



ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

ชื่อหัวข้อ โปรแกรมลดรูปสมการบูลีน โดยวิธี Quine-Mc-Cluskey
Boolean Expression Minimization Program by Quine-Mc-Cluskey Method

ชื่อนักศึกษา 1. นายฉัตรชัย อุทธา รหัสประจำตัว 42035294
2. นายชิน อรัญกานนท์ รหัสประจำตัว 42035296
3. นายพรอนันต์ ผ่านสำแดง รหัสประจำตัว 42035312

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์
อาจารย์ที่ปรึกษา ผศ.วิสุทธิ อธิพรธรรม
อาจารย์ที่ปรึกษาร่วม อาจารย์กิติพงศ์ มะโน

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. ผศ.วิสุทธิ อธิพรธรรม	
2. อาจารย์กิติพงศ์ มะโน	
3. อาจารย์สุชิน อาจหาญ	
4. อาจารย์ปิยะ จิตธรรมาภิรมย์	
5. อาจารย์ไพบุลย์ พวงวงศ์ตระกูล	

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 25 พฤศจิกายน พ.ศ. 2543 เวลา 10.30 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
ลงนาม.....
(ผศ.วิสุทธิ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 26 เดือน มกราคม พ.ศ. 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์

โปรแกรมลดรูปสมการบูลีนโดยวิธี QUINE-MC-CLUSKEY
BOOLEAN EXPRESSION MINIMIZATION PROGRAM BY
QUINE-MC-CLUSKEY METHOD

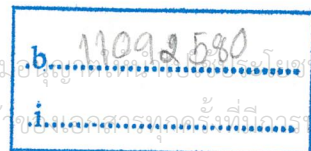


นายฉัตรชัย อุทธา
นายชยีน อรัญยกานนท์
นายพรอนันต์ ผ่านลำแดง

ปริญญาานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน.....40173
วัน, เดือน, ปี.....17 ส.ค. 2544



เอกสารนี้เป็นเอกสารลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่หรือใช้เพื่อการค้า
โดยไม่ได้รับอนุญาต หากมีข้อผิดพลาดหรือให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าภาพเอกสารฉบับนี้ที่มีลิขสิทธิ์ไปใช้

ปริญญานิพนธ์

เรื่อง โปรแกรมลดรูปสมการบูลีนโดยวิธี QUINE-MC-CLUSKEY

Boolean Expression Minimization Program by Quine-Mc-Cluskey Method

วัตถุประสงค์

1. เพื่อศึกษาวิธีการลดรูปสมการบูลีน โดยวิธี Quine-Mc-Cluskey
2. เพื่อออกแบบซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปร
3. เพื่อสร้างซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปร
4. เพื่อทดสอบซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปร
5. เพื่อนำซอฟต์แวร์ที่ได้ไปใช้ในการลดรูปสมการบูลีนหลายตัวแปร

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเข้าใจถึงวิธีการลดรูปสมการบูลีน โดยวิธี Quine-Mc-Cluskey ได้
2. สามารถออกแบบซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปรได้
3. สามารถสร้างซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปรได้
4. ได้ผลการทดสอบซอฟต์แวร์ที่ใช้ในการลดรูปสมการบูลีนหลายตัวแปร
5. ได้ซอฟต์แวร์ที่ใช้ลดรูปสมการบูลีนหลายตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	โปรแกรมลดรูปสมการบูลีนโดยวิธี QUINE-MC-CLUSKEY	
นักศึกษา	นายฉัตรชัย	อุทธา
	นายชิน	อรัญยกานนท์
	นายพรอนันต์	ผ่านสำแดง
อาจารย์ที่ปรึกษา	อาจารย์วิสุทธิ์	อชิพรธรรม
อาจารย์ที่ปรึกษาร่วม	อาจารย์ปิยะ	จิตรธรรมมาภิรมย์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์	
ปีการศึกษา	2543	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอโปรแกรมที่ใช้ในการลดรูปสมการบูลีน โดยใช้วิธีการของ Quine-Mc-Cluskey โดยสามารถลดรูปสมการบูลีนในแบบผลบวกของผลคูณ (sum of product) ได้ตั้งแต่ 2 – 10 ตัวแปร การแสดงผลที่ได้จากการลดรูปสมการจะมีอยู่ 3 แบบ คือ แสดงผลในรูปแบบของสมการบูลีน รูปวงจรถูก และโปรแกรมภาษา VHDL ซึ่งทำให้ประหยัดเวลาและลดความผิดพลาดเนื่องจากขั้นตอนการลดรูปสมการด้วยมือ เป็นผลทำให้วงจรมีขนาดเล็กและลดต้นทุนในการผลิต

Thesis Title	Boolean Expression Minimization Program by Quine-Mc-Cluskey Method	
Students	Mr.Chatchai	Utha
	Mr.Chayin	Arunyakanon
	Mr.Pornanun	Pansumdeang
Advisor	Assist. Prof. Wisuit	Atiporn tum
Co-Advisor	Mr.Piya	Jitthamapirom
Education Level	Bachelor of Science in Industrial Education	
Program in	Electronics and computer	
Academic Year	2000	

ABSTRACT

This thesis presents the Boolean simplifying program which based on the Quine-Mc-Cluskey's method. By this method, we can simplify the Boolean equations in the form of sum of product with 2 – 10 variables. The output, which obtained after program, has 3 forms as to low Boolean equations, gate circuits and VHDL program language. This method help us reduce errors which we've got by manual method, and also have advantages for industrial process the circuit is smaller and cheaper, for instance

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณอาจารย์ปิยะ จิตธรรมมาภิรมย์, ผ.ศ.วิสุทธิ อธิพรธรรม ภาควิชาครุศาสตร์ วิศวกรรม คณะครุศาสตร์อุตสาหกรรม และอาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความอนุเคราะห์อุปกรณ์ รวมทั้งยังให้คำแนะนำ แนวความคิด ความรู้ต่าง ๆ แนวทางการแก้ไขปัญหา ในการทำโปรแกรมและการจัดทำปริญญานิพนธ์ ขอขอบคุณห้องสมุดคณะครุศาสตร์ อุตสาหกรรม หอสมุดกลาง ที่ช่วยอำนวยความสะดวก และเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูล สุดท้ายที่ควรระลึกถึงอย่างยิ่ง คือ บิดา และ มารดา ที่เป็นผู้ให้ความสนับสนุนด้านการศึกษา และ เป็นผู้ให้กำลังใจด้วยดีตลอดมา ตั้งแต่อดีตจนถึงปัจจุบัน



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญาานิพนธ์	1
1.2 ชัดความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 แนวความคิดของการลดทอนฟังก์ชันให้น้อยที่สุด	3
2.2 วิธีหาค่าพหุคูณของเทอมผลคูณที่น้อยที่สุด	8
2.2.1 วิธีการแบบตาราง QM นี้แบ่งออกเป็น 2 ตอนดังนี้	9
1) การหาไพรม์แฟกเตอร์	9
2) ตารางไพรม์แฟกเตอร์	12
2.2.2 การหาแถวที่ต้องเลือก	14
2.3 Delphi	15
2.3.1 ความสามารถของ Delphi 5.0	15
1) การสร้างโปรแกรมประยุกต์ (Application) จาก VCL	16
2) สร้างโปรแกรมประยุกต์ใช้งานร่วมกับอินเทอร์เน็ต	16
2.3.2 ความสามารถด้านการเขียนโปรแกรม	17
2.4 สร้างโปรแกรมประยุกต์ แบบ Event-Driven	18
2.5 ระบบ Message Loop ของ Windows	18
2.6 วัตถุ (Object)	19
2.7 คุณสมบัติ (Property)	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.8 เมธอด (Method)	22
2.9 คอมโพเนนต์ของ Delphi	23
2.10 ส่วนประกอบการทำงานของ Delphi	24
2.10.1 Menu Bar	24
2.10.2 Form Designer	24
2.10.3 Object Inspector	25
2.10.4 Component Palette	25
2.10.5 Code Editor	25
2.11 ขั้นตอนการสร้างโปรแกรมประยุกต์เบื้องต้นกับ Delphi 5.0	26
2.11.1 ขั้นแรก ออกแบบการทำงาน และหน้าต่างของโปรแกรมประยุกต์	26
2.11.2 ขั้นที่สอง วางคอมโพเนนต์ที่ต้องการลงในฟอร์ม	26
2.11.3 ขั้นที่สาม กำหนดคุณสมบัติให้แก่แต่ละคอมโพเนนต์	28
2.11.4 ขั้นที่สี่ เขียนคำสั่งกำกับการทำงานของแต่ละคอมโพเนนต์	28
2.11.5 ขั้นที่ห้า ทำการแปล (Compile)	29
2.11.6 ขั้นที่หก ทดสอบการทำงานของโปรแกรมประยุกต์	30
2.11.7 ขั้นที่เจ็ด บันทึกโปรแกรมประยุกต์ที่สร้างไว้	31
2.12 แนะนำภาษา VHDL (VHSIC Hardware Description Language)	32
2.12.1 ลักษณะการทำงานแบบต่าง ๆ ของ VHDL	32
2.12.2 ข้อดีของ VHDL	33
2.12.3 ลักษณะของภาษา VHDL	33
1) รูปแบบการใช้ port	33
2) ทิศทางการไหลหรือเรียกว่า mode มีอยู่ด้วยกัน 5 mode	33
2.12.4 ชนิดของข้อมูลที่สัญญาณ	34
บทที่ 3 การออกแบบ การสร้างและการพัฒนา	35
3.1 วิธีการเขียนโปรแกรม	35
3.1.1 โครงสร้างของ Cellular (The Cellular Struction)	35

สารบัญ (ต่อ)

เรื่อง	หน้า
3.1.2 THE N-CUBE	36
3.1.3 การแปลงเป็นเลขฐานสิบ (The Decimal Transform)	37
3.2 คุณสมบัติของการบรรจุ (Containment properties)	38
3.3 สรุปความสัมพันธ์ของวิธีการลดรูปสมการ	43
3.4 อัลกอริทึมของการลดรูปสมการบูลีน (Minimization algorithm)	45
3.5 การแสดงโปรแกรมโดยใช้ Flowchart	47
3.6 Flowchart VHDL	49
3.7 แสดงการทำงานโดยโฟลชาร์ตของวงจร	50
บทที่ 4 การทดลองและผลการทดลอง	51
4.1 การทดลองลดรูปสมการขนาด 2 ตัวแปร	51
4.2 การทดลองลดรูปสมการขนาด 3 ตัวแปร	53
4.3 การทดลองลดรูปสมการขนาด 4 ตัวแปร	56
4.4 การทดลองลดรูปสมการขนาด 5 ตัวแปร	58
4.5 การทดลองลดรูปสมการขนาด 6 ตัวแปร	61
4.6 การทดลองลดรูปสมการขนาด 7 ตัวแปร	63
4.7 การทดลองลดรูปสมการขนาด 8 ตัวแปร	66
4.8 การทดลองลดรูปสมการขนาด 9 ตัวแปร	68
4.9 การทดลองลดรูปสมการขนาด 10 ตัวแปร	71
4.10 สรุปผลการทดลอง	73
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	74
5.1 บทสรุป	74
5.2 ปัญหา	74
5.3 แนวทางแก้ไข	74
5.4 แนวทางพัฒนา	74
ภาคผนวก ก โปรแกรมต้นแบบ	75
ภาคผนวก ข ผังงานและโปรแกรม	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บรรณานุกรม	117
ประวัติผู้แต่ง	118



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การลดรูปขั้นที่ 1	10
ตารางที่ 2.2 การลดรูปขั้นที่ 2	10
ตารางที่ 2.3 การลดรูปขั้นที่ 3	11
ตารางที่ 2.4 การลดรูปขั้นที่ 4	11
ตารางที่ 2.5 การหาค่า PI ที่น้อยที่สุด	12
ตารางที่ 2.6 ค่าสมการที่น้อยที่สุด	14
ตารางที่ 2.7 รายละเอียดขุมต่าง ๆ ของโปรแกรมประยุกต์	20



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 n-cube $n=0,1,2,3,4$	5
รูปที่ 2.2 (ก) 3-cube (ข) 4-cube	6
รูปที่ 2.3 การแทนเทอมของฟังก์ชันในรูปเรขาคณิต	6
รูปที่ 2.4 เว็บไซต์ของบริษัท Inprise	15
รูปที่ 2.5 ตัวอย่างโปรแกรมประยุกต์ที่สร้าง VCL	16
รูปที่ 2.6 ตัวอย่างโปรแกรมประยุกต์อินเตอร์เน็ต	17
รูปที่ 2.7 ลดความยุ่งยากในการเขียนโปรแกรมใน Delphi	17
รูปที่ 2.8 โปรแกรมประยุกต์ตัวอย่างที่สร้างจาก Delphi	18
รูปที่ 2.9 แนวคิดของ Message Loop ของ Windows	19
รูปที่ 2.10 องค์ประกอบของโปรแกรมประยุกต์	19
รูปที่ 2.11 ตัวอย่างการกำหนดค่าคุณสมบัติใน Object Inspector	20
รูปที่ 2.12 กำหนดค่าคุณสมบัติโดยระบุค่าอย่างอิสระ	21
รูปที่ 2.13 กำหนดค่าคุณสมบัติโดยเลือกจากตัวเลือก	21
รูปที่ 2.14 กำหนดค่าคุณสมบัติจากไดอะล็อกบ็อกซ์	22
รูปที่ 2.15 เรียกใช้เมธอดของวัตถุในโปรแกรม	23
รูปที่ 2.16 คอมโพเนนต์ต่าง ๆ ใน Component Palette	23
รูปที่ 2.17 องค์ประกอบการทำงานของ Delphi 5.0	24
รูปที่ 2.18 Code Explorer และ Code Editor	25
รูปที่ 2.19 หน้าต่างของโปรแกรมประยุกต์ตัวอย่าง	26
รูปที่ 2.20 เลือกคอมโพเนนต์จาก Component Palette	27
รูปที่ 2.21 การจัดวางคอมโพเนนต์ใน Form Designer	27
รูปที่ 2.22 หลังการกำหนดค่าคุณสมบัติให้คอมโพเนนต์	28
รูปที่ 2.23 การเขียนคำสั่งภายใน	29
รูปที่ 2.24 การแปลงโปรแกรมประยุกต์	30
รูปที่ 2.25 สังกัดสอบโปรแกรมประยุกต์	30
รูปที่ 2.26 บันทึกโปรแกรมที่เขียนขึ้น	31
รูปที่ 2.27 บันทึกโปรเจกต์	32

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.1 รูป Cellular 4-Cube	36
รูปที่ 3.2 ขั้นตอนการลดรูปของโปรแกรม	47
รูปที่ 3.3 ขั้นตอนการทำงานของโปรแกรม VHDL	49
รูปที่ 3.4 ขั้นตอนการวาดรูปวงจรถูก	50
รูปที่ 4.1 หน้าต่าง “กำหนดค่าเริ่มต้น”	51
รูปที่ 4.2 หน้าต่าง “ป้อนสมการลงในตาราง”	51
รูปที่ 4.3 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	52
รูปที่ 4.4 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	52
รูปที่ 4.5 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	53
รูปที่ 4.6 หน้าต่าง “กำหนดค่าเริ่มต้น”	53
รูปที่ 4.7 หน้าต่าง “ป้อนสมการลงในตาราง”	54
รูปที่ 4.8 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	54
รูปที่ 4.9 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	55
รูปที่ 4.10 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	55
รูปที่ 4.11 หน้าต่าง “กำหนดค่าเริ่มต้น”	56
รูปที่ 4.12 หน้าต่าง “ป้อนสมการลงในตาราง”	56
รูปที่ 4.13 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	57
รูปที่ 4.14 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	57
รูปที่ 4.15 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	58
รูปที่ 4.16 หน้าต่าง “กำหนดค่าเริ่มต้น”	68
รูปที่ 4.17 หน้าต่าง “ป้อนสมการลงในตาราง”	59
รูปที่ 4.18 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	59
รูปที่ 4.19 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	60
รูปที่ 4.20 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	60
รูปที่ 4.21 หน้าต่าง “กำหนดค่าเริ่มต้น”	61

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.22 หน้าต่าง “ป้อนสมการลงในตาราง”	61
รูปที่ 4.23 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	62
รูปที่ 4.24 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	62
รูปที่ 4.25 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	63
รูปที่ 4.26 หน้าต่าง “กำหนดค่าเริ่มต้น”	63
รูปที่ 4.27 หน้าต่าง “ป้อนสมการลงในตาราง”	64
รูปที่ 4.28 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	64
รูปที่ 4.29 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	65
รูปที่ 4.30 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	65
รูปที่ 4.31 หน้าต่าง “กำหนดค่าเริ่มต้น”	66
รูปที่ 4.32 หน้าต่าง “ป้อนสมการลงในตาราง”	66
รูปที่ 4.33 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	67
รูปที่ 4.34 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	67
รูปที่ 4.35 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	68
รูปที่ 4.36 หน้าต่าง “กำหนดค่าเริ่มต้น”	68
รูปที่ 4.37 หน้าต่าง “ป้อนสมการลงในตาราง”	67
รูปที่ 4.38 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	69
รูปที่ 4.39 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	70
รูปที่ 4.40 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	70
รูปที่ 4.41 หน้าต่าง “กำหนดค่าเริ่มต้น”	71
รูปที่ 4.42 หน้าต่าง “ป้อนสมการลงในตาราง”	71
รูปที่ 4.43 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”	72
รูปที่ 4.44 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”	72
รูปที่ 4.45 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”	73

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญญาประดิษฐ์

เทคโนโลยีดิจิทัล ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์ และเกือบจะแทนที่เทคโนโลยีแอนะล็อกแทบทั้งหมด ส่วนหนึ่งเป็นเพราะความแม่นยำ และมีเสถียรภาพของระบบดีกว่าแบบแอนะล็อก ซึ่งนั่นก็หมายความว่าผู้ประกอบการทั้งหลายก็ย่อมมีการแข่งขันกันทางการตลาดเพื่อผลิตอุปกรณ์เครื่องใช้ หรือผลิตภัณฑ์ทางด้านเทคโนโลยีต่างๆ เพื่อตอบสนองความต้องการของผู้ใช้งาน ที่ให้การยอมรับเทคโนโลยีดิจิทัลเป็นอย่างมาก

การพัฒนาเทคโนโลยีทางด้านดิจิทัลของสถานประกอบการนี้ จึงจำเป็นจะต้องอาศัยเครื่องมือ และเทคนิควิธีการที่จะทำให้ผลิตภัณฑ์ออกมามีคุณภาพ มีขั้นตอนการออกแบบที่รวดเร็ว และใช้ต้นทุนให้น้อยที่สุด

การลดรูปสมการดิจิทัลก็เป็นอีกวิธีหนึ่งที่จะทำให้วงจรที่ออกแบบมีการทำงานเหมือนเดิมกับที่ออกแบบไว้ แต่มีจำนวนอุปกรณ์หรือจำนวนเกตลดน้อยลง ซึ่งหมายถึงการลดต้นทุนในการผลิตนั่นเอง

ในสมัยก่อนการลดรูปสมการต้องใช้การลดรูปด้วยมือโดยอาศัยกฎความสัมพันธ์ทางดิจิทัล ซึ่งเสียทั้งเวลาในการจัดการ ทั้งในการดำเนินการผลิต และเกิดการผิดพลาดในขั้นตอนการออกแบบได้ง่าย

ปัจจุบันคอมพิวเตอร์เป็นเครื่องมือขั้นเยี่ยมในการคำนวณ และประมวลผลข้อมูล การลดรูปสมการดิจิทัลสามารถประยุกต์ขั้นตอนการทำงานให้อยู่ในรูปของโปรแกรม ซึ่งจะช่วยให้ประหยัดเวลาในการออกแบบเป็นอย่างมาก และลดข้อผิดพลาดจากการลดรูปสมการดิจิทัลด้วยมือ ทั้งยังสามารถมองเห็นผลลัพธ์ในรูปสมการ และวงจรต้นแบบได้อีกด้วย

1.2 ขีดความสามารถของโครงการ

โครงการนี้มีความสามารถดังนี้ คือ

1. สามารถลดรูปตัวแปรได้ตั้งแต่ 2-10 ตัวแปร โดยอยู่ในรูปของ Sum of product
2. สมการที่ลดรูปแล้วในข้อ 1 สามารถแปลงเป็นวงจรเกตได้
3. สมการที่ลดรูปแล้วในข้อ 1 สามารถแปลงเป็นแฟ้ม VHDL ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญยานิพนธ์ฉบับนี้ได้มีการแบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาสำคัญดังนี้

บทที่ 1 ได้กล่าวถึงที่มาของโครงการนี้ ชี้ความสามารถ และเนื้อหาพอสังเขป

บทที่ 2 ทฤษฎีในการลดรูปสมการบูลีน โดยใช้วิธี Quine-Mc-Cluskey (QM) ให้ได้สมการที่ลดรูปแล้วเหลือน้อยที่สุด และทฤษฎีการใช้งานของโปรแกรม Delphi เพื่อที่จะนำมาสร้างโปรแกรม

บทที่ 3 หลักการ โปรแกรมลดรูปสมการบูลีนของ Quine-Mc-Cluskey สรุปเนื้อหาความสัมพันธ์ของทฤษฎี และโครงสร้างของโปรแกรมลดรูปสมการ

บทที่ 4 ผลการทดลอง โปรแกรมลดรูปโดยใช้โปรแกรม Delphi ในการประมวลผล

บทที่ 5 สรุปปัญหาในการทำงาน แนวทางการแก้ไข และการพัฒนาต่อไปในอนาคต

และในภาคผนวก จะแสดงรายละเอียดของโปรแกรมที่เขียนขึ้นมาในส่วนต่างๆ และผังการทำงานในส่วนต่างๆ ของโปรแกรมลดรูปสมการ

บทที่ 2

ทฤษฎีและหลักการ

การออกแบบวงจรสวิตซ์ก็เช่นเดียวกับการออกแบบทางวิศวกรรมทั่วไป คือ ต้องนำเอาหลักเศรษฐศาสตร์มาเกี่ยวข้องด้วย โดยคำนึงถึงด้านราคาต้นทุน และการนำไปใช้งานวงจรที่ราคาประหยัดที่สุดอาจนำไปใช้งานได้ไม่ดี เพราะต้องเสียคุณสมบัติบางประการไป ดังนั้น การออกแบบจึงต้องพิจารณาให้เหมาะสมเพื่อให้ได้วงจรที่ราคาถูกและใช้งานได้ดี แต่ในขั้นการออกแบบเบื้องต้น เราจะพยายามลดทอนฟังก์ชันที่จะสร้างเป็นวงจร ให้สามารถสร้างด้วยเกตจำนวนน้อยที่สุด และมีอินพุทของเกตน้อยที่สุด หลังจากนั้นจึงมาพิจารณาในแง่การใช้งาน ในบทนี้จะแนะนำการลดทอนฟังก์ชันโดยใช้วิธีการทาง QM (Quine-Mc-Cluskey Method)

2.1 แนวความคิดของการลดทอนฟังก์ชันให้น้อยที่สุด

การลดทอนฟังก์ชันสวิตซ์ เป็นวิธีที่เราพยายามลดจำนวนเทอม (อาจเป็นเทอมผลบวกหรือ เทอมผลคูณ) ลดจำนวนตัวอักษร (ทั้งรูปปกติและรูปคอมพลิเมนต์) หรือลดผลรวมของจำนวนเทอม และ ตัวอักษรของนิพจน์ทางพีชคณิตให้น้อยลง จนไม่สามารถลดทอนได้อีก และถ้าหากยังลดตัวอักษร หรือ เทอมต่อไปอีกก็จะทำให้ค่าฟังก์ชันผิดไป การลดตัวอักษรและเทอมจะเป็นการลดแฟนนิน และ จำนวนเกตของวงจรมันเอง ทำให้ราคาต้นทุนของวงจรต่ำลงลงพิจารณาการลดทอน

$$F(X,Y,Z) = (1,2,3,4,5,6)$$

$$F(X,Y,Z) = X'Y'Z + X'YZ' + X'YZ + XY'Z' + XY'Z + XYZ'$$

$$F(X,Y,Z) = X'Z + X'Y + XZ' + XY' \quad (2.1)$$

นิพจน์ของฟังก์ชันที่ถูกลดทอนมาข้างบน เราจะเรียกว่าเป็นนิพจน์ที่ไม่สามารถลดทอนได้อีก (Irredundant Irreducible Expression) เพราะถ้าเราตัดตัวอักษร หรือ ตัดเทอมก็ทำให้ค่าฟังก์ชัน

เปลี่ยนไป นิพจน์ที่ไม่สามารถลดทอนได้อีกอาจมีได้หลายรูป เช่น เมื่อลดทอนโดยรวมเทอมที่ 1 กับ 2,3 กับ 4,5 กับ 6 ของฟังก์ชันได้

$$F(X,Y,Z) = X'Z + YZ + XY' \quad (2.2)$$

และเมื่อลดทอนโดยรวมเทอมที่ 2 กับ 3,4 กับ 5,6 กับ 1 จะได้

$$F(X,Y,Z) = X'Y + XZ' + Y'Z$$

นิพจน์ของฟังก์ชันที่ได้ใหม่ทั้งสอง ต่างเป็นนิพจน์ที่ไม่สามารถลดทอนได้อีกในจำนวนนิพจน์ของฟังก์ชันที่ลดทอนไม่ได้ อีกทั้งสามสองนิพจน์หลังจะเป็นนิพจน์ที่น้อยที่สุด

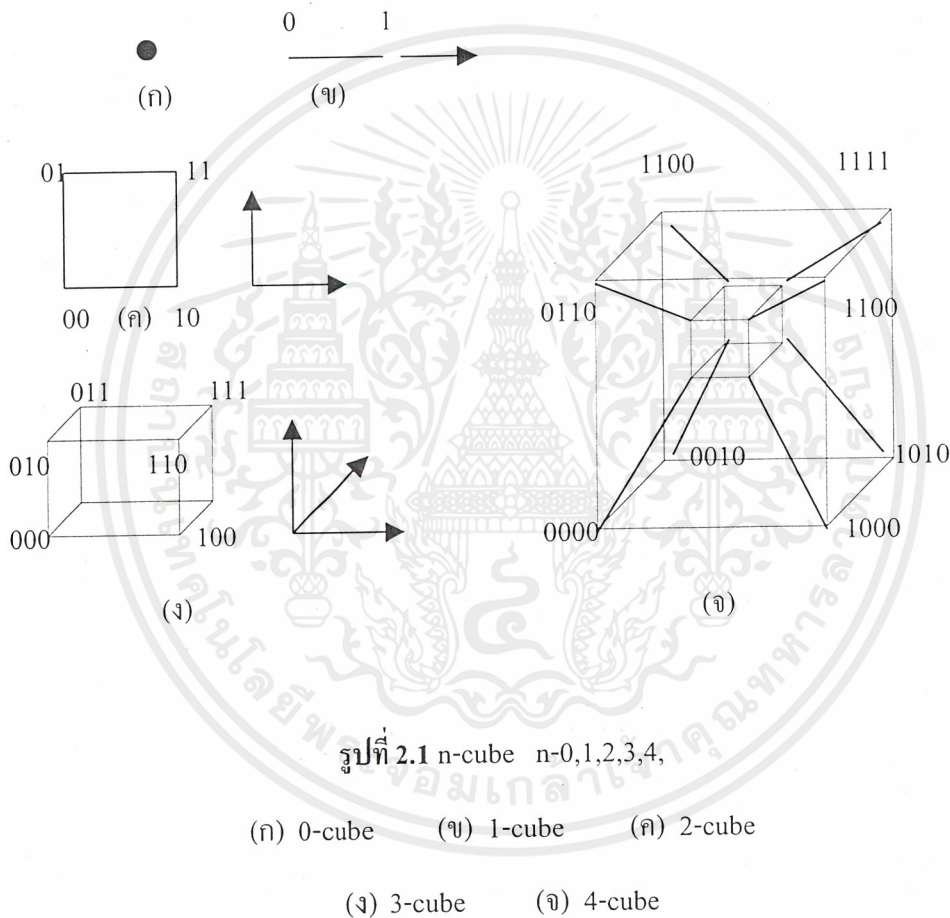
การออกแบบวงจรสวิตซ์ที่ประหยัด หรือราคาต้นทุนต่ำจะต้องพยายามลดทอนฟังก์ชันให้นิพจน์ทางพีชคณิตน้อยที่สุด เพื่อจำนวนแฟนอินและจำนวนเกต ของวงจรจะได้น้อยที่สุดจากนิพจน์ของฟังก์ชันข้างต้นทั้งสามเราสร้างเป็นวงจร 2 ระดับ AND-OR นิพจน์แรกใช้ AND เกท 4 ตัว และ OR เกท 4 แฟนอิน 1 ตัว ส่วนสองนิพจน์หลังใช้ AND เกท 3 ตัว และ OR เกท 3 แฟนอิน 1 ตัว แนวความคิดการลดทอนฟังก์ชันให้น้อยที่สุดที่กล่าวมานี้เป็นฟังก์ชันแบบผลบวกของเทอมผลคูณ สำหรับฟังก์ชันแบบผลคูณของเทอมผลบวกก็มีแนวความคิดทำนองเดียวกัน และเมื่อสร้างเป็นวงจรลอจิกจะเป็นวงจร 2 ระดับ OR-AND เกท

การลดทอนฟังก์ชันสวิตซ์โดยวิธีแผนผังคาโนห์และวิธีตาราง QM จะมีแนวความคิดมาจากรูปทรงทางเรขาคณิตของการแทนเลขไบนารี เลขไบนารี N บิตมีค่า 0 ถึง $2^N - 1$ แสดงด้วยจุดต่าง ๆ 2^N จุดใน N -Space ถ้าตัวแปรสวิตซ์ 1 ตัว จะแทนด้วยจุด 2 จุดใน 1-Space จุดหนึ่งแทน 0 และอีกจุดจะแทน 1 มีเส้นเชื่อมจุดทั้งสองเหมือนแกนเดียวเรียก 1-Cube ตัวแปร สวิตซ์ 2 ตัวจะแทนด้วยจุด 4 จุด ใน 2-Space ตามการจัดหมู่ของตัวแปร ($2^2 = 4$) แต่ละจุดแทน 00,01,10,11 เขียนได้เหมือน 2 แกนที่ตั้งฉากกัน (รูปที่ 2.1 ข.) เป็นรูปสี่เหลี่ยมแรก 2-Cube สำหรับตัวแปร สวิตซ์ 3 ตัวจะแทนด้วยจุด 8 จุด ใน 3-Space แต่ละจุดแทนเลขไบนารี 0 ถึง 7 เขียนได้เหมือนแกน 3 แกน ที่ตั้งฉากกัน (รูปที่ 2.1 ค.) เป็นลูกบาศก์เรียก 4-Cube เมื่อตัวแปรสวิตซ์มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

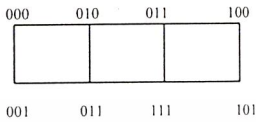
ขึ้น การแสดงรูปเรขาคณิตยิ่งยากขึ้น เพราะต้องใช้แกนมากขึ้นแต่วิธีการ จะเหมือนแบบที่อธิบาย มาแล้ว

ข้อสังเกตจากรูปที่ 2.1 จะเห็นว่าเส้นตรงที่ใช้เชื่อมจุดสองจะมีบิตที่ต่างกันเพียงบิตเดียว เท่านั้น และภายใน N-Cube หนึ่งจะประกอบด้วย Sub-cube หลายแบบเช่น 3-Cube จะประกอบด้วย 2-Sub-cube เป็นสี่เหลี่ยม 6 รูป หรือ 1-Sub-cube เป็นเส้นตรง 12 เส้น

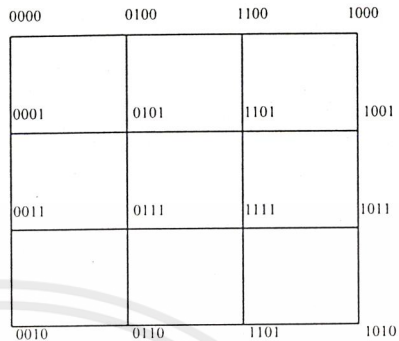


4-Cube จะประกอบด้วย 3-Sub-cube เป็นลูกบาศก์ 8 รูป หรือ 2-Sub-cube เป็นสี่เหลี่ยม 24 รูป หรือ 1-Sub-cube เป็นเส้นตรง 32 เส้น รูปทรงเรขาคณิตของการแทนเลขไบนารี หรือ ตัวแปร สวิตชิง อาจแสดงให้อยู่ในระนาบเดียวกันได้ อย่างเช่นกรณีตัวแปรสวิตชิง 3 ตัวลบเส้น เชื่อมจุดออกก่อน แล้วจึงค่อยมา เชื่อมภายหลังด้วยเส้นปะตามรูปที่ 2.1 (ก) ถ้าตัวแปรสวิตชิง 4 ตัว ก็ทำในทำนองเดียวกันตามรูปที่ 2.2 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



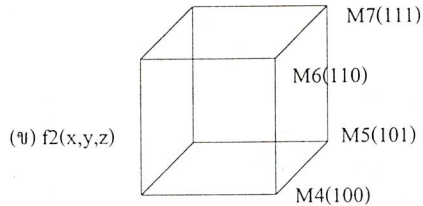
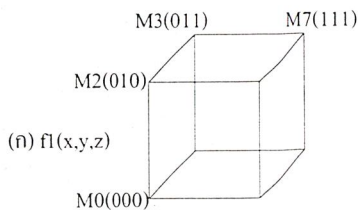
(ข)

รูปที่ 2.2 (ก) 3-cube (ข) 4-cube ให้อยู่ในระนาบเดียวกัน

จุดต่าง ๆ ของรูปทรงเรขาคณิตเลขไบนารีก็คือ มินเทอม (เขียนในรูปเต็มแบบ SP) หรือ แม็กเทอม (เขียนในรูปเต็มแบบ PS) เมื่อฟังก์ชันสวิทซ์ที่กำหนดเป็นมินเทอมมาเราจะเขียนในจุดตามที่กำหนดในรูปทรงเรขาคณิตและเชื่อมจุดตามที่เราทำได้ พิจารณาการรวมกันเป็น Sub-cube ของ จุดซึ่งจะช่วยลดทอนฟังก์ชันตามตัวอย่าง

$$F1(X,Y,Z) = m(0,2,3,7)$$

$$F2(X,Y,Z) = m(4,5,6,7)$$



รูปที่ 2.3 การแทนเทอมของฟังก์ชันในรูปทรงเรขาคณิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากฟังก์ชัน F1 (3-Cube) ที่แทนเทอมในรูปเรขาคณิตตามรูปที่ 2.3 เราสามารถเชื่อม M0 กับ M2, M3, และ M3, M7 เป็นเส้นตรง 3 เส้น (1-Sub-cube) ในเส้นตรงแต่ละเส้น จะแสดงที่บิตที่เหมือนกัน 2 บิต แตกต่างกัน 1 บิต สามารถรวมเทอมลดตัวแปรจาก 3 ตัว เป็น 2 ตัว โดยตัดตัวแปรของบิตที่ต่างกันออกไป

$$\begin{aligned} F1(X,Y,Z) &= M0 + M2 + M3 + M7 \\ &= X'Y'Z + X'Y'Z + X'YZ + XYZ \\ &= X'Z' + X'Y + YZ \end{aligned}$$

