การสังเคราะห์วงจรควบคุมอุณหภูมิแบบฟัซซี่ลอจิกโดยวีเอชดีแอล

VHDL SYNTHESISES OF FUZZY LOGIC TEMPERATURE CONTROLLER

โดย

นางสาวดาวีร์ ขันธะกาค เลขประจำตัว 36014143

นายดำรงพล เลิศพิรุฬห์วงศ์ เลขประจำตัว 36014144

อาจารย์ที่ปรึกษา

ดร.สมศักดิ์ ชุมช่วย

วัน เดือน ปี -1 ต.ค. 2541

เลขทะเบียน 038377

เลขเรียกหนังสือ T 39397 ด 371 ก.

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038377

ปริญญานิพนธ์ปีการศึกษา 2539

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การสังเคราะห์วงจรควบคุมอุณหภูมิแบบฟัซซี่ลอจิกโดยวีเอชดีแอล

ผู้จัดทำ

1. นางสาวดาวีร์ ขันธะกาค รหัสประจำตัว 36014143
2. นายดำรงพล เลิศพิรุฬห์วงศ์ รหัสประจำตัว 36014144

....อาจารย์ที่ปรึกษา

(ดร.สมศักดิ์ ชุมช่วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสังเคราะห์วงจรควบคุมอุณหภูมิแบบฟัซซี่ลอจิกโดยวีเอชดีแอล

VHDL SYNTHESISES OF FUZZY LOGIC TEMPERATURE CONTROLLER

นางสาวดาวีร์ ขันธะกาด 36014143

นายดำรงพล เลิศพิรุฬห์วงศ์ 36014144

โครงงานได้รับการตรวจสอบแล้ว พร้อมที่จะทำการสอบได้

....

(ดร.สมศักดิ์ ชุมช่วย)

อาจารย์ที่ปรึกษา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสังเคราะห์วงจรควบคุมอุณหภูมิแบบฟัซซี่ลอจิกโดยวีเอชดีแอล

นางสาวดาวีร์ ขันธะกาด
นายดำรงพล เลิศพิรุฬห์วงศ์
ดร.สมศักดิ์ ชุมช่วย อาจารย์ที่ปรึกษา
ปีการศึกษา 2539

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการประยุกต์การใช้งานของระบบควบคุมแบบดิจิตอล โดยจะเป็นการควบคุมอุณหภูมิแบบพีไอ (PI : Proportional Plus Integral Control) ที่ใช้ระเบียบวิธีทางฟัซซี่ลอจิก (Fuzzy Logic) ซึ่งระบบการควบคุมนี้ลักษณะสัญญาณและการคำนวณจะเป็นแบบดิจิตอลทั้งหมด ระบบการควบคุมแบบ PI อย่างเดียวนั้น เป็นระบบที่ไม่สามารถปรับตัวเองให้เหมาะสมกับการทำงานได้ทุกสภาวะ ฉะนั้นถ้าเพิ่มส่วนที่เป็นฟัซซี่ลอจิกเข้าไปเพื่อใช้ปรับคุณลักษณะการทำงานของระบบ PI ให้เหมาะสมกับสภาพแวดล้อมต่างๆ ก็จะทำให้ระบบควบคุมนี้สามารถปรับตัวให้เหมาะสมกับการทำงานในสภาวะต่างๆ ได้อย่างมีประสิทธิภาพ โดยที่ระบบรวมทั้งหมดนี้จะสังเคราะห์ขึ้นมาโดยภาษาวีเอชดีแอล (VHDL) ซึ่งเป็นภาษาที่ใช้อธิบายลักษณะการทำงานของวงจรดิจิตอลต่างๆ และเมื่อได้ผลลัพธ์จากการใช้ภาษาวีเอชดีแอลในการสังเคราะห์วงจรข้างต้น ผลลัพธ์นี้จะสามารถนำไปเข้าสู่ขบวนการสร้างไอซี (Fabrication) หรือนำไปโปรแกรมลงเอฟพีจีเอ (FPGA) เพื่อใช้งานจริงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VHDL SYNTHESISES OF FUZZY LOGIC TEMPERATURE CONTROLLER

Darwee Khunthakard
Damrongpone Lertpriroonwong
Dr.Somsak Choomchuay
1996

Abstract

This purpose of our thesis is the application of digital control system design emphasizes on · PI (Proportional Plus Integral Control) temperature controlling. Fuzzy Logic Technique is employed in this project to acheive the ease of controllability. Since PI Controlling can not adjust itself to serve all different circumstances, Fuzzy Logic Technique is implemented to solve this problem. This system then has the capability of self adjusting when working in different conditions. VHDL, the hardware description language, is used to synthesize the system . The result obtained is of beneficial not only to ensure the system functions, but also the VHDL output can be used to drive available programable hardware such as FPGA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
ABSTRACT	II
บทที่ 1 บทนำ	1
บทที่ 2 FUZZY LOGIC	3
2.1 FUZZY LOGIC	3
2.2 FUZZY CONTROL	5
2.3 การประยุกต์ใช้งานฟัซซี่ลอจิก	6
2.4 หลักการหา FUZZY CONTROL RULE	9
บทที่ 3 ภาษา VHDL	11
3.1 ภาษา VHDL	11
3.2 ประวัติความเป็นมาของภาษา VHDL	11
3.3 ความสามารถของภาษา VHDL	12
3.4 กระบวนวิธีในการออกแบบโดยภาษา VHDL	13
3.5 หลักการสร้างโมเดลโดยใช้ภาษา VHDL	14
3.6 ระดับของการอธิบายระบบ	18
3.7 สรุป	19
บทที่ 4 VLSI เบื้องต้น	20
4.1 ขั้นตอนในการออกแบบ	20
4.2 หน้าที่ในการออกแบบ	20
4.3 ขั้นตอนการสร้างไอซี	22
4.4 กฎของการออกแบบ	23
4.5 คอมพิวเตอร์ช่วยในการออกแบบ	24
บทที่ 5 การออกแบบในเบื้องต้น	26
5.1 FUZZIFICATION OF INPUT	27
5.2 DEFUZZIFICATION OF OUTPUT	30
บทที่ 6 การออกแบบทางด้าน HARDWARE	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1 ส่วนสแกนคีย์บอร์ด 38

