สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์

Computer Interfacing Unit for X-ray Controller

โดย

นายเสริมชัย ขุนจันทร์

นายเอกภพ งามละเมียด

เลขหมู่.....
เลขทะเบียน 50373
วัน,เดือน,ปี 13 พ.ค. 2547

.b.....
.i.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์

Computer Interfacing Unit for X-ray Controller

โดย

นายเสริมชัย ขุนจันทร์ 43015241

นายเอกภพ งามละเมียด 43015245

อาจารย์ที่ปรึกษา

ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์

(Computer Interfacing Unit for X-ray Controller)

ผู้จัดทำ

1. นายเสริมชัย ขุนจันทร์ 43015241
2. นายเอกภพ งามละเมียด 43015245

.....อาจารย์ที่ปรึกษา

(ดร.ชูชาติ ปิณฑวิรุจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์

นายเสริมชัย ขุนจันทร์ 43015241
นายเอกภพ งามละเมียด 43015245
ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
ภาคเรียนที่ 1 ปีการศึกษา 2545

บทคัดย่อ

เครื่องเอ็กซเรย์ที่ใช้งานในปัจจุบันมีส่วนประกอบหลัก 3 ส่วน คือ ชุดควบคุมแรงดันที่ใช้ในการเอ็กซเรย์ ชุดควบคุมกระแสที่ไส้หลอดเอ็กซเรย์และชุดควบคุมการฉายรังสี ซึ่งทั้งหมดควบคุมการทำงานด้วยมือ

ปริญญานิพนธ์ฉบับนี้ได้ทำการศึกษาเกี่ยวกับการออกแบบและการสร้างหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์ ซึ่งใช้อุปกรณ์ FPGA ในการเชื่อมต่อการทำงานระหว่างหน่วยควบคุมการทำงานด้วยมือและเครื่องคอมพิวเตอร์ส่วนบุคคล ในการใช้งานสามารถควบคุมการเอ็กซเรย์โดยตรงผ่านทางหน่วยควบคุมโดยไม่ต้องเชื่อมต่อการทำงานเข้ากับเครื่องคอมพิวเตอร์ได้ และจากการทดลองใช้งาน ผลการทดลองที่ได้ตรงตามหน้าที่การทำงานต่างๆ ตามที่ได้ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Computer Interfacing Unit for X-ray Controller

Mr.Sermchai Khunjan 43015241

Mr.Akkapob Ngamlamiat 43015245

Ph.D.Chuchart Pintavirooj (Advisor)

Academic Year 2002

Abstract

Traditionally, the X-ray unit including High Voltage Controller, Tube-filament Controller, Exposure Controller and so on is controlled manually.

This thesis study the design and construction of X-ray computer-controlled unit. We use the FPGA chip to communicate between the manual-controlled X-ray unit and the Personal Computer. This FPGA chip is also designed to control the X-ray without the connection with Personal Computer. The operating test shows successfully results.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงงานหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์นี้ ประสบผลสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับคำแนะนำ ตลอดจนแนวคิดในการทำงานจากอาจารย์ที่ปรึกษาโครงงาน ดร.ชูชาติ ปิณฑวิรุจน์ และรศ.ดร. มนัส สังวรศิลป์ รวมทั้งอาจารย์ทุกท่านในภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ขอขอบคุณบริษัท คงศักดิ์เอ็กซเรย์ การแพทย์อุตสาหกรรม ที่อนุญาตให้ไปฝึกงานเพื่อศึกษาเกี่ยวกับการทำงานของเครื่องเอ็กซเรย์ รวมทั้งคุณบุญอนันต์ เกียงเอีย ที่ได้ให้ความรู้รวมไปถึงการแนะนำแนวทางในการแก้ปัญหาจากการทดลอง รวมถึงแนวทางในการปรับปรุงโครงงาน ตลอดจนผู้เกี่ยวข้องทั้งหมดที่ไม่ได้กล่าวถึงในที่นี้

ทางผู้จัดทำขอแสดงความขอบคุณทุกท่านอีกครั้ง ณ โอกาสนี้

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เทคโนโลยีของ FPGA	3
2.1.1 Field-Programmable Gate Array (FPGA)	3
2.1.2 โครงสร้างภายในของ FPGA	4
2.1.3 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว	5
2.1.4 การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์	5
2.1.5 เครื่องมือสำหรับการออกแบบ FPGA	9
2.2 เทคโนโลยีการออกแบบวงจรลอจิก (VHDL : Logic Design Technology)	9
2.2.1 องค์ประกอบที่สำคัญของ VHDL	9
2.2.2 รูปแบบของภาษา VHDL	10
2.2.7 ข้อดีของ VHDL	10
2.2.8 VHDL ช่วยในการออกแบบได้อย่างไรบ้าง	10
2.3 Delphi คืออะไร	11
2.3.1 การสร้างแอพพลิเคชันจากเคลไฟล์	11
2.4 พอร์ตอนุกรม (Serial Port)	11
2.4.1 Hardware Properties	12
2.4.2 Serial Pinouts (DB25 & DB9)	12
2.4.3 Pin Functions	15
2.4.4 Null Modems	16
2.4.5 Loopback Plugs	16
2.4.6 DTE/DCE Speeds	16
2.4.7 Flow Control	17
2.4.8 RS-232 Waveforms	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.9 RS-232 Level Converters 18
2.5 เอ็กซเรย์ (X-ray) 19
2.5.1 ประวัติการค้นพบเอ็กซเรย์ 19
2.5.2 ลักษณะและคุณสมบัติของเอ็กซเรย์ 19
2.5.3 การเกิดการเอ็กซเรย์ 20
2.5.4 โครงสร้างหลอดเอ็กซเรย์ 21
บทที่ 3 การออกแบบและการสร้าง 25
บทที่ 4 ผลการทดลอง 38
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 47
ภาคผนวก
บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 โครงสร้างภายในของ FPGA ตระกูล FLEX 10K 4

รูปที่ 2.2 การโปรแกรมข้อมูลลงในอุปกรณ์ FPGA 5

รูปที่ 2.3 แสดง D-Type 25 Pin 13

รูปที่ 2.4 แสดง D-Type 9 Pin 14

รูปที่ 2.5 Null modem wiring diagram. 16

รูปที่ 2.6 Loopback plug wiring diagram. 16

รูปที่ 2.7 TTL/CMOS Serial Logic Waveform. 17

รูปที่ 2.8 RS-232 Logic Waveform. 18

รูปที่ 2.9 แสดงลักษณะของ IC MAX-232 18

รูปที่ 2.10 แสดงให้เห็นถึงลักษณะและส่วนประกอบที่สำคัญของ Stationary anode tube 21

รูปที่ 2.11 แสดงให้เห็นถึงลักษณะและส่วนประกอบที่สำคัญๆของ Rotating anode tube 22

รูปที่ 2.12 แสดงให้เห็นถึง Line-focus principle และการเอียง Target เป็นมุม 17°-20° เพื่อทำให้ได้ Effective focal spot ขนาดเล็ก 23

รูปที่ 3.1 แสดงแผนผังของการเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์ทั่วไป 24

รูปที่ 3.2 แสดงการทำงานโดยรวมของหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์ 27

รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานภายในส่วนควบคุมด้วยชิป FPGA 28

รูปที่ 3.4 แสดงวงจรของชุดทริก SCR 31

รูปที่ 3.5 แสดงวงจรควบคุมการทำงานของรีเลย์ 32

รูปที่ 3.6 แสดงวงจรควบคุมการทำงานของบัซเซอร์ 32

รูปที่ 3.7 การวางคอมโพเนนต์ต่างๆ ลงบนฟอร์ม 33

รูปที่ 3.8 แสดง Flow Chart แสดงการทำงานของฟอร์ม 35

รูปที่ 3.9 แสดง Flow Chart แสดงการทำงานของฟอร์ม (ต่อ) 36

รูปที่ 3.10 แสดงผลของการคอมไพล์และรันโปรแกรมที่ได้ออกแบบไว้ 37

รูปที่ 4.1 แสดงส่วนแผงควบคุมการทำงาน 38

รูปที่ 4.2 แสดงส่วนควบคุมการทำงานโดยใช้คอมพิวเตอร์ 39

รูปที่ 4.3 แสดงผลการควบคุมจากคอมพิวเตอร์ 41

รูปที่ 4.4 แสดงการผลการควบคุมจากคีย์บอร์ด 42

รูปที่ 4.5 แสดงผลการทำงานของชุดทริก SCR เพื่อเลือกค่าแรงดัน ค่ากระแส และเทคนิคในการเอ็กซเรย์ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 แสดงสัญญาณที่ใช้ในการทริก SCR ที่ 0.02s 43
รูปที่ 4.7 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.02s 43
รูปที่ 4.8 แสดงสัญญาณที่ใช้ในการทริก SCR ที่ 0.03s 44
รูปที่ 4.9 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.03s 44
รูปที่ 4.10 แสดงสัญญาณที่ใช้ในการทริก SCR ที่ 0.04s 45
รูปที่ 4.11 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.04s 45
รูปที่ 4.12 แสดงสัญญาณที่ใช้ในการทริก SCR ที่ 0.07s 46
รูปที่ 4.13 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.07s 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 25 PIN D-SUB MALE at the computer. 13

ตารางที่ 2.2 9 PIN D-SUB MALE at the Computer. 14

ตารางที่ 2.3 D-Type 25 and D-Type 9 Connector 15

ตารางที่ 2.4 แสดงหน้าที่ของขาต่าง ๆ 15

ตารางที่ 3.1 แสดงการกำหนดค่าของพรอบเพอร์ตี้ให้กับคอมโพเนนต์ต่างๆ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันพบว่าความก้าวหน้าของเทคโนโลยีทางการแพทย์มีการพัฒนาไปอย่างรวดเร็ว ซึ่งก็ควบคู่ไปกับการพัฒนาทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ โดยอุปกรณ์เอ็กซเรย์ก็เป็นอุปกรณ์อีกชนิดหนึ่งที่มีความต้องการที่จะพัฒนาเพื่อที่จะสามารถทำการควบคุมการทำงานร่วมกับคอมพิวเตอร์ได้ซึ่งจะทำให้การใช้งานง่ายขึ้น จากเดิมที่หน่วยควบคุมเครื่องเอ็กซเรย์ที่ใช้กันโดยทั่วไปจะใช้การทำงานในเชิงกลไก คือจะใช้สวิตซ์ซีเลคเตอร์ในการเลือกค่าแรงดันที่ต้องการใช้ในการเอ็กซ์เรย์ ซึ่งหากต้องการที่จะพัฒนารูปแบบการควบคุมการทำงานให้สามารถใช้การควบคุมด้วยคอมพิวเตอร์ควบคุมได้ ก็จำเป็นต้องเปลี่ยนลักษณะการทำงานมาเป็นเชิงตัวเลขซึ่งจะง่ายในการออกแบบเพื่อใช้งานร่วมกับคอมพิวเตอร์ สำหรับอุปกรณ์ FPGA (Field Programmable Gate Array) ก็เป็นอุปกรณ์ชนิดหนึ่งที่ได้มีการพัฒนาขึ้นมาเพื่อช่วยให้ผู้ใช้งานสามารถสร้างวงจรดิจิตอลที่ต้องการลงในตัวอุปกรณ์ได้เอง ซึ่งทำให้มีความสะดวกในการออกแบบวงจรอย่างมาก อีกทั้งการออกแบบเพื่อใช้งานร่วมกับคอมพิวเตอร์ก็ยังทำได้ง่ายอีกด้วย ในการออกแบบการทำงานของวงจรภายในอุปกรณ์ FPGA จะใช้ภาษา VHDL เขียนบรรยายการทำงานของวงจรร่วมกับการใช้การออกแบบด้วยแผนภาพ (Schematic)

1.1 ความเป็นมาของโครงงาน

จากความต้องการที่จะพัฒนาหน่วยควบคุมเอ็กซเรย์ที่สามารถควบคุมการทำงานผ่านทางคอมพิวเตอร์ได้นั้นจำเป็นที่จะต้องปรับเปลี่ยนลักษณะการทำงานของวงจรจากเดิมที่หน่วยควบคุมการเอ็กซเรย์โดยทั่วไปจะใช้การทำงานในเชิงกลไก ให้เปลี่ยนมาเป็นการทำงานในเชิงตัวเลขซึ่งในที่นี้เราจะใช้อุปกรณ์ FPGA ในการสร้างวงจรการทำงานเชิงตัวเลขซึ่งจะทำให้การออกแบบการทำงานร่วมกับเครื่องคอมพิวเตอร์สามารถทำได้ง่ายขึ้น

1.2 วัตถุประสงค์

- เพื่อศึกษาการออกแบบและการสร้างชุดควบคุมแรงดัน, ส่วนตั้งเวลา, กระแสและเทคนิคในการเอ็กซเรย์ของเครื่องเอ็กซเรย์ผ่านทางคอมพิวเตอร์
- เพื่อศึกษาการใช้งานอุปกรณ์ FPGAในการสร้างชุดควบคุมการทำงานของเครื่องเอ็กซเรย์
- เพื่อศึกษาภาษา VHDL ที่ใช้ในการอธิบายและออกแบบวงจรดิจิตอล
- เพื่อศึกษาการใช้งานโปรแกรมเดลไฟล์ เพื่อใช้ในการสร้างชุดควบคุมและการส่งข้อมูลออกทางพอร์ตอนุกรมมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 แนวความคิดและโครงสร้างของงาน

การนำอุปกรณ์ FPGA มาประยุกต์ใช้งานเป็นหน่วยควบคุมเอ็กซเรย์ โดยหลักการนั้นจะนำค่าที่รับมาจากคีย์บอร์ดหรือค่าที่รับมาจากชุดควบคุมโดยคอมพิวเตอร์ที่ส่งข้อมูลผ่านมาทางพอร์ตอนุกรมมาตรฐาน RS-232 ซึ่งเป็นคำสั่งในการควบคุมการทำงานของเอ็กซเรย์ โดยค่าเอาท์พุทที่ได้จากหน่วยควบคุมนี้จะเป็นสัญญาณเชิงตัวเลขซึ่งจะนำไปควบคุมการทำงานในส่วนต่างๆ ของเครื่องเอ็กซ์เรย์

1.4 คุณสมบัติโดยทั่วๆ ไปของโครงงาน

- สามารถเลือกค่าแรงดันได้ 35 KVp ถึง 113 KVp
- สามารถเลือกเวลาที่ใช้ในการเอ็กซเรย์ได้ 0.02s ถึง 5s
- สามารถเลือกค่ากระแสได้ 3 ค่า คือ 50S, 50L, 100L
- สามารถเลือกเทคนิคการเอ็กซเรย์ได้ 2 รูปแบบ คือ Radio และ Bucky
- การส่งข้อมูลผ่านพอร์ตอนุกรม มาตรฐาน RS-232 และสามารถเลือกพอร์ตสื่อสารเป็นพอร์ต COM1 หรือ COM2 ได้
- ใช้อัตราบอดเรตเท่ากับ 4800 บิทต่อวินาที

1.5 การนำเสนอโครงงาน

จะมีการนำเสนอรายละเอียดของโครงงานโดยสมบูรณ์ ตามหัวข้อต่อไปนี้

- บทที่ 1 บทนำ กล่าวถึงตัวโครงงานและคุณสมบัติการทำงานพื้นฐาน
- บทที่ 2 กล่าวถึงทฤษฎีที่เกี่ยวข้องกับการสร้างหน่วยควบคุมเอ็กซเรย์ ซึ่งประกอบด้วยรายละเอียดของอุปกรณ์ FPGA, การเขียนโปรแกรมบรรยายเชิงพฤติกรรมด้วยภาษา VHDL, การสร้างแอพพลิเคชันจากเคลไฟล์, การส่งข้อมูลผ่านพอร์ตอนุกรมมาตรฐาน RS-232 และทฤษฎีที่เกี่ยวข้องเช่นการเกิดรังสีเอ็กซเรย์ คุณสมบัติของรังสีเอ็กซเรย์
- บทที่ 3 กล่าวถึงรายละเอียดเกี่ยวกับการออกแบบหน่วยควบคุมเอ็กซเรย์ทั้งในส่วนที่ควบคุมผ่านทางคอมพิวเตอร์ที่เขียนขึ้นโดยโปรแกรมเคลไฟล์และส่วนที่ควบคุมการทำงานผ่านทางคีย์บอร์ดรวมถึงการสร้างการทำงานของวงจรขึ้นภายในอุปกรณ์ FPGA โดยการเขียนโปรแกรมบรรยายเชิงพฤติกรรมด้วยภาษา VHDL
- บทที่ 4 ผลการทดลอง
- บทที่ 5 สรุปและวิจารณ์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 เทคโนโลยีของ FPGA

2.1.1 Field-Programmable Gate Array (FPGA)

FPGA เป็นอุปกรณ์ในลักษณะของ Programmable Device ที่เราสามารถโปรแกรมตัวมันให้เป็นวงจรดิจิตอลใดๆก็ได้ สำหรับ FPGA แล้วนับว่าเป็นอุปกรณ์ตัวใหม่ในตระกูลของ ASIC ซึ่งมีการเจริญเติบโตอย่าง รวดเร็วและมีบทบาทที่สำคัญในการเข้ามาแทนที่ระบบอิเล็กทรอนิกส์ที่ใช้ TTL โครงสร้าง ภายในของ FPGA ประกอบไปด้วยอะเรย์ของลอจิกเกทต่างๆมากมาย ซึ่งในปัจจุบันความจุเกทภายในตัวชิพ FPGA ได้เพิ่มขึ้น จากระดับไม่กี่พันตัวจนถึงระดับล้านตัวซึ่งสามารถรองรับวงจรดิจิตอลที่มีความสลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการออกแบบพัฒนาและทดสอบก็ทำได้ง่ายซึ่งในปัจจุบัน การออกแบบวงจรโดยใช้ FPGA กำลังเป็นที่นิยมและมีแนวโน้มที่จะนำมาใช้งานมากขึ้นเรื่อยๆ ในปัจจุบันมี FPGA อยู่ 4 ชนิดที่วางขายอยู่ในท้องตลาด ได้แก่ Symmetrical Array, Row-Based, Hierarchical PLD และ Sea-of-Gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการโปรแกรมที่แตกต่างกันไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรม ซึ่งมีอยู่ 2 แบบคือ การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรม โดยการใช้หน่วยความจำ

1. การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

1.1 Fuse เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียวเท่านั้น ซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน

1.2 Anti Fuse เป็นวิธีการโปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรม แล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

2. การโปรแกรมโดยใช้หน่วยความจำ

2.1 EEPROM Based FPGA

FPGA ที่ใช้การโปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกทต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกทแต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิท ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

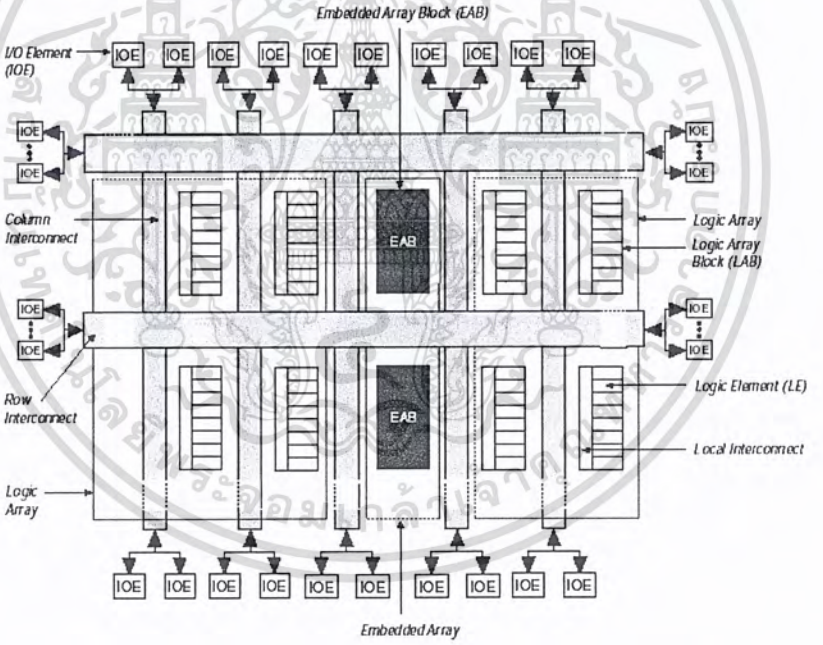
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 SRAM Based FPGA

FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถ โปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกทในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกท) ซึ่งข้อดีของ SRAM Based FPGA คือใช้เวลาในการ โปรแกรมน้อย (ระดับ nSec) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และ เหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมใน ภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิพในขณะที่เริ่มต้นใช้งาน

2.1.2 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอะเรย์ของบล็อกลอจิกที่สามารถทำการโปรแกรมได้ดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างภายในของ FPGA ตระกูล ACEX 1K

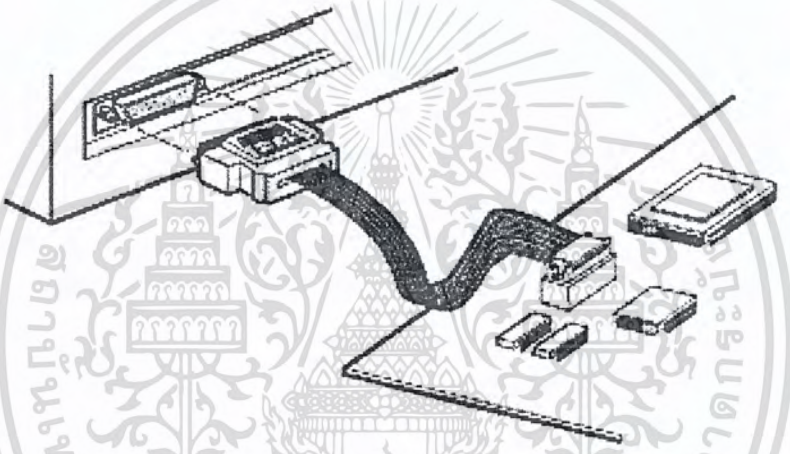
2.1.3 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว

1. ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพ เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึง ภาษา Assembly ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจร หรือ HDL (Hardware Description Language) เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and Optimize ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตราฐานเดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของ คอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก ดังรูปที่ 2.2 และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด



รูปที่ 2.2 การโปรแกรมลงในชิพ

2.1.4 การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์

ในการออกแบบวงจรดิจิตอลนั้นสามารถทำได้โดยการวาดวงจร (Schematic) หรือใช้ภาษาอธิบายพฤติกรรม (Hardware Description Language) ของฮาร์ดแวร์ ในกรณีของการออกแบบวงจรด้วย ASIC ชนิด Full Custom ผู้ออกแบบจะต้องเขียนวงจรด้วย Schematic จากนั้นจะนำวงจรที่ออกแบบไว้ไปทำการจำลองการทำงาน (Simulate) ซึ่งหากผลออกมาเป็นที่พอใจก็จะต้อง Layout เป็นชั้นสารและในการออกแบบ ASIC ชนิดนี้ผู้ออกแบบจำเป็นจะต้องทราบถึงเทคโนโลยีที่ใช้ในการสร้างด้วย หลังจากได้ Layout ที่สมบูรณ์แล้วจึงจะส่งไปเข้ากระบวนการสร้างไอซีหรือ Fabrication เพื่อสร้างเป็นชิพไอซีออกมาแต่ในการออกแบบวงจรด้วย FPGA โดยการใช้ Schematic หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำได้สะดวกกว่าเนื่องจากวิธีการนี้ผู้ออกแบบไม่จำเป็นต้องคำนึงถึงเทคโนโลยีที่จะใช้สร้างไอซีและที่สำคัญการออกแบบโดยวิธีนี้สามารถแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่าเพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยีสำหรับภาษาที่ใช้สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL, AHDL และ Verilog เป็นต้น ส่วนรายละเอียด ของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

1. การสังเคราะห์วงจร(Logic Synthesis)

ในขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์พฤติกรรม ของวงจรที่ได้จากการออกแบบด้วย Schematic หรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์ นั้นสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการหรือไม่ ตัวอย่างเช่น FPGA ของบริษัท XILINX และบริษัท ALTERA จะมีซอฟต์แวร์หลายตัวที่สามารถใช้ได้ เช่น Max Plus II ในขั้นตอนนี้ ซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL และทำการ Optimize เพื่อให้ได้วงจรตาม เทคโนโลยีที่เลือกใช้ในการสังเคราะห์วงจรนั้นวงจรระดับเกท (Gate Level) จะไม่เหมาะสมกับโครงสร้างที่มีอยู่ในอุปกรณ์ FPGA ดังนั้นในการ Optimize ซอฟต์แวร์สังเคราะห์วงจร จะต้องทำการ Optimize ให้ได้เป็นวงจรที่ประกอบ ด้วยกลุ่มของลอจิกที่เหมาะสมกับอุปกรณ์ FPGA นั้นๆจึงทำให้ผล ที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับ โมเดล แต่ละตัวได้ เช่น ข้อบังคับในเรื่องเวลา (Timing Constraints) หรือข้อบังคับในเรื่องของพื้นที่ (Area) หรือกำหนดชนิดและตำแหน่งของ I/O ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจร ที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการ Optimize คือการเทียบ (Mapping) โมเดลให้เข้ากับ เทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในอุปกรณ์ FPGA เมื่อทำ การสังเคราะห์วงจรเสร็จแล้ว ซอฟแวร์การสังเคราะห์วงจรก็จะมีการรายงานผลว่าโมเดลที่ออกแบบไปนั้น เป็นอย่างไร เช่น มีค่าความหน่วง (Delay) เท่าใด ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตาม ที่กำหนด

2. การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำโดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกท (gate), ฟลิปฟลอบ (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGAหลังจากทำขั้นตอนนี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงภายในแต่ละส่วนเท่านั้นหรือที่เรียกว่าความหน่วงลอจิก (logic delay) ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์จะรวมเอาซอฟต์แวร์ย่อยอื่นๆ อีก เพื่อให้การทำ Partitioning Placement & Routing เป็นไปอย่างต่อเนื่อง

3. การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควร จะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กัน เพื่อจะได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆให้อยู่ใกล้กัน โดยเฉพาะส่วน ที่มีการเชื่อมต่อสัญญาณด้วยกัน นอกจากนั้นการกำหนดตำแหน่งขา I/O (I/O pin) ตามตำแหน่งขา I/O ของ FPGA บนแผ่น PCB ก็จะมีผลโดยตรงเลยคือซอฟต์แวร์จะวาง I/O ลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่งบางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา I/O ควรกำหนดตำแหน่งให้เหมาะสมหรือไม่ก็ให้ซอฟต์แวร์จัดการเอง

4. การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA ขั้นตอนนี้จะทำ ต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมด (เนื่องจากจำนวนทรัพยากรสำหรับเชื่อมต่อสัญญาณนั้นมีอยู่จำกัด) หรือเกิดความหน่วงเกิน ค่าที่กำหนดในข้อบังคับ ผู้ออกแบบสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟท์แวร์หรือผู้ออกแบบจะทำการ เชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับ ทางเวลา จะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

5. ความหน่วงด้านเวลา (Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วงที่เกิดขึ้นนั้นแยกได้เป็นสองประเภทคือความหน่วงลอจิก (Logic delay) เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เองและความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ (Routing delay) เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA โดยปกติแล้วค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณมักจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้นและเพื่อให้ได้ผลลัพธ์ที่ดีขึ้นค่าความหน่วงที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการวางอุปกรณ์และเชื่อมต่อสัญญาณแล้วจะมีค่าความหน่วงที่ค่อนข้างแน่นอนซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้น เป็นไปตามข้อกำหนดหรือไม่

6. การจำลองการทำงานของวงจร (Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่งเพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการทำงานของโมเดลว่าถูกต้องหรือไม่มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะมีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจรควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้วเพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดลเกิดขึ้นตอนไหนจะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้เกิดข้อผิดพลาด นั่นคือการทำการจำลองการทำงานของวงจรต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจรและการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้นผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจสอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตาม ข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์ การเชื่อมต่อสัญญาณ (post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลยซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้อง ตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงานของวงจรควรใช้ ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่างๆ

7. การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็นข้อมูลวงจร (configuration data) ซึ่งอยู่ในรูปของบิตสตรีม (bit stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับอุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้นในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ serial PROM ด้วยเพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานโมเดลครั้งต่อไปไม่ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีกเพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วแต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยใช้วิธี EPROM หรือ Anti fuse ก็ไม่จำเป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจรเพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูล วงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

2.1.5 เครื่องมือสำหรับการออกแบบ FPGA

จะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้เวลาน้อยกว่ามากด้วย ส่วนสำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่เขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงใน อุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็นซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้สำหรับซอฟต์แวร์ที่ใช้ทำการจำลองการทำงานของวงจรนั้นต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบ เพราะโมเดลที่ได้จากการทำขั้นตอนต่างๆ ด้วยซอฟต์แวร์ ต่างๆ ต้องเอามาจำลองการทำงานได้และในการจำลองการทำงานของวงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์คสเตชั่น (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ ที่ใช้บนพีซี (PC) มากขึ้นซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

2.2 เทคโนโลยีการออกแบบวงจรลอจิก (VHDL : Logic Design Technology)

เทคโนโลยีการออกแบบวงจรลอจิกได้มีการพัฒนาอยู่เสมอและได้มีการพัฒนาโปรแกรมที่ใช้ในการออกแบบวงจรลอจิกขึ้นมา VHDL เป็นหนึ่งในนั้นที่ได้มีการพัฒนาขึ้นมา VHDL ย่อมาจาก VHSIC Hardware Description Language (VHSIC = Very High Speed Integrated Circuit) ซึ่งเป็นภาษาที่ใช้ในการออกแบบวงจรลอจิกเพื่อผลิตเป็นชิพไอซีขึ้นมา VHDL เป็นภาษาระดับสูงที่ใช้ในการออกแบบระบบและวงจรลอจิก ตัวภาษานั้นสามารถที่จะออกแบบระบบโดยไม่คำนึงกระบวนการหรือวิธีการของวงจรเพราะวงจรจะถูกสร้างในรูปแบบของฟังก์ชั่น เราจะพิจารณาเพียงจุดมุ่งหมายของการออกแบบวงจรเท่านั้น ภาษา VHDL เป็นภาษามาตรฐานที่ IEEE ได้ให้การยอมรับ ข้อดีคือเป็นภาษาที่สามารถปรับเปลี่ยนโครงสร้างให้เข้ากับระบบฮาร์ดแวร์ต่างๆได้เป็นอย่างดีและอ่านเข้าใจได้ง่าย

2.2.1 องค์ประกอบที่สำคัญของ VHDL

VHDL มีองค์ประกอบที่สำคัญอยู่ 2 ส่วนคือ Entity และ Architecture

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

2. Architecture ใช้ในการอธิบายโครงสร้างภายในของ Entity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 รูปแบบของภาษา VHDL

การเขียน Architecture ของ VHDL มีรูปแบบการเขียน 3 รูปแบบ

1. Structural Model การอธิบายวงจรโดยใช้โมดูลต่างๆ มาเชื่อมต่อกันให้เป็นโครงสร้างภายใน
2. Behavioral Model การอธิบายการทำงานของวงจรในระดับลอจิกเกท
3. Sequential Model การอธิบายขั้นตอนการทำงานโดยใช้ Sequential Statement ซึ่งจะทำงานแบบเป็นขั้นตอน

สำหรับรูปแบบการเขียนที่เราจะเลือกใช้นั้นขึ้นอยู่กับความเหมาะสมของฮาร์ดแวร์เราสามารถที่จะใช้รูปแบบต่างๆมาร่วมกันได้อยู่ใน Architecture ตัวเดียวกัน ซึ่งขึ้นอยู่กับความเหมาะสมและวิธีการออกแบบ แต่ทั้ง 3 รูปแบบก็ให้ผลลัพธ์เหมือนกัน

2.2.3 ข้อดีของ VHDL

1. รูปแบบของภาษาที่เข้าใจได้ง่าย
2. มีโครงสร้างของภาษาที่สามารถปรับเปลี่ยนให้เข้ากับ hardware ได้ง่าย
3. มี statement ให้ใช้งานอยู่หลายตัว
4. สามารถออกแบบในรูปแบบ top-down design ได้
5. มี library ต่างๆให้เลือกใช้งานได้ทำให้ออกแบบได้ง่ายขึ้น เป็นประโยชน์ต่อ programmer

2.2.4 VHDL ช่วยในการออกแบบได้อย่างไรบ้าง

1. การออกแบบวงจรสามารถทำได้โดยง่าย
2. เมื่อมีการแก้ไข หรือ เปลี่ยนแปลงวงจรสามารถทำได้โดยง่าย เพียงโปรแกรมวงจรใหม่ลงไป ไม่ต้องมีการเปลี่ยนแปลงตัว hardware
3. การทดสอบไม่จำเป็นต้องทดลองกับวงจรจริง สามารถใช้การ simulate ทดลองดูผลลัพธ์ของวงจรที่เราออกแบบได้ว่าถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Delphi คืออะไร

