สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT CONTROL SYSTEM

T055781

โดย

นาย จตุรธร ธีรพัฒนกุล

นาย มานนท์ แท่งทอง

นาย เสิศปัญญา สุทธิศิริกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่ 55781

เลขทะเบียน

วัน,เดือน,ปี 25 พ.ค. 2548

b

i

ระบบควบคุมสัญญาณไฟจราจร

Traffic Light Control System

โดย

นาย จตุรธร ธีรพัฒนกุล

นาย มานนท์ แท่งทอง

นาย เลิศปัญญา สุทธิศิริกุล

อาจารย์ที่ปรึกษา

รศ. พิชัย คูศิริวานิชกร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2546

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT CONTROL SYSETEM

ผู้จัดทำ

1. นาย จตุรธร ธีรพัฒนกุล รหัสประจำตัว 44015188
2. นาย มานนท์ แท่งทอง รหัสประจำตัว 44015209
3. นาย เลิศปัญญา สุทธิศิริกุล รหัสประจำตัว 44015213

..... อาจารย์ที่ปรึกษา

(รศ. พิชัย คูศิริวานิชกร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง ระบบควบคุมสัญญาณไฟจราจร

TRAFFIC LIGHT CONTROL SYSETEM

ผู้จัดทำ

1. นาย จตุรธร ธีรพัฒนกุล รหัสประจำตัว 44015188
2. นาย มานนท์ แท่งทอง รหัสประจำตัว 44015209
3. นาย เลิศปัญญา สุทธิศิริกุล รหัสประจำตัว 44015213

โครงงานได้นับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

.....

(.....)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมสัญญาณไฟจราจร

นาย จตุรธร ธีรพัฒนกุล
นาย มานนท์ แท่งทอง
นาย เลิศปัญญา สุทธิศิริกุล
รศ. พิชัย คูศิริวานิชกร อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้เป็นการจำลองระบบสัญญาณไฟจราจรขึ้นมา เพื่อนำมาใช้ในการศึกษาเกี่ยวกับระบบการควบคุมสัญญาณไฟจราจร โดยใช้ความรู้ในทางด้านดิจิตอล และการเขียนโปรแกรมไมโครคอนโทรลเลอร์ ในการออกแบบระบบควบคุมสัญญาณไฟจราจร โดยพิจารณาจากปริมาณรถที่กำหนดขึ้นมาเพื่อใช้ในการออกแบบ ในโครงงานนี้ตัวอุปกรณ์ที่นำมาใช้เพื่อสร้างเป็นแบบจำลองนั้นคือ อุปกรณ์ FPGA และไมโครคอนโทรลเลอร์ โดยอาศัยคุณสมบัติและความสัมพันธ์ระหว่างอุปกรณ์ทั้งสองในการออกแบบ โดยมีความมุ่งหมายว่าจะสามารถนำเอาความรู้ทางดิจิตอลและไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานได้อย่างไร และสามารถนำอุปกรณ์ FPGA มาใช้ประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRAFFIC LIGHT CONTROL SYSETEM

Mr.Jaturathorn Theerapattanakul

Mr.manon Thangthong

Mr.Lerdpunya Sutthisirikul

Assoc.Prof Phichai Kusiriwanchakorn Advisor

Year 2003

ABSTRACT

This project is a model of traffic of systems which was taken digital and microcontroller programming for design the traffic lights control systems. Amount of car where made for uses in design. Devices that use in this project are FPGA and microcontroller by using qualification and relationship between the devices for design. The objective is how to use digital knowledge microcontroller and FPGA for the benefit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRAC	II
สารบัญ	III
สารบัญภาพ	VI
สารบัญตาราง	VIII
กิตติกรรมประกาศ	IX
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 แนวทางในการทำโครงงาน	1
1.3 โครงสร้างของรายงาน	2
บทที่ 2 ทฤษฎีของโครงงาน	4
2.1 ASIC (Application Specific Integrated Circuits)	4
2.1.1 Full-custom	4
2.1.2 Semi-custom	5
2.1.3 Programmable	5
2.2 FPGA (FIELD PROGRAMMABLE GATE LOGIC)	9
2.2.1 การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ	9
2.2.2 การโปรแกรมโดยใช้หน่วยความจำ	10
2.3 โครงสร้างภายในของ FPGA	10
2.4 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว	11
2.5 การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์	12
2.5.1 การสังเคราะห์วงจร (Logic Synthesis)	13
2.5.2 การแบ่งวงจร (Partitioning)	13
2.5.3 การวางอุปกรณ์ (Placement)	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 การเชื่อมต่อสัญญาณ (Routing) 14

2.5.5 ความหน่วงด้านเวลา (Delay) 14

2.5.6 การจำลองการทำงานของวงจร (Simulation) 15

2.5.7 การโปรแกรมอุปกรณ์ FPGA (Configuration) 16

2.6 เครื่องมือสำหรับการออกแบบ FPGA 16

2.7 การใช้งานโปรแกรม Max Plus II เบื้องต้น 17

2.8 สรุปขั้นตอนการใช้ MAX+PLUS II 25

2.8.1 การวาดวงจร 25

2.8.2 การ Compile 25

2.8.3 การ Simulate 25

2.8.4 การกำหนดขา 25

2.8.5 โปรแกรม 26

2.9 การสื่อสารแบบอนุกรม 26

2.10 UART 27

2.11 โทโปโลยีเครือข่าย 28

2.11.1 โทโปโลยีแนวราบ 28

บทที่ 3 การออกแบบ 32

3.1 แนวคิดในการออกแบบ 32

3.2 ส่วนประกอบต่างๆของระบบ 32

3.3 ส่วนการตรวจสอบปริมาณรถ 33

3.3.1 วงจรจำลองปริมาณรถ 33

3.3.2 วงจรแสดงปริมาณรถ 34

3.4 ส่วนการประมวลผล 37

3.5 ส่วนการแสดงผล 38

3.5.1 การแสดงตารางสถานะของแยกชิดลมและแยกเพลินจิตร 38

3.5.2 วิธีการลดรูปสมการ แยกชิดลมและแยกเพลินจิตร 41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 ส่วนควบคุมการนับขึ้นนับลง 46

3.6 ส่วนประกอบอื่นๆของระบบ 50

บทที่ 4 ผลการทดลอง 54

4.1 วงจรส่วนต่างๆที่อยู่ใน อุปกรณ์ FPGA 54

4.2 ผลการทดลองลักษณะสัญญาณไฟแยกชิดลม 54

4.3 ผลการทดลองลักษณะสัญญาณไฟแยกเพลินจิตร 57

4.4 ผลการทดลองสัญญาณ UP-DOWN 60

บทที่ 5 สรุปและแนวทางการพัฒนา 62

5.1 สรุป 62

5.2 แนวทางการพัฒนา 62

บรรณานุกรม 64

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1.1 แสดงโครงสร้างของระบบ 2
รูปที่ 1.2 แสดงการกำหนดชื่อของเส้นทางและสัญญาณไฟจราจร 3
รูปที่ 2.1 ประเภทของ ASIC 4
รูปที่ 2.2 วงจรพื้นฐานของ PLD ซึ่งอยู่ในรูปผลคูณร่วมบวก 6
รูปที่ 2.3 ลักษณะของ PROM เมื่อเปรียบเทียบเป็นวงจรในรูปผลคูณร่วมบวก 7
รูปที่ 2.4 วงจรพื้นฐานภายในของ PLA 8
รูปที่ 2.5 วงจรพื้นฐานภายในของ PAL 8
รูปที่ 2.6 โครงสร้างภายในของ FPGA ตระกูล MAX7000S 10
รูปที่ 2.7 โครงสร้างภายในของ FPGA ตระกูล FLEX10K 11
รูปที่ 2.8 การโปรแกรมลงในชิพ 12
รูปที่ 2.9 โปรแกรม Max+Plus II 9.5 BASELINE 17
รูปที่ 2.10 การตั้งชื่อโปรเจค 17
รูปที่ 2.11 การเลือกประเภทไฟล์ 18
รูปที่ 2.12.วงจรตัวอย่าง 18
รูปที่ 2.13 การกำหนดสัญลักษณ์ 19
รูปที่ 2.14 การบันทึกข้อมูล 20
รูปที่ 2.15 การกำหนดอุปกรณ์ 21
รูปที่ 2.16 รูปการคอมไพล์ 21
รูปที่ 2.17 การกำหนดโหนดต่าง ๆ ที่ใช้ในการวิเคราะห์ Timming 22
รูปที่ 2.18 การจำลองการทำงาน 22
รูปที่ 2.19 การบันทึกรูปกราฟ 23
รูปที่ 2.20 การจำลองการทำงานอีกครั้ง 23
รูปที่ 2.21 การจำลองการหน่วงเวลา 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.22 การกำหนดขาใช้งาน 24

รูปที่ 2.23.รูปแบบอย่างง่ายที่สุดของการส่งข้อมูลแบบอนุกรม 26

รูปที่ 2.24 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 27

รูปที่ 2.25 โทโปโลยีแบบดาว (Star Topology) 29

รูปที่ 2.26 โทโปโลยีแบบบัส (Bus Topology) 30

รูปที่ 2.27 โทโปโลยีแบบวงแหวน (Ring Topology) 31

รูปที่ 3.1 แสดงการกำหนดชื่อของเส้นทาง 32

รูปที่ 3.2 บล็อกไดอะแกรมแสดงการทำงานของระบบ 33

รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานของส่วนตรวจสอบปริมาณรถ 33

รูปที่ 3.4 วงจรจำลองปริมาณรถ 34

รูปที่ 3.5 แนวคิดการเขียนโปรแกรมส่วนแสดงปริมาณรถ 35

รูปที่ 3.6 แนวคิดการเขียนโปรแกรมส่วนส่งข้อมูล 35

รูปที่ 3.7 วงจรทรานซิสเตอร์ที่ขับกระแสให้กับ 7 – Segment 36

รูปที่ 3.8 แนวคิดการเขียนโปรแกรมควบคุมวงจรประมวลผล 37

รูปที่ 3.9 แสดง Stage Diagram ของส่วนแสดงผลแยกชิดลม 49

รูปที่ 3.10 แสดง Stage Diagram ของส่วนแสดงผลแยกเพลินจิตร 50

รูปที่ 3.11 วงจรแสดงปริมาณรถ 51

รูปที่ 3.12 วงจรประมวลผล 52

รูปที่ 3.13 วงจรส่วนแสดงผล 53

รูปที่ 4.1 ผลการทดลองลักษณะสัญญาณไฟแยกชิดลม 57

รูปที่ 4.2 ผลการทดลองลักษณะสัญญาณไฟจราจรแยกเพลินจิตร 60

รูปที่ 4.3 ผลการทดลองสัญญาณ UP – DOWN 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 ตารางแสดงสถานะแยกชิดลม 38

ตารางที่ 3.2 ตารางแสดงสถานะแยกเพลินจิตร 39

ตารางที่ 3.3 ตารางลักษณะสัญญาณไฟจราจรแยกชิดลม 40

ตารางที่ 3.4 ตารางลักษณะสัญญาณไฟจราจรแยกเพลินจิตร 40

ตารางที่ 3.5 กำหนดการนับขึ้น นับลง แยกชิดลม 47

ตารางที่ 3.6 กำหนดการนับขึ้น นับลง แยกเพลินจิตร 47

ตารางที่ 3.7 ควบคุมการนับขึ้น นับลง ช่องจราจรที่ 5 48

ตารางที่ 3.8 ควบคุมการนับขึ้น นับลง ช่องจราจรที่ 6 48

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับคงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับการช่วยเหลือ และร่วมมือจากหลาย ๆ ท่านด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะท่านเป็นบุคคลที่มีความสำคัญมากและมีส่วนผลักดันให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ซึ่งท่านนั้นก็คือ รองศาสตราจารย์ พิชัย ดูศิริวานิชกร ท่านอาจารย์ที่ปรึกษาปริญญานิพนธ์ เพราะเนื่องจากความที่ท่านอาจารย์ได้กรุณาแนะนำช่วยเหลือและเอาใจใส่พวกเราด้วยดีเสมอมาจึงทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้

และบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ซึ่งได้แก่ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้าเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณทุกๆท่านมา ณ ที่นี้

นาย จตุรธร ธีรพัฒนกุล

นาย มานนท์ แท่งทอง

นาย เลิศปัญญา สุทธิศิริกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การจราจรขนส่งนั้นมีความสำคัญมากในโลกยุคปัจจุบัน ซึ่งหนึ่งในนั้นสัญญาณไฟจราจรก็เป็นส่วนประกอบหนึ่งที่สำคัญต่อการจราจรขนส่ง การทำให้การจราจรนั้นไม่ติดขัด และมีความสะดวกรวดเร็วในการจราจร ก็ขึ้นอยู่กับการออกแบบสัญญาณไฟจราจรที่มีประสิทธิภาพด้วยเช่นกัน

โครงการนี้ได้นำเอาแนวความคิดของ FPGA ซึ่งเป็นชิพที่สามารถออกแบบวงจรดิจิตอลภายในตัวมันได้ ซึ่งในปัจจุบันความจุเกตในตัวชิพ FPGA ได้เพิ่มจากระดับไม่กี่พันตัวจนถึงระดับล้านตัว ซึ่งสามารถรองรับวงจรดิจิตอลที่มีความสลับซับซ้อนได้เป็นอย่างดี มาทำการศึกษา เพื่อทำการสร้างโครงงานนี้ขึ้นมา

1.1 วัตถุประสงค์ของโครงงาน

- ศึกษาการทำงานของ FPGA และการนำเอา FPGA มาประยุกต์ใช้งาน
- ศึกษาการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ และ ชิพ FPGA ว่าทำงานสอดคล้องกันได้อย่างไร
- จำลองรูปแบบของสัญญาณไฟจราจร เพื่อเป็นแนวทางในการแก้ไขปัญหาการจราจรต่อไปในอนาคต

1.2 แนวทางในการทำโครงงาน

โครงสร้างของโครงงานนี้ประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

1. วงจรแสดงปริมาณรถซึ่งเป็นวงจรนับ (Counter) ที่นับจาก 00-99 จำนวน 11 ชุดโดยสร้างจาก MCS-51 เพื่อการสมมุติความหนาแน่นของการจราจรไปควบคุมสัญญาณไฟจราจรที่ออกแบบในชิพ FPGA

2. วงจรประกอบในส่วนของชิพ FPGA ซึ่งมีวงจรจ่ายไฟเลี้ยงให้กับตัวชิพ FPGA และวงจรประกอบพื้นฐานเพื่อทำให้ชิพ FPGA ทำงาน

3. ส่วนประมวลผลที่โปรแกรมลงไปในชิพ FPGA ซึ่งสามารถดูรายละเอียดขั้นตอนการออกแบบได้ในบทที่ 3

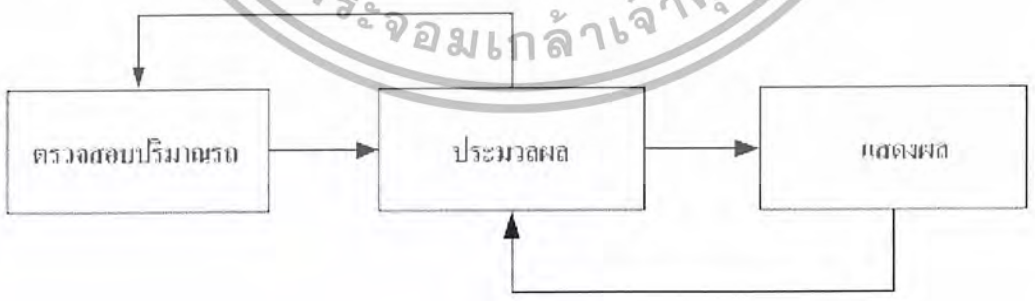
4. เอาท์พุทที่แสดงสัญญาณไฟจราจร ซึ่งประกอบไปด้วย ไฟแดง ไฟเขียว และไฟเหลือง โดยใช้ LED ในการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 โครงสร้างของรายงาน

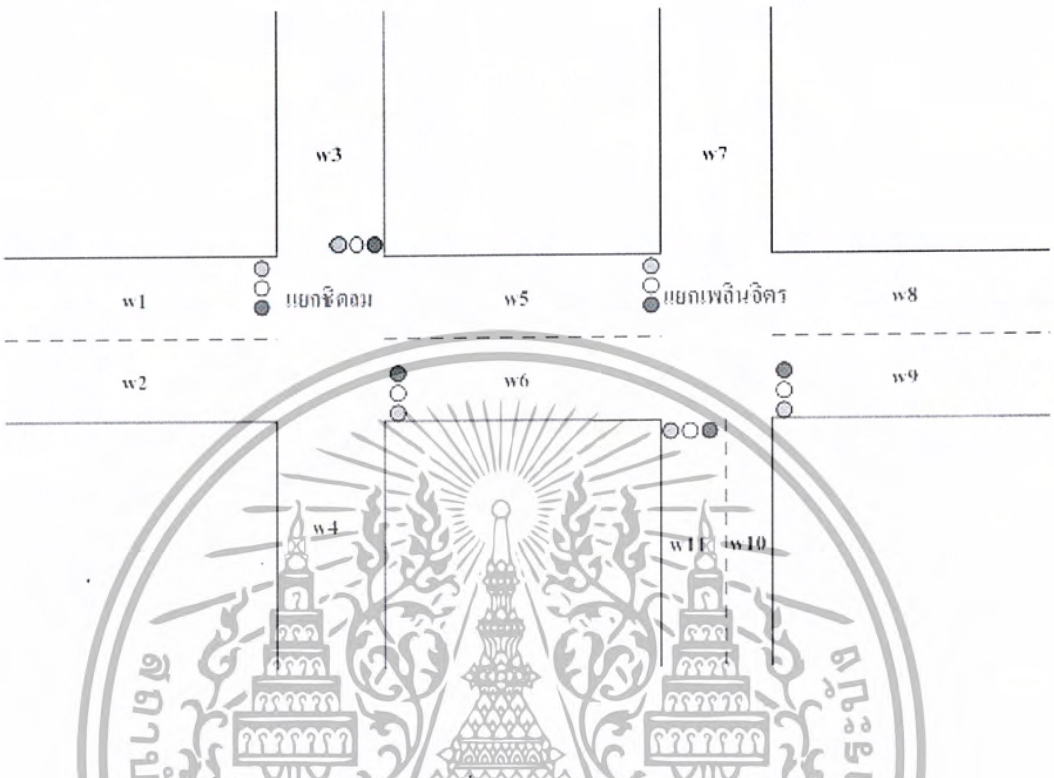
สำหรับรายงานฉบับนี้ ประกอบด้วยบทต่าง ๆ ดังนี้ ในบทที่ 2 จะกล่าวถึงเนื้อหาและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการทำโครงงาน ในบทที่ 3 กล่าวถึงขั้นตอนในการออกแบบวงจรดิจิตอลทั้งหมดที่ใช้โปรแกรมลงไปในชิพ FPGA และระบบอิเล็กทรอนิกส์ต่าง ๆ ที่ใช้ประกอบโครงสร้างพร้อมกับคำอธิบายการทำงานของวงจรอย่างละเอียด ในบทที่ 4 จะเป็นส่วนของวิธีการทดลอง และผลการทดลองของโครงงาน บทที่ 5 จะเป็นบทสรุปของโครงงาน และปัญหาต่าง ๆ พร้อมทั้งแนวทางแก้ไข

ในโครงงานนี้เราได้ทำการจำลองเอาแยกชิดลมและแยกเพลินจิตรมาใช้มาใช้ในการทำโครงงานโดยเราได้มีการศึกษาถึงช่องทางการจราจรและตำแหน่งของสัญญาณไฟจราจรที่เหมือนกันกับแยกชิดลมและแยกเพลินจิตร แล้วสร้างเป็น Model ขึ้นมาโดยการต่อเชื่อมกัน โดยเราได้กำหนดเอาปริมาณรถในแต่ละช่องทางอย่าคร่าว ๆ โดยคล้าย ๆ กับปริมาณจำนวนรถที่มีอยู่จริง แต่ไม่ใช่ปริมาณที่แท้จริง เพื่อมาใช้ในการกำหนดว่าจะควบคุมสัญญาณไฟจราจรได้อย่างไร ในโครงงานนี้นั้นเราสามารถควบคุมสัญญาณไฟจราจรได้โดย เมื่อช่องจราจรใดมีจำนวนรถเต็มตามที่เราได้ตั้งเอาไว้ เราจะทำการปล่อยรถออกจากช่องจราจรนั้นก่อน โดยให้เป็นสัญญาณไฟเขียว ส่วนช่องการจราจรอื่นที่มีสัญญาณไฟจะให้เป็นสัญญาณไฟแดง นอกจากนี้เรายังใช้เวลามาเป็นตัวกำหนดอีกด้วยว่า ถ้าไม่มีช่องจราจรใดแจ้งเข้ามาว่ามีรถเต็ม แต่ถ้าสัญญาณไฟที่ติดอยู่ในขณะนั้นติดอยู่เป็นเวลาเท่ากับที่เรากำหนดเอาไว้มันก็จะเปลี่ยนไปตาม stage ที่เราได้ตั้งเอาไว้ โดยพิจารณาจากจำนวนรถซึ่งแสดงโดย 7-segment 2 หลักในแต่ละช่องทาง รวมทั้งพิจารณาสถานะก่อนหน้านี้ด้วย ซึ่งส่วนประกอบของระบบโดยคร่าว ๆ จะเป็นดังรูป 1.1



รูปที่ 1.1 แสดงโครงสร้างของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 แสดงการกำหนดชื่อของเส้นทางและสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีของโครงงาน

2.1 ASIC (Application Specific Integrated Circuits)

จากการปรากฏตัวของ VLSI ทำให้วิศวกรเริ่มมีการออกแบบไอซีตามความต้องการของลูกค้าซึ่งใช้ใน ระบบที่เจาะจงนอกเหนือจากการใช้ไอซีมาตรฐานเพียงอย่างเดียว โดยไอซีเหล่านี้มีชื่อเรียกว่า ASIC: Application-Specific Integrated Circuit (ออกเสียงว่า เอ-ซิก) ซึ่งตัวอย่างของ ASIC ได้แก่ชิพไอซีที่ใช้ สำหรับตุ๊กตาของเล่นพูดได้ ดาวเทียม และ ชิพที่ภายในบรรจุด้วยไมโครโปรเซสเซอร์กับอุปกรณ์ทางลอจิกอื่นๆ

ASIC แบ่งเป็น 3 ประเภทใหญ่ๆ คือ Full-custom, Semi-custom และ Programmable ดังรูป 2.1

รูปที่ 2.1 ประเภทของ ASIC

2.1.1 Full-custom

ASIC ประเภทนี้ลูกค้าจะเป็นผู้ออกแบบเซลล์ลอจิก (เช่น แอนด์เกท ออร์เกท มัลติเพล็กเซอร์ และฟลิปฟลอบ) และลักษณะ การจัดวางอุปกรณ์บนตัวไอซีรวมถึงหน้ากากสำหรับควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเจือและสร้างชั้นสาร (Mask) ต่างๆ ที่ ใช้ในการทำไอซีเอง ดังนั้นค่าใช้จ่ายในการออกแบบและการผลิตจะสูงมาก

2.1.2 Semi-custom

ASIC ประเภทนี้เซลล์ลอจิกจะถูกออกแบบเอาไว้ก่อนแล้วในรูปแบบของไลบรารีและลูกค้าจะเป็นผู้ออกแบบ Mask ต่างๆเอง ตัวอย่างของไอซีประเภทนี้ได้แก่ Standard-Cell-Based ASIC และ Masked Gate-Array-Based ASIC

2.1.2.1 Standard-Cell-Based ASIC

ไอซีประเภทนี้จะมีพื้นที่สำหรับจัดวางเซลล์ลอจิกมาตรฐานซึ่งถูกออกแบบเอาไว้แล้ว ในบางครั้งเซลล์ มาตรฐานเหล่านี้จะถูกนำมาประกอบกันเป็นเซลที่มีขนาดใหญ่ขึ้นเรียกว่า Mega cell สำหรับการออก แบบนั้นผู้ออกแบบจะทำเพียงแค่กำหนดตำแหน่งของเซลล์มาตรฐานและการเชื่อมต่อภายในของแต่ ละเซลล์เท่านั้นแต่อย่างไรก็ดีเซลล์ต่างๆ เหล่านี้สามารถวางที่ตำแหน่งใดๆ ก็ได้บนแผ่นเวเฟอร์ซิลิกอน นั่นก็หมายความว่าชั้น Mask จะถูกจัดวางตามความต้องการของผู้ออกแบบ