6.2 ส่วนสแกนดิสเพลย์ 39

6.3 ส่วน BUFFER TEMPERATURE 40

6.4 ส่วน FUZZY ALGORITHM 40

6.5 ส่วนคำนวณแบบพีไอ 41

6.6 ส่วนสร้างสัญญาณนาฬิกา 2 เฟส 43

6.7 การทำงานโดยรวม 44

บทที่ 7 การเชื่อมต่อกับอุปกรณ์ภายนอก 44

7.1 ส่วนคีย์บอร์ด 45

7.2 ส่วนเซนเซอร์อุณหภูมิ 46

7.3 ส่วน ADC 47

7.4 ส่วนแสดงผล 48

7.5 ส่วนเอาท์พุท 49

7.6 ตัวอย่างการประยุกต์ใช้งานกับระบบทำความร้อน 49

บทที่ 8 โปรแกรมและการทดสอบ 52

8.1 โปรแกรมส่วนสแกนคีย์บอร์ด 52

8.2 โปรแกรมส่วนสแกนดิสเพลย์ 53

8.3 โปรแกรมส่วน BUFFER TEMPERATURE 55

8.4 โปรแกรมส่วน FUZZY ALGORITHM 55

8.5 โปรแกรมส่วนคำนวณแบบพีไอ 56

8.6 โปรแกรมส่วนสร้างสัญญาณนาฬิกา 2 เฟส 58

8.7 โปรแกรมโดยรวมทั้งหมด 59

8.8 การทดสอบโปรแกรม 60

บทที่ 9 การสังเคราะห์วงจร 69

9.1 ตัวอย่างของการออกแบบในขั้นสูง 69

9.2 การสังเคราะห์วงจร 70

บทที่ 10 สรุปและวิจารณ์ 75

10.1 ส่วนฟัซซี่ลอจิก 75

10.2 ส่วนพีไอคอนโทรล 75

10.3 ส่วน VHDL 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.4 สรุปการทำงาน 76

10.5 วิเคราะห์ผลงานและปัญหาในการทำงาน 77

บรรณานุกรม 78

ภาคผนวก

โปรแกรมภาษาวีเอชดีแอล ผ-1

ผ.1 โปรแกรมส่วนสแกนคีย์บอร์ด ผ-1

ผ.2 โปรแกรมส่วนสแกนดิสเพลย์ ผ-7

ผ.3 โปรแกรมส่วน BUFFER TEMPERATURE ผ-13

ผ.4 โปรแกรมส่วน FUZZY ALGORITHM ผ-14

ผ.5 โปรแกรมส่วนคำนวณแบบพีไอ ผ-27

ผ.6 ส่วนสร้างนาฬิกา 2 เฟส ผ-33

ผ.7 โปรแกรมโดยรวมทั้งหมด ผ-36

กิตติกรรมประกาศ X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 แสดงระบบที่ต้องการออกแบบ 2

รูปที่ 2.1 การเปรียบเทียบการทำนายและฟัซซี่ลอจิก 4

รูปที่ 2.2 คำจำกัดความของระบบควบคุมแบบฟัซซี่ 6

รูปที่ 2.3 แสดงบล็อกไดอะแกรมของระบบควบคุมแบบฟัซซี่ ซึ่งแสดงให้เห็นถึง FUZZIFIER (FUZZ), DEFUZZIFIER (DE-FUZZ) และ INFERENCE ENGINE 8