Delphi เป็นเครื่องมือสำหรับสร้างแอพพลิเคชันสำหรับรันบน Windows 95/98/2000 ที่ผลิตโดยบริษัท Inprise (ชื่อเดิมก็คือ Borland) ซึ่งเป็นบริษัทที่แวดวงของนักพัฒนาแอพพลิเคชันรู้จักและยอมรับในตัวผลิตภัณฑ์เป็นอย่างดี

Delphi นั้นเป็นเครื่องมือพัฒนาแอพพลิเคชันแบบ Visual Programming (เหมือนกับ Visual Basic, Visual C++, PowerBuilder ฯลฯ) ซึ่งทำให้เราสามารถเห็นผลลัพธ์การทำงานไปพร้อมๆ กับการลงมือสร้างแอพพลิเคชัน

จุดเด่นที่สำคัญมากของความเป็น Visual Programming คือช่วยลดเวลาของการสร้างแอพพลิเคชันนั่นเพราะแทนที่เราจะไปทุ่มเวลาไปปรับแต่งส่วนติดต่อผู้ใช้หรืองานที่ไม่จำเป็นหรืองานซ้ำซาก เราก็มอบภาระเหล่านี้ให้ Delphi เสีย, สำหรับเราก็มุ่งเข้าไปแก้ไขปัญหาที่เป็นหัวใจการทำงานของแอพพลิเคชันดีกว่า

2.3.1 การสร้างแอพพลิเคชันจากเดลไฟล์

แอพพลิเคชันที่สร้างจากเดลไฟล์นั้นมีวิธีการสร้างที่แตกต่างจากการเขียนโปรแกรมแบบเดิมๆ ที่เราเคยเรียนรู้มาก่อนหากเคยเขียนโปรแกรมจากภาษา Basic, C หรือ Pascal ก็จะเห็นว่ามีแนวคิดที่ข้อนข้างต่างกัน ซึ่งเราจะเรียกแอพพลิเคชันที่สร้างจากเดลไฟล์ว่า Event-Driven (แปลว่าเหตุการณ์พาไป)

Event-Driven ที่จริงก็คือการเขียนโปรแกรมในลักษณะที่ว่า “ถ้ามีเหตุการณ์เกิดขึ้น เราจะจัดการกับมันอย่างไร” เช่นถ้าผู้ใช้คลิกปุ่ม Exit เราจะทำอย่างไร? เราอาจจะถามผู้ใช้ว่า แน่ใจแล้วนะว่าจะจบการทำงานของโปรแกรม ถ้าผู้ใช้ยืนยันก็จบไป แต่ถ้าไม่ก็ให้แแอพพลิเคชันทำงานต่อไป เป็นต้น

2.4 พอร์ตอนุกรม (Serial Port)

การ Interface กับ Serial Port จะยากกว่าการ Interface กับ Parallel Port การสื่อสารของอุปกรณ์ที่ต่อกับ Serial Port จะถูกเปลี่ยน (Convert) เป็น สัญญาณแบบ Parallel แล้วจึงนำไป Process ต่อ ซึ่งจะใช้ Universal Asynchronous Receiver / Transmitter (UART) เป็นตัวทำหน้าที่นี้ ส่วนทางด้าน Software ก็มี Register ที่ต้องจัดการมากกว่า Standard Parallel Port (SPP) อีกหลายตัว Serial Port มีข้อดีกว่า Parallel Port ดังนี้

1. ระยะของสาย Serial สามารถมีความยาวได้มากกว่าสายของ Parallel มาก ทั้งนี้เพราะสัญญาณของ Serial Port ซึ่งส่วนใหญ่ใช้มาตรฐาน RS-232C จะมีค่า –3Volt ถึง –25Volt สำหรับ Logic ”1” หรือ “Mark” และมีค่า +3Volt ถึง +25Volt สำหรับ Logic “0” หรือ “Space” (สำหรับช่วง +3Volt ถึงถึง -3Volt เป็นช่วง Undefined) ส่วนสัญญาณของ Parallel นั้น Logic “1” จะมีค่า +5Volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ logic “0” จะมีค่า 0Volt ทำให้สัญญาณของ Serial สามารถรับการสูญเสียของสาย (Cable loss) ได้มากกว่าสัญญาณของ Parallel ปรกติสาย Parallel Port จะไปได้เพียง 5 ฟุต ส่วนสาย RS-232 จะไปได้ถึง 50 ฟุตที่ speed สูงสุดของมัน

2. Serial จะใช้จำนวนสายไฟน้อยกว่า Parallel ถ้าต่อในลักษณะ Null Modem จะใช้สายเพียง 3 เส้น ขณะที่แบบ Parallel จะต้องใช้สาย 19 ถึง 25 เส้น

3. การสื่อสารแบบไร้สายเช่นการใช้ Infra Red การส่งพร้อมกันทีละ 8 Bit แบบ Parallel จะทำให้ไม่สามารถแยกแยะได้ว่า Bit ใดเป็น Bit0 หรือ Bit1 … เป็นต้น ปัจจุบันอุปกรณ์ IrDA มีความเร็วไม่ต่ำกว่า 115.2K Baud แต่มี Pulse length เพียง 3/16 ของ RS-232 เพื่อประหยัดพลังงานเพราะส่วนมากใช้ในอุปกรณ์แบบพกพาเช่น Laptop หรือ Palmtop

4. ปัจจุบัน Microcontroller มักมีการผนวก Port การสื่อสารแบบ Serial ไว้ด้วย เพราะใช้จำนวนขาน้อยกว่าแบบ Parallel

2.4.1 Hardware Properties

อุปกรณ์สื่อสารจะจำแนกออกเป็น 2 กลุ่มคือ

1. อุปกรณ์ Data Communication Equipment (DCE)
2. อุปกรณ์ Data Terminal Equipment (DTE)

DEC คืออุปกรณ์ที่ทำหน้าที่สื่อสารเช่น Modem หรือ โทรศัพท์หรือ Fax เป็นต้น ส่วน DTE คืออุปกรณ์ที่ต่อกับ DCE เพื่อใช้รับและ/หรือส่งสาร ซึ่งก็คือเครื่อง Computer หรือ Terminal โดยใช้ UART เป็นตัว Interface นั่นเอง คุณสมบัติทางไฟฟ้าของ Serial Port ตามมาตรฐาน RS-232C ของ Electronic Industry Association (EIA) พอสรุปได้ดังนี้

1. Logic “0” หรือ “Space” มีค่า +3Volt ถึง +25Volt
2. Logic “1” หรือ “Mark” มีค่า -3Volt ถึง -25Volt
3. ช่วง +3Volt ถึง –3Volt เป็นช่วง Undefined
4. Open circuit voltage เมื่อเทียบกับ GND ต้องไม่เกิน 25Volt
5. Short circuit current ต้องไม่เกิน 500mA ซึ่ง Driver ต้องสามารถรองรับได้

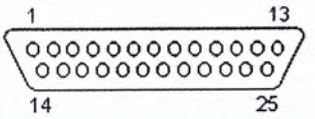
RS-232C กำหนด Baud rate ไว้ไม่เกิน 20K Baud ปัจจุบันได้แก้ไขให้รองรับกับ Technology ใหม่ได้จึงมีการปรับปรุงถึง RS-232E ซึ่งมีรายละเอียดอีกหลายอย่างถ้าสนใจขอให้ศึกษาจากเอกสาร RS-232E

2.4.2 Serial Pinouts (DB25 & DB9)

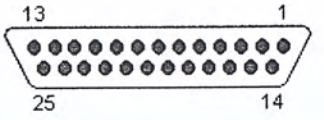
Serial Port มีขั้วต่อ 2 แบบคือ แบบ D-Type 25 Pin และแบบ D-Type 9 Pin ซึ่งทั้ง 2 แบบจะเป็นชนิดตัวผู้ทางด้านของ Computer ดังนั้นอุปกรณ์ที่จะนำมาต่อกับ Computer จึงต้องใช้ขั้วต่อชนิดตัวเมีย ตารางที่ 2.1 ข้างล่างแสดง ชื่อสัญญาณของขาต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D-Type 25 Pin



(Male at the computer side)



(Female at the cable side)

รูปที่ 2.3 แสดง D-Type 25 Pin

Pin	Name	Dir	Description
1	SHIELD	-	Shield Ground
2	TXD	→	Transmit Data
3	RXD	←	Receive Data
4	RTS	→	Request to Send
5	CTS	←	Clear to Send
6	DSR	←	Data Set Ready
7	GND	-	System Ground
8	CD	←	Carrier Detect
9	n/c	-	
10	n/c	-	
11	n/c	-	
12	n/c	-	
13	n/c	-	
14	n/c	-	
15	n/c	-	
16	n/c	-	
17	n/c	-	
18	n/c	-	
19	n/c	-	
20	DTR	→	Data Terminal Ready
21	n/c	-	
22	RI	←	Ring Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

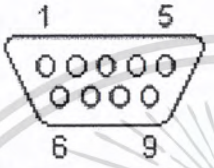
23	n/c	-	
24	n/c	-	
25	n/c	-	

ตารางที่ 2.1 25 PIN D-SUB MALE at the computer.

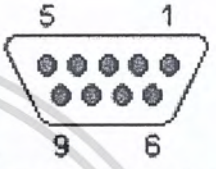
Note: Direction is DTE (Computer) relative DCE (Modem).

Note: Do not connect SHIELD(1) to GND(7).

D-Type 9 Pin



(Male at the computer side)



(Female at the cable side)

รูปที่ 2.4 แสดง D-Type 9 Pin

Pin	Name	RS232	V.24	Dir	Description
1	CD	CF	109	←	Carrier Detect
2	RXD	BB	104	←	Receive Data
3	TXD	BA	103	→	Transmit Data
4	DTR	CD	108.2	→	Data Terminal Ready
5	GND	AB	102	—	System Ground
6	DSR	CC	107	←	Data Set Ready
7	RTS	CA	105	→	Request to Send
8	CTS	CB	106	←	Clear to Send
9	RI	CE	125	←	Ring Indicator

ตารางที่ 2.2 9 PIN D-SUB MALE at the Computer.

Note: Direction is DTE (Computer) relative DCE (Modem).

Note: RS232 column is RS232 circuit name.

Note: V.24 column is ITU-TSS V.24 circuit name.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin No. (D-Type 25)	Pin No. (D-Type 9)	Abbreviation	Full Name
2	3	TD	Transmit Data
3	2	RD	Receive Data
4	7	RTS	Ready To Send
5	8	CTS	Clear To Send
6	6	DSR	Data Set Ready
7	5	SG	Signal Ground
8	1	(D)CD	(Data) Carrier Detect
20	4	DTR	Data Terminal Ready
22	9	RI	Ring Indicator

ตารางที่ 2.3 D-Type 25 and D-Type 9 Connector

2.4.3 Pin Functions

ตารางข้างล่างนี้แสดงถึงหน้าที่ของขาต่าง ๆ

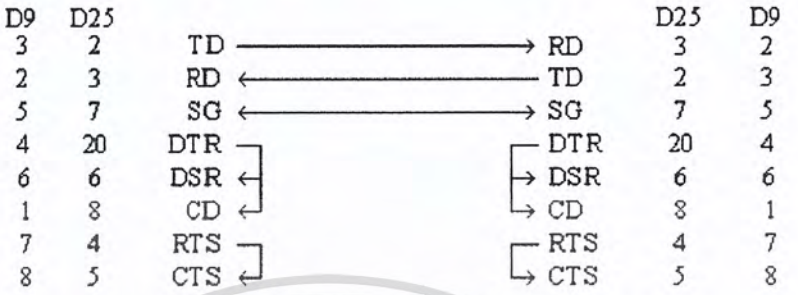
Abbreviation	Full Name	Originator	Function
TD	Transmit Data	DTE	Serial data output (TXD) from DTE.
RD	Receive Data	DCE	Serial data input (RXD) to DTE.
CTS	Clear To Send	DCE	Tell DTE that DCE is ready to exchange data.
(D)CD	(Data) Carrier Detect	DCE	Carrier from remote DCE is detected.
DSR	Data Set Ready	DCE	Tell DTE that DCE is ready to establish a link.
DTR	Data Terminal Ready	DTE	Tell DCE that DTE is ready to establish a link.
RTS	Ready To Send	DTE	Tell DCE that DTE is ready to exchange data.
RI	Ring Indicator	DCE	Ringing signal from the phone line is detected.

ตารางที่ 2.4 แสดงหน้าที่ของขาต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 Null Modems

Null Modem ใช้สำหรับเชื่อมโยงระหว่าง DTE 2 ตัวเข้าด้วยกันโดยตรงซึ่งโดยมากใช้ในการถ่ายข้อมูลระหว่าง Computer หรือใช้ในการพัฒนาระบบ Microprocessor หรือ Microcontroller ต่าง ๆ



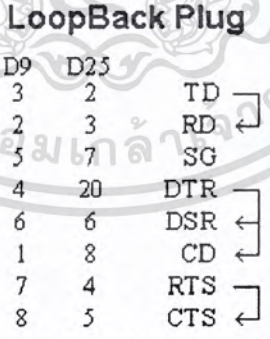
รูปที่ 2.5 Null modem wiring diagram.

จะพบว่าใช้สายเพียง 3 เส้นคือ TD, RD และ SG และมีการต่อ Jump ที่หัวต่อแต่ละข้างอีกเล็กน้อยเท่านั้น เพื่อหลอกให้ Computer คิดว่ามันกำลังคุยกับ DCE อยู่

หมายเหตุ TD ของ Computer เครื่องหนึ่งจะต้องต่อกับ RD ของ Computer อีกเครื่องหนึ่ง

2.4.5 Loopback Plugs

สำหรับ Plug ตัวนี้มีวัตถุประสงค์เพื่อช่วยในการเขียน Program ที่ติดต่อกับ Serial Port แต่อาจจะไม่สามารถใช้กับ Program ที่ใช้วิเคราะห์หาจุดเสีย (Diagnostic program) เพราะแต่ละ Program อาจจะออกแบบการเชื่อมต่อที่แตกต่างกันไป



รูปที่ 2.6 Loopback plug wiring diagram.

2.4.6 DTE/DCE Speeds

ความเร็วของการติดต่อระหว่าง DTE กับ DCE หรือระหว่าง Computer กับ Modem ซึ่งมักจะเรียกว่า Terminal speed และความเร็วระหว่าง DCE กับ DCE หรือระหว่าง Modem กับ Modem ซึ่งมักจะเรียกว่า Line speed ปรกติ Terminal speed มักจะเร็วกว่า Line speed เช่น Modem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

มี speed 56K baud ขณะที่ติดต่อกับ Computer ด้วย speed 115.2Kbps เป็นต้น ซึ่ง COM Port ของ Computer ต้อง set ให้เป็น 115.2Kbps ด้วย ปัจจุบัน UART เบอร์ 16550A จะรองรับ speed ได้ 115.2Kbps และเบอร์ 16C650 จะรองรับ speed ได้ 230.4Kbps

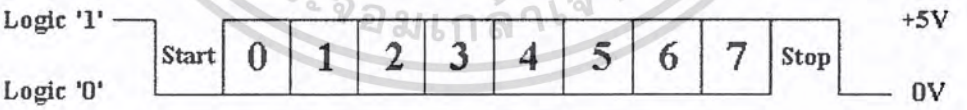
2.4.7 Flow Control

การติดต่อระหว่าง DTE และ DCE จะมีการควบคุมการไหลของข้อมูลเพื่อไม่ให้เกิดการ Overflow ขึ้นได้ ซึ่งมีอยู่ 2 แบบคือ Hardware flow control และ Software flow control

สำหรับ Software flow control มักจะเรียกว่า Xon/Xoff flow control ซึ่งใช้รหัส ASCII 17 เป็นสัญญาณ Xon และใช้รหัส ASCII 19 เป็นสัญญาณ Xoff หลักการทำงานก็ง่าย ๆ คือ Modem จะมี Buffer อยู่ เมื่อ Modem รับข้อมูลจาก Computer จน Buffer ใกล้จะเต็มมันก็จะส่งสัญญาณ Xoff ไปให้ Computer เพื่อให้ Computer หยุดส่งข้อมูลให้มันชั่วคราวและเมื่อ Buffer มีที่ว่างถึงระดับหนึ่ง Modem ก็จะส่งสัญญาณ Xon ไปให้ Computer เพื่อให้ Computer ส่งข้อมูลให้มันต่อการควบคุมโดยวิธีนี้ประหยัดสายสัญญาณเพราะรับ-ส่งผ่าน TD และ RD แต่อาจทำให้การสื่อสารช้าลงอย่างเห็นได้ชัดในกรณีที่ใช้กับการสื่อสารที่มี speed ต่ำเพราะแต่ละตัวอักษร ASCII ที่รับ-ส่งจะมีขนาด 10 Bit ส่วน Hardware flow control มักจะเรียกว่า RTS/CTS flow control จะใช้สายสัญญาณของ Serial Port ในการควบคุมทำให้ไม่บั่นทอนความเร็วของข้อมูล หลักการทำงานคือ เมื่อ Modem มีที่ว่างเพื่อรับข้อมูล มันก็จะส่งสัญญาณ CTS ไปให้ Computer และเมื่อมันใกล้จะเต็ม มันก็จะหยุดส่งสัญญาณ CTS ไปให้ Computer

2.4.8 RS-232 Waveforms

การสื่อสารโดย RS-232 เป็นการสื่อสารแบบ asynchronous หมายความว่าสัญญาณ clock ที่ใช้ควบคุมจังหวะไม่ได้ส่งไปพร้อมกับ Data แต่จะใช้ start bit เป็นตัว sync. ในแต่ละ word ของการสื่อสารและใช้สัญญาณ clock ภายในของแต่ละด้านเป็นตัวให้จังหวะเอง



รูปที่ 2.7 TTL/CMOS Serial Logic Waveform.

รูปที่ 2.18 แสดงลักษณะของสัญญาณจาก UART เมื่อใช้ format แบบ 8N1 คือ 8 data bits ไม่มี parity bit และมี 1 stop bit ขณะที่ idle จะอยู่ในสถานะ “Mark” หรือ logic “1” การส่งจะเริ่มจากการส่ง start bit คือ logic “0” และตามด้วย LSB bit จนหมด data bits และถ้ามี parity bit ก็จะส่งที่จุดนี้แล้วลงท้ายด้วย stop bit ซึ่งมีค่าเป็น logic “1” ในรูปได้แสดง bit ที่ต่อถัดจาก stop bit ซึ่งมีค่าเป็น logic “0” หมายความว่าเป็น start bit ของการส่ง word ถัดไปแต่ถ้ายังไม่มีการส่ง word ถัดไป ก็ต้องอยู่ในสถานะของ logic “1” ซึ่งเป็นสถานะของ idle และถ้าสายอยู่ในสถานะของ logic “0” นาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

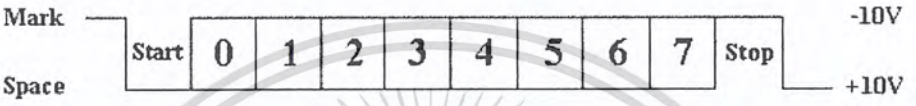
50373

กว่าเวลาของการส่ง 1 full word ระบบจะถือว่าเป็นสัญญาณ “Break” เพื่อหยุดการสื่อสาร ดังนั้นต้องไม่ลืมที่จะสั่งในสายกลับสู่สถานะ idle เมื่อสิ้นสุดการส่ง

การรับ-ส่งข้อมูลในลักษณะนี้เรียกว่าแบบ Frame คือมีกรอบปิดล้อมข้อมูลไว้ด้วย start bit และ stop bit

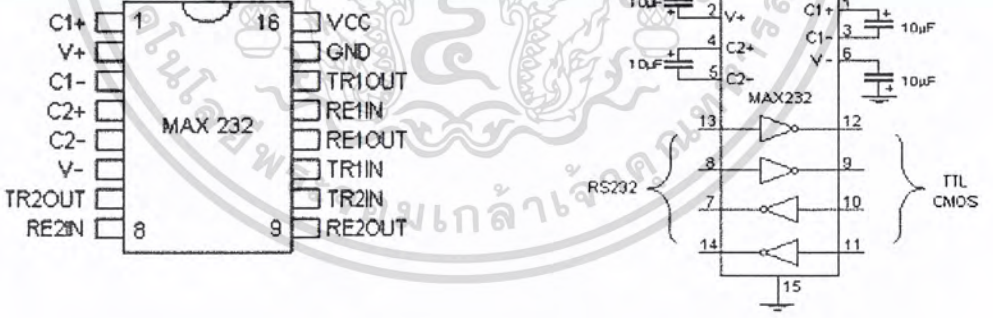
2.4.9 RS-232 Level Converters

สัญญาณ RS-232 มีค่าแรงไฟต่างจากที่ใช้ใน UART ดังแสดงใน Figure 5. ดังนั้นจึงต้องมี Converter เพื่อแปลงระดับสัญญาณให้เหมาะสมก่อนที่จะเชื่อมต่อกับ serial port หรือ RS-232 port ของ computer



รูปที่ 2.8 RS-232 Logic Waveform.

สำหรับสัญญาณ RS-232 นั้น logic “0” จะมีค่า +3 V ถึง +25 V และ logic “1” จะมีค่า –3 V ถึง –25 V ส่วนค่าระหว่าง –3 V ถึง +3 V เป็นค่า undefined ระดับสัญญาณนี้ใช้กับทุกสัญญาณไม่ใช่เฉพาะสัญญาณรับ-ส่งข้อมูลเท่านั้นแต่ยังรวมถึงสัญญาณควบคุมต่าง ๆ เช่น DTR, RTS, CTS, DCD, DSR เป็นต้น



รูปที่2.9 แสดงลักษณะของ IC MAX-232

ส่วนการที่เราจะนำข้อมูลมาใช้งานก็ต้องแปลงเป็น parallel ก่อนซึ่งเป็นหน้าที่ของ UART ซึ่งปัจจุบัน microprocessor และ microcontroller มักจะมี serial communication interface (SCI) อยู่ในตัว แต่อาจจะมีงานบางอย่างที่ไม่ได้ใช้ microcontroller และต้องการ process ข้อมูลกับ serial comm. เช่น ต่อ ADC เข้ากับ UART หรือต่อ LCD display เข้ากับ Serial comm. ก็ต้องใช้ UART ช่วย เช่นเบอร์ 8250 หรือ 16550A หรือเบอร์อื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 เอ็กซ์เรย์ (X-ray)

เอ็กซ์เรย์คือรังสีของแสงที่มองไม่เห็นด้วยตาเปล่าเช่นเดียวกับแสงสว่างธรรมดา เอ็กซ์เรซ์มีลักษณะเป็นทั้งคลื่นและอนุภาคของแม่เหล็กไฟฟ้าที่มีช่วงคลื่นสั้นมากความยาวคลื่นตั้งแต่ 0.04 ถึง 1000 อังสตรอม (1 อังสตรอมเท่ากับ 10^{-7}เซนติเมตร) หรืออยู่ระหว่างรังสีแกรมม่ากับรังสีอุลตร้าไวโอเลต คุณสมบัติของเอ็กซ์เรย์นั้นคล้ายคลึงกับแสงสว่างธรรมดาเป็นส่วนใหญ่ แต่คุณสมบัติพิเศษของมันคือมีอำนาจทะลุทะลวงผ่านวัตถุต่างๆได้มากบ้างน้อยบ้างขึ้นอยู่กับความแน่นทึบและน้ำหนักอะตอมของวัตถุที่ผ่านมัน นอกจากนั้นยังทำให้เกิดการเปลี่ยนแปลงทั้งทางเคมีชีวะและอื่นๆ อีกด้วย

2.5.1 ประวัติการค้นพบเอ็กซ์เรย์

ผู้ค้นพบเอ็กซ์เรย์เป็นคนแรกคือนักฟิสิกซ์ชาวอเมริกันชื่อ วิลเฮม คอนราด เรินท์เก้น (Wilhem Conrad Roentgen) การพบนี้เกิดขึ้นในตอนเย็นวันที่ 9 พฤศจิกายน ค.ศ.1895 ภายในห้องทดลอง ณ มหาวิทยาลัย วู๊ซบรุค ประเทศเยอรมัน เรินท์เก้นได้พบเอ็กซ์เรย์โดยบังเอิญ คือขณะที่เขากำลังทำการทดลองเกี่ยวกับเรื่อง “Absorption of Cathode rays “ โดยใช้หลอดทดลองเรียกว่า Crookes’ tube เขาสังเกตเห็นว่า Cathode rays ที่ออกมาจากหลอดทดลองทำให้กระดาษแข็งที่ฉาบด้วยแบเรี่ยมปลาติโนไซยาไนด์ เกิดเรืองแสงขึ้น ระยะไกลที่สุดที่ยังมีการเรืองแสงบนกระดาษแข็งคือ 120 เซนติเมตร

ในขณะเดียวกันเขาสังเกตุเห็นอีกว่า ตัวอักษร A ที่ทาบนแบเรียมปลาติโนไซยาไนด์อยู่ห่างออกไปเกือบสิบฟุตเกิดเรืองแสงขึ้นด้วยทั้งๆที่ไม่อยู่ในระยะของ Cathode rays เรินท์เก้นจึงคิดว่าเขาค้นพบชนิดใหม่ขึ้นแล้วและให้ชื่อว่า X-ray และเขาได้ใช้เวลาอีกหลายสัปดาห์ต่อมาทำการสังเกตุถึงการทะลุทะลวงของเอ็กซ์เรย์ผ่านกระดาษผ่านโลหะและแม้กระทั่งผ่านเนื้อหนังของคน และเขาได้ถ่ายภาพรังสีของมือของภรรยาเขาไว้ด้วย ในที่สุดเขาจึงประกาศให้โลกได้รู้ว่า เขาได้ค้นพบเอ็กซ์เรย์เป็นคนแรก

2.5.2 ลักษณะและคุณสมบัติของเอ็กซ์เรย์

1. เป็นรังสีคลื่นแม่เหล็กไฟฟ้ามีช่วงคลื่นสั้นมาก คืออยู่ในช่วงตั้งแต่ 0.04-1000 A°หรืออยู่ในช่วงระหว่างรังสีแกรมม่ากับรังสีอุลตร้าไวโอเลต

2. มีคุณสมบัติเหมือนแสงสว่างธรรมดา เดินทางเป็นเส้นตรง ถ้าเดินทางในสุญญากาศแล้วจะเดินทางด้วยความเร็วเท่ากับแสงคือ 186,000 ไมล์ หรือ $3x10^{8}$ เซนติเมตรต่อวินาที นอกจากนั้นแล้วยังมีการสะท้อนกลับ หักเห และเบี่ยงเบนเช่นเดียวกับแสงสว่างธรรมดา

3. ไม่หักเหโดยสนามแม่เหล็กหรือสนามไฟฟ้า

4. เกิดจากการที่อนุภาคอิเล็กตรอนที่มีความเร็วสูงวิ่งเข้าชนเป้าอิเล็คตรอนดังกล่าวนี้จะวิ่งไปชนอิเล็คตรอนตัวอื่นๆ ที่อยู่ในวงโคจรของอะตอมของเป้าให้หลุดกระเด็นออกนอกวงโคจร อิเล็กตรอนของเซลล์นอกถัดออกไปจะวิ่งเข้ามาแทนที่ และคายพลังงานส่วนเกินออกมาในรูปของเอ็กซ์เรย์และความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำให้เกิดการเรืองแสงในสารพิเศษบางอย่าง

6. ดูดกลืน (Absorbed) โดยสสาร (Matter) ทุกชนิดมากบ้างน้อยบ้างขึ้นอยู่กับความหนาแน่นและน้ำหนักของอตอมของสสารนั้น

7. ทำให้เกิดการปล่อยประจุไฟฟ้า (Ionization) เมื่อผ่านไปในอากาศหรือก๊าซ

8. ทำให้เกิดการเปลี่ยนแปลงทางเคมี เช่นเมื่อเอ็กซ์เรย์ไปถูกฟิล์มถ่ายรูปจะทำให้ฟิล์มนั้นดำจึงนำผลอันนี้มาใช้ในการบันทึกภาพรังสีบนแผ่นฟิล์มเอ็กซ์เรย์

9. ทำให้เกิดการเปลี่ยนแปลงทางชีวะ เป็นต้นว่าทำให้เซลล์ของร่างกายเปลี่ยนแปลงหรือเกิดการผ่าเหล่า (Genetic mutation) ถ้าได้รับรังสีเป็นจำนวนมาก

10. มีอำนาจในการทะลุทะลวงสูง สามารถทะลุผ่านเนื้อหนังของมนุษย์และสัตว์ได้แต่ไม่สามารถทะลุผ่านแผ่นตะกั่วหรือคอนกรีตหนาๆ ได้

2.5.3 การเกิดการเอ็กซ์เรย์

เอ็กซ์เรย์เกิดขึ้นโดยที่อนุภาคของอิเล็กตรอนที่มีความเร็วสูงวิ่งไปชนเป้า ผลทำให้เกิดเป็น เอ็กซเรย์และความร้อน

การเกิดของเอ็กซเรย์นี้อาศัยองค์ประกอบสำคัญ 5 ประการคือ

1. ทำให้เกิดอนุภาคอิเล็กตรอนหรือการแยกอนุภาคอิเล็กตรอนออกจากอะตอมของโลหะ

อิเล็กตรอนจะเกิดขึ้นหรือแยกจากอะตอมของโลหะทำเป็นใส้หลอดเอ็กซ์เรย์ โดยการที่เราผ่านกระแสไฟฟ้าเข้าไปในใส้หลอด (Filament) จนกระทั่งใส้หลอดร้อนขึ้นประมาณ 2000°เซลเซียส หรือมากกว่านั้นจะมีผลทำให้เกิดเทอร์มิโอนิคอิมิชั่น (Thermionic Emission) คือ อิเล็กตรอนหลุดออกจากเซลล์หรือวงโคจรของมันเมื่อถูกความร้อน อิเล็กตรอนที่หลุดออกมานี้จะมาออกันอยู่รอบๆ ผิวหน้าของโลหะจนกลายเป็นกลุ่ม (cloud)เรียกว่า Space charge สาเหตุที่อิเล็กตรอนไม่สามารถหลุดพ้นไปจากผิวหน้าของโลหะเพราะว่าแรงดึงดูดระหว่างอิเล็กตรอนและอะตอมของโลหะยังมีอยู่

2. การทำให้อนุภาคอิเล็กตรอนหลุดและเคลื่อนที่ไปด้วยความเร็วสูง

วิธีที่จะทำให้อนุภาคอิเล็กตรอนหลุดจากอะตอมไปได้คือต้องหาสิ่งที่มีพลังงานหรือแรงดึงดูดมากกว่าแรงดึงดูดระหว่างอิเล็กตรอนกับอะตอมของโลหะมาดึงดูดอิเล็กตรอน สิ่งนั้นก็คือไฟฟ้าแรงสูงไฟฟ้าแรงสูงนี้นอกจากจะทำให้อิเล็กตรอนหลุดจากผิวหน้าของโลหะแล้วยังทำให้อิเล็กตรอนเคลื่อนที่หรือวิ่งไปด้วยความเร็วสูงขึ้น ความต่างศักย์ 100 kilo-voltage จะทำให้อิเล็กตรอนมีความเร็วประมาณ 165 กิโลเมตรต่อวินาที

3. การทำให้เส้นทางอิเล็กตรอนวิ่งผ่านไปโดยไม่มีสิ่งกีดขวาง จึงจะทำให้อิเล็กตรอนวิ่งไปด้วยความเร็วสูง ดังนั้นภายในหลอดเอ็กซ์เรย์จึงเป็นสุญญากาศ เพราะมีอากาศหรือก๊าซอยู่ภายในหลอดแม้เพียงเล็กน้อยก็ตาม ความเร็วของอิเล็กตรอนลดลงเนื่องจากการเกิดไอออนไนเซชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การทำให้อิเล็กตรอนมีความเข้ม (Concentration of eletron)

หมายถึงจะต้องมีวิธีการที่จะทำให้อิเล็กตรอนเคลื่อนที่ไปในทิศทางเดียวกัน คือพุ่งไปหาจุดโฟกัสของเป้าด้วยปริมาณ (ความเข้ม) ที่มากพอควร ซึ่งทำได้โดยใช้เครื่องมือที่เรียกว่า "Electron focusing device" คอยควบคุมให้อิเล็กตรอนส่วนใหญ่ไปตกในบริเวณจุดโฟกัสของเป้า