นิพจน์ของฟังก์ชันที่หาได้นี้เห็นนิพจน์ที่ลดทอนไม่ได้อีกแล้ว แต่ไม่ใช่น้อยที่สุด เพราะถ้าเราตัดเส้นตรงที่เชื่อม M2 กับ M3 โดยเฉพาะเส้นตรงที่เชื่อม M0 กับ M2 และ M3 กับ M7 จะได้นิพจน์ที่น้อยที่สุดเหลือเพียง $X'Y' + YZ$ ซึ่งครอบคลุมถึง หรือมินเทอมทั้ง 4 ได้ครบ รายละเอียดจะกล่าวถึงในหัวข้อต่อไป

ฟังก์ชัน F2(3-Cube) ที่แทนมินเทอมในรูปทรงเรขาคณิตตามรูปที่ 2.3 ข. สามารถเชื่อม M4 M5 M6 และ M7 เป็นสี่เหลี่ยม 1 รูป (2-Sub-cube) มินเทอมทั้ง 4 ในสี่เหลี่ยมจะแสดงบิตที่เหมือนกัน 1 บิต แตกต่างกัน 2 บิต สามารถรวมเทอมลดตัวแปรจาก 3 ตัว เป็น 1 ตัว โดยตัดตัวแปร ของบิตที่ต่างออกไป

$$\begin{aligned} F2(X,Y,Z) &= M4 + M5 + M6 + M7 \\ &= XY'Z' + XY'Z + XYZ' + XYZ \\ &= XY' + XY \\ &= X \end{aligned}$$

เราอาจกล่าวสรุปได้ว่า “ฟังก์ชัน” F(N-Cube) ใดๆ ที่แทนเทอมในรูปทรงเรขาคณิต เมื่อประกอบด้วย P-Sub-cube จะมีบิตที่เหมือนกัน N-P แตกต่างกัน P บิต สามารถรวมเทอมลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

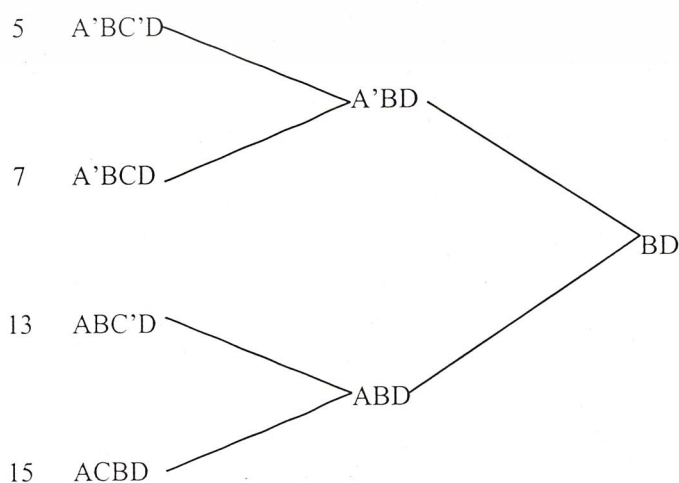
ลดตัวแปรจาก N ตัว ให้มีเทอมที่เหลือตัวแปรเพียง N-P บิต โดยตัดตัวแปรของบิตที่แตกต่างกันออกไป การรวมจุดหรือเทอมให้เป็น Sub-cube เราจะพิจารณา Sub-cube ที่ใหญ่กว่าก่อน โดยไม่พิจารณาถึง Sub-cube ย่อยที่ประกอบเป็น Sub-cube ใหญ่นั้นเพราะ Sub-cube ใหญ่จะครอบคลุมถึง Sub-cube ย่อยได้ เช่น รูปที่ 2.3 ข. เราจะพิจารณาถึงสี่เหลี่ยม 2- Sub-cube M4 M5 M6 +M7 เพียงรูปเดียว โดยไม่ต้องพิจารณาถึงเส้นตรง 1-Sub-cube ที่ประกอบกันเป็นสี่เหลี่ยม

2.2 วิธีหาฟังก์ชันผลบวกของเทอมผลคูณที่น้อยที่สุด

วิธี Quine-Mc-Cluskey (QM) นี้จะใช้หลักการพยายามจัดมินเทอมหรือจุดในรูปทรงเรขาคณิตของการแทนเลขไบนารีเป็น 1-Sub-cube เป็น 2-Sub-cube จัดต่อไปเรื่อย ๆ ให้ Sub-cube ใหญ่ขึ้น Sub-cube เก่าที่เป็นไปได้ต่าง ๆ ก็คือ PI นั่นเอง จากนั้นจึงเลือก PI เพื่อประกอบเป็นฟังก์ชัน ที่น้อยที่สุด การรวมมินเทอมให้เป็น Sub-cube มีขนาดใหญ่ขึ้นจะทำให้ตัวอักษรลดลงตามทฤษฎีของการรวม $Aa + Aa' = A(a+a') = A$ ดังนั้นเราจะกล่าวได้ว่า “เทอมผลคูณของตัวแปร K ตัว 2 เทอม สามารถรวมกันเป็นเทอมผลคูณของตัวแปร K-1 ตัวเพียงเทอมทั้งสองที่จะรวมกันต้องมีตัวอักษรที่เหมือนกัน K-1 ตัวและต่างกันเพียงตัวอักษรเดียว”

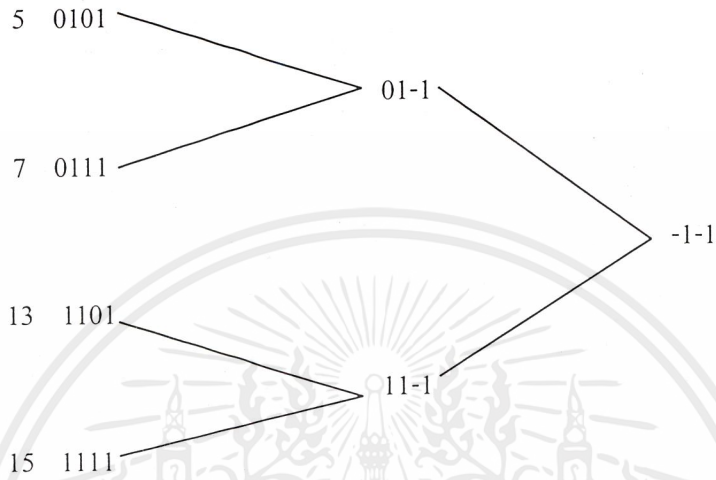
ตัวอย่างที่ 1

$$\begin{aligned} F1(A,B,C,D) &= m(5,7,13,15) \\ &= A'BC'D + A'BCD + ABC'D + ACBD \\ &\quad \text{1- Sub-cube} \qquad \qquad \text{2- Sub-cube} \end{aligned}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแทนมินเทอมต่าง ๆ เราจะแทนด้วยเลขไบนารีและบิตที่ต่างกันเมื่อรวมกันจะแทนด้วย “-” ซึ่งหมายความว่าเราสามารถตัดบิตนี้ออกไป



2.2.1 วิธีการแบบตาราง QM นี้ แบ่งออกเป็น 2 ตอน ดังนี้

- ก. หาไพรมอพลิแกนท์ (Prime Implicant) ทั้งหมดของฟังก์ชัน
- ข. หาชุดของ PI ที่ครอบคลุมมินเทอมที่แทนฟังก์ชันนั้นทุกตัว โดยมีราคาต้นทุนต่ำที่สุด โดยอาศัยตารางไพรมอพลิแกนท์

เราอธิบายวิธีการแบบตาราง QM ควบคู่ไปกับตัวอย่างข้างล่างนี้

ตัวอย่างที่ 2 จงหาฟังก์ชันที่ประหยัดที่สุดของ $F(A,B,C,D) = m(2,5,6,7,10,11,13,15)$

1) การหาไพรมอพลิแกนท์ทุกตัวของฟังก์ชัน

ขั้นที่ 1 เขียนรหัสไบนารีของตัวแปรเฉพาะค่าที่กำหนดไว้ในฟังก์ชัน $F = 1$

ขั้นที่ 2 คัดรหัสไบนารีที่มีเลข “1” เท่ากันอยู่กลุ่มเดียวกันและเรียงจากกลุ่มที่จำนวน “1” น้อยที่สุดไปหามากที่สุด จำนวนเลข “1” ในแต่ละกลุ่มเราเรียกว่า ดัชนี (INDEX)

ขั้นที่ 3 เปรียบเทียบเทอมที่อยู่ในกลุ่มดัชนีต่ำกับเทอมในกลุ่มที่มีดัชนีสูงกว่าถัดไปเมื่อใดที่

รหัสใบนารีของเทอมที่อยู่ในกลุ่มทั้งสอง คู่ใดต่างกันเพียงบิตเดียว ก็สามารถรวมเทอมคู่นั้นได้แล้ว เขียนเทอมใหม่แทนเทอมคู่นั้นให้มีเครื่องหมาย “-” แทนบิตที่ต่างกัน เช่น M2 (0010) เปรียบเทียบ M6 (0110) จะเขียนเทอมใหม่ได้เป็น 0-10 บันทึกเทอมใหม่ในตารางที่ 2 สำหรับเทอมที่ถูกรวมแล้วให้ทำเครื่องหมาย “ถูก” ไว้ข้างหลัง ให้ทำการเปรียบเทียบเช่นนี้เรื่อย ๆ สำหรับ เทอมที่อยู่ในกลุ่มดัชนีกับทุกเทอมในกลุ่มดัชนี 1+1 ที่สามารถเปรียบเทียบได้จนครบทุกกลุ่ม

ตารางที่ 2.1 การลดรูปขั้นที่หนึ่ง

ดัชนี	มินเทอม	A B C D	
1	2	0010	/
2	5	101	/
	6	110	/
	10	1010	/
3	7	111	/
	11	1011	/
	13	1101	/
	15	1111	/

ตารางที่ 2.2 การลดรูปขั้นที่สอง

ดัชนี	มินเทอม	A B C D
1	(2,6)	0-10
	(2,10)	-010
2	(5,7)	01-0
	(5,13)	-101
	(6,7)	011-
	(10,11)	101-
3	(7,15)	-111
	(11,15)	1-11
	(13,15)	11-1

หมายเหตุ ค่าในตารางที่ 2 เราหามาจากตารางที่ 1 ดังนี้

ก. กลุ่มดัชนี 1 ในตารางที่ 2 เกิดจากการเปรียบเทียบเทอมในกลุ่มดัชนี 1 และ 2 ของตารางที่ 1 M2 รวมกับ M5 ไม่ได้, M2 รวมกับ M6, M2 รวมกับ M10 ปิดเครื่องหมาย “ถูก” หลัง M2, M6, M10

ข. กลุ่มดัชนี 2 ในตารางที่ 2 เกิดจากการเปรียบเทียบเทอมในกลุ่มดัชนี 2 และ 3 ของตารางที่ 1 เริ่มจาก M5 รวมกับ M7 ปิดเครื่องหมาย “ถูก” หลัง M5 กับ M7 M5 รวม M11 ไม่ให้ผ่านไป, M5 รวมกับ M13 ปิดเครื่องหมาย “ถูก” หลัง M13 จะเห็นว่า M5 เปรียบเทียบหมดแล้วเราจึงเริ่ม M5 ซึ่งรวมกับ M7 ได้เท่านั้นสุดท้ายจึงเริ่ม M10 รวมกับ M11 ได้เท่านั้นปิดเครื่องหมาย “ถูก” หลัง M11 สำหรับเทอมที่มีเครื่องหมาย “ถูก” อยู่หลังจะไม่ต้องใส่ซ้ำอีก

ค. กลุ่มดัชนี 3 ในตารางที่ 2 เกิดจากการเปรียบเทียบเทอมในกลุ่มดัชนี 3 และ 4 ของตารางที่ 1 วิธีการก็เช่นเดียวกันกับข้อ ก. และ ข.

ขั้นที่ 4 ทำขั้นที่ 3 ซ้ำในตารางที่ 2 เพื่อให้ได้ตารางที่ 3 (ถ้ามี) จะได้เทอมใหม่ที่มีเครื่องหมาย “-” 2 บิตถ้ามีตารางที่ 3 ให้ทำขั้นที่ 3 ได้เทอมที่มีเครื่องหมาย “-” 3 บิต ทำในลักษณะเช่นนี้เรื่อย ๆ จนทำการเปรียบเทียบเทอมอีกไม่ได้ (การที่เครื่องหมาย “-” มากขึ้น แสดงว่าเราสามารถลดตัวอักษรในเทอมน้อยลง) ตั้งแต่ขั้น 3 เป็นต้นไป เทอมที่จะเปรียบเทียบรวมกันได้จะต้อง

ก. มีเครื่องหมาย “-” ในตำแหน่งเดียวกันทุกตำแหน่งของ “-”

ข. ต้องมีบิตต่างกัน 1 บิตเท่านั้น “-”

ตารางที่ 2.3 การลดรูปขั้นที่สาม

ดัชนี	มินเทอม	A B C D	
1	(2,6)	0-10	A
	(2,10)	-010	B
2	(5,7)	01-0	/
	(5,13)	-101	/
	(6,7)	011-	C
	(10,11)	101-	D
3	(7,15)	-111	/
	(11,15)	1-11	E
	(13,15)	11-1	/

ตารางที่ 2.4 การลดรูปขั้นที่สี่

ดัชนี	มินเทอม	A B C D
2	(5,7,13,15)	-1-1

หมายเหตุ ตารางที่ 2 เราจะรวมเทอมได้เฉพาะในกลุ่มดัชนี 2 และ 3

(5,7) รวมกับ (13,15) ได้ -1-1

(5,13) รวมกับ (7,15) ได้ -1-1 ซ้ำกันดังนั้นในตารางที่ 3 จะไม่บันทึกซ้ำแต่จะขีดเครื่องหมาย “ถูก” ไว้ข้างท้ายเท่านั้น

ขั้นที่ 5 เทอมทุกเทอมในทุกตารางที่ไม่มีเครื่องหมาย “ถูก” คือ ไพโรมิพลีเคนท์

ในตารางที่ 3 PI 1 ตัว คือ BD

ในตารางที่ 2 PI 5 ตัว คือ ACD, BCD, ABC, ABC, ACD

เราจะใส่ชื่อกำกับ PI ไว้ด้วยตามตารางที่แสดงมีชื่อ A,B,C,D,E,F ขั้นตอนต่อไปจะเป็นการเลือก PI เพื่อมาประกอบให้ได้ฟังก์ชันที่ประหยัดที่สุด

2) ตารางไพโรมิพลีเคนท์

เมื่อได้ PI ทุกตัวของฟังก์ชันแล้ว ต่อไปเราจะต้องหาจำนวน PI ที่มีราคาค้นทุนต่ำที่สุดซึ่งคลุมมินเทอมทุกเทอม เราจะเขียนเป็นตารางไพโรมิพลีเคนท์ดังนี้

ตารางที่ 2.5 การหาค่า PI ที่น้อยที่สุด

	2	5 /	6	7 /	10	11	13 /	15 /
A=a'cd'	X		X					
B=b'cd'	X				X			
C=a'bc			X	X				
d=ab'c					X	X		
E=acd						X		
F=bd		*		X			X	X /

ตาราง PI เป็นตารางที่มีแถวอนแทน PI และสดมภ์ (Column) แทนมินเทอมต่าง ๆ ตำแหน่งที่ตัดกันของ PI จะแทนมินเทอมจะแสดงถึงว่า PI ตัวนั้นคลุมมินเทอมนั้นหรือไม่ ถ้าคลุมให้ใส่เครื่องหมาย “ถูก” ไว้ เช่น A คลุม 2 และ 6 ก็ใส่เครื่องหมาย “ถูก” ที่แถว A ตำแหน่งที่ตัดกับ 2 และ 6 เป็นต้น การเลือกชุดของ PI ที่คลุมมินเทอมทั้งหมด เราสามารถหาออกมาเป็นรูปพีชคณิตสวิชซึ่งได้ โดยคิดว่าไพโรมิพลีเคนท์แต่ละตัวเป็นตัวแปรตัวหนึ่ง ในตารางข้างบนให้ชื่อเป็น A ถึง F จากตารางจะทำได้เป็นข้อความลอจิกได้ว่ามินเทอม M2 จะถูกคลุมด้วย A หรือ B, M5 ถูกคลุมด้วย F, M6 ถูกคลุมด้วย A หรือ C, M7 ถูกคลุมด้วย C หรือ F, M10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกคลุมด้วย B หรือ D, M11 ถูกคลุมด้วย D หรือ E, M13 ถูกคลุมด้วย F และ M15 ถูกคลุมด้วย E หรือ F เราสามารถถอดข้อความเป็นฟังก์ชันผลคูณของเทอมผลบวกแสดงถึงว่ามีเทอมทุกเทอมจะถูกครอบคลุมด้วย

$$(A+B) (F) (A+C) (C+F) (B+D) (D+E) (F) (E+F)$$

สามารถจัดใหม่ได้เป็น

$$(A+B) (B+D) (A+C) (C+F) (D+E) (E+F) F$$

และลดทอนได้เป็น

$$\begin{aligned} (A+B) (B+D) (A+C) (C+F) (D+E) (E+F) F &= (B+AD) (C+AF) (E+DF) F \\ &= BCEF + BCDF + ABEF + ADF \end{aligned}$$

กล่าวได้ว่ามีเทอมทุกเทอมจะถูกครอบคลุมจนหมดเมื่อเลือกไพรมพลิแกนท์ B,C,E,F หรือ B,C,D,F หรือ A,B,E,F หรือ A,D,F ใส่แต่ละชุดของ PI นี้จะประกอบกันเป็นนิพจน์ที่ลดทอนไม่ได้อีก จะเห็นว่าชุด ADF มีจำนวน PI น้อยที่สุด ในบางโอกาสชุดของ PI อาจมีจำนวน PI น้อยกว่าชุดอื่นมากกว่า 1 ชุดได้ การพิจารณาเพื่อลดทอนฟังก์ชัน ที่ประหยัดที่สุดจึงควรพิจารณาจำนวนตัวอักษรหรืออินพุตที่ประกอบอยู่ใน PI ด้วยในตัวอย่าง ข้างต้น A ถึง E ต่างมี 3 อินพุต ดังนั้นการเลือกชุด ADF จะประหยัดที่สุด

$$F(A,B,C,D) = ACD + ABC + BD$$

การพิจารณาเลือก PI จากตารางโดยวิธีทางพีชคณิตอาจยุ่งยากมาก ถ้ากรณีมีมินเทอมมากๆ ดังนั้นเราจะมีวิธีหาชุดของ PI ที่มีราคาค่าต้นทุนน้อยที่สุดจากตารางโดยตรงได้ตามหลักการต่อไปนี้

2.2.2 การหาแถวที่ต้องเลือก

สดมภ์ใดที่มีเครื่องหมาย “X” เพียงตัวเดียวแถวตรงกับ “X” นั้นเรียกว่า แถวที่ต้องเลือก (ESSENTIAL ROW TO SELECT) PI ของแถวนั้นเรียก PI ที่จำเป็น (ESSENTIAL PI) จากตารางแถว F เป็นแถวที่เลือก (เราทำเครื่องหมาย “X” หน้าแถว)

ดังนั้น $F = BD + \dots\dots\dots$

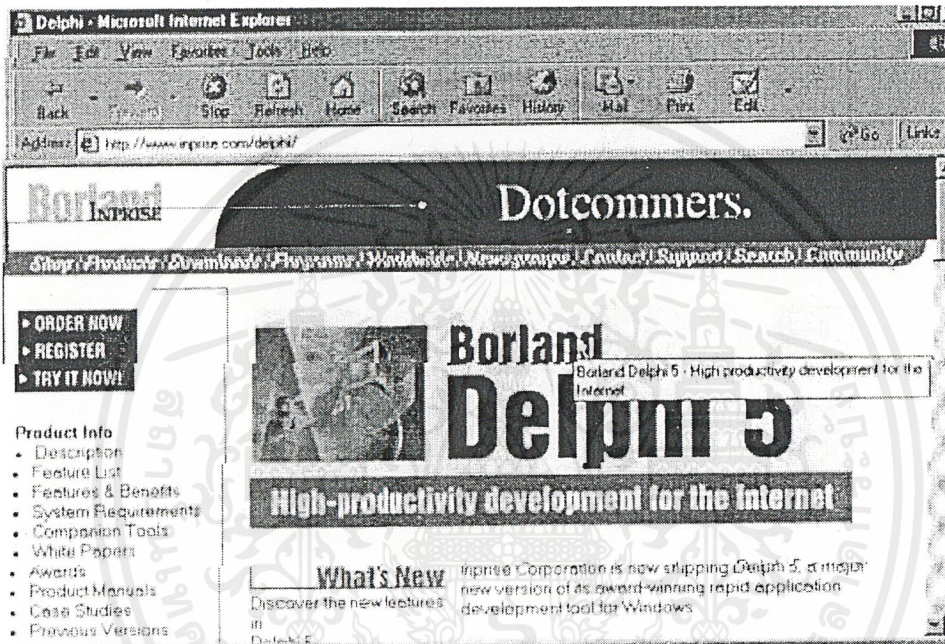
เมื่อเลือก F แล้วให้ตัดแถว F ออกจากตารางพร้อมกันนั้น สดมภ์ที่ตรงกับ “X” ทุกตัวของแถว F ให้ตัดทิ้งจากตารางด้วย เพื่อป้องกันความสับสนเราจะขีดเครื่องหมาย “ถูก” บนมินเทอมที่แสดงที่หัวสดมภ์ว่าสดมภ์ใดจะตัดออกจากตาราง ดังนั้นเราจะได้ตารางใหม่ดังนี้

ตารางที่ 2.6 ค่าสมการที่น้อยที่สุด

	2	6	10	11
A=a'cd'	X	X		
B=b'cd'	X		X	
C=a'bc		X		
D=ab'c			X	X
E=acd				X

2.3 Delphi

Delphi เป็นเครื่องมือสำหรับสร้างโปรแกรมใช้งานบนระบบปฏิบัติการ Windows และโปรแกรมประยุกต์แบบ Visual Programming (เหมือนกับ Visual Basic, Visual C++, PowerBuilder ฯลฯ)



รูปที่ 2.4 เว็บไซต์ของบริษัท Inprise

จุดเด่นที่สำคัญมากของความเป็น Visual Programming คือช่วยลดเวลาของการสร้างโปรแกรมประยุกต์ เพราะแทนที่เราจะไปหุ้มเวลาไปปรับแต่งส่วนติดต่อผู้ใช้ (User Interface) เราก็มอบภาระเหล่านี้ให้ Delphi สำหรับเราก็มุ่งเข้าไปแก้ไขปัญหาที่เป็นหัวใจการทำงานของโปรแกรมประยุกต์ดีกว่า

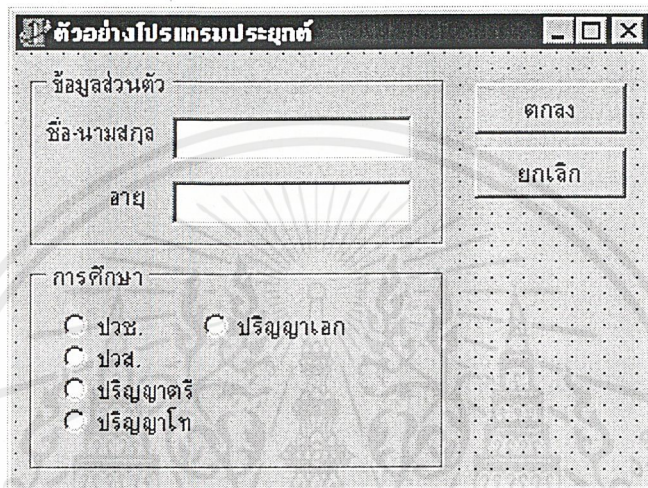
2.3.1 ความสามารถของ Delphi 5.0

Delphi ได้พัฒนามาจนถึงเวอร์ชัน 5.0 ซึ่งประกอบไปด้วยเครื่องมือช่วยในการออกแบบ, สร้าง และทดสอบโปรแกรมประยุกต์ที่หลากหลาย ช่วยให้ผลงานออกมาได้อย่างรวดเร็ว สำหรับความสามารถของ Delphi 5.0 ที่เราน่าจะได้อรรถาธิบายได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การสร้างโปรแกรมประยุกต์ (Application) จาก VCL

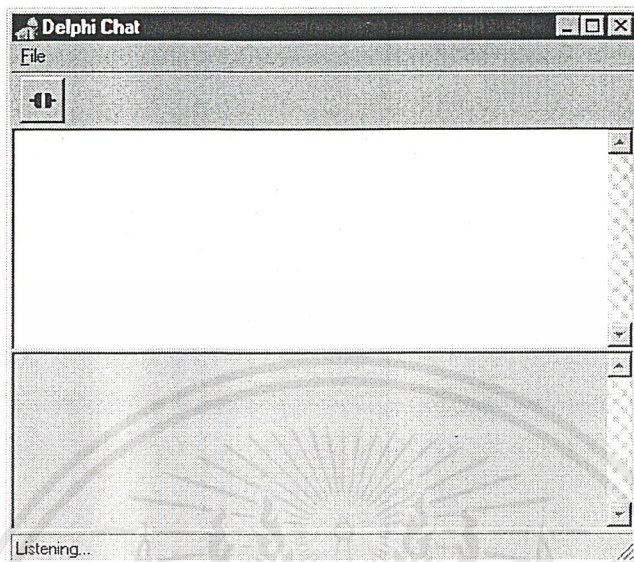
การสร้างโปรแกรมประยุกต์แบบ Visual Programming นั้น เกิดจากการนำเอาวัตถุ (Object) ต่างๆ มาประกอบกันเป็นโปรแกรมประยุกต์ ซึ่งวัตถุเหล่านั้นส่วนใหญ่จะอยู่ในรูปของส่วนติดต่อผู้ใช้ ซึ่งทำให้เราสามารถสร้างโปรแกรมประยุกต์ได้อย่างรวดเร็ว ใช้งานได้ง่าย และสวยงาม



รูปที่ 2.5 ตัวอย่างโปรแกรมประยุกต์ที่สร้างจาก VCL

2) สร้างโปรแกรมประยุกต์ใช้งานร่วมกับอินเทอร์เน็ต

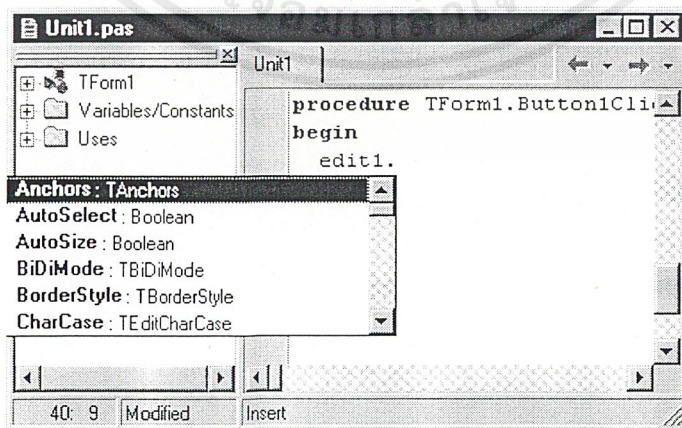
อินเทอร์เน็ตนั้น ไม่ได้มีเพียงแค่เว็บเพจที่สร้างจากภาษา HTML เท่านั้น การที่จะทำให้โปรแกรมประยุกต์ที่ใช้งานอินเทอร์เน็ตได้รับการยอมรับ นอกจากความสวยงามแล้ว สิ่งที่ต้องเพิ่มเข้ามาคือ ความสามารถในการทำงานของโปรแกรมประยุกต์ ต้องสามารถประยุกต์ใช้กับงานต่างๆ ได้เป็นอย่างดี ซึ่ง Delphi 5.0 ได้เตรียมความสะดวกสบายสำหรับการสร้างโปรแกรมประยุกต์เพื่อใช้งานร่วมกับอินเทอร์เน็ตไว้อย่างมากมาย



รูปที่ 2.6 ตัวอย่าง โปรแกรมประยุกต์อินเทอร์เน็ต

2.3.2 ความสามารถด้านการเขียนโปรแกรม

การเขียนโปรแกรมนั้นเป็นเรื่องของความมีระบบระเบียบโดยธรรมชาติอยู่แล้ว แต่เมื่อความซับซ้อนของโปรแกรมมีมากขึ้นทำให้การเขียนโปรแกรมเป็นเรื่องยุ่งยากอย่างหลีกเลี่ยงไม่ได้ ทำให้มีหลายๆ วิธีที่คิดขึ้นมาเพื่อลดความซับซ้อนลง ซึ่ง Delphi ก็มีหลากหลายวิธีที่ช่วยให้เราเขียนและทดสอบโปรแกรมที่เขียนได้อย่างสะดวกสบาย

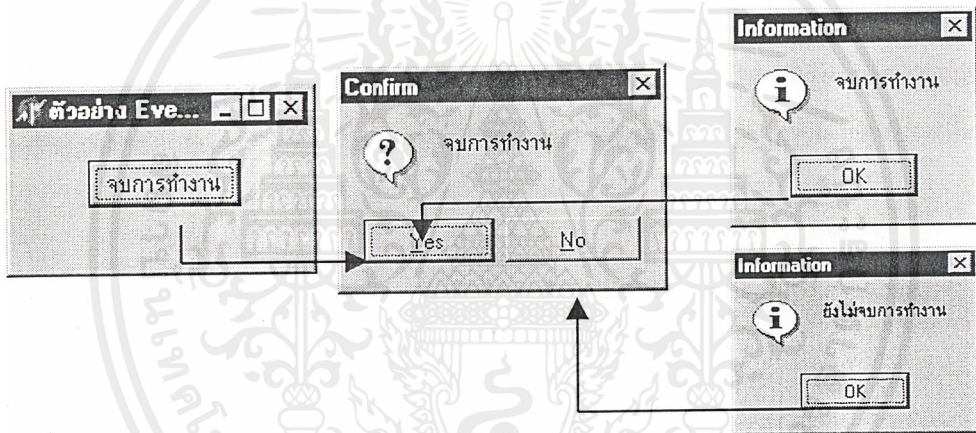


รูปที่ 2.7 ลดความยุ่งยากในการเขียน โปรแกรมใน Delphi 5.0

2.4 สร้างโปรแกรมประยุกต์ แบบ Event-Driven

โปรแกรมประยุกต์ ที่สร้างจาก Delphi นั้นมีวิธีการสร้างที่แตกต่างจากการเขียนโปรแกรมแบบเดิมๆ ที่เราเคยเรียนรู้มาก่อน หากเราอาจเคยเขียนโปรแกรมภาษา BASIC ,C หรือ Pascal ก็ จะเห็นว่ามีแนวคิดที่ค่อนข้างต่างกัน ซึ่งเราจะเรียกวิธีการที่สร้างโปรแกรมประยุกต์ด้วย Delphi ว่า Event-Driven (แปลว่า เหตุการณ์พาไป)

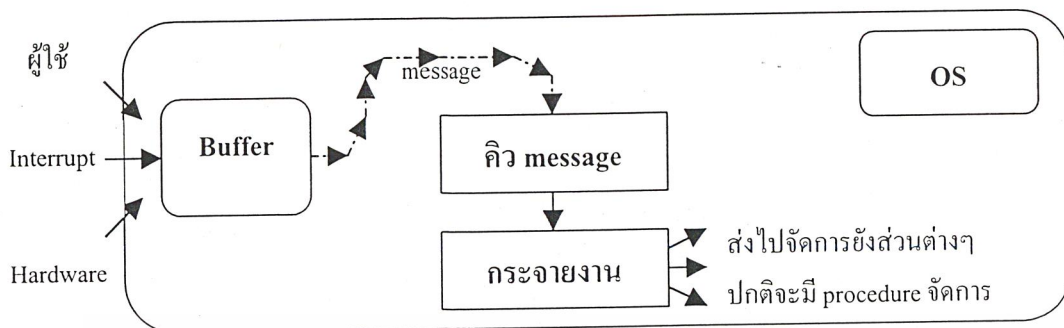
Event-Driven ที่จริงก็คือการเขียนโปรแกรมในลักษณะที่ว่า “ถ้ามีเหตุการณ์เกิดขึ้น เราจะจัดการกับมันอย่างไร ” เช่นถ้าผู้ใช้คลิกที่ปุ่ม Exit เราจะทำอย่างไร? เราอาจจะถามผู้ใช้งานว่า แน่ใจแล้วว่าจะจบการทำงานของโปรแกรม ถ้าผู้ใช้ยืนยันก็จบไป แต่ถ้าไม่ก็ให้โปรแกรมประยุกต์ทำงานต่อไป เป็นต้น



รูปที่ 2.8 โปรแกรมประยุกต์ตัวอย่างที่สร้างจาก Delphi

2.5 ระบุ Message Loop ของ Windows

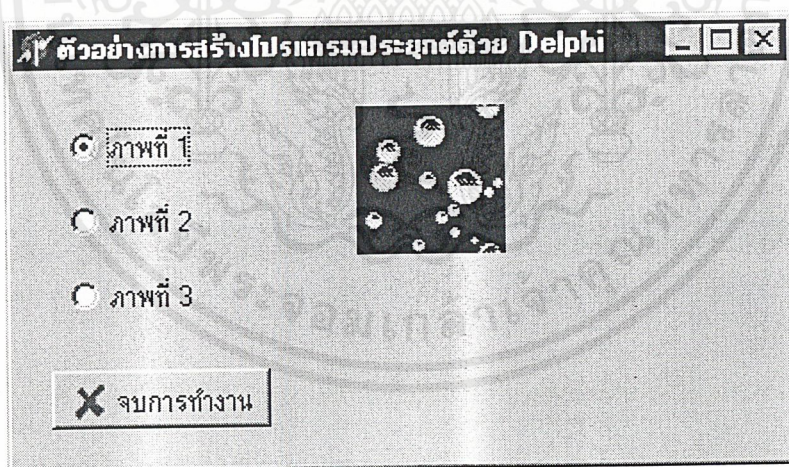
Event-Driven ถือได้ว่าเป็นเหมือนหลักการทำงานของ Windows ก็ว่าได้ ที่จริงแล้วเบื้องหลังการทำงานของ Windows ก็คือจะมีการรอรับการทำงานจากผู้ใช้ จากฮาร์ดแวร์ต่างๆ ส่งเหตุการณ์ต่างๆ เข้ามาสู่ Message Loop ซึ่งจะพิจารณาว่า มีเหตุการณ์อะไรเกิดขึ้น ซึ่งแต่ละเหตุการณ์ Windows ได้เตรียมวิธีการจัดการแต่ละเหตุการณ์เอาไว้แล้ว



รูปที่ 2.9 แนวคิดของ Message Loop ของ Windows

2.6 วัตถุ (Object)

จากโปรแกรมประยุกต์ที่ผ่านมามาดูเราจะเห็นว่า โปรแกรมประยุกต์ประกอบไปด้วย วินโดว์, ปุ่มกด หรือ ตัวเลือก ซึ่งเราจะเรียกสิ่งต่างๆ ที่นำมาสร้างเป็น โปรแกรมประยุกต์ นี้ว่า วัตถุ



รูปที่ 2.10 องค์ประกอบของ โปรแกรมประยุกต์

จะเห็นว่า โปรแกรมประยุกต์ตัวอย่างนั้นประกอบด้วยวัตถุต่างๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