รูปที่ 3.1 ระเบียบวิธีในการออกแบบโดยใช้ HARDWARE DESCRIPTION LANGUAGE 14

รูปที่ 4.1.1 แสดงขั้นตอนวงรอบของการออกแบบ 21

รูปที่ 4.1.2 ค่าความสัมพันธ์ระหว่าง EFFORT กับ COMPLEXITY 22

รูปที่ 4.2.1 ค่าความสัมพันธ์ระหว่างราคาของระบบกับความซับซ้อนของระบบ 23

รูปที่ 5.1 แสดงระบบควบคุมอุณหภูมิ 26

รูปที่ 5.1.1 (ก) แสดงค่า MEMBERSHIP ของ $|E(t)|$ 28

รูปที่ 5.1.1 (ข) แสดงค่า MEMBERSHIP ของ $|\Delta E|$ 29

รูปที่ 5.2.1 (ก) แสดงการหาค่า MEMBERSHIP FUNCTION ที่ $|E(t)| = 1.7$ 34

รูปที่ 5.2.1 (ข) แสดงการหาค่า MEMBERSHIP FUNCTION ที่ $|\Delta E| = 1.9$ 34

รูปที่ 6.1 แสดงวงจรที่ต้องการ 36

รูปที่ 6.2 แสดงรายละเอียดภายในของระบบที่ต้องการ 37

รูปที่ 6.3 แสดงรูปส่วนสแกนคีย์บอร์ด 38

รูปที่ 6.4 แสดงรูปส่วนสแกนดิสเพลย์ 39

รูปที่ 6.5 แสดงรูปส่วน BUFFER TEMPERATURE 40

รูปที่ 6.6 แสดงรายละเอียดของส่วน FUZZY ALGORITHM 41

รูปที่ 6.7 แสดงส่วนโครงสร้างแบบโดยตรง 2 42

รูปที่ 6.8 แสดงวงจรและสัญญาณนาฬิกาของวงจรสร้างสัญญาณนาฬิกา 2 เฟส 43

รูปที่ 7.1 แสดงถึงอุปกรณ์ภายนอกที่ระบบต้องการ 44

เอกสารนี้เป็นเอกสารที่ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.1.1 (ก)	แสดงตัวอย่างการเชื่อมต่อกับคีย์บอร์ด	45
รูปที่ 7.1.1 (ข)	แสดงการทำงานของ POWER ON RESET	45
รูปที่ 7.2.1	แสดงตัวอย่างเซนเซอร์อุณหภูมิโดยใช้ IC3911	46
รูปที่ 7.3.1	แสดงตัวอย่างการเชื่อมต่อกับ ADC เบอร์ ADC0802 หรือ ADC0804	47
รูปที่ 7.4.1	แสดงการเชื่อมต่อกับ SEVEN SEGMENT	48
รูปที่ 7.5.1	แสดงตัวอย่างของวงจร DAC	49
รูปที่ 7.6.1	แสดงตัวอย่างการประยุกต์ใช้งานกับระบบทำความร้อน	50
รูปที่ 7.6.2	แสดงการควบคุม POWER ที่ขดลวดความร้อน	50
รูปที่ 8.1	แสดง STATEMACHINE ของส่วนสแกนคีย์บอร์ด	52
รูปที่ 8.2	แสดงการทำงานของส่วนสแกนดิสเพลย์	54
รูปที่ 8.3	แสดงการทำงานของส่วน BUFFER TEMPERATURE	55
รูปที่ 8.4	แสดงการทำงานของส่วน FUZZY ALGORITHM	56
รูปที่ 8.5	แสดงการทำงานของการคำนวณแบบพีไอ	57
รูปที่ 8.6	การทำงานของส่วนสร้างสัญญาณนาฬิกา 2 เฟส	58
รูปที่ 8.7	แสดงการทำงานของระบบโดยรวม	59
รูปที่ 8.8.1	แสดงขั้นตอนการเขียนโปรแกรมภาษาวีเอชดีแอล	60
รูปที่ 8.8.2	แสดงการ COMPILE โดยใช้โปรแกรม V-SYSTEM	61
รูปที่ 8.8.3	TIMING DIAGRAM ของส่วนสแกนคีย์บอร์ด	62
รูปที่ 8.8.4	TIMING DIAGRAM ของส่วนสแกนดิสเพลย์	63
รูปที่ 8.8.5	TIMING DIAGRAM ของส่วน BUFFER TEMPERATURE	64
รูปที่ 8.8.6	TIMING DIAGRAM ของส่วน FUZZY ALGORITHM	65
รูปที่ 8.8.7	TIMING DIAGRAM ของส่วน PI CONTROLLER	66
รูปที่ 8.8.8	TIMING DIAGRAM ของส่วนสร้างสัญญาณนาฬิกา 2 เฟส	67
รูปที่ 8.8.9	TIMING DIAGRAM ของระบบโดยรวมทั้งหมด	68
รูปที่ 9.1.1	แสดงขั้นตอนการออกแบบขั้นสูง	69
รูปที่ 9.2.1	แสดงขั้นตอนการสังเคราะห์วงจร	70
รูปที่ 9.2.2	แสดงเมนูของโปรแกรมกาลิเลโอ LOGIC EXPLORER	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 9.2.3 แสดงรายงานผลการสังเคราะห์วงจรของโปรแกรมกาลิเลโอ LOGIC EXPLORER 72

รูปที่ 9.2.4 แสดงเมนูของโปรแกรมกาลิเลโอ TIME EXPLORER 73

รูปที่ 9.2.5 แสดงรูปวงจรที่ได้จากการใช้โปรแกรมกาลิเลโอสังเคราะห์ขึ้นมา 74

รูปที่ ผ.1 แสดง STATEMACHINE ของส่วนสแกนคีย์บอร์ด ผ-1

รูปที่ ผ.2 แสดงการทำงานของส่วนสแกนดิสเพลย์ ผ-7

รูปที่ ผ.3 แสดงการทำงานของส่วน BUFFER TEMPERATURE ผ-13

รูปที่ ผ.4 แสดงการทำงานของส่วน FUZZY ALGORITHM ผ-14

รูปที่ ผ.5 แสดงการทำงานของส่วนคำนวณแบบพีไอ ผ-27

รูปที่ ผ.6 แสดงการทำงานของส่วนสร้างสัญญาณนาฬิกา 2 เฟส ผ-33

รูปที่ ผ.7 แสดงการทำงานของระบบโดยรวม ผ-36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การควบคุมอุณหภูมิให้มีประสิทธิภาพ คือ สามารถปรับค่าอุณหภูมิให้ลู่เข้าค่าใดค่าหนึ่งได้รวดเร็ว และมีความไว (Sensitivity) ในการปรับค่าอุณหภูมิของระบบเมื่อมีการเปลี่ยนแปลงอุณหภูมิเกิดขึ้น ซึ่งในปริญญานิพนธ์ฉบับนี้เป็นการประยุกต์ใช้งานของระบบควบคุมแบบดิจิ-ตอล โดยเป็นการควบคุมอุณหภูมิแบบพีไอ (PI : Proportional Plus Integral Control) ซึ่งระบบควบคุมพีไอนี้มีฟังก์ชันถ่ายโอน (Transfer Function) เป็น

$$T(s) = K_p + K_i/s$$

ซึ่งผลตอบสนองของระบบ เช่น Overshoot, Rise Time, Settling Time ขึ้นอยู่กับอัตราส่วนของสัมประสิทธิ์ของ K_p/K_i นั้นการเปลี่ยนแปลงค่า K_p และ K_i ก็จะมีผลต่อเสถียรภาพของระบบด้วย จึงเป็นข้อจำกัดของระบบที่ควบคุมโดยลักษณะนี้คือ

1. เมื่อเลือกอัตราส่วนของ K_p/K_i ค่ามาก จะทำให้เกิด Overshoot ขึ้นก่อนที่ระบบจะเข้าใกล้ Steady State

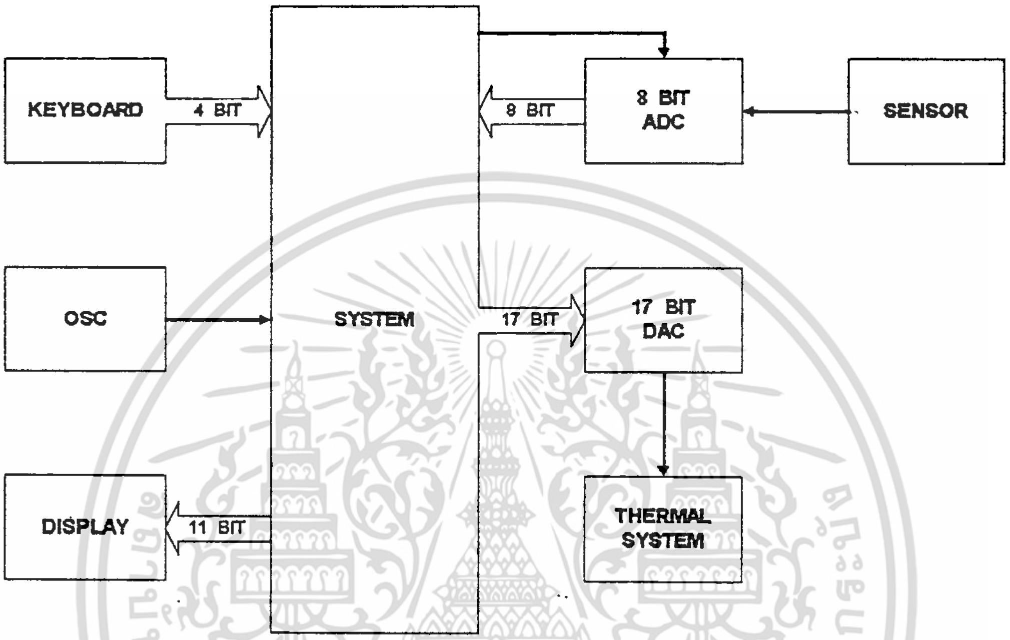
2. เมื่อเลือกอัตราส่วนของ K_p/K_i ค่าน้อย ถึงแม้ว่าจะลด Overshoot ได้ แต่ก็จะทำให้ความไวของวงจรลดลงด้วย