5. การทำให้อิเล็กตรอนหยุดวิ่งในทันที

การที่จะทำให้อิเล็กตรอนที่วิ่งมาด้วยความเร็วสูงนี้หยุดวิ่งได้ในทันทีทันใด ทำได้โดยหาวัตถุมากั้นหรือขวางอิเล็กตรอนไว้ วัตถุนั้นก็คือเป้า ผลที่เกิดขึ้นคือเกิดการชนหรือกระทบกันระหว่างอิเล็กตรอนกับเป้าอย่างแรงอิเล็กตรอนดังกล่าวจะไปชนอิเล็กตรอนที่อยู่ในวงโคจรของอะตอมของเป้านั้นให้หลุดกระเด็นออกนอกวงโคจรและจะมีอิเล็กตรอนจากวงโคจรอื่นๆที่อยู่ถัดไปวิ่งเข้ามาแทน แต่เนื่องจากพลังงานของอิเล็กตรอนในแต่ละวงจรจะไม่เท่ากัน ดังนั้นจึงมีการคายพลังส่วนเกินออกมาในรูปเอ็กซ์เรย์และความร้อนกล่าวคือจากพลังงานของอิเล็กตรอนทั้งหมดที่วิ่งเข้าสู่เป้านั้น 99.8 %จะเปลี่ยนเป็นความร้อน และ 0.2 % เป็นเอ็กซ์เรย์

2.5.4 โครงสร้างหลอดเอ็กซ์เรย์

หลอดเอ็กซ์เรย์ที่ใช้กันในวงการแพทย์มีด้วยกัน 2 ชนิด คือ

1. หลอดเอ็กซ์เรย์ชนิด Stationary anode tube

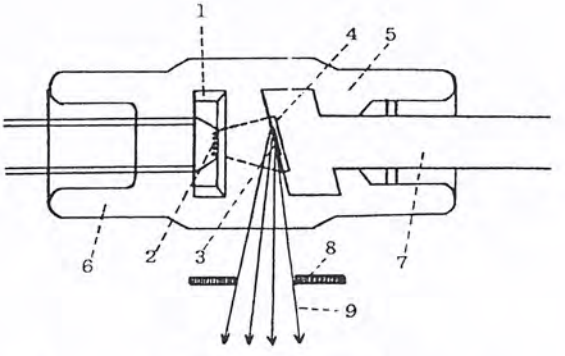
หมายถึงหลอดเอ็กซ์เรย์ชนิดที่ขั้วบวก อยู่คงที่ กล่าวคือขั้วบวกมักจะทำเป็นแท่งทองแดงลักษณะเป็นบล็อกและมีเป้าเป็นบริเวณสีเหลี่ยมเล็กๆ ทำด้วยแผ่นโลหะทังสเตนซึ่งฝังลงในบล็อกทองแดงอีกทีหนึ่ง

2. หลอดเอ็กซ์เรย์ชนิด Rotating anode tube

หมายถึงหลอดเอ็กซ์เรย์ชนิดที่ขั้วบวกไม่อยู่คงที่ กล่าวคือจะหมุนรอบตัวตลอดเวลาที่ทำการถ่ายเอ็กซ์เรย์ ขั้วบวกของหลอดชนิดนี้ จะมีรูปร่างคล้ายจาน ทำด้วยทังสเตนและมีแกนหรือด้ามลักษณะเรียวเล็กทำด้วยโมลิบดินั่ม และมีส่วนที่ทำให้เกิดการหมุนอยู่ตรงปลายของด้ามนั้น

ส่วนประกอบที่สำคัญของหลอดเอ็กซ์เรย์ทั้ง 2 ชนิด ดูได้จาก Diagrams 2 รูปต่อไปนี้

Stationary anode tube



รูปที่ 2.10 แสดงให้เห็นถึงลักษณะ และส่วนประกอบที่สำคัญของ Stationary anode tube โดยลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ 1 = ขั้วลบ

2 = ไส้หลอด

3 = จุดโฟกัส

4 = Tungsten target

5 = บริเวณสุญญากาศ

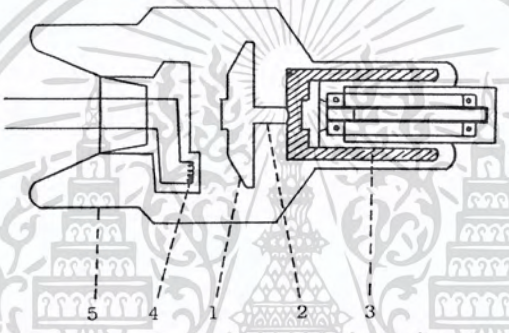
6 = หลอดแก้ว

7 = แกนของขั้วบวก

8 = ที่ควบคุมขนาดของลำเอ็กซ์เรย์

9 = ลำแสงเอ็กซ์เรย์ที่นำมาใช้ประโยชน์

Rotating anode tube



รูปที่ 2.11 แสดงให้เห็นถึงลักษณะและส่วนประกอบที่สำคัญๆของ Rotating anode tube

1 = Tungsten target

2 = แกนของขั้วบวกทำด้วยโมลิปดินัม

3 = ระบบลูกปืน

4 = ขั้วลบพร้อมกับใส้หลอด

5 = หลอดแก้วสุญญากาศ

จากรูปที่ 2.10 และรูปที่ 2.11 พอจะสรุปส่วนประกอบหรือโครงสร้างสำคัญของหลอดเอ็กซ์เรย์ทั้ง 2 ชนิดดังนี้

1.โครงสร้างทั้งหมดของเอ็กซ์เรย์จะถูกบรรจุอยู่ในหลอดแก้วที่ผนึกอย่างดี ทั้งนี้เพื่อให้ภายในหลอดแก้วเป็นสุญญากาศ

2. มีขั้วไฟฟ้า 2 ขั้ว คือ ขั้วบวก (Anode) และขั้วลบ (Cathode)

Anode มี 2ชนิดคือ

- Anode ชนิดอยู่กับที่ชนิดนี้ทำด้วยทองแดง ปลายด้านหนึ่งติดอยู่กับหลอดแก้ว ส่วนอีกปลายหนึ่งมีแผ่นโลหะทังสเตนขนาดครึ่งตารางนิ้วติดอยู่ตรงกลาง เรียกว่า เป้า การที่ใช้โลหะทังสเตนเป็นเป้าก็ด้วยสาเหตุคือทังสเตนเป็นโลหะที่มีจุดหลอมเหลวสูง ต้านทานความร้อนได้ดี

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงไม่ละลาย กับทั้งยังเป็นโลหะที่มีจุดหลอมเหลวสูง เป็นผลให้สามารถผลิตแสงเอ็กซ์เรย์ได้ดีมากกว่าโลหะที่มีน้ำหนักอะตอมต่ำ

- Anode ชนิดที่หมุนได้ชนิดนี้มีลักษณะคล้ายๆจานกลมทำด้วยโลหะทังสเตน หมุนอยู่กับแกนที่ทำด้วยโมลิปดินัมที่ทำเป็นจานก็เพื่อที่จะเพิ่มเนื้อที่ของ Anode ให้มากขึ้นและช่วยระบายความร้อนอีกด้วย

สำหรับขั้วลบนั้นประกอบด้วยลวดทังสเตนที่พันเป็นขดเรียกว่าไส้หลอด ขดลวดนี้จะบรรจุอยู่ในที่รองรับที่มีลักษณะถ้วยเรียกว่า '' Focusing cup" ซึ่งเป็นส่วนที่ต่อออกไปภายนอกหลอดแก้วเพื่อต่อกับวงจรไฟฟ้า

3. เป้าเป็นส่วนสำคัญที่จะทำให้เกิดเอ็กซ์เรย์ทั้งนี้อิเล็กตรอนต้องวิ่งมาชนเป้านี้จึงจะเกิดเอ็กซ์เรย์ จุดหรือบริเวณที่เป็นเป้าให้อิเล็กตรอนวิ่งเข้าชนนี้เรียกว่าจุดโฟกัส จุดนี้คือจุดกำเนิดของเอ็กซ์เรย์

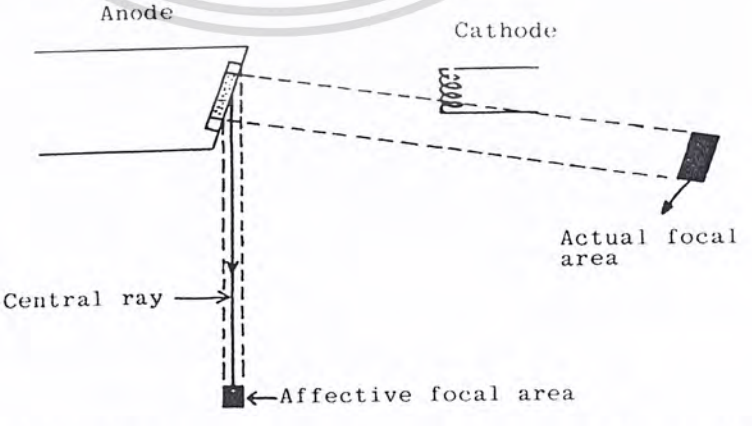
ขนาดของจุดโฟกัสนั้นมีความสำคัญเกี่ยวกับการถ่ายภาพเอ็กซ์เรย์มาก โฟกัสที่มีขนาดเล็กจะให้คุณภาพที่มีคุณภาพดี มีรายละเอียดและความชัดเจนมากกว่า ส่วนจุดโฟกัสที่มีขนาดใหญ่จะมีความทนทานต่อความร้อนได้มากกว่าแต่ภาพที่ได้รับจะมีความชัดเจนน้อยกว่าภาพที่ถ่ายโดยใช้จุดโฟกัสขนาดเล็ก

ดังนั้นเพื่อที่จะได้ภาพที่มีความชัดเจนและเพื่อให้จุดโฟกัสมีความทนทานต่อความร้อนได้เป็นอย่างดีจึงได้มีการหาวิธีที่จะแก้ปัญหาดังกล่าวนี้ ทำได้ 2 วิธีคือ

1 ใช้วิธีสร้าง Line-focus principle

2 ใช้วิธีทำให้ Anode หมุน

Line-focus principle นั้น อาศัยหลักการคือ บังคับให้ลำอิเล็กตรอนวิ่งไปสู่เป้าในลักษณะเป็นสี่เหลี่ยมมุมฉากแคบๆและสร้างให้ด้านหน้าของเป้า (Target) เอียงเป็นมุมประมาณ 17° - 20° กับด้านหน้าของคาโถด (Cathode) ดังในรูปที่ (2.12)



รูปที่ 2.12 แสดงให้เห็นถึง Line-focus principle และการเอียง Target เป็นมุม 17°-20° เพื่อทำให้ได้ Effective focal spot ขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างให้เป้าเอียง 17°-20°นี้มีวัตถุประสงค์ คือ

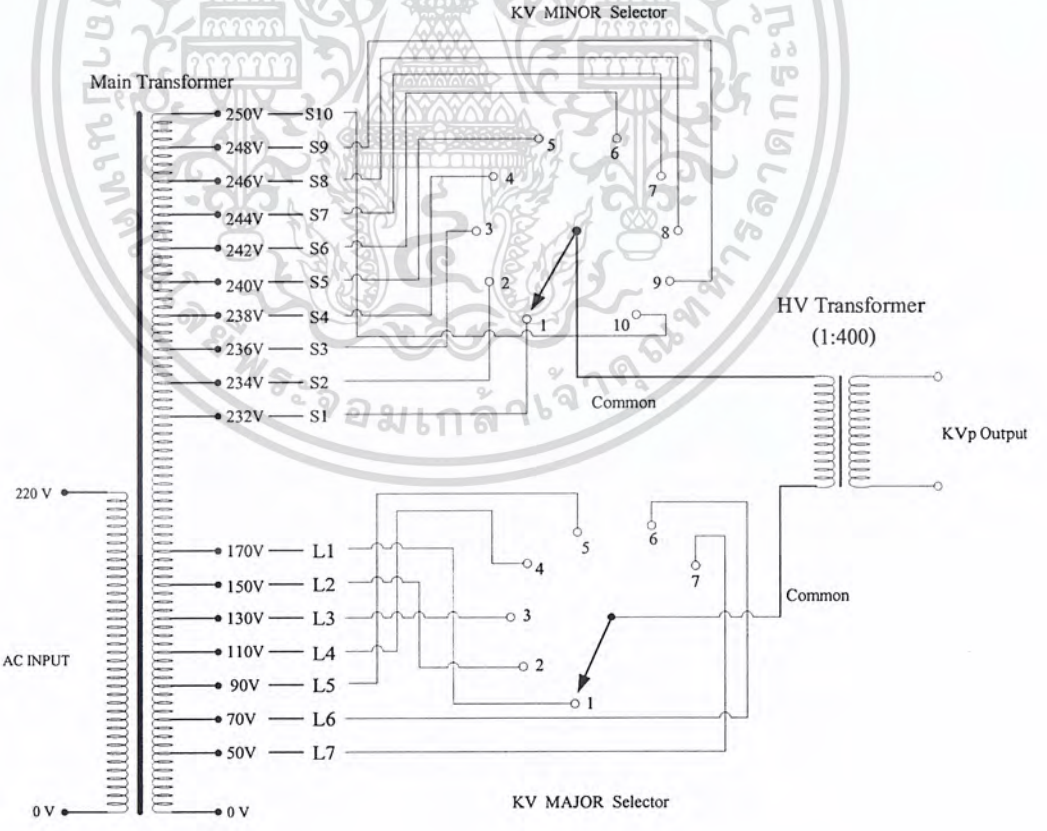
1. ทำให้ Effective focal area มีขนาดเล็ก ภาพที่ได้จะชัดเจนและคม
2. ทำให้ Actual focal area มีขนาดใหญ่ รับความร้อนได้มากและทนความร้อนได้ดี

สำหรับวิธีที่ทำให้ Anode หมุนนั้นก็เพื่อจะเพิ่มเนื้อที่ของ Anode ให้มากขึ้น ผลคือทนทานต่อความร้อนได้มากขึ้นและตำแหน่งของ Focal spot ที่ถูกอิเล็กตรอนชนจะเปลี่ยนที่ไปเนื่องจาก Anode มีการหมุนต่อเนื่องตลอดเวลาเป็นผลให้ระบายความร้อนไปภายในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

ในการออกแบบและการสร้างหน่วยควบคุมเอ็กซเรย์ เนื่องจากเครื่องเอ็กซเรย์ที่ใช้งานโดยทั่วไปนั้นมักจะมีการเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์ในรูปแบบการเลือกที่สวิตซ์ซีเลคเตอร์เพื่อทำการเลือกแรงดัน ส่วนการเลือกเทคนิคที่ใช้ในการเอ็กซเรย์ตลอดจนการเลือกค่ากระแสที่ใช้ในการเอ็กซเรย์จะเป็นการควบคุมโดยการกดสวิตซ์เพื่อทำการเลือกค่าที่ต้องการใช้งาน แต่ในการนำเครื่องเอ็กซเรย์มาพัฒนาเพื่อเชื่อมต่อการทำงานกับเครื่องคอมพิวเตอร์ เพื่อให้สามารถทำการควบคุมการเอ็กซเรย์ผ่านทางคอมพิวเตอร์ได้นั้น ถ้าเรายังคงใช้เครื่องเอ็กซเรย์ในรูปแบบเก่า คือใช้ระบบควบคุมการทำงานในเชิงกลไกก็จะเป็นการยากในการควบคุมการทำงานผ่านทางคอมพิวเตอร์ ดังนั้นเราต้องทำการเปลี่ยนส่วนควบคุมการทำงานของเครื่องเอ็กซเรย์ให้อยู่ในรูปแบบการควบคุมในเชิงตัวเลขหรือการควบคุมการทำงานด้วยระบบดิจิตอลเพื่อความสะดวกในการพัฒนาและการนำหน่วยควบคุมเอ็กซเรย์ไปเชื่อมต่อการทำงานกับเครื่องคอมพิวเตอร์



รูปที่ 3.1 แสดงแผนผังของการเลือกค่าแรงดันที่ใช้ในเครื่องเอ็กซเรย์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 แสดงแผนผังของการเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์ ซึ่งจะทำการเลือกค่าแรงดันโดยการเลือกที่สวิตช์ซีเลคเตอร์ซึ่งมีอยู่ 2 ตัว คือ Major และ Minor ซึ่ง Major จะทำการเลือกค่าแรงดันอย่างหยาบๆ คือ ซีเลคเตอร์ในแต่ละตำแหน่งจะมีค่าแรงดันต่างกันประมาณ 10 Kvp ส่วน Minor จะทำการเลือกค่าแรงดันอย่างละเอียด คือ ซีเลคเตอร์ในแต่ละตำแหน่งจะมีค่าแรงดันต่างกันประมาณ 1 Kvp โดยค่า Kvp output สามารถคำนวณได้จากสมการที่ 3.1

$$\text{KVp output} = \sqrt{2}\,(\text{Sn} - \text{Lm}) \times 400 \qquad (3.1)$$

เมื่อ Sn : KV Minor Selector ที่ตำแหน่ง n ต่างๆ ; n = 1,2,3,...,10

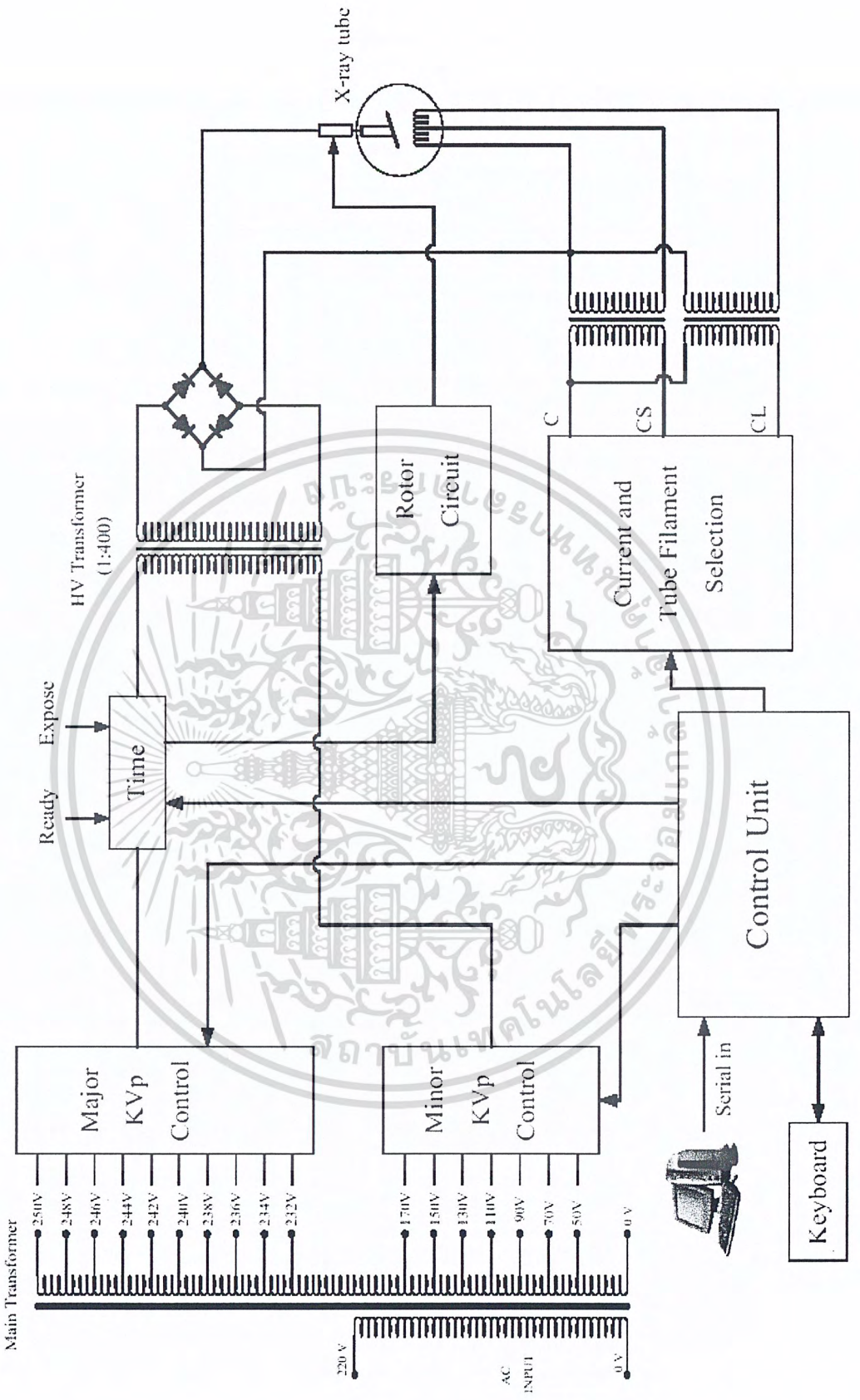
Lm : KV Major Selector ที่ตำแหน่ง n ต่างๆ ; m = 1,2,3,...,7

โดยที่ค่า 400 คืออัตราส่วนของ HV Transformer (1:400) ในการใช้งานเพื่อให้ได้แรงดัน KVp Output ตรงตามความต้องการเราจะทำการปรับที่สวิตช์ซีเลคเตอร์ทั้ง 2 ตัว ส่วนในการเลือกค่าเทคนิคตลอดจนค่ากระแสและค่าเวลาที่ใช้ในการเอ็กซเรย์จะเป็นการเลือกที่สวิตช์แบบกดเพื่อเลือกรูปแบบในการทำงาน

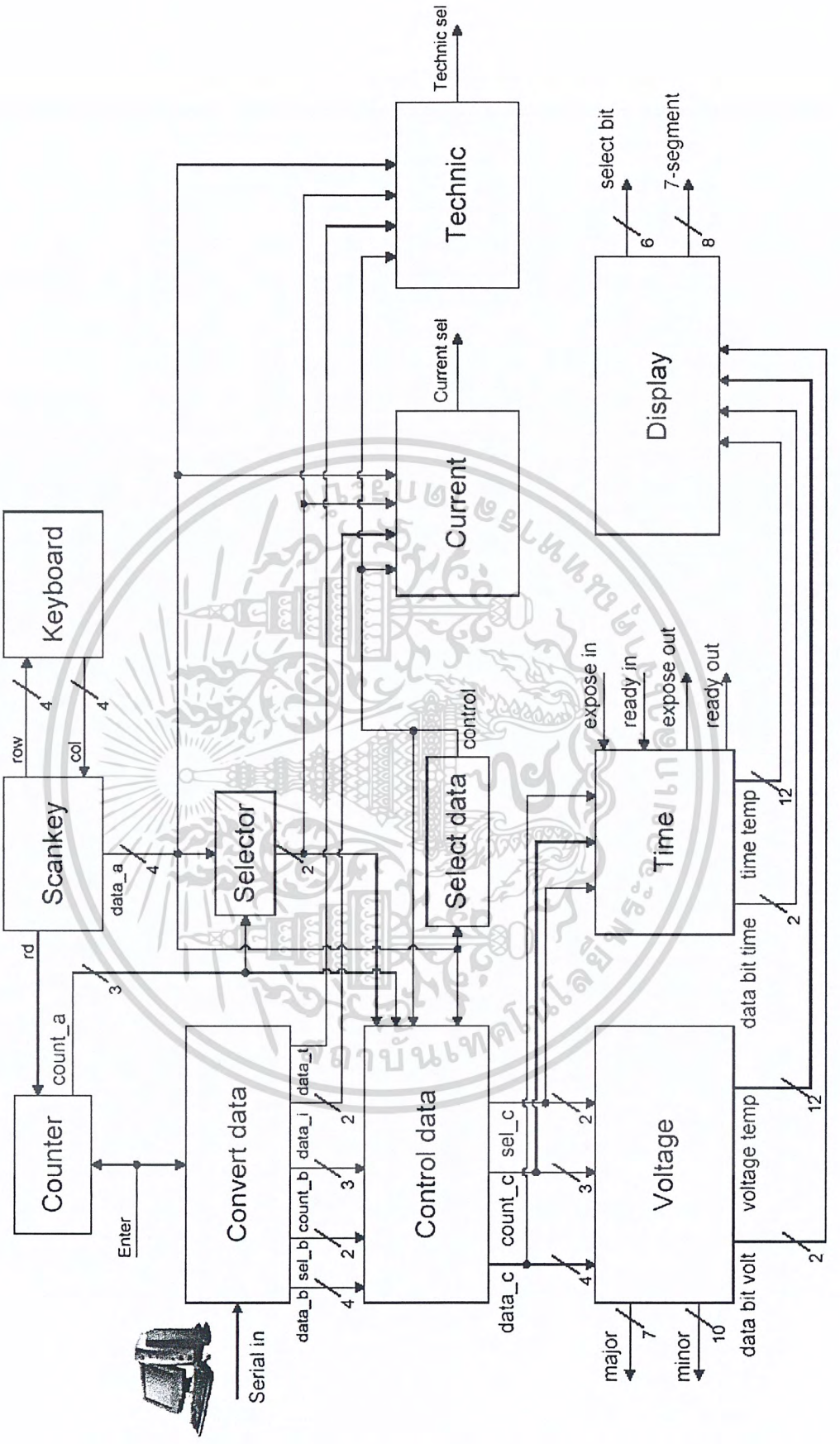
จากรูปแบบในการเลือกค่าเวลาที่ใช้ในการถ่ายเอ็กซเรย์, ค่าเทคนิคและค่ากระแสที่ใช้ในการเอ็กซเรย์จะพบว่าถ้าเราควบคุมให้เป็นระบบดิจิตอลนั้นก็สามารถทำได้ แต่ในการเลือกค่าแรงดันจะพบว่าจะต้องมีการเปลี่ยน ขดหม้อแปลงซึ่งแต่เดิมจะใช้สวิตช์ซีเลคเตอร์ในการเปลี่ยนขด ดังนั้นถ้าเราจะใช้ระบบควบคุมแบบดิจิตอลเราจำเป็นจะต้องเปลี่ยนสวิตช์ซีเลคเตอร์ในแต่ละตำแหน่งมาเป็น SCR แทนเพื่อให้สามารถใช้การควบคุมด้วยระบบดิจิตอลได้ เมื่อเราสามารถเปลี่ยนการควบคุมทั้งหมดให้อยู่ในระบบดิจิตอลได้แล้วการนำเครื่องเอ็กซเรย์ไปเชื่อมต่อกับเครื่องคอมพิวเตอร์ก็สามารถทำได้โดยไม่ยากจนเกินไป

ในส่วนของการควบคุมด้วยระบบดิจิตอลจะใช้อุปกรณ์ FPGA ในการออกแบบจำเป็นจะต้องเขียนโปรแกรมต่างๆ เพื่อให้อุปกรณ์ FPGA ทำหน้าที่เป็นส่วนควบคุมการทำงานของเครื่องเอ็กซเรย์โดยมีบล็อกไดอะแกรมแสดงการทำงานดังรูปที่ 3.2 และรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงการทำงานโดยรวมของหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์



รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานภายในส่วนควบคุมด้วยชิป FPGA

จากบล็อกไดอะแกรมสามารถอธิบายการทำงานในแต่ละส่วนได้ดังนี้

Keyboard

ทำหน้าที่ในการรับข้อมูลจากผู้ใช้ซึ่งคีย์บอร์ดที่ใช้มีขนาด 4x4 โดยจะทำงานร่วมกับบล็อก Scankey เพื่อตรวจเช็คตำแหน่งของคีย์ที่ถูกกด

Scankey

ทำหน้าที่ตรวจสอบการกดคีย์บอร์ด โดยมีอินพุทของการตรวจเช็คค่าคือ col ซึ่งเป็นข้อมูลขนาด 4 บิตและเอาท์พุทที่ใช้ในการตรวจเช็คคือ row ซึ่งเป็นข้อมูลขนาด 4 บิตเช่นเดียวกัน โดยการทำงานของบล็อกนี้จะมีดังนี้คือ เมื่อมีการปล่อยสัญญาณตรวจเช็คออกไปทาง row ถ้าไม่มีการกดคีย์บอร์ดก็จะไม่เกิดการเปลี่ยนแปลงขึ้นที่ col ซึ่งเป็นอินพุทของบล็อก แต่ถ้ามีการกดคีย์บอร์ดก็จะทำให้เกิดการเปลี่ยนแปลงขึ้นที่ col ซึ่งมีผลทำให้สามารถตรวจสอบตำแหน่งของคีย์บอร์ดที่ถูกกดได้จากข้อมูลที่ส่งออกไปทาง row และข้อมูลที่รับเข้ามาทาง col ส่วนเอาท์พุทที่ได้จากบล็อกการทำงานนี้คือสัญญาณ data_a ซึ่งเป็นข้อมูลขนาด 4 บิตจะถูกส่งไปยังบล็อก Selector, Control data, Select data, Current และ Technic

Counter

ทำหน้าที่เช็คลำดับของการกดคีย์บอร์ดว่ามีการกดคีย์ใดก่อนหลัง ซึ่งจะมีการจัดทำเป็นลำดับของข้อมูลที่ผู้ใช้ป้อนเข้ามาสู่ส่วนควบคุมการทำงานโดยผ่านทางคีย์บอร์ด โดยที่การทำงานของบล็อก Counter จะมีสัญญาณที่ควบคุมการทำงาน คือ สัญญาณ rd จากบล็อก Scankey และสัญญาณ Enter จากแผงควบคุมที่ใช้เมื่อต้องการป้อนข้อมูลที่ต้องการเข้าสู่หน่วยควบคุม ส่วนเอาท์พุทที่ออกจากบล็อกการทำงานนี้คือ count_a ซึ่งเป็นข้อมูลขนาด 3 บิตที่จะนำไปควบคุมการทำงานของบล็อก Selector และ Control data

Selector

ทำหน้าที่เลือกชนิดของข้อมูลที่ผู้ใช้ป้อนเข้ามา เช่น การเลือกปรับค่าแรงดัน, เลือกค่าเวลาที่ใช้ในการเอ็กซเรย์, เลือกค่ากระแสไฟฟ้าหรือเลือกค่าเทคนิคที่ต้องการใช้ในการถ่ายเอ็กซเรย์

Convert data

เนื่องจากโครงงานนี้ได้มีการนำเอาเครื่องคอมพิวเตอร์เข้ามาควบคุมการทำงานของเครื่องเอ็กซเรย์โดยใช้รูปแบบของการส่งข้อมูลผ่านทางพอร์ตอนุกรม ดังนั้นบล็อกการทำงานนี้มีหน้าที่แปลงรูปแบบของข้อมูลจากข้อมูลแบบอนุกรมให้เป็นข้อมูลแบบขนานและนำไปเก็บไว้ในบัฟเฟอร์เพื่อพักข้อมูลที่ถูกส่งมาจากเครื่องคอมพิวเตอร์ หลังจากนั้นจะมีการแบ่งข้อมูลที่ได้ออกเป็นส่วนๆ ตามหน้าที่การทำงานโดยเอาท์พุทของบล็อกการทำงานนี้คือ data_b, sel_b, data_i, data_t และ count_b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control

ทำหน้าที่ควบคุมข้อมูลที่จะส่งไปยังบล็อก Voltage และ Time ซึ่งจะควบคุมการทำงานด้วยสัญญาณ control ที่ส่งมาจากบล็อก Select data

Select data

ทำหน้าที่ส่งสัญญาณ control ไปยังบล็อก Control data, Current และ Technic เพื่อเป็นการกำหนดข้อมูลที่จะใช้ในการควบคุมเครื่องเอ็กซเรย์ ซึ่งจะเป็นการเลือกข้อมูลระหว่างข้อมูลที่ได้จากการป้อนค่าโดยตรงจากคีย์บอร์ดและข้อมูลในบัฟเฟอร์ที่ส่งมาจากเครื่องคอมพิวเตอร์

Voltage

ทำหน้าที่เลือกขดของหม้อแปลงให้ตรงกับความต้องการ โดยแบ่งออกเป็น Major ซึ่งเป็นข้อมูลขนาด 7 บิตทำหน้าที่ควบคุมการเลือกขดของหม้อแปลงจำนวน 7 ขดโดยเป็นการเลือกค่าแรงดันแบบหยาบ (ความแตกต่างของแรงดันระหว่างหม้อแปลงแต่ละขดประมาณ 10 KV) และสัญญาณ Minor ซึ่งเป็นข้อมูลขนาด 10 บิตซึ่งจะควบคุมการเลือกขดของหม้อแปลงจำนวน 10 ขด และเป็นการเลือกค่าแรงดันแบบละเอียด (ความแตกต่างของแรงดันระหว่างหม้อแปลงแต่ละขดประมาณ 1 KV) แต่ในการเลือกขดของหม้อแปลงนั้นจำเป็นต้องใช้การควบคุมโดยผ่านทางอุปกรณ์ออปโต้คัปเปิลเพื่อทำการแยกส่วนควบคุมการทำงานออกจากไฟฟ้าที่มีค่าแรงดันสูงทางด้านขดของหม้อแปลงออกจากกัน ซึ่งหลังจากผ่านออปโต้คัปเปิลแล้วจะเป็นชุดทริก SCR ซึ่งทำหน้าที่ในการเลือกขดของหม้อแปลงให้ตรงตามความต้องการ