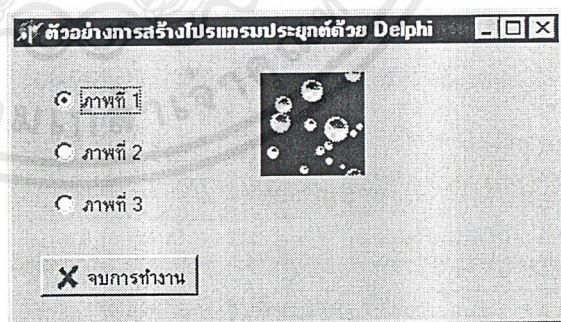
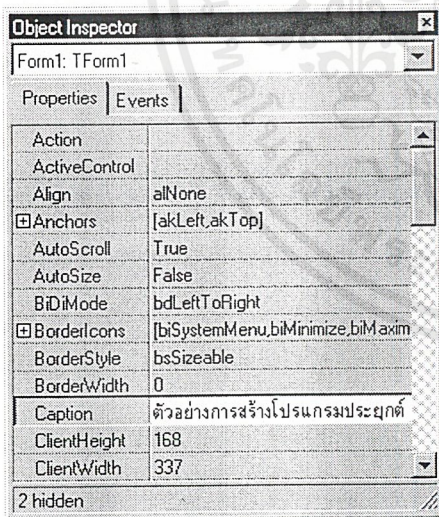
ตารางที่ 2.7 รายละเอียดปุ่มต่างๆของโปรแกรมประยุกต์

วัตถุ	ลักษณะของวัตถุ
ฟอร์ม (Form)	เป็นวัตถุที่เรามองเห็นเป็นวินโดว์ ซึ่งมีวัตถุอื่นๆ วางอยู่ภายในมีข้อความอยู่ด้านบน
ปุ่มกด (button)	เป็นวัตถุที่ใช้การคลิกเมาส์เพื่อรับคำสั่งหรือกด Enter
ปุ่มตัวเลือก (RadioButton)	เป็นวัตถุที่ให้เรากดคลิกเมาส์เพื่อเลือก ตัวใดตัวหนึ่ง
รูปภาพ (Image)	เป็นวัตถุที่ทำหน้าที่แสดงรูปภาพ

2.7 คุณสมบัติ (Property)

จะเห็นได้ว่าแต่ละวัตถุมีลักษณะเฉพาะตัวที่แสดงความเป็นวัตถุ เช่น ปุ่มกด ก็จะมีข้อความที่อยู่บนตัวมัน เช่น จบการทำงาน หรือ บอกเวลา ซึ่งเราเรียกลักษณะเฉพาะตัวนี้ว่า คุณสมบัติ

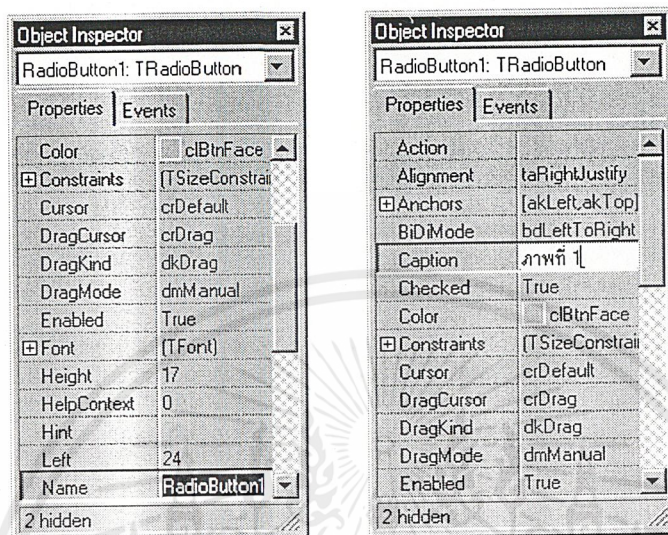
ในวัตถุชนิดเดียวกันก็จะมีคุณสมบัติอย่างเดียวกัน แต่อาจจะมีค่าของแต่ละคุณสมบัติที่แตกต่างกัน โดยเราสามารถกำหนดค่าคุณสมบัติได้ใน Object Inspector



รูปที่ 2.11 ตัวอย่างการกำหนดค่าคุณสมบัติใน Object Inspector

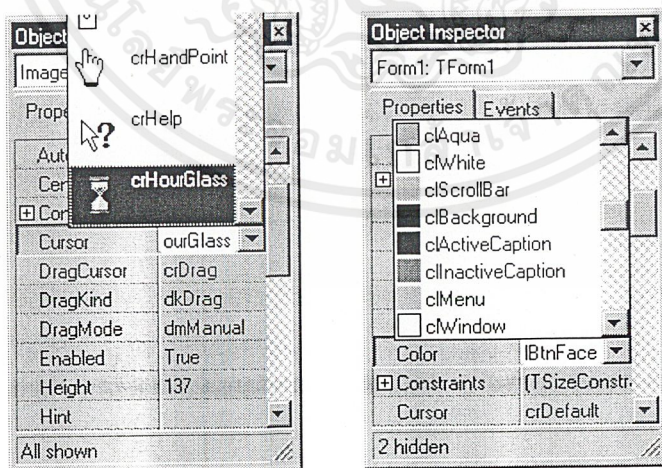
ลักษณะการกำหนดคุณสมบัติใน Delphi นั้นมี 3 รูปแบบ ได้แก่

1. การกำหนดค่าอย่างอิสระ เช่น คุณสมบัติ Caption เป็นต้น



รูปที่ 2.12 กำหนดค่าคุณสมบัติ โดยระบุค่าอย่างอิสระ

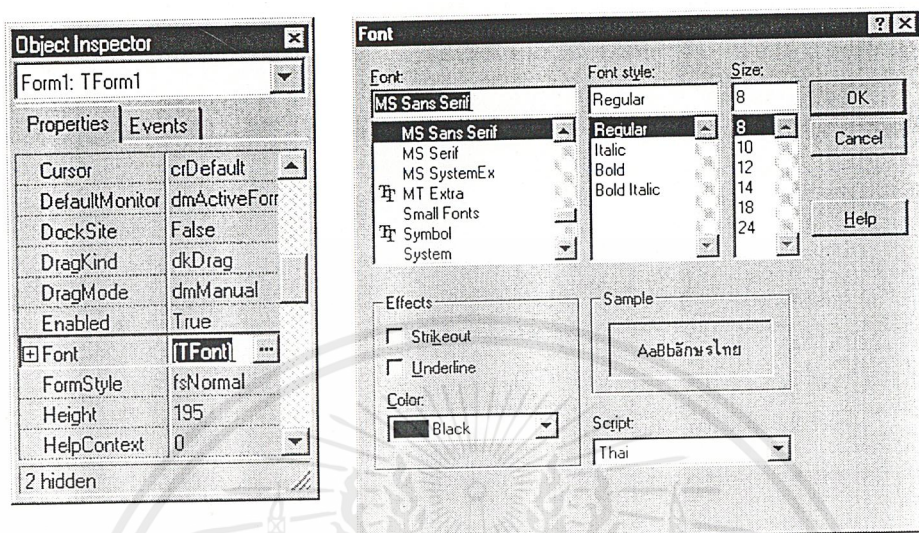
2. การกำหนดค่าจากตัวเลือกที่มีให้ใน Object Inspector เช่น คุณสมบัติ Cursor , คุณสมบัติ BackColor เป็นต้น



รูปที่ 2.13 กำหนดค่าคุณสมบัติโดยเลือกจากตัวเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การกำหนดค่าจากไดอะล็อกบ็อกซ์ เช่น คุณสมบัติ Font เป็นต้น



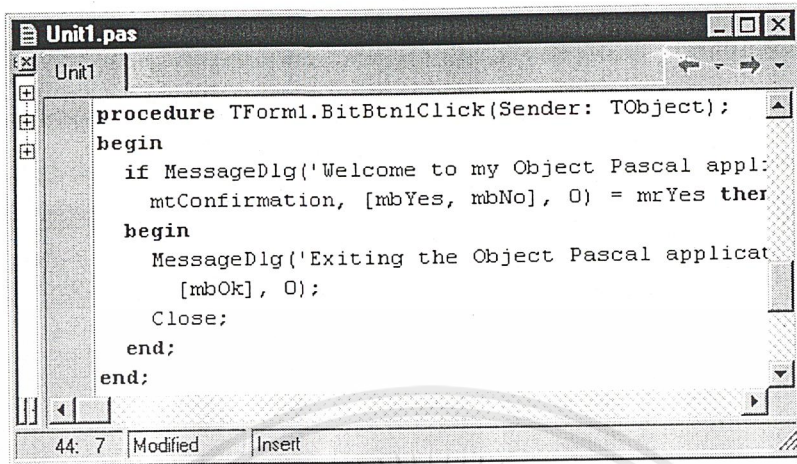
รูปที่ 2.14 กำหนดค่าคุณสมบัติจากไดอะล็อกบ็อกซ์

นอกจากการระบุค่าคุณสมบัติจาก Object Inspector ซึ่งถือเป็นการกำหนดค่าก่อนการทำงานของโปรแกรมประยุกต์ (Design-Time) แล้ว เรายังสามารถกำหนดค่า หรือเปลี่ยนค่าคุณสมบัติได้โดยการเขียนลงไปโปรแกรมของเรา ซึ่งก็จะทำให้มีการกำหนดค่า หรือเปลี่ยนแปลงค่าในขณะที่โปรแกรมประยุกต์ ทำงาน (Run-Time) ได้เช่นกัน

2.8 เมธอด (Method)

นอกจากจะใช้คุณสมบัติในการบอกความแตกต่างของวัตถุแต่ละตัวแล้ว ยังมีอีกสิ่งหนึ่งที่วัตถุแต่ละตัวมักจะมีนั่นก็คือ เมธอด

เมธอด คือ ความสามารถในการทำงานอย่างใดอย่างหนึ่งของวัตถุ เช่น วัตถุ Form มีความสามารถในการวาดวงกลมได้ เป็นต้น ในการเรียกใช้เมธอดเราจะเรียกโดยการเขียนโปรแกรม



รูปที่ 2.15 เรียกใช้เมธอดของวัตถุในโปรแกรม

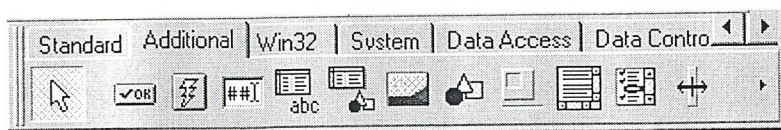
2.8.1 อีเวนต์ (Event)

อีเวนต์ คือ เหตุการณ์ที่เกิดขึ้นกับแต่ละวัตถุ เช่น ปุ่มก็จะมีเหตุการณ์ที่ผู้ใช้คลิกเมาส์ (อีเวนต์ OnClick) หรือ กรอบข้อความ จะมีเหตุการณ์เมื่อผู้ใช้กรอกข้อความ (อีเวนต์ OnKeyPress) เราสามารถเขียนคำสั่งต่างๆ เพื่อจัดการกับเหตุการณ์ ที่เกิดขึ้น โดยการเลือกเหตุการณ์ที่ต้องการจากแท็บ Events ใน Object Inspector

2.9 คอมโพเนนต์ของ Delphi

ข้อดีของการเขียนโปรแกรมเพื่อสร้างโปรแกรมประยุกต์ใน Delphi ก็คือ เมื่อเราสมมติสิ่งต่าง ๆ เป็นวัตถุที่มีทั้งคุณสมบัติ, เมธอดแล้ว เราก็สามารถนำมาใช้งานได้อย่างเรื่อยๆ (Reusable Object) ทำให้ไม่ต้องเสียเวลาสร้างกันใหม่ทุกครั้งที่เราสร้างโปรแกรมประยุกต์

Delphi จะเรียกวัตถุต่างๆ ที่สามารถนำกลับมาใช้ในการสร้างโปรแกรมประยุกต์ว่า คอมโพเนนต์ (Component) ซึ่งจะถูกเก็บไว้ใน Component Palette พร้อมให้เรานำมาใช้งาน

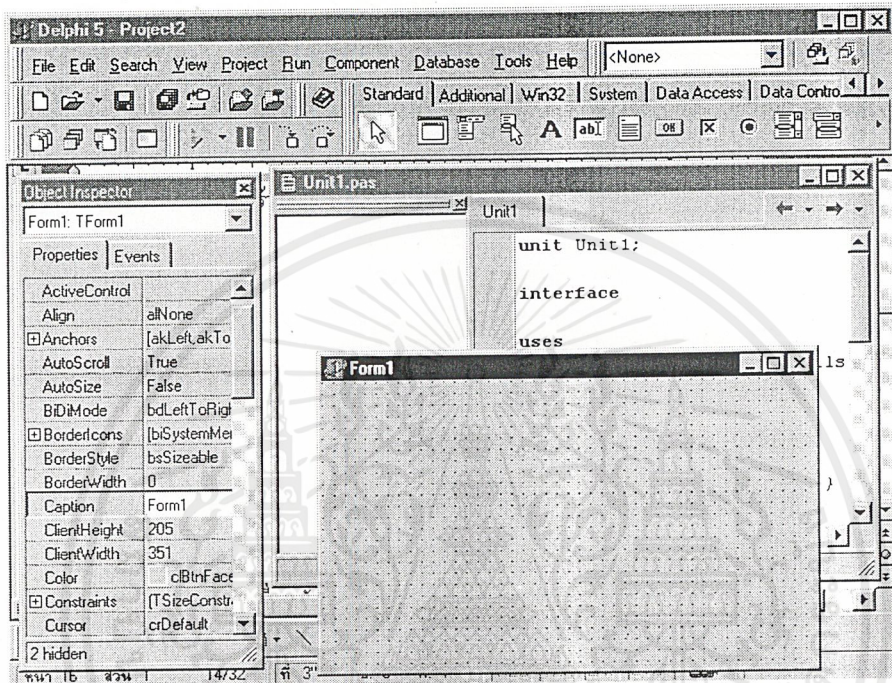


รูปที่ 2.16 คอมโพเนนต์ต่างๆ ใน Component Palette

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 ส่วนประกอบการทำงานของ Delphi

การทำงานเพื่อสร้างโปรแกรมประยุกต์กับ Delphi 5.0 นั้นเราจะอยู่ภายใต้สภาวะแวดล้อมการพัฒนา นั่นคือ ประกอบด้วยเครื่องมือชนิดต่าง ๆ ที่นำมาสร้าง และทดสอบโปรแกรมประยุกต์



รูปที่ 2.17 องค์ประกอบการทำงานของ Delphi 5.0

2.10.1 Menu Bar

Menu Bar เป็นส่วนที่เก็บคำสั่งเพื่อเรียกใช้งาน Delphi 5.0 ในรูปแบบ ซึ่งอยู่บนสุดของการแสดงผล

2.10.2 Form Designer

Form Designer เป็นส่วนหนึ่งที่ออกแบบหน้าต่างของฟอร์ม ซึ่งถือเป็นพื้นที่แสดงผลหลักที่เราสามารถนำเอาวัตถุต่าง ๆ มาวางไว้ภายในให้เป็นโปรแกรมประยุกต์ตามที่เรากำลังต้องการได้

2.10.3 Object Inspector

เราสามารถกำหนดค่าคุณสมบัติของวัตถุต่าง ๆ ผ่านทาง Object Inspector นอกจากนี้เรายังเริ่มต้นจัดการกับอ็อบเจกต์ต่าง ๆ ของวัตถุผ่านทาง Object Inspector เช่นเดียวกัน

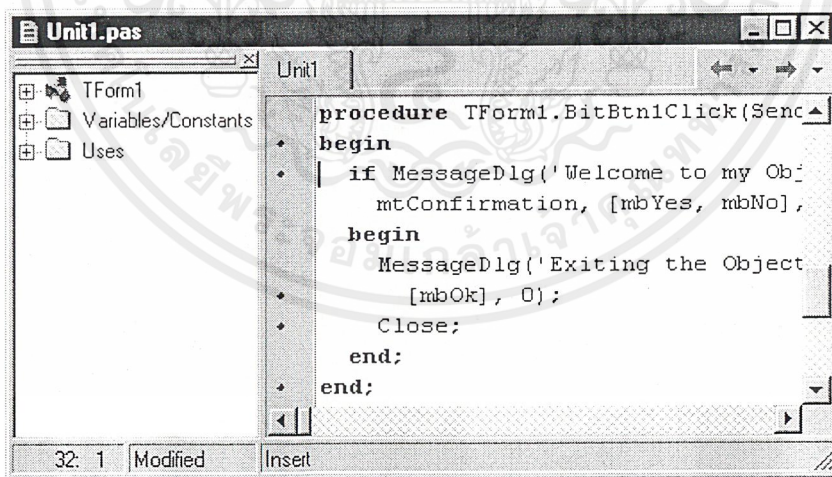
2.10.4 Component Palette

Component palette คือส่วนที่เก็บคอมโพเนนต์ต่าง ๆ แยกเป็นหมวดหมู่เอาไว้ พร้อมให้เรานำไปประกอบในโปรแกรมประยุกต์ที่สร้างขึ้น และไม่ว่าจะมีเพียงแต่คอมโพเนนต์ที่ Delphi 5.0 เตรียมไว้ให้เท่านั้น เรายังสามารถสร้างคอมโพเนนต์มาใช้งานเอง แล้วบรรจุไว้ใน Component palette ได้ด้วย

2.10.5 Code Editor

Code Editor เป็นส่วนที่เราใช้ในการเขียนคำสั่งต่าง ๆ เข้าไปเพื่อกำกับการทำงานของโปรแกรมประยุกต์ให้สอดคล้องกันซึ่งคำสั่งต่าง ๆ ในภาษาปาสคาลที่อยู่ใน Code Editor จะประกอบไปด้วยส่วนที่ Delphi เตรียมไว้ให้ และส่วนที่เราต้องเขียนเพิ่มเติมเข้าไป

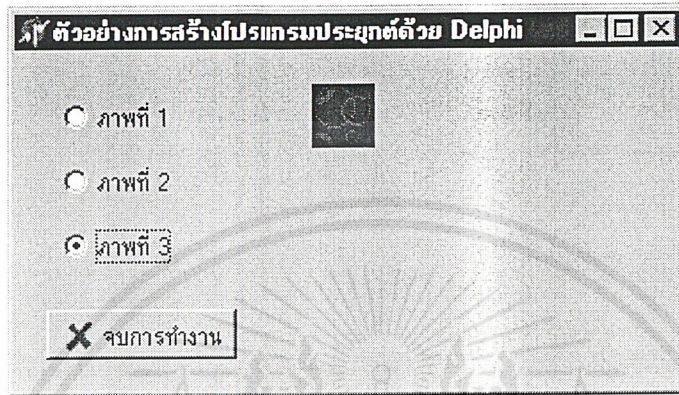
ใน Code Editor เรามักจะเห็น Code Explorer ควบคู่ไปด้วย ซึ่ง Code Explorer จะช่วยให้เราเห็นตัวแปร, โปรแกรมย่อย วัตถุ, คุณสมบัติ ของวัตถุต่าง ๆ ที่ประกอบกันเป็นโปรแกรมประยุกต์



รูปที่ 2.18 Code Explorer และ Code Editor

2.11 ขั้นตอนการสร้างโปรแกรมประยุกต์เบื้องต้นกับ Delphi 5.0

เพื่อความเข้าใจในวิธีการสร้างโปรแกรมประยุกต์ ตอนนี้เราจะมาทดลองสร้างโปรแกรมประยุกต์ด้วยตัวของเราเองที่ละขั้นตอนดังต่อไปนี้



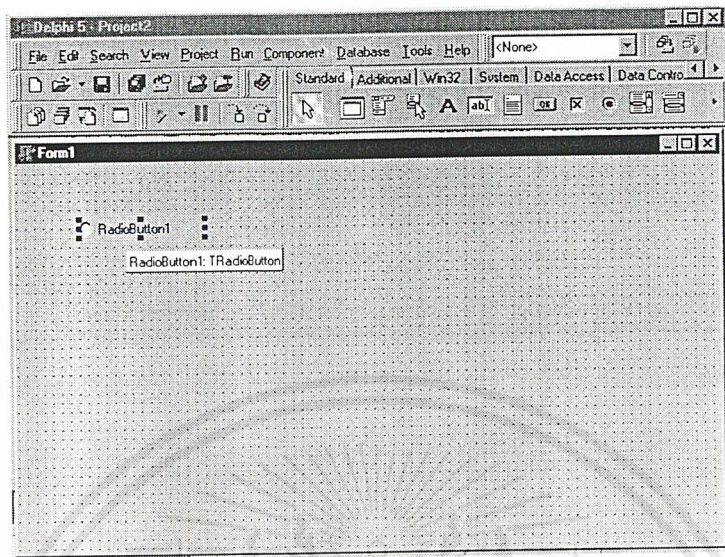
รูปที่ 2.19 หน้าตาของโปรแกรมประยุกต์ตัวอย่าง

2.11.1 ขั้นแรก : ออกแบบการทำงาน และหน้าตาของโปรแกรมประยุกต์

ถือเป็นขั้นตอนแรกที่ต้องทำ ต้องนึกให้คิดว่าอยากจะให้หน้าตาของโปรแกรมประยุกต์เป็นอย่างไรถ้า นึกไม่ออก หรือมีความซับซ้อน ก็วาดลงในกระดาษก่อน

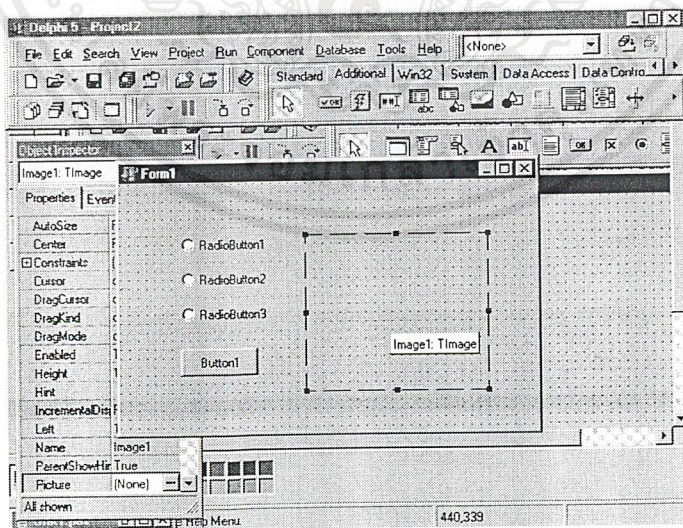
2.11.2 ขั้นสอง : วางคอมโพเนนต์ที่ต้องการลงในฟอร์ม

ขั้นต่อมา เราก็นำเอาคอมโพเนนต์ต่าง ๆ จาก Component Palette มาวางบนฟอร์มใน Form Designer โดยให้เราคลิกเลือกที่คอมโพเนนต์ที่ต้องการ จากนั้นมาคลิกที่ฟอร์ม



รูปที่ 2.20 เลือกคอมโพเนนต์จาก Component Palette

จัดวางคอมโพเนนต์ต่าง ๆ ให้อยู่ในตำแหน่งที่ต้องการอย่างสวยงาม สังเกตว่าคอมโพเนนต์ต่าง ๆ ไม่ห่าง หรือไม่ชิดกันเกินไป

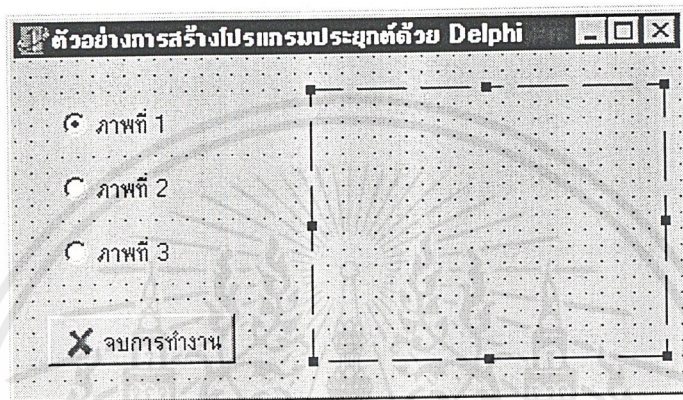


รูปที่ 2.21 การจัดวางคอมโพเนนต์ใน Form Designer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.3 ขั้นที่สาม : กำหนดคุณสมบัติให้แก่คอมโพเนนต์

เมื่อได้ส่วนติดต่อกับผู้ใช้แล้วแล้ว เราก็มากำหนดคุณสมบัติให้แก่คอมโพเนนต์ โดยใช้ Object Inspector กำหนดค่าที่ต้องการ หลังการกำหนดคุณสมบัติให้กับคอมโพเนนต์ต่าง ๆ แล้วจะได้หน้าต่างโปรแกรมประยุกต์ ดังนี้

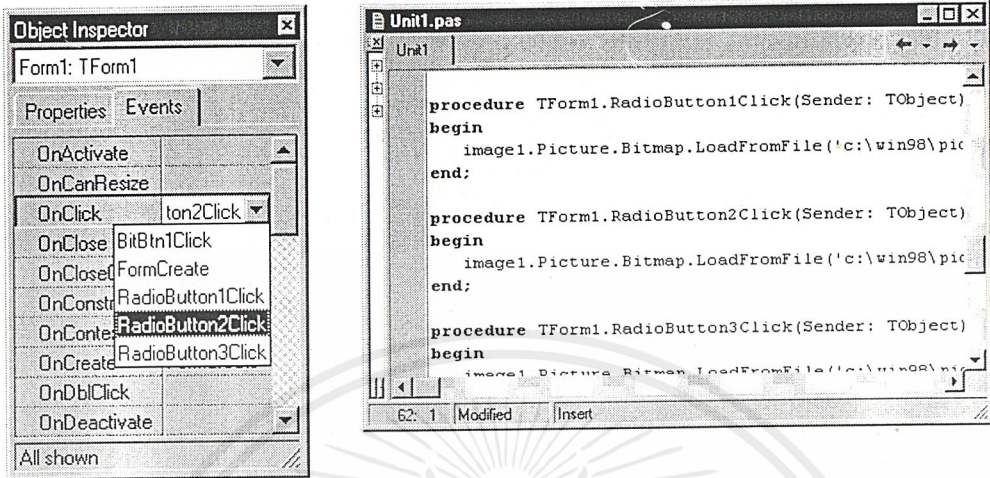


รูปที่ 2.22 หลังการกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์

2.11.4 ขั้นที่สี่ : เขียนคำสั่งเกี่ยวกับการทำงานของแต่ละคอมโพเนนต์

เพื่อให้การทำงานร่วมกันของวัตถุ หรือคอมโพเนนต์ต่าง ๆ เป็นไปอย่างราบรื่น เราจำเป็นต้องเขียนคำสั่งขึ้นมากำกับการทำงาน โดยเราจะเขียนไว้ใน Code Editor

เราเรียก Code Editor มาใช้งานโดยเลือกเมนู View > Code Editor หรือใช้การดับเบิ้ลคลิกที่คอมโพเนนต์ใด ๆ ในฟอร์ม



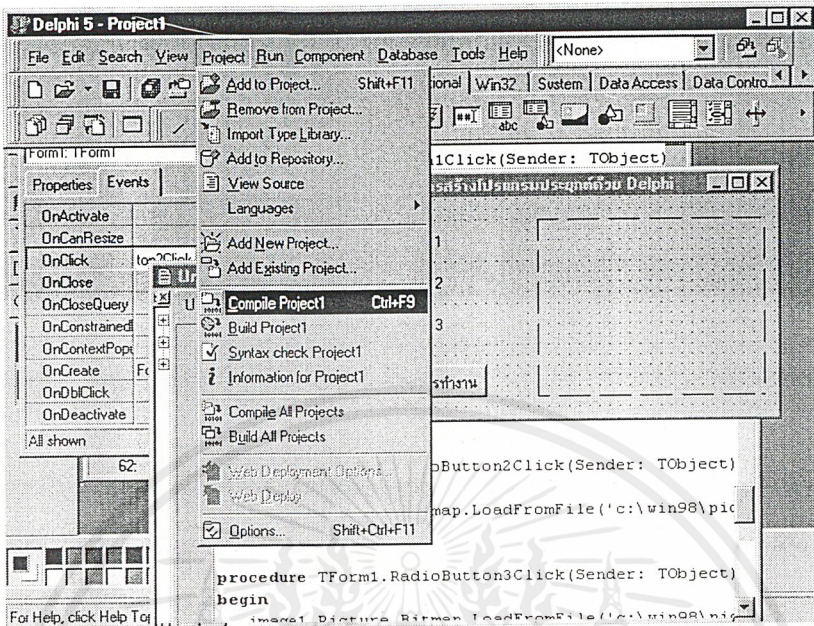
รูปที่ 2.23 การเขียนคำสั่งภายใน

จากคำสั่งที่เราเขียนขึ้น เราสั่งให้มีการโหลดภาพที่เราชื่นชอบเมื่อคลิกที่ RadioButton แต่ละตัว และที่ปุ่ม จบการทำงาน ให้มีการสอบถามก่อนว่า ต้องการจบการทำงานหรือไม่ ถ้าคลิกปุ่ม Yes ก็ให้สิ้นสุดการทำงาน แต่ถ้าคลิกปุ่ม No ก็ให้ยังคงทำงานต่อไป

2.11.5 ขั้นที่ห้า : ทำการแปล (Compile)

เพื่อให้คอมพิวเตอร์เข้าใจคำสั่งต่าง ๆ ที่เราเขียนขึ้น เราจะต้องทำการแปล (Compile) คำสั่งที่เขียนขึ้น ซึ่งก็คือการแปลคำสั่งต่าง ๆ ที่เขียนขึ้น (Source Code) จากภาษาปาสคาล ซึ่งมนุษย์อย่างเราเข้าใจ และอ่านได้รู้เรื่อง ให้เป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ (Object Code) และทำงานได้

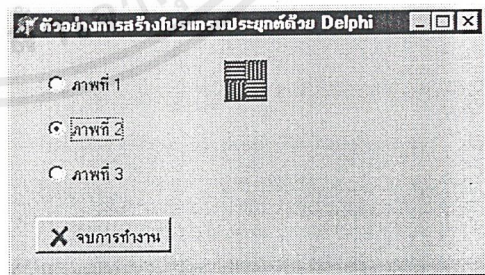
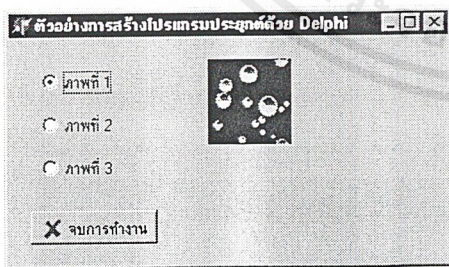
การแปลทำได้โดยเลือกเมนู Project > Compile ชื่อโปรเจกต์ในการแปลนั้น Delphi จะตรวจสอบความถูกต้องของคำสั่งที่ใช้ให้ด้วย ถ้าผิดพลาดจะแจ้งให้ทราบ



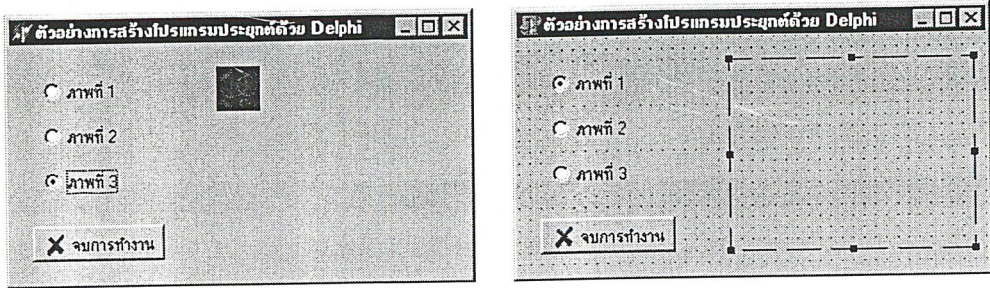
รูปที่ 2.24 การแปลงโปรแกรมประยุกต์

2.11.6 ขั้นที่หก : ทดสอบการทำงานของโปรแกรมประยุกต์

เมื่อแปลงเสร็จ ขั้นตอนถัดไปคือ ทดสอบการทำงาน ว่ามีความถูกต้อง และตรงกับความต้องการของเราหรือไม่ โดยเลือกเมนู Run > Run ซึ่งจะเป็นการสั่งให้โปรแกรมประยุกต์ที่สร้างขึ้นเริ่มทำงาน



รูปที่ 2.25 สั่งทดสอบ โปรแกรมประยุกต์

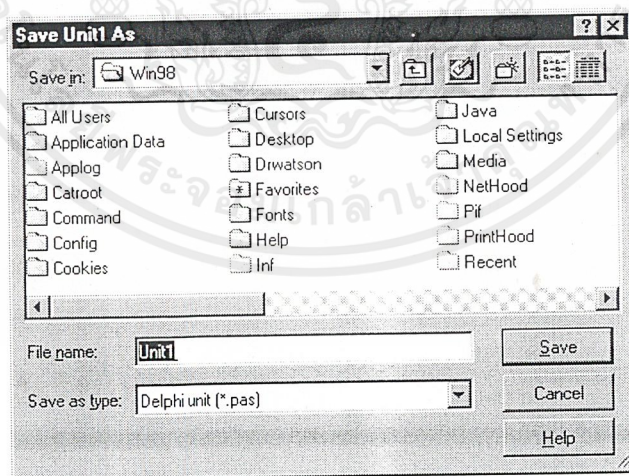


รูปที่ 2.25 (ต่อ) สิ่งทดสอบโปรแกรมประยุกต์

เราสามารถลดขั้นตอนในการแปลงให้เริ่มทำงานได้เลย แต่ที่จริงแล้ว Delphi จะแปลงก่อน จากนั้นจึงสั่งให้โปรแกรมประยุกต์เริ่มทำงานตามที่เราต้องการ

2.11.7 ขั้นที่เจ็ด : บันทึกโปรแกรมประยุกต์ที่สร้างไว้

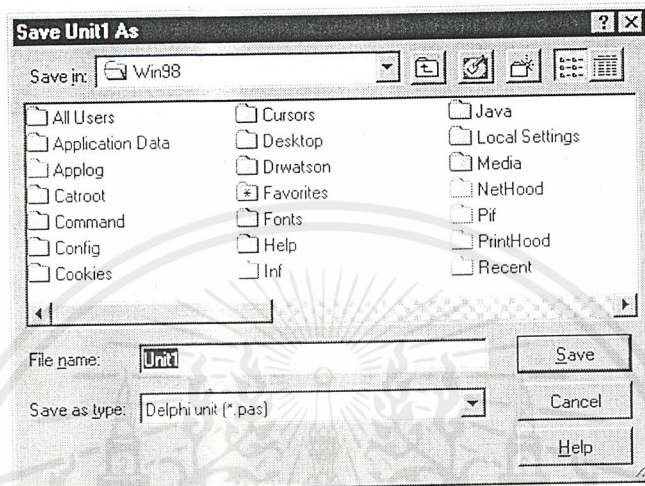
เมื่อสร้างโปรแกรมประยุกต์จนเสร็จแล้ว เราสามารถบันทึกไฟล์ต่างๆเก็บไว้ เพื่อการเรียกใช้งาน หรือแก้ไขในครั้งต่อไปได้ ในการบันทึกจะเริ่มจากบันทึกรายละเอียดของฟอร์ม และโปรแกรมที่เราเขียนขึ้นมา โดยคลิกที่ปุ่ม Save หรือเรียกเมนู File > Save ซึ่งไฟล์ที่บันทึกจะมีนามสกุลเป็น PAS



รูปที่ 2.26 บันทึกโปรแกรมที่เขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นเราจะบันทึกโปรเจกต์ ซึ่งได้ชื่อว่าเป็นตัวแทนของการเรียกใช้โปรแกรมประยุกต์ (เราจะได้เรียนรู้รายละเอียดเกี่ยวกับโปรเจกต์ในบทถัดไป) โดยเรียกเมนู File>Save Project As... ซึ่งโปรเจกต์ที่บันทึกจะมีนามสกุลเป็น .DPR