ดังนั้นเพื่อให้ได้ค่าอัตราส่วนที่เหมาะสม จึงต้องมีการเพิ่มส่วนที่เป็น Fuzzy Logic เข้าช่วย เพื่อให้ระบบมีการปรับคุณลักษณะการทำงานให้เหมาะสมกับสภาพแวดล้อมต่างๆ ได้อย่างมีประสิทธิภาพ

ระบบรวมทั้งหมดนี้จะถูกสังเคราะห์ขึ้นมาโดยภาษาวีเอชดีแอล (VHDL Language) ซึ่งเป็นภาษาที่ใช้อธิบายสถาปัตยกรรมของระบบและรวมไปถึงลักษณะการทำงานของวงจรดิจิตอล เมื่อเปรียบเทียบกับการออกแบบระบบโดยวิธีอื่น เช่น การควบคุมผ่านไมโครโปรเซสเซอร์ ซึ่งเป็นอุปกรณ์มีความสามารถในการทำงานได้กว้างขวาง จึงทำให้เกิดการใช้ทรัพยากรหรือความสามารถที่มีอยู่ในตัวบอร์ดไม่เต็มประสิทธิภาพ ผลที่ตามมาอีกอย่างคือ ระบบจะมีขนาดใหญ่ ราคาแพง แต่ในทางกลับกันถ้าใช้การออกแบบโดยภาษาวีเอชดีแอล ซึ่งสามารถนำผลลัพธ์จากการสังเคราะห์ระบบข้างต้นที่ผ่านมา จะสามารถนำไปผ่านขบวนการสร้างไอซี (Fabrication) ทำให้ได้ระบบมีขนาดเล็กลง ประหยัดกว่า และมีความเร็วในการทำงานสูงขึ้น หรือมิเช่นนั้นก็นำผลลัพธ์จากการสังเคราะห์ไปโปรแกรมลงเอฟพีจีเอ (FPGA) เพื่อใช้งานจริงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมาข้างต้น สามารถออกแบบระบบได้ดังรูป



รูปที่ 1.1 แสดงระบบที่ต้องการออกแบบ

จากรูปสามารถอธิบายได้ดังนี้ ระบบที่จะออกแบบนั้นจะมีการรับอินพุตจากสิ่งแวดล้อม โดยผ่านวงจร ADC ขนาด 8 บิต รวมทั้งรับอินพุตจากมนุษย์ ซึ่งผ่านมาทางการกดคีย์บอร์ด สาเหตุที่ต้องใช้วงจร ADC เนื่องมาจากระบบที่ออกแบบจะใช้การคำนวณแบบดิจิตอลทั้งหมด และสาเหตุที่ต้องใช้ในขนาด 8 บิตนั้น ก็เพื่อความละเอียดในการคำนวณ เมื่อระบบได้รับอินพุตดังกล่าวข้างต้น ก็จะนำทำการคำนวณค่าเอาท์พุตออกมาควบคุมระบบความร้อน (Thermal System) ที่ต้องการ โดยจะมีอัลกอริทึมที่ใช้ในการคำนวณจะใช้การคำนวณแบบพีไอ ที่ค่าสัมประสิทธิ์ของสมการควบคุมนั้นจะถูกเลือกให้เหมาะสมโดยใช้อัลกอริทึมของฟัซซี่ลอจิก และที่เอาท์พุตที่มีค่าขนาด 17 บิต ก็เพราะว่าเป็นขนาดที่เหมาะสมสำหรับการคำนวณ เพื่อที่จะทำให้โอกาสที่จะเกิดโอเวอร์โฟลว์มีน้อยลง ดังจะได้กล่าวในรายละเอียดของการคำนวณในบทต่อๆ ไป โดยมีเอาท์พุตอีกส่วนไว้แสดงสถานะของระบบ และมีสัญญาณนาฬิกาเป็นตัวกำหนดการทำงานให้สอดคล้องกับอัตราสุ่ม (Sampling Rate) .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
FUZZY LOGIC

มีเทคโนโลยีหลายแบบที่ช่วยลดค่าใช้จ่ายสำหรับการนำไปประยุกต์ใช้งาน หนึ่งในเทคโนโลยีดังกล่าวคือ ระบบผู้เชี่ยวชาญ (Expert System) ซึ่งก็จะรวมถึง Fuzzy Expert System ทั้ง Fuzzy Logic และ Fuzzy Expert Control System ต่างก็มักพบกับการประยุกต์ใช้งานอย่างกว้างขวางในทุกระดับ การนำ Fuzzy Logic มาใช้งานจริงครั้งแรกเกิดขึ้นในอุตสาหกรรมของญี่ปุ่นโดย Lotfi A. Zadeh ในปี 1965 ซึ่ง Zadeh ได้แสดงให้เห็นว่า Fuzzy Logic เป็นพื้นฐานของ Logic ต่างๆ โดย Fuzzy Set สามารถเป็นได้ทั้งภายในขอบเขตและขอบเขต ตัวอย่างเช่น องค์ประกอบสีต่างๆ ภายในเซต มิได้แทนเฉพาะขอบเขตสีขาวหรือสีดำเท่านั้น แต่ประกอบด้วย Spectrum ของสีเทาในระดับความเข้มต่างๆ อยู่ระหว่างขอบเขตสีขาวและดำด้วย