2.1.2.2 Masked Gate-Array-Based ASIC

ไอซีชนิดนี้จะมีทรานซิสเตอร์หรือเกทถูกสร้างมาในลักษณะของอะเรย์สองมิติบนแผ่นเวเฟอร์ซิลิกอน และผู้ออกแบบจะทำการออกแบบ Mask เพื่อใช้สำหรับกำหนดการต่อเชื่อมของทรานซิสเตอร์แต่ละตัว

2.1.3 Programmable

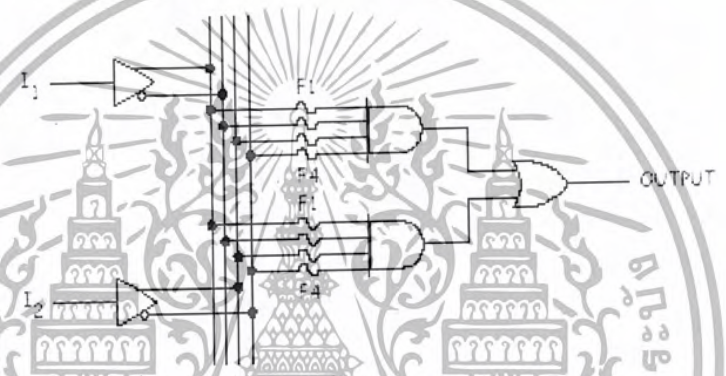
ASIC ประเภทนี้เซลล์ลอจิกจะถูกออกแบบไว้ก่อนเช่นเดียวกับ Semi-Custom แต่ชั้นของ Mask จะไม่สามารถเปลี่ยนแปลงได้ ตามความต้องการของผู้ออกแบบ ไอซีประเภทนี้ยังแบ่งออกเป็น 2 ชนิดคือ Programmable Logic Device (PLD) และ Field Programmable Gate Array (FPGA)

2.1.3.1 Programmable Logic Device (PLD)

มีโครงสร้างภายในเป็นวงจรพื้นฐานทางด้านลอจิกต่อกันอยู่เป็นกลุ่มซึ่งมีทั้งวงจรคอมบิเนชัน(Combination) และซีเควนเชียล (Sequential) สำหรับเทคโนโลยีของวงจรที่ใช้สร้าง PLD จะมีทั้ง TTL, ECL และ CMOS ตามความเหมาะสมของแต่ละระบบ ไอซี PLD ทุกชนิดมีหลักการพื้นฐานของวงจรภายในที่เหมือนกันโดยมี วงจรคอมบิเนชันที่เป็นผลคูณร่วม บวก (Sum of product) ประกอบไปด้วยชุดของแอนด์เกทต่อร่วมกับออร์ เกทและในการ โปรแกรมจะเป็นการเลือกว่า

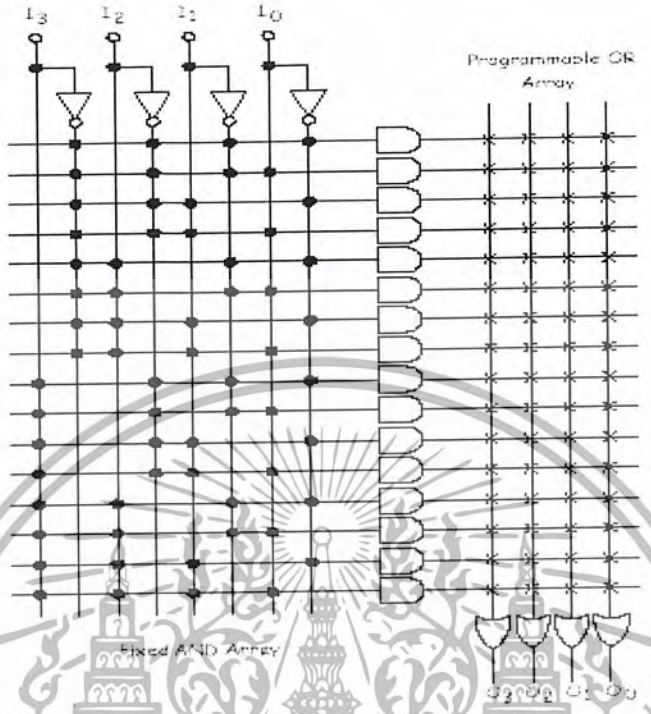
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุทภายในของแอนด์เกทกับสัญญาณอินพุทใดบ้างที่จะต้อง ต่อ ถึงกันซึ่งมีทั้งจากภายนอกและสัญญาณป้อนกลับจากเอาท์พุทภายในเอง เช่น การติดต่ออินพุทของออร์ เกทกับเอาท์พุทของแอนด์เกทตัวต่างๆ สำหรับการโปรแกรมทางกายภาพนั้นอินพุท ต่างๆ ของอุปกรณ์ทุกตัว จะถูกต่อผ่านฟิวส์เข้ากับแหล่งสัญญาณ ซึ่งถ้าไม่ต้องการใช้สัญญาณ ใดก็จะตัดฟิวส์ตัวนั้นทิ้งทำให้สามารถโปรแกรมได้เพียงครั้งเดียว ไอซี PLD บางชนิดใช้ มอสทรานซิสเตอร์แทนฟิวส์ทำให้สามารถโปรแกรมโดย ใช้กระแสไฟฟ้าและสามารถลบแล้ว โปรแกรมเข้าไปใหม่ได้อีก สำหรับไอซีในตระกูล PLD ได้แก่ PROM, PAL, PLA และ EPLD



รูปที่ 2.2 วงจรพื้นฐานของ PLD ซึ่งอยู่ในรูปผลคูณร่วมบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ลักษณะของ PROM เมื่อเปรียบเทียบเป็นวงจรในรูปผลคูณร่วมบวก

2.1.3.1.1 PROM (Programmable Read Only Memory)

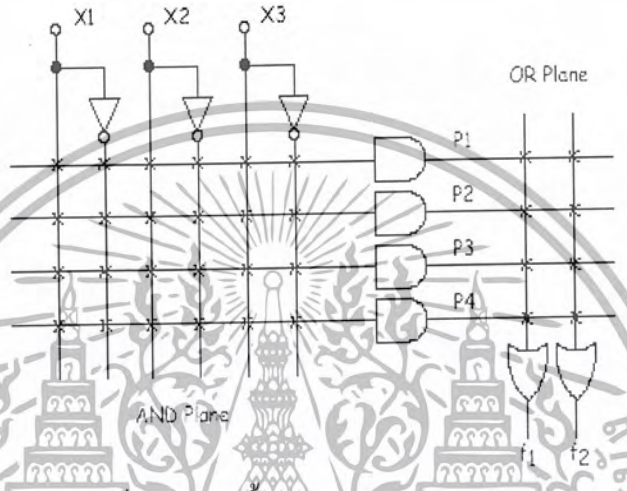
PROM คือหน่วยความจำประเภท ROM ซึ่งนับว่าเป็นไอซี PLD ชนิดหนึ่งซึ่งวงจรภายในของ PROM ประกอบไปด้วยอะเรย์ของแอนด์และออร์เกท (And - Or Array) ผลลัพธ์ที่ขาดาต้าเอาท์พุทสามารถแสดงได้ในสมการของฟังก์ชันผลคูณร่วมบวก (Sum of product) ของสัญญาณอินพุทที่ขาแอดเดรส

รูป 2.3 แสดงถึงลักษณะการเชื่อมต่อแอนด์เกทและออร์เกทของ PROM ขนาด 16x4 บิต วงจรทางด้านซ้ายบนสุดเป็นแอนด์เกทจะให้ผลคูณ (Product) ของกรณีที่อินพุทเป็น0000 แอนด์เกทที่อยู่ถัดลงมาเป็นผลคูณของกรณีที่อินพุทเป็น 0001, 0010, ...จนถึงตัวล่างสุดคือ ผลคูณในกรณีที่อินพุทเป็น 1111 ซึ่งสำหรับ PROM ที่มีจำนวนอินพุท n ตัวจะมีค่าอินพุทที่ เป็นไปได้ทั้งหมดเท่ากับ 2n และค่าอินพุทเหล่านี้จะถูกจัดวางอยู่ในส่วนอะเรย์ของ AND ซึ่งไม่สามารถแก้ไขได้ แต่ในส่วนของ OR จะเป็นส่วนที่อนุญาตให้ทำการโปรแกรมได้ และเนื่องจากการที่ด้าน AND ของ PROM มีการคอมบิเนชันของอินพุทที่เป็นไปได้ทั้งหมด ดังนั้นผู้ออกแบบจึงไม่จำเป็นต้องทำการลดรูปของฟังก์ชันลอจิกที่ออกแบบไว้เลยแต่อย่าง ไรก็ดีการกระทำเช่นนี้อาจทำให้เกิดจำนวนวงจรที่ไม่มีประสิทธิภาพจำนวนมากบนตัว ชิพได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.1.2 PLA (Programmable Logic Array)

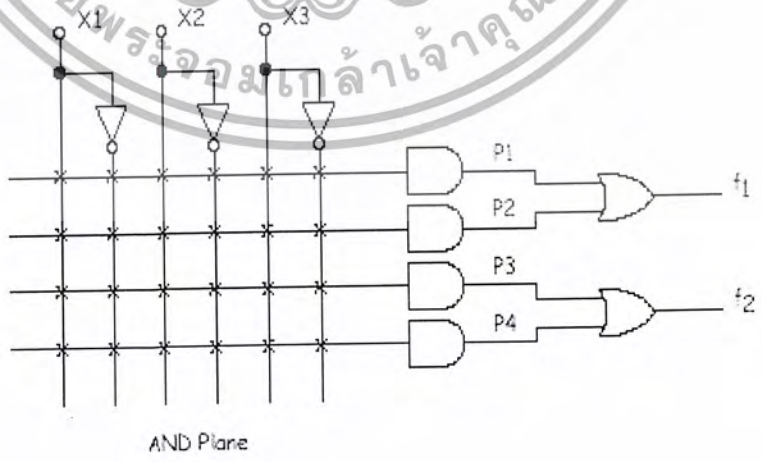
ลักษณะเด่นของ PLA คือสามารถโปรแกรมการเชื่อมต่อได้ทั้งทางด้าน AND และด้าน OR ทำให้มีความยืดหยุ่นในการใช้งานมาก แต่อย่างไรก็ดีข้อเสียที่เห็นได้อย่างชัดเจน ของ PLA คือ ความยุ่งยากในการสร้างและคุณสมบัติทางด้านความเร็วที่ลดลงเนื่องจาก สัญญาณจะต้องวิ่งผ่านอะเรย์ของ AND และ OR



รูปที่ 2.4 วงจรพื้นฐานภายในของ PLA

2.1.3.1.3 PAL (Programmable Array Logic)

PAL มีลักษณะโครงสร้างที่ใกล้เคียงกับ PROM และ PLA มาก แต่การโปรแกรม PAL จะสามารถทำได้เพียงด้าน AND เท่านั้น



รูปที่ 2.5 วงจรพื้นฐานภายในของ PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.1.4 EPLD (Erasable Programmable Logic Device)

EPLD เป็นอุปกรณ์ที่สามารถทำการโปรแกรมได้หลายครั้งซึ่งเหมาะสำหรับการ ทำวงจรต้นแบบ สำหรับเทคโนโลยีที่ใช้ในการสร้างจะเหมือนกับ CMOS EPROM คือ ใช้มอสทรานซิสเตอร์เชื่อมต่อระหว่างสัญญาณอินพุทกับจุดที่ต้องการแทนการ ใช้ฟิวส์แบบเดิมทำให้สามารถโปรแกรมการต่อวงจรภายในอุปกรณ์ด้วยการจ่าย ไฟฟ้าตามขนาดที่กำหนดไว้และลบได้โดยใช้แสงอัลตร้าไวโอเลทฉาย ผ่านช่อง หน้าต่างกระจกของตัวชิพ

2.1.3.2 Field-Programmable Gate Array (FPGA)

เป็นอุปกรณ์ที่มีความซับซ้อนมากกว่า PLD ไปอีกระดับหนึ่ง ซึ่งในความเป็นจริงแล้ว PLD และ FPGA แตก ต่างกันน้อยมาก สำหรับ FPGA แล้วนับว่าเป็นอุปกรณ์ตัวใหม่ในตระกูลของ ASIC ซึ่งมีการเจริญเติบโตอย่าง รวดเร็วและมีบทบาทที่สำคัญในการเข้ามาแทนที่ระบบอิเล็กทรอนิกส์ที่ใช้ TTL โครงสร้าง ภายในของ FPGA ประกอบไปด้วยอะเรย์ของลอจิกเกทต่างๆ มากมาย ซึ่งในปัจจุบันความจุเกทภายใน ตัวชิพ FPGA ได้เพิ่มขึ้น จากระดับไม่กี่พันตัวจนถึงระดับล้านตัวซึ่งสามารถรองรับวงจรดิจิตอลที่มีความ สลับซับซ้อนได้เป็นอย่างดี นอกจากนี้ในด้านการออกแบบพัฒนาและทดสอบก็ทำได้ง่ายซึ่งในปัจจุบัน การออกแบบวงจรโดยใช้ FPGA กำลังเป็นที่นิยมและมีแนวโน้มที่จะนำมาใช้งานมากขึ้นเรื่อย

2.2 FPGA (FIELD PROGRAMMABLE GATE LOGIC)

ในปัจจุบันมี FPGA อยู่ 4 ชนิดที่วางขายอยู่ในท้องตลาดได้แก่ Symmetrical Array, Row-Based, Hierarchical PLD และ Sea-of-Gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการโปรแกรม ที่แตกต่างกันไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรม ซึ่งมีอยู่ 2 แบบคือ การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรม โดยการใช้หน่วยความจำ

2.2.1 การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

2.2.1.1 Fuse เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากทำโปรแกรมแล้วจุดเชื่อม ต่อจะขาดจากกัน

2.2.1.2 Anti Fuse เป็นวิธีการโปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรม แล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การโปรแกรมโดยใช้หน่วยความจำ

2.2.2.1 EEPROM Based FPGA

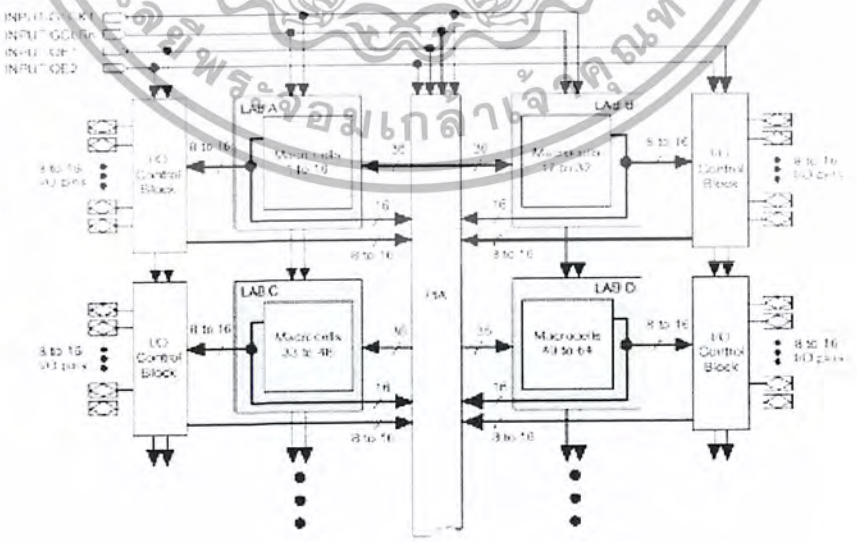
FPGA ที่ใช้การโปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจุของเกทต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกท แต่ข้อดีของ EEPROM Based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิท ซึ่งการโปรแกรมสามารถทำได้ประมาณ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

2.2.2.2 SRAM Based FPGA

FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถ โปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกท ในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกท) ซึ่งข้อดีของ SRAM Based FPGA คือใช้เวลาในการโปรแกรมน้อย (ระดับ nSec) การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิพในขณะที่เริ่มต้นใช้งาน

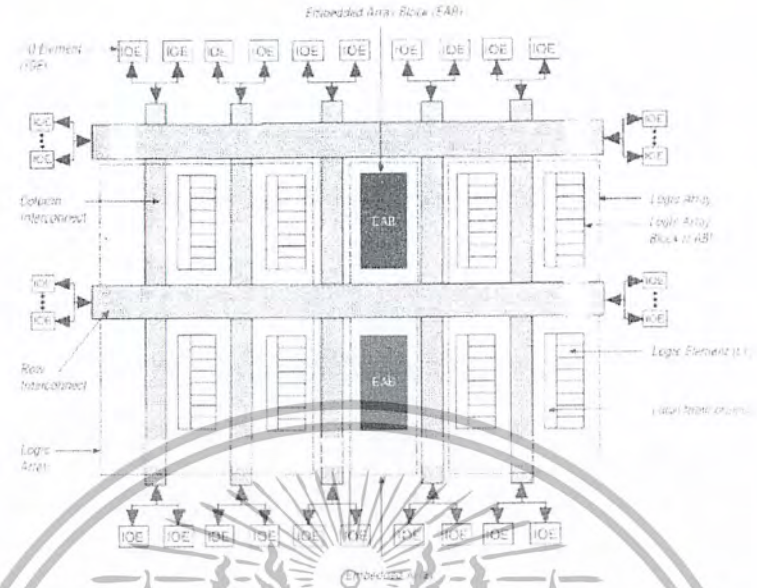
2.3 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอะเรย์ของบล็อคลอจิกที่สามารถทำการโปรแกรมได้ดังรูป 2.6 และ 2.7



รูปที่ 2.6 โครงสร้างภายในของ FPGA ตระกูล MAX7000S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 โครงสร้างภายในของ FPGA ตระกูล FLEX10K

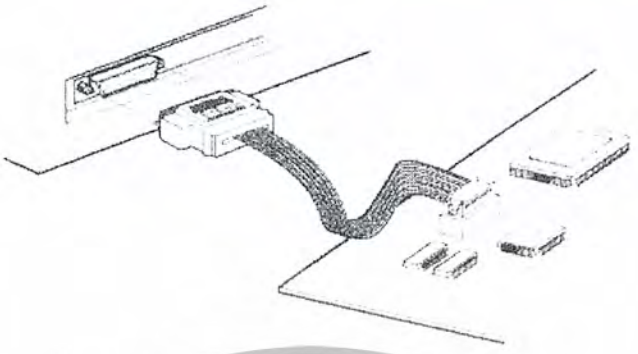
2.4 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว

1. ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพ เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออก แบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึง ภาษา Assembly ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2. มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจร หรือ HDL (Hardware Description Language)เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็น ต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and Optimize ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตราฐาน เดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของ คอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก ดังรูป 2.8 และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสีย ค่าใช้จ่ายเพิ่มแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การโปรแกรมลงในชิพ

2.5 การออกแบบโดยใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์

ในการออกแบบวงจรดิจิตอลนั้นสามารถทำได้โดยการวาดวงจร (Schematic) หรือใช้ภาษาอธิบายพฤติกรรม (Hardware Description Language) ของฮาร์ดแวร์ จากที่ได้กล่าวไปแล้วในบทที่ 1 ในกรณีของการออก แบบวงจรด้วย ASIC ชนิด Full Custom ผู้ออกแบบจะต้องเขียนวงจรด้วย Schematic จากนั้นจะนำวงจรที่ ออกแบบไว้ไปทำการจำลองการทำงาน (Simulate) ซึ่งหากผลออกมาเป็นที่พอใจก็จะต้อง Layout เป็นชั้นสาร และในการออกแบบ ASIC ชนิดนี้ผู้ออกแบบจำเป็นจะต้องทราบถึงเทคโนโลยีที่ใช้ในการสร้างด้วย หลังจากได้ Layout ที่สมบูรณ์แล้วจึงจะส่งไปเข้ากระบวนการสร้างไอซีหรือ Fabrication เพื่อสร้างเป็นชิพไอซีออกมา แต่ในการออกแบบวงจรด้วย FPGA โดยการใช้ Schematic หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำ ได้สะดวกกว่า เนื่องจากวิธีการนี้ผู้ออกแบบไม่จำเป็นต้องคำนึงถึงเทคโนโลยีที่จะใช้สร้างไอซีและที่สำคัญการออก แบบโดยวิธีนี้สามารถแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจร ใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยีสำหรับภาษาที่ใช้ สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL, AHDL และ Verilog เป็นต้น ส่วนรายละเอียด ของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 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) จะไม่เหมาะสมกับโครง สร้างที่มีอยู่ในอุปกรณ์ 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 เช่นมีค่าความหน่วง (Delay) เท่าใด ใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เมื่อมาถึงขั้น ตอนนี้ ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตาม ที่กำหนด

2.5.2 การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในตอนทำการเชื่อม ต่อสัญญาณ (routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำโดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกท (gate), ฟลิป-ฟลอบ (flip-flop) ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGAหลังจากทำขั้นตอน นี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงภายในแต่ละส่วนเท่านั้น หรือที่เรียกว่าความ หน่วงลอจิก(logic delay) ส่วนซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะรวมเอาซอฟต์แวร์ย่อยอื่นๆ อีก เพื่อให้การทำ PPR (Partitioning Placement & Routing) เป็นไปอย่างต่อเนื่อง

2.5.3 การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควร จะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กัน เพื่อจะ ได้ค้นหาเส้นทางได้ (route) ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมี ความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆให้อยู่ใกล้กันโดยเฉพาะส่วน ที่มีการเชื่อมต่อสัญญาณด้วยกันนอกจากนั้นการกำหนดตำแหน่งขา I/O (I/O pin) ตามตำแหน่งขา I/O ของ FPGA บนแผ่น PCB ก็จะมีผลโดยตรงเลยคือซอฟต์แวร์จะวาง I/O ลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่ง บางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา I/O ควรกำหนดตำแหน่งให้เหมาะสม หรือ ไม่ก็ให้ซอฟต์แวร์จัดการเอง

2.5.4 การเชื่อมต่อสัญญาณ (Routing)

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

2.5.5 ความหน่วงด้านเวลา (Delay)

ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นเป็นความหน่วงที่เกิดจากการวางตำแหน่ง (layout) ของอุปกรณ์ ซึ่งผู้ออกแบบไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วง ที่เกิดขึ้นนั้นแยกได้เป็นสองประเภทคือ

1. ความหน่วงลอจิก (Logic delay) เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ (Routing delay) เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA โดยปกติแล้ว ค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการ เชื่อมต่อสัญญาณมักจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์ และเชื่อมต่อสัญญาณผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น และเพื่อ ให้ได้ผลลัพธ์ที่ดีขึ้นค่าความหน่วงที่ได้หลังจากการวางอุปกรณ์ และเชื่อมต่อสัญญาณแล้วจะมีค่าความ หน่วงที่ค่อนข้างแน่นอน ซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้นเป็นไปตามข้อกำหนด หรือไม่

2.5.6 การจำลองการทำงานของวงจร (Simulation)

ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการ ทำงานของโมเดลว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้ จะมีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่ เช่น Model Sim ของบริษัท Model Technology หรือ Max Plus II ของบริษัท Altera ในการจำลองการทำงานของวงจร ควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดล เกิดขึ้นตอน ไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลย ไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้ เกิดข้อผิดพลาด นั่นคือการทำการจำลองการทำงานของวงจร ต้องทำทั้งหลังการเขียนโค้ด, การสังเคราะห์วงจร และการทำ PPR การจำลองการทำงานของวงจรหลังจากที่เขียนโค้ดเสร็จแล้วนั้น ผู้ออก แบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น (functional test) ยังไม่สามารถตรวจ สอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจร แล้ว เพื่อจรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตาม ข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้อง