Time

ทำหน้าที่ตั้งเวลาที่ใช้ในการถ่ายเอ็กซเรย์ในการควบคุมการเริ่มต้นของส่วนจับเวลาจะควบคุมโดย Ready in และ Expose in ซึ่ง Ready in จะถูกกดก่อนการการ Expose เป็นเวลา 2 วินาทีเพื่อให้หลอดเอ็กซเรย์ชนิด Stationary หมุนด้วยรอบการทำงานที่คงที่ก่อนการ Expose หลังจากนั้นสวิตซ์ Expose จะถูกกดเพื่อทำการถ่ายเอ็กซเรย์ ซึ่งเป็นการเริ่มต้นการทำงานของการจับเวลาและในการตั้งเวลาสามารถทำการตั้งเวลาได้ตั้งแต่ 0.02 วินาทีจนถึง 5 วินาที

Display

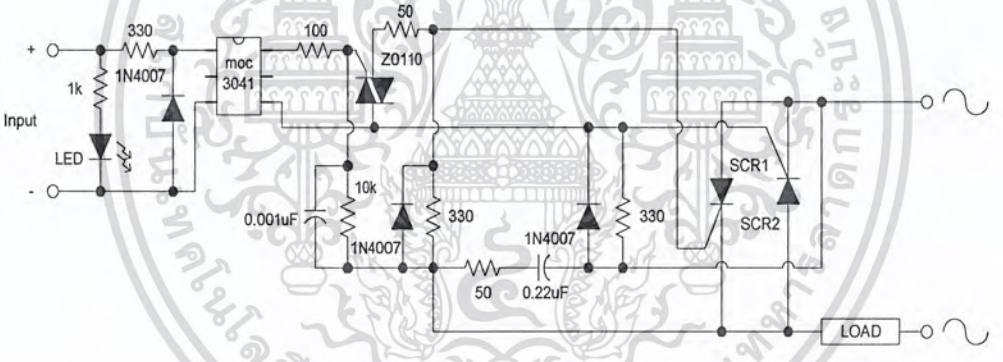
ทำหน้าที่ควบคุมการแสดงผลของค่าแรงดันและเวลาที่เลือกใช้ในการถ่ายเอ็กซเรย์โดยข้อมูลที่ใช้ในการแสดงผลจะส่งมาจากบล็อก Voltage และ Time คือ Voltage_temp และ Time_temp ตามลำดับ ส่วนสัญญาณที่ใช้ในการควบคุมตำแหน่งของแอลอีดีแบบ 7- Segment ที่ต้องการให้ติดสว่างคือ data_bit_volt และ data_bit_time ส่วนเอาท์พุทของบล็อกการทำงานนี้คือ Select_bit และ Segment ซึ่งจะถูกส่งไปยังส่วนแสดงผลที่ติดตั้งอยู่ที่แผงควบคุม

ในแต่ละบล็อกการทำงานจะเขียนบรรยายโครงสร้างภายในด้วยภาษา VHDL ซึ่งเมื่อได้ทำการเขียนโครงสร้างภายในของแต่ละบล็อกแล้วจึงทำการเรียบเรียง (Complier) และจำลองการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน (Simulate) เพื่อดูผลการทำงานของบล็อกการทำงานที่ได้ออกแบบไว้ เมื่อผลการทำงานของแต่ละบล็อกเป็นไปตามที่ต้องการแล้วก็ให้นำแต่ละบล็อกการทำงานมาเชื่อมต่อกันเพื่อให้เป็นบล็อกการทำงานที่สมบูรณ์ของหน่วยควบคุมเครื่องเอ็กซเรย์ด้วยระบบดิจิตอลซึ่งสามารถควบคุมการทำงานผ่านทางเครื่องคอมพิวเตอร์ได้ หลังจากนั้นจึงจำลองการทำงานของทั้งระบบเพื่อดูผลการทำงานที่ได้ว่าเป็นไปตามที่ต้องการหรือไม่และมีข้อบกพร่องในส่วนใดบ้างหรือไม่ หลังจากที่ทำการปรับปรุงจนได้ผลการจำลองการทำงานเป็นไปตามที่ต้องการแล้ว ในขั้นตอนสุดท้ายก็จะเป็นการป้อนโปรแกรมลงในอุปกรณ์ FPGA เพื่อทดสอบและดูผลการทำงานจริงเป็นอันดับต่อไป

ในส่วนของการเลือกขดหม้อแปลงนั้น ทำได้โดยการต่อ SCR ที่ตำแหน่งเดิมของซีเลคเตอร์ที่แสดงดังรูปที่ 3.1 ทำให้จำเป็นต้องใช้ SCR เป็นจำนวน 17 ตัวในการเลือกขดหม้อแปลงทั้ง Major selector และ Minor selector โดยในการทริก SCR ให้ทำงานนั้นจำเป็นต้องมีวงจรสำหรับทริก SCR ให้เริ่มต้นการทำงาน ซึ่งวงจรของชุดทริก SCR ได้แสดงในรูปที่ 3.3

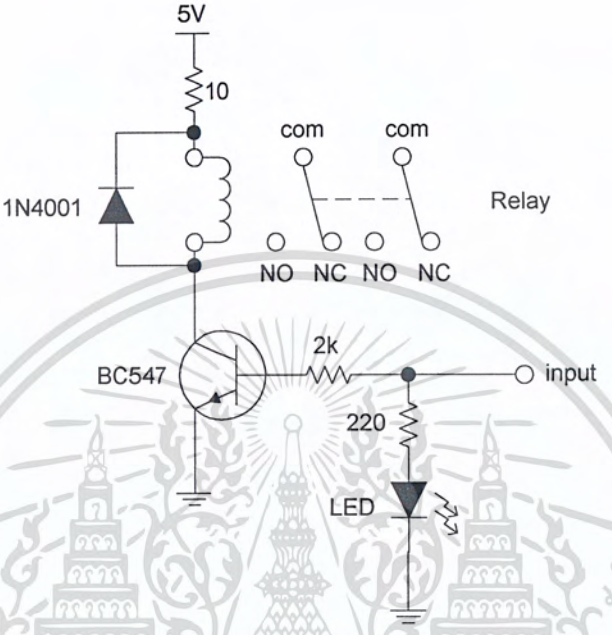


รูปที่ 3.4 แสดงวงจรของชุดทริก SCR

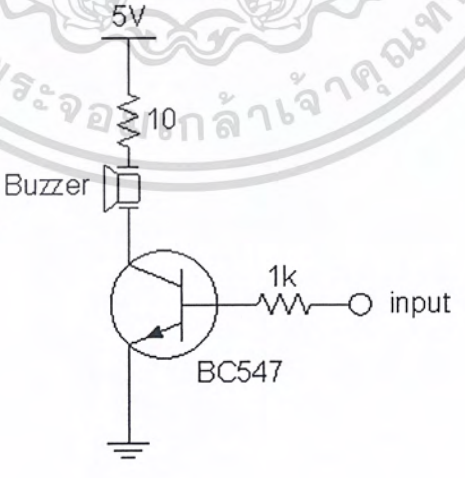
จากรูปที่ 3.4 ได้แสดงวงจรที่ใช้ในการทริก SCR ให้เริ่มการทำงาน โดยที่ในการทริกนั้นจะกระทำผ่านทางออปโต้คัปเปิล ซึ่งออปโต้คัปเปิลที่เลือกใช้เป็นชนิด zero crossing ซึ่งจะเปิดปิดการทำงานของ SCR ในช่วงที่แรงดันไฟสลับมีค่าศูนย์โวลต์ (เฟส 0) ทำให้ในช่วงเริ่มการทำงานไม่มีแรงดันตกคร่อม SCR มีผลทำให้ลดการเสียหายของ SCR ลงได้ และระหว่างขาคาโทดทั้งสองของ SCR ได้มีวงจรสนับเบอร์ต่อไว้เนื่องจากโหลดของ SCR เป็นหม้อแปลง ซึ่งทำให้เกิดแรงดันเกินค่าปกติขึ้นได้ในช่วงสนามแม่เหล็กของหม้อแปลงเกิดการขยายตัว ดังนั้นจึงต้องต่อวงจรสนับเบอร์ซึ่งประกอบด้วยความต้านทานค่า 50 Ω และตัวเก็บประจุค่า 0.22 uF ไว้เพื่อป้องกันแรงดันเกินค่าปกติทำให้ SCR เกิดความเสียหายได้ ในส่วนของการควบคุมการทำงานของรีเลย์ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ในการเลือกเทคนิคและค่ากระแสที่ใช้ในการเอ็กซเรย์ รวมถึงการควบคุมการทำงานของบัซเซอร์ที่ใช้ในการเตือนเมื่อมีความผิดปกติในการทำงานได้มีวงจรควบคุมการทำงานดังแสดงในรูปที่ 3.5 และรูปที่ 3.6



รูปที่ 3.5 แสดงวงจรควบคุมการทำงานของรีเลย์

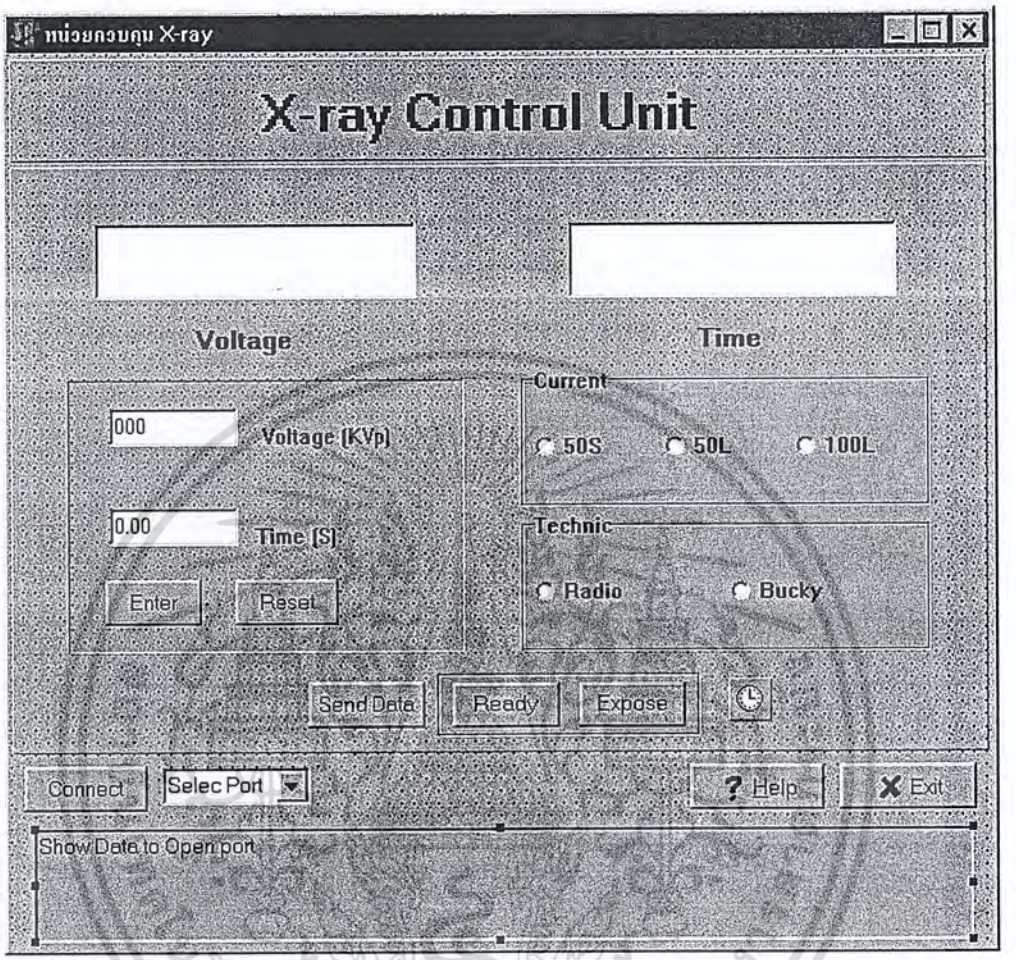


รูปที่ 3.6 แสดงวงจรควบคุมการทำงานของบัซเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการออกแบบหน่วยควบคุมเอ็กซเรย์ผ่านทางคอมพิวเตอร์โดยใช้โปรแกรมเดลไฟ (Delphi)

1. เริ่มต้นสร้างโปรเจกต์ใหม่ (โดยเลือกเมนู File > New Application) แล้ววางคอมโพเนนต์ต่างๆ บนฟอร์มดังนี้



รูปที่ 3.7 การวางคอมโพเนนต์ต่างๆ ลงบนฟอร์ม

2. กำหนดค่าของพรอบเพอร์ตี้ให้กับคอมโพเนนต์ต่างๆ ดังนี้

คอมโพเนนต์	พรอพเพอร์ตี้	ค่าที่กำหนด
ฟอร์ม	Name	X-ray
	Caption	หน่วยควบคุม X-ray
Label	Name	LbHead
	Caption	X-ray Control Unit
Edit1	Name	EdShowvoltage
Edit2	Name	EdShowtime

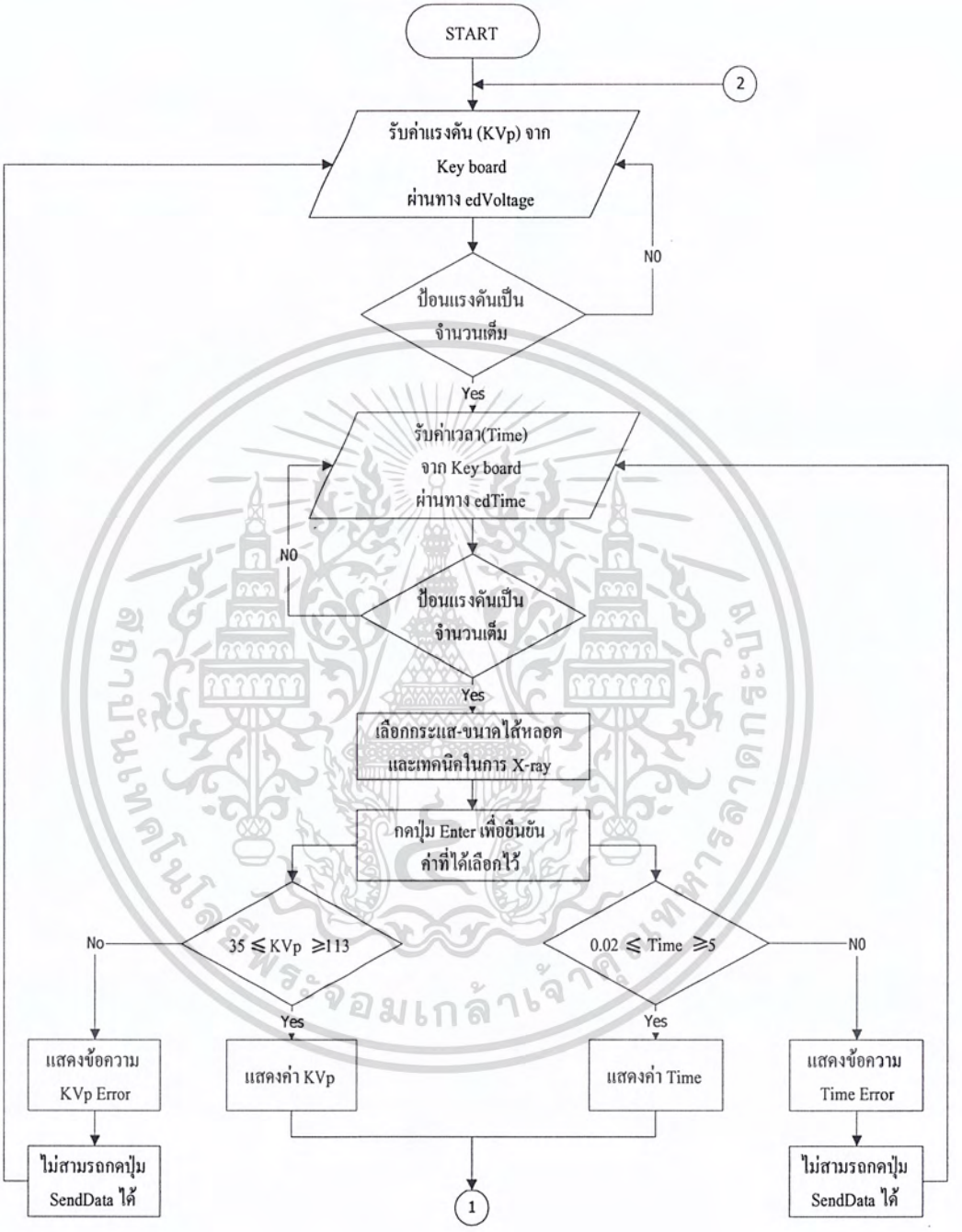
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit3	Name	EdVoltage
Edit4	Name	EdTime
Button1	Name	BEnter
	Caption	Enter
Button2	Name	BReset
	Caption	Reset
RadioGroup1	Name	RgCurrent
	Caption	Current
	Item	50S, 50L, 100L
RadioGroup2	Name	RgTechnic
	Caption	Technic
	Item	Radio, Bucky
Button3	Name	BsendData
	Caption	SendData
Button4	Name	BReady
	Caption	Ready
Button5	Name	BExpose
	Caption	Expose
Button6	Name	BConnect
	Caption	Connect
Button7	Name	Bdisconnect
	Caption	Disconnect
BitBtn1	Name	BbHelp
	Caption	Help
BitBtn2	Name	BbExit
	Caption	Exit
Timer	Name	Timer1

ตารางที่ 3.1 แสดงการกำหนดค่าของพรอบเพอร์ตี้ให้กับคอมโพเนนต์ต่างๆ

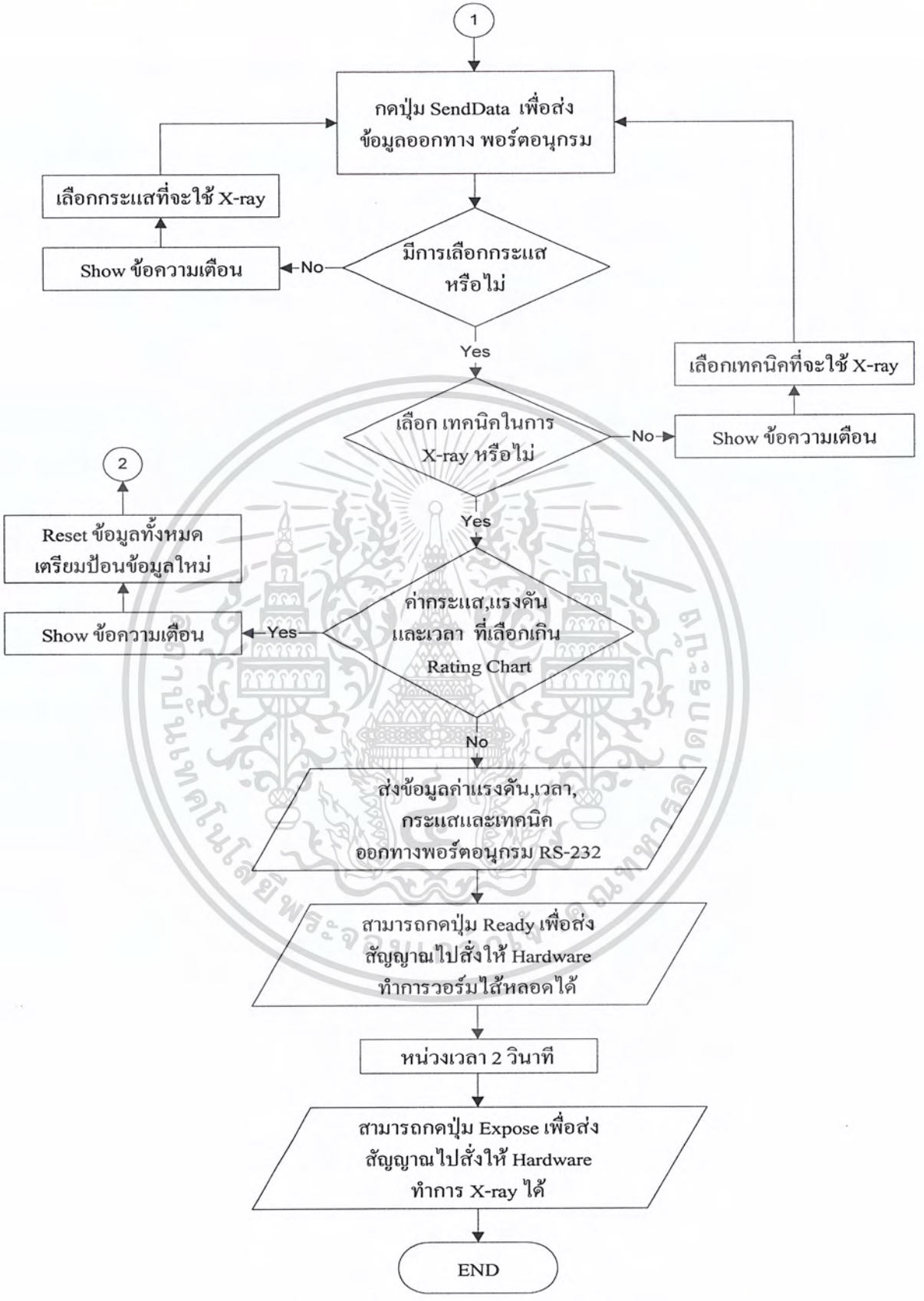
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการเขียนโปรแกรมควบคุมการทำงานของคอมโพเนนต์ต่างๆ โดยเริ่มจากการเขียน FlowChart แสดงการทำงานทั้งหมดของฟอร์ม



รูปที่ 3.8 แสดง Flow Chart แสดงการทำงานของฟอร์ม

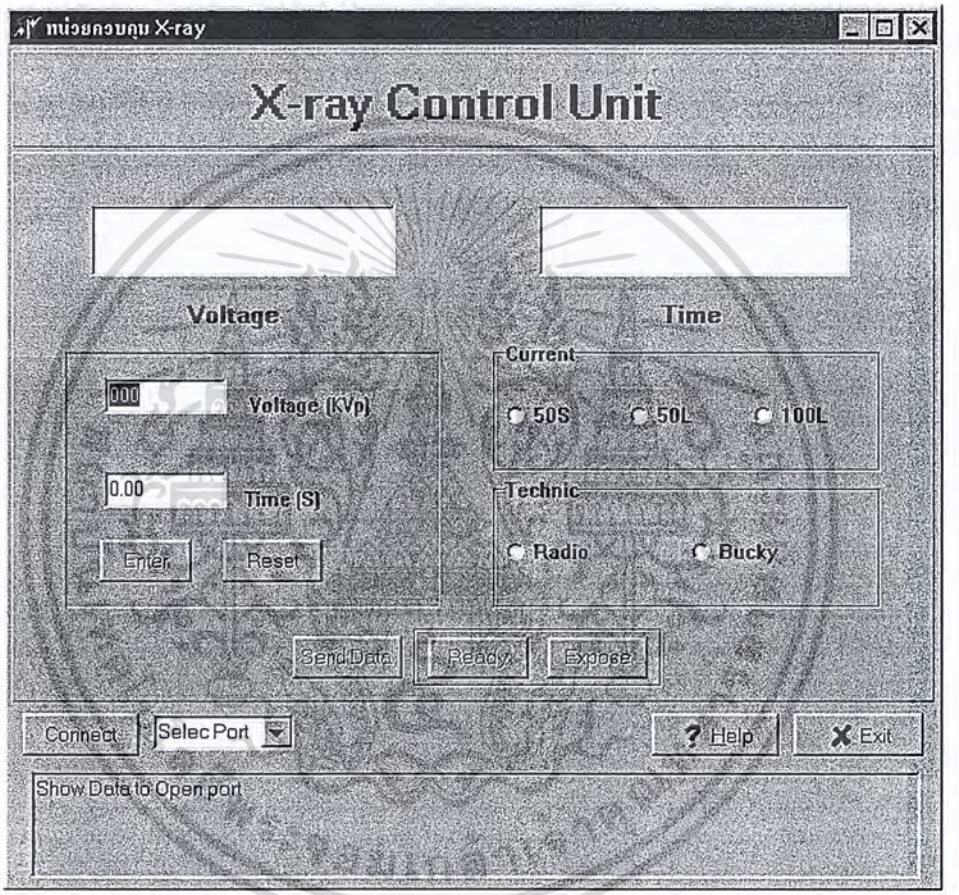
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดง Flow Chart แสดงการทำงานของฟอร์ม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นำลำดับการทำงานตามที่เขียน Flow Chart ไว้มาทำการเขียนโปรแกรมโดยสามารถดู Source Code ได้จากภาคผนวก
5. ทำการเซฟไฟล์โดยต้องทำการเซฟเป็น 2 ไฟล์คือ ไฟล์ยูนิตและไฟล์โปรเจกต์ (โดยใช้เมนู File >SaveAS เซฟไฟล์ยูนิตเป็นชื่อ uXray.pas และเมนู File > Save Project As เซฟไฟล์โปรเจกต์เป็นชื่อ pjXray.dpr)
6. ทำการคอมไพล์ (โดยใช้เมนู Project >Compile pjXray) และรันโปรแกรม (โดยใช้เมนู Run> Run) โปรแกรมจะได้หน้าตาของฟอร์มที่ได้ออกแบบไว้ดังรูปที่ 3.10



รูปที่ 3.10 แสดงผลของการคอมไพล์และรันโปรแกรมที่ได้ออกแบบไว้

7. ทำการทดสอบโปรแกรมโดยการป้อนค่าต่างๆ ตามที่ต้องการจะใช้ในการเอ็กซเรย์ แล้วจึงทำการทดสอบการส่งขอมูลโดยทำการเลือกพอร์ตสื่อสาร (COM1 หรือ COM2) แล้วทำการกดปุ่ม Connect เพื่อเปิดพอร์ตสื่อสาร จากนั้นทำการป้อนข้อมูลที่ต้องการแล้วจึง Send Data เมื่อข้อมูลได้ส่งไปแล้วจึงจะสามารถทำการ Ready และ Expose ได้
8. หลังจากทดสอบโปรแกรมและการทำงานเป็นไปตามที่ต้องการแล้วให้ทำการสร้างไฟล์ (.exe) ซึ่งสามารถนำไปใช้งานได้กับคอมพิวเตอร์ทุกเครื่องโดยไม่ต้องรันผ่านโปรแกรมเดลไฟ (โดยใช้เมนูProject > Build pjXray)

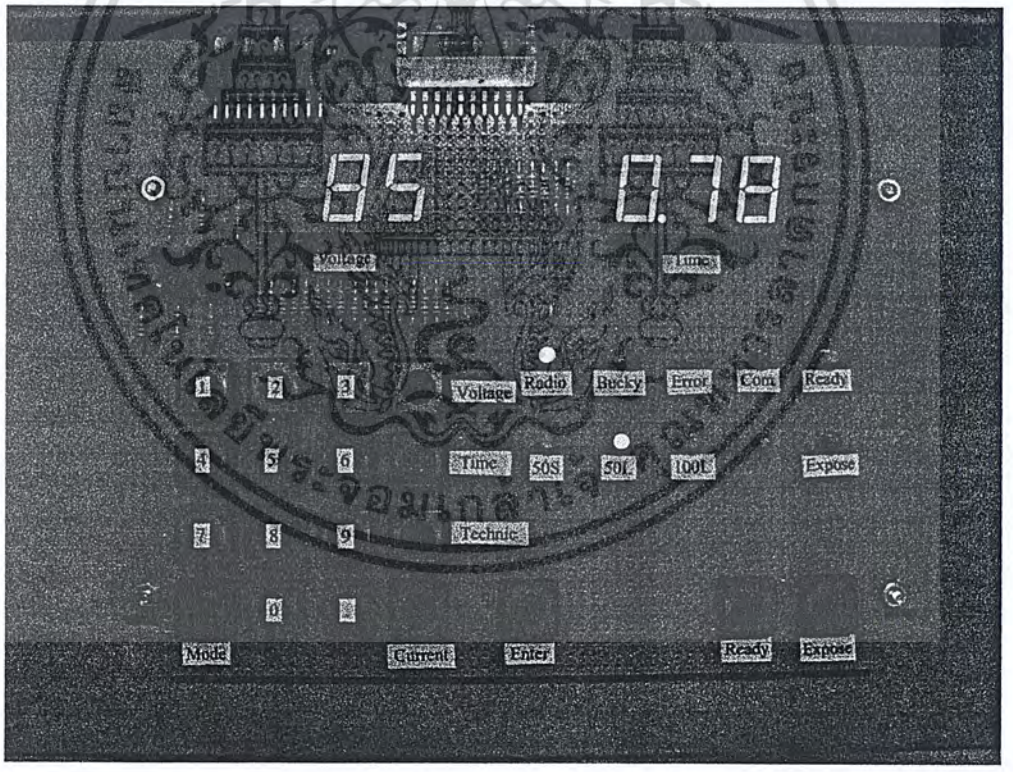
<u>หมายเหตุ</u> วิธีการใช้งานชุดควบคุมสามารถดูได้โดยกดปุ่ม Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในการทดลองหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์นี้ เป็นการทดสอบการทำงานของหน่วยควบคุมเอ็กซเรย์ โดยหน่วยควบคุมที่สร้างขึ้นนี้สามารถทำการควบคุมโดยตรงที่แผงควบคุม (ชุดควบคุมด้วยคีย์บอร์ด)หรือจะควบคุมผ่านทางคอมพิวเตอร์ก็ได้ การทดสอบจะทำการป้อนค่าต่างๆ ที่จะใช้ในการควบคุมการเอ็กซเรย์ทั้งจากคอมพิวเตอร์และจากส่วนแผงควบคุม โดยที่การแสดงผลนั้นจะมีการแสดงผลที่แผงควบคุมและแอลอีดีที่ชุดวงจรทริก SCR ซึ่งจะเป็นการบอกให้รู้ว่าชุดทริกนั้นได้มีการทำงานและที่ส่วนของการ Expose นั้นจะมีการต่อ SCR และหลอดไฟเพื่อแสดงให้เห็นถึงการทำงานที่ชัดเจนขึ้น



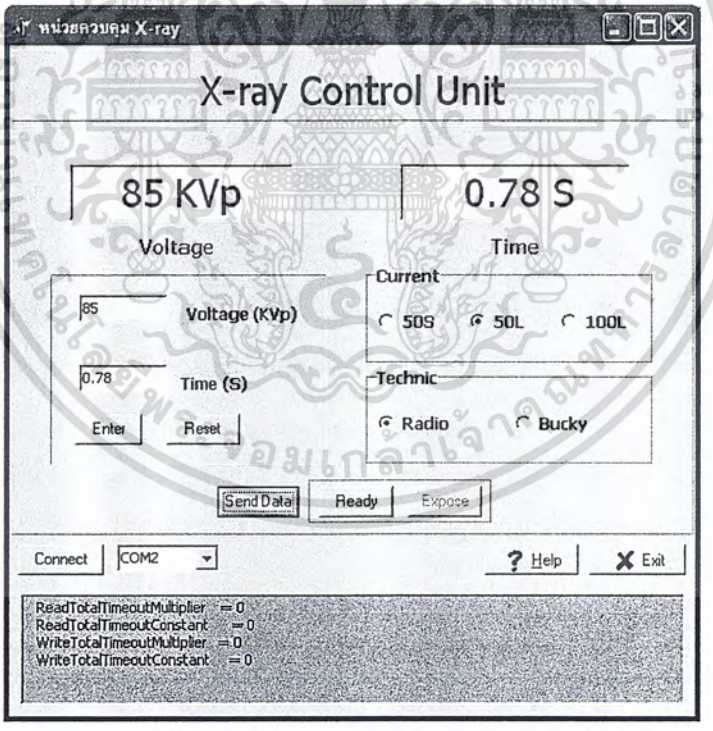
รูปที่ 4.1 แสดงส่วนแผงควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะพบว่าส่วนควบคุมการทำงานจะแบ่งออกเป็น 4 ส่วนใหญ่ๆ คือ

1. ส่วนคีย์บอร์ด ทำหน้าที่รับข้อมูลตลอดจนคำสั่งในการทำงาน
2. ส่วนแสดงผล ประกอบด้วยการแสดงผล 3 ส่วนคือ
   2.1 แสดงค่า KVp ซึ่งเป็นค่าแรงดันที่ใช้ในการเอ็กซเรย์
   2.2 แสดงเวลาที่ใช้ในการเอ็กซเรย์
   2.3 แสดงตัวเลือกค่ากระแสที่ใช้ในการเอ็กซเรย์และขนาดของไส้หลอดที่ใช้ โดยมี 3 ตัวเลือกคือ
   - 50S ค่ากระแส 50 mA ไส้หลอดเล็ก
   - 50L ค่ากระแส 50 mA ไส้หลอดใหญ่
   - 100L ค่ากระแส 100 mA ไส้หลอดใหญ่

   2.4 แสดงตัวเลือกเทคนิคที่ใช้ในการเอ็กซเรย์ โดยแบ่งเป็น 2 ชนิดคือ
   - Radio
   - Bucky



รูปที่ 4.2 แสดงส่วนควบคุมการทำงานโดยใช้คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งานหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์จะมีขั้นตอนต่างๆ ดังต่อไปนี้ โดยขั้นแรกจะเป็นการเลือกโหมดในการทำงานว่าจะเป็นการควบคุมโดยคอมพิวเตอร์หรือจะทำการควบคุมจากคีย์บอร์ด