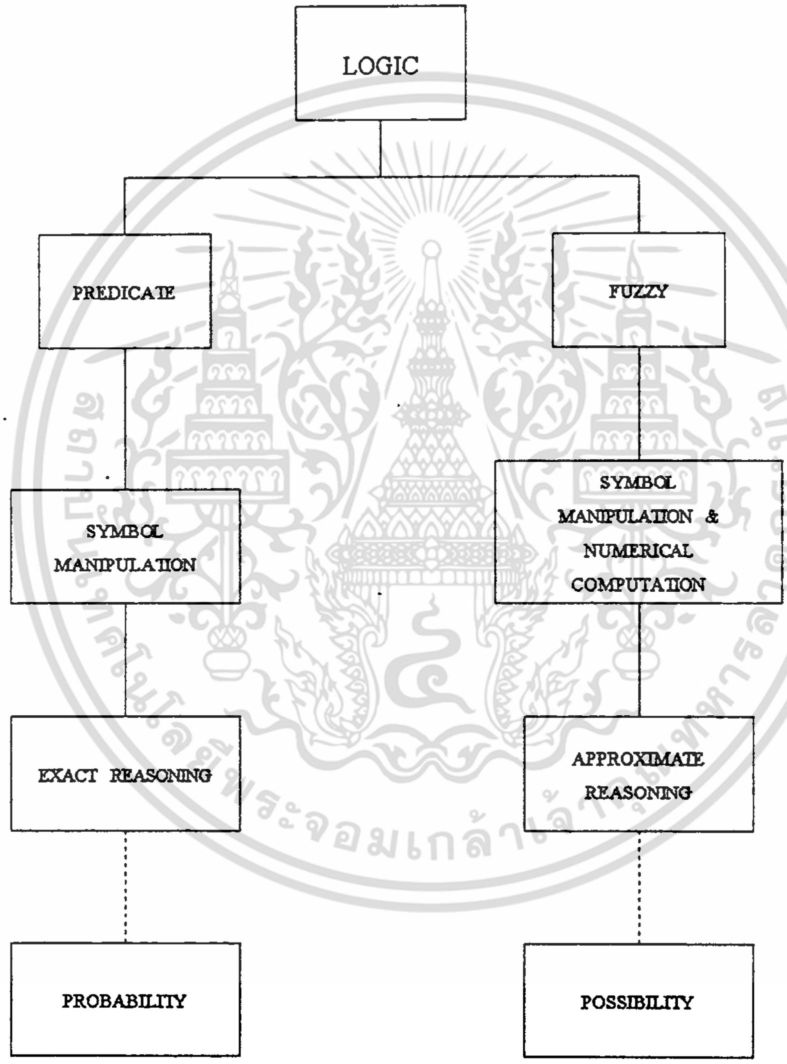
การประยุกต์ใช้งานอย่างหนึ่งของ Fuzzy Logic คือ การควบคุมระบบ การใช้ Fuzzy Controller จะทำให้ระบบมีการปรับตัวได้ค่าใกล้เคียงกับความต้องการของมนุษย์มากกว่ากฎขอบเขตของ Crisp Set (Hard-Boundary Crisp Rules) กฎของการใช้ Fuzzy Logic มาควบคุมดังกล่าวคือ มี Degree ค่าต่างๆ แบบต่อเนื่องทำให้เกิดตอบสนองของระบบในหลายๆ ลักษณะ นำมารวมกันได้เป็นผลลัพธ์ที่ต้องการ กระบวนการที่มีผลลัพธ์หรือข้อมูลไม่แน่นอน ประหยัดพลังงานโดยใช้กฎของ Common Sense และใช้คำสั่งที่ยึดหลักความเป็นจริง ต่างก็เป็นพื้นฐานของระบบควบคุมแบบฟัซซี่ ข้อมูลจาก Sensor ในระบบควบคุมในงานหลายๆ ประเภทมักจัดการโดยอาศัยเวียนรอบของการตัดสินใจของมนุษย์ (Decision Loop) เช่น นักบินอวกาศปรับตำแหน่งของดาวเทียมหรือปรับแนววงโคจรให้เหมาะสม หรือคนขับรถปรับเครื่องปรับอากาศที่ติดรถ เป็นต้น งานเหล่านี้สามารถหาคำตอบได้จากกฎเกณฑ์ต่างๆ มาใช้ตัดสินหรือหาค่าผลลัพธ์ของเซต คือมนุษย์ใช้เกณฑ์จากความเชี่ยวชาญจากการเรียนรู้และประสบการณ์ที่ผ่านมา ซึ่งเกณฑ์ต่างๆ เหล่านี้ไม่ใช่ Crisp แต่เป็นคุณสมบัติของการตัดสินใจของตัวแปรฟัซซี่รวมทั้งกฎต่างๆ ซึ่งพยายามทำให้มีค่าใกล้เคียงกับการตัดสินใจของมนุษย์

2.1 FUZZY LOGIC

ความต้องการที่จะใช้ประโยชน์จากระดับลอจิกหลายๆ ระดับ (Multi-Level Logic) ได้มีมาตั้งแต่สมัยอดีต แต่เดิมมีแค่ 2 ค่าระดับลอจิก (Two-Values Logic) ตัวอย่างเช่น ค่าจริงและค่าเท็จ แต่ต่อมาการพัฒนามาเป็น 3 ค่าระดับลอจิก (Three-Values Logic) ตัวอย่างคือ ค่าจริง(1), ค่าเท็จ(0) และ Neuter(1/2) ซึ่งเป็นค่าที่เป็นตัวแทนครึ่งหนึ่งของจริงและเท็จ จนในที่สุดก็มีการใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์จาก Multi Level Logic ได้สำเร็จ จึงเป็นต้นกำเนิดของทฤษฎีฟัซซี่เซตและทฤษฎีฟัซซี่ลอจิก โดยนิยามคำว่า " Fuzz " ว่าเป็นค่าลอจิกที่ต่อเนื่องระหว่าง 0 (Complete False) และ 1 (Complete True) ซึ่งเป็นเรื่องที่เกี่ยวข้องกับการตีค่าและตัดสินใจในเรื่องหรือข้อมูลต่างๆ ที่คลุมเครือ สามารถแสดงการเปรียบเทียบความแตกต่างของการทำนายค่าข้อมูลที่แน่นอนและการประมาณค่าของข้อมูลที่คลุมเครือได้ดังรูปที่ 1



รูปที่ 2.1 การเปรียบเทียบการทำนาย (Predicate) และ Fuzzy Logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุเบื้องต้นที่ทำให้ Fuzzy Logic ได้รับนิยมเชื่อถือคือ การตัดสินใจที่ถูกต้องแน่นอนจะไม่จำเป็นในงานบางประเภท ทำให้สิ้นเปลืองค่าใช้จ่าย ปัญหาเหล่านี้ต้องการการประมาณค่าก็เพียงพอในงานแล้ว ซึ่งจุดมุ่งหมายเบื้องต้นของ Fuzzy Logic จะมีแนวความคิดรวบยอดและเทคนิคที่เกี่ยวข้องกับการจัดการระบบจริงแบบประมาณค่า