ในการจำลองการทำงานของวงจรหลังจากที่ทำการวางอุปกรณ์ การเชื่อมต่อสัญญาณ (post layout simulation) แล้วก็มีความสำคัญเช่นกันเพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลย ซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้อง ตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการทำ PPR ในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไรนั่นเอง ในการจำลองการทำงาน ของวงจรควรใช้ ซอฟต์แวร์ตัวเดียวกันตอลดเพื่อจะได้เปรียบเทียบผลที่ได้จากขั้นตอนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่โมเดลผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement & Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็นข้อมูลวงจร (configuration data) ซึ่งอยู่ในรูปของบิตสตรีม (bit stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มี ฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการ ซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับอุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือ ในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้น ในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM หรือ serial PROM ด้วยเพื่อจะใช้งานสะดวกขึ้น คือในการใช้งานโมเดลครั้งต่อไปไม่ ต้องดาวน์โหลดข้อมูลวงจรจากเครื่องคอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิดที่โปรแกรมโดยใช้วิธี EPROM หรือ Anti fuse ก็ไม่จำ เป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไป ข้อมูลที่ดาวน์โหลดลงไปก็ยังคงอยู่ในอุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออก แบบไว้ได้เลย

2.6 เครื่องมือสำหรับการออกแบบ FPGA

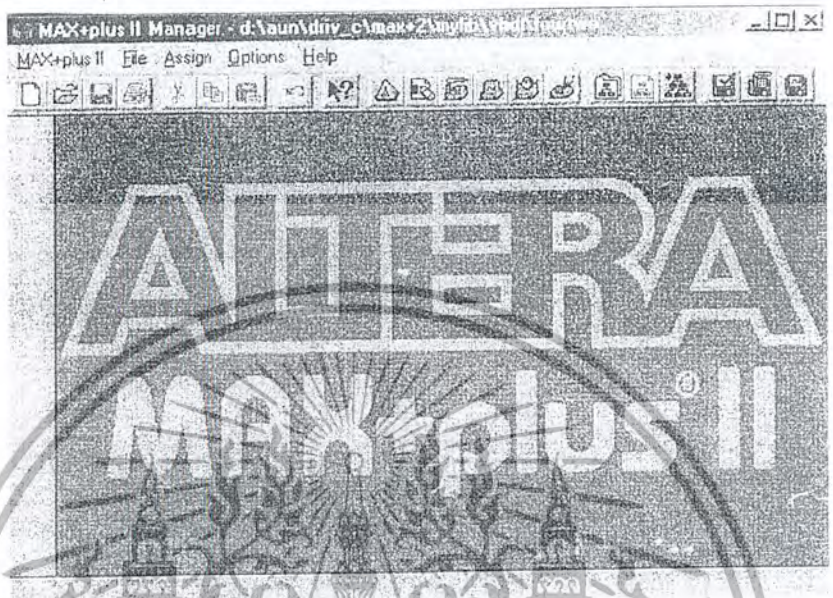
จะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้เวลาน้อยกว่ามากด้วย ส่วน สำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่เขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่วดาวน์โหลดลงใน อุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็น ซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้ สำหรับซอฟต์แวร์ที่ใช้ทำการ จำลองการทำงานของวงจรนั้น ต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบ เพราะโมเดลที่ได้จาก การทำขั้นตอนต่างๆ (ด้วยซอฟต์แวร์ต่างๆ ต้องเอามาจำลองการทำงานได้ และในการจำลองการทำงานของ วงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่ จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์คสเตชั่น (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ ที่ใช้บนพีซี (PC) มากขึ้นซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

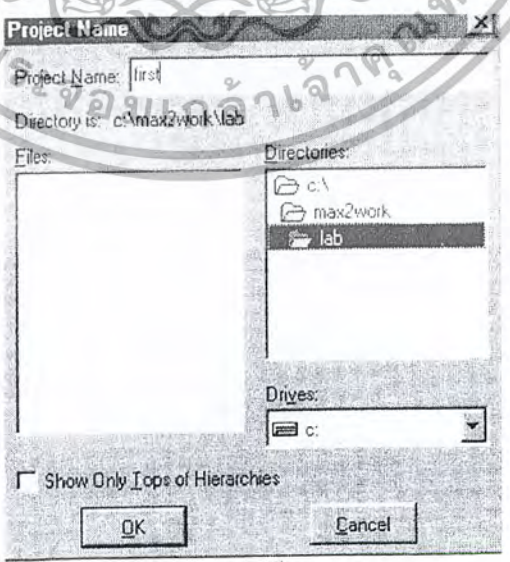
2.7 การใช้งานโปรแกรม Max Plus II เบื้องต้น

2.7.1 Run โปรแกรม Max+Plus II 9.5 BASELINE ขึ้นมา



รูปที่ 2.9 โปรแกรม Max+Plus II 9.5 BASELINE

จากนั้นทำการตั้งชื่อของโปรเจ็ค ที่เราจะสร้างขึ้นมา ชื่อของไฟล์ทุกไฟล์ที่อยู่ในโปรเจ็คเดียวกันจะมีชื่อที่เหมือนกันแตกต่างกันที่นามสกุล การตั้งชื่อโปรเจ็คเริ่มจาก **File / Project / Name** เลือก Folder ที่จะเก็บโปรเจ็คของเราและใส่ชื่อของโปรเจ็คเป็น **First**

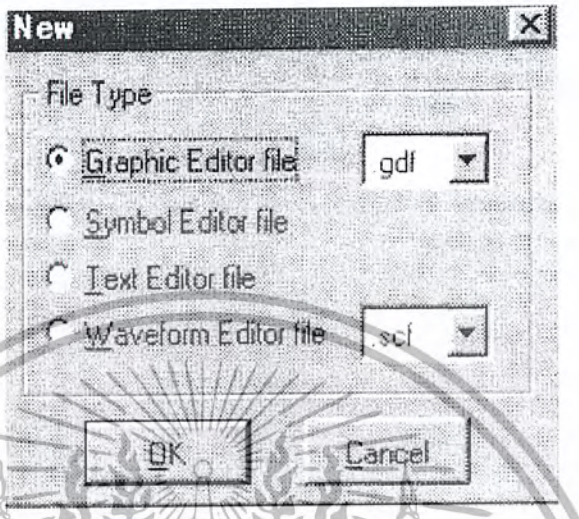


รูปที่ 2.10 การตั้งชื่อโปรเจค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55781

2.7.2 เลือกประเภทไฟล์ที่จะสร้าง **File / New / Graphic Editor File**จะใช้งานขึ้นมาวางไว้ที่ Graphic Form



รูปที่ 2.11 การเลือกประเภทไฟล์

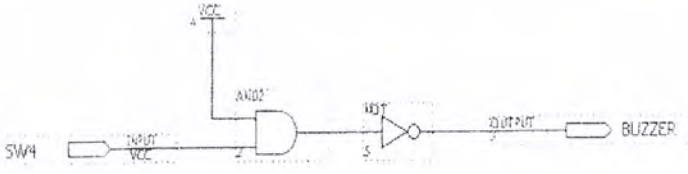
Graphic Editor File: เป็นไฟล์กราฟฟิกส์ที่เรานำอุปกรณ์ต่างๆใน Library มาวางต่อกันได้

Symbol Editor File: เป็นไฟล์ที่ใช้เก็บสัญลักษณ์เพื่อสื่อให้รู้ว่าโมเดลที่เราได้สร้างขึ้นมีอินพุท, เอาท์พุทเป็นอย่างไร

Text Editor File: เป็น Text file ใช้สำหรับเขียน Source code เพื่ออธิบายพฤติกรรมของวงจรหรือโมเดลต่างๆที่เราจะสร้างขึ้นหรือเพื่อไว้สำหรับเก็บข้อความทั่วๆไป

Waveform Editor File: เป็นไฟล์ไว้สำหรับการกำหนดรูปแบบของสัญญาณอินพุทเพื่อใช้ในการ Simulate และไว้สำหรับให้ Max Plus II แสดงผลของเอาท์พุทที่ได้จากการ Simulate

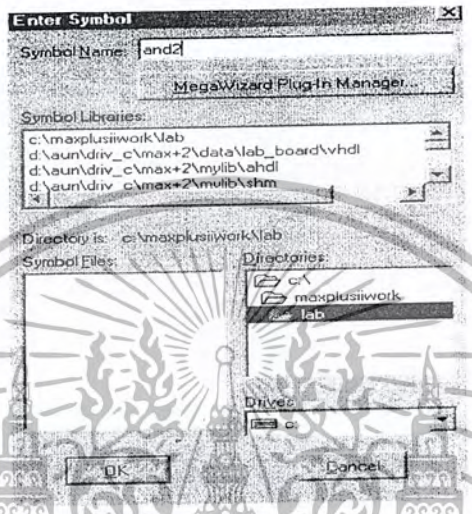
2.7.3 เราจะใช้วงจรดังรูป 2.12 ประกอบคำอธิบาย



รูปที่ 2.12 วงจรตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.12 AND GATE, NOT GATE, INPUT, OUTPUT, VCC จะเป็นอุปกรณ์ที่มาพร้อมกับโปรแกรม Max Plus II ในการที่จะเอาอุปกรณ์ต่างๆ ใน Library ของ Max Plus II มาใช้ให้ทำการ Double Click ที่ Graphic Form จะมี Dialog Enter Symbol ปรากฏขึ้นมา ให้เลือกอุปกรณ์ที่เราต้องการจะใช้งานขึ้นมาวางไว้ที่ Graphic Form



รูปที่ 2.13 การกำหนดสัญลักษณ์

โดยที่

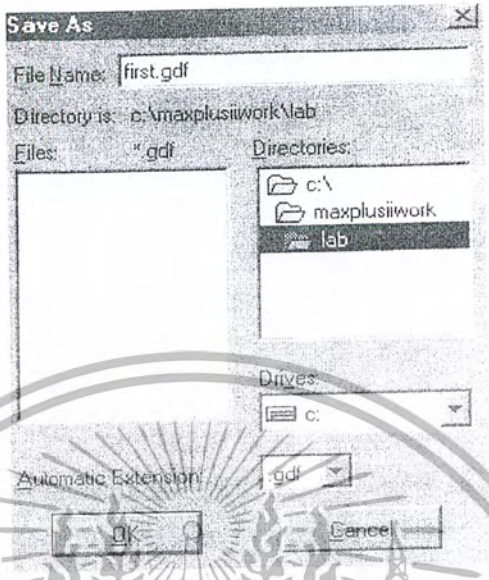
- AND 2 input จะใช้ชื่อ Symbol Name ว่า **and2**
- NOT 1 input จะใช้ชื่อ Symbol Name ว่า **not**
- Input จะใช้ชื่อ Symbol Name ว่า **input**
- Output จะชื่อ Symbol Name ว่า **output**
- VCC จะใช้ชื่อ Symbol Name ว่า **vcc**

อุปกรณ์ทั้งหมดจะอยู่ใน Symbol Library ชื่อ **...\maxplusii\max2lib\primg** เมื่อทำการวางอุปกรณ์จนครบทุกตัวแล้ว ให้ทำการเชื่อมต่ออุปกรณ์ต่างๆเข้าด้วยกันโดยเลื่อน pointer ของ mouse ไปที่ขาต่างๆของอุปกรณ์ pointer ของ mouse จะเปลียนเป็นเครื่องหมาย + ให้ทำการ Click ขวาค้างไว้แล้วลากเส้นไปต่อกับ input หรือ output ของอุปกรณ์ตัวอื่น เมื่อเชื่อมต่ออุปกรณ์ต่างๆจนครบทั้งหมดแล้วให้ทำการเปลียน PIN_NAME ของอุปกรณ์ input กับ output โดยการไป **Double Click** ที่ PIN_NAME ของ input แล้วเปลี่ยนชื่อเป็น **SW1** สำหรับ PIN_NAME ของ output

เปลี่ยนเป็น **BUZZER** เมื่อวาดรูปวงจรเสร็จเรียบร้อยแล้วให้ทำการบันทึกวงจรที่เราสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

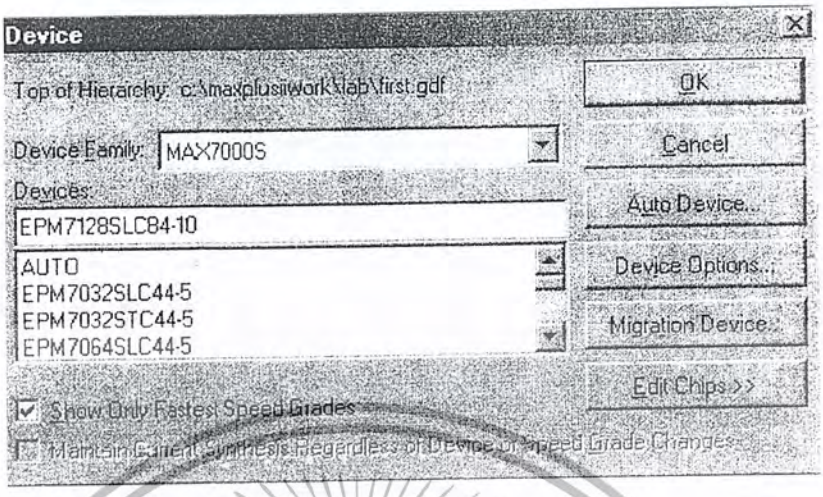
ขึ้นมา **File / Save as** ให้ใส่ชื่อไฟล์ปีน **first.gdf** แล้วก็ OK



รูปที่ 2.14 การบันทึกข้อมูล

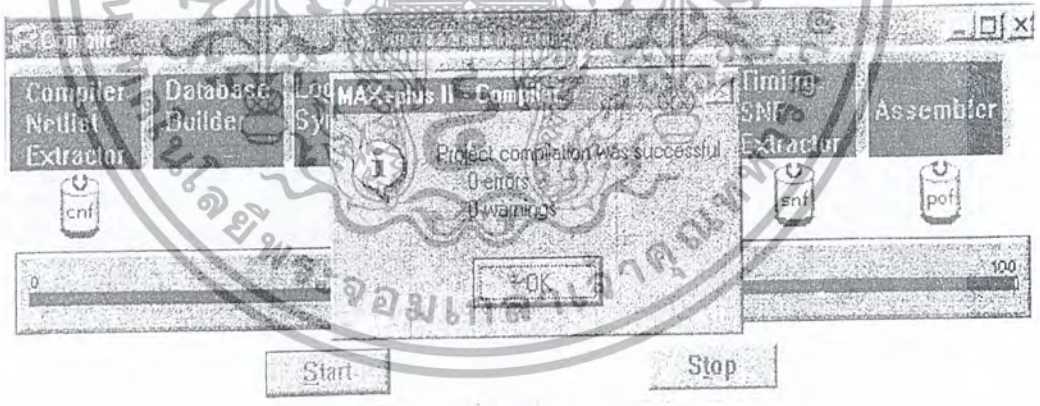
2.7.4 ทำการระบุเบอร์ของชิพที่เราจะใช้ **Assign / Device** จะมีไดอะล็อก Device ปรากฏขึ้นมา ให้ทำการเลือก Device Family เป็น **MAX700S** และเลือก Device เป็น **EPM7128SLC84-10**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 การกำหนดอุปกรณ์

2.7.5 ทำการคอมไพล์วงจรที่เราได้สร้างขึ้นมา **Max+Plus II / Compiler / Start** เมื่อคอมไพล์เสร็จจะมีหน้าต่างรายงานผลการคอมไพล์ error กับ warring หากมีความผิดพลาดเกิดขึ้นจะมีข้อความสีแดงบอกว่า error และจะบอกด้วยว่า error เพราะอะไร

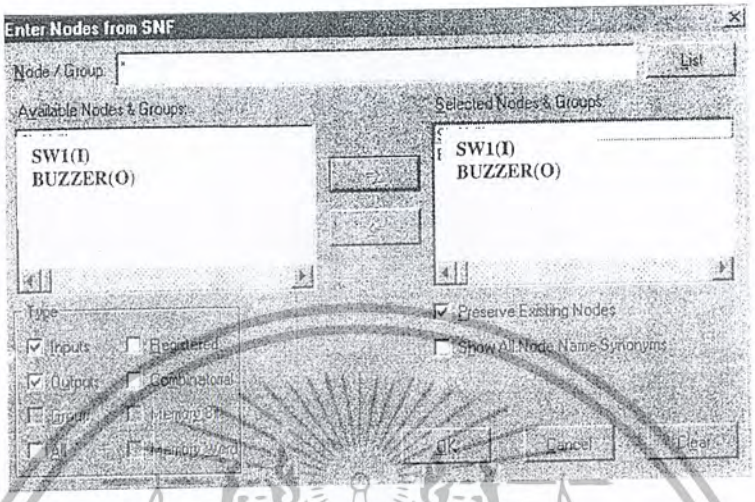


รูปที่ 2.16 รูปการคอมไพล์

2.7.6 ทำการ Simulate โดยจะต้องสร้างไฟล์ Waveform ขึ้นมาก่อน **File / New / Waveform Editer file** (จะได้หน้าต่างดังรูป 2.18) ก่อนที่จะทำการ Simulate จะต้องทำการกำหนดอินพุทให้แก่วงจรก่อน โดยโหลด Noad ต่างๆ เข้ามา **Noad / Enter Noad from SNF จะมีไดอะล็อก** Enter Nodes from SNF ปรากฏขึ้นมาให้คลิกที่ **List** จะมี Noad ต่างๆ ที่อยู่ในวงจรปรากฏขึ้นมาให้เราเลือก Noad ที่เป็น input กับ output ดังรูป 2.17 โดยการกดปุ่มที่มี **เครื่องหมายลูกศรชี้ไปทางขวา**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

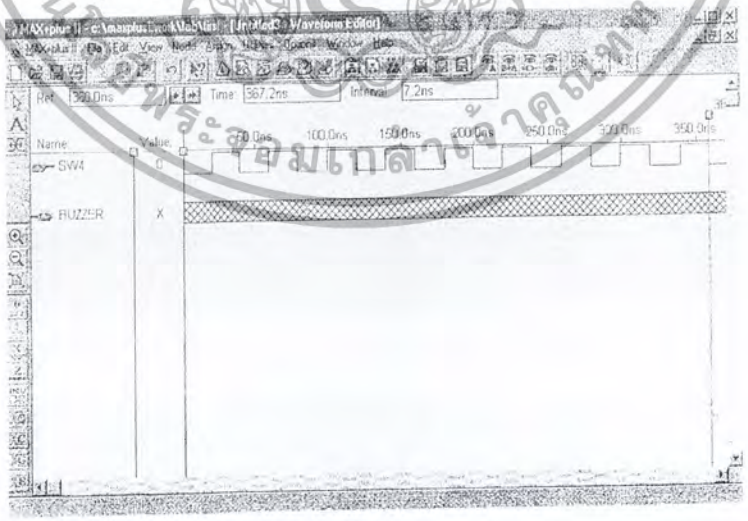
- กำหนดเวลาสิ้นสุดการ Simulate (End Time) ให้กับตัวโปรแกรม **File / End Time** ให้ค่า End Time เท่ากับ **1.0 us** ซึ่งจะเป็นการบอกว่าให้โปรแกรมทำการ Simulate ตั้งแต่ 0.0 us - 1.0 us



รูปที่ 2.17 การกำหนดโหนดต่าง ๆ ที่ใช้ในการวิเคราะห์ Timming

- กำหนดขนาดของกริด **Option / Gride Size** กำหนดให้กริดมีขนาดเท่ากับ **10.00ns**

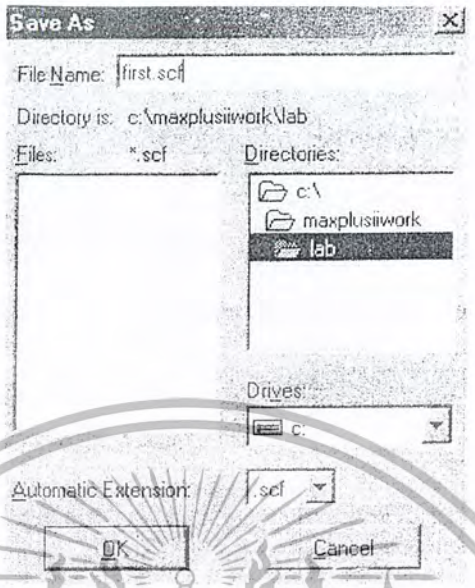
- ทำการกำหนดรูปแบบสัญญาณให้กับ Node Input โดยที่ คลิกที่ SW1 แถบดำปรากฏขึ้นมา หลังจากนั้นทำการกำหนดรูปแบบสัญญาณให้มีลักษณะเป็นพัลซ์ **Edit / Over write / Clock ในช่อง Multiply By** ให้ใส่ **2** แล้วคลิก OK ซึ่งจะเป็นการกำหนดให้สัญญาณในช่วง Logic 1 และ Logic 0 มีค่าเวลาเป็น 2 เท่าของกริด



รูปที่ 2.18 การจำลองการทำงาน

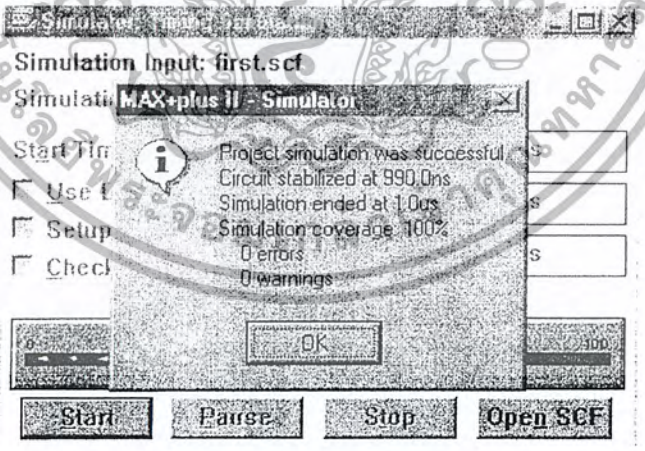
- บันทึกไฟล์ Waveform ที่ได้สร้างขึ้น **File / Save as** กำหนดให้ชื่อไฟล์ที่จะบันทึกเป็น**First.scf**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การบันทึกรูปกราฟ

- ทำการจำลองการทำงานของวงจร Max+**Plus II / Simulation** เมื่อ Simulate เสร็จจะมีไดอะล็อกขึ้นมารายงานผลการ Simulate ว่ามี error หรือ warring หรือไม่ และเราสามารถดูผลการ Simulate ได้ที่ Form ของ Waveform Editor พร้อมทั้งวาด **Waveform** ต่างๆไว้

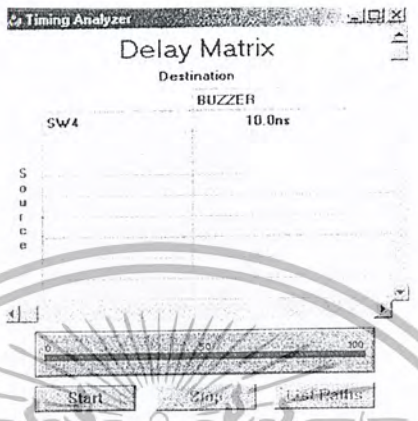


รูปที่ 2.20 การจำลองการทำงานอีกครั้ง

2.7.7 วิเคราะห์ Timing Analyzer จะใช้สำหรับหาค่า Delay Time ระหว่าง Noad ต่างๆ **Max+Plus II / Timing Analysis** จะมี ไดอะล็อก Timing Analyzer ปรากฏขึ้นมาให้เลือกชนิดของการวิเคราะห์ ให้ทำการเลือกวิเคราะห์ค่าเวลาหน่วง **Analysis / Delay Time**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.8 การเปลี่ยนแปลงตำแหน่งขาของชิพเราสามารถเปลียนแปลงตำแหน่งขาของชิพโดย คลิกที่ Max+Plus / Floorplan Editor จะปรากฏฟอร์มของ Floorplan Editor ขึ้นมาเลือกที่เมนู Assign / Back-Annotate Project คลิกที่ **Chip,Pin & Device / OK**



รูปที่ 2.21 การจำลองการหน่วงเวลา

สังเกตุว่าเมื่อผ่านขั้นตอนนี้เราจะเห็นตำแหน่งขาที่โปรแกรม Max+Plus II ได้เลือกให้เรา ที่นี้เราสามารถที่จะเปลี่ยนแปลงตำแหน่งขาต่างๆเหล่านี้ได้เพื่อให้เหมาะสมกับบอร์ดทดลองของเราโดยคลิกที่ Layout / Current Assignment Foorplan ให้ทำการเปลี่ยนตำแหน่งขาของ sw1 เป็นขาที่ 49 และตำแหน่งของขาของ Buzzer เป็นขาที่ 10 โดยการคลิกขวาที่ขา sw1 ค้างไว้แล้วลากมาที่ขา 49 การเปลี่ยนแปลงตำแหน่งของขา Buzzer ก็ให้ทำเช่นเดียวกัน