การใช้การควบคุมผ่านทางเครื่องคอมพิวเตอร์

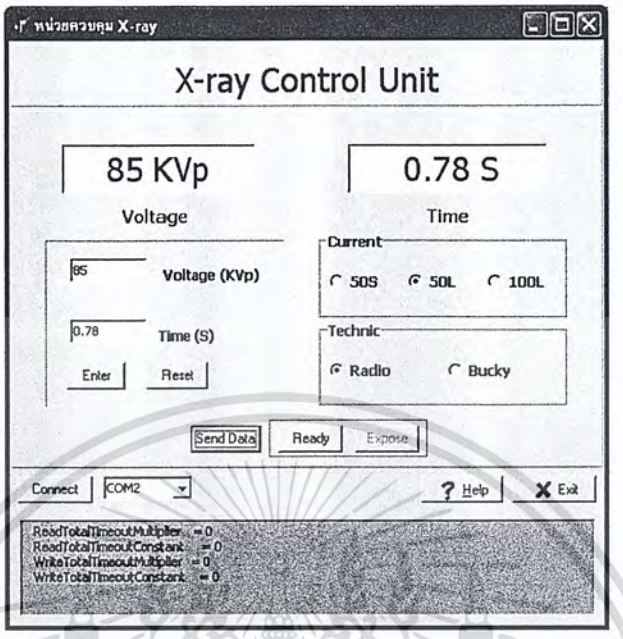
1. เลือกพอร์ตสื่อสารโดยสามารถเลือกเป็น COM1 หรือ COM2
2. ทำการเปิดพอร์ตสื่อสารโดยการกดปุ่ม Connect
3. ทำการป้อนค่าแรงดัน (KVp) และเวลา (Time)
4. ทำการกดปุ่ม Enter เพื่อยืนยันค่าต่างๆ ที่ได้เลือกไว้
5. ทำการเลือกค่ากระแสและไส้หลอดที่จะใช้ในการเอ็กซเรย์รวมทั้งเทคนิคในการเอ็กซเรย์
6. ทำการส่งข้อมูลผ่านพอร์ตอนุกรมโดยการกดปุ่ม Send Data
7. ทำการกดปุ่ม Ready และหลังจากนั้น 2 วินาทีก็จะสามารถทำการ Expose เพื่อทำการถ่ายเอ็กซเรย์ได้

การควบคุมโดยใช้แผงควบคุม (ชุดควบคุมโดยคีย์บอร์ด)

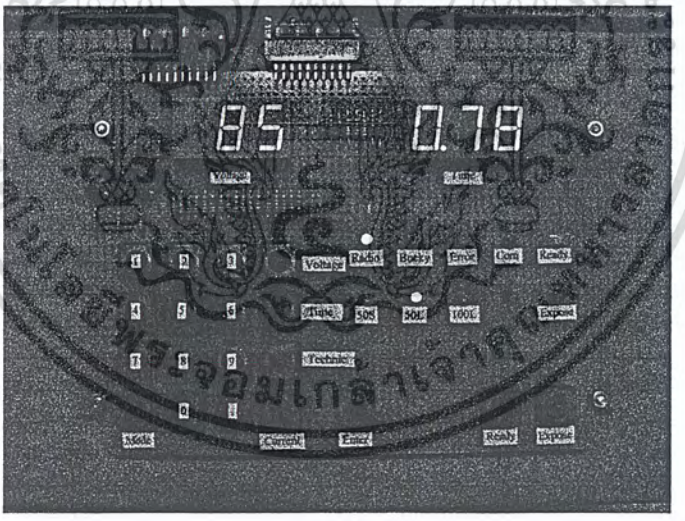
1. กดปุ่ม Enter เพื่อเตรียมความพร้อมในการใช้งาน
2. กดเลือกหน้าที่ในการทำงาน ซึ่งมี 4 ตัวเลือกคือ Voltage, Time, Current และ Technic ซึ่งในการใช้งานสามารถเลือกหน้าที่ใดก่อนก็ได้ ในกรณีที่มีการเลือกค่าแรงดันที่จะใช้ในการเอ็กซเรย์ก่อนให้กดปุ่ม Voltage 1 ครั้งเพื่อทำการเลือกหน้าที่ในการทำงานตามด้วยค่าแรงดันที่ต้องการซึ่งจะอยู่ในช่วง 35 KVp – 113 KVp โดยการกดที่ปุ่มตัวเลขต่างๆ ส่วนการเลือกเวลา(Time)ที่ใช้ในการถ่ายเอ็กซเรย์จะมีวิธีการเช่นเดียวกับการเลือกค่าแรงดันที่ใช้ในการถ่าย สำหรับค่ากระแสที่ใช้ในการเอ็กซ์เรย์และเทคนิคที่ใช้ในการเอ็กซเรย์จะมีวิธีการใช้ที่เหมือนกันคือ กดปุ่ม Current หรือ Technic 1 ครั้งก่อนเพื่อเลือกหน้าที่ในการทำงานและกดปุ่มเดิมซ้ำไปเรื่อยๆ จนกว่าจะได้ค่าที่ต้องการ (โดยการทำงานของทั้ง 2 ปุ่มนี้จะทำงานในลักษณะวนลูปไปเรื่อยๆ) ในการเลือกค่าแรงดัน, เวลา, กระแส และเทคนิคที่ใช้ในการเอ็กซเรย์ เมื่อได้ค่าที่ต้องการแล้วจะต้องกดปุ่ม Enter ทุกครั้งเพื่อเป็นการยืนยันคำสั่งในการทำงาน

โดยในรูปที่ 4.3 ถึง 4.4 แสดงการเลือกค่าแรงดัน, เวลา, กระแสและเทคนิคที่ใช้ในการเอ็กซเรย์ที่ค่าต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(A)

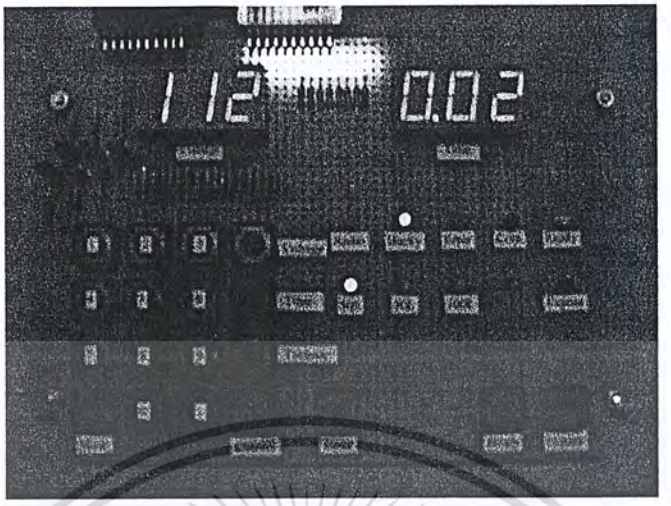


(B)

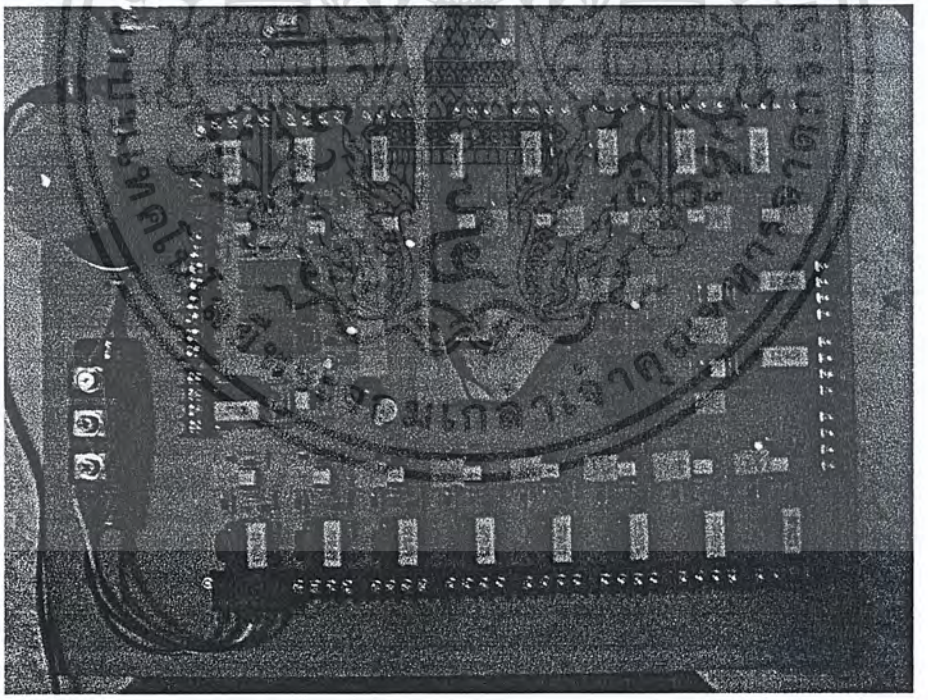
รูปที่ 4.3 **(A)** แสดงการควบคุมจากคอมพิวเตอร์เพื่อเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 85KVp, เวลาที่ใช้ในการเอ็กซเรย์เท่ากับ 0.78 S, ค่ากระแสและเทคนิคที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 50L และ Radio ตามลำดับ

(B) แสดงผลการเลือกค่าที่ส่งมาจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

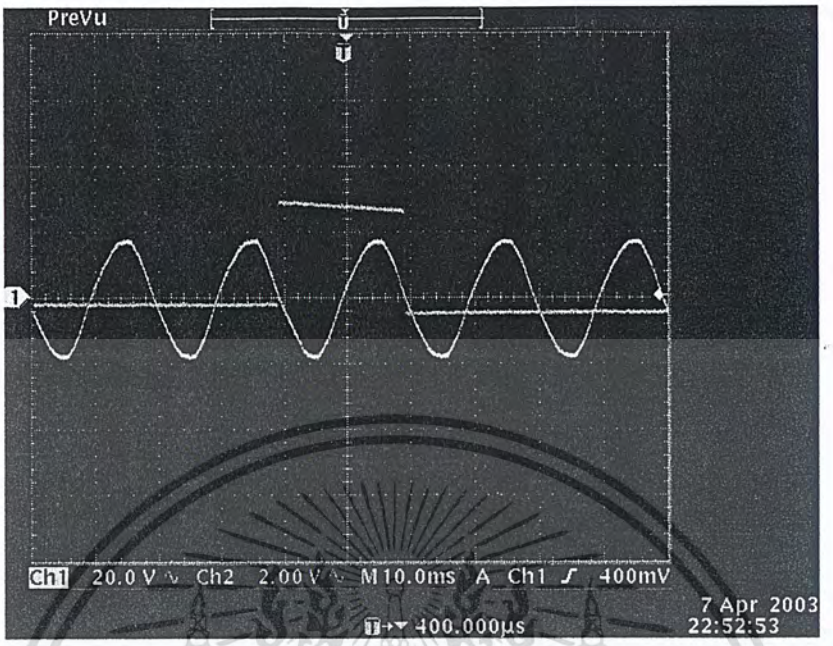


รูปที่ 4.4 แสดงการควบคุมจากคีย์บอร์ดเพื่อเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 112 KVp, เวลาที่ใช้ในการเอ็กซเรย์เท่ากับ 0.02 S, ค่ากระแสและเทคนิคที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 50S และ Bucky ตามลำดับ

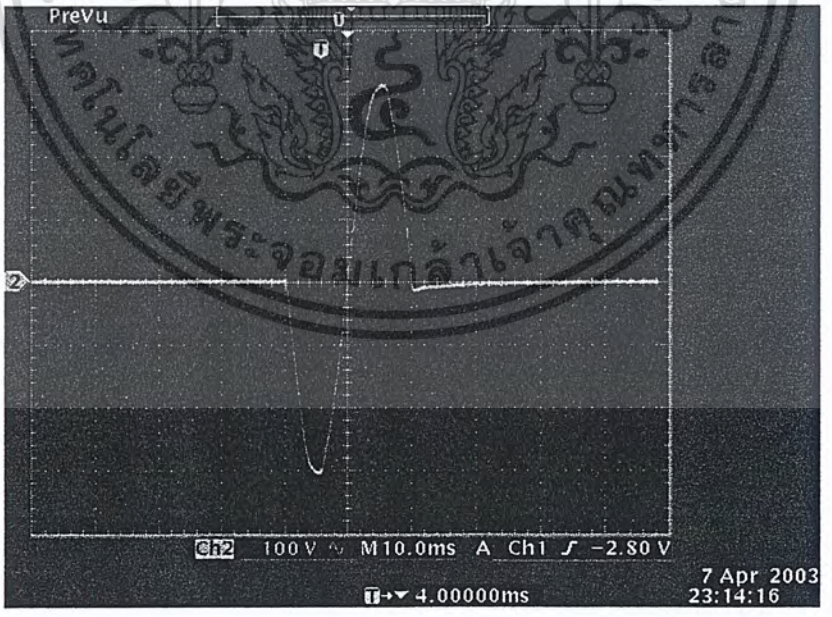


รูปที่ 4.5 แสดงการค่าแรงดันที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 37 KVp, ค่ากระแสและเทคนิคที่ใช้ในการเอ็กซเรย์มีค่าเท่ากับ 100L และ Bucky ตามลำดับ เพื่อแสดงการทริก SCR ในการเลือกขดของหม้อแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

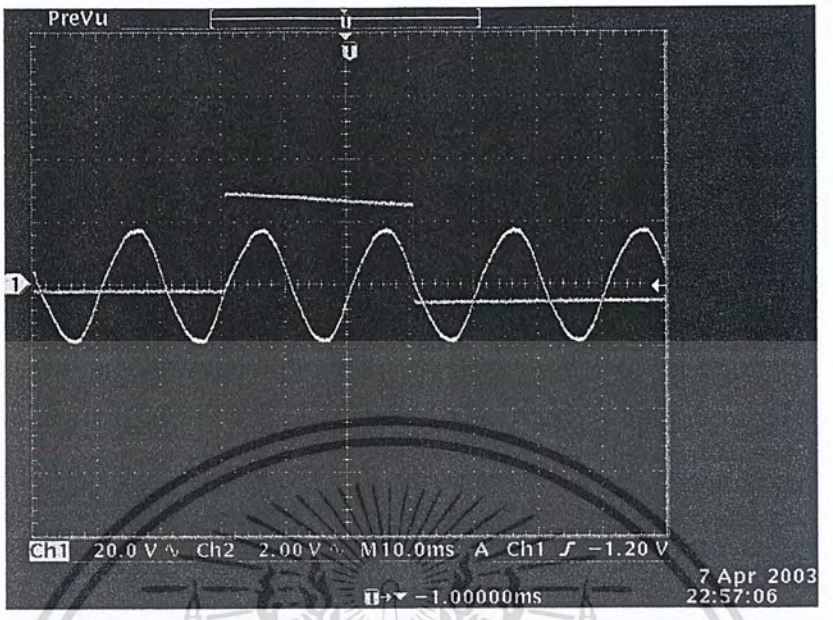


รูปที่ 4.6 แสดงสัญญาณที่ใช้ในการทริก SCR เพื่อควบคุมเวลาที่ใช้ในการถ่ายเอ็กซเรย์ โดยที่ Ch1 แสดงสัญญาณความถี่ 50 Hz ของไฟฟ้ากระแสสลับที่นำมาเปรียบเทียบและ Ch2 แสดงสัญญาณทริก SCR ที่มีค่าเวลาในการถ่ายเอ็กซเรย์เท่ากับ 0.02 S

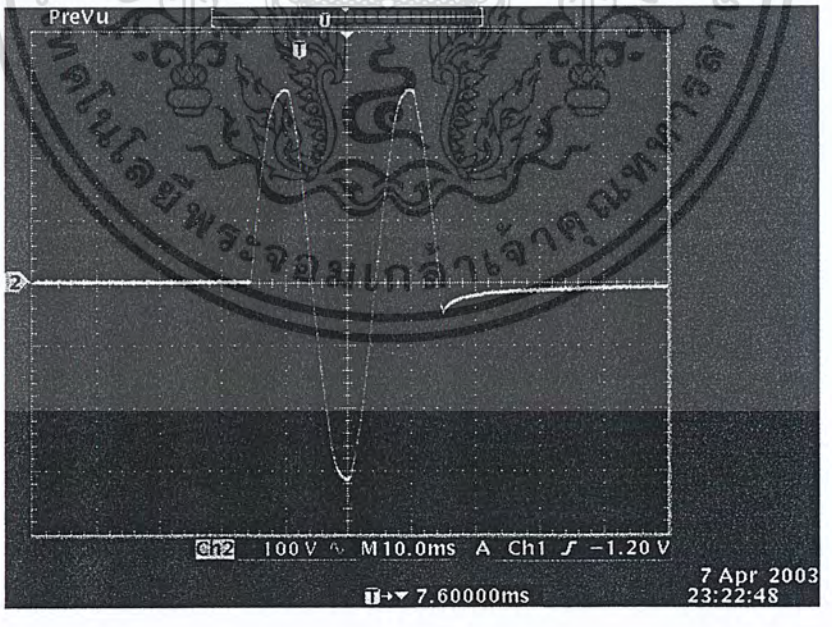


รูปที่ 4.7 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.02 S ที่ได้จากการทริกที่ SCR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

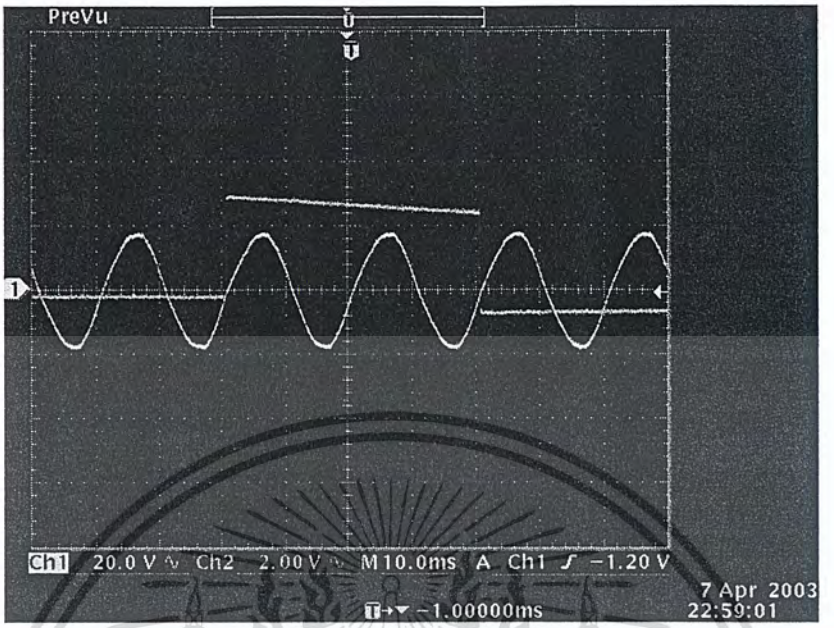


รูปที่ 4.8 แสดงสัญญาณที่ใช้ในการทริก SCR เพื่อควบคุมเวลาที่ใช้ในการถ่ายเอ็กซเรย์ โดยที่ Ch1 แสดงสัญญาณความถี่ 50 Hz ของไฟฟ้ากระแสสลับที่นำมาเปรียบเทียบและ Ch2 แสดงสัญญาณทริก SCR ที่มีค่าเวลาในการถ่ายเอ็กซเรย์เท่ากับ 0.03 S

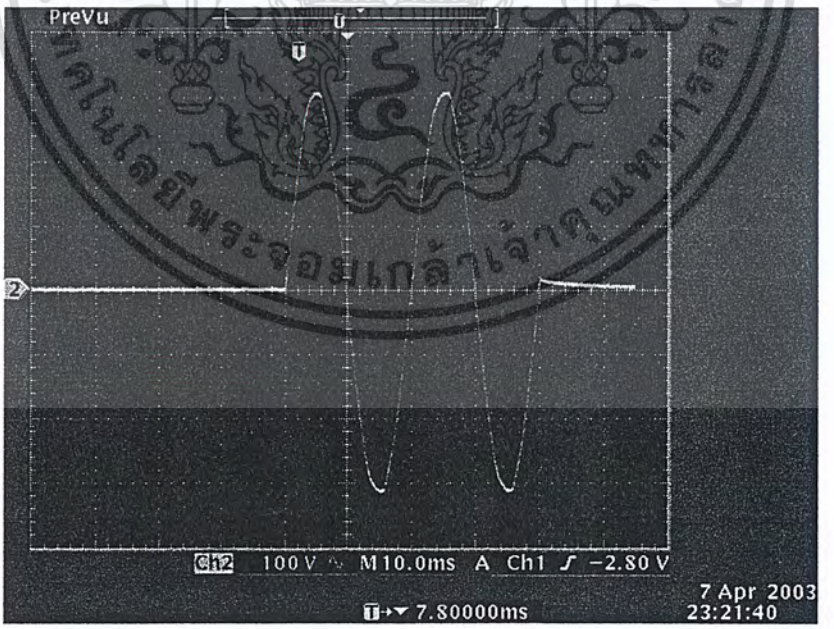


รูปที่ 4.9 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.03 S ที่ได้จากการทริกที่ SCR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

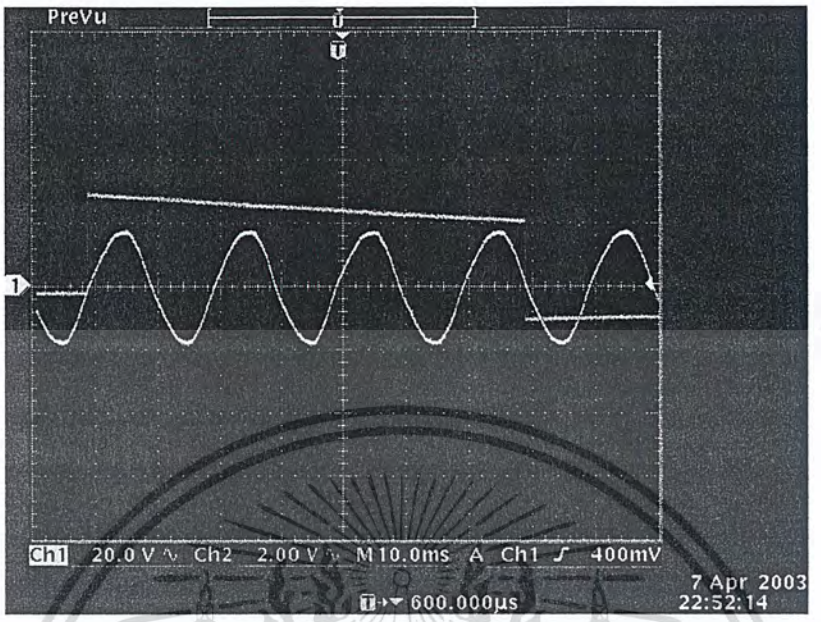


รูปที่ 4.10 แสดงสัญญาณที่ใช้ในการทริก SCR เพื่อควบคุมเวลาที่ใช้ในการถ่ายเอ็กซเรย์ โดยที่ Ch1 แสดงสัญญาณความถี่ 50 Hz ของไฟฟ้ากระแสสลับที่นำมาเปรียบเทียบและ Ch2 แสดงสัญญาณทริก SCR ที่มีค่าเวลาในการถ่ายเอ็กซเรย์เท่ากับ 0.04 S

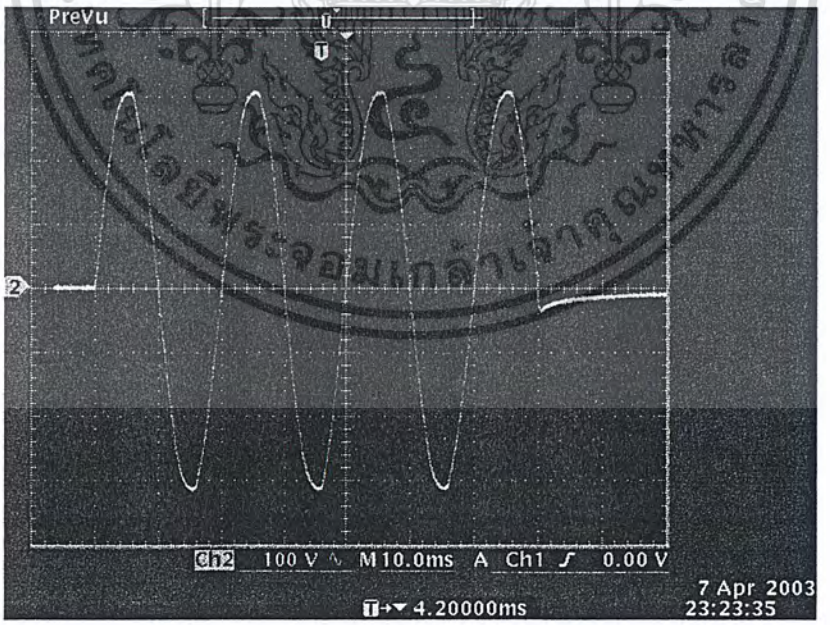


รูปที่ 4.11 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.04 S ที่ได้จากการทริกที่ SCR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงสัญญาณที่ใช้ในการทริก SCR เพื่อควบคุมเวลาที่ใช้ในการถ่ายเอ็กซเรย์ โดยที่ Ch1 แสดงสัญญาณความถี่ 50 Hz ของไฟฟ้ากระแสสลับที่นำมาเปรียบเทียบและ Ch2 แสดงสัญญาณทริก SCR ที่มีค่าเวลาในการถ่ายเอ็กซเรย์เท่ากับ 0.07 S



รูปที่ 4.13 แสดงแรงดันที่ตกคร่อมโหลดซึ่งเป็นหม้อแปลงในช่วงเวลา 0.07 S ที่ได้จากการทริกที่ SCR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

จากการทดลองหน่วยควบคุมเอ็กซเรย์โดยผ่านทางคอมพิวเตอร์นี้ ในเบื้องต้นของการทดลองจะยังไม่มีการเชื่อมต่อกับคอมพิวเตอร์เนื่องจากเครื่องเอ็กซเรย์ในระบบเดิมจะเป็นการควบคุมด้วยระบบกลไกในหลายๆ หน้าที่การทำงาน ดังนั้นจึงมีความจำเป็นที่จะต้องเปลี่ยนการควบคุมด้วยระบบกลไกของระบบเดิมให้มาอยู่ในรูปของการควบคุมด้วยระบบเชิงตัวเลข เพื่อความสะดวกในการเชื่อมต่อกับคอมพิวเตอร์และจากการออกแบบได้ทำการออกแบบให้สามารถควบคุมการทำงานของเครื่องได้ทั้งจากตัวเครื่องโดยตรงและผ่านทางคอมพิวเตอร์เพื่อความสะดวกในการใช้งาน จากผลการทดลองพบว่าหน่วยควบคุมเอ็กซเรย์ที่ได้จากการออกแบบสามารถทำงานได้อย่างถูกต้องตรงตามหลักการและหน้าที่การใช้งานต่างๆ ตามที่ได้ออกแบบไว้ทั้งในส่วนของการเลือกขดหม้อแปลงเพื่อเลือกค่าแรงดันที่ใช้ในการเอ็กซเรย์และส่วนของการควบคุมเวลาที่ใช้ในการถ่ายเอ็กซเรย์ซึ่งมีความแม่นยำในการทำงานสูงดังที่แสดงไว้ในส่วนของผลการทดลอง