รูปที่ 2.27 บันทึกโปรเจกต์

2.12 แนะนำภาษา VHDL (VHSIC Hardware Description Language)

VHDL คือ โปรแกรมภาษาที่สามารถสร้างรูปแบบการทำงานของระบบสัญญาณดิจิทัล (VHDL สามารถใช้กับระบบดิจิทัลเท่านั้น)

2.12.1 ลักษณะการทำงานแบบต่างๆของ VHDL

1. สร้างแบบจำลองการทำงานของระบบตั้งแต่เกทง่ายๆ จนถึงระดับ ไมโครโปรเซสเซอร์
2. สังเคราะห์ระบบจากแบบจำลอง คือ เมื่อได้แบบจำลองระบบที่เป็นโปรแกรมภาษา VHDL สามารถที่จะแปลงเป็นวงจรซึ่งอาจจะเขียนลงในชิพแล้ว ชิปนั้นก็มีความสมบัติตามที่เราเขียนโปรแกรม เช่น Xilinx หรืออาจจะเป็นขบวนการผลิตในโรงงานเลย
3. ทดสอบแบบจำลองที่สร้างขึ้น เช่น มีการเขียนวงจรการนับขึ้นมา ก็จะมีการสร้าง Seven Segment จาก VHDL เพื่อที่จะเอามาทดสอบการทำงานของวงจรมัน
4. เป็นมาตรฐานโปรแกรมที่จำลองการทำงานของระบบดิจิทัลภาษาต่างๆ

2.12.2 ข้อดีของ VHDL

1. สามารถสร้างระบบต่างๆ ตามความต้องการได้ง่ายเพราะเป็นการจำลองไม่เหมือนที่ต้องไปสร้างระบบจริงเลย
2. การส่งข้อมูลของการผลิตเป็นไปได้ง่ายเพราะอยู่ในรูปของไฟล์
3. สามารถสร้างระบบต่างๆ ได้ราคาถูกและประหยัดเวลา
4. สามารถทำระบบให้อยู่ในรูปของชิพตัวเดียวได้โดยการ โปรแกรมชิพพวก Xilinx

2.12.3 ลักษณะของภาษา VHDL

VHDL Module ประกอบด้วย Statements และ code segment

```
--Interface
entity XOR2_OP is
--Input / Output ports
port
    (A,B : in BIT; Z: out BIT);
end XOR2_OP;
--Body
architecture EX_DISJUNCTION of XOR2_OP is
begin
    z <= A xor B;
end EX_DISJUNCTION;
```

1) รูปแบบการใช้ port

1. ชื่อ
2. ทิศทางการไหลของข้อมูล
3. ชนิดของข้อมูลที่สัญญาณ

2) ทิศทางการไหลหรือเรียกว่า mode มีอยู่ด้วยกัน 5 mode คือ

1. in ข้อมูลไหลเข้า design entity อย่างเดียว
2. out ข้อมูลไหลออกจาก design entity อย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. inout ข้อมูลทั้งไหลเข้าและออกจาก design entity
4. linkage
5. buffer

2.12.4 ชนิดของข้อมูลที่สัญญาณ

BIT คือ “0”, “1” คำว่า BIT เป็นชนิดของข้อมูลที่กำหนดไว้ล่วงหน้า

Signal assignment statement

$Z <= A \text{ xor } B$ ← Assignment operator

Logic Operator

And conjunction

Or disjunction

Xor exclusive disjunction

Xnor complement exclusive disjunction

Nand complement disjunction

. Not complement

Operand สามารถมีพร้อมกันหลายๆตัวเช่น

$A * B * C$ $Z <= A \text{ and } B \text{ and } C$

$(/A * B) + (A * /B)$ $Z <= (\text{not } A \text{ and } B) \text{ or } (A \text{ and } \text{not } B)$

บทที่ 3

การออกแบบ การสร้างและการทำงาน

3.1 วิธีการเขียนโปรแกรม

ในบทนี้เป็นการแนะนำวิธีการเขียนโปรแกรมลดรูปสวิตซ์ฟังก์ชันซึ่งจะมีการกล่าวถึงโครงสร้างการเขียนโปรแกรมและการแปลงเลขฐานสิบ เพื่อจะอยู่ในรูป ของ (CELL C) เพื่อที่จะนำไปใช้กล่าวในคุณสมบัติของการบรรจุ และรวบรวมข้อมูลทั้งหมด ซึ่งนำไปประยุกต์เป็นทฤษฎีที่จะคำนวณหาค่าไพรม์อิมพลิเมนต์ และเอสเซนเชียลไพรม์อิมพลิเมนต์ (Essential Prime Implicant) ซึ่งจะใช้ในการลดรูป

3.1.1 โครงสร้างของ Cellular (The Cellular Struction)

ในการใช้ควินแมคคลอสกี้นั้นตัวแปรที่เป็นคอมพลิเมนต์จะแสดงค่าเป็น 0 และตัวแปรที่ไม่อยู่ในรูปของคอมพลิเมนต์จะแทนค่าด้วย 1 สุดท้ายในตำแหน่งที่ไม่มีการการแสดงค่าของตัวแปรจะถูกแทนด้วย - เมื่อมีการแสดงเทอมผลของบูลีนฟังก์ชันอาจจะถูกเขียนได้เป็น

$$C = (C_n C_{n-1} \dots C_1) \quad (3.1)$$

เมื่อ C เป็นสมาชิกของเซต (0,1,-) และเทอมผลคูณ (Product term) C จะเรียกว่า CELL ของ Cellular Struction

ตัวอย่าง

$$\begin{aligned} F(A,B,C,D) &= ABC'D + AB'D + A'D' + B'C \\ &= C1 + C2 + C3 + C4 \\ &= (1101) + (10-1) + (0- - 0) + (-01-) \end{aligned}$$

3.1.2 THE N-CUBE

เซตของ $C = \{0,1\}$ เป็นส่วนหนึ่งของเซตที่ถูกจัดลำดับ ซึ่งลำดับของความสัมพันธ์ (\geq) ถูกกำหนดดังนี้

$$\begin{aligned} 0,0 &\geq 1,0 & 0,0 &\geq 0,1 \\ 1 &\geq 1,0 & 0 &\geq 0,1 \end{aligned}$$

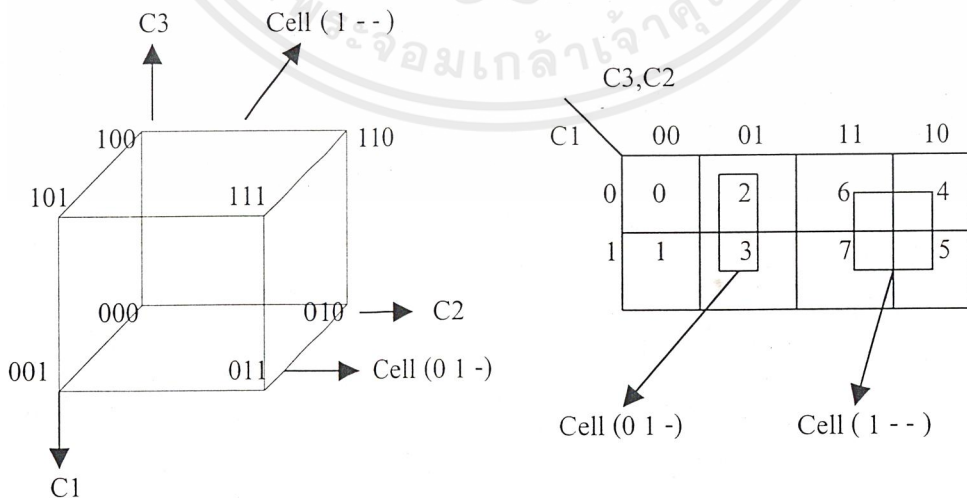
ความสัมพันธ์ \geq อาจจะถูกขยายไปถึง CELL เช่น ถ้า $X = (X_n X_{n-1} \dots X_1)$ และ $Y = (Y_n Y_{n-1} \dots Y_1)$ เป็น CELL และ $X_i \geq Y_i$ สำหรับ $i = 1, \dots, n$ เราจะสามารถบอกได้ว่า $X \supseteq Y$ หรือ X บรรจุใน Y เขียนอยู่ในรูปทั่วไปได้ว่า

$$C^n = \{ (C_n C_{n-1} \dots C_1) \mid C_i \in C \}$$

ตัวอย่าง

สมมุติ $(X) = (11-1)$ และ $(Y) = (1101)$ เป็นสมาชิกของ CELULAR 4-CUBES ดังนั้น $X \supseteq Y$ เพราะว่า $1 \geq 1, 1 \geq 1, - \geq 0, 1 \geq 1$

สังเกตได้ว่าความสัมพันธ์ของเครื่องหมายนั้นไม่ใช่ทางด้านคณิตศาสตร์ แต่มีความหมายว่าเป็นความสัมพันธ์ทางลำดับ $(C^4, \supseteq) (C^n, \supseteq)$



รูปที่ 3.1 รูป Cellular 4-Cubes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การแปลงเป็นเลขฐานสิบ (The decimal transform)

พิจารณามินเทอม $V = (V_{n-1} \dots V_1)$ N-CUBLE (C^n, \supseteq)

$$\delta(V) = \sum_{i=1}^n V_i * 2^{i-1}$$

เมื่อ V_i เป็นค่า 0 หรือ 1

ในการทำงานเดียวกัน ในแต่ละ CELL ของ N-CUBE (C^n, \subseteq) จะแสดงว่าเป็นเลขฐานสิบได้
ดังนี้

$$D(c) = (\delta(\min(c)), \delta(\max(c)))$$

ตัวอย่าง กำหนดค่ามินเทอม $V = (1001)$ ทำเป็นเลขฐานสิบ คือ

$$\begin{aligned} \delta(V) &= (1*2^3) + (0*2^2) + (0*2^1) + (1*2^0) \\ &= (9)_{10} \end{aligned}$$

ตัวอย่าง จงหาค่าของเลขฐานสิบที่แทน CELL $C = (10-1)$

$$\max(c) = (1001) \quad \delta(\max(c)) = 11$$

$$\min(c) = (1001) \quad \delta(\min(c)) = 9$$

ดังนั้น $D(c) = (9, 11)$

ถ้ากำหนดให้

$$V = \delta(V)$$

การแปลงค่าเป็นเลขฐานสิบใน (2) นั้น นำมาแทนค่าลงใน (3) อาจเขียนใหม่ได้ว่า

$$D(c) = (I, J)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} \text{เมื่อ } I &= \delta(\min(c)) \\ J &= \delta(\max(c)) \end{aligned}$$

ตัวอย่าง จงหามินเทอมของ CELL $C = (0-1-01)$ และหาค่าของการแปลงเลขฐานสิบ

$$V_1 = \min(c) = (001001) = V_1 = 9 = I$$

$$V_2 = \min(c) = (001101) = V_2 = 13$$

$$V_3 = \min(c) = (011001) = V_3 = 25$$

$$V_4 = \min(c) = (011101) = V_4 = 29 = J$$

ดังนั้นการแปลงเลขฐานสิบคือ

$$D(c) = (I, J) = (9, 29)$$

สังเกตเห็นได้ว่า $9 < 13 < 25 < 29$

ดังนั้นเราเขียนอยู่ในรูปตัวแปรว่า

$$I < V_i < J$$

สำหรับทุกๆ VERTEX V_i ของ CELL C ใน n -CUBE c (C^n, \subseteq)

3.2 คุณสมบัติของการบรรจุ (Containment properties)

กำหนดให้ตัวอักษรตัวแปรใหญ่เป็นการแสดงแทนค่าของเลขฐานสิบและตัวอักษรตัวเล็กแทนค่าของ Component ตั้งแต่ 1, n คุณสมบัติบางประการของการแสดงแทนค่าของเลขฐานสิบ CELLS ของ n -CUBE เป็นประโยชน์ในการกำหนดความสัมพันธ์ระหว่าง CELLS และมินเทอมได้ ซึ่งคุณสมบัติต่างๆ เหล่านี้จะถูกนำไปใช้ในอัลกอริทึมที่จะลดรูปตัวแปรให้ต่ำสุด โดยไม่มีการซ้ำซ้อนลอจิก AND ของค่าตัวแปรบูลีนนั้นเป็นการปฏิบัติการของระหว่าง 2 มินเทอมที่มีไม่ขึ้นต่อกันจะได้มินเทอมอื่นขึ้นมา ดังนั้น

$$V_i * V_j = V_k$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ตามการแปลงค่าเลขฐานสิบของ V_i และ V_j จะต้องถูกแปลงกลับไปเป็นส่วนประกอบของเลขไบนารีเพื่อที่จะกระทำการปฏิบัติทาง AND ได้

ตัวอย่าง กำหนดให้สองมินเทอม ของ 5-CUBE เป็น

$$V_1 = (10011), V_2 = (10101)$$

ทำการ AND จะได้

$$\begin{aligned} V_1 \cdot V_2 &= (10011) * (10101) \\ &= (1*1, 0*0, 0*1, 1*1) \\ &= (10001) \end{aligned}$$

หรือจะเขียนได้ว่า $V_i * V_j = V_k$

3.2.1 อัลกอริทึมของการบรรจุ (Containment algorithm)

ถ้ามินเทอม 2 มินเทอม V_1 และ V_2 ของ n -CUBE และถ้าความสัมพันธ์ปรากฏ เป็น

$$V_1 * V_2 = V_1$$

จะกล่าวได้ว่า V_1 ถูกสัมพันธ์กับ V_2 เขียนเป็น

$$V_2 \text{ -----} > V_1$$

หมายความว่า V_1 เป็น CONTAINED ใน V_2 ในการแปลงฐานสิบจะได้

$$V_1 * V_2 = V_1$$

$$V_2 \text{ -----} > V_1$$

การปฏิบัติการของ AND จะถูกใช้แสดงการบรรจุ ซึ่งจะเป็นสิ่งที่จำเป็นในการหาค่าในโปรแกรม ในการหาค่าต่ำสุดของอัลกอริทึมต่อไปจะเป็นทฤษฎีบทที่จะเป็นส่วนประกอบของการออกแบบอัลกอริทึม

วิธีที่ 1

สำหรับทุกๆ CELL C ของ n-CUBE (C^n, \subseteq), $\max(c) \rightarrow \min(c)$ หรือ $I * J = 1$

เมื่อ

$$D(c) = (I, J)$$

วิธีนี้เป็นการคำนวณหาคู่ของมินเทอมของบูลีนฟังก์ชันที่จะสามารถฟอร์มตัวกันเป็น CELL ได้

ตัวอย่าง กำหนดให้ 2 มินเทอม 10 และ 14 ของลำดับ ที่มี 4 ลำดับ คำนวณว่าจะจะเป็น CELL ของ (C, \supseteq) หรือไม่จาก

$$I = 10, \quad i = (1010)$$

$$J = 14, \quad j = (1110)$$

$$I * J, \quad i * j = (1010) * (1110) \\ = (1010) = i$$

ดังนั้น $I * J = I, J \rightarrow I$ (โดยความสัมพันธ์ J จะสัมพันธ์กับ I) ดังนั้น (10,14) เป็น CELL ของ (10) เพราะมินเทอม คือ (1010) และมินเทอม คือ (1110), CELL(I, J) ถูกแทนด้วย (1-10)

วิธีที่ 2

ถ้า VERTEX V จะถูกบรรจุใน CELL C จะถูกแสดงเป็น

$$C \supseteq V$$

$$(I, J) \supseteq V$$

และถ้า $\max(c) \rightarrow V \rightarrow \min(c)$

$$\text{หรือ } I * V = I, \quad V * J = V$$

วิธีนี้เป็นการใช้ในการคำนวณหาถ้ามินเทอมใดๆ ของ n-CUBE เป็นการถูกบรรจุใน CELL (I, J) หนึ่งใน CELL นี้จะถูกแทนค่าด้วยการแสดงเป็นกลุ่มของมินเทอม

ตัวอย่าง จงหาค่ามินเทอม m_{11} ถูกคลุม (COVERED) โดย CELL (10,15) มินเทอมเขียนอยู่ในรูปฐานสิบและไบนารีได้เป็น

$$I = 10, \quad i = (1010)$$

$$V = 11, \quad v = (1011)$$

$$J = 15, \quad j = (1111)$$

เพราะว่า

$$I * V = 10 * 11 = 10 = I$$

$$V * J = 11 * 15 = 11 = V$$

ดังนั้นมินเทอม V ถูกบรรจุใน CELL (I,J) เขียนได้เป็น

$$(10,15) \supseteq 11$$

วิธีที่ 3

$$\text{ถ้า } \text{Min}(C2) \text{ ----} > \text{Min}(C1)$$

$$\text{Max}(C1) \text{ ----} > \text{Max}(C2)$$

เมื่อ

$$C1 \supseteq C2$$

ในการสังเกตในเลขฐานสิบ

$$I_1 * I_2 = I_1 \text{ และ } J_2 * J_1 = J_2$$

$$\text{เมื่อ } (I_1, J_1) \supseteq (I_2, J_2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีนี้จะให้กฎเกณฑ์สำหรับการบรรจุของ CELL หนึ่งไปอีก CELL หนึ่ง ซึ่งเป็นพื้นฐานของลอจิกในการปฏิบัติทาง AND ทำให้ดิจิทัลคอมพิวเตอร์ทำการทดสอบได้ถ้าเพียงแต่มี Product term เทอมหนึ่งเทอมใดของบูลีนฟังก์ชันบรรจุเทอมอื่นๆและวิธีนี้เป็นการใช้ปฏิบัติกับงานที่สมบูรณ์เพื่อที่จะให้สร้างตารางของการลดรูปต่ำสุดได้และรายชื่อของไพริมพลีแคนท์

ตัวอย่าง ให้แสดงว่า $(10,15) \supseteq (14,15)$ ในเลขฐานสิบและไบนารี

$$(I_1, J_1) = (10,15), C1 = (101-)$$

$$(I_2, J_2) = (14,15), C2 = (111-)$$

จาก

$$I_1 * J_1 = (10) * (14) = (1010) * (1110) = (1010) = I_1$$

$$I_2 * J_2 = (15) * (15) = (1111) * (1111) = (1111) = J_2$$

วิธีที่ 3

$$(10,15) \supseteq (14,15)$$

จากตัวอย่างนี้จะเห็นว่า CELL C1 และ C2 ทั้งสองนี้มีความแตกต่างกันของตำแหน่งเพียง 1 COMPONENT คือ CELL C1 มี ในตำแหน่งนี้ C2 มีค่าเป็น 1 ดังนั้นจาก $- > = 1, C1_C2$

วิธีที่ 4

ถ้า V1 และ V2 เป็น VERTICES ของ n-CUBE หนึ่ง และ $V2 \rightarrow V1$

ดังนั้น $\delta(V1) \leq \delta(V2)$ หรือ $V1 \leq V2$

เมื่อ \leq เป็นสัญลักษณ์ทางคณิตศาสตร์หมายถึง “น้อยกว่าหรือเท่ากับ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีนี้ใช้เพื่อจะให้ VERTEX V2 CONTAIN VERTEX V1 ซึ่งเป็นการแปลงเลขฐานสิบ
อย่างหนึ่ง

วิธีที่ 5

ถ้า C_1 และ C_2 เป็น CELLS ของ n -CUBE หนึ่ง และ $C_1 \supseteq C_2$
ดังนั้น

$$\delta(\min(c_1)) \leq \delta(\min(c_2))$$

$$\delta(\min(c_2)) \leq \delta(\min(c_1))$$

หรือสังเกตจากเลขฐานสิบถ้า

$$(I_1, J_1) \supseteq (I_2, J_2)$$

ดังนั้น $I_1 \leq I_2 \leq J_2 \leq J_1$

ตัวอย่าง จาก CELL ของ

$$C_1 = (10, 15), C_2 = (14, 15)$$

เราทราบว่า $(10, 15) \supseteq (14, 15)$

$$10 \leq 14 \leq 15 \leq 15$$

3.3 สรุปความสัมพันธ์ของวิธีการลดรูปสมการ

ในการรวบรวมความสัมพันธ์ของวิธีที่ 3, 4 และ 5 นี้ ใช้ในการทดสอบเงื่อนไข ต่อไปนี้

1) VERTEX V ใดๆ ถูกคลุม โดย CELL(I,J) ถ้าเพียงแต่

$$I \leq V \leq J$$

$$I * V = I$$

และ $V * J = V$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) CELL(I2 ,J2) ถูกครอบคลุมโดย CELL (I1,J1) อีก CELL หนึ่ง ได้ถ้า

$$I1 \leq I2 \leq J2 \leq J1$$

$$I1 * I2 = I1 , J2 * J1 = J2$$

ในสมการที่อยู่ในข้อ 1 และ 2 จะถูกใช้ในการคำนวณหา PRIME IMPLICANTS ดังตัวอย่างต่อไปนี้

ตัวอย่าง

พิจารณาบูลีนฟังก์ชัน

$$F(W,X,Y,T) = 1, 3, 5, 7$$

จะสังเกตเห็นว่าเป็นฟังก์ชันที่มี VERTICES ของ 4 – CUBE โดยให้

$$V1 = 1, V2 = 3, V3 = 5, V4 = 7$$

จะใช้วิธีที่ 1 ในการหา CELL ต่างๆ โดยทดสอบกันเป็นคู่ๆ ไปในแต่ละคู่ของ VERTICES ต่อไปนี้

$$1 * 7 = (0001) * (0111) = (0001) = 1$$

(1,7) เป็น CELL (0 -- 1)

$$1 * 5 = (0001) * (0101) = (0001) = 1$$

(1,5) เป็น CELL (0 -01)

$$1 * 3 = (0001) * (0011) = (0001) = 1$$

(1,3) เป็น CELL (00 -1)

$$3 * 7 = (0011) * (0111) = (0001) = 3$$

(3, 7) เป็น CELL (0 -11)

$$3 * 5 = (0011) * (0101) = (0001) = 1$$

(3, 5) ไม่เป็น CELL

$$5 * 7 = (0101) * (0111) = (0101) = 5$$

(5,7) เป็น CELL (01- 1)

หรือ CELLS อาจจะถูกหาได้จากการใช้สมการในข้อ 2 ทดสอบหาการบรรจุดังต่อไปนี้
เซ็่คคู่ของ (1,7) และ (1,5)

$$1 <= 1 <= 5 <= 7$$

$$1 * 1 = 1$$

$$5 * 7 = 5$$

ดังนั้น (1,7) ---> (1,5) และยังสามารถแสดงได้อีกอย่างว่า (1,7) นั้นยัง CONTAIN คู่ของ (1,3) (3,7) และ (5,7) อีกด้วย ดังนั้น CELL(1, 7) CONTAIN MINTERM ทั้งหมดของฟังก์ชัน ถ้าเมื่อมันคลุมได้หมดอย่างสมบูรณ์ก็จะมีเพียง PRIME IMPLICANT ตัวเดียวดังนั้น

$$\begin{aligned} F(W,X,Y,T) &= (1,7) = (0- -1) \\ &= W'Z \end{aligned}$$

3.4 อัลกอริทึมของการลดรูปสมการบูลีน (Minimization algorithm)

จากวิธีการที่ได้ศึกษามาตั้งแต่ต้น ทำให้เราสามารถเขียนโปรแกรมได้เป็นขั้นตอนโดยจะมีการอธิบายโดยย่อของแต่ละขั้นตอน ซึ่งเป็นหลักการใหญ่ๆ ในการเขียนโปรแกรมให้ลดรูปสมการบูลีนได้เป็นผลสำเร็จ ดังต่อไปนี้

- 1) ขั้นที่ 1 ทำการกำหนดลำดับของบูลีน ฟังก์ชันที่เป็นค่าของมินเทอม โดยมีการจัดเรียงลำดับของเทอมที่มีค่าเพิ่มขึ้น จาก 0,1,2,4...(2_n-1 หมายถึงเลขจำนวนเต็มบวก)
- 2) ขั้นที่ 2 เลือกคู่ (II,JJ) เมื่อ II เป็นเทอมที่มีขนาดต่ำสุดและเทอม JJ เป็นเทอมที่มีขนาดสูงสุด
- 3) ขั้นที่ 3 เซ็่คคู่ (II,JJ) ว่าจะฟอร์มตัวเป็น Cell ได้หรือไม่ โดยใช้สมการ

$$II * JJ = II$$

ถ้าคู่ที่กำลังพิจารณานั้นฟอร์มเป็น Cell ก็จะผ่านไปยังขั้นที่ 4 และถ้าไม่สามารถฟอร์มตัวเป็น Cell ได้ก็ให้มีการเลือกค่าสูงสุดถัดเข้ามาโดยเก็บค่าเป็น JJ แล้วก็กระทำการเช็คดูว่าฟอร์มตัวเป็น Cell ได้หรือไม่อีกครั้ง

4) ขั้นที่ 4 เป็นขั้นที่ต้องหาค่าของเทอมทั้งหมดที่เกิดขึ้นจาก Cell ที่ได้จากขั้นที่ 3 และถ้าเทอมที่เกิดจาก Cell นั้นเป็นค่าที่ถูกรวมอยู่ในฟังก์ชันของมินเทอมทั้งหมด จะผ่านไปยังขั้นที่ 5 แต่ถ้าไม่ถูกรวมอยู่ในฟังก์ชันจะเป็นการย้อนกลับไปหาค่าสูงสุดเข้ามาและกลับเข้าขั้นที่ 3 ใหม่

5) ขั้นที่ 5 ทำการเช็ค Cell (II, JJ) เพื่อที่จะสังเกตว่า Cell นั้นจะถูกบรรจุภายในตารางไพรม์อิมพลิแคนท์ (ข้อสังเกต สำหรับ Cell ตัวแรกที่ผ่านอัลกอริทึมนี้ ค่าของตาราง PI นั้นจะว่าเปล่า) ถ้าไม่ถูกบรรจุจะผ่านไปยังขั้นที่ 6 แต่ถ้าบรรจุจะกลับไปเลือกค่าสูงสุดตัวถัดเข้ามาในฟังก์ชันเป็นค่า JJ และเข้าไปยังขั้นที่ 3 ใหม่

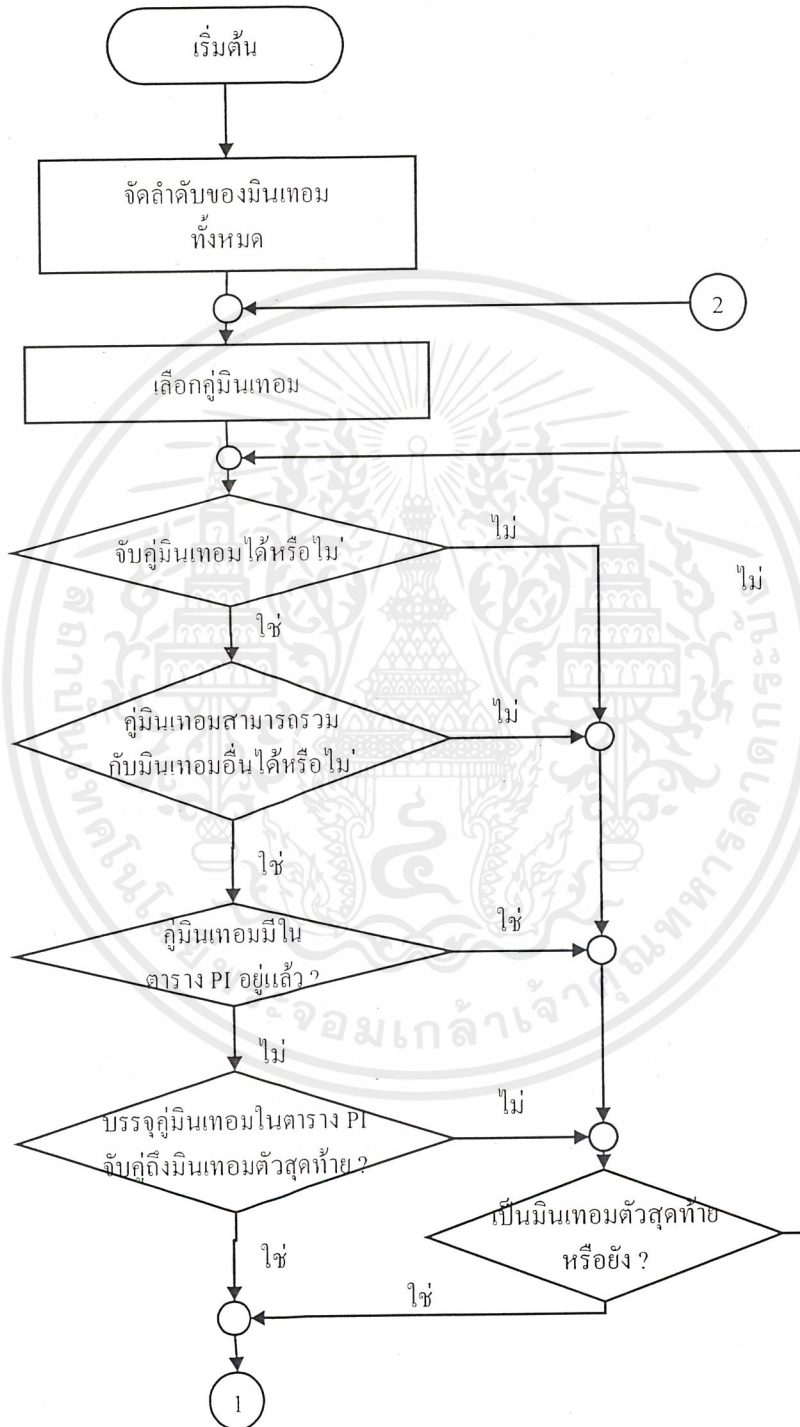
6) ขั้นที่ 6 หลังจากผ่านขั้นที่ 5 แล้ว Cell (II, JJ) จะเป็น PI ตัวใหม่และจะนำเข้าไปเก็บที่ตาราง PI และถ้าค่า II ถูกนำไปเปรียบเทียบหรือจับคู่กับเทอมอื่นๆ ที่ปรากฏอยู่ในฟังก์ชันและมีค่ามากกว่า II ทั้งหมดตัวแล้วก็จะผ่านเข้าไปปฏิบัติต่อในขั้นที่ 7 แต่ถ้าค่า II นำไปเทียบหรือจับคู่กับเทอมอื่นไม่หมดก็ต้องเลือกค่าสูงสุดตัวถัดไปในฟังก์ชันให้เป็น JJ แล้วเข้าขั้นที่ 3 ที่จุดนี้ค่าของตาราง PI จะบรรจุ PI ทั้งหมดนี้คลุมเทอมอยู่ก่อนจะเข้าขั้นที่ 7

7) ขั้นที่ 7 เช็คจำนวนของไพรม์อิมพลิแคนท์ ที่คลุม II ถ้ามีไพรม์อิมพลิแคนท์เพียงตัวเดียว ที่คลุม II ได้ ค่าของไพรม์อิมพลิแคนท์นั้นอาจจะเป็น Essential PI ได้ ดังนั้นจึงมีการทดสอบ PI ที่ได้นี้ อยู่ในตาราง EPI ก่อนแล้วหรือไม่ถ้าถูกบรรจุอยู่ก่อนแล้วจะผ่านออกไปขั้นถัดไป แต่ถ้ายังไม่ได้บรรจุอยู่ EPI ก็จะนำค่า PI ที่ได้ในขั้นนี้เข้าไปเก็บในตาราง EPI

8) ขั้นที่ 8 เช็คค่า II หรือค่าต่ำสุดของฟังก์ชัน นั้นได้ถูกแทนหมดแล้วหรือไม่ ถ้ายังให้แทนค่า II ด้วยค่าต่ำสุดตัวถัดไป แล้วนำกลับไปเข้าขั้นที่ 2 กระทำถึงขั้นที่ 7 ใหม่จนกระทั่ง II นั้นเท่ากับ JJ ให้หยุด จากขั้นนี้เราจะพบว่าสามารถหาค่าของไพรม์อิมพลิแคนท์ทั้งหมดของฟังก์ชันได้ และเลือก Essential Prime Implicant ได้ทั้งหมดด้วย

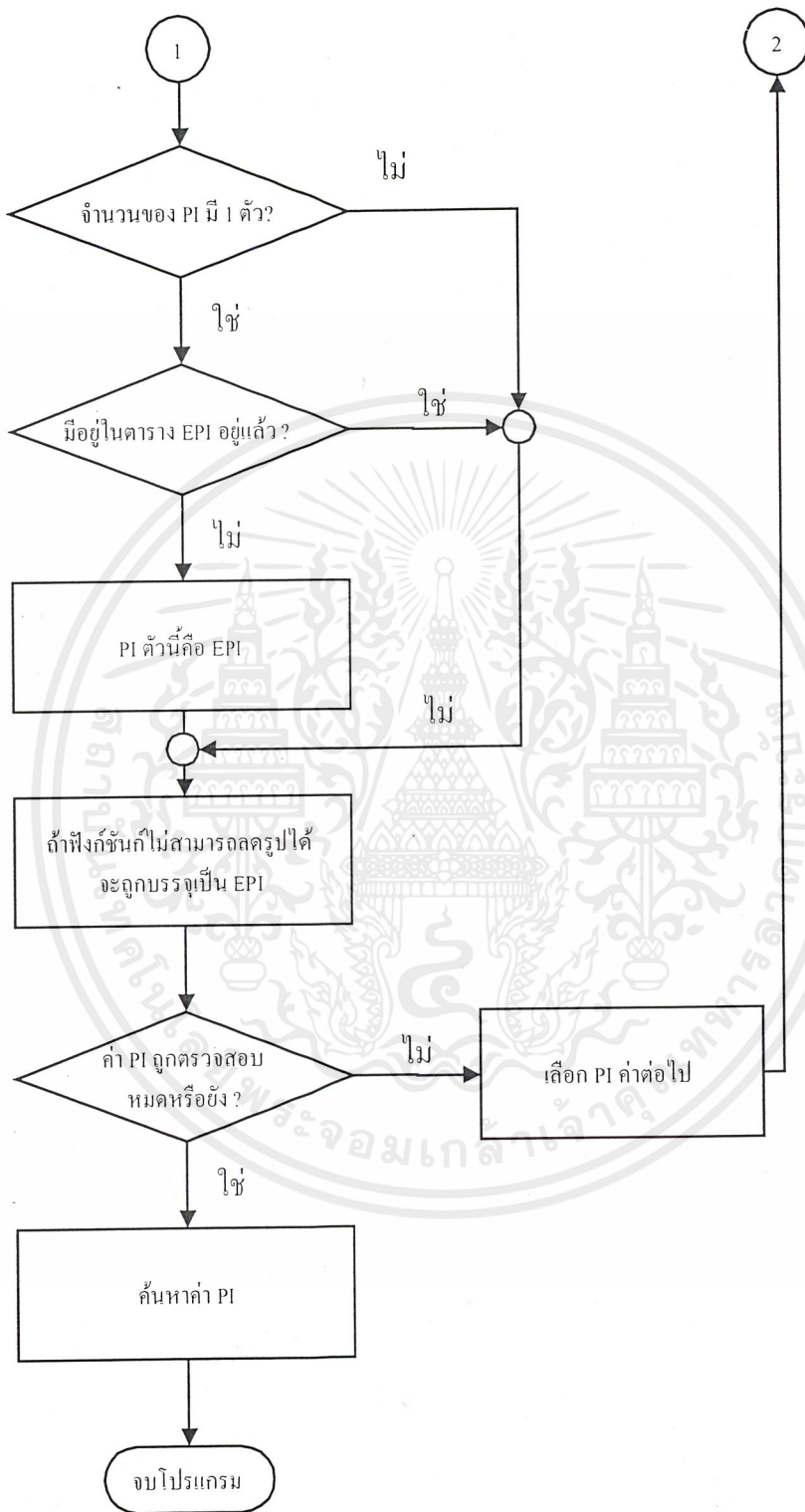
9) ขั้นที่ 9 เป็นการหามินเทอมทั้งหมดที่ไม่ได้ถูกคลุมด้วย Essential Prime Implicant ทำการเลือกชุดที่ดีที่สุดของ Prime Implicant ซึ่งคลุม PI ตัวอื่นๆ ได้มากที่สุดหรือทั้งหมดจากค่า PI ที่ได้จากการเลือกนี้จะเป็น Non Essential Prime Implicant