รูปที่ 2.22 การกำหนดขาใช้งาน

เมื่อทำการเปลี่ยนแปลงตำแหน่งขาของชิพเรียบร้อยแล้วให้ save และ คอมไพล์ใหม่อีกครั้ง Project / Save Compile & Simulate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.9 การโปรแกรมวงจรลงในชิพ Max+**Plus II / Programmer / Program** ไฟล์ข้อมูลที่จะโปรแกรมลงในชิพหากเลือก Device ของ FPGA เป็นอุปกรณ์ประเภท EEPROM – Base FPGA หรือที่เรียกว่า CPLD (เบอร์ IC จะขึ้นต้นด้วย EPM) ไฟล์ที่จะโปรแกรมจะมีนามสกุล *.pof

2.7.10 ทดสอบการทำงานจริงของวงจรที่เราได้โปรแกรมลงในชิพ แล้ว**อธิบายการทำงาน**

2.8 สรุปขั้นตอนการใช้ MAX+PLUS II

2.8.1 การวาดวงจร

1. Run โปรแกรม **Max+Plus II 9.5 BASELINE** ขึ้นมา
2. ตั้งชื่อโปรเจ็คเริ่มจาก **File / Project / Name** เลือก Folder ที่จะเก็บโปรเจ็คของเราและใส่ชื่อของ โปรเจ็คตามต้องการ
3. สร้างไฟล์วงจร **File / New / Graphic Editor File** แล้ววาดวงจรที่ต้องการ แล้ว **save**

2.8.2 การ Compile

4. ระบุเบอร์ของชิพ **Assign / Device** ให้ทำการเลือก Device Family เป็น **MAX700S** และเลือก Device เป็น **EPM7128SLC84-10**
5. คอมไพล์วงจรที่เราได้สร้างขึ้นมา **Max+Plus II / Compiler / Start**

2.8.3 การ Simulate

6. Simulate โดยจะต้องสร้างไฟล์ Waveform ขึ้นมาก่อน **File / New / Waveform Editer file**
7. โหลด Noad ต่างๆ เข้ามา **Noad / Enter Noad from SN**
8. กำหนด **File / End Time** เท่ากับ **1.0 us**
9. กำหนด **Option / Gride Size** มีค่าเท่ากับ **10.00ns**
10. กำหนดรูปแบบสัญญาณให้กับ **Node Input**
11. บันทึกไฟล์ Waveform ที่ได้สร้างขึ้น **File / Save as**
12. การจำลองการทำงานของวงจร **Max+Plus II / Simulation**

2.8.4 การกำหนดขา

13. เปลี่ยนแปลงตำแหน่งขาของชิพ **Max+Plus II / Floorplan Editor**
14. เลือกที่เมนู **Assign / Back-Annotate Project** คลิกที่ **Chip,Pin & Device / OK**
15. คลิกที่ **Layout / Current Assignment Foorplan**
16. ทำการ **Drag and Drop** ขาต่างๆให้ตรงกับความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

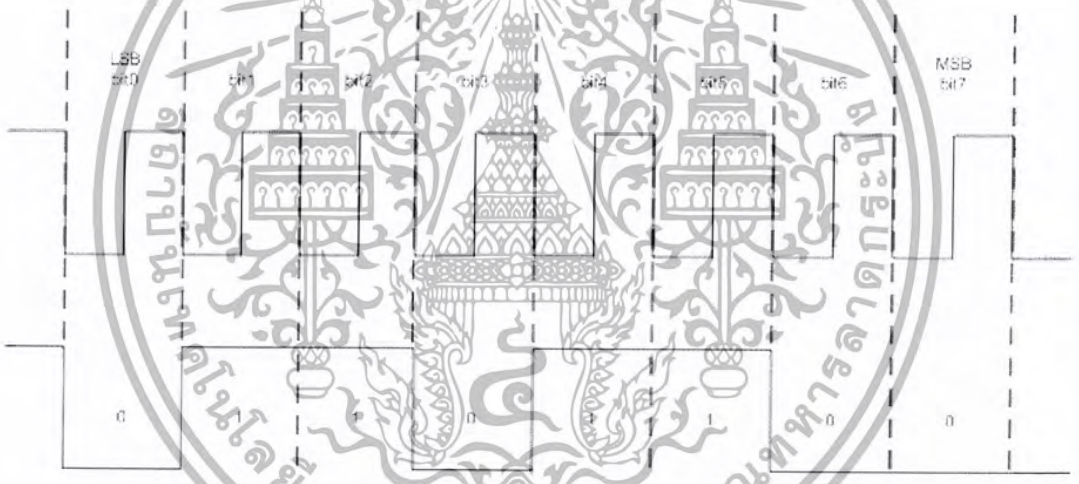
2.8.5 โปรแกรม

17. โปรแกรมวงจรลงในชิพ **Max+Plus II / Programmer / Program**

18. ทดลองการทำงานจริงของวงจร

2.9 การสื่อสารแบบอนุกรม

การสื่อารแบบอนุกรมแบบได้เป็น 2 แบบคือการสื่อสารข้อมูลแบบซิงโครนัสและการสื่อสารข้อมูลแบบอะซิงโครนัส การสื่อสารแบบอะซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีบอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายไปในการเชื่อมต่ออย่างน้อย 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์ รูปที่ 2.23 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.23 รูปแบบอย่างง่ายที่สุดของการส่งข้อมูลแบบอนุกรม

2.9.1 การส่งข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนการส่งข้อมูลแบบซิงโครนัส แต่จะใช้ในการกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากันซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการส่งทอดข้อมูล หรือ บอดเรต(Baud rate) มีหน่วยเป็นบิตต่อวินาที (bit per secone: bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วนกันคือ

1. บิตเริ่มต้น (Start bit) ซึ่งจะมีขนาด 1 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตสุดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

รูปที่ 1-2 แสดงรูปแบบของข้อมูลแบบอะซิงโครนัสเมื่อไม่มีข้อมูลที่จะส่งขา DATA จะมีสถานะลอจิก"1"ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิตเรียกบิตนี้ว่าบิตเริ่มต้นจากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้จากนั้นตามด้วยบิตพาริตี้ซึ่งใช้เพื่อการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1,1.5 หรือ 2 บิตเพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

รูปที่ 2.24 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

2.10 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารข้อมูล

หน้าที่หลักของ UART คือทาหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนจะส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะเป็นตัวส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลต่างๆให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการส่งข้อมูล (บอดเรต) ,รูปแบบการส่งข้อมูล,ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล(ผิดพลาดจากพาริตี้,เฟรมข้อมูล,โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1-65,535 UART สามารถส่งข้อมูลได้ทั้งแบบฮาล์ฟดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพล็กซ์ และฟลูดูเพล็กซ์ โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งข้อมูลแบบทิศทางเดียว ส่วนการส่งแบบฟลูดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียว

2.11 โทโปโลยีเครือข่าย

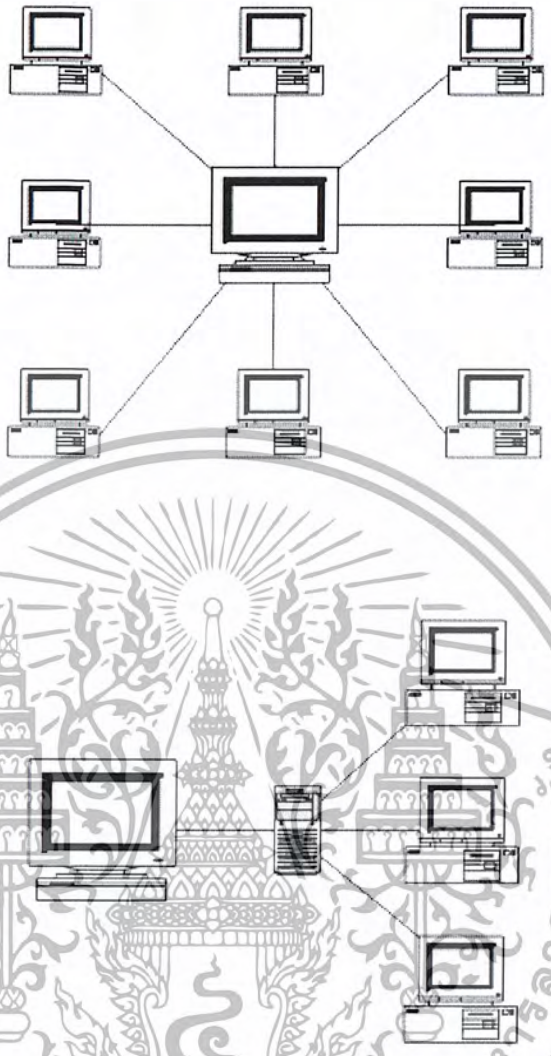
โทโปโลยีเครือข่าย (network topology) หมายถึง ความสัมพันธ์เฉพาะระหว่างส่วนต่างๆของเครือข่ายคอมพิวเตอร์ในแง่ของ การต่อถึงระหว่างกัน (interconnection) หน้าที่การทำงาน และตำแหน่ง ที่ตั้ง โดยทั่วไปโทโปโลยีเครือข่ายอาจแบ่งได้เป็นหลายแบบด้วยกัน เราเรียกคอมพิวเตอร์ซึ่งมีส่วนร่วมในการส่งผ่านข้อมูลข้ามเครือข่ายว่า โนด (node) ซึ่งอาจเป็นคอมพิวเตอร์ตัวเดียวโดดๆ หรือแผ่นบอร์ดภายในคอมพิวเตอร์ที่มีหน้าที่เฉพาะสำหรับการเคลื่อนย้ายข้อมูลผ่านเครือข่าย โนดอาจทำการติดต่อกับคอมพิวเตอร์อื่นที่มีขนาดใหญ่กว่า หรือคอมพิวเตอร์หลัก (host) ซึ่งสนับสนุนการทำงานของผู้ใช้โดยการเก็บและจัดการข้อมูลของผู้ใช้ ในการประยุกต์ใช้งานบางอย่าง เราอาจได้หน้าที่การทำงานของเครือข่ายอย่างหนึ่งจากโนด และได้หน้าที่การทำงานอื่นจากคอมพิวเตอร์หลัก ยกตัวอย่างเช่น โนดอาจทำหน้าที่กำหนดเส้นทางการส่งข้อมูลข้ามเครือข่าย ในขณะที่คอมพิวเตอร์หลักอาจทำหน้าที่ตรวจจับและแก้ไขความผิดพลาดในการส่งข้อมูล การแบ่งโทโปโลยีแบ่งได้เป็น 2 ลักษณะคือ โทโปโลยีแนวราบ และ โทโปโลยีแนวตั้ง แต่ในที่นี้จะอธิบายเกี่ยวกับโทโปโลยีแนวราบเท่านั้น เนื่องจากโทโปโลยีแนวราบนั้นนำไปใช้เป็นส่วนประกอบในการทำระบบนี้

2.11.1 โทโปโลยีแนวราบ(Horizontal topologies) โทโปโลยีแนวราบที่ใช้กันมาก ในระบบเครือข่ายมี 3 ประเภท คือ

1) โทโปโลยีแบบดาว(Star Topology)

โทโปโลยีแบบดาวเป็นโทโปโลยีแบบง่ายที่สุดใช้แทนโครงข่ายที่มี โนดหลายๆโนดเชื่อมโยงเข้ากับโนดสวิตเชิงศูนย์กลาง (central switching node) ดังแสดงในรูป 2.25 การสื่อสารในเครือข่ายทั้งหมดจะต้องเดินทางผ่านศูนย์กลางนี้ หน้าที่สำคัญของโนดศูนย์กลางคือ การทำหน้าที่อำนวยความสะดวกในกางสื่อสางระหว่างโนดรอบนอกของรูปดาว กล่าวคือ เหมือนกับเป็นอุปกรณ์ควบคุมการสื่อสาร (communication controler) ข้อดีของโทโปโลยีแบบดาวคือ โครงสร้างมีรูปแบบที่ง่าย และเราสามารถเพิ่มหรือลอจำนวนโนดรอบนอกของรูปดาวได้ โดยเพียงแต่แก้ไขตัวโนดศูนย์กลาง ส่วนข้อเสียที่สำคัญคือ เครือข่ายมีโอกาสล้มเหลวได้ง่าย ถ้าโนดศุนกลางเกิดล้มเหลวขึ้นมาดังนั้นโดยปกติในทางปฏิบัติจะมีการสำรอง (back up) โนดศูนย์กลางเพื่อทำให้เครือข่ายเชื่อถือได้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



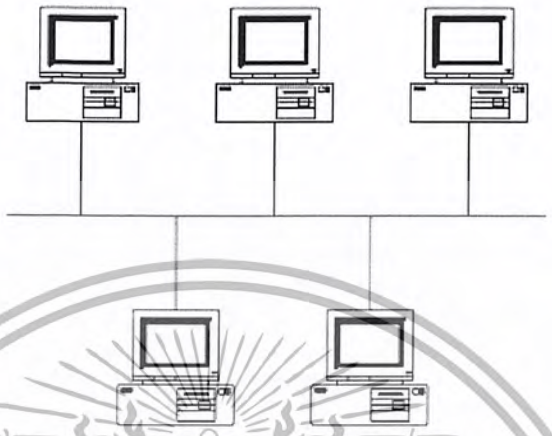
รูปที่ 2.25 โทโปโลยีแบบดาว (Star Topology)

2) โทโปโลยีแบบบัส (Bus topology)

ในโทโปโลยีแบบบัส โนดทุกโนดของเครือข่ายติดต่อสื่อสารผ่านตัวกลางในการส่งข้อมูลตัวเดียวกัน (common transmission medium) ดังในรูป 2.26 ทั้งสัญญาณที่เป็นข้อมูลและสัญญาณควบคุมถูกแพร่กระจาย(broadcast) ไปยังโนดทุกโนดพร้อมกันโนดแต่ละโนดมีแอดเดรสเฉพาะตัวดังนั้นถึงแม้ว่าโนดทุกโนดจะได้รับข่าวสารเหมือนกันทั้งหมดจากบัส จะมีเพียงโนดที่มีแอดเดรสตรงกับที่กำหนดโดยผู้ส่งเท่านั้นที่จะรับทราบข้อมูลนั้นวิธีการที่โนดเข้าถึงตัวกลางเพื่อส่งข้อมูลนั้นมีหลายวิธี เช่น คอมพิวเตอร์แต่ละเครื่องอาจได้รับอนุญาตให้เข้าถึงตัวกลางได้จากคอมพิวเตอร์ตัวอื่นภายในเครือข่าย(เรียกว่า วิธีการเข้าถึงตัวกลางแบบ token – passing) 1หรือได้รับอนุญาตจากโนดหลักหรือ master node (วิธีนี้เรียกว่าการเข้าถึงตัวกลางแบบ polling)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของโทโปโลยีแบบบัสคือ 1) เพิ่มหรือลดจำนวนผู้ใช้ง่าย 2)ส่งข้อมูลได้ด้วยความเร็วสูง เนื่องจากไม่ต้องเลือกเส้นทางในการส่งข้อมูลผ่านโนดระหว่างทาง



รูปที่ 2.26 โทโปโลยีแบบบัส (Bus Topology)

3) โทโปโลยีแบบวงแหวน(Ring topology)

เป็นโทโปโลยีแบบวงแหวนมีลักษณะดังแสดงในรูป 2.27 กล่าวคือ มีเส้นทางระหว่างโนดของโครงข่ายบรรจบเป็นวงกลม โดยที่แต่ละโนดต่อถึงกันกับโนดอื่น 2 โนดที่อยู่ใกล้กัน การไหลของข้อมูลอาจเป็นแบบไหลไปทิศทางเดียว (unidirectional) หรือไหลได้ 2 ทิศทาง (bidirectional)

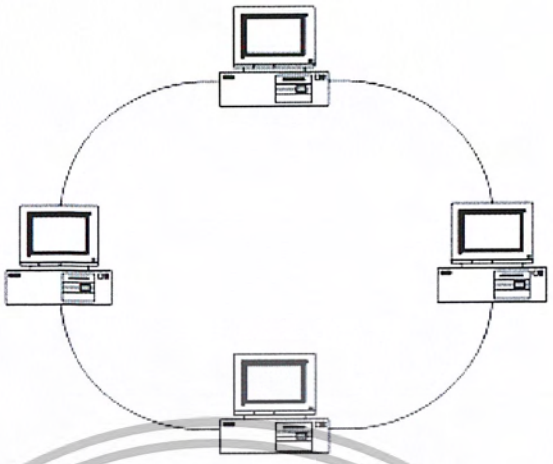
วิธีการในการแบ่งกันใช้ตัวกลางรูวงแหวนนี้มี 2 วิธีดังนี้

1) ให้ผู้ส่งแต่ละคนแย่งกันเข้าถึงตัวกลาง

2) ให้สิทธิในการเข้าถึงตัวกลางโดยการ token

ในประเทศสหรัฐอเมริกา token- passing เป็นวิธีการเข้าถึงตัวกลางที่นิยมมากที่สุดสำหรับโทโปโลยีแบบวงแหวน วิธี token- passing ทำให้เกิดการประวิงเวลาที่โนดในการตรวจสอบดูว่าข้อมูลที่ส่งเข้ามายังโนดนั้นๆ เป็น token หรือข่าวสารแบบอื่น ข้อดีของ token –passing ring คือ สมรรถนะที่มีเสถียรภาพ แม้ว่าจะอยู่ภายใต้สภาวะที่มีโหลดมาก ข้อเสียคือ เกิดการประวิงเวลาค่อนข้างมาก ตั้งแต่เริ่มมีคงามต้องการเข้าถึงตัวกลางจนกระทั่งได้รับ token และในกรณีที่ token สูญหายไปกรประวิงเวลาจะยิ่งนานขึ้นไปอีกเนื่องจากต้องรอให้โครงข่ายฟื้นตัวสู่สภาพปกติ เสียก่อนอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 โทโปโลยีแบบวงแหวน (Ring Topology)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ

3.1 แนวคิดในการออกแบบ

แนวคิดในการออกแบบระบบควบคุมสัญญาณไฟจราจรนั้นมีแนวคิดที่ว่าในการปล่อยรถในแต่ละเส้นทางนั้นเราจะปล่อยรถในเส้นทางที่มากที่สุดก่อน โดยเราสามารถควบคุมการจราจรของในแต่ละแยกให้สัมพันธ์ได้โดยอาศัยเส้นทางที่ร่วมกันดังแสดงในรูปที่ 3.1 โดยในโครงงานระบบควบคุมสัญญาณไฟจราจรนี้เราได้ทำการจำลอง 2 แยกที่มีเส้นทางร่วมกันคือ แยกเพลินจิตรและ แยกชิดลม โดยเส้นทางที่รวมกันคือ w5 และ w6

รูปที่ 3.1 แสดงการกำหนดชื่อของเส้นทาง

3.2 ส่วนประกอบต่างๆของระบบ

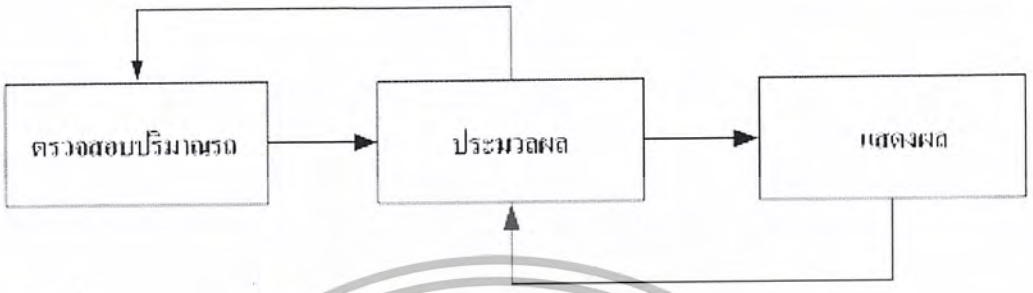
ระบบควบคุมสัญญาณไฟจราจรนี้สามารถแบ่งการทำงานของระบบออกได้เป็น 3 ส่วนใหญ่ๆ ได้ดังนี้คือ

1. การตรวจสอบปริมาณรถ
2. การประมวลผล
3. การแสดงผล

โดยการทำงานของทั้ง 3 ส่วนนี้จะทำงานสัมพันธ์ซึ่งสามารถแสดงได้รูปที่ 3.2 โดยการสื่อสารข้อมูลระหว่างส่วนของการตรวจสอบปริมาณรถและส่วนของการประมวลผลนั้นจะใช้การสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

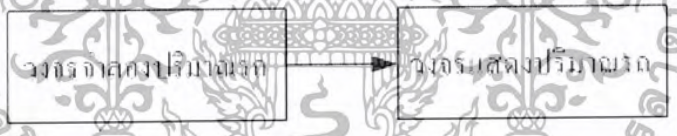
ข้อมูลอนุกรมโดยมีอัตราการสื่อสารข้อมูลที่ 9600 บิตต่อวินาทีและเป็นการสื่อสารข้อมูลแบบแบ่งช่วงเวลากัน (Time Division Multiplex)



รูปที่ 3.2 บล็อกไดอะแกรมแสดงการทำงานของระบบ

3.3 ส่วนการตรวจสอบปริมาณรถ

เป็นส่วนที่ใช้ในการตรวจสอบปริมาณรถในแต่ละเส้นทางว่าในขณะนั้นมีรถอยู่ในเส้นทางนั้นปริมาณเท่าใด โดยเราสามารถแสดงการทำงานของส่วนนี้ได้ตามรูปที่ 3.3



รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงานของส่วนตรวจสอบปริมาณรถ

3.3.1 วงจรจำลองปริมาณรถ

วงจรในส่วนนี้จะเป็นการจำลองปริมาณรถที่มีอยู่ในแยก โดยผลลัพธ์ต้องการของวงจรในส่วนนี้คือสัญญาณพัลซ์ที่มีความถี่ประมาณ 0.72 – 1.44 Hz โดยความถี่ของสัญญาณพัลซ์ที่ได้นี้จะแสดงถึงความเร็วของรถที่วิ่งในเส้นทางนั้นและสมมุติให้พัลซ์ 1 ลูกแทนจำนวนรถ 1 คัน ดังนั้นในการออกแบบวงจรนี้เราจะใช้IC เบอร์ 555 ซึ่งเป็นIC Timer ในการออกแบบวงจรเพื่อให้ได้ผลลัพธ์ตามที่เราต้องการ โดยจะออกแบบเป็นวงจรอะสเตเบิลมัลติไวเบรเตอร์ซึ่งสามารถคำนวณค่าอุปกรณ์ต่างได้ดังนี้

จากสูตร

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} \quad \text{Hz} \qquad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉะนั้นถ้าเรากำหนดให้ R_2 = 50 KΩ, C_1 = 10 μF และ f = 0.72 Hz เราจะได้ R_2 จากสมการ

$$R_1 = \left(\frac{1.44}{(f * C_1)}\right) - 2R_2 \quad \Omega \qquad (3.2)$$

ดังนั้นเราจะได้ R_1 = 100 KΩ

และถ้า R_1 = 0 Ω, R_2 = 50 KΩ และ C_1 = 10 μF เราจะได้สัญญาณพัลซ์ที่ความถี่ 1.44 Hz เพราะฉนั้นเพื่อให้ได้สัญญาณพัลซ์ตามที่เราต้องการเราจึงเลือกใช้R_1ให้เป็นตัวต้านทานปรับค่าได้ ซึ่งสามารถแสดงวงจรได้ดังรูปที่ 3.4