ในการทดลองได้ทำการออกแบบหน่วยควบคุมเอ็กซเรย์โดยใช้อุปกรณ์ FPGA ของบริษัท ALTERA ตระกูล ACEX 1K ซึ่งโครงสร้างภายในจะเป็น SRAM ซึ่งไม่สามารถเก็บข้อมูลของโปรแกรมการทำงานต่างๆ ไว้ได้เมื่อไม่มีไฟจ่ายให้กับตัวอุปกรณ์ ดังนั้นจึงได้มีการนำอุปกรณ์ที่เก็บข้อมูลที่ใช้ในการโปรแกรม (Configuration Device) มาใช้งานร่วมกับอุปกรณ์ FPGA เพื่อแก้ไขปัญหาในการที่ต้องโปรแกรมการทำงานใหม่ทุกครั้งที่มีการจ่ายไฟเลี้ยงให้กับตัวอุปกรณ์ซึ่งทำให้มีความสะดวกในการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Delphi ที่ใช้ในการสร้างหน่วยควบคุมโดยผ่านทางคอมพิวเตอร์

```
unit uXray;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons,uComm32,uComProc;

type
  TXray = class(TForm)
    lbHead: TLabel;
    edVoltage: TEdit;
    edShowvoltage: TEdit;
    edTime: TEdit;
    edShowtime: TEdit;
    lbVoltage: TLabel;
    lbTime: TLabel;
    Bevel2: TBevel;
    bEnter: TButton;
    bClear: TButton;
    lbshowvoltage: TLabel;
    lbShowtime: TLabel;
    Bevel1: TBevel;
    bbExit: TBitBtn;
    Bevel3: TBevel;
    bbHelp: TBitBtn;
    bSendData: TButton;
    groupboxExpose: TGroupBox;
    bReady: TButton;
    bExpose: TButton;
    Timer1: TTimer;
    rgTechnic: TRadioGroup;
    rgCurrent: TRadioGroup;
    cmbSelecPort: TComboBox;
    bConnect: TButton;
    Memo1: TMemo;
    procedure bEnterClick(Sender: TObject);
    procedure bClearClick(Sender: TObject);
    procedure bbExitClick(Sender: TObject);
    procedure edVoltageKeyDown(Sender: TObject; var Key: Word;
          Shift: TShiftState);
    procedure edTimeKeyDown(Sender1: TObject; var Key1: Word;
          Shift1: TShiftState);
    procedure FormCreate(Sender: TObject);
    procedure bReadyClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure bExposeClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure bSendDataClick(Sender: TObject);
    procedure bbHelpClick(Sender: TObject);
    procedure bConnectClick(Sender: TObject);
  private
    { Private declarations }
     commInfo : TcommInfo;
     commProc : Tcomm32;
    procedure KeyCheck(CHKey:Word;var DetResu:Boolean);
    procedure KeyCheck1(CHKey1:Word;var DetResu1:Boolean);
  public
    { Public declarations }
  end;

var
  Xray: TXray;
  v1,v2,v3,T1,T2,T3:string;
  kvp:integer;
  time:extended;

implementation

uses Unit2;

{$R *.DFM}

procedure TXray.bEnterClick(Sender: TObject); //enter to show data--------
var    time1,time2 ,v1i,v2i,v3i:integer;
       time_time1:extended;
       stime:string;
begin
       kvp:=StrTOInt(edVoltage.Text);
       if ((kvp>=35)and(kvp<=113)) then
          begin
             edshowvoltage.Text:='   '+IntToStr(kvp)+' KVp';
//---separate KVp voltage to V1,V2 and V3 for send voltage-------------------
             if (kvp<>100)then
                  begin
                  if (kvp<100) then
                     begin
                       v1i:=kvp div 10;
                       v1:= IntTostr(v1i);
                       v2i:=kvp mod 10;
                       v2:= IntTostr(v2i);
                       v3:='0';
                     end
                  else
                     if (kvp<110) then
                       begin
                         v1:='1';
                         v2:='0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    v3i:=kvp mod 10;
                    v3:= FloatTostr(v3i);
                  end
                else
                  begin
                    v1:='1';
                    v2:='1';
                    v3i:=kvp mod 11;
                    v3:= FloatTostr(v3i);
                  end
              end
            else
              begin
                v1:='1';
                v2:='0';
                v3:='0';
              end
          end
        else
          begin
            edshowvoltage.Text:='  KVp Error ';
            v1:='0';
            v2:='0';
            v3:='0';
          end; //-------------------------------------------------------------------

        time:=StrTOFloat(edTime.Text);
        if ((time>=0.02)and(time<=5)) then
          begin
            edshowtime.Text:='    '+FloatToStr(time)+' S';
//--separate Time to T1,T2 and T3 for send voltage-----------------------------
            time_time1:=time*100;
            stime:=FloatToStr(time_time1);
            time1:=StrToInt(stime);
            time2:=time1 mod 100;
            if (time1<100) then
              if (time1<=9) then
                begin
                  T1:='0';
                  T2:='0';
                  T3:=IntToStr(time2);
                end
              else
                begin
                  T1:='0';
                  T2:=IntToStr(time1 div 10);
                  T3:=IntToStr(time1 mod 10);
                end
            else
              if (time2<=9) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            begin
              T1:=IntToStr(time1 div 100);
              T2:='0';
              T3:=IntToStr(time2 mod 10);
            end
          else
            begin
              T1:=IntToStr(time1 div 100);
              T2:=IntToStr(time2 div 10);
              T3:=IntToStr(time2 mod 10);
            end
      end
    else
      begin
        edshowtime.Text:='  Time Error ';
        T1:='0';
        T2:='0';
        T3:='0';
      end;
    if (edShowvoltage.Text='    KVp Error ') or
      (edShowtime.Text='   Time Error ' )then
       bSendData.Enabled:=False
    else
       bSendData.Enabled:=True;
end;//-------------------------------------------------------------------------------

procedure TXray.bClearClick(Sender: TObject);//clear data
begin
    edvoltage.text:='000';
    edshowvoltage.text:='';
    edtime.text:='0.00';
    edshowtime.text:='';
    rgCurrent.ItemIndex:=-1;
    rgTechnic.ItemIndex:=-1;
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
end;//-------------------------------------------------------------------------------

procedure TXray.bbExitClick(Sender: TObject);
begin
    close  //close program----------------------------------------------------
end;

// protect  insert Char  procedure Keycheck ----------------------------------
procedure TXray.KeyCheck(CHKey:Word;var DetResu:Boolean);
var    KeyID : extended;
begin
    keyID:= Ord(CHkey);
    if(((KeyID>=32)and(KeyID<48))or
      ((KeyID>=58)and(KeyID<96))or
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ((KeyID>=106)and(KeyID<112))or
        ((KeyID>=128)and(KeyID<255)))then
        begin
         ShowMessage('¡ÃØ³Ò»éÍ¹à»ç¹µÑÇàÅ¢¨Ó¹Ç¹àµçÁµÑé§áµè 35 ¶Ö§ 113');
         DetResu:=False;
        end
       else
        DetResu:=True;
end;

procedure TXray.edVoltageKeyDown(Sender: TObject; var Key: Word;
 Shift: TShiftState);
var    MyDet:boolean;
begin
      KeyCheck(Key,myDet);
      if Mydet=False then
      edVoltage.text:='';
end;

procedure TXray.KeyCheck1(CHKey1:Word;var DetResu1:Boolean);
var    KeyID1 : extended;
begin
      keyID1:= Ord(CHkey1);
      if(((KeyID1>=32)and (KeyID1<48))or
      ((KeyID1>=58)and(KeyID1<96))or
      ((KeyID1>=106)and (KeyID1<110))or
      ((KeyID1>=128)and(KeyID1<255)))then
       begin
        ShowMessage('¡ÃØ³Ò»éÍ¹à»ç¹µÑÇàÅ¢·È¹ÔÂÁµÑ§áµè 0.02 ¶Ö§ 5');
        DetResu1:=False;
       end
      else
       DetResu1:='True;
end;

procedure TXray.edTimeKeyDown(Sender1: TObject; var Key1: Word;
 Shift1: TShiftState);
var    MyDet1:boolean;
begin
      KeyCheck1(Key1,myDet1);
      if Mydet1=False then
      edTime.text:='';
end; //-------------------------------------------------------------------------------

procedure TXray.FormCreate(Sender: TObject);
begin
  bEnter.Enabled:=False;
  bReady.Enabled:=False;
  bExpose.Enabled:=False;
  bSendData.Enabled:=False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
procedure TXray.bReadyClick(Sender: TObject);
var r : shortString; rP : pChar ;
begin
  Timer1.Enabled:=True;
  if ( commProc=nil ) then exit;   // Send Signal Ready ----------------------------
      r:='j' ; rP:=@r[1];
      commProc.SendData( rP, length( r ) );
end;

procedure TXray.Timer1Timer(Sender: TObject);
begin
  bExpose.Enabled:=True;
end;

procedure TXray.bExposeClick(Sender: TObject);
var e : shortString; eP : pChar ;
begin
  Timer1.Enabled:=False;
  if ( commProc=nil ) then exit;  // Send Signal Expose ------------------------
     begin
      e:='o' ; eP:=@e[1];
      commProc.SendData( eP, length( e ) );
     end;
   bExpose.Enabled:=False;

end;

procedure TXray.bSendDataClick(Sender: TObject);
var current,technic:string ;
   s : shortString; sP : pChar ;
begin
    bExpose.Enabled:=False;
    if (rgCurrent.ItemIndex=-1)then
      begin
       showMessage('¡ÃØ³ÒàÅ×Í¡¤èÒ¡ÃÐáÊ·Õè¨Ðãªéã¹¡ÒÃ X-ray');
       Exit;
      End
    else
      if (rgTechnic.ItemIndex=-1)then
        Begin
         showMessage('¡ÃØ³ÒàÅ×Í¡ Technic ·Õè¨Ðãªéã¹¡ÒÃ X-ray');
         Exit;
        End
      else
        if rgTechnic.ItemIndex=0 then
          technic:='1';
        if rgTechnic.ItemIndex=1 then
          technic:='2';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if rgCurrent.ItemIndex=0 then
  current:='3';
if rgCurrent.ItemIndex=1 then
  current:='2';
if rgCurrent.ItemIndex=2 then
  begin
    current:='1';
    if (kvp>78)and(time>=0.02)then
       begin
         ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
                +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
         bSendData.Enabled:=False;
         bReady.Enabled:=False;
         edVoltage.Text:='000';
         edTime.Text:='0.00';
         Exit;
       end
    else
    if (kvp=78)and(time>0.02)Then
      begin
        ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
                +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
        bSendData.Enabled:=False;
        bReady.Enabled:=False;
        edVoltage.Text:='000';
        edTime.Text:='0.00';
        Exit;
      end
    else
    if (kvp=77)and(time>0.03)Then
      begin
        ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
                +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
        bSendData.Enabled:=False;
        bReady.Enabled:=False;
        edVoltage.Text:='000';
        edTime.Text:='0.00';
        Exit;
      end
    else
    if (kvp=76)and(time>0.04)Then
       begin
         ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
                +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
         bSendData.Enabled:=False;
         bReady.Enabled:=False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=75)and(time>0.05)Then
   begin
     ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
           +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
           +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
     bSendData.Enabled:=False;
     bReady.Enabled:=False;
     edVoltage.Text:='000';
     edTime.Text:='0.00';
     Exit;
   end
else
if (kvp=74)and(time>0.06)Then
   begin
     ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
           +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
           +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
     bSendData.Enabled:=False;
     bReady.Enabled:=False;
     edVoltage.Text:='000';
     edTime.Text:='0.00';
     Exit;
   end
else
if (kvp=73)and(time>0.075)Then
   begin
     ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
           +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
           +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
     bSendData.Enabled:=False;
     bReady.Enabled:=False;
     edVoltage.Text:='000';
     edTime.Text:='0.00';
     Exit;
   end
else
if (kvp=72)and(time>0.08)Then
   begin
     ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
           +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
           +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
     bSendData.Enabled:=False;
     bReady.Enabled:=False;
     edVoltage.Text:='000';
     edTime.Text:='0.00';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Exit;
   end
else
if (kvp=71)and(time>0.09)Then
   begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
   end
else
if (kvp=70)and(time>0.1)Then
   begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
   end
else
if (kvp=69)and(time>0.125)Then
   begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
   end
 else
 if (kvp=68)and(time>0.15)Then
   begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
   end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
if (kvp=67)and(time>0.175)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=66)and(time>0.21)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=65)and(time>0.25)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=64)and(time>0.28)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=63)and(time>0.33)Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=62)and(time>0.36)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=61)and(time>0.41)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=60)and(time>0.46)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
    edTime.Text:='0.00';
    Exit;
  end
else
if (kvp=59)and(time>0.6)Then
  begin
    ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=58)and(time>0.7)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=57)and(time>0.8)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=56)and(time>0.9)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=55)and(time>1.1)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=54)and(time>1.3)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=53)and(time>1.45)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=52)and(time>1.65)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=51)and(time>1.8)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=50)and(time>2)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=49)and(time>2.5)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=48)and(time>2.8)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
else
if (kvp=47)and(time>3.2)Then
    begin
      ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Exit;
      end
    else
    if (kvp=46)and(time>3.6)Then
      begin
        ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
              +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
              +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
        bSendData.Enabled:=False;
        bReady.Enabled:=False;
        edVoltage.Text:='000';
        edTime.Text:='0.00';
        Exit;
      end
    else
    if (kvp=45)and(time>4.25)Then
      begin
        ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
              +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
              +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
        bSendData.Enabled:=False;
        bReady.Enabled:=False;
        edVoltage.Text:='000';
        edTime.Text:='0.00';
        Exit;
      end
    else
    if (kvp>=44)and(time>4.6)Then
      begin
        ShowMessage('¡ÃÐáÊ 100 mA ,áÃ§´Ñ¹ '
              +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
              +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
        bSendData.Enabled:=False;
        bReady.Enabled:=False;
        edVoltage.Text:='000';
        edTime.Text:='0.00';
        Exit;
      end
  end
else
bReady.Enabled:=True;

if (kvp>=100)and(time>2)Then
  begin
    ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
          +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
          +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
    bSendData.Enabled:=False;
    bReady.Enabled:=False;
    edVoltage.Text:='000';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=99)and(time>2.5)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=98)and(time>2.7)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=97)and(time>2.9)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=96)and(time>3.1)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end
  else
  if (kvp=95)and(time>3.4)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=54)and(time>3.75)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=93)and(time>4)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
  if (kvp=92)and(time>4.25)Then
    begin
      ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
            +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
            +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
      bSendData.Enabled:=False;
      bReady.Enabled:=False;
      edVoltage.Text:='000';
      edTime.Text:='0.00';
      Exit;
    end
  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (kvp=91)and(time>4.5)Then
          begin
            ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
                  +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                  +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
            bSendData.Enabled:=False;
            bReady.Enabled:=False;
            edVoltage.Text:='000';
            edTime.Text:='0.00';
            Exit;
          end
        else
        if (kvp>=90)and(time>4.8)Then
          begin
            ShowMessage('¡ÃÐáÊ 50 mA ,áÃ§´Ñ¹ '
                  +IntToStr(kvp)+' KVp áÅÐ Time à»ç¹ '
                  +FloatToStr(time)+' S à¡Ô¹ Rating Chart');
            bSendData.Enabled:=False;
            bReady.Enabled:=False;
            edVoltage.Text:='000';
            edTime.Text:='0.00';
            Exit;
          end
        else
      bReady.Enabled:=True;
// Send Data----------------------------------------------------------------------
    if ( commProc=nil ) then exit;
      s:= v1; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= v2; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= v3; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= T1; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= T2; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= T3; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= current; sP:=@s[1];
      commProc.SendData( sP, length( s ) );sleep(100);
      s:= Technic; sP:=@s[1];
      commProc.SendData( sP, length( s ) );
end;

procedure TXray.bbHelpClick(Sender: TObject);
begin
    fmHelp.Show; // link to help ----------------------------------------------------
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TXray.bConnectClick(Sender: TObject);
begin
   if (cmbSelecPort.ItemIndex=-1) then
      begin
        ShowMessage ('¡ÃØ³ÒàÅ×Í¡ Port Ê×èÍÊÒÃ');
        Exit;
      end;
   with commInfo do
     begin
       commPort:=cmbSelecPort.ItemIndex+1;
       rxBuffer:=8072;
       txBuffer:=2048;
       baudRate:=CBR_4800;
       byteSize:=8;
       parity:=NOPARITY;
       stopBits:=ONESTOPBIT;
     end;
   commProc:=Tcomm32.Create( memo1.lines, commInfo );
   bEnter.Enabled:=True;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unit uComm32 ใช้ในการ open port

```
unit uComm32;

interface

uses Classes, Dialogs, Messages, SysUtils, Windows;

const
 // define max comm. port possible
 MAX_COMM_PORT = 4;

type

 TCommInfo = record
  hCommFile : THandle;
  commPort  : byte;
  rxBuffer  : integer;
  txBuffer  : integer;
  baudRate  : integer;
  byteSize  : integer;
  parity    : integer;
  stopBits  : integer;
  errorFlg  : boolean;
 end;

 Tcomm32 = class( tobject )
 public
    myCommInfo : TCommInfo;
   Constructor Create( dspStr:Tstrings; commInfo:TCommInfo );
   Destructor  Free;

   function  SendData(dataP:pointer; dataLen:dword) : integer;
   function  ReadData(dataP:pointer; dataLen:dword) : integer;

 private
     commDcb     : Tdcb;
     commProp    : TCommProp;
     evtMask     : dword;
     commTimeOuts : TCommTimeouts;
     dispStr     : Tstrings;
     overlapWrite : TOverlapped;
     overlapRead  : TOverlapped;
    procedure Echo( content : shortString );

    procedure DispComDCB( var dspDcb : Tdcb );
    procedure DispComProperties( var dspProp : TCommProp );
    procedure DispComTimeouts( var dspComTmeOut : TCommTimeouts );

    procedure VerifyCommInfo( var commInfo : TCommInfo );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure OpenComm32( var commInfo : TCommInfo );
    procedure CloseComm32;
  end;

implementation
//---------------------------------------------------------

constructor Tcomm32.Create(dspStr:Tstrings; commInfo:TCommInfo);
begin
  inherited Create;
  dispStr:=dspStr;
  fillchar( commDcb, sizeof( Tdcb ),0 );
  fillchar( commProp, sizeof( TCommProp ),0 );
  fillchar( commTimeOuts, sizeof( TCommTimeouts ),0 );
  fillchar( overlapWrite, sizeof( TOverlapped ),0 );
  fillchar( overlapRead, sizeof( TOverlapped ),0 );
  OpenComm32( commInfo );
end;

destructor Tcomm32.Free;
begin
  CloseComm32;
end;

//---------------------------------------------------------

procedure Tcomm32.Echo( content : shortString );
begin
  if ( dispStr=nil ) then exit;
  if ( dispStr.count>300 ) then dispStr.delete(0);
  dispStr.add( content );
end;

procedure Tcomm32.DispComDCB( var dspDcb : Tdcb );
begin
  with dspDcb do
  begin
    Echo( 'Communications Device Control Block information : ' );
    Echo( '  DCBLength = '+IntToStr( DCBLength ) );
    Echo( '  BaudRate  = '+IntToStr( BaudRate ) );
    Echo( '  Flags     = '+IntToStr( Flags ) );
    Echo( '  wReserved = '+IntToStr( wReserved ) );
    Echo( '  XonLim    = '+IntToStr( XonLim ) );
    Echo( '  XoffLim   = '+IntToStr( XoffLim ) );
    Echo( '  ByteSize  = '+IntToStr( byteSize ) );
    Echo( '  Parity    = '+IntToStr( Parity ) );
    Echo( '  StopBits  = '+IntToStr( StopBits ) );
    Echo( '  XonChar   = '''+XonChar+'''' );
    Echo( '  XffChar   = '''+XoffChar+'''' );
    Echo( '  ErrorChar = '''+errorChar+'''' );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Echo( '  EofChar   = "'+EofChar+"" );
  Echo( '  EvtChar   = "'+EvtChar+"" );
  Echo( '  wReserved1= '+IntToStr( wReserved1 ) ); Echo( ' ' );
 end;
end;

procedure Tcomm32.DispComProperties( var dspProp : TCommProp );
begin
 with dspProp do
 begin
  Echo('Communications Properties : ' );
  Echo('  wPacketLength     = '+IntToStr(wPacketLength));
  Echo('  wPacketVersion    = '+IntToStr(wPacketVersion));
  Echo('  dwServiceMask     = '+IntToStr(dwServiceMask));
  Echo('  dwReserved1       = '+IntToStr(dwReserved1));
  Echo('  dwMaxTxQueue      = '+IntToStr(dwMaxTxQueue));
  Echo('  dwMaxRxQueue      = '+IntToStr(dwMaxRxQueue));
  Echo('  dwMaxBaud         = '+IntToStr(dwMaxBaud));
  Echo('  dwProvSubType     = '+IntToStr(dwProvSubType));
  Echo('  dwProvCapabilities = '+IntToStr(dwProvCapabilities));
  Echo('  dwSettablePrrams  = '+IntToStr(dwSettableParams));
  Echo('  dwSettableBaud    = '+IntToStr(dwSetTableBaud));
  Echo('  wSettableData     = '+IntToStr(wSetTableData));
  Echo('  wSetTableStopParity= '+IntToStr(wSetTableStopParity));
  Echo('  dwCurrentTxQueue  = '+IntToStr(dwCurrentTxQueue ));
  Echo('  dwCurrentRxQueue  = '+IntToStr(dwCurrentRxQueue ));
  Echo('  dwProvSpec1       = '+IntToStr(dwProvSpec1));
  Echo('  dwProvSpec2       = '+IntToStr(dwProvSpec2));
  Echo(' ' );
 end;
end;

procedure Tcomm32.DispComTimeouts(var dspComTmeOut:TCommTimeouts);
begin
 with dspComTmeOut do
 begin
  Echo('Communications Timeouts information : ' );
  Echo('  ReadIntervalTimeout         = '+
      IntToStr( ReadIntervalTimeout ) );
  Echo('  ReadTotalTimeoutMultiplier  = '+
      IntToStr( ReadTotalTimeoutMultiplier ) );
  Echo('  ReadTotalTimeoutConstant    = '+
      IntToStr( ReadTotalTimeoutConstant ) );
  Echo('  WriteTotalTimeoutMultiplier = '+
      IntToStr( WriteTotalTimeoutMultiplier ) );
  Echo('  WriteTotalTimeoutConstant   = '+
      IntToStr( WriteTotalTimeoutConstant ) );
  Echo(' ' );
 end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-------------------------------------------------------------

procedure Tcomm32.VerifyCommInfo( var commInfo : TCommInfo );
begin
 with commInfo do
 begin
  if ( not commPort in [1..MAX_COMM_PORT] ) then commPort:=1;
  if ( rxBuffer<4096 ) then rxBuffer:=4096;
  if ( txBuffer<1024 ) then txBuffer:=1024;
  if ( baudRate<1 ) then baudRate:=CBR_9600;
  if ( not byteSize in [ 4..8 ] ) then byteSize:=8;
  if ( not parity in [ EVENPARITY,MARKPARITY,NOPARITY,
    ODDPARITY ] ) then parity:=NOPARITY;
  if not stopBits in [ONESTOPBIT,ONE5STOPBITS,TWOSTOPBITS] then
    stopBits:=ONESTOPBIT;
 end;
end;

procedure Tcomm32.OpenComm32( var commInfo : TCommInfo );
var commPortP : pchar; commPortStr,s : ShortString;
begin
 myCommInfo:=commInfo; commPortP:=@commPortStr[1];
 VerifyCommInfo( myCommInfo );
 fillchar( commPortStr, sizeof( ShortString ), 0 );
 with myCommInfo do
 begin
  commPortStr:='COM'+IntToStr( commPort );
  hCommFile:=CreateFile(commPortP, GENERIC_READ+GENERIC_WRITE,
          0, nil, OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);
  // comm. handle is valid?
  if ( hCommFile=INVALID_HANDLE_VALUE ) then
  begin
   s:='Error on open '+commPortStr;
   Echo( s+', error code = '+IntToStr( GetLastError ) );
   ShowMessage( s ); exit;
  end;
  if ( GetFileType( hCommFile )<>FILE_TYPE_CHAR ) then
  begin
   s:='Handle is not COM port.'; Echo( s );
   ShowMessage( s ); exit;
  end;
  // get all comm. properties
  GetCommState( hCommFile, commDcb );
  GetCommProperties( hCommFile, commProp );
  GetCommMask( hCommFile, evtMask );
  GetCommTimeOuts( hCommFile, commTimeOuts );
  // config necessary comm. properties
  SetupComm( hCommFile, rxBuffer, txBuffer );
  commTimeOuts.ReadIntervalTimeout:= MAXDWORD; // set timeout
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  commTimeOuts.ReadTotalTimeoutMultiplier:=0;
  commTimeOuts.ReadTotalTimeoutConstant:=0;
  SetCommTimeouts( hCommFile, commTimeOuts );
  Echo( 'evtmask = '+IntToStr( evtMask ) );
  evtMask:=EV_RXCHAR;
  SetCommMask( hCommFile, evtMask );
  commDcb.baudRate:=baudRate;
  commDcb.parity:=Parity;
  commDcb.stopBits:=StopBits;
  commDcb.byteSize:=byteSize;
  SetCommState( hCommFile, commDcb );
  errorFlg:=false;
  // Display comm. properties
  Echo( commPortStr+' open successfully.' );
  DispComDCB( commDcb );
  DispComProperties( commProp );
  DispComTimeouts( commTimeOuts );
 end;
end;

procedure Tcomm32.CloseComm32;
begin
 if ( myCommInfo.hCommFile>0 ) then
  CloseHandle( myCommInfo.hCommFile );
end;

function Tcomm32.SendData( dataP : pointer;
                dataLen : dword ) : integer;
var byteSend : dword;
begin
 result:=-1; byteSend:=0;
 if ( not WriteFile( myCommInfo.hCommFile, dataP^, dataLen,
  byteSend, @overlapWrite ) ) then exit;
 result:=byteSend;
end;

function Tcomm32.ReadData( dataP : pointer;
                  dataLen : dword ) : integer;
var byteRead : dword; overlapReadP : POverlapped;
  nNumberOfBytesRead : dword; ret : boolean;
begin
 byteRead:=0; overlapReadP:=@overlapRead; nNumberOfBytesRead:=0;
 if ( not ReadFile( myCommInfo.hCommFile, dataP^, dataLen,
  byteRead, overlapReadP ) ) then exit;
 ret:=GetOverlappedResult( myCommInfo.hCommFile, overlapReadP^,
      nNumberOfBytesRead, true );
 result:=byteRead;
end;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit uComProc;

interface

uses
  Classes, Windows;

type
  TcommProcess = class(TThread)
  private
    { Private declarations }
  protected

  public
    constructor Create( suspend : boolean );
  end;

implementation

uses  uXray;

constructor TcommProcess.Create( suspend : boolean );
begin
  freeOnTerminate:=true;
  inherited Create( suspend );
end;
end.
```

Unit2 สร้างส่วน Help ใช้เป็นเมนูช่วยเหลือการใช้งานโปรแกรม

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Buttons;

type
  TfmHelp = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button1: TButton;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Button2: TButton;
    Label8: TLabel;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

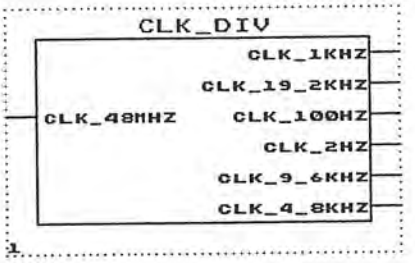
```
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Button3: TButton;
    Label18: TLabel;
    Label19: TLabel;
    Label20: TLabel;
    Label21: TLabel;
    Label22: TLabel;
    Button4: TButton;
    Label23: TLabel;
    Label24: TLabel;
    Button5: TButton;
    Label25: TLabel;
    Label26: TLabel;
    Button6: TButton;
    Label27: TLabel;
    BitBtn1: TBitBtn;
    Label28: TLabel;
    Label29: TLabel;
    Label30: TLabel;
    Label31: TLabel;
    Label34: TLabel;
    Label35: TLabel;
    BitBtn2: TBitBtn;
    Label36: TLabel;
    Label37: TLabel;
    Label38: TLabel;
    ComboBox1: TComboBox;
    Label39: TLabel;

  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  fmHelp: TfmHelp;

implementation
{$R *.DFM}

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 การเชื่อมต่อวงจรย่อยต่างๆ ด้วยภาษา VHDL

รูปที่ 3.5 การเชื่อมต่อวงจรย่อยต่างๆ ด้วยภาษา VHDL

โครงสร้างของบล็อก clk_div



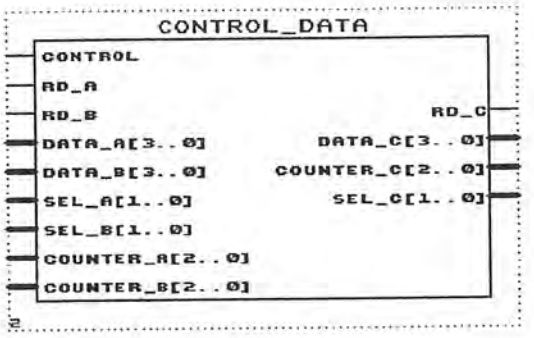

```
library ieee;
use ieee.std_logic_1164.all;
entity clk_div is
port(clk_48MHz        :in std_logic;
     clk_1kHz,clk_19_2kHz,clk_100Hz,clk_2Hz,clk_9_6kHz,clk_4_8kHz  :out
std_logic
      );
end clk_div;
architecture rtl of clk_div is

signal temp_1:integer range 0 to 23999;
signal temp_2,temp_4,temp_6,temp_8,temp_9,temp_10:std_logic;
signal temp_3:integer range 0 to 1249;
signal temp_5:integer range 0 to 4;
signal temp_7:integer range 0 to 24;

begin
 process(clk_48MHz)
        begin
         if clk_48MHz'event and  clk_48MHz='1' then
                if temp_1=23999 then
                        temp_1<=0;
                        temp_2<=not temp_2;
                else
                        temp_1<=temp_1+1;
                end if;
                if temp_3=1249 then
                        temp_3<=0;
                        temp_4<=not temp_4;
                else
                        temp_3<=temp_3+1;
                end if;
            end if;
                        clk_1kHz<=temp_2;
                        clk_19_2kHz<=temp_4;
  end process;
  process(temp_4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
         if temp_4'event and temp_4='1' then
                temp_9<=not temp_9;
         end if;
           clk_9_6kHz<=temp_9;
        end process;

        process(temp_9)
        begin
        if temp_9'event and temp_9='1' then
                temp_10<= not temp_10;
        end if;
        clk_4_8kHz<=temp_10;
end process;
  process(temp_2)
         begin
                if temp_2'event and temp_2='1' then
                 if temp_5=4 then
                        temp_5<=0;
                        temp_6<=not temp_6;
                 else
                        temp_5<=temp_5+1;
                 end if;
                end if;
                        clk_100Hz<=temp_6;
         end process;

         process(temp_6)
          begin
                if temp_6'event and temp_6='1' then
                 if temp_7=24 then
                        temp_7<=0;
                        temp_8<=not temp_8;
                 else
                        temp_7<=temp_7+1;
                 end if;
                end if;
                        clk_2Hz<=temp_8;
         end process;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

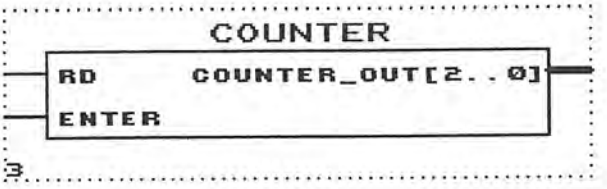
โครงสร้างของบล็อก control_data




```
library ieee;
use ieee.std_logic_1164.all;
entity control_data is
port(control,rd_a,rd_b                    :in std_logic;
        data_a,data_b                     :in std_logic_vector(3 downto 0);
        sel_a,sel_b                       :in std_logic_vector(1 downto 0);
        counter_a,counter_b  :in std_logic_vector(2 downto 0);
        rd_c                              :out std_logic;
        data_c                            :out std_logic_vector(3 downto 0);
        counter_c                         :out std_logic_vector(2 downto 0);
        sel_c                             :out std_logic_vector(1 downto 0)
       );
end control_data;
architecture rtl of control_data is
begin
 process(control)
  begin
        case control is
              when '1' => data_c<=data_a;
                          sel_c<=sel_a;
                          counter_c<=counter_a;
                          rd_c<=rd_a;
              when others=>data_c<=data_b;
                           sel_c<=sel_b;
                           counter_c<=counter_b;
                           rd_c<=rd_b;

    end case;
 end process;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

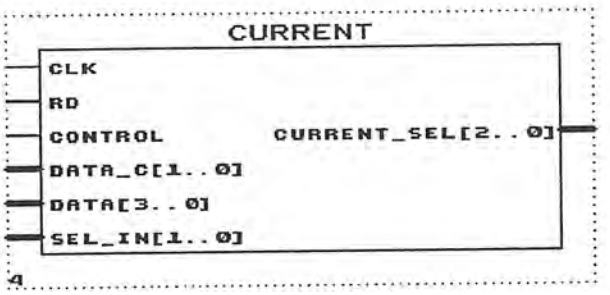
โครงสร้างของบล็อก counter




```
library ieee;
use ieee.std_logic_1164.all;
entity counter is
port(rd,enter              :in std_logic;
        counter_out    :out integer range 0 to 7
        );
end counter;
architecture rtl of counter is
begin

process(enter,rd)
variable temp:integer range 0 to 7;
 begin
  if enter='0' then
    temp:=0;
   elsif rd'event and rd='0' then
          if temp<6 then
                temp:=temp+1;
    end if;
  end if;
   counter_out<=temp;
end process;

end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก current

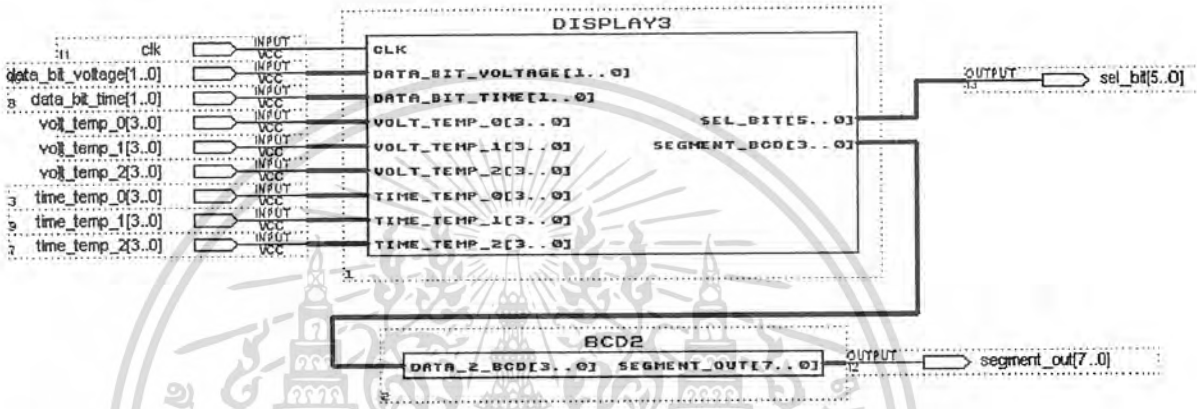



```
library ieee;
use ieee.std_logic_1164.all;

entity current is
port(clk,rd,control          : in std_logic;
        data_c               : in integer range 0 to 3;
    data                     : in std_logic_vector(3 downto 0);
    sel_in                   : in std_logic_vector(1 downto 0);
    current_sel              : out std_logic_vector(2 downto 0)
     );
end current;

architecture rtl of current is
signal temp_0:integer range 0 to 3;
signal temp_1:integer range 0 to 3;
begin

 process(clk)
        begin
         if clk'event and clk='1' then
                if sel_in="11" and data="0000" and rd='0' then
                 if temp_0<3 then
                        temp_0<=temp_0+1;
                 else
                        temp_0<=1;
                 end if;
                end if;
    end if;
        end process;
process(control)
 begin
        case control is
                when '1' =>     temp_1<=temp_0;
                when others=> if sel_in="11" then
                                                temp_1<=data_c;
                                end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   end case;
end process;
                current_sel <= "001" when temp_1=1 else          --50S
                               "010" when temp_1=2 else          --50L
                               "100" when temp_1=3;              --100L
end rtl;
```

โครงสร้างของบล็อก display