ความพยายามที่จะแปลงข้อมูลจริงในกระบวนการเช่น แรงดันที่ Terminal ต่างๆ มาเป็นภาษาของระบบฟัซซี่ จะต้องผ่านกระบวนการ Fuzzification ในระบบคริฟเซตจะกำหนดเพียงว่า X เป็นสมาชิกของ A หรือไม่เป็นสมาชิกของ A แทนด้วยสูตรทางคณิตศาสตร์ที่เรียกว่าระดับการเป็นสมาชิก (Membership Function) เป็น 0 หรือ 1 แต่ฟัซซี่เซตจะยอมรับสมาชิกที่ถูกเพียงบางส่วนและผิดเป็นบางส่วน ไม่มีขอบเขตที่แน่นอน การประยุกต์ใช้ทฤษฎีฟัซซี่เซตจะต้องแสดงค่าระดับ (Degree) ซึ่งเป็นค่าที่เป็นไปได้ที่จะเป็นสมาชิกภายในเซต สามารถเขียนแทนด้วยสัญลักษณ์

$$\mu A(X) \rightarrow [0,1]$$

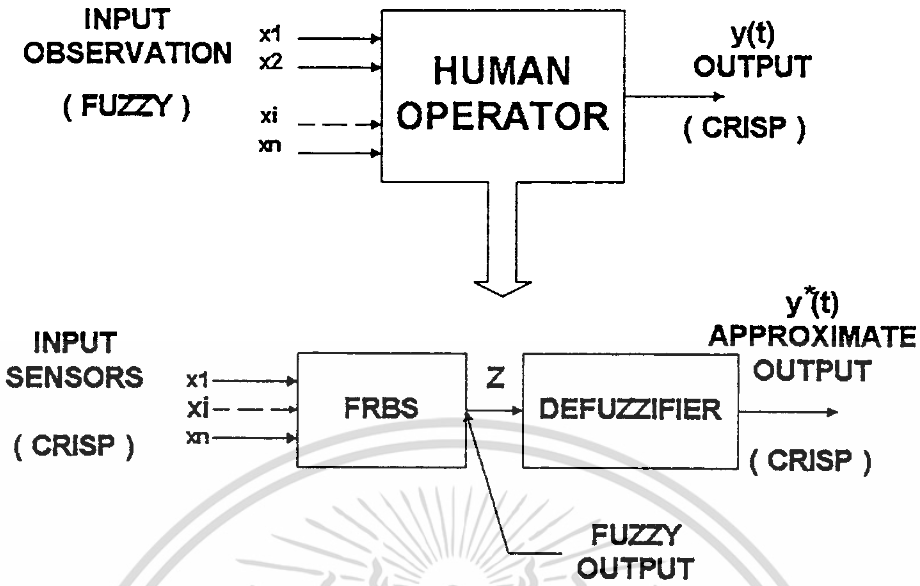
และมีค่าระดับความเป็นสมาชิกระหว่าง 0 ถึง 1 ค่า Membership Function สามารถเรียกว่า ค่าฟังก์ชันความเป็นไปได้ (Possibility Function) แต่ไม่ใช่ค่าความน่าจะเป็น (Probability Function)

2.2 FUZZY CONTROL

ระบบควบคุมฟัซซี่ (Fuzzy Control) มีแนวความคิดพื้นฐานมาจากทฤษฎีฟัซซี่ลอจิก (Fuzzy Logic) และทฤษฎีฟัซซี่เซต (Fuzzy Set) ฟัซซี่ลอจิก (Fuzzy Logic) มีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซต ซึ่งจะใช้เป็นตัวแทนของตัวปฏิบัติการที่มีความชำนาญเหมือนมนุษย์และอาศัยกฎพื้นฐานของฟัซซี่ รูปที่ 2.2 แทนการทำงานของระบบที่ใช้ Fuzzy ในการควบคุม

พบว่าข้อมูล Crisp หลายข้อมูลส่งตรงมาจาก Sensor ต่างๆ ผ่านกระบวนการ Fuzzification ซึ่งข้อมูลเหล่านี้จะถูกแปลงเป็น Fuzzy Membership Function ค่าจากการแปลงเหล่านี้ในเซตของฟัซซี่ นำมาใช้กฎ IF-THEN ซึ่งเปรียบได้กับระบบผู้เชี่ยวชาญ (Expert System) ทำให้ได้เอาท์พุท Z และก็ต้องแปลงกลับมาเป็น Crisp Values ดังเดิมโดยผ่านไปยังกระบวนการที่เรียกว่า Defuzzification โดยใช้ วิธีหาค่าน้ำหนักเฉลี่ย เช่นวิธี Centroidal Method เป็นต้น ทำให้ได้ค่าประมาณผลลัพธ์เป็นเอาท์พุท $y^*(t)$ จากเอาท์พุทจริงคือ $y(t)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTE : FRBS... FUZZY RULE-BASE SYSTEM

$y^*(t)$ is an approximation of $y(t)$

รูปที่ 2.2 คำจำกัดความของระบบควบคุมแบบฟัซซี่

2.3 การประยุกต์ใช้งานฟัซซี่ลอจิก (FUZZY LOGIC APPLICATION)

โดยทั่วไปจะมีลักษณะโครงสร้างพื้นฐานแบ่งได้ 3 ส่วนหลัก คือ

2.3.1 การเปลี่ยน Crisp Set เป็น Fuzzy Set (FUZZIER)

2.3.2 กฎการวินิจฉัย (INFERENCE MECHANISM THAT EMPLOYS RULES)

2.3.3 การเปลี่ยนฟัซซี่เป็น Crisp Values (DEFUZZIFIER)

2.3.1 FUZZIER

ในการใช้กับระบบจะต้องเปลี่ยนให้อยู่ในรูปของฟัซซี่โดเมน (Fuzzy Domain) เคลื่อนย้ายประมวลผลข้อมูลแล้วเปลี่ยนกลับให้อยู่ในรูป Crisp Domain ตามเดิมซึ่งเหมือนกับการกระทำทางอนาลอก (Analog) คือ เปลี่ยนให้อยู่ในรูปของโดเมนความถี่ (Frequency Domain) จากข้อมูลในโดเมนเวลา (Time Domain) เพราะว่าในกระบวนการในโดเมนความถี่จะง่ายกว่าโดเมนเวลา ในระบบฟัซซี่ (Fuzzy System) กฎพื้นฐานสามารถอธิบายการทำงานของระบบในรูปของฟัซซี่ได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราจะเปลี่ยนค่าอินพุทในรูปของ Crisp ไปอยู่ในฟัซซี่โดเมนมากกว่าจะเปลี่ยนจากฟัซซี่ไปอยู่ในรูป Crisp Domain