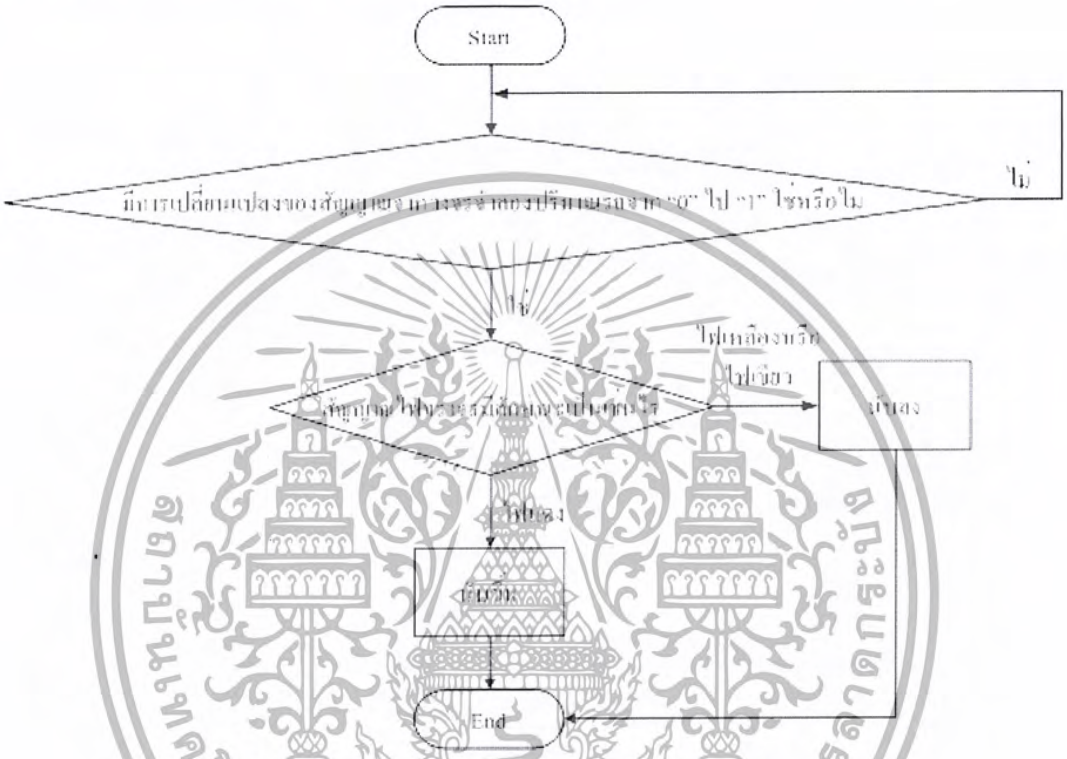
รูปที่ 3.4 วงจรจำลองปริมาณรถ

3.3.2 วงจรแสดงปริมาณรถ

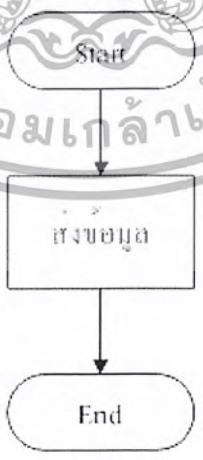
วงจรในส่วนนี้เป็นวงจรที่ทำหน้าที่ในการแสดงปริมาณรถ และส่งข้อมูลของปริมาณรถที่มีอยู่ในเส้นทางนั้นออกไปให้กับส่วนประมวลผล เมื่อมีการร้องขอมาจากส่วนประมวลผล โดยสัญญาณข้อมูลที่ส่งออกไปนั้นจะถูกส่งออกไปเป็นแบบอนุกรม ที่มีอัตราการส่งและรับข้อมูลที่ 9600 บิตต่อวินาที สำหรับวงจรในส่วนนี้จะออกแบบโดยใช้ไมโครคอลโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในส่วนของการแสดงผลนั้นจะใช้ 7-Segment ในการแสดงผล สามารถแสดงแนวคิดในการเขียนโปรแกรมของวงจรแสดงปริมาณรถได้ดังรูปที่ 3.5 และ 3.6 โดยเราจะแบ่งโปรแกรมออกเป็น 2 ส่วนหลักคือ ส่วนแสดงปริมาณรถ และ ส่วนส่งข้อมูลไปยังส่วนประมวลผล ในส่วนของการส่งข้อมูลนั้นจะทำงานเมื่อมีการร้องขอมาจากส่วนประมวลผล



รูปที่ 3.5 แนวคิดการเขียนโปรแกรมส่วนแสดงปริมาณรถ



รูปที่ 3.6 แนวคิดการเขียนโปรแกรมส่วนส่งข้อมูล

เนื่องจากความสามารถในการขับกระแสที่พอร์ตของไมโครคอนโทรลเลอร์เบอร์ AT89C2051 ซึ่งเป็นไมโครคอลโทรเลอร์ที่เราเลือกใช้ นั้นมีความสามารถในการขับกระแสที่ไม่สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นักแต่ในการแสดงผลซึ่งเราใช้ 7- Segment แบบคอมมอนอาโหนดซึ่งจำเป็นต้องใช้กระแสที่ค้อนข้างสูงดังนั้นเพื่อความปลอดภัยของตัวอุปกรณ์และการแสดงผลที่ชัดเจนเราจึงใช้BJTทรานซิสเตอร์ชนิดPNPมาช่วยในการขับกระแสในที่นี้เราใช้ทรานซิสเตอร์เบอร์ 2N4403 โดยเราสามารถคำนวณหาค่าของอุปกรณ์ต่างๆในวงจรได้จากเงื่อนไขต่างๆของ 7 – Segment และ ทรานซิสเตอร์ ดังนี้คือ ทรานซิสเตอร์มี β = 180 และ $I_{c(max)}$ = 1 A โดยทราซิสเตอร์จะต้องขับกระแสสูงสุดที่ 70 mA ดังแสดงในรูปที่ 3.7โดยสามารถหาค่าR_BโดยอาศัยทฤษฎีของKVLและสามารถเขียนสมการ KVLได้ดังนี้

$$V_{EE} = I_B R_B + \left|V_{BE(FA)}\right| \qquad (3.3)$$

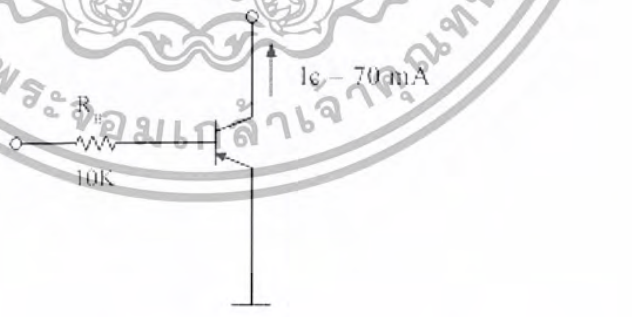
$$\therefore \quad R_B = \frac{V_{EE} - \left|V_{BE(FA)}\right|}{I_B} \qquad (3.4)$$

แทนค่า V_{EE} = 5 V. และ $\left|V_{BE(FA)}\right|$ = 0.7 และ $I_c = \beta I_B$

$$\therefore \quad R_B = \frac{(5-0.7)(180)}{(70\times10^{-3})}$$

$$= 11.1 \quad k\Omega$$

แต่เราเลือกใช้ค่า 10 kΩ ดังนั้น กระแสสูงสุดที่ได้จะมากกว่า 70 mA เล็กน้อย

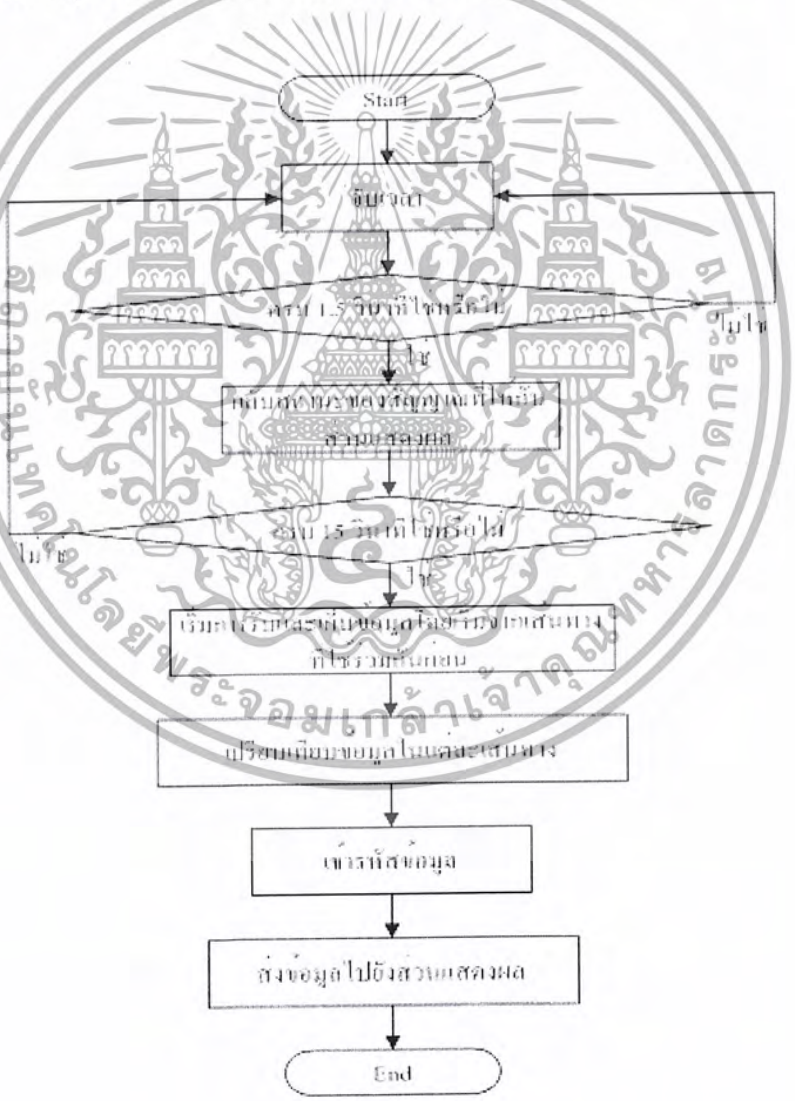


รูปที่ 3.7 วงจรทราซิสเตอร์ที่ขับกระแสให้กับ 7 – Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนการประมวลผล

การทำงานของส่วนนี้จะทำหน้าที่เป็นส่วนประมวลผลและตัดสินใจว่าควรจะปล่อยหรือหยุดรถในเส้นทางใด ด้วยหลักการที่ว่าถ้าเส้นทางใดมีปริมาณรถมากที่สุดก็จะปล่อยเส้นทางนั้นก่อน และผลลัพธ์ที่ได้จากการประมวลผลนี้ก็จะถูกส่งไปให้ยังส่วนแสดงผลต่อไป โดยจะมีการร้องขอข้อมูลจากวงจรแสดงปริมาณรถทุกๆ 15 วินาที และจะมีการส่งสัญญาณนาฬิกาไปยังส่วนแสดงผลที่ความถี่ประมาณ 0.67 Hz หรือก็คือมีการเปลี่ยนแปลงสถานะของสัญญาณทุกๆ 1.5 วินาที และข้อมูลที่ส่งมาให้นั้นจะเป็นข้อมูลแบบอนุกรมที่มีอัตราการส่งและรับข้อมูลที่ 9600 บิตต่อวินาที และในส่วนนี้เราจะใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 เป็นตัวประมวลผลข้อมูล สำหรับการเขียนโปรแกรมควบคุมวงจรประมวลผลนั้นสามารถแสดงได้ดังรูปที่ 3.8



รูปที่ 3.8 แนวคิดการเขียนโปรมแกรมควบคุมวงจรประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ส่วนการแสดงผล

3.5.1 การแสดงตารางสถานะของแยกชิดลมและแยกเพลินจิตร

ตารางสถานะแยกชิดลม

NS= A,B,C (รหัสสถานะสัญญาณไฟ state ถัดไป)

PS= X,Y,Z (รหัสสถานะสัญญาณไฟปัจจุบัน)

รหัสสถานะสัญญาณไฟปัจจุบัน คือ code หรือรหัสสัญญาณที่ส่งไปให้หลอดไฟจราจรแสดงออกเป็นลักษณะของสัญญาณไฟที่แสดงอยู่ ณ เวลาปัจจุบัน โดยรหัสนี้ได้รับการป้อนกลับมาจากสัญญาณเอาท์พุท(output)

รหัสสถานะสัญญาณไฟ state ถัดไป คือ code หรือรหัสสัญญาณที่จะต้องแสดงต่อไปหลังจากสถานะปัจจุบันโดยcodeนี้ได้มาจากการแสกนค่าของจำนวนรถแต่ละช่องทาง แล้วนำมาเปรียบเทียบกัน

การทำงานในส่วนนี้คือเริ่มแรกเราจะดูลักษณะสัญญาณไฟในปัจจุบันนั้นว่าเป็นอะไร หลังจากนั้นประมาณ 15 วินาทีในส่วนของการตรวจจอบปริมาณรถจะทำการเช็คปริมาณรถเพื่อเปรียนเทียบกันแล้วส่ง code ออกมาซึ่งก็คือลักษณะสัญญาณไฟ state ถัดไปนั่นเอง แล้วหลังจากนั้นลักษณะสัญญาณไฟจราจรที่แยกก็จะเปลี่ยนไปตามอินพุทดัง ตารางที่ 3.1 (แยกชิดลม) และตารางที่ 3.2 (แยกเพลินจิตร)

XYZ \ ABC	000	001	010	011	100	101	110	111
000	000	011	100	011	100	000	111	111
001	000	001	101	001	101	000	111	111
010	110	110	010	110	010	110	111	111
011	001	001	100	001	100	001	111	111
100	111	111	010	111	010	111	111	111
101	111	111	010	111	010	111	111	111
110	000	001	111	001	111	000	111	111
111	000	001	010	001	010	000	111	111

ตารางที่ 3.1 ตารางแสดงสถานะแยกชิดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสถานะแยกเพลินจิตร

NS= D,E,F (รหัสสถานะสัญญาณไฟ state ถัดไป)

PS= U,V,W (รหัสสถานะสัญญาณไฟปัจจุบัน)

DEF / UVW	000	001	010	011	100	101	110	111
000	000	101	000	101	101	101	101	110
001	011	001	011	011	011	011	001	110
010	100	100	100	010	010	010	100	110
011	000	110	000	010	010	010	110	110
100	000	001	000	110	110	110	001	110
101	110	001	110	010	010	010	001	110
110	000	001	000	010	010	010	001	110
111	000	001	000	010	010	010	001	110

ตารางที่ 3.2 ตารางแสดงสถานะแยกเพลินจิตร

ต่อไปคือการกำหนดการติดดับของลักษณะสัญญาณไฟทั้งสองแยกโดย code ที่ได้ก็คือรหัสที่ใช้กำหนดเอาพุทว่ารหัสนี้ลักษณะสัญญาณไฟจราจรจะเป็นอย่างไรโดยถ้าเป็น 0 คือกำหนดให้ไฟดวงนั้นติด ถ้าเป็น 1 คือกำหนดให้ดับซึ่งการกำหนดลักษณะสัญญานไฟจะเป็นดังตารางที่ 3.3 และ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code(XYZ)	A	B	C	D	E	F	G	H	I
000	1	0	0	0	0	1	1	0	0
001	1	0	0	0	0	1	0	0	1
010	0	0	1	1	0	0	0	0	1
011	1	0	0	0	0	1	0	1	0
100	0	1	0	0	0	1	0	1	0
101	0	1	0	0	0	1	0	0	1
110	0	0	1	0	1	0	0	0	1
111	0	0	1	0	0	1	0	0	1

ตารางที่ 3.3 ตารางลักษณะสัญญาณไฟจราจรแยกชิดลม

Code(UVW)	J	K	L	M	N	O	P	Q	R
000	1	0	0	0	0	1	0	0	1
001	0	0	1	1	0	0	0	0	1
010	0	0	1	0	0	1	1	0	0
011	0	0	1	0	1	0	0	0	1
100	0	0	1	0	0	1	0	1	0
101	0	1	0	0	0	1	0	0	1
110	0	0	1	0	0	1	0	0	1
111	0	0	1	0	0	1	0	0	1

ตารางที่ 3.4 ตารางลักษณะสัญญาณไฟจราจรแยกเพลินจิตร

จากตารางที่ 3.3 และ 3.4 เราจะได้สมการของตำแหน่งสัญญาณไฟต่างๆดังนี้

$A = \overline{X}\overline{Y} + \overline{X}Z$ $\quad D = \overline{X}Y\overline{Z}$ $\quad G = \overline{X}YZ$

$B = X\overline{Y}$ $\quad E = XY\overline{Z}$ $\quad H = X\overline{Y}\overline{Z} + \overline{X}YZ$

$C = XY + Y\overline{Z}$ $\quad F = \overline{Y} + Z$ $\quad I = \overline{Y}Z + Y\overline{Z} + XY$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$J = \overline{U}V\overline{W}$ $M = \overline{U}\overline{V}W$ $P = \overline{U}V\overline{W}$

$K = U\overline{V}W$ $N = \overline{U}VW$ $Q = U\overline{V}\overline{W}$

$L = V + \overline{U}W + U\overline{W}$ $O = U + \overline{W}$ $R = W + \overline{U}\overline{V} + UV$

จากนั้นนำสมการเหล่านี้ไปเขียน Schematic ลงใน FPGA ต่อไป

3.5.2 วิธีการลดรูปสมการ แยกชิดลมและแยกเพลินจิตร

การลดรูปแยกชิดลม

สมการ X แยกชิดลม

A = 0

B = 0, CD \ EF	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	1	1	0
10	1	1	1	1

B = 1, CD \ EF	00	01	11	10
00	1	0	1	0
01	1	1	0	0
11	1	1	1	0
10	0	1	1	1

A = 1

B = 0, CD \ EF	00	01	11	10
00	1	0	0	1
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

B = 1, CD \ EF	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	0	1	1	0
10	1	1	1	0

$= DE + A\overline{B}F + A\overline{B}\overline{D}\overline{E} + AB\overline{C}D\overline{F} + AB\overline{C}E\overline{F} + \overline{A}\overline{B}D\overline{F} + \overline{A}\overline{B}E\overline{F} + \overline{A}B\overline{C}\overline{D}\overline{E}$

$+ \overline{A}B\overline{C}F + \overline{A}BCD\overline{F} + \overline{A}BCE\overline{F}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ Y แยกชิดลม

B = 0

A = 0

EF \ CD	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	1	0
10	0	1	1	0

B = 1

EF \ CD	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	1	1	0
10	1	1	1	0

A = 1

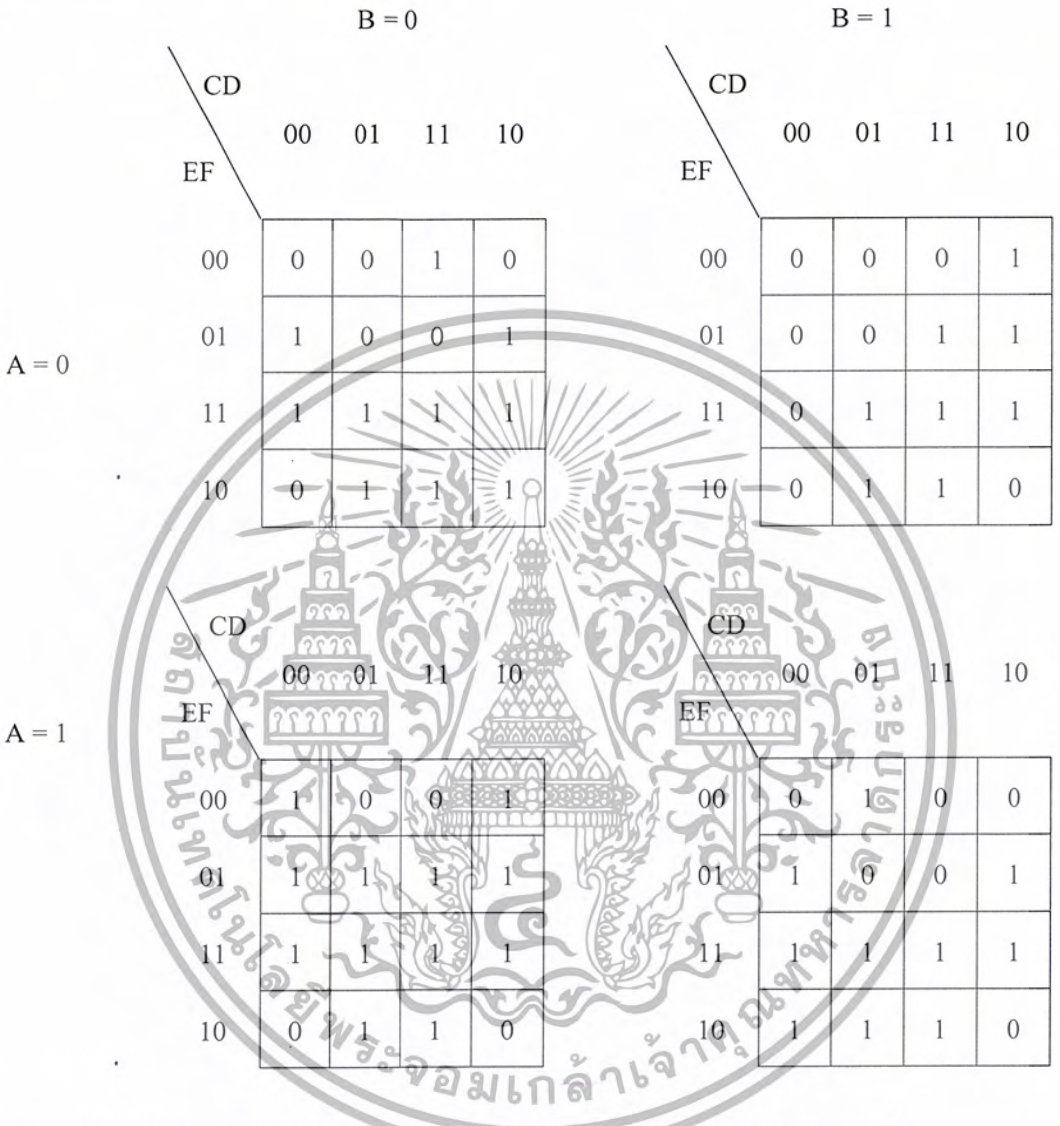
EF \ CD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

EF \ CD	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	1	1	0
10	1	1	1	1

$$= A\overline{B} + AE\overline{F} + AD\overline{F} + DE + \overline{A}\overline{B}\overline{C} + \overline{B}CD\overline{F}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ Z แยกชิดลม

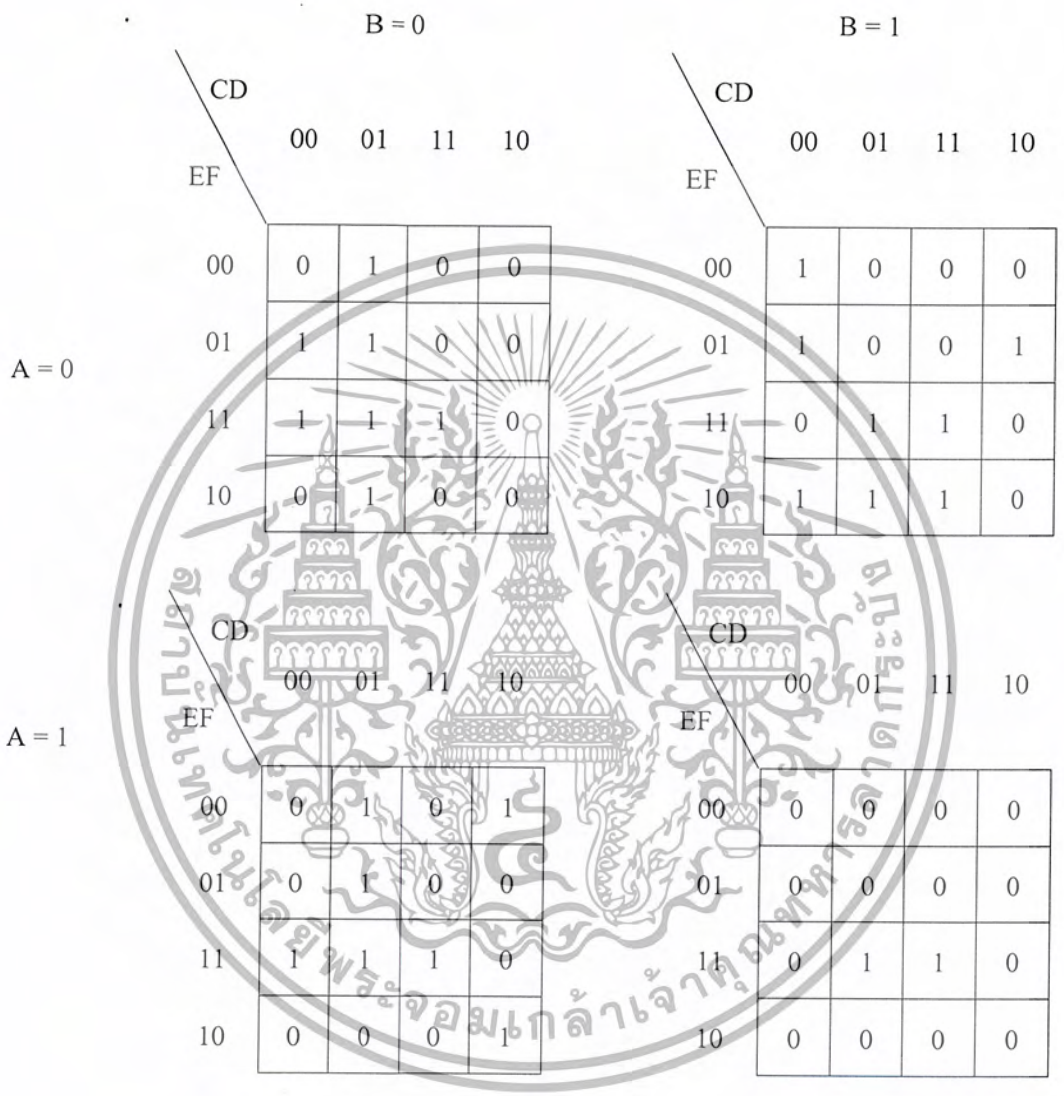


$$= DE + \overline{B}\overline{D}F + A\overline{D}\overline{F} + AB\overline{C}E + AB\overline{C}D\overline{F} + A\overline{B}D\overline{E} + A\overline{B}DF + \overline{A}\overline{B}CD\overline{F}$$
$$+ \overline{A}\overline{B}CE + \overline{A}BCF + \overline{A}BCD\overline{E}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การลดรูปแยกเพลินจิตร