```
library ieee;
use ieee.std_logic_1164.all;
entity display3 is
port(clk                                        : in std_logic;
        data_bit_voltage,data_bit_time          : in integer range 0 to 3;
        volt_temp_0,volt_temp_1,volt_temp_2     : in std_logic_vector(3 downto 0);
        time_temp_0,time_temp_1,time_temp_2 : in std_logic_vector(3 downto 0);
        sel_bit                                 : out std_logic_vector(5 downto 0);
        segment_bcd                             : out std_logic_vector(3 downto 0)
       );
end display3;
architecture rtl of display3 is
signal temp:integer range 0 to 7;

begin
 process(clk)
  begin
        if clk'event and clk='1' then
                if temp<5 then
                        temp<=temp+1;
                else temp<=0;
                end if;

        case temp is
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        when 0 =>case data_bit_time is
        when 0 =>       segment_bcd<=time_temp_0;
                        sel_bit<="111110";
        when 1 =>       case time_temp_0 is
        when "0010" =>segment_bcd<="0011";
        when "1111" =>segment_bcd<="0100";
        when "1110" =>segment_bcd<="0000";
        when "1101" =>segment_bcd<="1000";
        when "1011" =>segment_bcd<="1100";
        when "1010" =>segment_bcd<="0001";
        when others =>null;
end  case;
        sel_bit<="111110";
when 2 =>       segment_bcd<=time_temp_1;
                sel_bit<="111110";
when 3 =>       segment_bcd<=time_temp_2;
                sel_bit<="111110";
when others=>sel_bit<="111111";
                end case;
        when 1 =>case data_bit_time is
        when 2 =>       case time_temp_0 is
        when "0010" =>segment_bcd<="0011";
        when "1111" =>segment_bcd<="0100";
        when "1110" =>segment_bcd<="0000";
        when "1101" =>segment_bcd<="1000";
        when "1011" =>segment_bcd<="1100";
        when "1010" =>segment_bcd<="0001";
        when others =>null;
        end  case;
        sel_bit<="111101";
        when 3 =>       segment_bcd<=time_temp_1;
                        sel_bit<="111101";
        when others=> sel_bit<="111111";
                        end case;
        when 2 =>case data_bit_time is
        when 3 =>       case time_temp_0 is
        when "0010" =>segment_bcd<="0011";
        when "1111" =>segment_bcd<="0100";
        when "1110" =>segment_bcd<="0000";
        when "1101" =>segment_bcd<="1000";
        when "1011" =>segment_bcd<="1100";
        when "1010" =>segment_bcd<="0001";
        when others =>null;
        end  case;
        sel_bit<="111011";
        when others=> sel_bit<="111111";
                        end case;
        when 3 =>case data_bit_voltage is
        when 0 =>       segment_bcd<=volt_temp_0;
                        sel_bit<="110111";
```

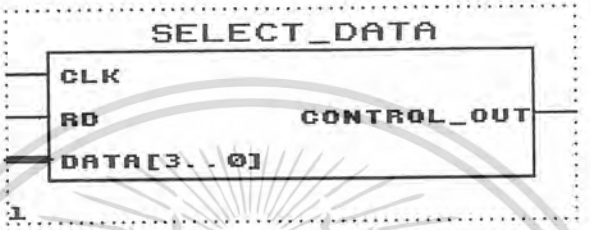
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                when 1 =>       segment_bcd<=volt_temp_1;
                                sel_bit<="110111";
                when 2 =>       segment_bcd<=volt_temp_2;
                                sel_bit<="110111";
                when others =>sel_bit<="111111";
                                end case;
                when 4 =>       case data_bit_voltage is
                when 1 =>       segment_bcd<=volt_temp_0;
                                sel_bit<="101111";
                when 2 =>       segment_bcd<=volt_temp_1;
                                sel_bit<="101111";
                when others =>sel_bit<="111111";
                                end case;
                when 5 =>       case data_bit_voltage is
                when 2 =>       segment_bcd<=volt_temp_0;
                                sel_bit<="011111";
                when others =>sel_bit<="111111";
                                end case;
                when others => sel_bit<="111111";
          end case;
        end if;
end process;

end rtl;


library  ieee;
use ieee.std_logic_1164.all;

entity bcd2 is
port(data_2_bcd                         : in std_logic_vector(3 downto 0);
        segment_out                     : out std_logic_vector(7 downto 0)
        );
end bcd2;

architecture rtl of bcd2 is
begin
        with data_2_bcd select
        segment_out<= "00000011" when "0010", --0
                      "10011111" when "1111", --1
                      "00100101" when "1110", --2
                      "00001101" when "1101", --3
                      "10011001" when "1011", --4
                      "01001001" when "1010", --5
                      "01000001" when "1001", --6
                      "00011111" when "0111", --7
                      "00000001" when "0110", --8
                      "00001001" when "0101", --9
                      "00000010" when "0011", --0.
                      "10011110" when "0100", --1.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "00100100" when "0000", --2.
                    "00001100" when "1000", --3.
                    "10011000" when "1100", --4.
                    "01001000" when "0001", --5.
                    "11111111" when others;
end rtl;
```

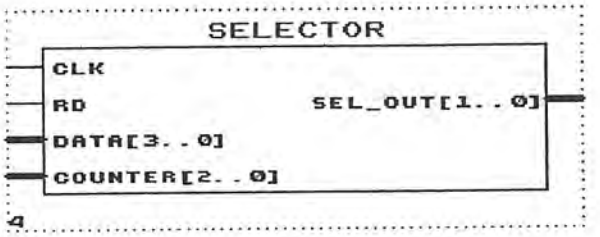
โครงสร้างของบล็อก select_data




```
library ieee;
use ieee.std_logic_1164.all;
entity select_data is
port(clk,rd                     :in std_logic;
        data                    :in std_logic_vector(3 downto 0);
        control_out             :out std_logic
       );
end select_data;
architecture rtl of select_data is
signal temp:std_logic;
begin
 process(clk)
        begin
                if clk'event and clk='1' then
          if data="0011" and rd='0' then
                        temp<=not temp;
                 end if;
                end if;
  end process;
        control_out<=temp;
 end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

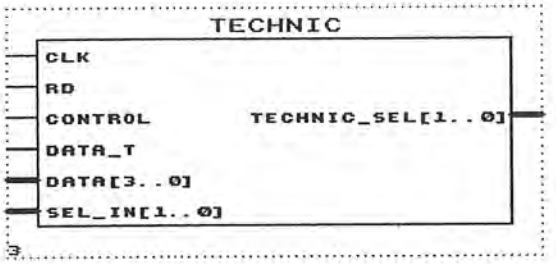
โครงสร้างของบล็อก selector




```
library ieee;
use ieee.std_logic_1164.all;
entity selector is
port(clk,rd                    :in std_logic;
        data                   :in std_logic_vector(3 downto 0);
        counter                :in integer range 0 to 7;
        sel_out                :out std_logic_vector(1 downto 0)
       );
end selector;
architecture rtl of selector is
begin

 process(clk)
        begin
                if clk'event and clk='1' then
                if counter=1 and rd='0' then
                 case data is
                        when "1100" => sel_out <= "00" ; -- Voltage
                        when "1000" => sel_out <= "01" ; -- Time
                        when "0100" => sel_out <= "10" ; -- Technic
                        when "0000" => sel_out <= "11" ; -- current
                        when others => null;
          end case;
                  end if;
                 end if;
 end process;

end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

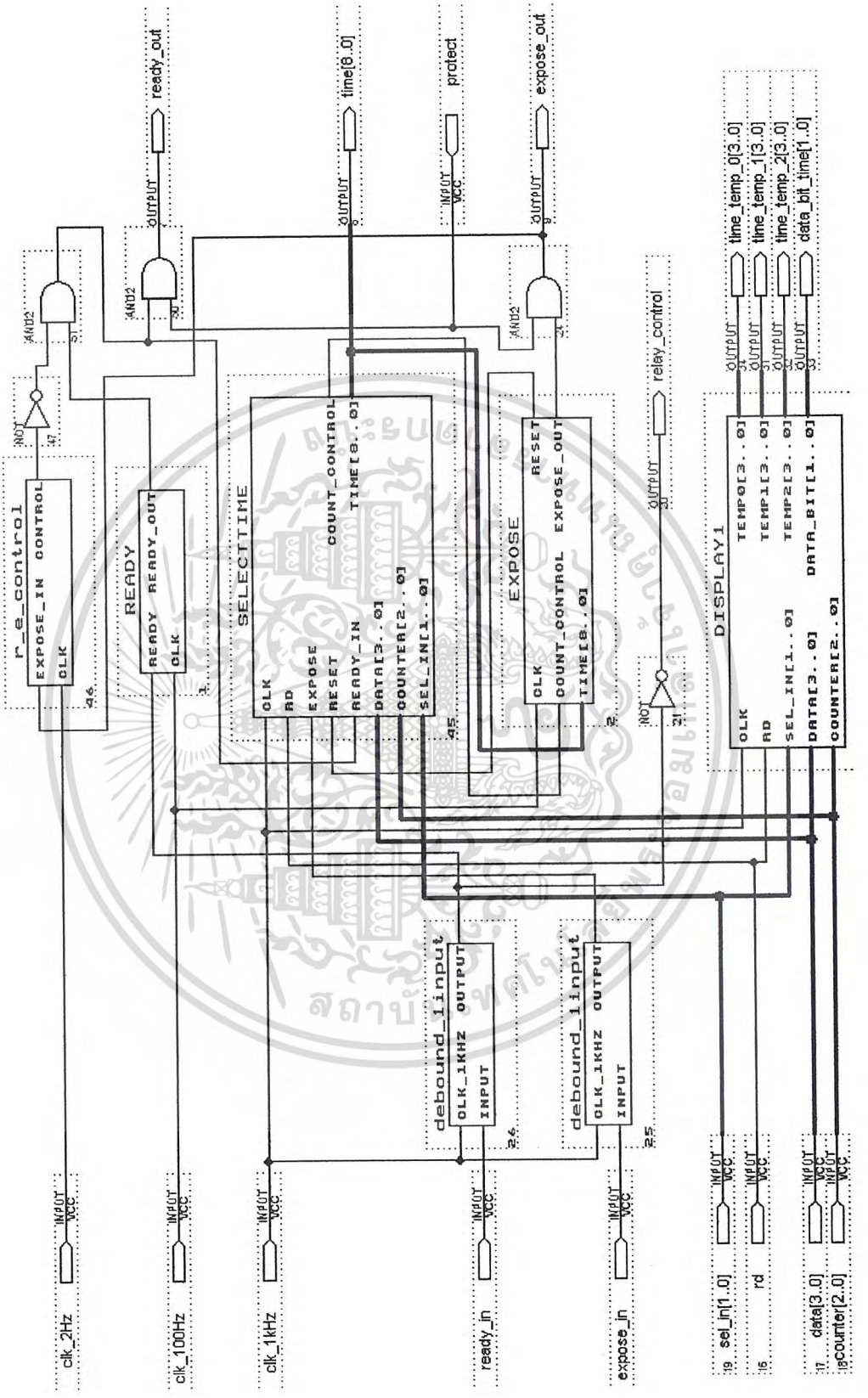
โครงสร้างของบล็อก technic




```
library ieee;
use ieee.std_logic_1164.all;
entity Technic is
port(clk,rd,control,data_t          :in std_logic;
        data                        :in std_logic_vector(3 downto 0);
        sel_in                      :in std_logic_vector(1 downto 0);
        Technic_sel                 :out std_logic_vector(1 downto 0)
       );
end Technic;
architecture rtl of Technic is
signal temp_0:std_logic;
signal temp_1:std_logic;
begin
 process(clk)
       begin
              if clk'event and clk='0' then
               if sel_in="10" and data="0100" and rd='0' then
                     temp_0<=not temp_0;
                end if;
              end if;
 end process;
 process(control)
  begin
        case  control is
              when '1' => temp_1<=temp_0;
              when others=> if sel_in="10" then
                    temp_1<=data_t;
              end if;
        end case;
 end process;
       Technic_sel<="01" when temp_1='0' else
                        "10";
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก time



โครงสร้างของบล็อก ready

```
library ieee;
use ieee.std_logic_1164.all;

entity ready is
port(clk,ready                  : in std_logic;
        ready_out                       : out std_logic
        );
end ready;

architecture rtl of ready is
signal temp:integer range 0 to 200;
begin

 process(clk)
        begin
        if clk'event and clk='1' then
         if ready='0' then
                if temp<=200 then
                        temp<=temp+1;
          end if;
    else temp<=0;
        end if;
         case temp is
                when  0 to 199 => ready_out<='0';
                when  others=> ready_out<='1';
          end case;
        end if;
end process;

end rtl;
```

โครงสร้างของบล็อก selecttime

```
library ieee;
use ieee.std_logic_1164.all;

entity selecttime is
port(clk,rd,expose,reset,ready_in       : in std_logic;
        count_control                   : out std_logic;
        data                            : in std_logic_vector(3 downto 0);
        counter                         : in integer range 0 to 7;
        time                            : out integer range 0 to 511;
        sel_in                          : in std_logic_vector(1 downto 0)
        );
end selecttime;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
architecture rtl of selecttime is
signal time1:integer range 0 to 511;
signal time2:integer range 0 to 127;
signal time3:integer range 0 to 15;
begin

 process(clk)
  begin
                if clk'event and clk='1' then
                 if data="1000" and counter=1  then
                                time1<=0;
                                time2<=0;
                                time3<=0;
                  elsif sel_in="01" and rd='0'  then
                   case counter is
                         when 2 =>
                                case data is
                                        when "1111" => time1<=100; --1s
                                        when "1110" => time1<=200; --2s
                                        when "1101" => time1<=300; --3s
                                        when "1011" => time1<=400; --4s
                                        when "1010" => time1<=500; --5s
                                        when others => time1<=0;   --0s

                                end case;
        --when 3 =>0.__
                           when 4 =>
                                case data is
                                        when "1111" => time2<=10; --0.1s
                                        when "1110" => time2<=20; --0.2s
                                        when "1101" => time2<=30; --0.3s
                                        when "1011" => time2<=40; --0.4s
                                        when "1010" => time2<=50; --0.5s
                                        when "1001" => time2<=60; --0.6s
                                        when "0111" => time2<=70; --0.7s
                                        when "0110" => time2<=80; --0.8s
                                        when "0101" => time2<=90; --0.9s
                                        when others => time2<=0;  --0.0_

                                end case;
                          when 5=>
                                case data is
                                        when "1111" => time3<=1; --0.01s

                                        when "1110" => time3<=2; --0.02s
                                        when "1101" => time3<=3; --0.03s
                                        when "1011" => time3<=4; --0.04s
                                        when "1010" => time3<=5; --0.05s
                                        when "1001" => time3<=6; --0.06s
                                        when "0111" => time3<=7; --0.07s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        when "0110" => time3<=8; --0.08s
                                        when "0101" => time3<=9; --0.09s
                                        when others => time3<=0;
                                end case;
                        when others => null;
                        end case;
                end if;
                                time<=time1+time2+time3;
        end if;
end process;

process(expose,ready_in)
        begin
                if ready_in='0' or reset='1' then
                                count_control<='0';
                elsif expose'event and expose='0' then
                                count_control<='1';
                end if;
end process;

end rtl;
```

โครงสร้างของบล็อก expose

```
library ieee;
use ieee.std_logic_1164.all;

entity expose is
port(clk,count_control          : in std_logic;
        reset,expose_out        : out std_logic;
        time                    : in integer range 0 to 511
        );
end expose;

architecture rtl of expose is
signal count:integer range 0 to 511;
begin

 process(clk)
        begin
         if clk'event and clk='1' then
                if count_control='1' then
                        expose_out<='1';
                        count<=count+1;
                else
                        expose_out<='0';
                        count<=0;
                end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end if;
end process;

process(count)
        begin
    if count=time then
                reset<='1';
        else
                reset<='0';
        end if;
end process;

end rtl;
```

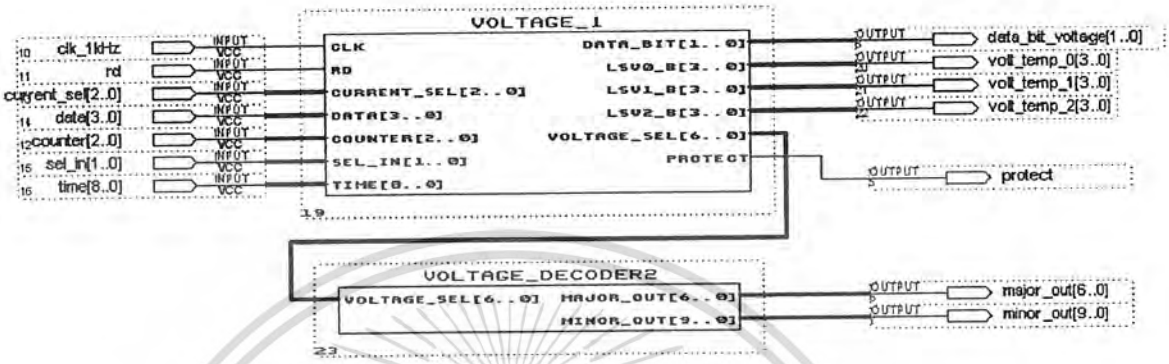
โครงสร้างของบล็อก display1

```
library ieee;
use ieee.std_logic_1164.all;

entity display1 is
port(   clk,rd                          : in std_logic;
        sel_in                          : in std_logic_vector(1 downto 0);
        data                            : in std_logic_vector(3 downto 0);
        counter                         : in integer range 0 to 7;
        temp0,temp1,temp2               : out std_logic_vector(3 downto 0);
        data_bit                        : out integer range 0 to 3
        );
end display1;

architecture rtl of display1 is
begin

 process(clk)
        begin
                if clk'event and clk='1' then
                 if sel_in="01" and rd='0' then
                  case counter is
            when 2 => temp0<=data;
                                                data_bit<=0;
                                when 3 => data_bit<=1;
            when 4 => temp1<=data;
                                                data_bit<=2;
                                when 5 => temp2<=data;
                                                data_bit<=3;
                        when others => null;
            end case;
                  end if;
                 end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end process;

end rtl;
```

โครงสร้างของบล็อก voltage



โครงสร้างของบล็อก voltage_1

```
library ieee;
use ieee.std_logic_1164.all;

entity voltage_1 is
port(clk,rd             : in std_logic;
        current_sel     : in std_logic_vector(2 downto 0);
        data            : in std_logic_vector(3 downto 0);
        counter         : in integer range 0 to 7;
        sel_in          : in std_logic_vector(1 downto 0);     -- select function input
        data_bit        : out integer range 0 to 3;
        lsv0_b,lsv1_b,lsv2_b : out std_logic_vector(3 downto 0);
        voltage_sel     : out std_logic_vector(6 downto 0);    -- select voltage output
        time            : in integer range 0 to 511;
        protect         : out std_logic
       );
end voltage_1;

architecture rtl of voltage_1 is
 signal lsv0,lsv1,lsv2:integer range 0 to 9;
begin

 process(clk)
  variable temp_lsv:integer range 0 to 9;
  variable lsv_a:std_logic_vector(3 downto 0);
        begin
                if clk'event and clk='1' then
                  if sel_in="00" and rd='0' then
            lsv_a:=data;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    case data is
                            when "0010" => temp_lsv := 0;
                            when "1111" => temp_lsv := 1;
                            when "1110" => temp_lsv := 2;
                            when "1101" => temp_lsv := 3;
                            when "1011" => temp_lsv := 4;
                            when "1010" => temp_lsv := 5;
                            when "1001" => temp_lsv := 6;
                            when "0111" => temp_lsv := 7;
                            when "0110" => temp_lsv := 8;
                            when "0101" => temp_lsv := 9;
                            when others => null;
                    end case;
                            case counter is
                when 2 => lsv0<=temp_lsv;
                                                    lsv0_b<=lsv_a;
                                                    data_bit<=0;
                when 3 => lsv1<=temp_lsv;
                                                    lsv1_b<=lsv_a;
                                                    data_bit<=1;
                                    when 4 => lsv2<=temp_lsv;
                                                    lsv2_b<=lsv_a;
                                                    data_bit<=2;
                when others => null;
                end case;
                        end if;
                        end if;
        end process;

-- protection
        protect <='0' when      lsv0=4 and lsv1=3 and current_sel="100" and time>500  else
                --100mA
                                '0' when        lsv0=4 and lsv1=4 and current_sel="100" and
time>460  else
                                '0' when        lsv0=4 and lsv1=5 and current_sel="100" and
time>425  else
                                '0' when        lsv0=4 and lsv1=6 and current_sel="100" and
time>360  else
                                '0' when        lsv0=4 and lsv1=7 and current_sel="100" and
time>320  else
                                '0' when        lsv0=4 and lsv1=8 and current_sel="100" and
time>280  else
                                '0' when        lsv0=4 and lsv1=9 and current_sel="100" and
time>250  else

                                '0' when        lsv0=5 and lsv1=0 and current_sel="100" and
time>200  else
                                '0' when        lsv0=5 and lsv1=1 and current_sel="100" and
time>180  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    '0' when    lsv0=5 and lsv1=2 and current_sel="100" and
time>165 else
                    '0' when    lsv0=5 and lsv1=3 and current_sel="100" and
time>145 else
                    '0' when    lsv0=5 and lsv1=4 and current_sel="100" and
time>130 else
                    '0' when    lsv0=5 and lsv1=5 and current_sel="100" and
time>110 else
                    '0' when    lsv0=5 and lsv1=6 and current_sel="100" and
time>90 else
                    '0' when    lsv0=5 and lsv1=7 and current_sel="100" and
time>80 else
                    '0' when    lsv0=5 and lsv1=8 and current_sel="100" and
time>70 else
                    '0' when    lsv0=5 and lsv1=9 and current_sel="100" and
time>60 else

                    '0' when    lsv0=6 and lsv1=0 and current_sel="100" and
time>46 else
                    '0' when    lsv0=6 and lsv1=1 and current_sel="100" and
time>41 else
                    '0' when    lsv0=6 and lsv1=2 and current_sel="100" and
time>36 else
                    '0' when    lsv0=6 and lsv1=3 and current_sel="100" and
time>33 else
                    '0' when    lsv0=6 and lsv1=4 and current_sel="100" and
time>28 else
                    '0' when    lsv0=6 and lsv1=5 and current_sel="100" and
time>25 else
                    '0' when    lsv0=6 and lsv1=6 and current_sel="100" and
time>21 else
                    '0' when    lsv0=6 and lsv1=7 and current_sel="100" and
time>17 else
                    '0' when    lsv0=6 and lsv1=8 and current_sel="100" and
time>15 else
                    '0' when    lsv0=6 and lsv1=9 and current_sel="100" and
time>12 else

                    '0' when    lsv0=7 and lsv1=0 and current_sel="100" and
time>10 else
                    '0' when    lsv0=7 and lsv1=1 and current_sel="100" and
time>9 else
                    '0' when    lsv0=7 and lsv1=2 and current_sel="100" and
time>8 else
                    '0' when    lsv0=7 and lsv1=3 and current_sel="100" and
time>7 else
                    '0' when    lsv0=7 and lsv1=4 and current_sel="100" and
time>6 else
                    '0' when    lsv0=7 and lsv1=5 and current_sel="100" and
time>5 else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    '0' when    lsv0=7 and lsv1=6 and current_sel="100" and
time>4 else
                    '0' when    lsv0=7 and lsv1=7 and current_sel="100" and
time>3 else
                    '0' when    lsv0=7 and lsv1=8 and current_sel="100" and
time>2 else
                    '0' when    lsv0=7 and lsv1=9 and current_sel="100" and
time>1 else

                    '0' when    lsv0=8 and current_sel="100" else
                    '0' when    lsv0=9 and current_sel="100" else
                    '0' when    lsv0=1 and current_sel="100" else

                    '0' when    lsv0=9 and lsv1=0 and current_sel="001" and
time>480 else        --50mA
                    '0' when    lsv0=9 and lsv1=0 and current_sel="010" and
time>480 else
                    '0' when    lsv0=9 and lsv1=1 and current_sel="001" and
time>450 else
                    '0' when    lsv0=9 and lsv1=1 and current_sel="010" and
time>450 else
                    '0' when    lsv0=9 and lsv1=2 and current_sel="001" and
time>425 else
                    '0' when    lsv0=9 and lsv1=2 and current_sel="010" and
time>425 else
                    '0' when    lsv0=9 and lsv1=3 and current_sel="001" and
time>400 else
                    '0' when    lsv0=9 and lsv1=3 and current_sel="010" and
time>400 else
                    '0' when    lsv0=9 and lsv1=4 and current_sel="001" and
time>375 else
                    '0' when    lsv0=9 and lsv1=4 and current_sel="010" and
time>375 else
                    '0' when    lsv0=9 and lsv1=5 and current_sel="001" and
time>340 else
                    '0' when    lsv0=9 and lsv1=5 and current_sel="010" and
time>340 else
                    '0' when    lsv0=9 and lsv1=6 and current_sel="001" and
time>310 else
                    '0' when    lsv0=9 and lsv1=6 and current_sel="010" and
time>310 else
                    '0' when    lsv0=9 and lsv1=7 and current_sel="001" and
time>290 else
                    '0' when    lsv0=9 and lsv1=7 and current_sel="010" and
time>290 else
                    '0' when    lsv0=9 and lsv1=8 and current_sel="001" and
time>270 else
                    '0' when    lsv0=9 and lsv1=8 and current_sel="010" and
time>270 else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    '0' when    lsv0=9 and lsv1=9 and current_sel="001" and
time>250  else
                    '0' when    lsv0=9 and lsv1=9 and current_sel="010" and
time>250  else
                    '0' when    lsv0=1 and current_sel="001" and time>200
else
                    '0' when    lsv0=1 and current_sel="010" and time>200
else
                    '1';

-- select voltage
 voltage_sel<=

--35KVp - 39KVp

                                "0000000" when lsv0=3 and lsv1=5 and
current_sel="001"  else             --50S
                                "0000001" when lsv0=3 and lsv1=5 and
current_sel="010"  else             --50L
                                "0000010" when lsv0=3 and lsv1=5 and
current_sel="100"  else             --100L

                                "0000001" when lsv0=3 and lsv1=6 and
current_sel="001"  else
                                "0000010" when lsv0=3 and lsv1=6 and
current_sel="010"  else
                                "0000011" when lsv0=3 and lsv1=6 and
current_sel="100"  else

                                "0000010" when lsv0=3 and lsv1=7 and
current_sel="001"  else
                                "0000010" when lsv0=3 and lsv1=7 and
current_sel="010"  else
                                "0000010" when lsv0=3 and lsv1=7 and
current_sel="100"  else

                                "0000011" when lsv0=3 and lsv1=8 and
current_sel="001"  else
                                "0000011" when lsv0=3 and lsv1=8 and
current_sel="010"  else
                                "0000011" when lsv0=3 and lsv1=8 and
current_sel="100"  else

                                "0000100" when lsv0=3 and lsv1=9 and
current_sel="001"  else
                                "0000100" when lsv0=3 and lsv1=9 and
current_sel="010"  else
                                "0000100" when lsv0=3 and lsv1=9 and
current_sel="100"  else
--40KVp - 49KVp
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                "0000101" when lsv0=4 and lsv1=0 and
current_sel="001"  else
                                "0000101" when lsv0=4 and lsv1=0 and
current_sel="010"  else
                                "0000101" when lsv0=4 and lsv1=0 and
current_sel="100"  else

                                "0000110" when lsv0=4 and lsv1=1 and
current_sel="001"  else
                                "0000110" when lsv0=4 and lsv1=1 and
current_sel="010"  else
                                "0000110" when lsv0=4 and lsv1=1 and
current_sel="100"  else

                                "0000110" when lsv0=4 and lsv1=2 and
current_sel="001"  else
                                "0000110" when lsv0=4 and lsv1=2 and
current_sel="010"  else
                                "0000110" when lsv0=4 and lsv1=2 and
current_sel="100"  else

                                "0000111" when lsv0=4 and lsv1=3 and
current_sel="001"  else
                                "0000111" when lsv0=4 and lsv1=3 and
current_sel="010"  else
                                "0000111" when lsv0=4 and lsv1=3 and
current_sel="100"  else

                                "0001000" when lsv0=4 and lsv1=4 and
current_sel="001"  else
                                "0001000" when lsv0=4 and lsv1=4 and
current_sel="010"  else
                                "0001000" when lsv0=4 and lsv1=4 and
current_sel="100"  else

                                "0001001" when lsv0=4 and lsv1=5 and
current_sel="001"  else
                                "0001001" when lsv0=4 and lsv1=5 and
current_sel="010"  else
                                "0001001" when lsv0=4 and lsv1=5 and
current_sel="100"  else

                                "0010000" when lsv0=4 and lsv1=6 and
current_sel="001"  else
                                "0010000" when lsv0=4 and lsv1=6 and
current_sel="010"  else
                                "0010000" when lsv0=4 and lsv1=6 and
current_sel="100"  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    "0010001" when lsv0=4 and lsv1=7 and
current_sel="001"   else
                                    "0010001" when lsv0=4 and lsv1=7 and
current_sel="010"   else
                                    "0010001" when lsv0=4 and lsv1=7 and
current_sel="100"   else

                                    "0010010" when lsv0=4 and lsv1=8 and
current_sel="001"   else
                                    "0010010" when lsv0=4 and lsv1=8 and
current_sel="010"   else
                                    "0010010" when lsv0=4 and lsv1=8 and
current_sel="100"   else

                                    "0010011" when lsv0=4 and lsv1=9 and
current_sel="001"   else
                                    "0010011" when lsv0=4 and lsv1=9 and
current_sel="010"   else
                                    "0010011" when lsv0=4 and lsv1=9 and
current_sel="100"   else
--50KVp - 59KVp

                                    "0010100" when lsv0=5 and lsv1=0 and
current_sel="001"   else
                                    "0010100" when lsv0=5 and lsv1=0 and
current_sel="010"   else
                                    "0010100" when lsv0=5 and lsv1=0 and
current_sel="100"   else

                                    "0010100" when lsv0=5 and lsv1=1 and
current_sel="001"   else
                                    "0010100" when lsv0=5 and lsv1=1 and
current_sel="010"   else
                                    "0010100" when lsv0=5 and lsv1=1 and
current_sel="100"   else

                                    "0010101" when lsv0=5 and lsv1=2 and
current_sel="001"   else
                                    "0010101" when lsv0=5 and lsv1=2 and
current_sel="010"   else
                                    "0010101" when lsv0=5 and lsv1=2 and
current_sel="100"   else

                                    "0010110" when lsv0=5 and lsv1=3 and
current_sel="001"   else
                                    "0010110" when lsv0=5 and lsv1=3 and
current_sel="010"   else
                                    "0010110" when lsv0=5 and lsv1=3 and
current_sel="100"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    "0010111" when lsv0=5 and lsv1=4 and
current_sel="001"  else
                                    "0010111" when lsv0=5 and lsv1=4 and
current_sel="010"  else
                                    "0010111" when lsv0=5 and lsv1=4 and
current_sel="100"  else

                                    "0011000" when lsv0=5 and lsv1=5 and
current_sel="001"  else
                                    "0011000" when lsv0=5 and lsv1=5 and
current_sel="010"  else
                                    "0011000" when lsv0=5 and lsv1=5 and
current_sel="100"  else

                                    "0011001" when lsv0=5 and lsv1=6 and
current_sel="001"  else
                                    "0011001" when lsv0=5 and lsv1=6 and
current_sel="010"  else
                                    "0011001" when lsv0=5 and lsv1=6 and
current_sel="100"  else

                                    "0100000" when lsv0=5 and lsv1=7 and
current_sel="001"  else
                                    "0100000" when lsv0=5 and lsv1=7 and
current_sel="010"  else
                                    "0100000" when lsv0=5 and lsv1=7 and
current_sel="100"  else

                                    "0100001" when lsv0=5 and lsv1=8 and
current_sel="001"  else
                                    "0100001" when lsv0=5 and lsv1=8 and
current_sel="010"  else
                                    "0100001" when lsv0=5 and lsv1=8 and
current_sel="100"  else

                                    "0100010" when lsv0=5 and lsv1=9 and
current_sel="010"  else
                                    "0100010" when lsv0=5 and lsv1=9 and
current_sel="010"  else
                                    "0100010" when lsv0=5 and lsv1=9 and
current_sel="100"  else

--60KVp - 69KVp

                                    "0100010" when lsv0=6 and lsv1=0 and
current_sel="001"  else
                                    "0100010" when lsv0=6 and lsv1=0 and
current_sel="010"  else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "0100010" when lsv0=6 and lsv1=0 and
current_sel="100"   else

                    "0100011" when lsv0=6 and lsv1=1 and
current_sel="001"   else
                    "0100011" when lsv0=6 and lsv1=1 and
current_sel="010"   else
                    "0100011" when lsv0=6 and lsv1=1 and
current_sel="100"   else

                    "0100100" when lsv0=6 and lsv1=2 and
current_sel="001"   else
                    "0100100" when lsv0=6 and lsv1=2 and
current_sel="010"   else
                    "0100100" when lsv0=6 and lsv1=2 and
current_sel="100"   else

                    "0100101" when lsv0=6 and lsv1=3 and
current_sel="001"   else
                    "0100101" when lsv0=6 and lsv1=3 and
current_sel="010"   else
                    "0100101" when lsv0=6 and lsv1=3 and
current_sel="100"   else

                    "0100110" when lsv0=6 and lsv1=4 and
current_sel="001"   else
                    "0100110" when lsv0=6 and lsv1=4 and
current_sel="010"   else
                    "0100110" when lsv0=6 and lsv1=4 and
current_sel="100"   else

                    "0100111" when lsv0=6 and lsv1=5 and
current_sel="001"   else
                    "0100111" when lsv0=6 and lsv1=5 and
current_sel="010"   else
                    "0100111" when lsv0=6 and lsv1=5 and
current_sel="100"   else

                    "0101000" when lsv0=6 and lsv1=6 and
current_sel="001"   else
                    "0101000" when lsv0=6 and lsv1=6 and
current_sel="010"   else
                    "0101000" when lsv0=6 and lsv1=6 and
current_sel="100"   else

                    "0101001" when lsv0=6 and lsv1=7 and
current_sel="001"   else
                    "0101001" when lsv0=6 and lsv1=7 and
current_sel="010"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "0101001" when lsv0=6 and lsv1=7 and
current_sel="100"   else

                    "0101001" when lsv0=6 and lsv1=8 and
current_sel="001"   else
                    "0101001" when lsv0=6 and lsv1=8 and
current_sel="010"   else
                    "0101001" when lsv0=6 and lsv1=8 and
current_sel="100"   else

                    "0110000" when lsv0=6 and lsv1=9 and
current_sel="001"   else
                    "0110000" when lsv0=6 and lsv1=9 and
current_sel="010"   else
                    "0110000" when lsv0=6 and lsv1=9 and
current_sel="100"   else
--70KVp - 79KVp

                    "0110001" when lsv0=7 and lsv1=0 and
current_sel="001"   else
                    "0110001" when lsv0=7 and lsv1=0 and
current_sel="010"   else
                    "0110001" when lsv0=7 and lsv1=0 and
current_sel="100"   else