2.3.2 กฎการวินิจฉัย (INFERENCE)

สำหรับกฎการวินิจฉัยค่าอินพุทและค่าความจริง จะสนับสนุนเงื่อนไขสำหรับการสร้างส่วนกำหนดรูปแบบฟัซซี่คอมไพลเลอร์ จะสุ่มค่าของอินพุทและนำมาวินิจฉัยค่า เพื่อให้ได้ผลออกมาทางเอาท์พุทของระบบตามทฤษฎี ระบบจะรวมเอาค่าที่เป็นไปได้ทั้งหมดของอินพุท เพื่อนำมาวินิจฉัยและประมวลผล แต่จริงๆ แล้วการครอบคลุมค่าเหล่านี้ไม่จำเป็นในการใช้งานปกติ

กฎการวินิจฉัยนั้นได้มาจากการสร้างความสัมพันธ์ระหว่างอินพุทและเอาท์พุทขึ้นมา เพื่อนำไปสู่เอาท์พุทที่ต้องการ ซึ่งความสัมพันธ์ระหว่างอินพุทและเอาท์พุทก็คือ Rule ต่างๆ นั่นเอง นำค่า Rule ทุกกรณีที่สนใจมารวมกันจะได้เป็นค่า Membership Function เพื่อผ่านยังกระบวนการถัดไป

2.3.3 DEFUZZIER

เทคนิคและวิธีการในการเปลี่ยนฟัซซี่เป็น Crisp ดังจะกล่าวได้ดังต่อไปนี้

2.3.3.1 เทคนิค Maximizer เลือกค่าสูงสุดจากหลายๆ แบบมาเพียงหนึ่ง

เป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิก จากการกระทำหลายๆ แบบแล้วเลือกกระทำเพียงหนึ่งรูปแบบ ถ้าหากเกิดการกระทำที่มีค่า μ สูงสุดเท่ากัน 2 อย่าง จะต้องใช้รูปการแก้ปัญหาอีกลักษณะหนึ่ง คือ ใช้ค่าเฉลี่ยของค่าเอาท์พุท หรือเลือกการกระทำที่สัมพันธ์กับค่าระดับของระบบพื้นฐาน ถึงแม้เทคนิค Maximizer จะเป็นวิธีที่ง่ายที่สุด แต่ก็ไม่มีประสิทธิภาพเท่าที่ควร

2.3.3.2 ทฤษฎีค่าน้ำหนักเฉลี่ย

จะใช้ค่าเฉลี่ยของการกระทำหลังจากการกำหนดค่าระดับของระดับการเป็นสมาชิกไว้ล่วงหน้าแล้ว เป็นวิธีที่ง่ายและใช้การคำนวณเพียงเล็กน้อย แต่ก็ยังให้ค่าที่ไม่ค่อยชัดเจน เช่นเดียวกับเทคนิค Maximizer ที่เกิดความไม่ชัดเจน

2.3.3.3 กรรมวิธีค่าศูนย์กลาง

เป็นการแสดงค่าเอาท์พุทที่สัมพันธ์กับค่าจุดศูนย์กลางมวลของเอาท์พุท ในระดับที่ทำงานเพราะว่าเราไม่ใช้ค่าขอบของฟังก์ชันระดับความเป็นสมาชิก และจะไม่เกิดความไม่ชัดเจนอีกต่อไป กรรมวิธีค่าศูนย์กลางเป็นการคำนวณที่แน่นอนและเป็นการแก้ไขข้อบกพร่องที่เกิดขึ้นในวิธีอื่นๆ ค่าศูนย์กลางมักจะสัมพันธ์กับค่าเอาท์พุทหนึ่งค่า ผลที่ได้จะเป็นการกระทำอย่างหยาบๆ ภายในช่วงของเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

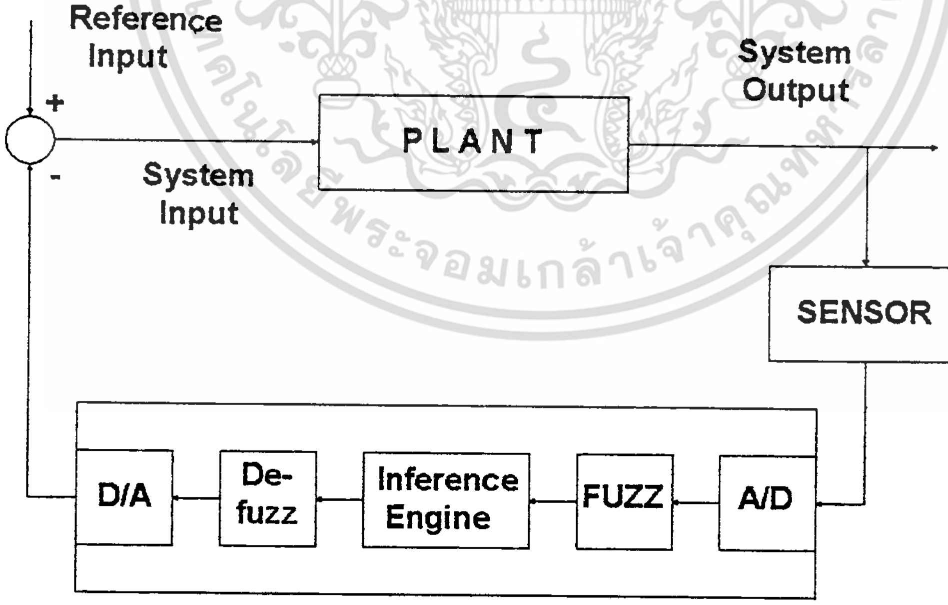
กฎทั่วๆ ไปจะต้องมีการปฏิบัติการเพียงจุดเดียวของแต่ละระบบ ในกรณีที่ต้องทำพร้อมๆ กันหลายกฎ ในการกระทำหนึ่งครั้ง จะต้องเกิดการซ้อนทับกัน (Overlap) ของค่าฟังก์ชันอินพุทของระดับการเป็นสมาชิก เพื่อแก้ไขความไม่ต่อเนื่องของเอาท์พุท ถึงแม้ว่าจะมีข้อบกพร่องแต่ก็เป็นวิธีที่ดีที่สุดในการรวมกัน (Combination) และการแปลงค่าฟัซซี่กลับคืน(Defuzzification)