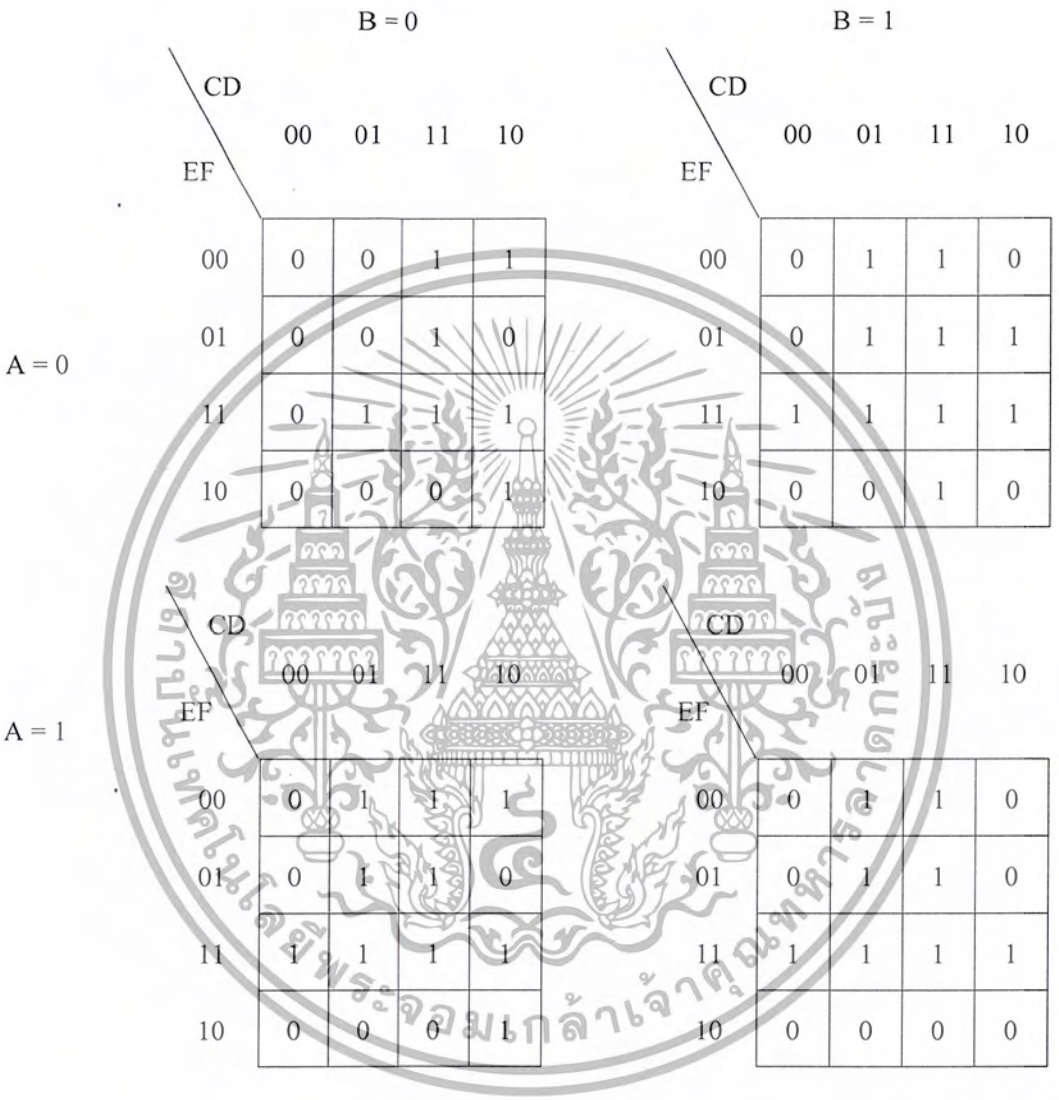
สมการ U แยกเพลินจิตร



$$= DEF + \overline{B}\overline{C}D\overline{E} + \overline{A}\overline{B}\overline{C}F + \overline{A}\overline{B}\overline{C}DE + A\overline{B}C\overline{D}\overline{F} + A\overline{B}\overline{C}EF + \overline{A}BC\overline{D}F + \overline{A}BC\overline{D}E$$
$$+ \overline{A}B\overline{D}\overline{E}F + \overline{A}BDE$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

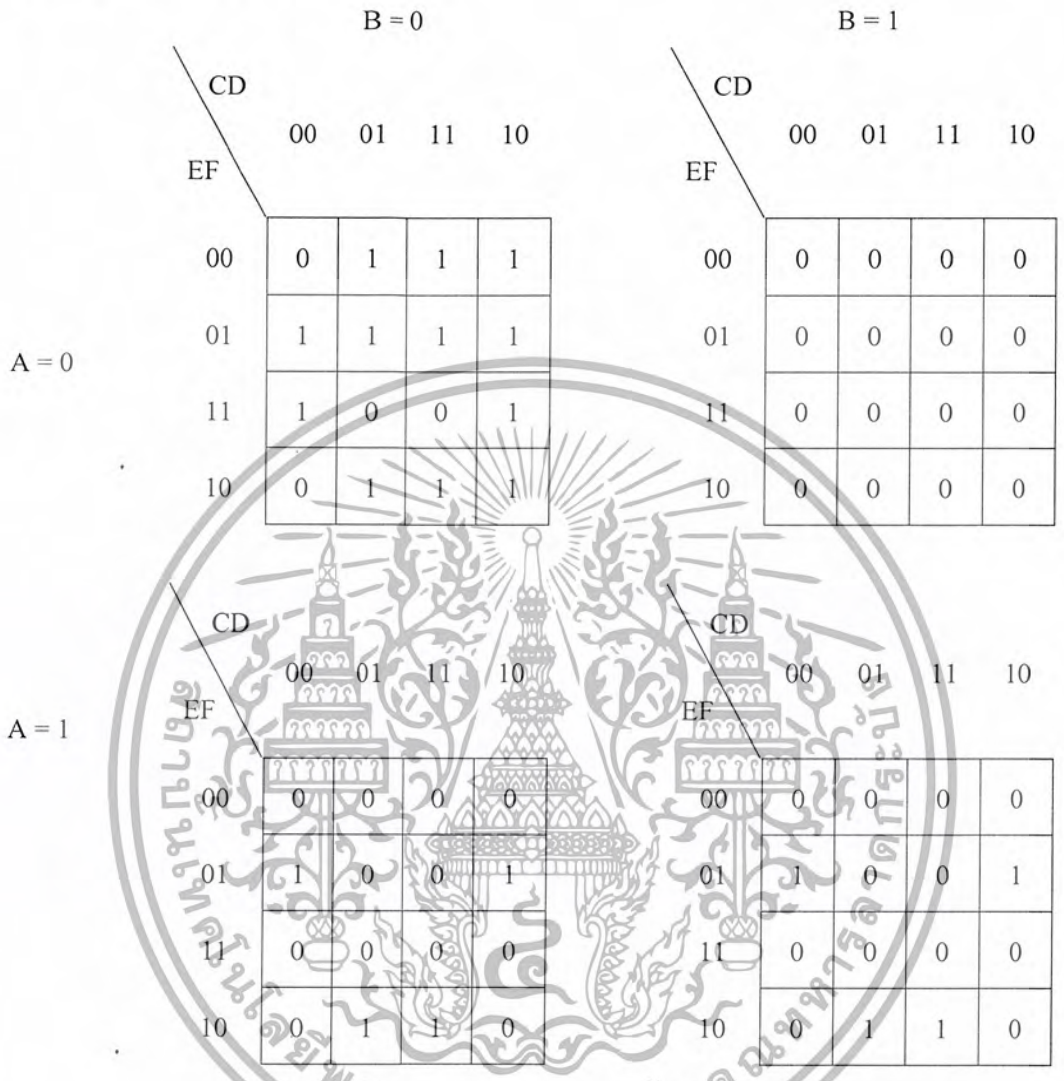
สมการ V แยกเพลินจิตร



$$= AEF + BEF + AD\overline{E} + BD\overline{E} + A\overline{B}C\overline{D}\overline{F} + \overline{A}BCD + \overline{A}BCF + \overline{A}\overline{B}CE\overline{F} + \overline{A}\overline{B}CD\overline{E}$$
$$+ \overline{A}\overline{B}C\overline{D}E + \overline{A}\overline{B}DEF$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการ W แยกเพลินจิตร



$$= ADEF + A\overline{DE}F + \overline{AB}[D\overline{F} + \overline{D}F + C\overline{D} + D\overline{E}]$$

สมการที่ได้นี้คือสมการ เอาท์พุทซึ่งอยู่ในรูปของรหัสเพื่อให้ง่ายแก่การออกแบบ จากนั้นเราจะนำเอาไปสร้างเป็นวงจรต่อไป

3.5.3 ส่วนควบคุมการนับขึ้นนับลง

ในส่วนของการนับขึ้นนับลงนี้จะใช้เอาท์มาทำการควบคุม โดยจะนำเอารหัสของสัญญาณไฟมากำหนดโดยดูว่าสถานะเอาท์พุทในปัจจุบันเป็นลักษณะสัญญาณไฟแบบใด และลักษณะแบบนี้นั้นช่องจราจรใดจะต้องมีปริมาณรถลดลง หรือเพิ่มขึ้นอย่างไร ถ้าช่องจราจรใดเพิ่มเราจะส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะ 1 ออกไปให้กลับส่วนตรวจสอบปริมาณรถ ถ้าช่องจราจรใดลดลงเราจะส่งสถานะ 0 ให้กลับส่วนตรวจสอบปริมารถเช่นกัน แล้วส่วนตราจสอบปริมาณรถก็จะทำงานตามสถานะที่ได้รับต่อไป การกำหนดสถานะเป็นดังตารางข้างล่างนี้

ABC	UD1	UD2	UD3	UD4	UD5	UD6
000	0	1	1	1	1	0
001	0	0	1	0	1	1
010	1	1	0	1	1	1
011	0	1	1	1	1	0
100	0	1	1	1	1	0
101	0	0	1	0	1	1
110	1	1	0	1	0	1
111	1	0	1	0	0	1

ตารางที่ 3.5 กำหนดการนับขึ้น นับลง แยกชิดลม

ABC	UD5	UD6	UD7	UD8	UD9	UD10	UD11
000	0	0	0	1	1	0	1
001	1	1	1	0	1	0	0
010	1	1	1	0	0	1	1
011	1	1	1	0	1	0	0
100	1	1	1	0	0	1	1
101	0	0	0	1	1	0	1
110	1	0	0	0	1	0	1
111	1	0	0	0	1	0	1

ตารางที่ 3.6 กำหนดการนับขึ้น นับลง แยกเพลินจิตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะสังเกตุเห็นว่าในช่องจราจรที่ 5 และช่องจราจรที่ 6 นั้นจะมีความสัมพันธ์กันอยู่คือ จากตารางเราจะเห็นว่าในการทำงานจริงนั้นช่องจราจรที่ 5 และช่องจราจรที่ 6 จะได้รับอินพุทมาจากทั้งสองแยกทำให้เกิดการสับสนได้ ดังต้องมีการกำหนดสถานะใหม่ดังนี้

สถานะแยกชิดลม/ เพลินจิตร(AB)	สัญญาณ UD5	สัญญาณ clock
00	0	1
01	ไม่มีการนับ	0
10	ไม่มีการนับ	0
11	1	1

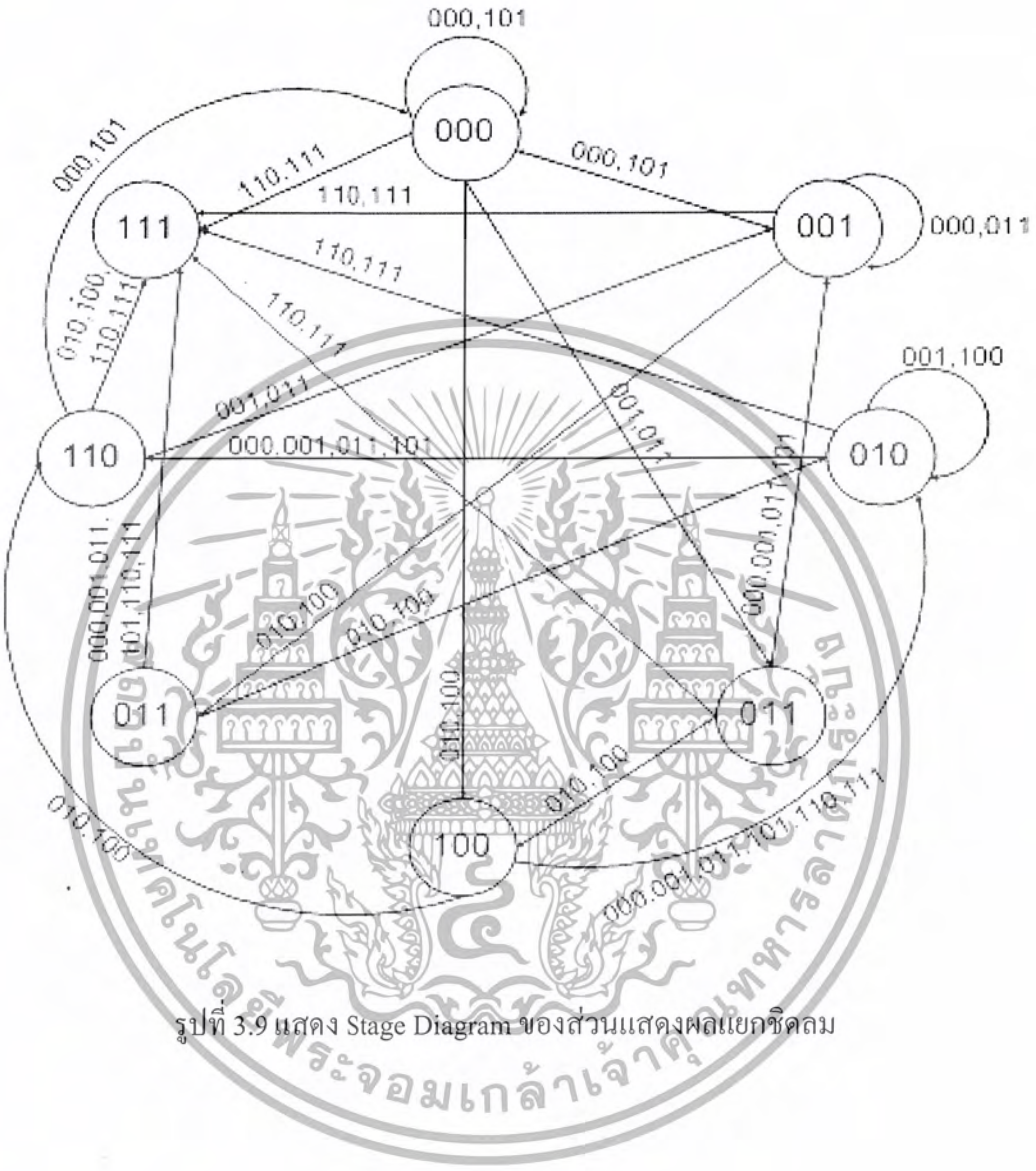
ตารางที่ 3.7 ควบคุมการนับขึ้น นับลง ช่องจราจรที่ 5

สถานะแยกชิดลม/ เพลินจิตร(AB)	สัญญาณ UD6	สัญญาณ clock
00	0	1
01	ไม่มีการนับ	0
10	ไม่มีการนับ	0
11	1	1

ตารางที่ 3.8 ควบคุมการนับขึ้น นับลง ช่องจราจรที่ 6

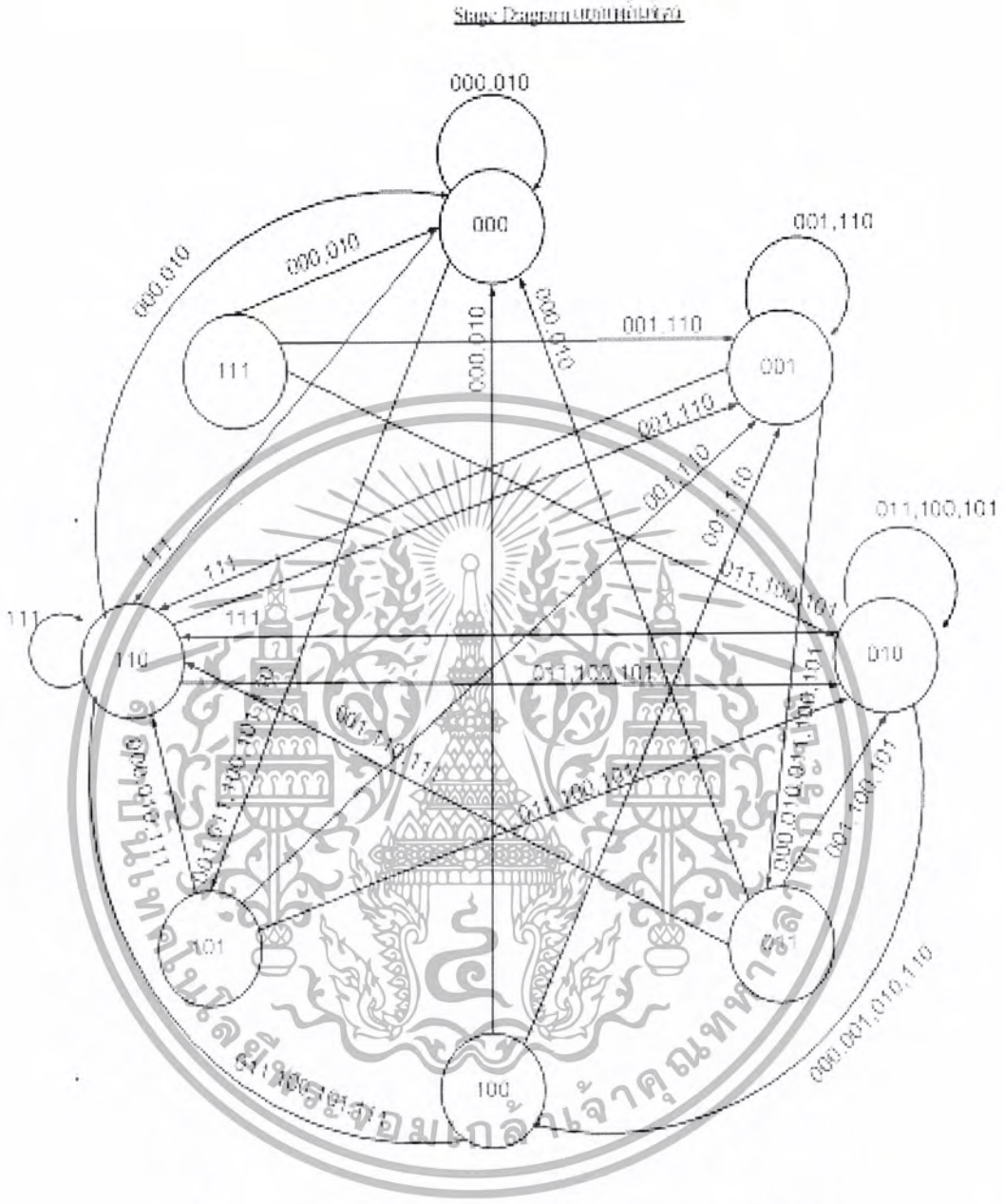
จากตารางที่ 3.8 เราจะสังเกตุเห็นว่าจะมีสถานะที่ไม่มีการนับเกิดขึ้นคือไม่มีการนับขึ้นนับลง เราสามารถสร้างวงจรควบคุมการนับของช่องที่ 5 และ 6 ได้โดย จากตารางในช่อง สัญญาณ clock ที่เป็น 1 เราจะทำการนับปกติโดยอาจนับขึ้นหรือนับลงก็แล้วแต่สัญญาณ UD ส่วนที่เป็น 0 จะไม่มีการนับเกิดขึ้น คือไม่มี clock ออกมานั่นเอง เนื่องจาสัญญาณ clock ที่ใช้ควบคุมการนับที่มาจาก IC 555 ของช่องที่ 5 และช่องที่ 6 นั้นจะต้องผ่านมาที่ส่วนของการควมคุมการนับนี้ก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดง Stage Diagram ของส่วนแสดงผลแยกชิดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



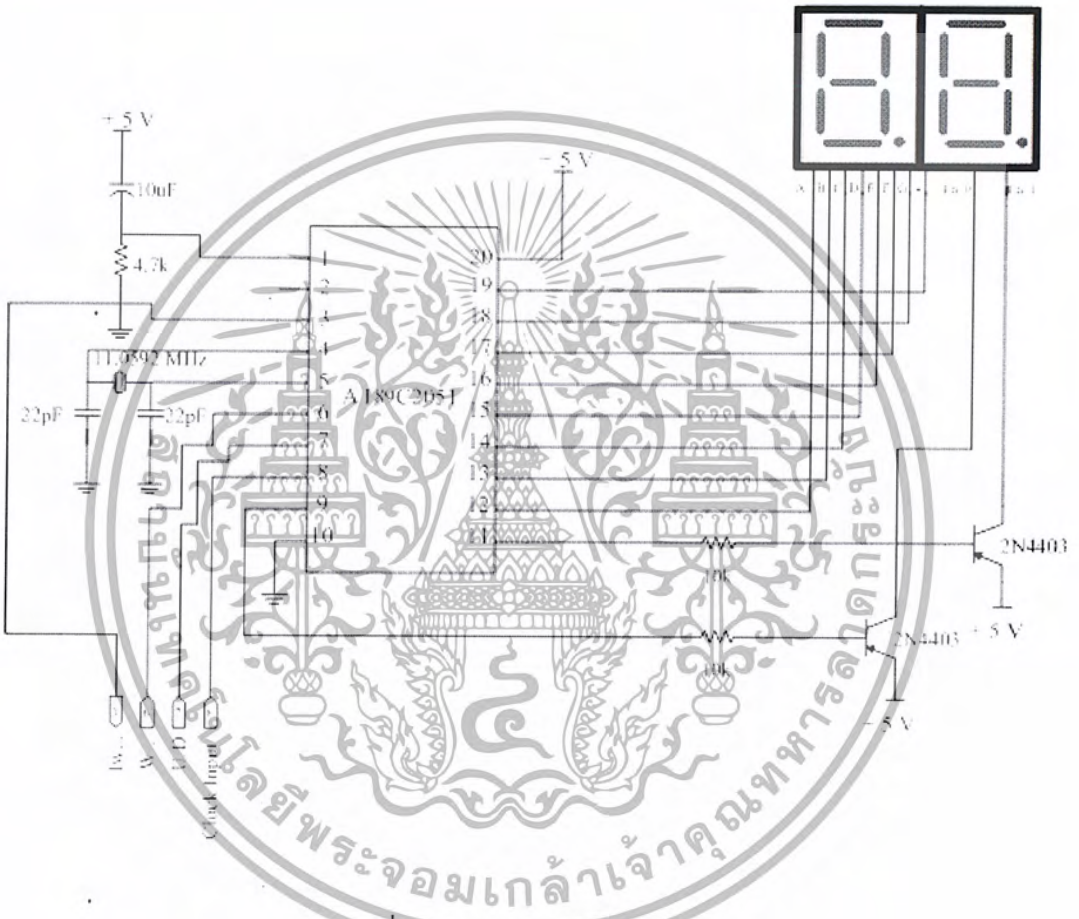
รูปที่ 3.10 แสดง Stage Diagram ของส่วนแสดงผลแยกเพลินจิตร