                    "0110010" when lsv0=7 and lsv1=1 and
current_sel="001"   else
                    "0110010" when lsv0=7 and lsv1=1 and
current_sel="010"   else
                    "0110010" when lsv0=7 and lsv1=1 and
current_sel="100"   else

                    "0110011" when lsv0=7 and lsv1=2 and
current_sel="001"   else
                    "0110011" when lsv0=7 and lsv1=2 and
current_sel="010"   else
                    "0110011" when lsv0=7 and lsv1=2 and
current_sel="100"   else

                    "0110100" when lsv0=7 and lsv1=3 and
current_sel="001"   else
                    "0110100" when lsv0=7 and lsv1=3 and
current_sel="010"   else
                    "0110100" when lsv0=7 and lsv1=3 and
current_sel="100"   else

                    "0110101" when lsv0=7 and lsv1=4 and
current_sel="001"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "0110101" when lsv0=7 and lsv1=4 and
current_sel="010"   else
                    "0110101" when lsv0=7 and lsv1=4 and
current_sel="100"   else

                    "0110110" when lsv0=7 and lsv1=5 and
current_sel="001"   else
                    "0110110" when lsv0=7 and lsv1=5 and
current_sel="010"   else
                    "0110110" when lsv0=7 and lsv1=5 and
current_sel="100"   else

                    "0110110" when lsv0=7 and lsv1=6 and
current_sel="001"   else
                    "0110110" when lsv0=7 and lsv1=6 and
current_sel="010"   else
                    "0110110" when lsv0=7 and lsv1=6 and
current_sel="100"   else

                    "0110111" when lsv0=7 and lsv1=7 and
current_sel="001"   else
                    "0110111" when lsv0=7 and lsv1=7 and
current_sel="010"   else
                    "0110111" when lsv0=7 and lsv1=7 and
current_sel="100"   else

                    "0111000" when lsv0=7 and lsv1=8 and
current_sel="001"   else
                    "0111000" when lsv0=7 and lsv1=8 and
current_sel="010"   else
                    "0111000" when lsv0=7 and lsv1=8 and
current_sel="100"   else

                    "0111001" when lsv0=7 and lsv1=9 and
current_sel="001"   else
                    "0111001" when lsv0=7 and lsv1=9 and
current_sel="010"   else
                    "0111001" when lsv0=7 and lsv1=9 and
current_sel="100"   else

--80KVP - 89KVp

                    "1000000" when lsv0=8 and lsv1=0 and
current_sel="001"   else
                    "1000000" when lsv0=8 and lsv1=0 and
current_sel="010"   else
                    "1000000" when lsv0=8 and lsv1=0 and
current_sel="100"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "1000001" when lsv0=8 and lsv1=1 and
current_sel="001"   else
                    "1000001" when lsv0=8 and lsv1=1 and
current_sel="010"   else
                    "1000001" when lsv0=8 and lsv1=1 and
current_sel="100"   else

                    "1000010" when lsv0=8 and lsv1=2 and
current_sel="001"   else
                    "1000010" when lsv0=8 and lsv1=2 and
current_sel="010"   else
                    "1000010" when lsv0=8 and lsv1=2 and
current_sel="100"   else

                    "1000011" when lsv0=8 and lsv1=3 and
current_sel="001"   else
                    "1000011" when lsv0=8 and lsv1=3 and
current_sel="010"   else
                    "1000011" when lsv0=8 and lsv1=3 and
current_sel="100"   else

                    "1000100" when lsv0=8 and lsv1=4 and
current_sel="001"   else
                    "1000100" when lsv0=8 and lsv1=4 and
current_sel="010"   else
                    "1000100" when lsv0=8 and lsv1=4 and
current_sel="100"   else

                    "1000101" when lsv0=8 and lsv1=5 and
current_sel="001"   else
                    "1000101" when lsv0=8 and lsv1=5 and
current_sel="010"   else
                    "1000101" when lsv0=8 and lsv1=5 and
current_sel="100"   else

                    "1000101" when lsv0=8 and lsv1=6 and
current_sel="001"   else
                    "1000101" when lsv0=8 and lsv1=6 and
current_sel="010"   else
                    "1000101" when lsv0=8 and lsv1=6 and
current_sel="100"   else

                    "1000110" when lsv0=8 and lsv1=7 and
current_sel="001"   else
                    "1000110" when lsv0=8 and lsv1=7 and
current_sel="010"   else
                    "1000110" when lsv0=8 and lsv1=7 and
current_sel="100"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    "1000111" when lsv0=8 and lsv1=8 and
current_sel="001"   else
                                    "1000111" when lsv0=8 and lsv1=8 and
current_sel="010"   else
                                    "1000111" when lsv0=8 and lsv1=8 and
current_sel="100"   else

                                    "1001000" when lsv0=8 and lsv1=9 and
current_sel="001"   else
                                    "1001000" when lsv0=8 and lsv1=9 and
current_sel="010"   else
                                    "1001000" when lsv0=8 and lsv1=9 and
current_sel="100"   else

--90KVp - 99KVp

                                    "1001001" when lsv0=9 and lsv1=0 and
current_sel="001"   else
                                    "1001001" when lsv0=9 and lsv1=0 and
current_sel="010"   else
                                    "1001001" when lsv0=9 and lsv1=0 and
current_sel="100"   else

                                    "1010000" when lsv0=9 and lsv1=1 and
current_sel="001"   else
                                    "1010000" when lsv0=9 and lsv1=1 and
current_sel="010"   else
                                    "1010000" when lsv0=9 and lsv1=1 and
current_sel="100"   else

                                    "1010001" when lsv0=9 and lsv1=2 and
current_sel="001"   else
                                    "1010001" when lsv0=9 and lsv1=2 and
current_sel="010"   else
                                    "1010001" when lsv0=9 and lsv1=2 and
current_sel="100"   else

                                    "1010010" when lsv0=9 and lsv1=3 and
current_sel="001"   else
                                    "1010010" when lsv0=9 and lsv1=3 and
current_sel="010"   else
                                    "1010010" when lsv0=9 and lsv1=3 and
current_sel="100"   else

                                    "1010010" when lsv0=9 and lsv1=4 and
current_sel="001"   else
                                    "1010010" when lsv0=9 and lsv1=4 and
current_sel="010"   else
                                    "1010010" when lsv0=9 and lsv1=4 and
current_sel="100"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "1010011" when lsv0=9 and lsv1=5 and
current_sel="001"   else
                    "1010011" when lsv0=9 and lsv1=5 and
current_sel="010"   else
                    "1010011" when lsv0=9 and lsv1=5 and
current_sel="100"   else

                    "1010100" when lsv0=9 and lsv1=6 and
current_sel="001"   else
                    "1010100" when lsv0=9 and lsv1=6 and
current_sel="010"   else
                    "1010100" when lsv0=9 and lsv1=6 and
current_sel="100"   else

                    "1010101" when lsv0=9 and lsv1=7 and
current_sel="001"   else
                    "1010101" when lsv0=9 and lsv1=7 and
current_sel="010"   else
                    "1010101" when lsv0=9 and lsv1=7 and
current_sel="100"   else

                    "1010110" when lsv0=9 and lsv1=8 and
current_sel="001"   else
                    "1010110" when lsv0=9 and lsv1=8 and
current_sel="010"   else
                    "1010110" when lsv0=9 and lsv1=8 and
current_sel="100"   else

                    "1010111" when lsv0=9 and lsv1=9 and
current_sel="001"   else
                    "1010111" when lsv0=9 and lsv1=9 and
current_sel="010"   else
                    "1010111" when lsv0=9 and lsv1=9 and
current_sel="100"   else
--100KVp - 109KVp

                    "1011000" when lsv0=1 and lsv1=0 and lsv2=0 and
current_sel="001"   else
                    "1011000" when lsv0=1 and lsv1=0 and lsv2=0 and
current_sel="010"   else
                    "1011000" when lsv0=1 and lsv1=0 and lsv2=0 and
current_sel="100"   else

                    "1011001" when lsv0=1 and lsv1=0 and lsv2=1 and
current_sel="001"   else
                    "1011001" when lsv0=1 and lsv1=0 and lsv2=1 and
current_sel="010"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "1011001" when lsv0=1 and lsv1=0 and lsv2=1 and
current_sel="100"   else

                    "1100000" when lsv0=1 and lsv1=0 and lsv2=2 and
current_sel="001"   else
                    "1100000" when lsv0=1 and lsv1=0 and lsv2=2 and
current_sel="010"   else
                    "1100000" when lsv0=1 and lsv1=0 and lsv2=2 and
current_sel="100"   else

                    "1100000" when lsv0=1 and lsv1=0 and lsv2=3 and
current_sel="001"   else
                    "1100000" when lsv0=1 and lsv1=0 and lsv2=3 and
current_sel="010"   else
                    "1100000" when lsv0=1 and lsv1=0 and lsv2=3 and
current_sel="100"   else

                    "1100001" when lsv0=1 and lsv1=0 and lsv2=4 and
current_sel="001"   else
                    "1100001" when lsv0=1 and lsv1=0 and lsv2=4 and
current_sel="010"   else
                    "1100001" when lsv0=1 and lsv1=0 and lsv2=4 and
current_sel="100"   else

                    "1100010" when lsv0=1 and lsv1=0 and lsv2=5 and
current_sel="001"   else
                    "1100010" when lsv0=1 and lsv1=0 and lsv2=5 and
current_sel="010"   else
                    "1100010" when lsv0=1 and lsv1=0 and lsv2=5 and
current_sel="100"   else

                    "1100011" when lsv0=1 and lsv1=0 and lsv2=6 and
current_sel="001"   else
                    "1100011" when lsv0=1 and lsv1=0 and lsv2=6 and
current_sel="010"   else
                    "1100011" when lsv0=1 and lsv1=0 and lsv2=6 and
current_sel="100"   else

                    "1100100" when lsv0=1 and lsv1=0 and lsv2=7 and
current_sel="001"   else
                    "1100100" when lsv0=1 and lsv1=0 and lsv2=7 and
current_sel="010"   else
                    "1100100" when lsv0=1 and lsv1=0 and lsv2=7 and
current_sel="100"   else

                    "1100101" when lsv0=1 and lsv1=0 and lsv2=8 and
current_sel="001"   else
                    "1100101" when lsv0=1 and lsv1=0 and lsv2=8 and
current_sel="010"   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    "1100101" when lsv0=1 and lsv1=0 and lsv2=8 and
current_sel="100"   else

                                    "1100110" when lsv0=1 and lsv1=0 and lsv2=9 and
current_sel="001"   else
                                    "1100110" when lsv0=1 and lsv1=0 and lsv2=9 and
current_sel="010"   else
                                    "1100110" when lsv0=1 and lsv1=0 and lsv2=9 and
current_sel="100"   else

--110KVp - 113KVp

                                    "1100111" when lsv0=1 and lsv1=1 and lsv2=0 and
current_sel="001"   else
                                    "1100111" when lsv0=1 and lsv1=1 and lsv2=0 and
current_sel="010"   else
                                    "1100111" when lsv0=1 and lsv1=1 and lsv2=0 and
current_sel="100"   else

                                    "1100111" when lsv0=1 and lsv1=1 and lsv2=1 and
current_sel="001"   else
                                    "1101000" when lsv0=1 and lsv1=1 and lsv2=1 and
current_sel="010"   else
                                    "1101001" when lsv0=1 and lsv1=1 and lsv2=1 and
current_sel="100"   else

                                    "1101000" when lsv0=1 and lsv1=1 and lsv2=2 and
current_sel="001"   else
                                    "1101000" when lsv0=1 and lsv1=1 and lsv2=2 and
current_sel="010"   else
                                    "1101000" when lsv0=1 and lsv1=1 and lsv2=2 and
current_sel="100"   else

                                    "1101001" when lsv0=1 and lsv1=1 and lsv2=3 and
current_sel="001"   else
                                    "1101001" when lsv0=1 and lsv1=1 and lsv2=3 and
current_sel="010"   else
                                    "1101001" when lsv0=1 and lsv1=1 and lsv2=3 and
current_sel="100"   else

                                    "1111111";

end rtl;
```

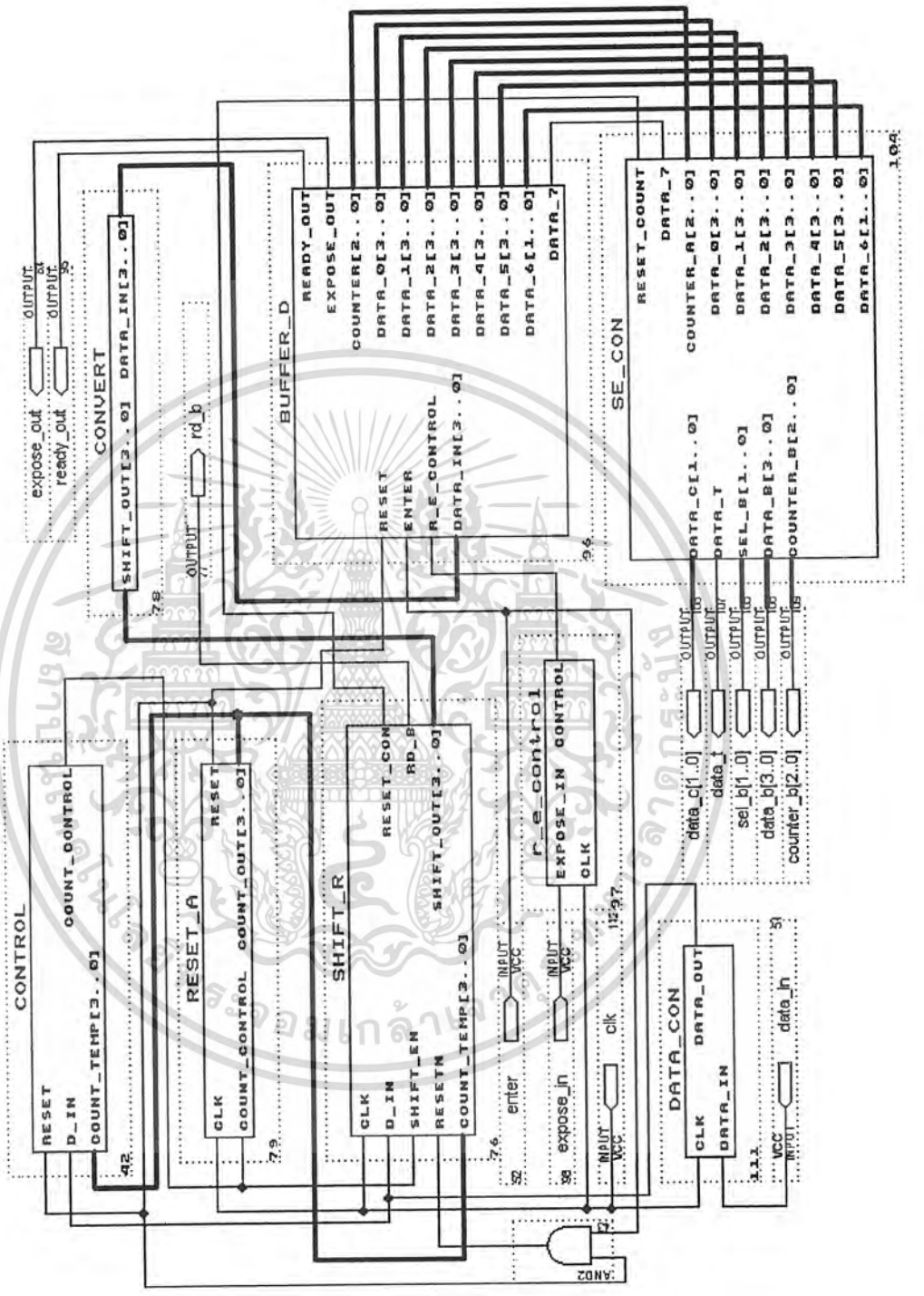
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก voltage_decoder2

```
library ieee;
use ieee.std_logic_1164.all;
entity voltage_decoder2 is
port(voltage_sel              :in std_logic_vector(6 downto 0);
        major_out             :out std_logic_vector(6 downto 0);
        minor_out             :out std_logic_vector(9 downto 0)
       );
end voltage_decoder2;
architecture rtl of voltage_decoder2 is
begin
 process(voltage_sel)
        begin
        case voltage_sel(6 downto 4) is
                when "000" => major_out <= "0000001"; -- L1
                when "001" => major_out <= "0000010"; -- L2
                when "010" => major_out <= "0000100"; -- L3
                when "011" => major_out <= "0001000"; -- L4
                when "100" => major_out <= "0010000"; -- L5
                when "101" => major_out <= "0100000"; -- L6
                when "110" => major_out <= "1000000"; -- L7
                when others =>major_out <= "0000000";
        end case;
        case voltage_sel(3 downto 0) is
                when "0000" => minor_out <= "0000000001"; -- S1
                when "0001" => minor_out <= "0000000010"; -- S2
                when "0010" => minor_out <= "0000000100"; -- S3
                when "0011" => minor_out <= "0000001000"; -- S4
                when "0100" => minor_out <= "0000010000"; -- S5
                when "0101" => minor_out <= "0000100000"; -- S6
                when "0110" => minor_out <= "0001000000"; -- S7
                when "0111" => minor_out <= "0010000000"; -- S8
                when "1000" => minor_out <= "0100000000"; -- S9
                when "1001" => minor_out <= "1000000000"; -- S10
                when others => minor_out <= "0000000000";
        end case;
 end process;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก convert_data



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก data_con

```
library ieee;
use ieee.std_logic_1164.all;
entity data_con is
port(clk,data_in        :in std_logic;
     data_out           :out std_logic
    );
end data_con;
architecture rtl of data_con is
signal buf:std_logic;
begin
      buf<=data_in;
      process(clk)
            begin
             if clk'event and clk='1' then
                  data_out<=buf;
             end if;
      end process;
end rtl;
```

โครงสร้างของบล็อก shift_r

```
library ieee;
use ieee.std_logic_1164.all;

entity shift_r is
port(clk,d_in,shift_en,resetn  :in std_logic;
     count_temp                :in integer range 0 to 15;
     reset_con,rd_b            :out std_logic;
     shift_out                 :out std_logic_vector(3 downto 0)
    );
end shift_r;

architecture rtl of shift_r is
signal shift_reg:std_logic_vector(7 downto 0);
signal temp:integer range 0 to 8;
begin

 process(clk)
      begin
            if resetn='0' then
             shift_reg<=(others=>'0');
            elsif clk'event and clk='0' then
                  if count_temp=9 then
              shift_out<=shift_reg(3 downto 0);
             end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if shift_en='1' then
                        shift_reg(0)<=shift_reg(1);
                        shift_reg(1)<=shift_reg(2);
                        shift_reg(2)<=shift_reg(3);
                        shift_reg(3)<=shift_reg(4);
                        shift_reg(4)<=shift_reg(5);
                        shift_reg(5)<=shift_reg(6);
                        shift_reg(6)<=shift_reg(7);
                        shift_reg(7)<=d_in;
                end if;
        end if;
        case count_temp is
                when 3 to 9 => reset_con <='0';
                when others => reset_con <='1';
        end case;
        case count_temp is
                when 1|6        => rd_b<='0';
                when others => rd_b<='1';
        end case;

end process;

end rtl;
```

โครงสร้างของบล็อก buffer_d

```
library ieee;
use ieee.std_logic_1164.all;
entity buffer_d is
port(reset,enter,r_e_control        :in std_logic;
        data_in                     :in std_logic_vector(3 downto 0);
        ready_out,expose_out        :out std_logic;
        counter                     :out integer range 0 to 7;
        data_0,data_1,data_2,data_3 :out std_logic_vector(3 downto 0);
        data_4,data_5               :out std_logic_vector(3 downto 0);
        data_6                      :out std_logic_vector(1 downto 0);
        data_7                      :out std_logic
        );
end buffer_d;
architecture rtl of buffer_d is
signal count:integer range 0 to 7;
begin
        process(reset,enter)
                begin
                        if enter='0' or r_e_control='1' then
                                count<=0;
                                data_0 <="0000";
                                data_1 <="0000";
                                data_2 <="0000";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    data_3 <="0000";
                                    data_4 <="0000";
                                    data_5 <="0000";
                                    ready_out<='1';
                                    expose_out<='1';
                        elsif reset'event and reset='1' then
        case  data_in is
                when "0000" => ready_out<='0';
                when "0001" => expose_out<='0';
                when others => ready_out<='1';
                                        expose_out<='1';
                                count<=count+1;
                                counter<=count;
                        case count is
                          when 0 =>          data_0 <=data_in;
                          when 1 =>          data_1 <=data_in;
                          when 2 =>          data_2 <=data_in;
                          when 3 =>          data_3 <=data_in;
                          when 4 =>          data_4 <=data_in;
                          when 5 =>          data_5 <=data_in;
                          when 6 =>          data_6 <=data_in(1 downto 0);
                          when 7 =>          data_7 <=data_in(0);
                        end case;
        end case;
                        end if;
        end process;
end rtl;
```

โครงสร้างของบล็อก se_con

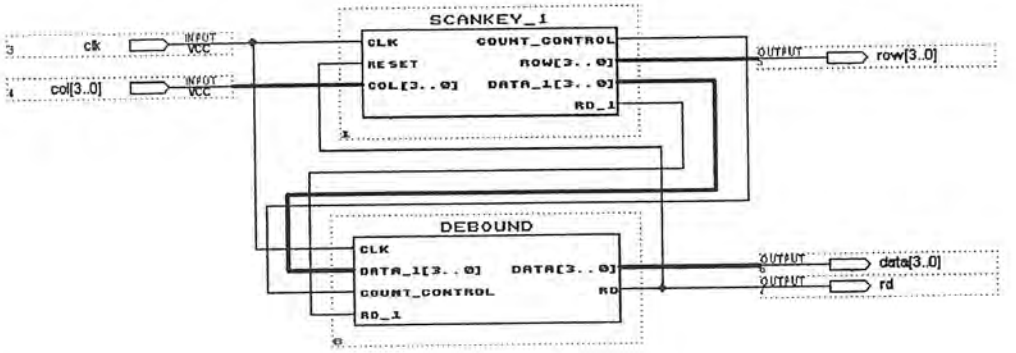
```
library ieee;
use ieee.std_logic_1164.all;
entity se_con is
port(reset_count,data_7         :in std_logic;
        counter_a               :in integer range 0 to 7;
        data_0,data_1,data_2,data_3 :in std_logic_vector(3 downto 0);
        data_4,data_5           :in std_logic_vector(3 downto 0);
        data_6                  :in std_logic_vector(1 downto 0);
        data_c                  :out std_logic_vector(1 downto 0);
        data_t                  :out std_logic;
        sel_b                   :out std_logic_vector(1 downto 0);
        data_b                  :out std_logic_vector(3 downto 0);
        counter_b               :out integer range 0 to 7
        );
end se_con;
architecture rtl of se_con is
signal temp:std_logic;
begin
 process(counter_a)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
        case counter_a is
                when 0 =>       data_b<=data_0;
                                        sel_b<="00";
                                        counter_b<=2;
                                if data_0="1111" then
                                                temp<='1';
                                else temp<='0';
                                end if;
                when 1 =>       data_b<=data_1;
                                        sel_b<="00";
                                        counter_b<=3;
                when 2 => case data_2 is
                when "0010" => if temp='1' then
                                        data_b<=data_2;
                                        sel_b<="00";
                                        counter_b<=4;
                                end if;
                when others =>  data_b<=data_2;
                                        sel_b<="00";
                                        counter_b<=4;
                                end case;
                when 3 => if reset_count='1' then
                                        data_b<="1000";
                                        sel_b<="01";
                                        counter_b<=1;
                                else
                                        data_b<=data_3;
                                        sel_b<="01";
                                        counter_b<=2;
                                end if;
                when 4 => if reset_count='1' then
                                        data_b<="0001";
                                        sel_b<="01";
                                        counter_b<=3;
                                else
                                        data_b<=data_4;
                                        sel_b<="01";
                                        counter_b<=4;
                                end if;
                when 5 => data_b<=data_5;
                                sel_b<="01";
                                counter_b<=5;
                when 6 => data_c<=data_6;
                                sel_b<="11";
                when 7 =>data_t<=data_7;
                                sel_b<="10";

        end case;
    end process;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของบล็อก scankey



โครงสร้างของบล็อก scankey1

```
library ieee;
use ieee.std_logic_1164.all;
entity scankey_1 is
port(   clk                 :in std_logic;
        reset               :in std_logic;
        count_control       :out std_logic;
        col                 :in std_logic_vector(3 downto 0);
        row                 :out std_logic_vector(3 downto 0);
        data_1              :out std_logic_vector(3 downto 0);
        rd_1                :out std_logic
       );
end scankey_1;
architecture rtl of scankey_1 is
signal row_data:std_logic_vector(3 downto 0);
signal data_temp:std_logic_vector(3 downto 0);
signal rd_temp:std_logic;
begin
        row<=row_data;
process(clk)
variable count:integer range 0 to 3;
 begin
        if clk'event and clk='1' then
                if count <=3 then
                  count:=count+1;
                else
                  count:=0;
                end if;
          case count is
                when 0=>row_data<="1110";
                when 1=>row_data<="1101";
                when 2=>row_data<="1011";
                when 3=>row_data<="0111";
          end case;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end if;
end process;
process(clk,col)
begin
  if clk'event and clk='0' then
    case row_data is
                when "1110"=>
                 case col is
                       when"1110"=>data_temp<="0000";rd_temp<='0'; --0
Technic
                       when"1101"=>data_temp<="0001";rd_temp<='0'; --1  .
                       when"1011"=>data_temp<="0010";rd_temp<='0'; --2  0
                       when"0111"=>data_temp<="0011";rd_temp<='0'; --3  Cancel
                       when others =>rd_temp<='1';
                 end case;

                when "1101"=>
                 case col is
                       when"1110"=>data_temp<="0100";rd_temp<='0'; --4  Current
                       when"1101"=>data_temp<="0101";rd_temp<='0'; --5  9
                       when"1011"=>data_temp<="0110";rd_temp<='0'; --6  8
                       when"0111"=>data_temp<="0111";rd_temp<='0'; --7  7
                       when others =>rd_temp<='1';
                 end case;

                when "1011"=>
                 case col is
                       when"1110"=>data_temp<="1000";rd_temp<='0'; --8  Time
                       when"1101"=>data_temp<="1001";rd_temp<='0'; --9  6
                       when"1011"=>data_temp<="1010";rd_temp<='0'; --10 5
                       when"0111"=>data_temp<="1011";rd_temp<='0'; --11 4
                       when others =>rd_temp<='1';
                 end case;

                when "0111"=>
                 case col is
                       when"1110"=>data_temp<="1100";rd_temp<='0'; --12
 Voltage
                       when"1101"=>data_temp<="1101";rd_temp<='0'; --13 3
                       when"1011"=>data_temp<="1110";rd_temp<='0'; --14 2
                       when"0111"=>data_temp<="1111";rd_temp<='0'; --15 1
                       when others =>rd_temp<='1';
                 end case;
                when others=>rd_temp<='1';
    end case;
  end if;
 end process;

 process(reset,rd_temp)
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if reset='0' then
                    count_control<='0';
            elsif rd_temp'event and rd_temp='0' then
                    data_1<=data_temp;
                    count_control<='1';
            end if;
 end process;
rd_1<=rd_temp;

end rtl;
```

โครงสร้างของบล็อก debound

```
library ieee;
use ieee.std_logic_1164.all;
entity debound is
port(clk:in std_logic;
        data_1:in std_logic_vector(3 downto 0);
        data:out std_logic_vector(3 downto 0);
        count_control:in std_logic;
        rd_1:in std_logic;
        rd:out std_logic
       );
end debound;
architecture rtl of debound is
signal  count_temp:integer range 0 to 31;
begin
 process(clk)
        begin
            if clk'event and clk='0' then
             if count_control='1' then
                        count_temp<=count_temp+1;
             else
                        count_temp<=0;
              end if;
            end if;
            if clk'event and clk='1' then
                    case count_temp is
                        when  24 =>   if rd_1='1' then
                                            rd<= '0';
                                      end if;
                        when others  => rd<= '1';
                    end case;
            end if;
 end process;

 process(count_temp)
        begin
            case count_temp is
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                when 24 =>    if rd_1='1' then
                                    data<=data_1;
                              end if;
                        when others =>null;
                end case;
 end process;
end rtl;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EXPOSURE CHART 100MA

PART	PROJECTION	MA	SEC	MAS	KV	FDD(CM	BUCKY
CHEST	PA	100	0.1	10	60-70	180	NO
	LAT	100	0.2	20	70-80	180	NO
ABMODEN	AP	50	1.5	75	70-75	100	YES
	LAT	50	2.0	100	80-85	100	YES
KUB	AP	50	1.5	75	70-75	100	YES
SKULL	AP	100	0.6	60	75-80	100	YES
	LAT	100	0.5	50	70-75	100	YES
C-SPINE	AP	100	0.3	30	65-70	100	NO
	LAT	100	0.4	40	70-75	180	NO
T-SPINE	AP	100	0.4	40	70-75	100	YES
	LAT	100	0.6	60	75-80	100	YES
S-SPINE	AP	50	1.5	75	75-80	100	YES
	LAT	50	2.0	100	80-85	100	YES
PELVIS GIRDLE	AP	50	1.5	80	75-80	100	YES
FEMUR	AP	100	0.1	10	70	100	NO
	LAT	100	0.1	10	70	100	NO
KNEE JOINT	AP	50	0.2	10	62	100	NO
	LAT	50	0.2	10	62	100	NO
LEG	AP	50	0.2	10	65	100	NO
	LAT	50	0.2	10	65	100	NO
ANKLE JOINT	AP	50	0.2	10	55	100	NO
	LAT	50	0.2	10	55	100	NO
FOOT	AP	50	0.2	10	53	100	NO
	OBLIQUE	50	0.2	10	55	100	NO
HAND	AP	50	0.2	10	50	100	NO
	LAT	50	0.2	10	52	100	NO
WRIST JOINT	AP	50	0.2	10	55	100	NO
	LAT	50	0.2	10	53-55	100	NO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PART	PROJECTION	MA	SEC	MAS	KV	FDD(CM	BUCKY
FOREARM	AP	50	0.2	10	55-58	100	NO
	LAT	50	0.2	10	58-60	100	NO
ELBOW JOINT	AP	50	0.2	10	57-60	100	NO
	LAT	50	0.2	10	60-62	100	NO
HUMERUS	AP	50	0.2	10	57-62	100	NO
	LAT	50	0.2	10	60-65	100	NO
SHOULDER	AP	100	0.3	30	75	100	NO
GIRDLE	LAT	100	0.5	50	75	100	NO
SINUS	CALDWELL'S	100	0.6	60	75-80	100	YES
	WATER'S	100	0.8	80	80-83	100	YES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงค่าการเลือกขดของหม้อแปลงที่ค่าแรงดันต่างๆ

	L1	L2	L3	L4	L5	L6	L7
S1	35 KVp	46 KVp	57 KVp	69 KVp	80 KVp	91 KVp	102,103 KVp
S2	36 KVp	47 KVp	58 KVp	70 KVp	81 KVp	92 KVp	104 KVp
S3	37 KVp	48 KVp	59,60 KVp	71 KVp	82 KVp	93,94 KVp	105 KVp
S4	38 KVp	49 KVp	61 KVp	72 KVp	83 KVp	95 KVp	106 KVp
S5	39 KVp	50,51 KVp	62 KVp	73 KVp	84 KVp	96 KVp	107 KVp
S6	40 KVp	52 KVp	63 KVp	74 KVp	85,86 KVp	97 KVp	108 KVp
S7	41,42 KVp	53 KVp	64 KVp	75,76 KVp	87 KVp	98 KVp	109 KVp
S8	43 KVp	54 KVp	65 KVp	77 KVp	88 KVp	99 KVp	110,111 KVp
S9	44 KVp	55 KVp	66 KVp	78 KVp	89 KVp	100 KVp	112 KVp
S10	45 KVp	56 KVp	67,68 KVp	79 KVp	90 KVp	101 KVp	113 KVp

บรรณานุกรม

1. ประพนธ์ อัศวภาณุวัตน์, "Delphi Episode II เทคนิคและการพัฒนาโปรแกรมด้วยเดลไฟ", ซีเอ็ดยูเคชั่น, กรุงเทพ, หน้า 291-318, 2543
2. สัจจะ จรัสรุ่งเรือง และ จักรพงษ์ สุขประเสริฐ , " Delphi 5 ฉบับสมบูรณ์ " , อินโฟเพรส, กรุงเทพ, หน้า 21-349, 2543
3. Astron Logic Research And Development, "ทำความรู้จักกับ FPGA" , Online http:/www.astronlogic.com
4. Frank A.Scarpino,Ph.D., "VHDL And AHDL Digital System Implementation" , Prentice Hall PTR , 40-316. , 1998
5. Stefan Sjoholm And Lennart Lindh , "VHDL for Designers" , Prentice Hall , 1-156., 1997

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้