3.5 การแสดงโปรแกรมโดยใช้ Flowchart



รูปที่ 3.2 ขั้นตอนการลดรูปของโปรแกรม

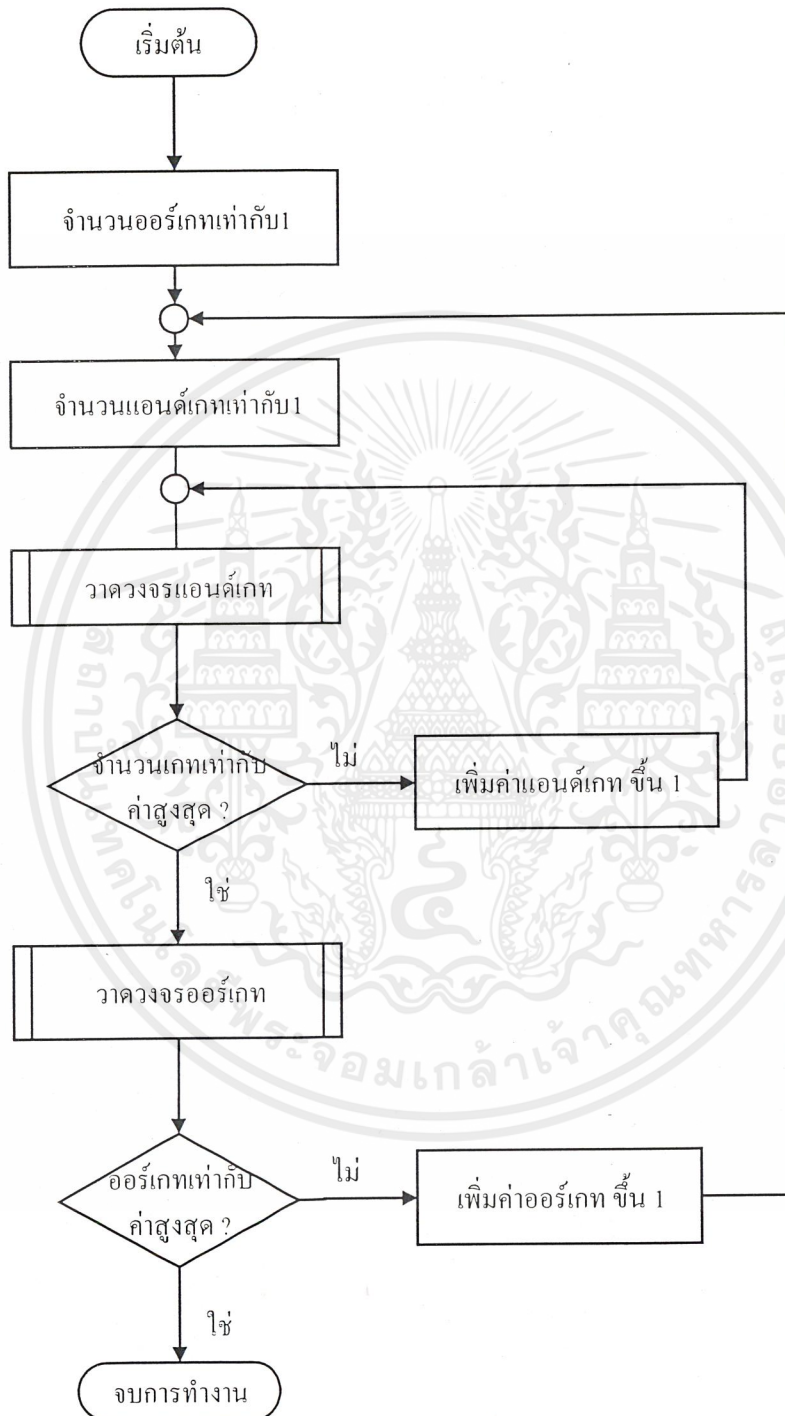
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 (ต่อ) ขั้นตอนการลสรุปของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

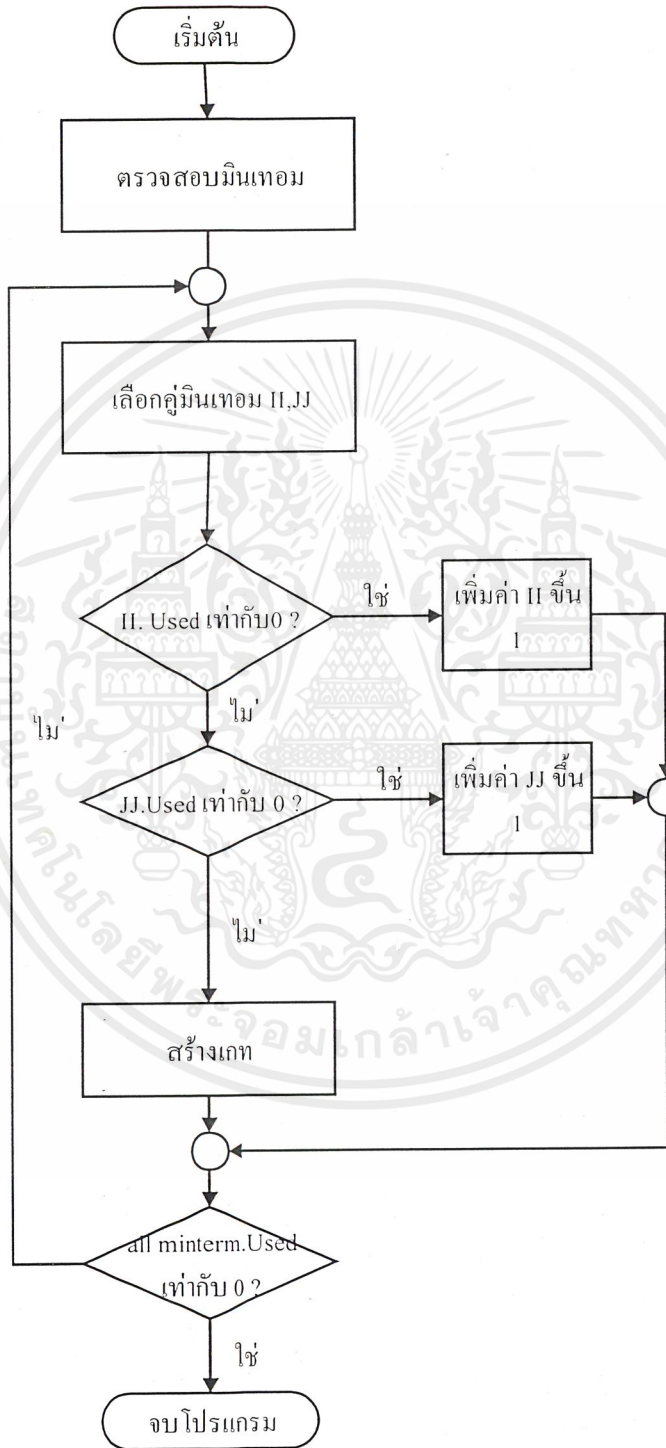
3.6 Flowchart VHDL



รูปที่ 3.3 ขั้นตอนการทำงานของโปรแกรม VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 แสดงการทำงานโดยไฟลัซาร์ตของวงจร



รูปที่ 3.4 ขั้นตอนในการวาดรูปวงจรถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองลดรูปสมการขนาด 2 ตัวแปร

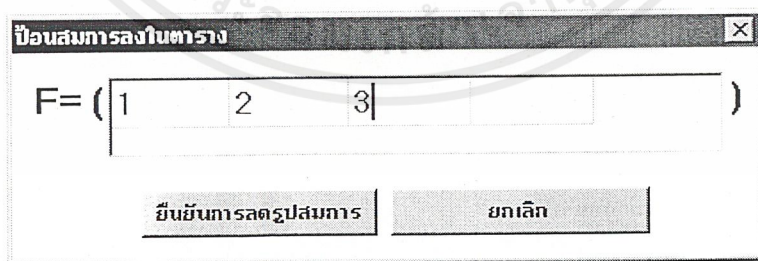
4.1.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 2 ตัวแปรดังรูปที่ 4.1



รูปที่ 4.1 หน้าต่าง “กำหนดค่าเริ่มต้น”

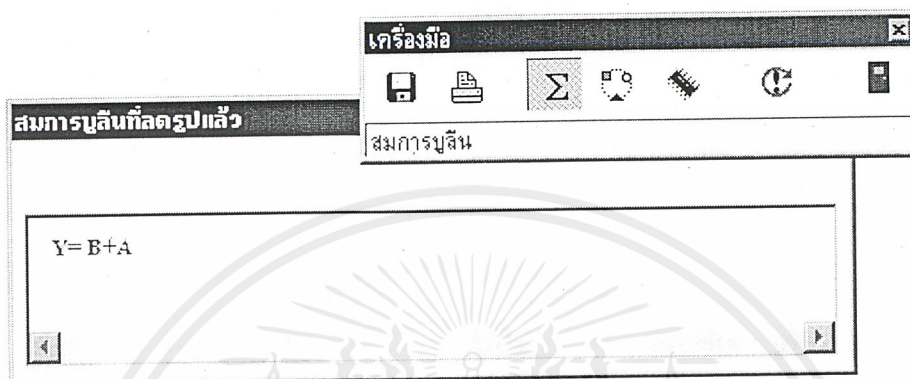
- 2) พิมพ์ตัวเลข 1, 2 และ 3 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ” ดังรูปที่ 4.2



รูปที่ 4.2 หน้าต่าง “ป้อนสมการลงในตาราง”

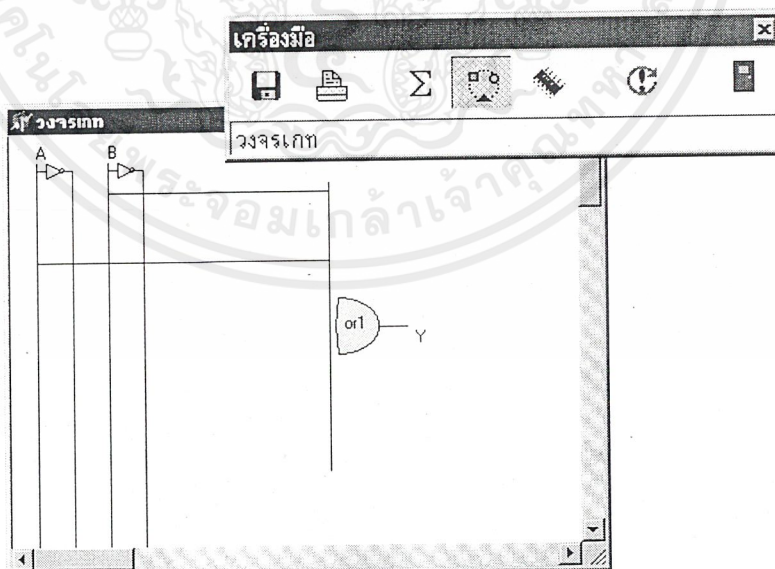
4.1.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.3



รูปที่ 4.3 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

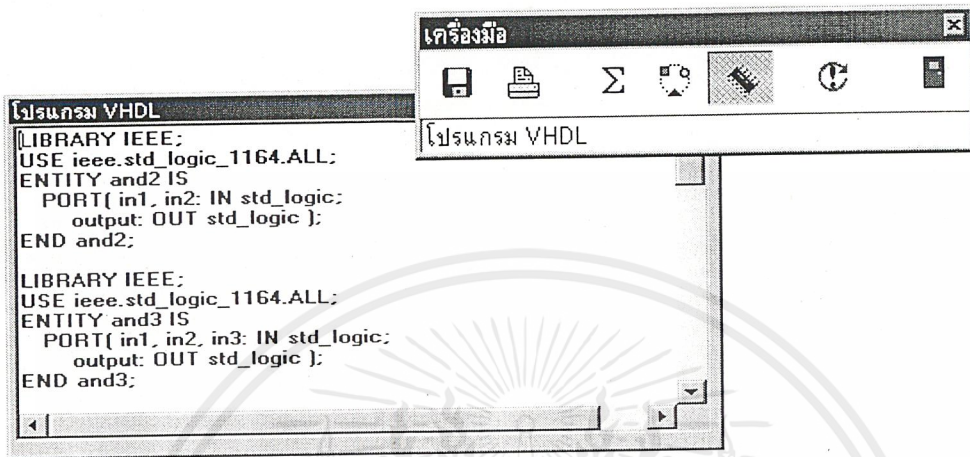
2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูก จะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.4



รูปที่ 4.4 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรมี VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.5

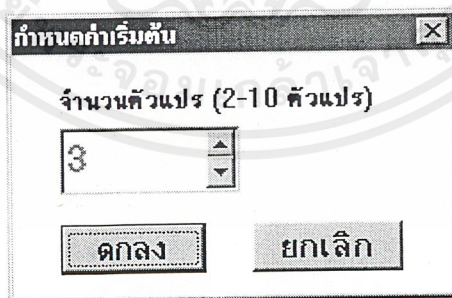


รูปที่ 4.5 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.2 การทดลองลดรูปสมการขนาด 3 ตัวแปร

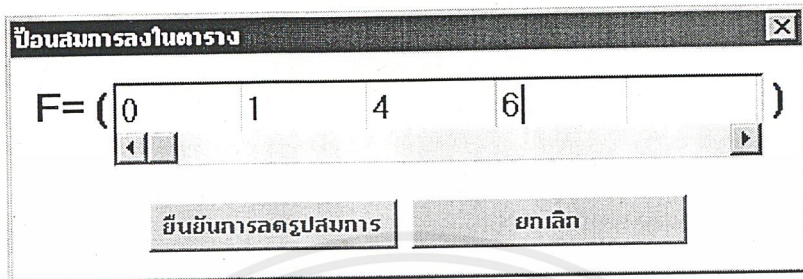
4.2.1 ลำดับขั้นการทดลอง

1) ป้อนจำนวนตัวแปรให้เท่ากับ 3 ตัวแปรดังรูปที่ 4.6



รูปที่ 4.6 หน้าต่าง “กำหนดค่าเริ่มต้น”

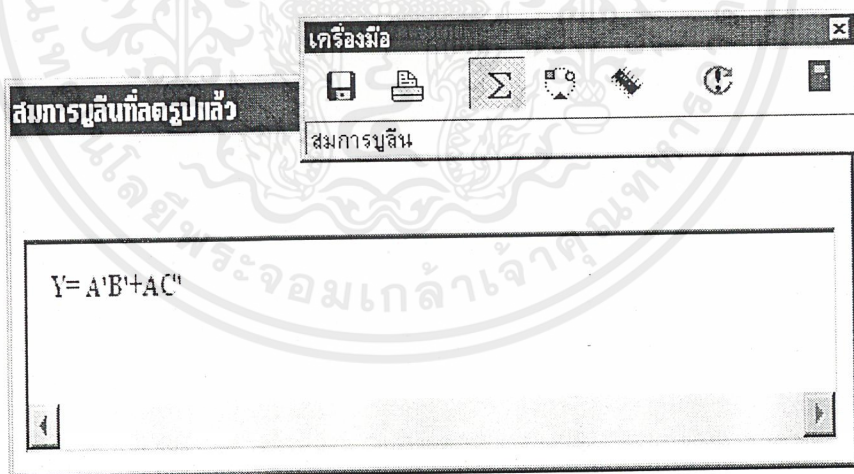
2) พิมพ์ตัวเลข 0, 1, 4 และ 6 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ”
 ดังรูปที่ 4.7



รูปที่ 4.7 หน้าต่าง “ป้อนสมการลงในตาราง”

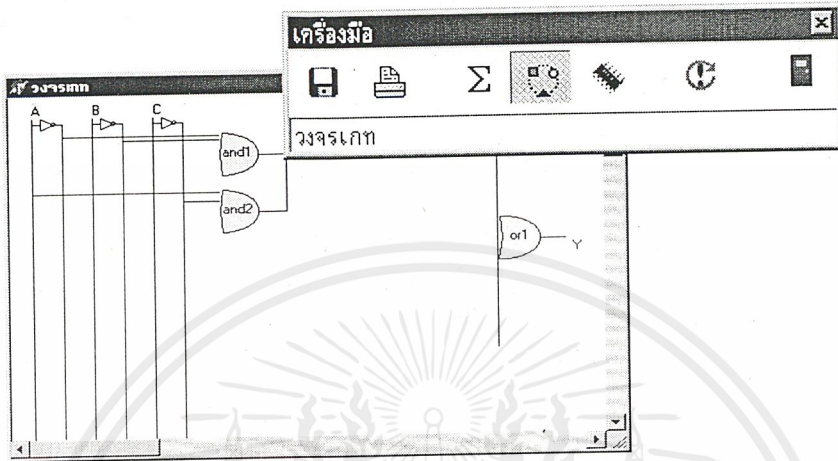
4.2.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.8



รูปที่ 4.8 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นวงจรถูก จะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.9



รูปที่ 4.9 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นวงจรถูกเพิ่ม VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.10

```

โปรแกรม VHDL
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and2 IS
  PORT( in1, in2: IN std_logic;
        output: OUT std_logic );
END and2;

LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and3 IS
  PORT( in1, in2, in3: IN std_logic;
        output: OUT std_logic );
END and3;

```

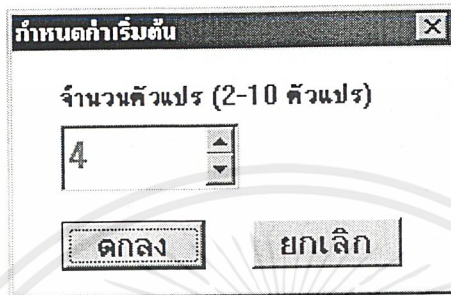
รูปที่ 4.10 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองลดรูปสมการขนาด 4 ตัวแปร

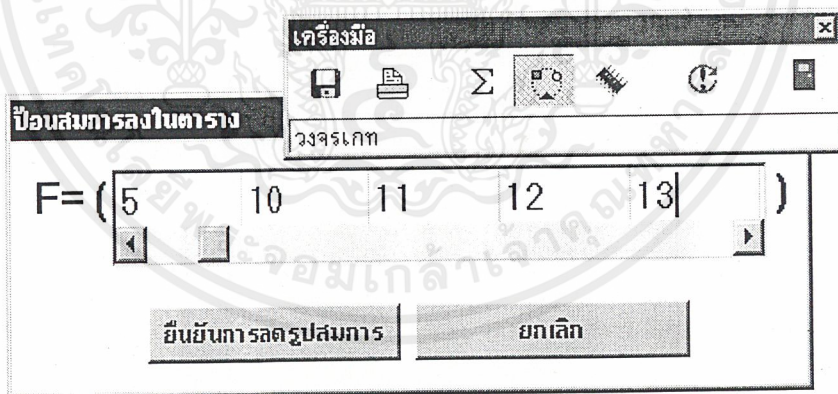
4.3.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 4 ตัวแปรดังรูปที่ 4.11



รูปที่ 4.11 หน้าต่าง “กำหนดค่าเริ่มต้น”

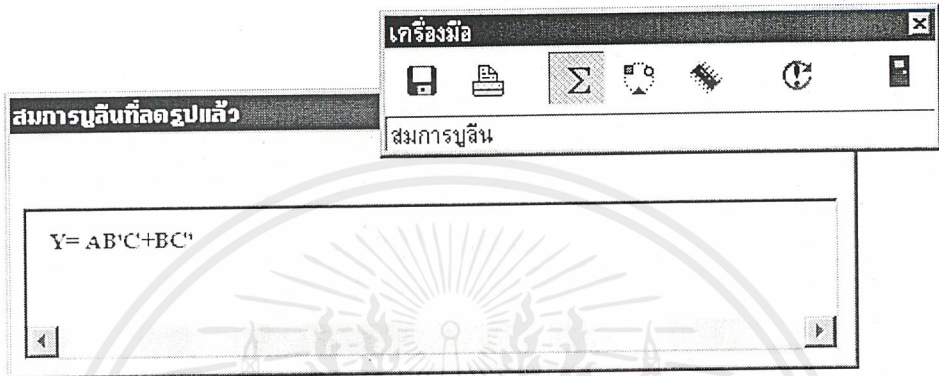
- 2) พิมพ์ตัวเลข 4, 5, 10, 11, 12, และ 13 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูป สมการ” ดังรูปที่ 4.12



รูปที่ 4.12 หน้าต่าง “ป้อนสมการลงในตาราง”

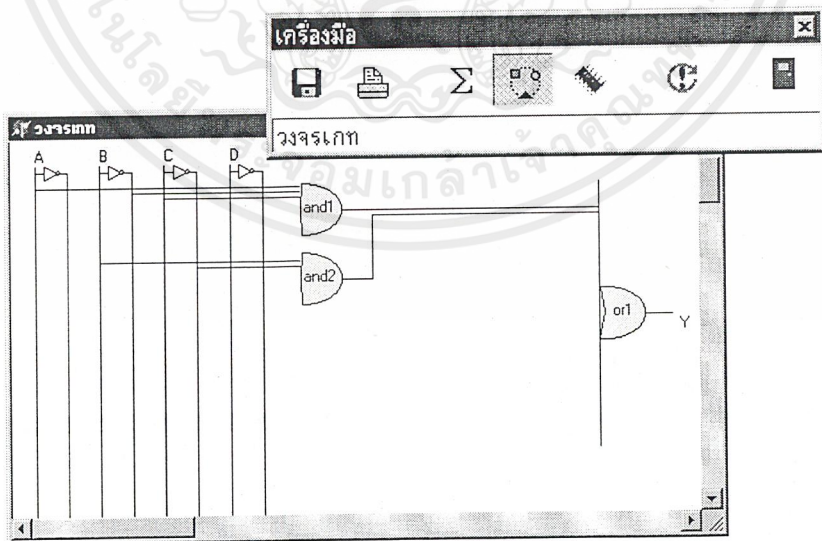
4.3.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.13



รูปที่ 4.13 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

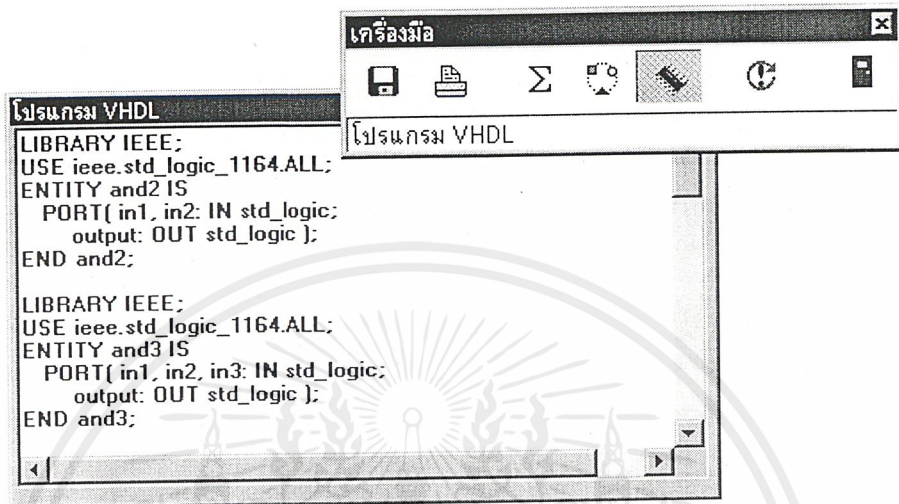
2) กดที่ปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นวงจรถูกจะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.14



รูปที่ 4.14 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรมี VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.15

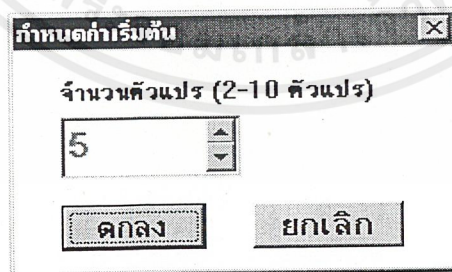


รูปที่ 4.15 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.4 การทดลองลดรูปสมการขนาด 5 ตัวแปร

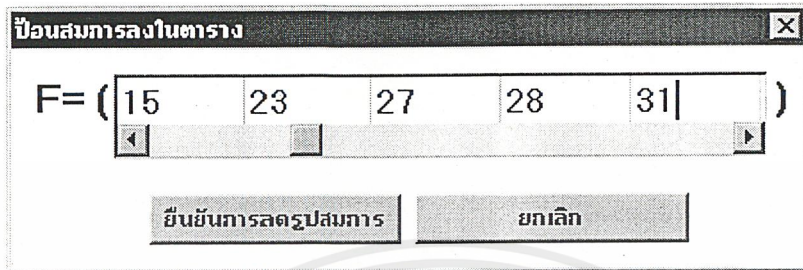
4.4.1 ลำดับขั้นการทดลอง

1) ป้อนจำนวนตัวแปรให้เท่ากับ 5 ตัวแปรดังรูปที่ 4.16



รูปที่ 4.16 หน้าต่าง “กำหนดค่าเริ่มต้น”

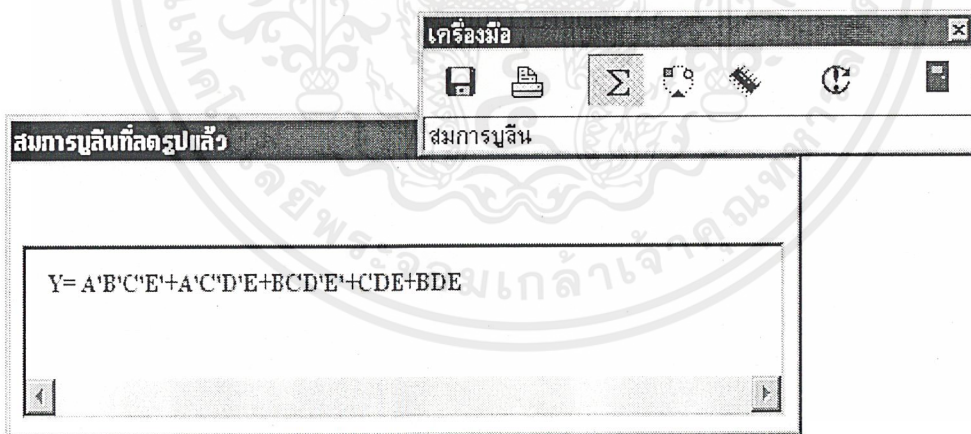
2) พิมพ์ตัวเลข 0, 1, 2, 7, 9, 11, 12, 15, 23, 27, 28 และ 31 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ” ดังรูปที่ 4.17



รูปที่ 4.17 หน้าต่าง “ป้อนสมการลงในตาราง”

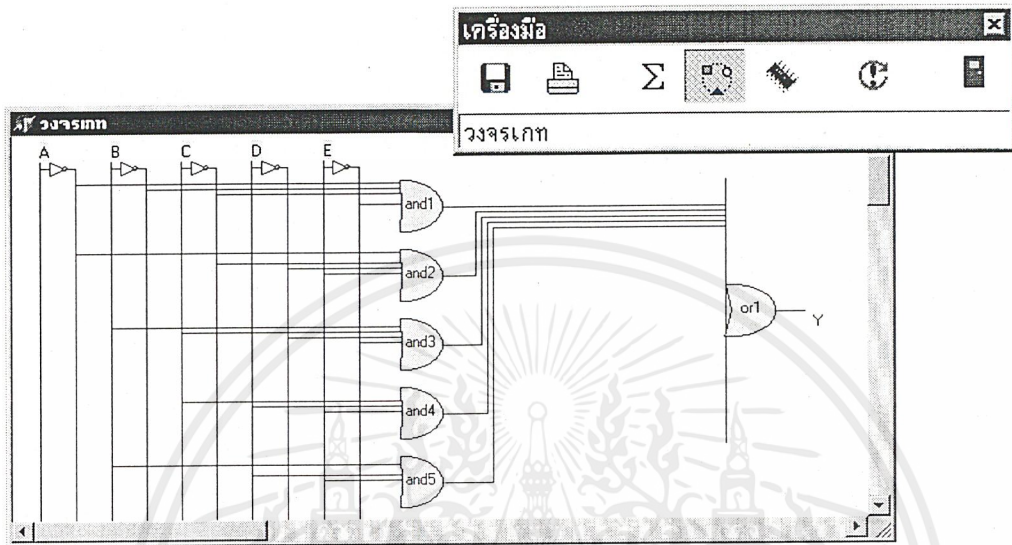
4.4.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.18



รูปที่ 4.18 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูก จะปรากฏหน้าต่าง “วงจรถูก”
 ดังรูปที่ 14.9



รูปที่ 4.19 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูกเพิ่ม VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.20

```

LIBRARY IEEE;
USE ieee.std_logic_1164.A;
ENTITY and2 IS
  PORT( in1, in2: IN std_logic;
        output: OUT std_logic );
END and2;

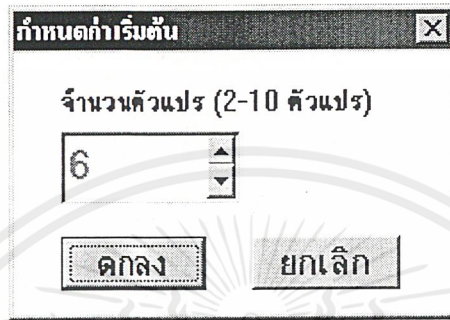
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and3 IS
  PORT( in1, in2, in3: IN std_logic;
        output: OUT std_logic );
END and3;
  
```

รูปที่ 4.20 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.5 การทดลองสตรูปสมการขนาด 6 ตัวแปร

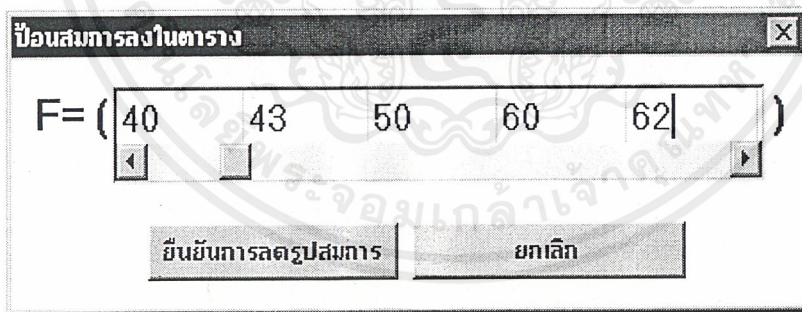
4.5.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 6 ตัวแปรดังรูปที่ 4.21



รูปที่ 4.21 หน้าต่าง “กำหนดค่าเริ่มต้น”

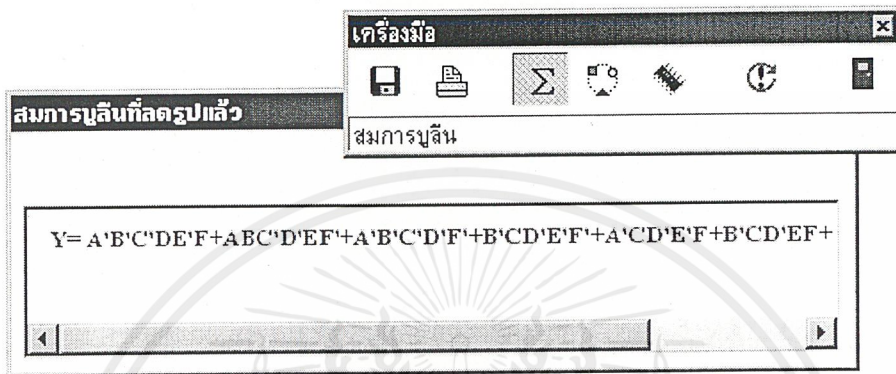
- 2) พิมพ์ตัวเลข 0, 2, 5, 8, 9, 11, 25, 28, 40, 43, 50, 60 และ 62 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ” ดังรูปที่ 4.22



รูปที่ 4.22 หน้าต่าง “ป้อนสมการลงในตาราง”

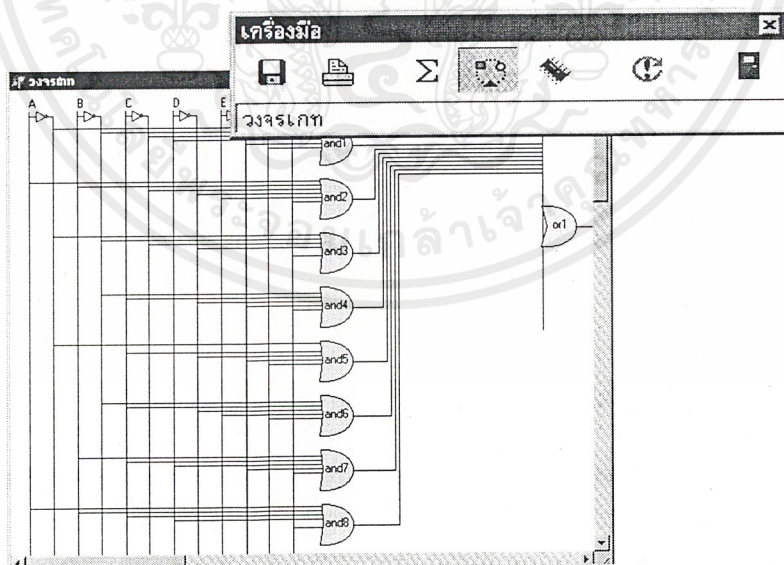
4.5.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.23



รูปที่ 4.23 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

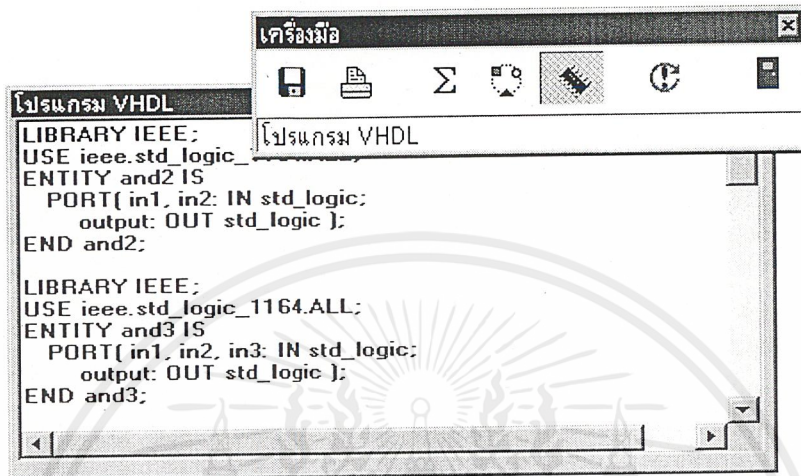
2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูกจะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.24



รูปที่ 4.24 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรมุมเพิ่ม VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.25

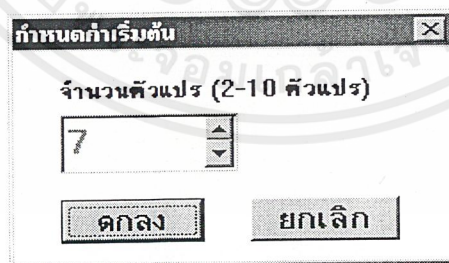


รูปที่ 4.25 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.6 การทดลองลดรูปสมการขนาด 7 ตัวแปร