3.6 ส่วนประกอบอื่นๆของระบบ

นอกจากส่วนประกอบหลักๆที่กล่าวมาข้างต้นแล้วการที่ระบบจะสามารถทำงานได้อย่างสมบูรณ์นั้นจำเป็นจะต้องมีวงจรต่างๆเข้ามามีส่วนร่วมในระบบด้วยเช่น ในการสื่อสารข้อมูลแบบแบ่งเวลานั้นอุปกรณ์ที่เราเลือกใช้ก็คือ ไอซีมัลติเพล็กเบอร์ 74HC251 และในการที่จะทำให้การทำงานของสัญญาณไฟในแยกต่างๆทำงานสัมพันธ์กันนั้นในเส้นทางที่ร่วมกันเมื่อมีการส่งข้อมูลมาที่ส่วนประมวลผลของแต่ละแยกนั้นจำเป็นจะต้องส่งข้อมูลให้พร้อมกันเพื่อมิให้เกิดความผิดพลาด

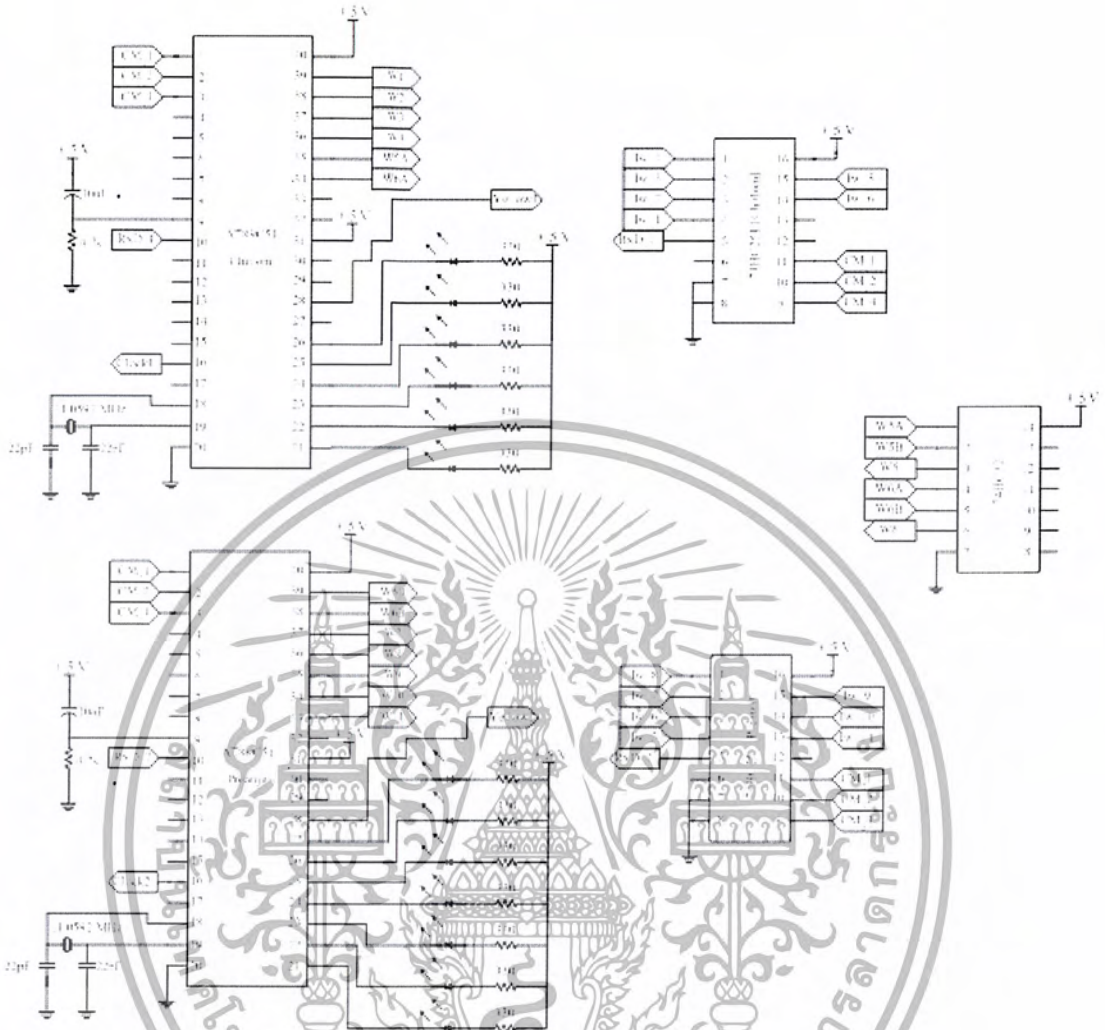
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูลเกิดขึ้น ดังนั้นในระบบจึงจำเป็นต้องมีการออกแบบวงจรมารองรับส่วนนี้ด้วย สำหรับวงจรในส่วนนี้นั้นเราใช้คุณสมบัติของ OR-GATE มาใช้งานเพราะว่าสัญญาณที่ร้องของข้อมูลนั้นเป็นสัญญาณ “0” ซึ่งสามารถแสดงวงจรทั้งหมดได้ดังรูปที่ 3.11- 3.13 โดยที่ทั้ง 3 รูปนั้นมีการทำงานที่สัมพันธ์กัน



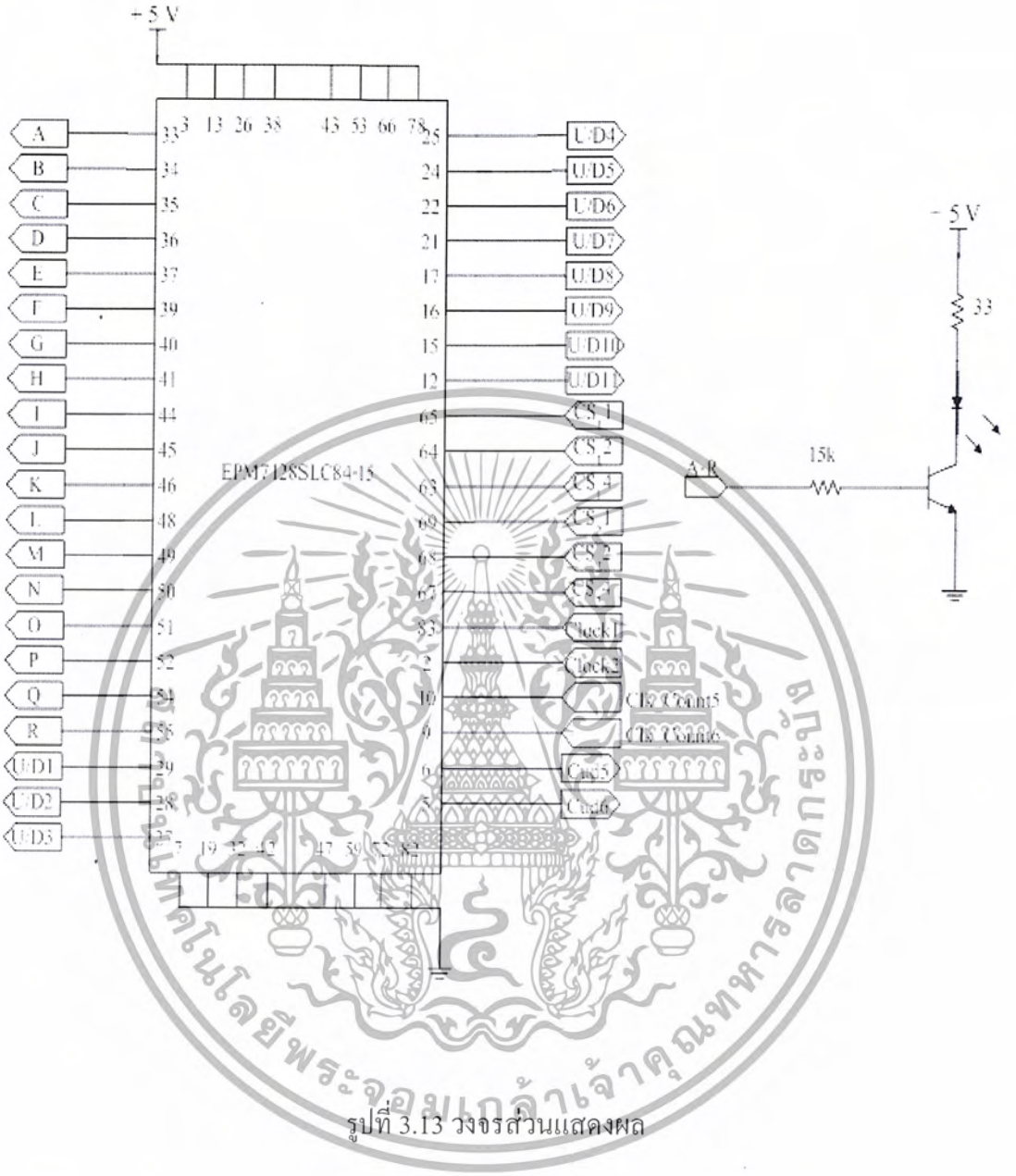
รูปที่ 3.11 วงจรแสดงปริมาณรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 วงจรประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 วงจรส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 วงจรส่วนต่างๆที่อยู่ใน อุปกรณ์ FPGA

เนื่องจากวงจรส่วนใหญ่ในโครงงานนี้ได้ถูกออกมาเป็นระบบดิจิตอลแทบทั้งหมด ซึ่งเป็นการนำเอาเกทต่างๆมาต่อกันเป็นจำนวนมาก ดังนั้นเพื่อความประหยัดทางด้านค่าใช้จ่ายต่างๆและความสะดวกในการพัฒนาวงจรเราจึงได้นำอุปกรณ์FPGAมาใช้งาน ซึ่งในที่นี้เป็นอุปกรณ์เบอร์ EPM7128SLC84-15 ซึ่งสามารถบรรจุเกทได้ขนาดกลางประมาณ 2500 เกท เมื่อทำการต่อวงจรเสร็จตามที่ออกแบบเอาไว้แล้ว จะทำการ SIMULATE ด้วยโปรแกรม MAX-PLUS II ซึ่งได้ผลการทดลองดังนี้

4.2 ผลการทดลองลักษณะสัญญาณไฟแยกชิดลม

ผลการทดลองเป็นไปตามรูปที่ 4.1(ก) ถึง 4.1(ก) ซึ่งอธิบายได้ดังนี้

CLK 1 คือ สัญญาณ Clock ที่ป้อนให้กับ FPGA ทำงาน

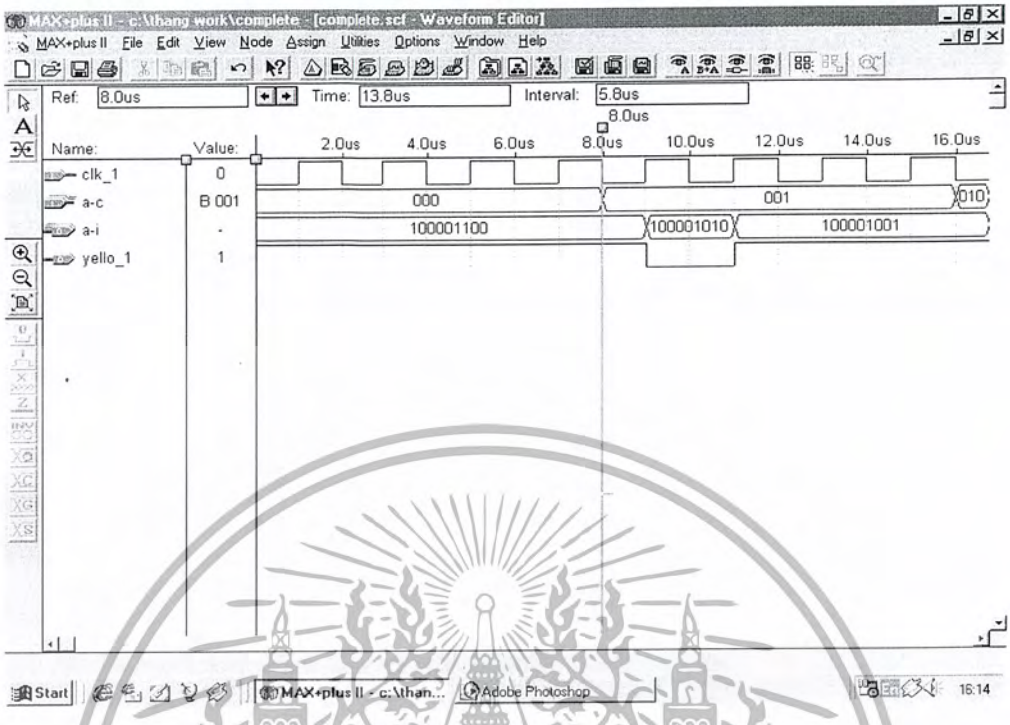
A - C คือ อินพุทที่ป้อนมาจาก ไมโครคอนโทรลเลอร์ซึ่งเป็น code ที่ได้มจากการเช็คปริมาณรถในแต่ละช่องจราจร

A – I คือ เอาพุทหรือลักษณะสัญญาณไฟที่แยกชิดลมตามตารางที่ 3.3

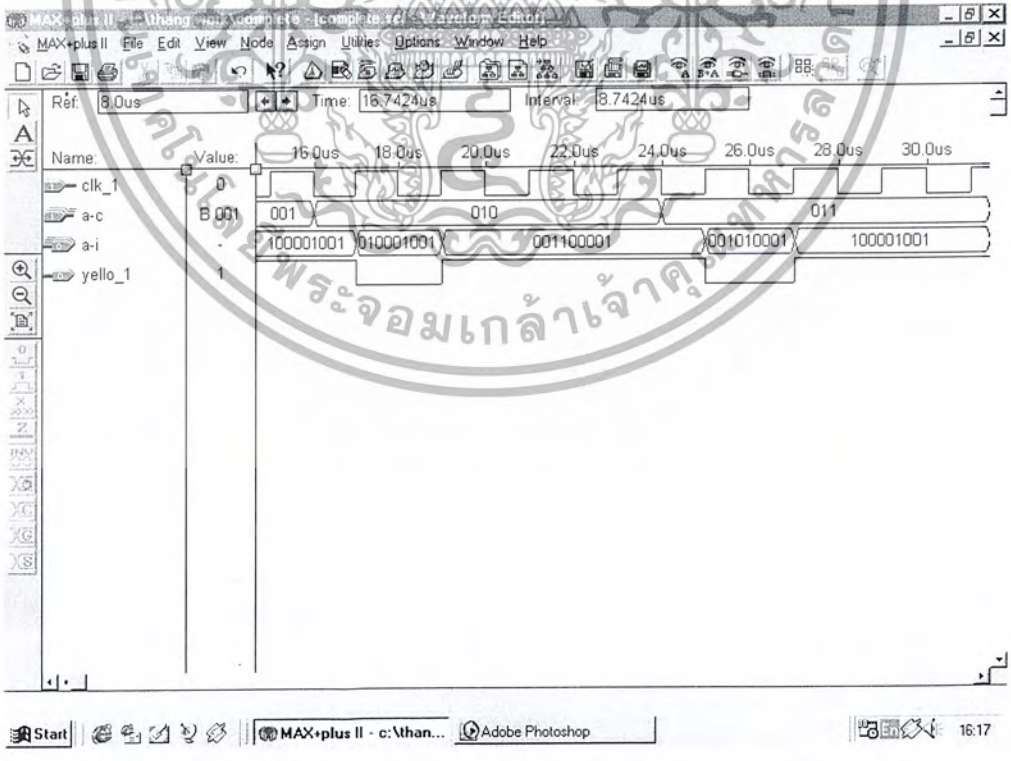
Yello_1 คือ Timming Diagram ของไฟเหลือง

การทำงานสามารถอธิบายได้ดังนี้คือ ในรูป 4.1(ก) เริ่มจาก อินพุทเป็น 000 เอาพุทจะเป็น 100001100 ตามตารางที่ 3.3 หลังจากนั้นจะมีการสแกนเจออินพุทเป็น 001 จากตารางที่ 3.1 เมื่ออินพุทเป็น 000 แล้วสแกนเจอ 001 เอาพุทจะกลายเป็น 011 แล้วลักษณะสัญญาณไฟจะเป็น 100001010 ตามตารางที่ 3.3 ซึ่งจะเห็นว่ามีสัญญาณไฟเหลืองด้วย เพราะฉนั้นสัญญาณ Yello_1 จะมี Logic เป็น 0 จำนวน 1 Clock และกลายเป็น ลักษณะสัญญาณไฟต่อไปตามตารางที่ 3.1 แล้วจะเป็นอย่างนี้ไปเรื่อย ๆ ซึ่งสามารถดูได้ตามรูปที่ 4.1(ก) - 4.1(จ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

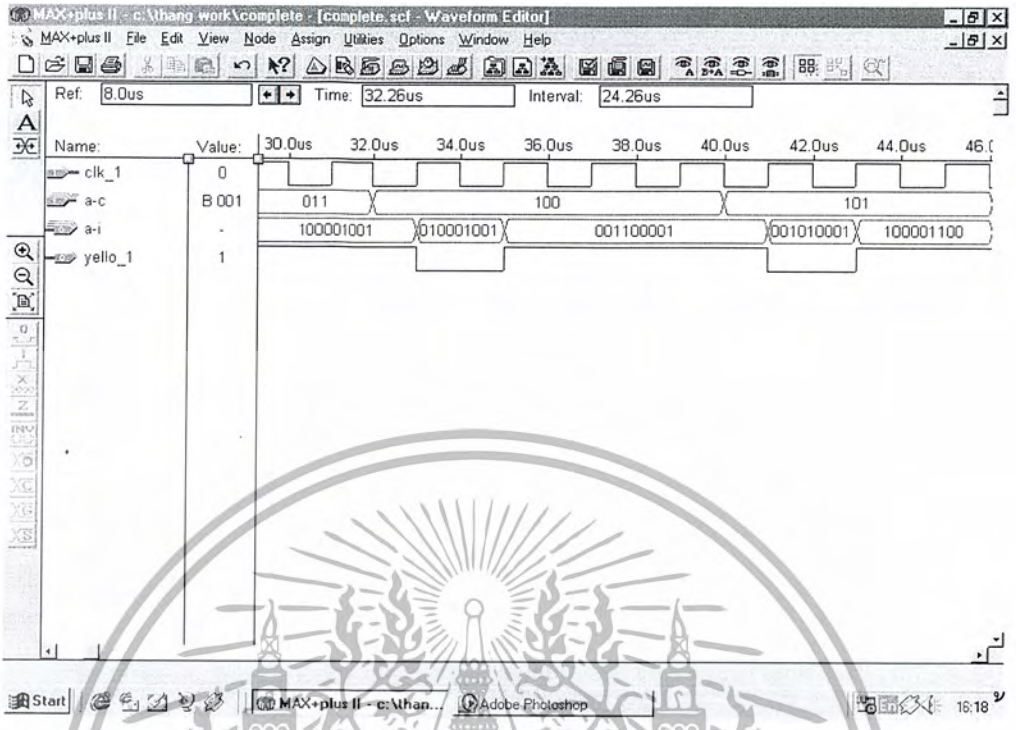


รูปที่ 4.1 (ก)

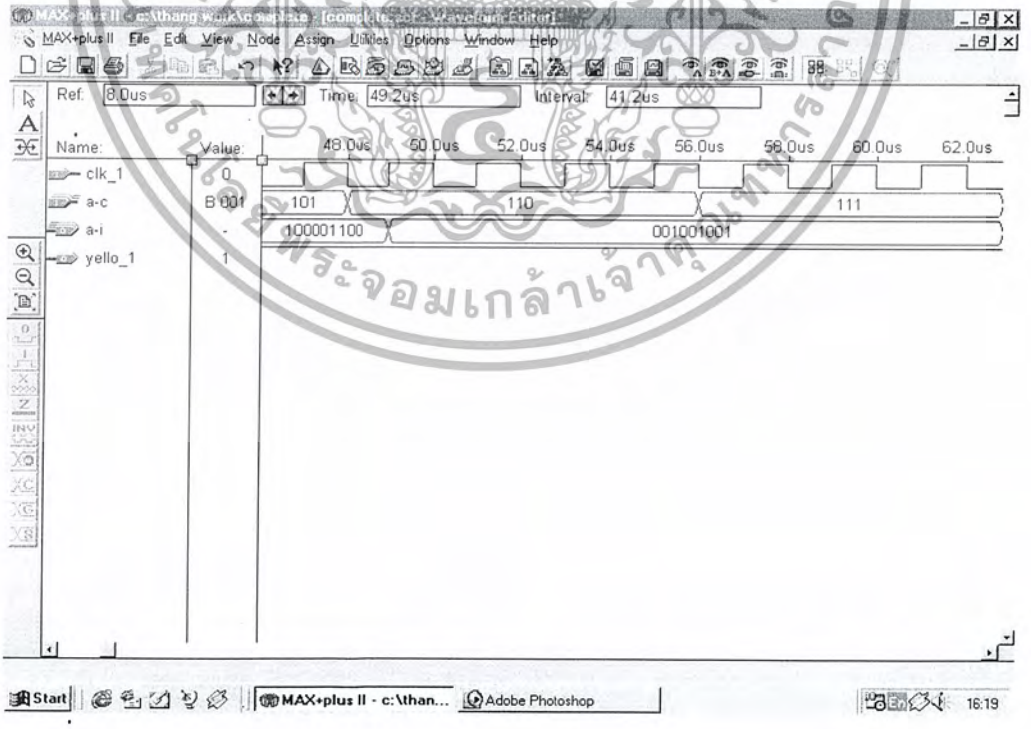


รูปที่ 4.1 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

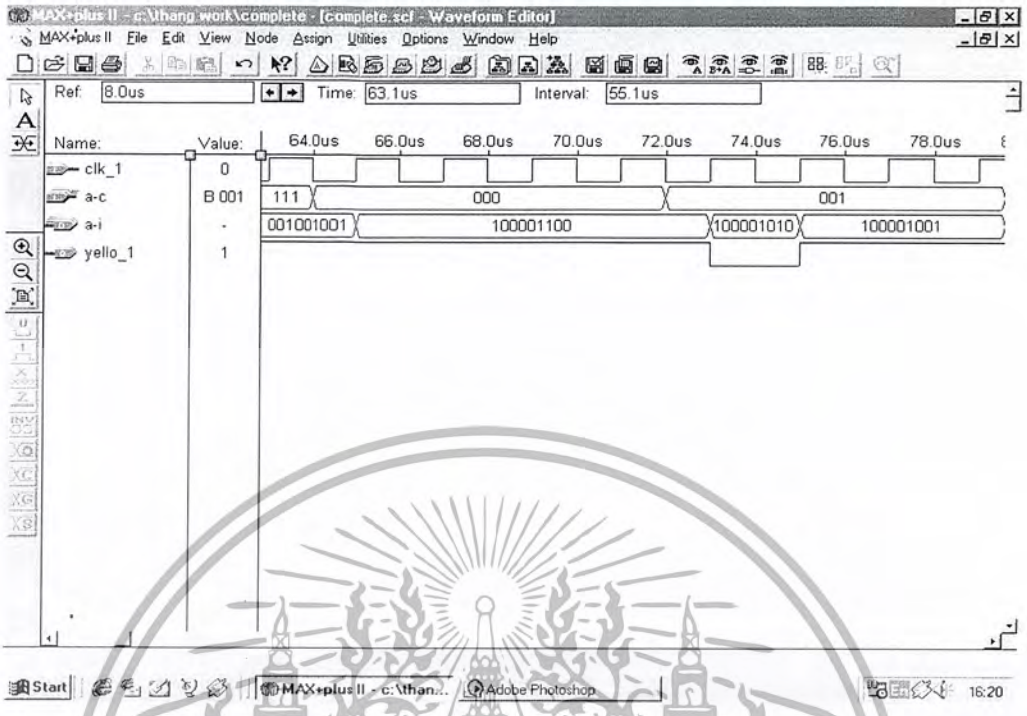


รูปที่ 4.1 (ค)



รูปที่ 4.1 (ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 (จ)

รูปที่ 4.1 ผลการทดลองลักษณะสัญญาณไฟแยกชิดลม

4.3 ผลการทดลองลักษณะสัญญาณไฟแยกเพลินจิตร

ผลการทดลองเป็นไปตามรูปที่ 4.1(ก) ถึง 4.1(ก) ซึ่งอธิบายได้ดังนี้

CLK 2 คือ สัญญาณ Clock ที่ป้อนให้กับ FPGA ทำงานซึ่งในที่นี้เหมือนกับ CLK 1 จึงไม่ได้แสงไว้

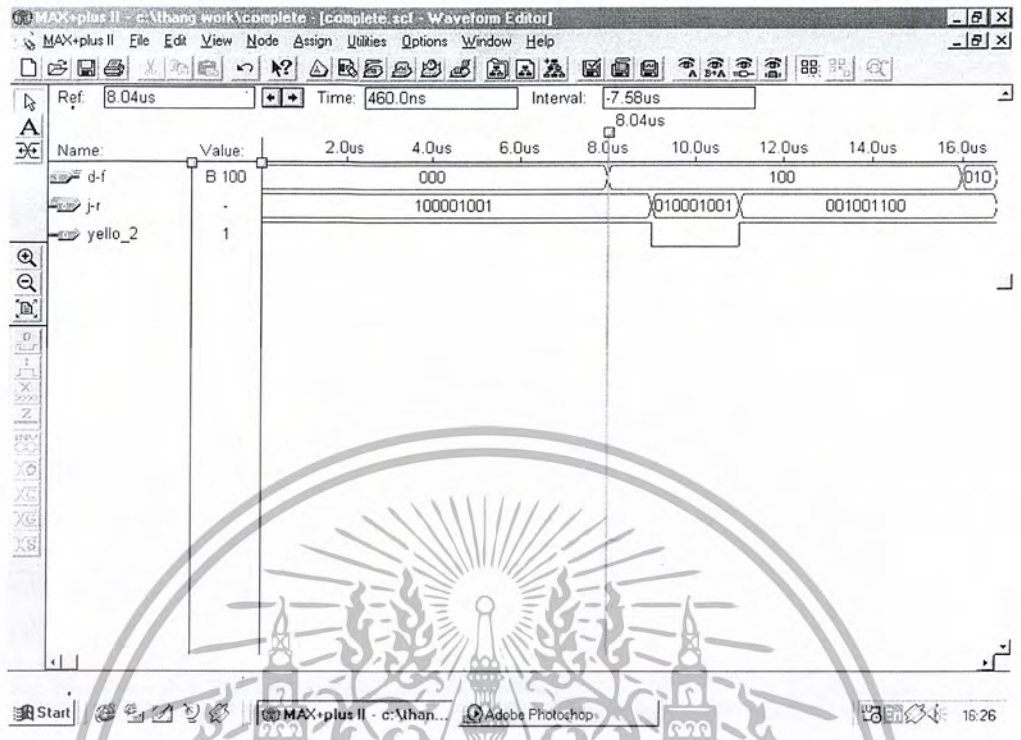
D - F คือ อินพุทที่ป้อนมาจาก ไมโครคอนโทรลเลอร์ซึ่งเป็น code ที่ได้มจากการเช็คปริมาณรถในแต่ละช่องจราจร

J – R คือ เอาพุทหรือลักษณะสัญญาณไฟที่แยกชิดลมตามตารางที่ 3.3

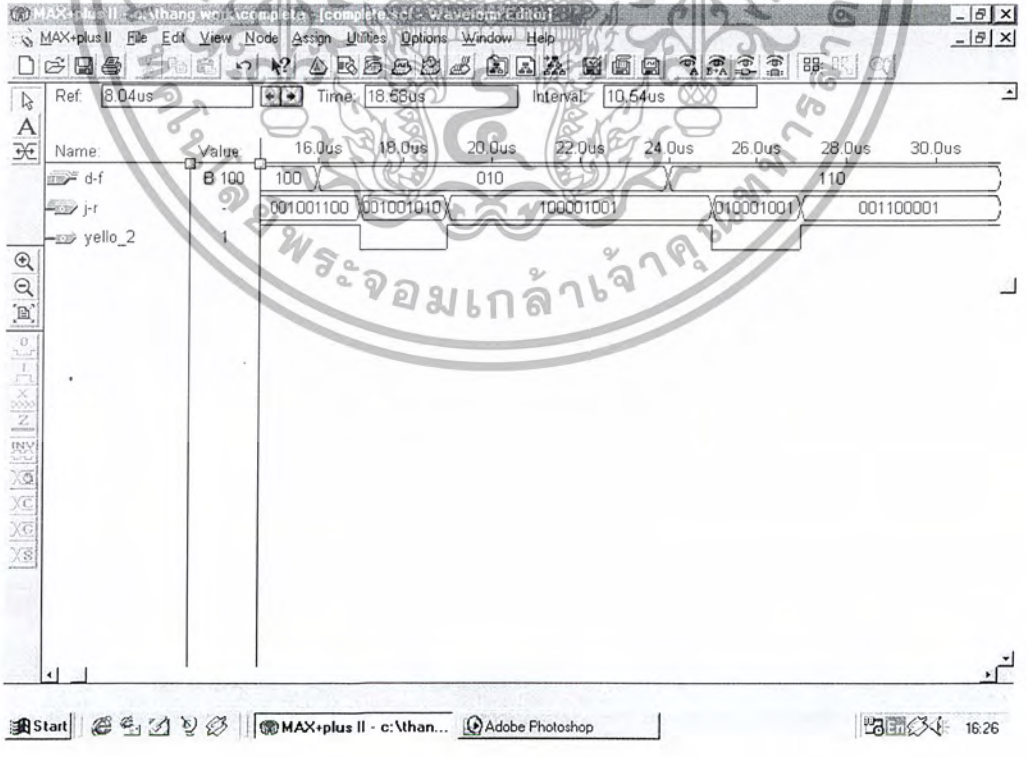
Yello_2 คือ Timing Diagram ของไฟเหลือง

การทำงานสามารถอธิบายได้ดังนี้คือ ในรูป 4.2(ก) เริ่มจาก อินพุทเป็น 000 เอาพุทจะเป็น 100001001 ตามตารางที่ 3.4 หลังจากนั้นจะมีการสแกนเจออินพุทเป็น100 จากตารางที่ 3.2 เมื่ออินพุทเป็น 000 แล้วสแกนเจอ 100 เอาพุทจะกลายเป็น 101 แล้วลักษณะสัญญาณไฟจะเป็น 010001001 ตามตารางที่ 3.4 ซึ่งจะเห็นว่ามีสัญญาณไฟเหลืองด้วย เพราะฉนั้นสัญญาณ Yello_1 จะมี Logic เป็น 0 จำนวน 1 Clock และกลายเป็น ลักษณะสัญญาณไฟต่อไปตามตารางที่ 3.2 แล้วจะเป็นอย่างนี้ไปเรื่อย ๆ ซึ่งสามารถดูได้ตามรูปที่ 4.2(ก) - 4.2(จ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

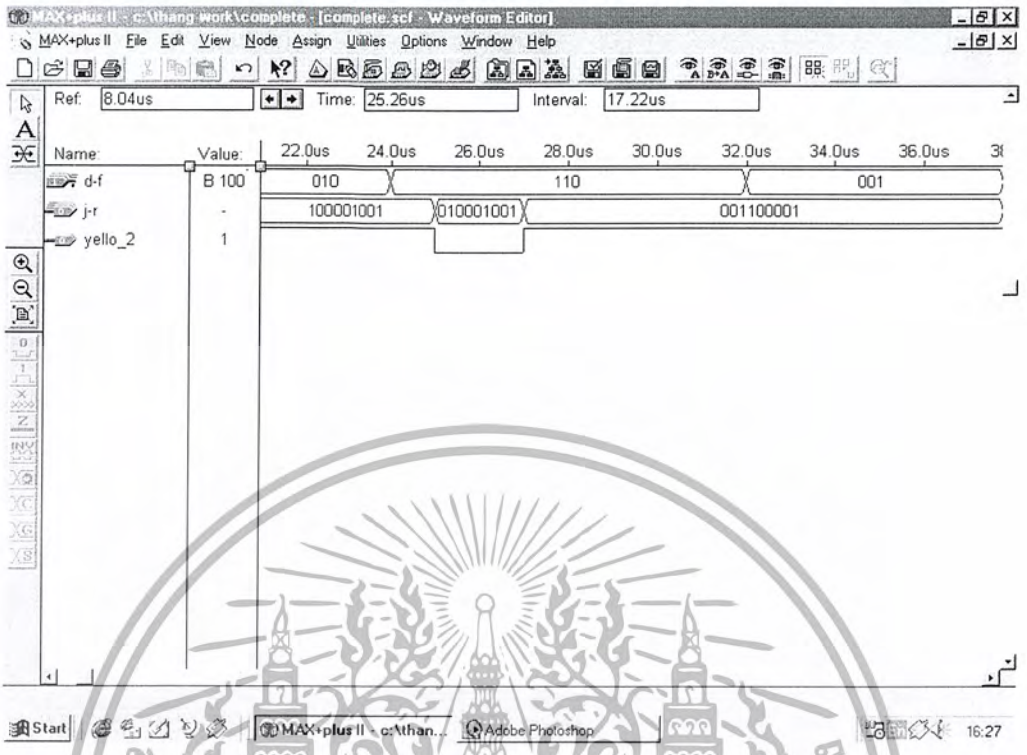


4.2(ก)

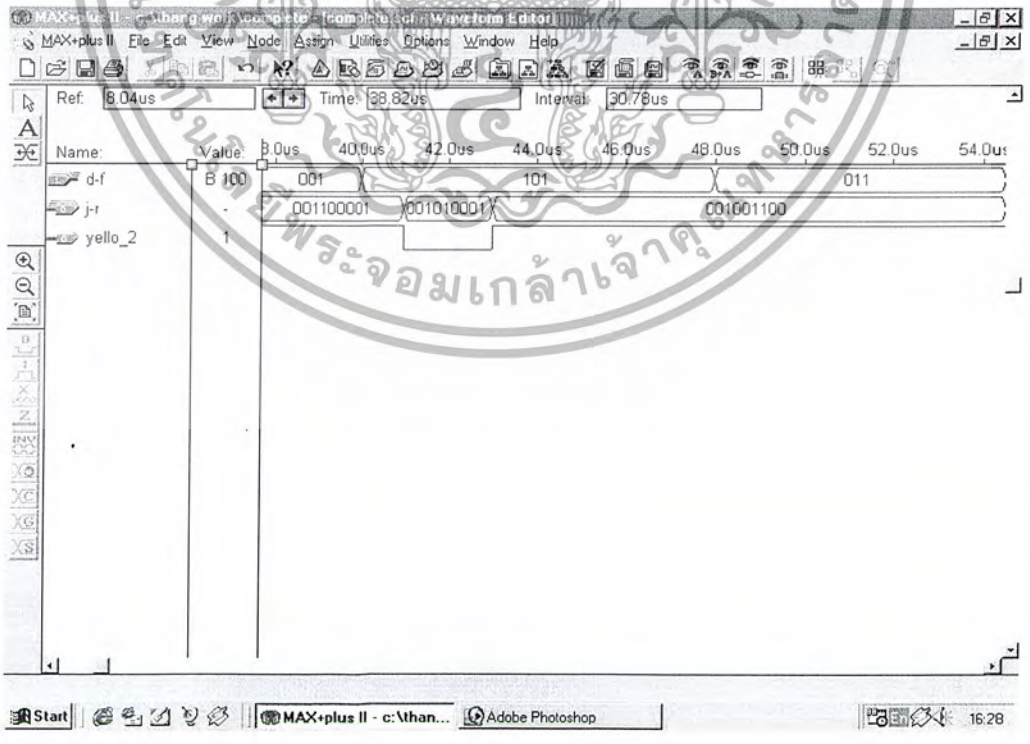


4.2(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

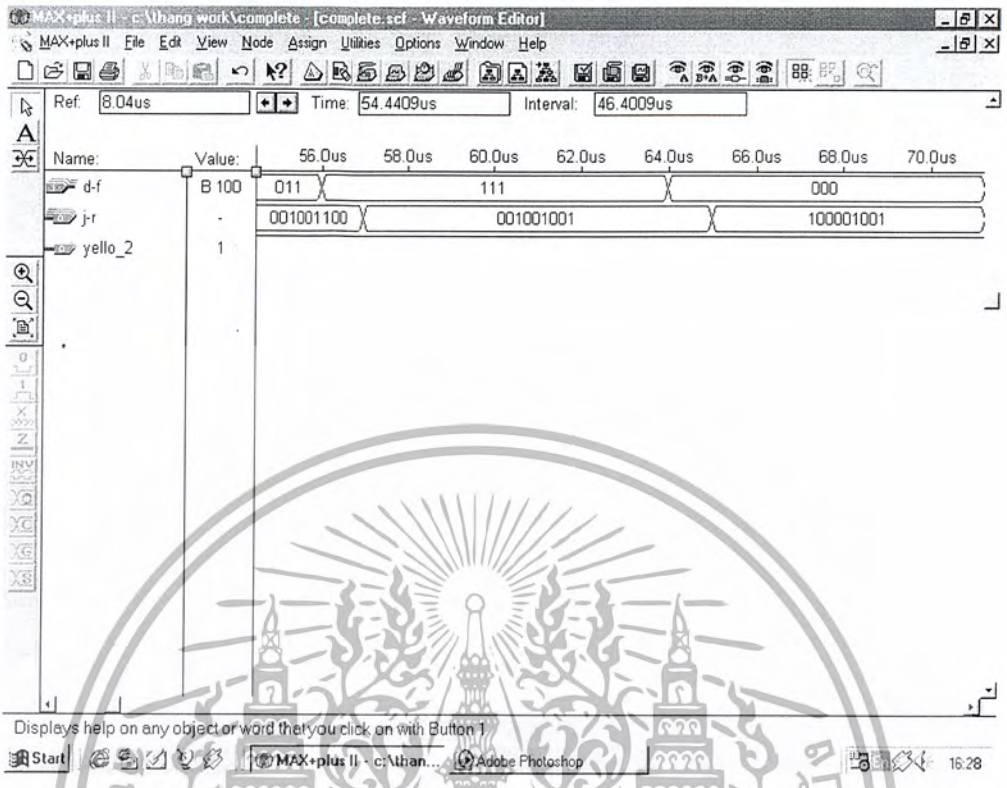


4.2(ก)



4.2(ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



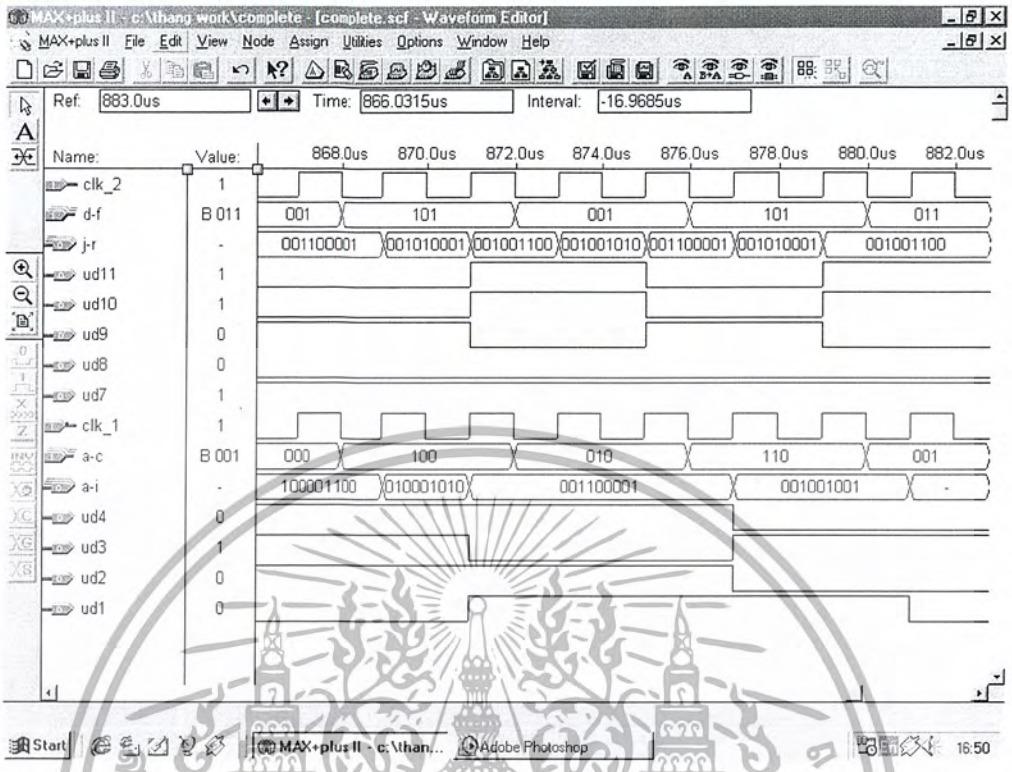
4.2(จ)

รูปที่ 4.2 ผลการทดลองลักษณะสัญญาณไฟจราจรแยกเพลินจิตร

4.4 ผลการทดลองสัญญาณ UP-DOWN

โดยสัญญาณ UP-DOWN ที่เป็น 0 คือมีการนับลง และถ้าเป็น 1 คือ มีการนับขึ้นซึ่งการกำหนดการนับขึ้นนับลงนั้นจะเปลี่ยนแปลงไปโดยการควบคุมโดยสัญญาณอินพุทและสัญญาณเอาพุทที่กำหนดเอาไว้ ตามข้อ 3.5.3 หลังจากการ Simulate แล้วพบว่าเป็นไปตามที่กำหนดเอาไว้โดยมีลักษณะ ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ผลการทดลองสัญญาณ UP - DOWN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและแนวทางการพัฒนา

5.1 สรุป

จากผลการทดลองที่ผ่านมาระบบจะพบได้ว่า สัญญาณไฟจราจรของแต่ละเส้นทางนั้นจะมีลักษณะเช่นใดนั้นจะขึ้นอยู่กับปริมาณรถที่มีอยู่ในแต่ละเส้นทาง โดยถ้าเส้นทางใดมีปริมาณรถมากที่สุดเมื่อมีการตรวจสอบพบสัญญาณจราจรในเส้นทางนั้น ก็จะปล่อยให้รถออกไปได้โดยการตรวจสอบปริมาณรถนั้นจะทำการตรวจสอบทุกๆ 15 วินาที นั้นแสดงให้เห็นว่าสัญญาณจราจรในแต่ละเส้นทางนั้นอาจจะมีการเปลี่ยนแปลงได้ทุกๆ 15 วินาที ทั้งนี้ขึ้นอยู่กับปริมาณรถของแต่ละเส้นทางเองว่ายังมากกว่าเส้นทางอื่นหรือไม่ และถ้าสัญญาณไฟจราจรในทุกเส้นทางของแยกนั้นมีลักษณะเป็นไฟแดงทั้งหมดหรือไม่ยอมให้รถวิ่งผ่านไปได้นั้น แสดงว่าขณะนั้นระบบได้มีการสื่อสารข้อมูลที่ผิดพลาดเกิดขึ้นซึ่งอาจเกิดจากสายสัญญาณ หรือ ส่วนประกอบใดๆชำรุดเมื่อมีการซ่อมแซมแล้วทำการรีเซ็ตระบบใหม่ระบบก็จะสามารถทำงานได้ดังเดิม

5.2 แนวทางการพัฒนา

แนวทางการพัฒนาระบบควบคุมสัญญาณไฟจราจรนั้นเราสามารถพัฒนาให้ระบบมีขนาดใหญ่ขึ้นได้โดยง่าย เนื่องจากการแนวคิดในการในการออกแบบระบบในตอนแรกนั้นเราได้กำหนดให้แต่ละแยกนั้นทำงานโดยเป็นอิสระต่อกันให้ได้มากที่สุดโดยจะอาศัยข้อมูลของเส้นทางที่ร่วมกันเท่านั้นที่เกี่ยวข้องกัน และลักษณะการส่งข้อมูลของแต่ละแยกนั้นจะเป็นแบบTDMซึ่งเหมาะกับระบบขนาดเล็กที่มีจำนวนเส้นทางร่วมกันไม่มาก ดังนั้นในการรับข้อมูลนั้นเพื่อให้มีความถูกต้องและชัดเจนมากที่สุดจึงควรให้มีการส่งข้อมูลของเส้นทางร่วมก่อนและพร้อมกันในแยกที่เกี่ยวข้องกับเส้นทางร่วมนั้นๆ แต่ถ้าระบบมีขนาดใหญ่มากขึ้นการส่งแบบTDMอาจใช่ไม่ได้ดีนักเนื่องจากอาจจะต้องใช้เวลาในการเก็บข้อมูลมาก เนื่องจากต้องรอให้ส่งข้อมูลพร้อมกันในทางร่วม ดังนั้นในการส่งข้อมูลจากเส้นทางอาจเปลี่ยนแปลงได้ โดยส่งข้อมูลทุกแยกพร้อมๆกันโดยอาศัยสัญญาณพาหะของข้อมูลในแต่ละเส้นทางที่ความถี่ที่ต่างกัน ก็จะทำให้เราสามารถส่งข้อมูลได้รวดเร็วมากขึ้น และในส่วนของการแสดงจำนวนรถในแต่ละเส้นทางนั้นได้ออกแบบไว้ให้แสดงปริมาณรถ 1 คัน ต่อ สัญญาณพัลซ์ 1 ลูก ในจะแสดงปริมาณรถที่เพิ่มขึ้นหรือลดลงนั้นขึ้นอยู่กับการแจ้งให้วงจรในส่วนนี้ทราบว่าเป็นรถที่เข้ามาในเส้นทางหรือรถที่ออกจากเส้นทาง ดังนั้นในการใช้งานจริงถ้ามีการสร้างวงจรการตรวจนับรถขึ้นมาสิ่งที่วงจรแสดงปริมาณรถต้องการก็คือ พัลซ์ 1 ลูก ต่อ รถ 1 คันและต้องแจ้งให้ทราบว่าเข้าหรือออกจากเส้นทาง โดยรถที่เข้าเส้นทางแจ้งสัญญาณโลจิก “1” และ รถที่ออกจากเส้นแจ้งสัญญาณโลจิก “0” และเพื่อความถูกต้องและแม่นยำในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลข้อมูล เพื่อให้มีความถูกต้องมากที่สุดนั้นเราจะนับแต่ปริมาณรถอย่างเดียวไม่ได้ เนื่องจากว่าในแต่ละเส้นทางนั้นขนาดของเส้นทางนั้นไม่เท่ากัน ดังนั้นความสามารถในการรองรับรถของแต่ละเส้นทางนั้นย่อมไม่เท่ากัน และขนาดของรถที่เข้ามาในเส้นทางหนึ่งๆย่อมมีขนาดไม่เท่ากันเช่นรถบรรทุก 18 ล้อ 1 คัน ก็จะมีขนาดประมาณรถยนต์ส่วนบุคคล 4 ที่นั่ง จำนวน 2-3 คัน เลยทีเดียว ดังนั้นถ้าเราไม่ได้มีการตรวจสอบว่ารถที่เข้ามาเป็นรถชนิดใดและแต่ละเส้นทางมีขนาดเท่าใดนั้นการประมวลผลข้อมูลก็อาจจะไม่ค่อยมีประสิทธิภาพในการทำงานเท่าที่ควร ดังนั้นในการพัฒนานั้น ถ้าเราสามารถกำหนดอัตราส่วนของแต่ละเส้นทางได้ว่ามีอัตราส่วนต่อเส้นทางที่เป็นมาตรฐานเท่าใด และสามารถแยกชนิดของรถที่เข้าไปในแยกได้ก็จะทำให้ระบบควบคุมสัญญาณไฟจราจรนั้นมีประสิทธิภาพในการทำงานมากขึ้นตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. รศ.พิชัย คูศิริวานิชกร, "เอกสารคำสอนวิชา การออกแบบระบบดิจิตอล"
2. น.อ.ธวัชชัย เลื่อนฉวี , พ.ต.อนุรักษ์ เถื่อนศิริ "ดิจิตอลเทคนิค เล่ม 1" ,2537
3. รศ. สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51",หน้า 119-247 ,2543
4. Astron logic research & development,"เปิดโลก FPGA" ,หน้า 1-56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้