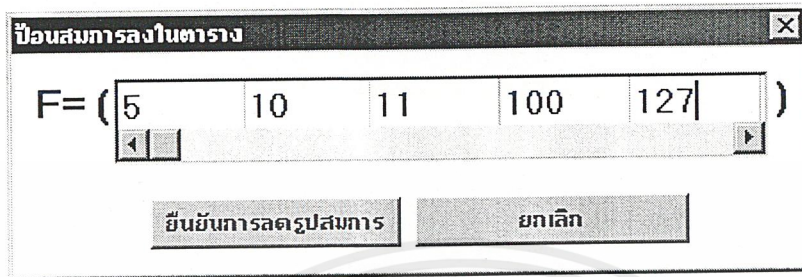
4.6.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 7 ตัวแปรดังรูปที่ 4.26



รูปที่ 4.26 หน้าต่าง “กำหนดค่าเริ่มต้น”

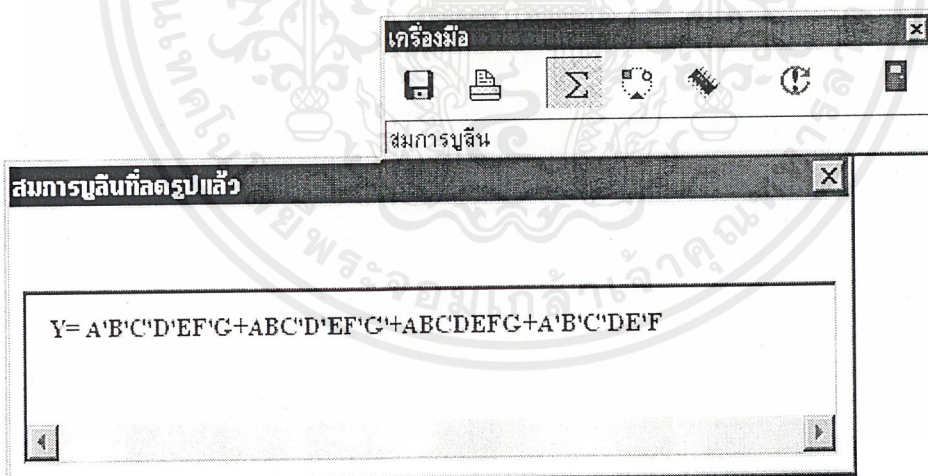
2) พิมพ์ตัวเลข 5, 10, 11, 100 และ 127 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูป สมการ” ดังรูปที่ 4.27



รูปที่ 4.27 หน้าต่าง “ป้อนสมการลงในตาราง”

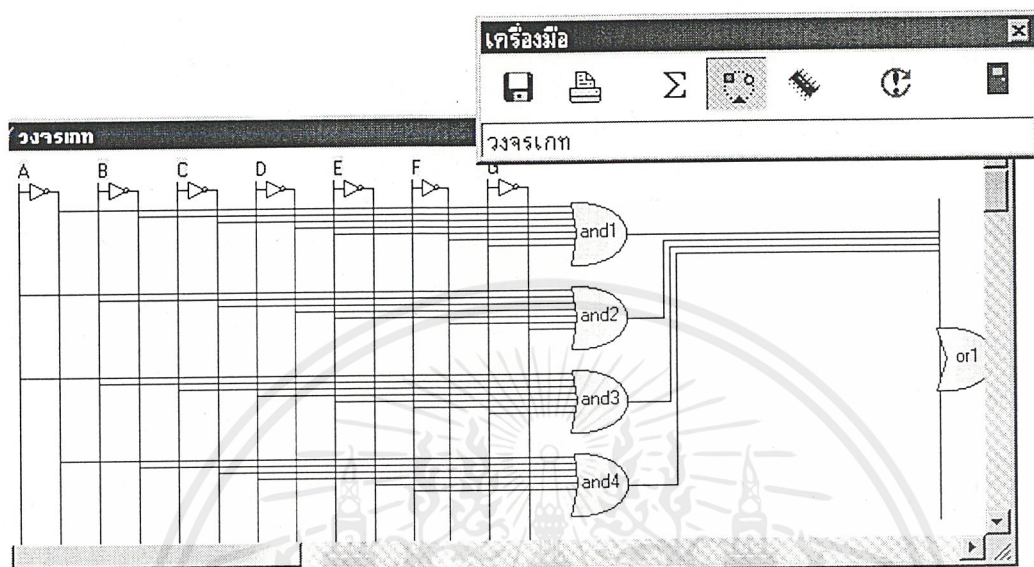
4.6.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.28



รูปที่ 4.28 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นวงจรถูก จะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.29



รูปที่ 4.29 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นวงจรถูกเพิ่ม VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.30

```

โปรแกรม VHDL
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and2 IS
    PORT( in1, in2: IN std_logic;
          output: OUT std_logic );
END and2;

LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and3 IS
    PORT( in1, in2, in3: IN std_logic;
          output: OUT std_logic );
END and3;
    
```

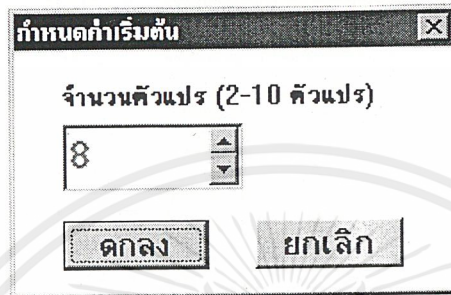
รูปที่ 4.30 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การทดลองลดรูปสมการขนาด 8 ตัวแปร

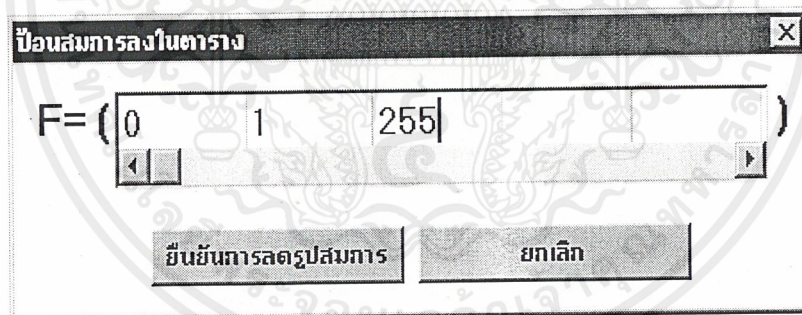
4.7.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 8 ตัวแปรดังรูปที่ 4.31



รูปที่ 4.31 หน้าต่าง “กำหนดค่าเริ่มต้น”

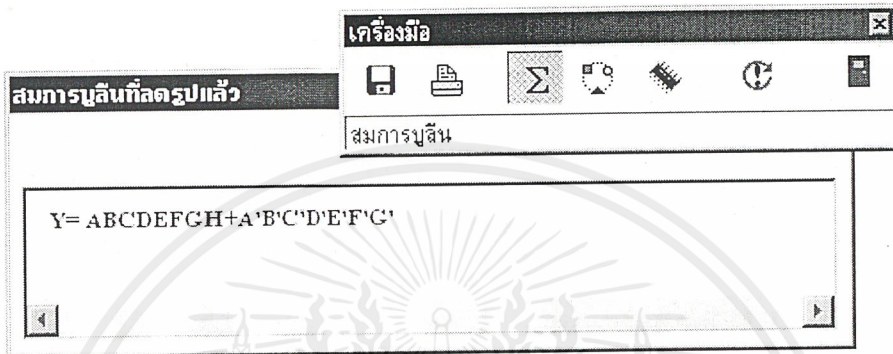
- 2) พิมพ์ตัวเลข 0, 1 และ 255 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ” ดังรูปที่ 4.32



รูปที่ 4.32 หน้าต่าง “ป้อนสมการลงในตาราง”

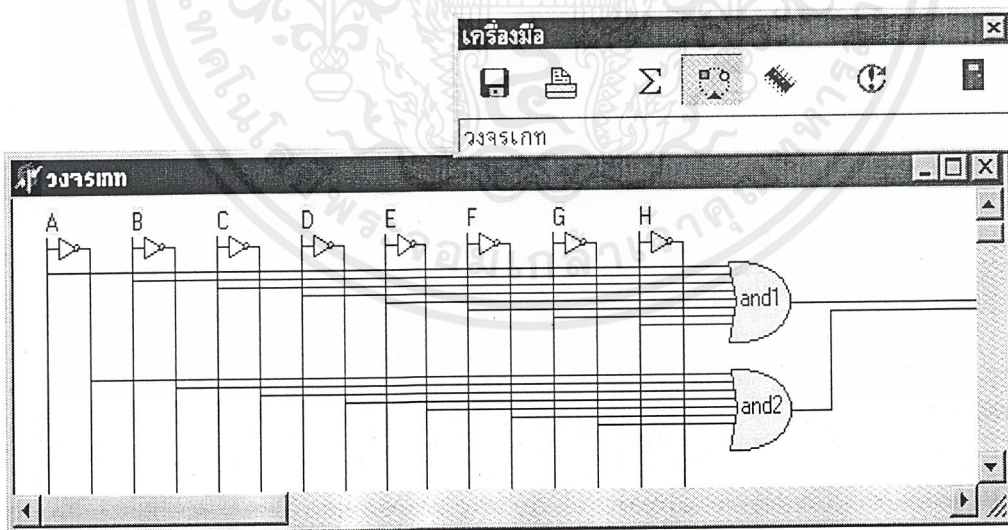
4.7.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.33



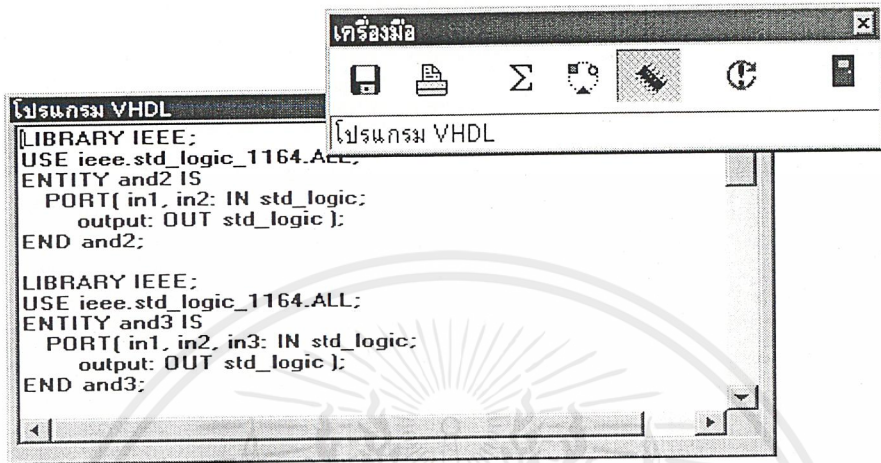
รูปที่ 4.33 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูกจะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.34



รูปที่ 4.34 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรมี VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.35

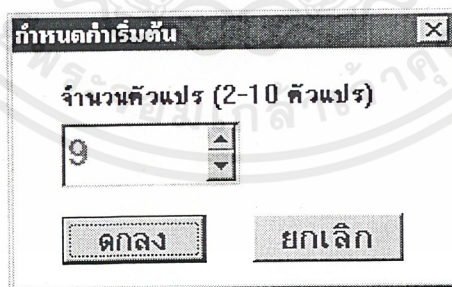


รูปที่ 4.35 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.8 การทดลองลดรูปสมการขนาด 9 ตัวแปร

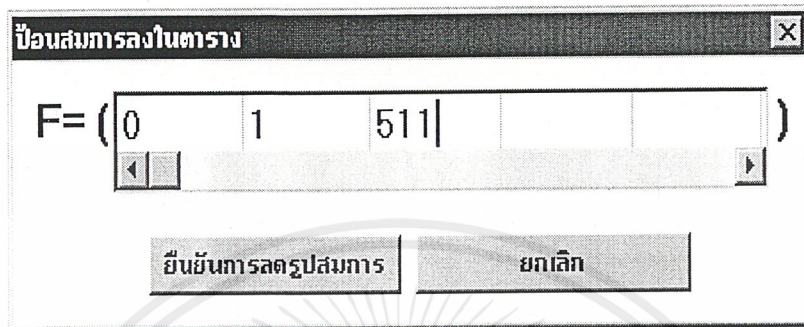
4.8.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 9 ตัวแปรดังรูปที่ 4.36



รูปที่ 4.36 หน้าต่าง “กำหนดค่าเริ่มต้น”

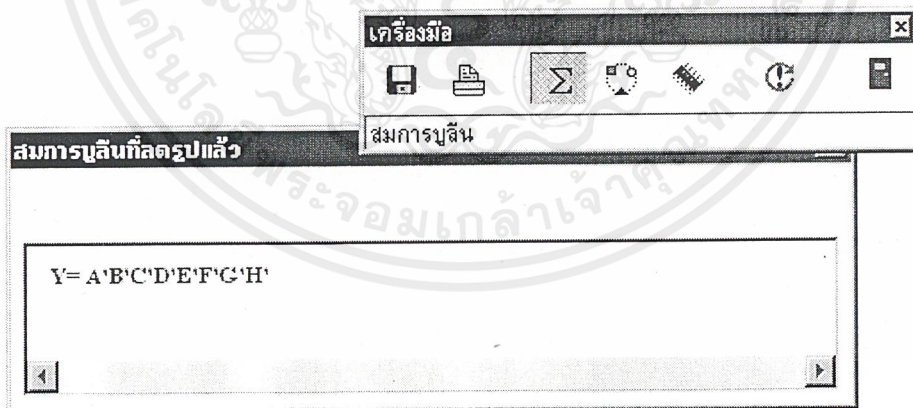
2) พิมพ์ตัวเลข 0 และ 1 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ”
 ดังรูปที่ 4.37



รูปที่ 4.37 หน้าต่าง “ป้อนสมการลงในตาราง”

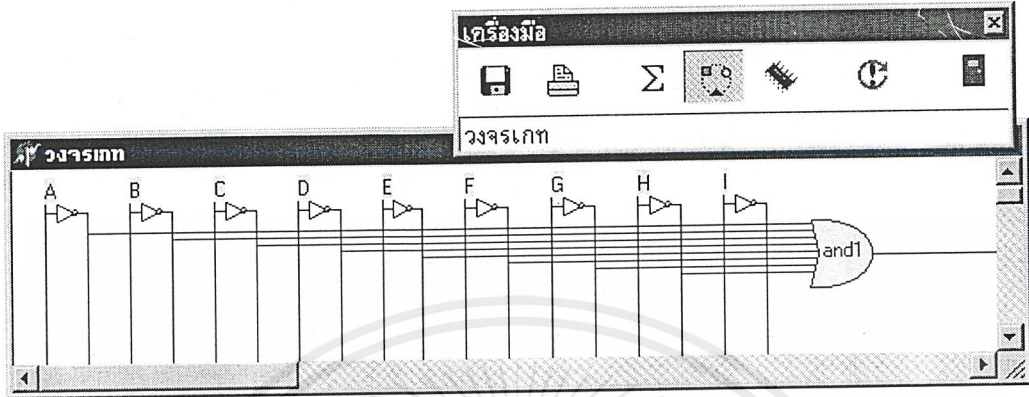
4.8.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่ให้แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.38



รูปที่ 4.38 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูก จะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.39



รูปที่ 4.39 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูกเพิ่ม VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.40

```

[PROGRAM VHDL]
[PROGRAM VHDL]
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and2 IS
  PORT( in1, in2: IN std_logic;
        output: OUT std_logic );
END and2;

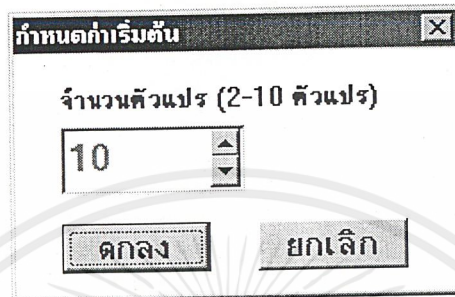
LIBRARY IEEE;
USE ieee.std_logic_1164.ALL;
ENTITY and3 IS
  PORT( in1, in2, in3: IN std_logic;
        output: OUT std_logic );
END and3;
  
```

รูปที่ 4.40 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.9 การทดลองลดรูปสมการขนาด 10 ตัวแปร

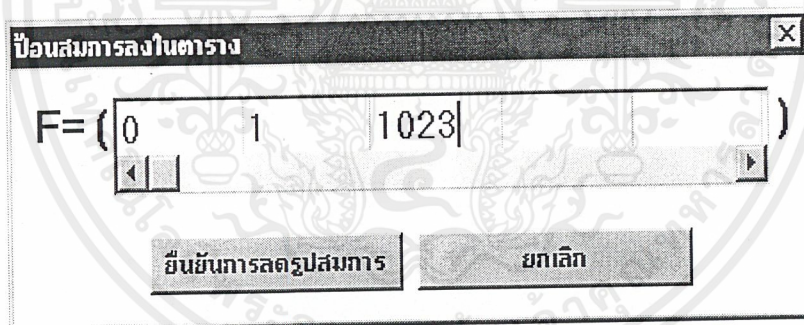
4.9.1 ลำดับขั้นการทดลอง

- 1) ป้อนจำนวนตัวแปรให้เท่ากับ 10 ตัวแปรดังรูปที่ 4.41



รูปที่ 4.41 หน้าต่าง “กำหนดค่าเริ่มต้น”

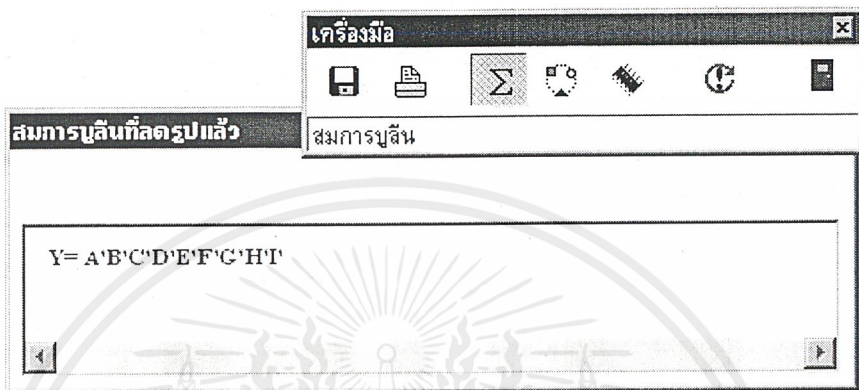
- 2) พิมพ์ตัวเลข 0,1 และ 1023 ลงในตาราง แล้วกดปุ่ม “ยืนยันการลดรูปสมการ” ดังรูปที่ 4.42



รูปที่ 4.42 หน้าต่าง “ป้อนสมการลงในตาราง”

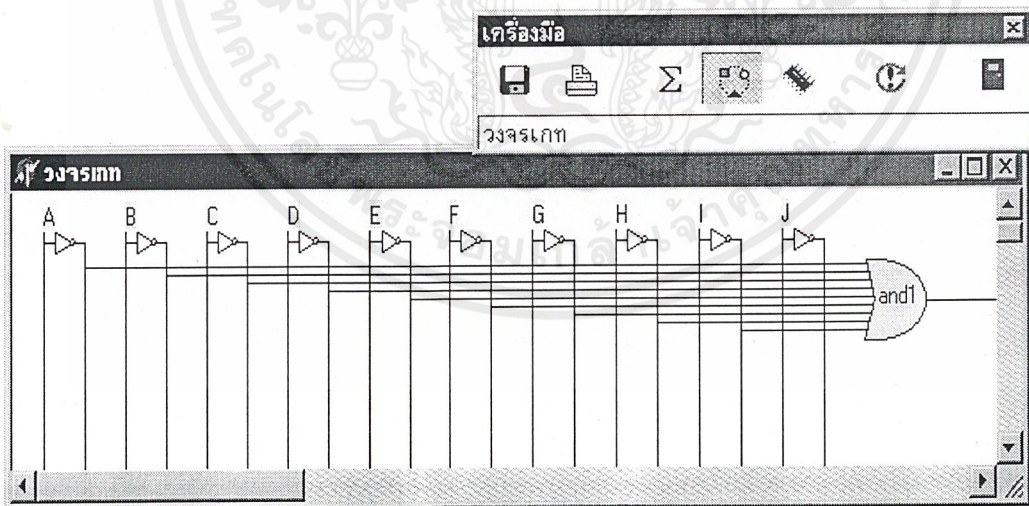
4.9.2 ผลการทดลอง

1) จะปรากฏหน้าต่างเครื่องมือขึ้น ให้กดปุ่มสัญลักษณ์ที่แสดงผลเป็นสมการ จะปรากฏหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว” ดังรูปที่ 4.43



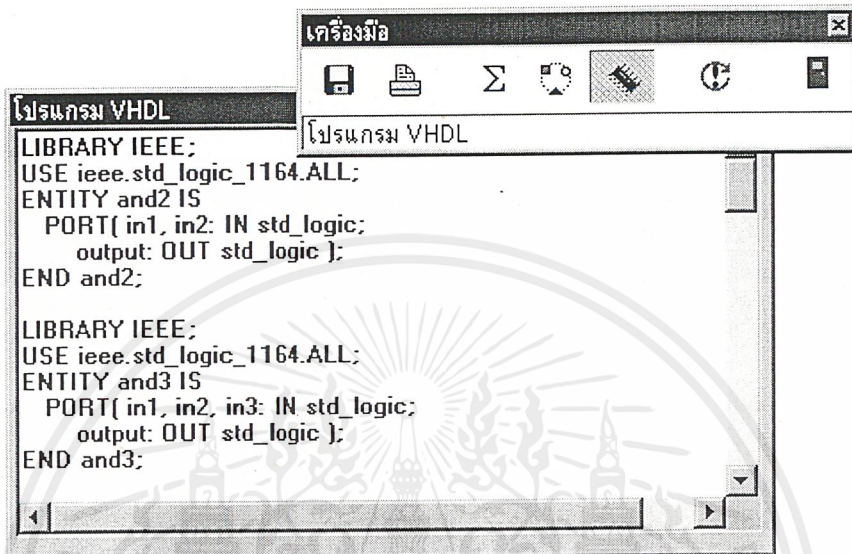
รูปที่ 4.43 หน้าต่าง “เครื่องมือ” และหน้าต่าง “สมการบูลีนที่ลดรูปแล้ว”

2) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรถูกจะปรากฏหน้าต่าง “วงจรถูก” ดังรูปที่ 4.44



รูปที่ 4.44 หน้าต่าง “เครื่องมือ” และหน้าต่าง “วงจรถูก”

3) กดที่ปุ่มสัญลักษณ์ที่แสดงผลเป็นวงจรมี VHDL จะปรากฏหน้าต่าง “โปรแกรม VHDL” ดังรูปที่ 4.45



รูปที่ 4.45 หน้าต่าง “เครื่องมือ” และหน้าต่าง “โปรแกรม VHDL”

4.10 สรุปผลการทดลอง

การทดลองในบทที่ 4 การทดลองและผลการทดลอง เป็นการทดลองการทำงานของโปรแกรมลดรูปสมการบูลีนตั้งแต่ 2 – 10 ตัวแปร ซึ่งผลการทดลองแสดงให้เห็นว่าโปรแกรมลดรูปสมการบูลีนให้ผลการทำงานที่ถูกต้อง ทั้งสมการ รูปวงจรถ และสมการ VHDL แต่การแสดงผลในรูปของวงจรถ เกิดความผิดพลาดขึ้นบ้าง เนื่องจากผลการทำงานของโปรแกรมที่ไม่สมบูรณ์

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

โปรแกรมลดรูปสมการบูลีน โดยวิธีการควินแมคคอสกีสามารถลดรูปสมการได้ตั้งแต่ 2 ถึง 10 ตัวแปร แสดงสมการที่ลดรูปให้เป็นวงจรถูกได้ และสามารถที่จะแปลงผลลัพธ์ให้อยู่ในรูปของโปรแกรมภาษา VHDL เพื่อใช้ในงานที่เกี่ยวข้องกับการโปรแกรมอุปกรณ์โปรแกรมได้ (Programmable Logic Devices) เช่น อุปกรณ์ FPGA เป็นต้น

5.2 ปัญหา

5.2.1 เมื่อทำการปิดแฟ้มโปรเจกต์แล้ว ส่วนของการแสดงผลวงจรถูกกลับไม่ลบหน้าจอ

5.2.2 ในขณะที่เปิดหน้าต่างที่ใช้แสดงวงจรถูก จะต้องรอประมาณ 3 วินาที เนื่องจากโปรแกรมกำลังทำการวาดวงจรถูกอยู่เบื้องหลัง

5.3 แนวทางแก้ไข

5.3.1 ได้ใช้การเขียนภาพว่างเปล่าสีขาวทับลงไปเพื่อลบหน้าจอ

5.3.2 นำคำสั่งแสดงหน้าต่างไปไว้หลังคำสั่งวาดวงจรถูก

5.4 แนวทางพัฒนา

โปรแกรมลดรูปสมการบูลีน โดยวิธีการควินแมคคอสกีในปริญญาณิพนธ์ฉบับนี้ สามารถลดรูปสมการบูลีนได้สูงสุด 10 ตัวแปร 1 ฟังก์ชัน ซึ่งยังมีขีดจำกัดอยู่ จึงอยากจะแนะนำให้ผู้ศึกษาไปพัฒนาต่อได้แก้ไขและเพิ่มเติมส่วนต่างๆดังนี้

5.3.1 ให้สามารถลดรูปสมการบูลีนให้ได้มากกว่า 10 ตัวแปร

5.3.2 ให้สามารถลดรูปสมการบูลีนให้ได้มากกว่า 1 ฟังก์ชัน

5.3.3 ให้สามารถวาดวงจรถูกได้ทั้งเกทแบบออร์เกท (Or Gate), แอนด์เกท (And Gate), นีอทเกท (Not Gate), แนนเกท (Nand Gate), และ นอร์เกท (Nor Gate)

5.3.4 ในส่วนของอินพุทให้มีส่วนของการป้อนค่าแบบ “Don't Care Term”

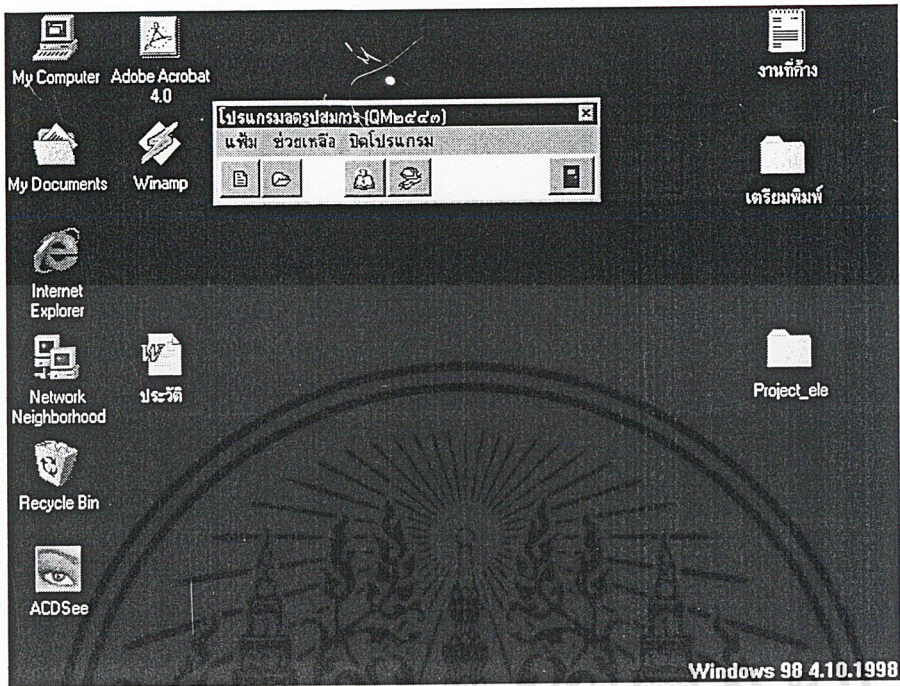
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



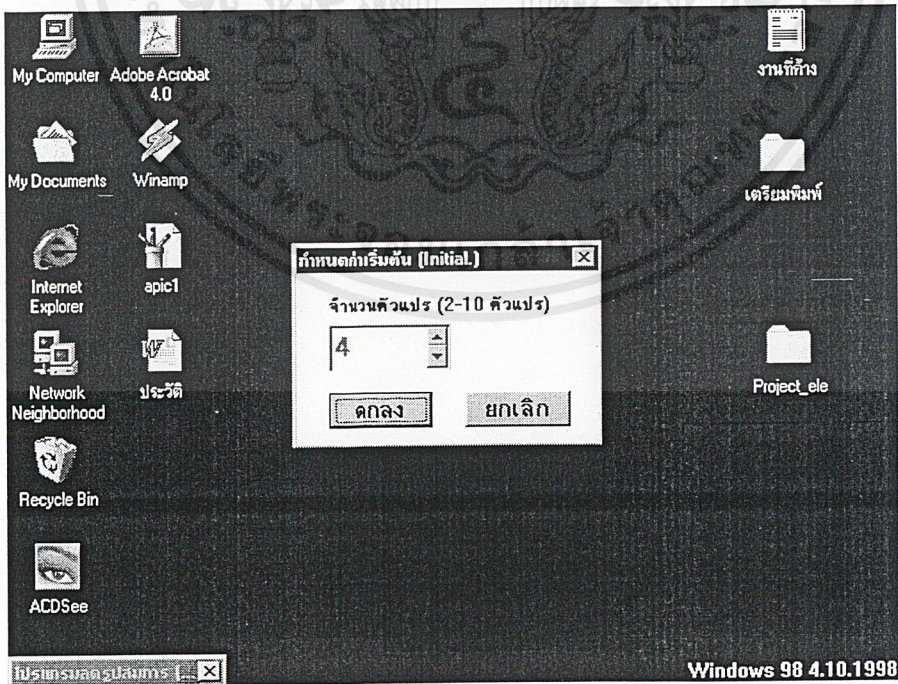
ภาคผนวก ก

โปรแกรมต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

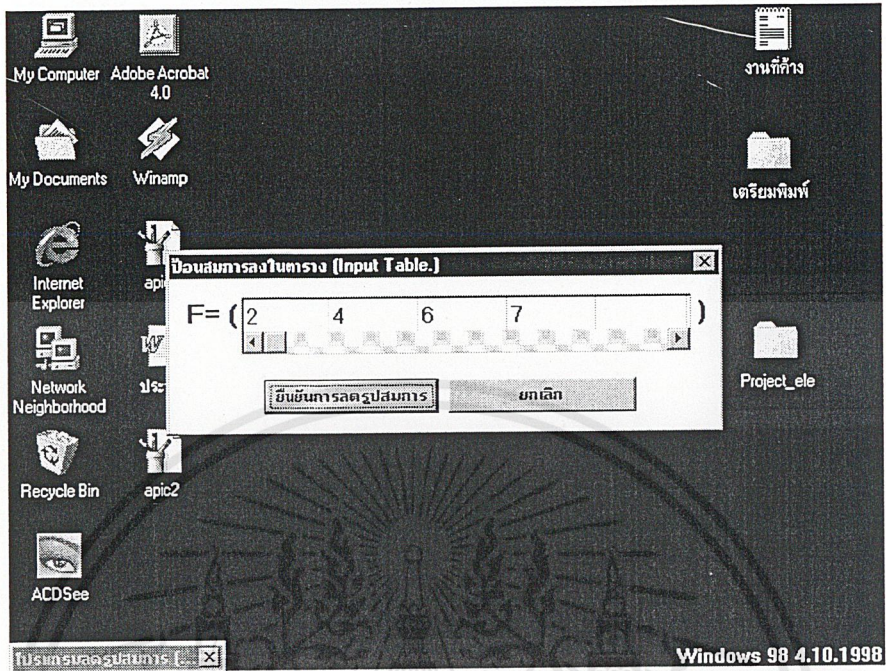


รูปที่ ก.1 หน้าต่างโปรแกรมหลัก

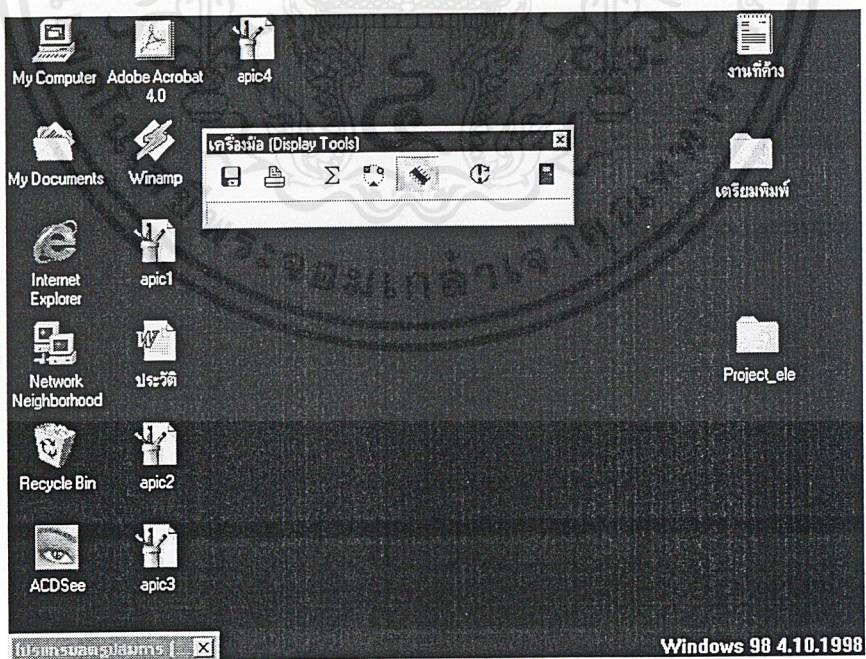


รูปที่ ก.2 หน้าต่างกำหนดการป้อนจำนวนตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

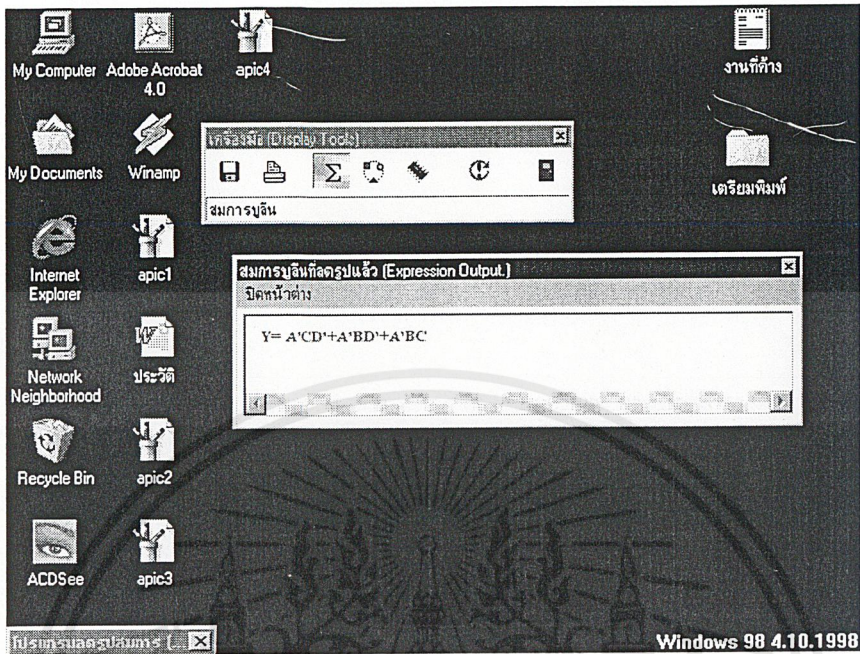


รูปที่ ก.3 หน้าต่างป้อนค่าของสมการบูลีน

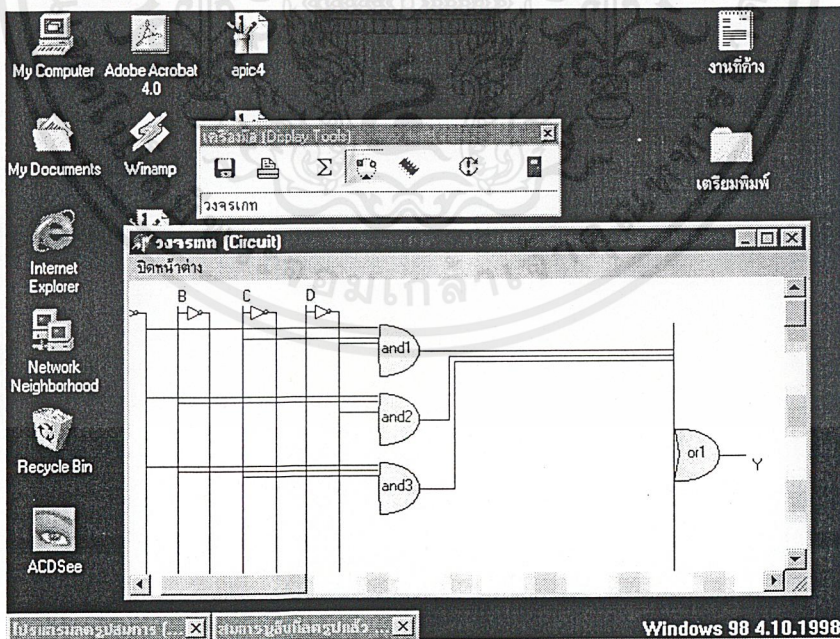


รูปที่ ก.4 หน้าต่างเครื่องมือช่วยในการเลือกรูปแบบการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

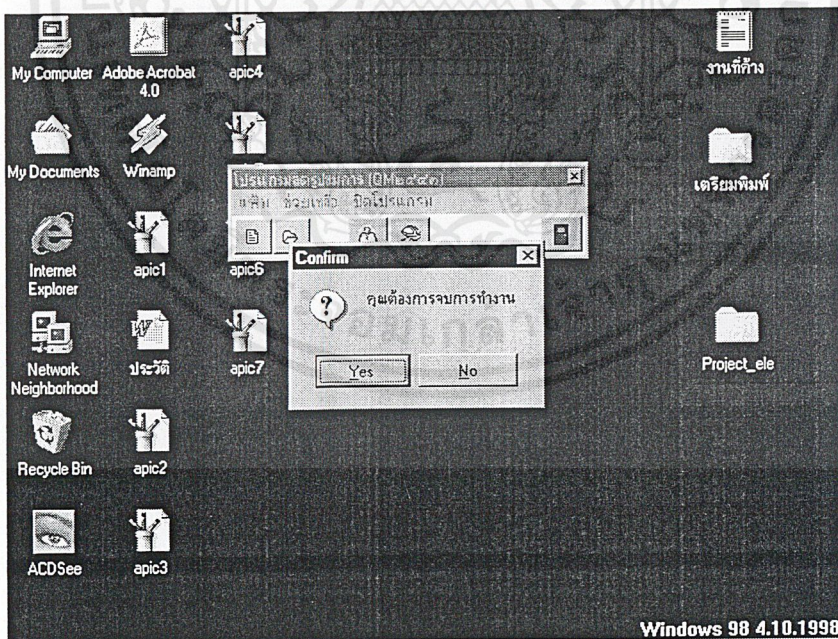
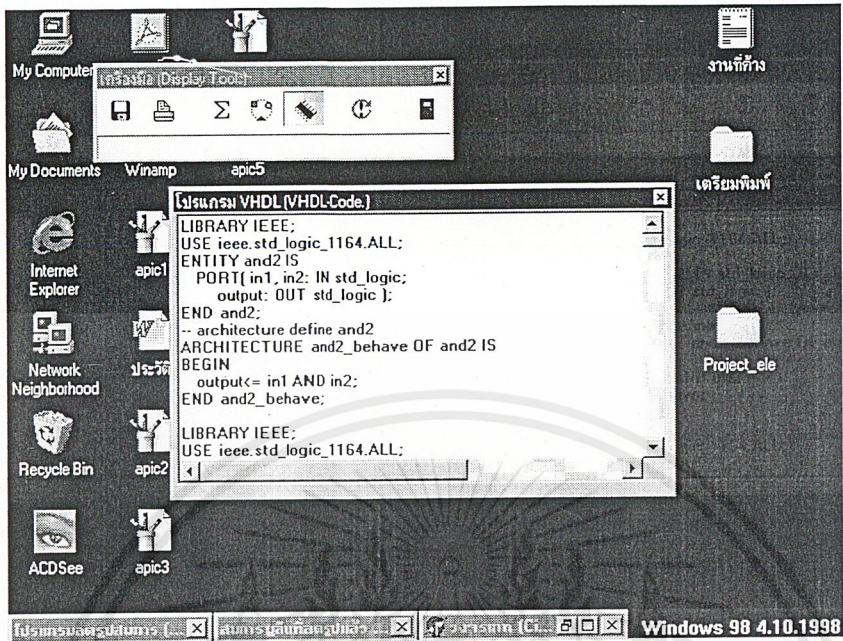


รูปที่ ก.5 การเลือกแสดงผลในแบบสมการบูลีน



รูปที่ ก.6 การเลือกแสดงผลในแบบรูปวงจรถรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

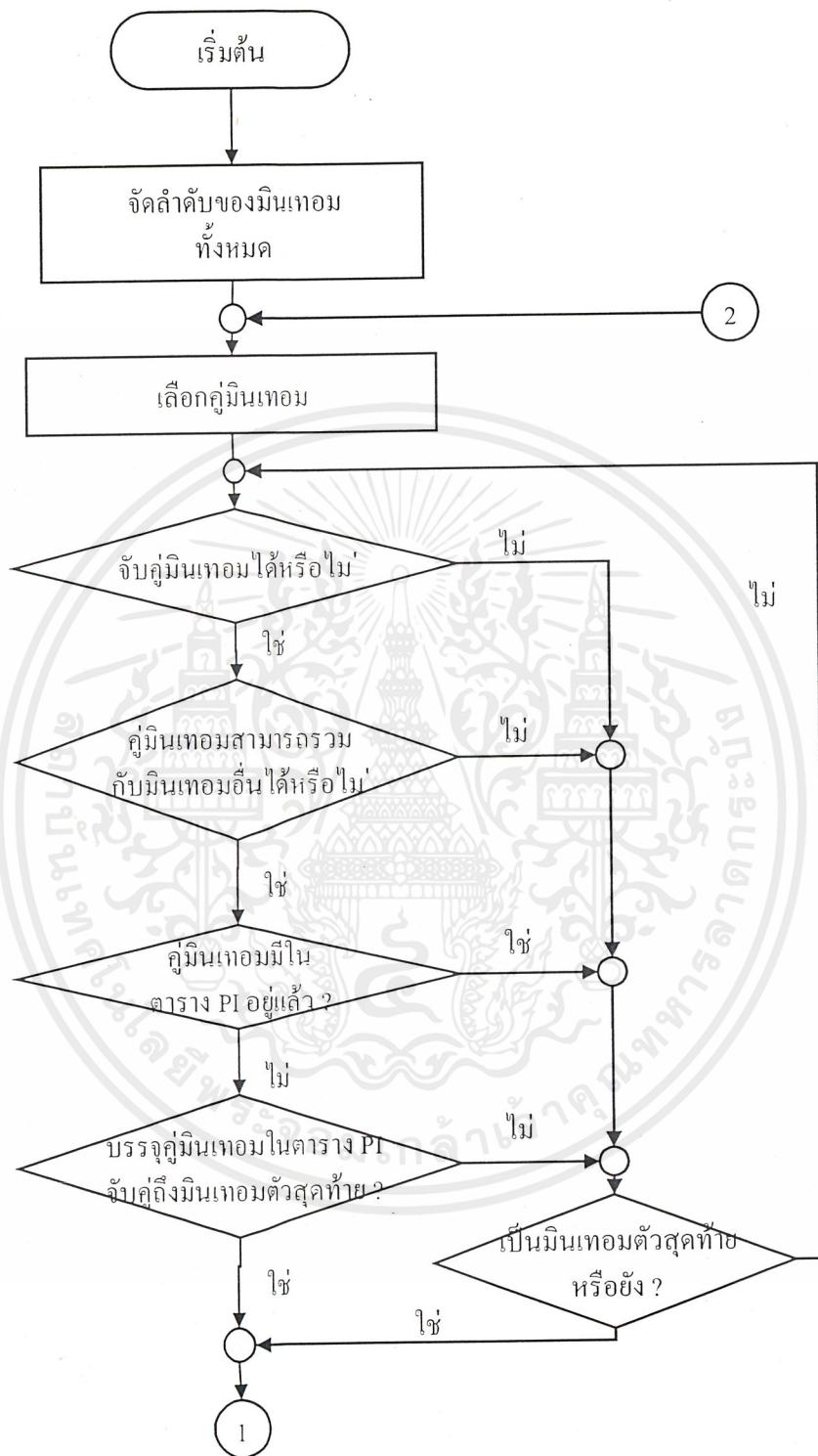


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



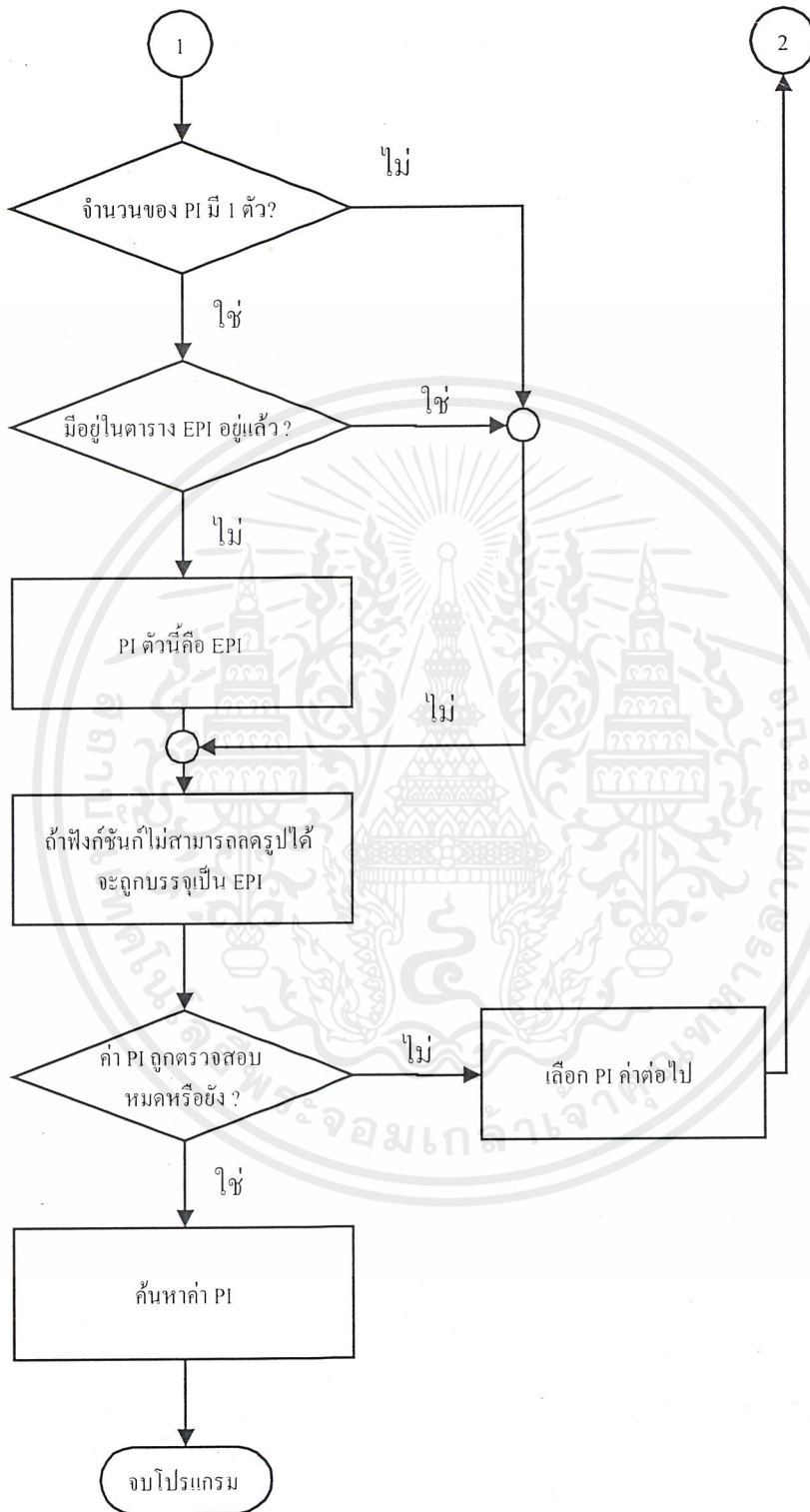
ภาคผนวก ข
ผังงานและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



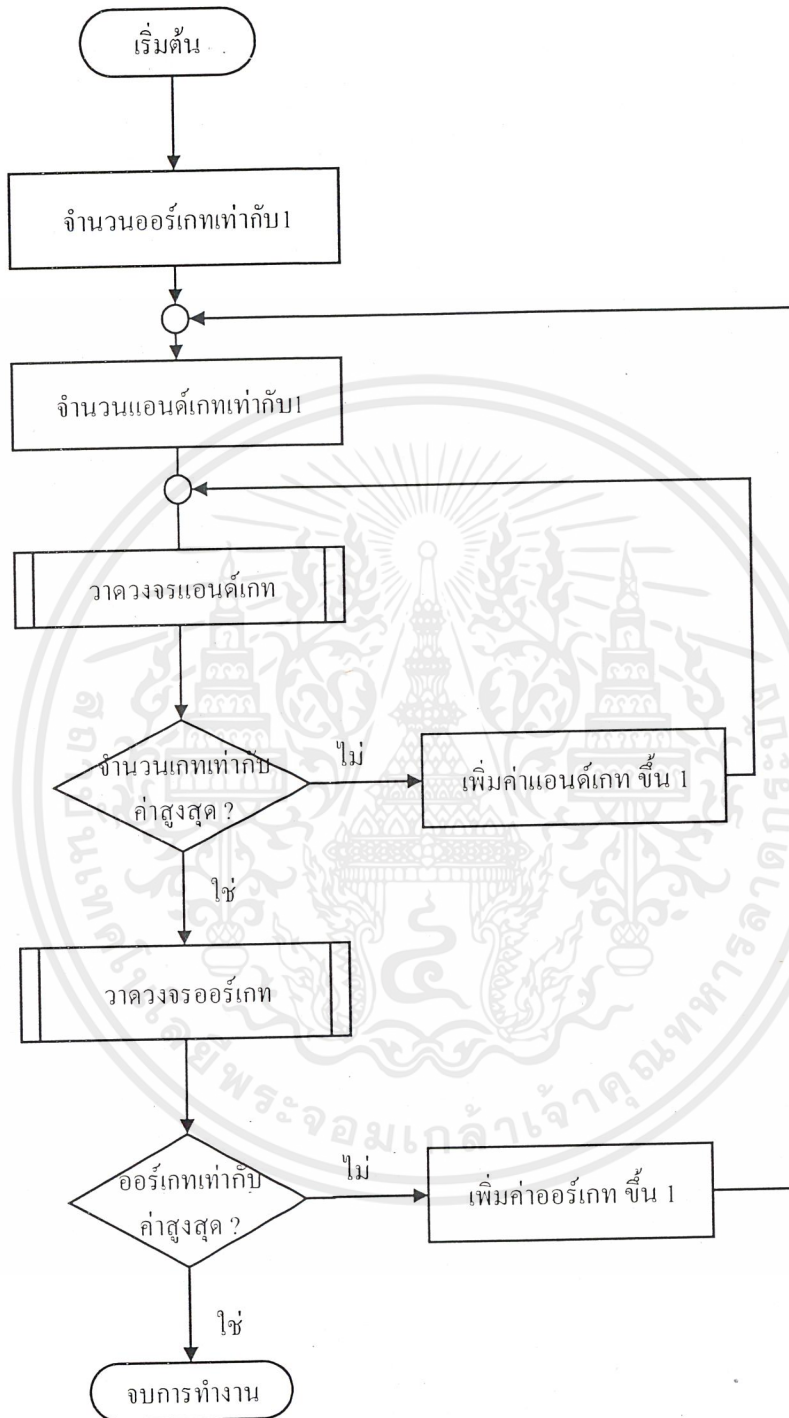
รูปที่ ข.1 ผังงานการลดรูปสมการบูลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



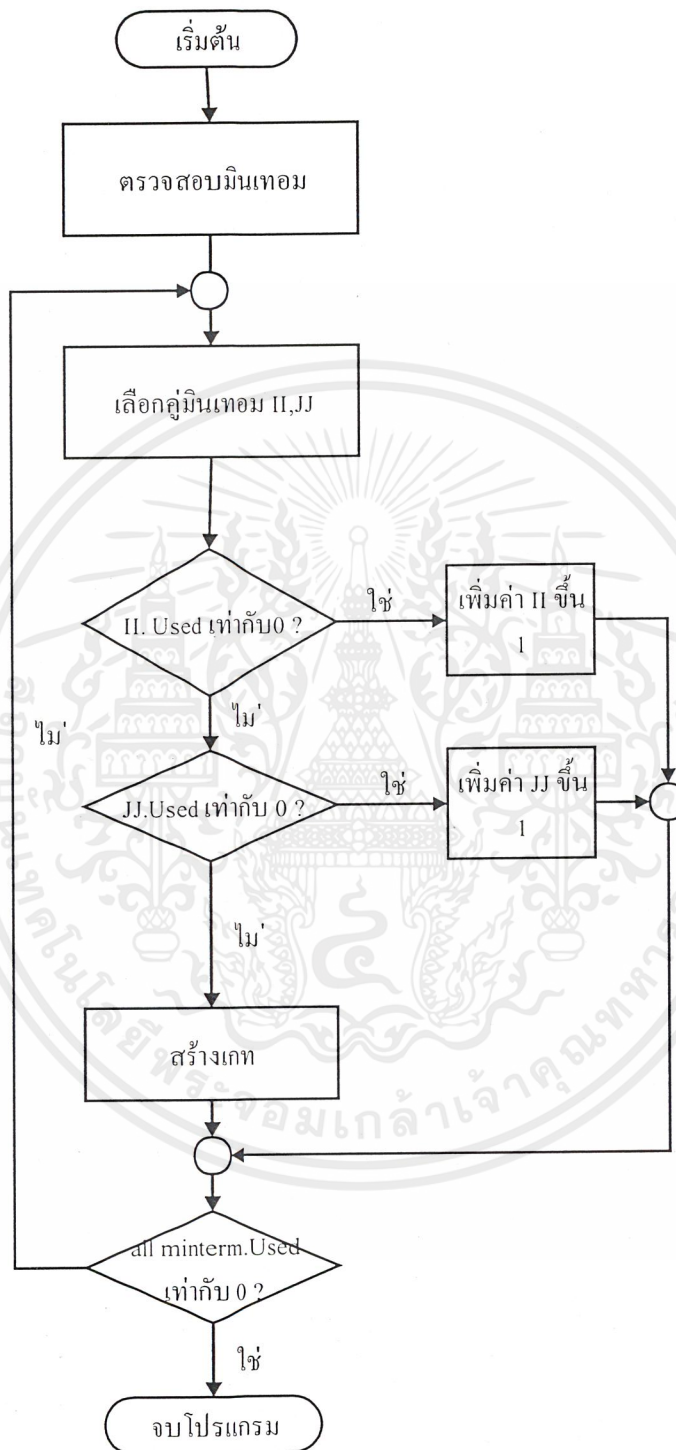
รูปที่ ข.1 (ต่อ) ผังงานการลดรูปสมการบูลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 ผังงานการแปลงสมการบูลีนเป็นโปรแกรม VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 ผังงานการแปลงสมการบูลีนเป็นรูปวงจรถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit FrmConsol;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls, Menus, ImgList, ExtCtrls, ComCtrls,
  ToolWin, Buttons, jpeg;
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N5: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N15: TMenuItem;
    N16: TMenuItem;
    N17: TMenuItem;
    ImageList1: TImageList;
    N6: TMenuItem;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    OpenFileDialog1: TOpenDialog;
    SpeedButton5: TSpeedButton;
    procedure Button4Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
CONST
  maxClearArray = 1000;
  maxVars_ = 10;
  maxCint_ = 1023;  maxCubes = 1000;
TYPE
  integer_ = 0..maxCint_;
  cube = RECORD t: integer_;
  f: integer_;

END;
TYPE

```

```

logic = 0..2;
Blogic = RECORD t: logic;

    F: logic;
END;
TYPE
maxSet = 0..255;
Setopr = RECORD sa: SET OF maxSet;
END;
VAR
cubes : ARRAY[0..maxVars_,1..maxCubes] OF cube;
covered : ARRAY[0..maxVars_,1..maxCubes] OF
boolean;
numCubes : ARRAY[0..maxVars_] OF 0..maxCubes;
oprC : ARRAY[0..maxCint_] OF Blogic;
m : 0..maxVars_;
total_,chk,tt,c,i,l,ck,q,Last,z,u,v : integer;
j,k,p : 1..maxCubes;
tempCube : cube;
found : boolean;
CC : SET OF maxSet;
PP,W,PPBuff : ARRAY[1..maxClearArray] OF
Setopr;
maxVars,maxCint : integer;
ChrOut : string;
F : TextFile;
Chkover : array[0..9] of boolean;
PROCEDURE ClrArray;stdcall;
FUNCTION EqualCubes(C1,C2: cube):
boolean;stdcall;
FUNCTION Combinable(C1,C2: cube):
boolean;stdcall;
FUNCTION powC(c : integer): integer;stdcall;
PROCEDURE ProcessCover;stdcall;
PROCEDURE ShowSet;stdcall;
PROCEDURE Cover_;stdcall;
PROCEDURE FindPI;stdcall;
PROCEDURE SRCVHDL;stdcall;
Procedure ExpressionProc;stdcall;
Procedure CircuitProc;stdcall;

implementation
uses FrmInit,FrmInput,FrmConfirm,FrmDisp, FrmGate,
    FrmVHDL, FrmSelect;
{$R *.DFM}

PROCEDURE SRCVHDL;
var

    jj_,lll_,ll_,ff,z,z_,no,aa,u,mm,v,c,chkOne,data
run,no_signal,dd,count_ : integer;
StrPort : set of 'A'..'J';

BEGIN
    AssignFile(F, 'tempVHD.vhd');

```

```

Rewrite(F);
no_signal:=0;dd:=1;jj_:=1;
for no:=2 to 10 do
begin
  Writeln(F, 'LIBRARY IEEE;');
  Writeln(F, 'USE
  ieee.std_logic_1164.ALL;');
  Writeln(F, 'ENTITY and',no,' IS');
  Write(F, '    PORT( ');
  aa:=1;
  while aa<no do
  begin
    Write(F, 'in',aa,', ');
    aa:=aa+1;
  end;
  Writeln(F, 'in',aa,': IN std_logic;');
  Writeln(F, '    output: OUT
  std_logic );');
  Writeln(F, 'END and',no,');');
  Writeln(F, '-- architecture define
  and',no);
  Writeln(F, 'ARCHITECTURE and',no, '_behave
  OF and',no, ' IS');
  Writeln(F, 'BEGIN');
  Write(F, '    output<= ');
  for aa:=1 to no do
  begin
    Write(F, 'in',aa);
    if aa<>no then
    begin
      Write(F, ' AND ');
    end;
  end;
  Writeln(F, ');');
  Writeln(F, 'END and',no, '_behave;');
  Writeln(F, '');
end;
Writeln(F, 'LIBRARY IEEE;');
Writeln(F, 'USE ieee.std_logic_1164.ALL;');
Writeln(F, 'ENTITY inv1 IS');
Writeln(F, '    PORT( in1: IN std_logic;');
Writeln(F, '    output: OUT std_logic );');
Writeln(F, 'END inv1;');
Writeln(F, '');
Writeln(F, '-- architecture define inv1');

Writeln(F, 'ARCHITECTURE inv1_behave OF inv1
IS');
Writeln(F, 'BEGIN');
Writeln(F, '    output<=NOT(inv1);');
Writeln(F, 'END inv1_behave;');
Writeln(F, '');
Writeln(F, 'LIBRARY IEEE;');
Writeln(F, 'USE ieee.std_logic_1164.ALL;');

```

```

Writeln(F, 'ENTITY or10 IS');
Writeln(F, '    PORT(in1,in2,in3,in4,
in5,in6,in7,in8,in9,in10: IN std_logic;');
Writeln(F, '        output: OUT std_logic );');
Writeln(F, 'END or10;');
Writeln(F, '');
Writeln(F, '-- architecture define or10');
Writeln(F, 'ARCHITECTURE or10_behave OF or10
IS');
Writeln(F, 'BEGIN');
Write(F, '    output<= -- user define --');
Writeln(F, 'END inv1_behave;');
Writeln(F, '');
Writeln(F, '--
*****');
Writeln(F, '-- ***** Port Map Process
*****');
Writeln(F, '--
*****');
Writeln(F, 'LIBRARY ieee;');
Writeln(F, 'USE ieee.std_logic_1164.ALL;');
Writeln(F, 'ENTITY aoi IS');
Write(F, 'PORT(');
u:=0;v:=1;
for mm:=1 to q DO
BEGIN
    IF W[mm].sa<>[] THEN u:=u+1;
END;
StrPort:=[];
for mm:=1 to u do
begin
    datarun:=0;
    while not(datarun in W[mm].sa) do
        datarun:=datarun+1;
        ff := 1;
        FOR c:=0 TO maxVars-1 DO
        BEGIN
            IF (datarun and ff)= 0 THEN
                oprC[c].t := 0
            ELSE oprC[c].t := 1;
                ff := ff shl 1;
        END;

    datarun:=1023;
    while not(datarun in W[mm].sa) do
        datarun:=datarun-1;
        ff := 1;
        FOR c:=0 TO maxVars-1 DO
        BEGIN
            IF (datarun and ff)= 0 THEN
                oprC[c].f := 0
            ELSE oprC[c].f := 1;
                ff := ff shl 1;
        END;
    END;

```

```

chkOne:=0;
FOR c:=0 TO maxVars-1 DO
BEGIN
  IF (oprC[c].t<>oprC[c].f)
  THEN
  begin
    oprC[c].t := 2;
    chkOne:=chkOne+1;
  end;
END;
count_:=0;
for c:=maxVars-1 downto 0 do
begin
  if (oprC[c].t=0) then
    count_:=count_+1;
end;
z:=0;
for c:=maxVars-1 downto 0 do
begin
  if (oprC[c].t=1) then
  begin
    StrPort:=StrPort+
      [chr(65+z)];
    end;
    z:=z+1;
  end;
end;{end for}
z:=0;ChkOne:=0;
while chr(65+z)<> 'k' do
begin
  if chr(65+z) in StrPort then
    ChkOne:=ChkOne+1;
    z:=z+1;
  end;
  z:=0;c:=0;
  while chr(65+z)<> 'k' do
  begin
    if chr(65+z) in StrPort
    then

      begin
        WRITE(F,chr(65+z));
        c:=c+1;
        if c<=ChkOne-1 then
          WRITE(F,',');
        end;
        z:=z+1;
      end;
  end;
  WRITELN(F,': IN std_logic;');
  WRITELN(F,'  aoi_output: OUT
std_logic ');');
  WRITELN(F,'END aoi;');
  WRITELN(F,'ARCHITECTURE struct
OF aoi IS');

```

```

for no:=2 to 10 do
begin
  Writeln(F, '');
  Writeln(F, '    COMPONENT
and', no);
  Write(F, '    PORT( ');
  aa:=1;
  while aa<no do
  begin
    Write(F, 'in', aa, ', ');
    aa:=aa+1;
  end;
  Writeln(F, 'in', aa, ': IN
std_logic;');
  Writeln(F, '    output:
OUT std_logic ');
  Writeln(F, '    END
COMPONENT;');

end;
Writeln(F, '    COMPONENT inv1');
Writeln(F, '    PORT( in1: IN
std_logic;');
Writeln(F, '    output: OUT
std_logic ');
Writeln(F, '    END COMPONENT;');
Writeln(F, '');
Writeln(F, '    COMPONENT or10');
Writeln(F, '    PORT(
in1,in2,in3,in4,in5,in6,in7,in8,
in9,in10: IN std_logic;');
Writeln(F, '    output: OUT
std_logic ');
Writeln(F, '    END COMPONENT;');
Writeln(F, '');
Writeln(F, '
*****
*****');
Writeln(F, '-- ***** End Port
Map Process *****');
Writeln(F, '
*****
*****');
no_signal:=no_signal+u;// total
of signal
Write(F, '    SIGNAL ');
for aa:=1 to no_signal do
begin
  Write(F, 'internal', aa);
  if aa<>no_signal then Write
(F, ', ');
end;
Writeln(F, ': std_logic;');
Writeln(F, 'BEGIN');

```

```

ll_:=1;
for mm:=1 to u do
begin
  datarun:=0;
  while not(datarun in W
[mm].sa) do
    datarun:=datarun+1;
    ff := 1;
    FOR c:=0 TO maxVars-1 DO
    BEGIN
    IF (datarun and ff)= 0 THEN
      oprC[c].t := 0
    ELSE oprC[c].t := 1;
      ff := ff shl 1;
    END;
    datarun:=1023;
    while not(datarun in W[mm].sa) do
      datarun:=datarun-1;
      ff := 1;
      FOR c:=0 TO maxVars-1 DO
      BEGIN
      IF (datarun and ff)= 0 THEN
        oprC[c].f := 0
      ELSE oprC[c].f := 1;
        ff := ff shl 1;
      END;
      chkOne:=0;
      FOR c:=0 TO maxVars-1 DO
      BEGIN
      IF (oprC[c].t<>oprC[c].f)
      THEN
        begin
          oprC[c].t := 2;
        end;
      END;
      z:=0;
      for c:=maxVars-1 downto 0 do
      begin
      if ((oprC[c].t=0) and (chkover
[c]=false)) then
      begin
        Writeln(F,'  u',dd,': inv1
PORT MAP (' ,chr(65+z),',
internal',z+1,');');
        chkover[c]:=true;
        dd:=dd+1;
        ll_:=ll_+1;
      end;
      z:=z+1;
      end;
      no_signal:=0;
      for c:=maxVars-1 downto 0 do
      begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if ((oprC[c].t=0) or (oprC[c].t=1))
        then
        begin
            no_signal:=no_signal+1;
        end;
    end;
    lll_:=1;
    if (no_signal<>1) then
    begin
        Write(F,'    u',dd,': and',no_signal,
        PORT MAP (');
        z_:=0;
        aa:=1;
        for c:=maxVars-1 downto 0 do
        begin
            if (oprC[c].t=0) then
            begin
                z_:=z_+1;
                Write(F,'internal',z_);
                if (aa<>no_signal) then
                    Write(F,', ');
                    aa:=aa+1;
            end;
            if (oprC[c].t=1) then
            begin
                Write(F,chr(65+z_));
                z_:=z_+1;
                if (aa<>no_signal) then
                    Write(F,', ');
                    aa:=aa+1;
            end;
            if (oprC[c].t=2) then z_:=z_+1;
        end;
        Write(F,',external',jj_,'');
        Writeln(F,'');dd:=dd+1;jj_:=jj_+1;
    end;
    lll_:=1;
    if (no_signal=1) then
    begin
        Write(F,'    u',dd,': or',u,' PORT MAP
        (');
        z_:=0;aa:=1;
        c:=maxVars-1;
        while (oprC[c].t=2) do
        begin
            c:=c-1;z_:=z_+1;lll_:=lll_+1;
        end;
        if (oprC[c].t=0) then
        begin
            Write(F,'internal',lll_);
        end;
        if (oprC[c].t=1) then
        begin

```

```

        Write(F,chr(65+z_));
    end;
    Writeln(F,',external',jj_,'');dd:=dd+1;
    jj_:=jj_+1;
end;
end;
Writeln(F,'END struct;');
CloseFile(F);
END;

PROCEDURE ClrArray;stdcall;
BEGIN
    {Initialize number of m-cubes at each level m.}
    CC:=[];
    FOR m:= 0 to maxVars DO numCubes[m]:=0;
    FOR l:= 1 to maxClearArray DO
    BEGIN
        PP[l].sa:=[];W[l].sa:=[];PPBuff[l].sa:=[];
    END;
    FOR l:=0 to 9 DO chkover[l]:=false;
END;

FUNCTION EqualCubes(C1,C2: cube): boolean;
BEGIN
    EqualCubes := (C1.t = C2.t) and (C1.f = C2.f);
END;

FUNCTION Combinable(C1,C2: cube): boolean;
BEGIN
    tt := 1;chk:=0;
    FOR c:=0 TO maxVars-1 DO
    BEGIN
        IF ((C1.t and tt) xor (C2.t and tt)) = 0 THEN
            oprC[c].t := 0
        ELSE oprC[c].t := 1;
        IF ((C1.f and tt) xor (C2.f and tt)) = 0 THEN
            oprC[c].f := 0
        ELSE oprC[c].f := 1;
        tt := tt shl 1;
    END;

    FOR c:=0 TO maxVars-1 DO
    BEGIN
        IF (oprC[c].t and oprC[c].f)=1 THEN chk:=chk+1;
    END;

    IF chk <> 1 THEN Combinable := false
    ELSE
    BEGIN
        chk:=0;
        FOR c:=0 TO maxVars-1 DO
        BEGIN
            IF (oprC[c].t = oprC[c].f) THEN BEGIN END

```

```

    ELSE chk:=chk+1;
END;
IF chk=0 THEN Combinable := true
ELSE Combinable:=false;
END;
END;

FUNCTION powC(c : integer): integer;stdcall;
BEGIN
    chk:=1;
    FOR i := 1 TO c DO chk := 2 * chk;
    powC := chk;
END;

PROCEDURE ProcessCover;
var
    two,zz,maxdata,ww,xx,no_shl,ii,jj,ss : integer;
BEGIN
    {***** You number 2 *****}
    two:=0;
    for zz:=maxVars-1 downto 0 do
    begin
        if oprC[zz].t =2 then two:=two+1;
    end;

    if two=0 then
    begin
        ck:=0;
        for ii:= maxVars-1 downto 0 do ck:=ck+oprC
[ii].t*powC(ii);
        PP[q].sa := PP[q].sa+[ck];
        PPBuff[q].sa:=PPBuff[q].sa+[ck];
    end
    else
    begin
        maxdata:=0;
        maxdata:=powC(two);
        {*****}
        for ww:=0 to maxdata-1 do{possible of data which
shift}
            begin
                for ii:=0 to maxVars-1 do oprC[ii].f:=oprC[ii].t;
                no_shl:=1;ii:=0;{jj:=0;{<-scan 2 stay}
                repeat
                while oprC[jj].t<>2 do jj:=jj+1;{if oprC is not found
then inc 1 step}
                {writeln(jj);}
                if (ww and no_shl)=0 then oprC[jj].t:=0
                else oprC[jj].t:=1;
                no_shl:=no_shl shl 1;
                ii:=ii+1;{check complete for scan 2}
                until ii=two;
                {*****}
                ck:=0;

```

```

    for ii:= maxVars-1 downto 0 do ck:=ck+oprC
    [ii].t*powC(ii);
    PP[q].sa := PP[q].sa+[ck];
    PPBuff[q].sa:=PPBuff[q].sa+[ck];
    {***** Store to buffer *****}
    for ii:=0 to maxVars-1 do oprC[ii].t:=oprC[ii].f;
    {*****}
end;
end;
END;

PROCEDURE Cover_;stdcall;
var
oo, ll, mm, qq, rr, yy, xx, ss, vv, minrow, ii, jj, hh, chkCo, a1,
a2: integer;
    alfa, errSet : set of maxSet;
label Finish_, cyclic;
BEGIN
//////////
    minrow:=1; alfa:=[]; errSet:=[];

    q := 1;
    for m:=0 to maxVars DO
    BEGIN
        FOR j:=1 TO numCubes[m] DO
        BEGIN
            IF NOT covered[m,j] THEN
            BEGIN
                tt:=1;
                for c:=0 TO maxVars-1 DO
                begin
                    if ((Cubes[m,j].t and tt) = (Cubes[m,j].f
and tt)) then oprC[c].t:=2
                    else if (Cubes[m,j].t and tt) = 0 then
oprC[c].t:=0
                    else oprC[c].t:=1;
                    tt := tt shl 1;
                end;
                ProcessCover;
                q:=q+1;
            END;
        END;
    END;
    q:=q-1; {***}
    for ii:=1 to q do
    begin
        for jj:=1 to q do
        begin
            if (ii<>jj) and (PP[ii].sa<>[]) and (PP[jj].sa<>[])
then
            begin
                if (PP[ii].sa*PP[jj].sa)=PP[ii].sa then PP
                [ii].sa:=[];

```

```

        end;
    end;
end;
cyclic:
for yy:=1 to q do// number of Pi Implicant
begin
////////// Identify-EPI //////////
xx:=1;alfa:=[];
alfa:=PP[yy].sa;
repeat
    if (xx<>yy) then
    begin
        alfa:=alfa-alfa*PP[xx].sa;
        xx:=xx+1;
    end
    else xx:=xx+1;
until xx>q;
if alfa<>[] then//EPI

begin
W[minrow].sa:=PPBuff[yy].sa;minrow:=minrow+1;
CC:=CC-(CC*PP[yy].sa);// Update-Set
xx:=1;    repeat
    if xx<>yy then
    begin
        PP[xx].sa:=PP[xx].sa-(PP[xx].sa*PP
[yy].sa);
        xx:=xx+1;
    end
    else xx:=xx+1;
until xx>q; // end Update
PP[yy].sa:=[];
for ii:=1 to q do
begin
    for jj:=1 to q do
    begin
if (ii<>jj) and (PP[ii].sa<>[]) and (PP[jj].sa<>[])
then
        begin
            if (PP[ii].sa*PP[jj].sa)=PP[ii].sa then
PP[ii].sa:=[];
            end;
        end;
end;////////// end remove redundante //////////
jj:=0;
for ii:=1 to q do
    if PP[ii].sa<>[] then jj:=jj+1;
//////////
if CC=[] then goto Finish_
else if (CC<>[]) and (jj=1) then
begin
    ss:=1;
    while PP[ss].sa=[] do ss:=ss+1;
W[minrow].sa:=PPBuff[ss].sa;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CC:=CC-(CC*PP[ss].sa); PP[ss].sa:=[];
        goto Finish_;
    end
    else if yy=q then //cyclic cover
    begin
        chkCo:=0;    for a1:=1 to q do
        begin
            a2:=1; errSet:=PP[a1].sa;
            repeat
                if a1<>a2 then
                begin
                    errSet:=errSet-(errSet*PP
                    [a2].sa);
                    a2:=a2+1;
                end
                else a2:=a2+1;
            until a2>q;
            if errSet<>[] then chkCo:=chkCo+1;
        end;
        if chkCo<>0 then goto cyclic;
    end;
end;{end if EPI}
end;{end for}
Finish_ :
    END;
PROCEDURE ShowSet;stdcall;
var str : string;
    u,v : integer;
BEGIN
    for u := 1 to 10 do{row}
    begin
        str:='';
        for v:=0 to maxCint_ do{data}
        begin
            if v in W[u].sa then
            begin
                str:=str+inttostr(v)+',';
            end;    end;
        showmessage(str);
    end;
END;

PROCEDURE FindPI;stdcall;
BEGIN
    FOR m:=0 to maxVars-1 DO
        begin
            FOR j:=1 TO numCubes[m] DO
                FOR k:=j+1 TO numCubes[m] DO
                    BEGIN
                        IF Combinable(cubes[m,j],cubes[m,k]) THEN
                            BEGIN
                                covered[m,j]:=true;covered[m,k]:=true;
                            tt := 1;

```

```

FOR c:=0 TO maxVars-1 DO
BEGIN
  IF ((Cubes[m,j].t and tt) and (Cubes[m,k].t and
tt)) = 0 THEN oprC[c].t := 0
  ELSE oprC[c].t := 1;
  IF ((Cubes[m,j].f and tt) and (Cubes[m,k].f and
tt)) = 0 THEN oprC[c].f := 0
  ELSE oprC[c].f := 1;
  tt := tt shl 1;
END;
tempCube.t := 0;
tempCube.f := 0;

FOR c:=0 TO maxVars-1 DO
BEGIN
  tempCube.t := tempCube.t+oprC[c].t*powC(c);
  tempCube.f := tempCube.f+oprC[c].f*powC(c);
END;

  found:=false;
  FOR p:=1 TO numCubes[m+1] DO
    IF EqualCubes(cubes[m+1,p],tempCube) THEN
      BEGIN
        found := true;
      END;
    IF NOT found THEN
      BEGIN
        numCubes[m+1]:=numCubes[m+1]+1;
        cubes[m+1,numCubes[m+1]]:=tempCube;
        covered[m+1,numCubes[m+1]]:=false;
      END;
    END;
  END;
end;
END;

Procedure ExpressionProc;stdcall;
var
  u,v,mm,c,ff,no_and,datarun,chkOne : integer;
begin
  u:=0;v:=1;ChrOut:='';
  for mm:=1 to q DO
  BEGIN
    IF W[mm].sa<>[] THEN u:=u+1;
  END;
  no_and:=u;
  for mm:=1 to u do
  begin
    datarun:=0;
    while not(datarun in W[mm].sa) do
    datarun:=datarun+1;
    ff := 1;
    FOR c:=0 TO maxVars-1 DO
    BEGIN

```

```

        IF (datarun and ff)= 0 THEN oprC[c].t := 0
        ELSE oprC[c].t := 1;
        ff := ff shl 1;
    END;
    datarun:=1023;
    while not (datarun in W[mm].sa) do
        datarun:=datarun-1;
    ff := 1;
    FOR c:=0 TO maxVars-1 DO

BEGIN
    IF (datarun and ff)= 0 THEN oprC[c].f := 0
    ELSE oprC[c].f := 1;
        ff := ff shl 1;
    END;
    chkOne:=0;
    FOR c:=0 TO maxVars-1 DO
    BEGIN
        IF (oprC[c].t<>oprC[c].f) THEN
            begin
                oprC[c].t := 2;
                chkOne:=chkOne+1;
            end;
    END;
    if chkOne=maxVars then ChrOut:=ChrOut+'1';
    z:=0;
    for c:=maxVars-1 downto 0 do
        begin
            if oprC[c].t=0 then ChrOut:=ChrOut+chr
(65+z)+Chr(39);
            if oprC[c].t=1 then ChrOut:=ChrOut+chr(65+z);
            z:=z+1;
        end;
        if ((v <= u-1)and (u<>1)) then
            begin
                ChrOut:=ChrOut+chr(43);
                v:=v+1;
            end;
        end;{end for}
    end;

    Procedure CircuitProc;stdcall;
    var
    mm,no_and,ll,c,oo,yy,gg,datarun,ff,xx,rr,qq,vv,ss,ch
    kOne,no_var : integer;
    begin
        Form6.Image1.canvas.Brush.Color:=clYellow;
        u:=0;
        for mm:=1 to q DO
            BEGIN
                IF W[mm].sa<>[] THEN u:=u+1;// number of
and gate
            END;
        no_and:=u;

```

```

ll:=0;
for oo:= 0 to maxVars-1 do
begin
  Form6.Image1.Width:=2000;
  Form6.Image1.Height:=2000;
  with Form6.Image1.Canvas do

    begin
      TextOut(20+ll,20,chr(65+oo));
      MoveTo(20+ll,35);//line 1
      LineTo(20+ll,1950);
      MoveTo(20+ll,40);//not gate
      LineTo(20+ll+7,40);
      LineTo(20+ll+7,35);
      LineTo(20+ll+18,40);
      MoveTo(20+ll+7,40);
      LineTo(20+ll+7,45);
      LineTo(20+ll+18,40);
      MoveTo(20+ll+18,40);//not gate
      Ellipse(20+ll+16,38,20+ll+20,42);
      MoveTo(20+ll+18,40);//not gate
      LineTo(20+ll+25,40);
      MoveTo(20+ll+25,40);//line 2
      LineTo(20+ll+25,1950);
    end;
    ll:=ll+50;
  end;
  u:=0;v:=1;
  yy:=52;
  for gg:=1 to q DO
  BEGIN
    IF W[gg].sa<>[] THEN u:=u+1;// number of
    and gate
  END;

  for gg:=1 to u do
  begin
    datarun:=0;
    while not(datarun in W[gg].sa) do
      datarun:=datarun+1;
    ff := 1;
    FOR c:=0 TO maxVars-1 DO
    BEGIN
      IF (datarun and ff)= 0 THEN oprC
      [c].t := 0
      ELSE oprC[c].t := 1;
      ff := ff shl 1;
    END;
    datarun:=1023;
    while not(datarun in W[gg].sa) do
      datarun:=datarun-1;
    ff := 1;
    FOR c:=0 TO maxVars-1 DO
    BEGIN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF(datarun and ff)=0 THEN
oprC[c].f:= 0
ELSE oprC[c].f := 1;

    ff := ff shl 1;
END;
chkOne:=0;
FOR c:=0 TO maxVars-1 DO
BEGIN
    IF (oprC[c].t<>oprC[c].f) THEN
    begin
        oprC[c].t := 2;
        chkOne:=chkOne+1;
    end;
END;
//////////////////////////////////**input gate //
if (chkOne<>maxVars) then
begin
xx:=20;ll:=0;
rr:=0;
qq:=4;
vv:=0;
oo:=maxVars-1;no_var:=1;
repeat
    if(oprC[oo].t=2) then
    begin
        rr:=rr+qq;
        ll:=ll+50;
    end;
    if(oprC[oo].t=0) then
    begin
        ll:=ll+25;//inc x
        Form6.Imagel.canvas.MoveTo
        (xx+ll,yy+rr);
        Form6.Imagel.canvas.LineTo
        ((50*maxVars)+25,yy+rr);
        if chkOne=maxVars-1 then
        begin
            Form6.Imagel.canvas.LineTo
            ((50*maxVars)+128,yy+rr);
        end;
        ll:=ll+25;
        rr:=rr+qq;//inc gap of leg gate
    end;
    if(oprC[oo].t=1) then
    begin
        Form6.Imagel.Canvas.MoveTo
        (xx+ll,yy+rr);
        Form6.Imagel.canvas.LineTo
        ((50*maxVars)+25,yy+rr);
        if chkOne=maxVars-1 then
        begin
            Form6.Imagel.canvas.LineTo
            ((50*maxVars)+128,yy+rr);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end;
        rr:=rr+qq;
        ll:=ll+50;
    end;
    oo:=oo-1;
until oo<0;
    yy:=yy+52;
end;{**}
end;
if (chkOne<>maxVars) and (chkOne<>maxVars-1)then
begin
ss:=50;yy:=0;mm:=(50*maxVars)+25;
for oo:=1 to u do
begin
with Form6.Image1.Canvas do
begin
Pie(mm-30,ss,mm+32,ss+40,250,1000,250,-
1000);
TextOut(mm+2,ss+10,'and'+inttostr(oo));
MoveTo(mm+32,ss+20);
LineTo(mm+50+yy,ss+20);
LineTo(mm+50+yy,70+yy);
LineTo(mm+225,70+yy);
END;
ss:=ss+52;yy:=yy+4;
end;
end;
if (no_and > 1) then
begin
with Form6.Image1.Canvas do
begin
MoveTo(mm+225,50);
LineTo(mm+225,250);
Pie(mm+200,130,mm+260,170,300,1000,
300,-1000);
TextOut(mm+235,140,'or1');
MoveTo(mm+260,150); LineTo
(mm+280,150);
Brush.Color:=clWhite;TextOut
(mm+285,150,'Y');
end;
end;
if chkOne=maxVars then
begin
Form6.Image1.Canvas.TextOut
(mm+200,150,'Y=1');
end;
end;
end;

procedure TForm1.Button4Click(Sender: TObject);

begin

```

```

Form8.visible:=true;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
Form1.Top:=10;
Form1.Left:=10;
ClrArray;
end;
procedure TForm1.N7Click(Sender: TObject);
begin
ClrArray;
Form1.WindowState:=wsminimized;
Form2.ShowModal;
Form2.spinedit1.value:=2;
end;

procedure TForm1.N12Click(Sender: TObject);
begin
Form5.Close;
Form6.Close;
Form7.Close;
Form8.Close;
ClrArray;
end;

procedure TForm1.N6Click(Sender: TObject);
var conf_ : integer;
begin
conf_:=messagedlg('คุณต้องการจบการทำงาน',mtconfirmation,
[mbyes,mbno],0);
if(conf_=mryes) then self.close;
end;
procedure TForm1.SpeedButton5Click(Sender: TObject);
var conf_ : integer;
begin
conf_:=messagedlg('คุณต้องการจบการทำงาน',mtconfirmation,
[mbyes,mbno],0);
if(conf_=mryes) then self.close;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
ClrArray;
Form1.WindowState:=wsminimized;
Form2.ShowModal;
Form2.spinedit1.value:=2;
Form2.SpinEdit1.TabOrder:=0;
Form3.StringGrid1.TabOrder:=0;

end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
begin

```

```

showmessage ('>> ทิมพัฒนาโครงการน <<' + #13#10#13#10+ ' ฉัตรชัย อุทธา: ลด
รูปสมการ '+ #13#10+ ' ซิน อรัณยกานนท์: วงจรเกท '+ #13#10+ ' พรอนันต์ ผ่านสำแดง: แปลง
ไฟล์ *.VHDL' );
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
var
  jpp: TJpegImage;
  Myexp : textfile;
  aline : string[80];
  amt   : integer;
begin
  if OpenDialog1.Execute then
  begin
    if OpenDialog1.FilterIndex=1 then
    begin
      ClrArray;
      amt:=0;CC:=[];numCubes[0]:=0;
      with form1 do
      begin
        AssignFile(Myexp, opendialog1.FileName);
        Reset(Myexp);
        readln(Myexp,aline);
        maxVars:=strtoint(aline); // maxVars
        maxCint := powC(maxVars)-1;

        while not eof(Myexp) do
        begin
          readln(Myexp,aline);
          amt:=amt+1;
          Cubes[0,amt].t:=strtoint(aline);
          Cubes[0,amt].f:=maxCint-Cubes[0,amt].t;
          covered[0,amt]:=false;
          CC:=CC+[cubes[0,amt].t];
        end;
        numCubes[0]:=amt;
        closefile(Myexp);

        FindPI;
        Cover_;
        ExpressionProc;
        CircuitProc;
        Form1.WindowState:=wsminimized;
        Form8.showmodal;
      end;
    end;
  end

  else if OpenDialog1.FilterIndex=2 then
  begin
    form5.Memo1.lines.Delete(0);
    with form1 do
    begin
      AssignFile(Myexp, opendialog1.FileName);
      Reset(Myexp);

```

```

        readln(Myexp, ChrOut);
        form5.memo1.Lines.Add('Y= '+ChrOut);
        closefile(Myexp);
        Form5.ShowModal;
    end;
    end
    else if OpenDialog1.FilterIndex=3 then
    begin
        form6.ShowModal;
        jp := TJpegImage.Create;
        try
            with jp do
            begin
                Assign(form6.Image1.Picture.Bitmap);
                form6.image1.picture.LoadFromFile
                (opendialog1.FileName);
            end;
            finally
                jp.Free;
            end;
        end
    end;
end;
end;
end.

unit FrmDisp;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs,
    StdCtrls, Buttons, Menus;
type
    TForm5 = class(TForm)
        Memo1: TMemo;
        MainMenu1: TMainMenu;
        N2: TMenuItem;
        SaveDialog1: TSaveDialog;
        procedure Button1Click(Sender: TObject);
        procedure N2Click(Sender: TObject);
        procedure N1Click(Sender: TObject);
    private
        { Private declarations }
    public

        { Public declarations }
    end;
var
    Form5: TForm5;
implementation
uses FrmConsol;
{$R *.DFM}
procedure TForm5.Button1Click(Sender: TObject);
begin
    Form5.Close;

```

```

end;

procedure TForm5.N2Click(Sender: TObject);
begin
self.close;
end;

procedure TForm5.N1Click(Sender: TObject);
var
  Myexp : textfile;
begin
  if savedialog1.Execute then
  begin
    with form1 do
    begin
      AssignFile(Myexp, savedialog1.FileName+'.exp');
      Rewrite(Myexp);
      Writeln(Myexp, ChrOut);
      closefile(Myexp);
    end;
  end;
end;
end.

unit FrmGate;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Menus, jpeg, ExtDlgs ;
type
  TForm6 = class(TForm)
    Image1: TImage;
    MainMenu1: TMainMenu;
    N2: TMenuItem;
    procedure FormPaint(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form6: TForm6;
implementation
uses FrmConsol;
{$R *.DFM}
procedure TForm6.FormPaint(Sender: TObject);
begin
with Form1 do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    Cover_;
end;
end;

procedure TForm6.N7Click(Sender: TObject);
begin
self.close;
end;
procedure TForm6.N11Click(Sender: TObject);
begin
end;

procedure TForm6.N2Click(Sender: TObject);
begin
self.close;
end;end.
unit FrmInit;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs,
    StdCtrls, Spin, Math;
type
    TForm2 = class(TForm)
        Label1: TLabel;
        Button1: TButton;
        Button2: TButton;
        SpinEdit1: TSpinEdit;
        procedure Button2Click(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
    private
    public

        end;
var
    Form2: TForm2;

implementation
uses FrmConsol, FrmInput;
procedure TForm2.Button2Click(Sender: TObject);
begin
with Form1 do
begin
    ClrArray;
end; Self.close;
Form1.WindowState:=wsnormal;
end;

```

```

procedure TForm2.Button1Click(Sender: TObject);
var
  cc : integer;
begin
  with Form1 do
  begin
    maxVars := Form2.spinedit1.value;
    maxCint := powC(maxVars)-1;
    Form3.StringGrid1.ColCount:=maxCint+1;

    for cc:=0 to maxCint+1 do
    begin
      Form3.StringGrid1.cells[cc,0] := '';
    end;
  end;
  self.Close;
  Form3.visible:=true;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  with Form1 do
  begin
    ClrArray;
  end; end;
end
unit FrmInput;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  StdCtrls, Grids;
type
  TForm3 = class(TForm)

    Button1: TButton;
    StringGrid1: TStringGrid;
    Button2: TButton;
    Label2: TLabel;
    Label1: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
implementation
uses FrmConsol, FrmConfirm, FrmSelect, FrmVHDL,
  FrmDisp, FrmGate;

```

```

{$R *.DFM}
procedure TForm3.FormCreate(Sender: TObject);
begin
with Form1 do
begin
ClrArray;
end;
end;

procedure TForm3.Button1Click(Sender: TObject);
var
  ab,zz,qq,nn,ll,uu,yy,dd,con_,hh : integer;
  done : integer;
begin
nn:=0;uu:=0;done:=0;yy:=0;con_:=0;qq:=0;hh:=0;
zz:=0;
with Form1 do
begin
for ll:=0 to maxCint+1 do
begin
if (Form3.StringGrid1.cells[ll,0]='') then
uu:=uu+1;
end;
ab:=maxCint+1;
while (ab>=0) and (Form3.StringGrid1.cells
[ab,0]='') do
begin
zz:=zz+1; ab:=ab-1;
end;
for ll:=0 to maxCint+1 do
begin
if (Form3.StringGrid1.cells[ll,0]<>'') then
begin
con_:=strtoint(Form3.StringGrid1.cells
[ll,0]);
if ((con_<0)or(con_>maxCint)) then
nn:=nn+1;
end;
end;
if (nn>0) then showmessage('ค่ามันเกินที่ป้อนต้องอยู่ในช่วง 0 ถึง
'+inttostr(maxCint)+' เท่านั้น')
else if (uu-1)=maxCint+1 then showmessage('กรุณากรอกข้อ
มูลด้วยครับ')
else if (zz<>uu) then showmessage('อย่าเว้นช่องว่างระหว่างข้อ
มูล')
else {form4.Showmodal;} {pass}
begin
with Form1 do
begin
numCubes [0] :=0;dd:=0;CC:=[];
while (Form3.StringGrid1.cells[dd,0]<>'') do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        begin
            numCubes[0] := numCubes[0] + 1;
            cubes[0, dd+1].t := strtoint(Form3.StringGrid1.cells
            [dd, 0]);
            cubes[0, dd+1].f := maxCint - cubes[0, dd+1].t;
            covered[0, dd+1] := false; CC := CC + [strtoint
            (Form3.StringGrid1.cells[dd, 0])];
            dd := dd + 1;
        end;
    end;
    FindPI;
    Cover_;
    ExpressionProc;
    CircuitProc;
SRCVHDL;
    Form3.Close;
    Form8.showmodal;
    end;

end;
Form5.Visible := false;
Form6.Visible := false;
Form7.Visible := false;
end;

procedure TForm3.Button2Click(Sender: TObject);
begin
    Form3.Close;

    Form1.WindowState := wsnormal;
end;
end.

unit FrmSelect;
interface
uses
    Windows, Messages, SysUtils, Classes, Graphics,
    Controls, Forms, Dialogs,
    StdCtrls, Buttons, ComCtrls, ExtDlgs;
type
    TForm8 = class(TForm)
        SpeedButton1: TSpeedButton;
        SpeedButton2: TSpeedButton;
        SpeedButton3: TSpeedButton;

        StatusBar1: TStatusBar;
        SpeedButton4: TSpeedButton;
        SpeedButton5: TSpeedButton;
        SpeedButton6: TSpeedButton;
        SpeedButton7: TSpeedButton;
        PrintDialog1: TPrintDialog;
        SaveDialog1: TSaveDialog;
    end;

```

```

    procedure Button2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form8: TForm8;
implementation
uses FrmConsol, FrmDisp, FrmGate, FrmVHDL, Main;
{$R *.DFM}
procedure TForm8.Button2Click(Sender: TObject);
var
  gg,rr,mm,m,ext1,datarun,c,ff,ss,oo,yy,ll,vv,xx,qq,no
  _and : integer;
begin
begin
u:=0;v:=1;ChrOut:='';
for mm:=1 to q DO
BEGIN
  IF W[mm].sa<>[] THEN u:=u+1;
END;
no_and:=u;
for mm:=1 to u do
begin
  datarun:=0;
  while not(datarun in W[mm].sa) do
    datarun:=datarun+1;
  ff := 1;
  FOR c:=0 TO maxVars-1 DO
  BEGIN
  IF (datarun and ff)= 0 THEN oprC[c].t := 0
  ELSE oprC[c].t := 1;
  ff := ff shl 1;
  END;
  datarun:=1023;
  while not(datarun in W[mm].sa) do
    datarun:=datarun-1;
  ff := 1;
  FOR c:=0 TO maxVars-1 DO
  BEGIN
  IF (datarun and ff)= 0 THEN oprC[c].f := 0
  ELSE oprC[c].f := 1;
  ff := ff shl 1;

```

```

        END;
    FOR c:=0 TO maxVars-1 DO
    BEGIN
        IF (oprC[c].t<>oprC[c].f) THEN oprC[c].t := 2;
    END;
    z:=0;
    for c:=maxVars-1 downto 0 do
    begin
        if oprC[c].t=0 then ChrOut:=ChrOut+chr
            (65+z)+Chr(39);
        if oprC[c].t=1 then ChrOut:=ChrOut+chr(65+z);
        z:=z+1;
    end;
    if ((v <= u-1)and (u<>1)) then
    begin
        ChrOut:=ChrOut+chr(43);
        v:=v+1;
    end;
    end; {end for}
    Form5.visible:=true;
    Form5.Memol.Enabled:=True;
    Form5.Memol.Lines.Delete(0);

    Form5.Memol.Lines.Add('Y= '+ChrOut);
    end;
begin
    Form6.visible:=true;
    u:=0;
    for mm:=1 to q DO
    BEGIN
        IF W[mm].sa<>[] THEN u:=u+1; // number of
and gate
    END;
    no_and:=u; {***** }
    ll:=0;
    for oo:= 0 to maxVars-1 do
    begin
        Form6.Imagel.Width:=2000;
        Form6.Imagel.Height:=2000;
        with Form6.Imagel.Canvas do
        begin
            TextOut(20+ll,20,chr(65+oo));
            MoveTo(20+ll,35); //line 1
            LineTo(20+ll,1950);
            MoveTo(20+ll,40); //not gate
            LineTo(20+ll+7,40);
            LineTo(20+ll+7,35);
            LineTo(20+ll+18,40);
            MoveTo(20+ll+7,40);
            LineTo(20+ll+7,45);
            LineTo(20+ll+18,40);
            MoveTo(20+ll+18,40); //not gate

            Ellipse(20+ll+16,38,20+ll+20,42);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MoveTo(20+ll+18,40); //not gate
LineTo(20+ll+25,40);
MoveTo(20+ll+25,40); //line 2

LineTo(20+ll+25,1950);
end;
ll:=ll+50;
end;
u:=0;v:=1; yy:=52;
for gg:=1 to q DO
BEGIN
IF W[gg].sa<>[] THEN u:=u+1;
END;
for gg:=1 to u do
begin
datarun:=0;
while not(datarun in W[gg].sa) do
datarun:=datarun+1;
ff := 1;

FOR c:=0 TO maxVars-1 DO
BEGIN
IF (datarun and ff)= 0 THEN oprC
[c].t := 0
ELSE oprC[c].t := 1;
ff := ff shl 1;
END;
datarun:=1023;
while not(datarun in W[gg].sa) do
datarun:=datarun-1;
ff := 1;
FOR c:=0 TO maxVars-1 DO
BEGIN
IF(datarun and ff)=0 THEN oprC[c].f:= 0
ELSE oprC[c].f := 1; ff := ff shl 1;
END;
FOR c:=0 TO maxVars-1 DO
BEGIN
IF (oprC[c].t<>oprC[c].f) THEN oprC
[c].t := 2;
END;
xx:=20;ll:=0; rr:=0;
qq:=4; vv:=0;
oo:=maxVars-1;
repeat
if(oprC[oo].t=2) then
begin
rr:=rr+qq;
ll:=ll+50;
end;
if(oprC[oo].t=0) then
begin
ll:=ll+25; //inc x
Form6.Image1.canvas.Brush.Color:=clYe

```

```

llow;
    Form6.Image1.canvas.MoveTo
        (xx+ll,yy+rr);
    Form6.Image1.canvas.LineTo
        ((50*maxVars)+25,yy+rr);
    ll:=ll+25;
    rr:=rr+qq;//inc gap of leg gate
end;
if(oprC[oo].t=1) then
begin
    Form6.Image1.Canvas.MoveTo
        (xx+ll,yy+rr);
    Form6.Image1.canvas.LineTo
        ((50*maxVars)+25,yy+rr);
    rr:=rr+qq;
    ll:=ll+50;

    end;
    oo:=oo-1;
until oo<0;
    yy:=yy+52;
end;
ss:=50;yy:=0;mm:=(50*maxVars)+25;
for oo:=1 to u do
begin
    with Form6.Image1.Canvas do
    begin
        Pie(mm30,ss,mm+32,ss+40,250,1000,250,-
            1000);
        TextOut(mm+2,ss+10,'and'+inttostr(oo));
        MoveTo(mm+32,ss+20);
        LineTo(mm+50+yy,ss+20);
        LineTo(mm+50+yy,70+yy);
        LineTo(mm+225,70+yy);
    END;
    ss:=ss+52;yy:=yy+4;
end;
if (no_and > 1) then
begin
    with Form6.Image1.Canvas do
    begin
        MoveTo(mm+225,50);
        LineTo(mm+225,250);

        Pie(mm+200,130,mm+260,170,300,1000,
            300,-1000);
        TextOut(mm+235,140,'or1');
        MoveTo(mm+260,150);
        LineTo(mm+280,150);
        Brush.Color:=clWhite;
        TextOut(mm+285,150,'Y');
    end;
end;
end;

```

```

end; {end redio2}
begin
  Form7.Visible:=true;
  Form7.memc1.Lines.LoadFromFile('c:\MIN.vhd');
  End ;end;
procedure TForm8.SpeedButton1Click(Sender: TObject);
begin
  StatusBar1.SimpleText:='สมการบูลีน';
with form1 do
begin
  Form5.Memo1.Enabled:=True;
  Form5.Memo1.Lines.Delete(0);
  Form5.Memo1.Lines.Add('Y= '+ChrOut);

  Form6.WindowState:=wsminimized;
  Form7.WindowState:=wsminimized;
  Form5.Visible:=true;
  Form5.WindowState:=wsnormal;
end;
end;

procedure TForm8.SpeedButton2Click(Sender: TObject);
begin
  StatusBar1.SimpleText:='วงจรถูก';
  Form5.WindowState:=wsminimized;
  Form7.WindowState:=wsminimized;
  Form6.Visible:=true;
  Form6.WindowState:=wsnormal;
end;
procedure TForm8.SpeedButton3Click(Sender: TObject);
begin
  StatusBar1.SimpleText:='โปรแกรม VHDL';
  Form6.WindowState:=wsminimized;
  Form5.WindowState:=wsminimized;
  Form7.Visible:=true;
  Form7.WindowState:=wsnormal;
form7.Memo1.Lines.LoadFromFile('tempVHD.vhd');
end;
procedure TForm8.SpeedButton4Click(Sender: TObject);
begin
  Form5.close;
  Form6.Image1.canvas.Brush.Color:=clwhite;

  Form6.Image1.Canvas.Rectangle(0,0,form6.Image1.Width
,form6.Image1.Height);
  Form6.close;
  Form7.close;
  Form1.WindowState:=wsnormal;
  self.Close;
end;

procedure TForm8.SpeedButton7Click(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var
  mm : integer;
begin
  StatusBar1.SimpleText := 'รายงาน';
  total_:=0;
  for mm:=1 to q DO
  BEGIN
    IF W[mm].sa<>[] THEN total_:=total_+1;
  END;

  showmessage ('คุณลดจำนวนเกทได้: '+inttostr(numcubes[0] -
  total_)+ ' ตัว');
end;
procedure TForm8.SpeedButton5Click(Sender: TObject);
var F: TextFile;
    no,aa : integer;
begin
  StatusBar1.SimpleText := 'บันทึกโปรแกรม (*.QMC)';
  if savedialog1.Execute then
  begin
  with form1 do
  begin
    AssignFile(F,savedialog1.FileName+'.qmc');
    Rewrite(F);
    Writeln(F,maxVars);
    aa:=1;
    while aa<=numCubes[0] do
    begin
      Writeln(F,Cubes[0,aa].t);
      aa:=aa+1;
    end;
    CloseFile(F);
  end;
  end;
  end;
  end;
  procedure TForm8.FormCreate(Sender: TObject);
  begin
  Form8.Top:=10;
  Form8.Left:=10;
  end;
  procedure TForm8.SpeedButton6Click(Sender: TObject);
  begin
    StatusBar1.SimpleText := 'พิมพ์รายงานผลการลดรูปสมการ';
  end;
  end.

```

รูปที่ ข.4 โปรแกรมการลดรูปสมการบูลีนทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ธวัชชัย เลื่อนฉวี.ดิจิตอลเทคนิค 2.กรุงเทพฯ:สุภลาลัย มีเดีย.2532

นุกูล กระจาย.การเขียนโปรแกรมและประมวลผลข้อมูลด้วยเทอร์โบปาสคาล.กรุงเทพฯ:

เอช.เอ็น.กรุ๊ป.2537

สัจจะ จรัสรุ่งรวิวรร และจักรพงษ์ สุขประเสริฐ.Delphi 5.กรุงเทพฯ:คำานสุทานการพิมพ์.2543

Victor,P.N.et.at.Digital Logic Circuit Analysis and Design.Unites States of America:

Prentice-Hill.1995



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานิพนธ์	นายฉัตรชัย อุทรา
วันเดือนปีเกิด	5 พฤษภาคม พ.ศ.2522
สถานที่เกิด	จังหวัดอุบลราชธานี
ภูมิลำเนาเดิม	เลขที่ 74/12 ถ.สถิตนิมานกาล ต.ท่าช้าง กิ่งอ.สว่างวีรวงศ์ จ.อุบลราชธานี 34190
ที่อยู่ปัจจุบัน	เลขที่ 13/10 หอพักชายVP ซ. เก็กิงาม1 ถ.ฉลองกรุง แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	(02) 7392700

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเทศบาลวารินวิชาชาติ
มัธยมศึกษาตอนต้น	โรงเรียนปทุมพิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคอุบลราชธานี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	อย่าคิดว่ายาก ถ้ายังไม่ได้อ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นายชยีน อรัญยกานนท์
วันเดือนปีเกิด	12 ธันวาคม พ.ศ. 2521
สถานที่เกิด	จังหวัดนนทบุรี
ภูมิลำเนาเดิม	เลขที่ 7/18 ถนนประชาราษฎร์ ต.สวนใหญ่ อ.เมือง จ.นนทบุรี 11000
ที่อยู่ปัจจุบัน	เลขที่ 112/45 โครงการสินธร-อ่อนนุช หมู่ 3 ถนนอ่อนนุช เขตลาดกระบัง กรุงเทพ 10520
โทรศัพท์	(02) 7392783

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลนนทบุรี
มัธยมศึกษาตอนต้น	โรงเรียนรัตนาธิเบศร์
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตนนนทบุรี
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม เรียงความวันแม่ดีเด่น
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	คุณค่าของตัวคน ไม่ได้ขึ้นอยู่กับหน้าตา อยู่ที่ผลงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาบัตร	นายพรอนันต์ ฝ่าสำแดง
วันเดือนปีเกิด	11 กุมภาพันธ์ พ.ศ.2518
สถานที่เกิด	จังหวัดอุดรธานี
ภูมิลำเนาเดิม	เลขที่ 97 หมู่ 1 ต.แซแล อ.กุมภวาปี จ.อุดรธานี 41110
ที่อยู่ปัจจุบัน	เลขที่ 260/72 ซอยโรงเจ ถนนหลวงแพ่ง แขวงทับยาว เขตลาดกระบัง กรุงเทพฯ 10520
โทรศัพท์	(02) 7380728 กด 0

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านเหล่าแซแลหนองแขวง
มัธยมศึกษาตอนต้น	โรงเรียนกุมภวาปี
มัธยมศึกษาตอนปลาย	โรงเรียนกุมภวาปี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	อดีต คือมหารัษฎ์แห่งชีวิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้