กรรมวิธีนี้จะรวมค่าเอาท์พุทของการกระทำหลายๆ ค่า เป็นค่าค่าเดียวสำหรับใช้ในระบบ และค่าเอาท์พุทค่าเดียวนี้เป็นค่าน้ำหนักเฉลี่ยของศูนย์กลาง (Centroid) ของแต่ละฟังก์ชันระดับการเป็นสมาชิก

2.3.3.4 กรรมวิธีซิงเกิลตัน

เป็นเทคนิคพิเศษของแบบกรรมวิธีค่าศูนย์กลาง หรือจะเรียกอีกอย่างหนึ่งว่ากรรมวิธี Remaining Combination / Defuzzification วิธีนี้เป็นการนำค่าเอาท์พุทของฟัซซี่เซตมาใช้เป็นค่าเอาท์พุทค่าเดียว โดยใช้ค่าน้ำหนักเฉลี่ยจากการกระทำรวมหลายๆ อย่าง วิธีนี้ได้ค่าความถูกต้องน้อยกว่ากรรมวิธีศูนย์กลาง และยังคงต้องการการซ้อนทับกันของอินพุทฟังก์ชัน เพื่อเป็นการหลีกเลี่ยงความไม่ต่อเนื่องของเอาท์พุท ด้วยหลักการและการคำนวณที่ไม่ยุ่งยากและยังไม่มีใครคิดค้นวิธีใหม่และดีกว่านี้ กรรมวิธีนี้จึงน่าที่จะใช้แทนกรรมวิธีค่าศูนย์กลางได้ดีที่สุด

สามารถเขียนเป็น Block Diagram ของระบบควบคุมฟัซซี่ได้ดังรูปที่ 2.3



รูปที่ 2.3 แสดง Block Diagram ของระบบควบคุมแบบฟัซซี่ซึ่งแสดงให้เห็นถึง Fuzzifier (FUZZ), Defuzzifier (DE-Fuzz) และ Inference Engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการหา FUZZY CONTROL RULE

ในการออกแบบ Fuzzy Controller มีปัญหาสำคัญอันหนึ่งคือ การหา Fuzzy Control Rule ที่เหมาะสม ซึ่งวิธีที่สามารถนำมาใช้ได้จริงในงาน Process Control มีอยู่ 3 วิธี คือ

2.4.1 จากความรู้และประสบการณ์ของผู้เชี่ยวชาญ

โดยจะอ้างอิงกับความรู้และประสบการณ์ของวิศวกรควบคุม ซึ่งอันที่จริงแล้วสามารถกล่าวได้ว่า Fuzzy Control นั้น เป็น Application ที่ใช้ได้จริงอันแรกทางด้านระบบผู้เชี่ยวชาญ (Expert System) เลยทีเดียว

การออกแบบด้วยวิธีนี้เหมาะกับระบบที่มนุษย์เป็นผู้ควบคุมเป็นหลักสำคัญ Control Rule นั้นก็ได้มาจากการ List วิธีการควบคุมต่างๆ โดยวิศวกรควบคุม แต่ข้อเสียของวิธีนี้ก็คือ ส่วนมากผู้ควบคุมจะไม่สามารถถ่ายทอดออกมาได้เป็นลายลักษณ์อักษร โดยเฉพาะเมื่อระบบมีความซับซ้อนมาก

2.4.2 จากการจำลองลักษณะการทำงานของผู้ควบคุม

เมื่อทักษะของผู้ควบคุมเป็นสิ่งที่สำคัญ จึงจำเป็นมากที่จะหา Fuzzy Control Rule มาจากการจำลองลักษณะการทำงานของผู้ควบคุม ซึ่งวิธีนี้สามารถเป็นไปได้ที่จะแปลงการทำงานของผู้ควบคุมให้เป็นอินพุทและเอาท์พุทของ Controller

วิธีค่อนข้างคล้ายคลึงกับการจำลองกระบวนการ แต่ว่าการจำลองการทำงานของผู้ควบคุมนั้นจะง่ายกว่าการจำลองกระบวนการ เพราะอินพุทของระบบหาได้ง่ายกว่า แต่ในสถานการณ์จริงนั้นก็ควรจะรวมวิธีที่ 1 และ 2 เข้าด้วยกัน

2.4.3 จากการจำลองกระบวนการ

ในวิธีแรกนั้นมีพื้นฐานมาจากความคิดคร่าวๆ ที่เกี่ยวกับคุณลักษณะของกระบวนการ เช่น เอาท์พุทเพิ่มขึ้นเมื่ออินพุทเพิ่มขึ้น ส่วนวิธีที่ 2 จะใช้เฉพาะตัวแปรที่มีให้แก่ผู้ควบคุมกระบวนการ ซึ่งทั้ง 2 วิธีนี้จะใช้ได้ดีเฉพาะในกรณีที่ผู้ควบคุมมีบทบาทสำคัญในการควบคุมกระบวนการเท่านั้น

แต่ถ้าไม่ต้องการขึ้นอยู่กับผู้ควบคุมและต้องการให้ได้ผลการควบคุมดีกว่าด้วยก็จะมีการออกแบบอีกวิธีหนึ่งคือ การจำลองกระบวนการ ซึ่งเป็นวิธีที่ซับซ้อนกว่า การออกแบบวิธีนี้ได้มีการศึกษาวิจัยมาหลายครั้งด้วยกัน การจำลองกระบวนการในที่นี้จะเป็นการแสดงคุณลักษณะของ Process ออกมาด้วย Fuzzy Set โดยพิจารณาจากอินพุทตัวแปรสถานะและเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีแนวความคิดสองทางด้วยกันในการออกแบบ Fuzzy Controller จากแบบจำลองฟัซซี่

-ทางแรก คือการออกแบบให้ Control Rule นั้น ทำการ Compensate ลักษณะที่ไม่ต้องการของกระบวนการเพื่อให้ได้ตามเป้าหมายที่ต้องการ

-ทางที่สองคือ เป็นไปตามทฤษฎีของ Optimal Control ซึ่งจะให้โครงสร้างและ Parameter ของ Control Rule ทำให้ระบบที่มี Fuzzy Controller ควบคุมอยู่นั้นเป็นไปตามเป้าหมายที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ภาษา VHDL

บทนี้จะแนะนำประวัติความเป็นมาอย่างย่อๆ ของภาษา VHDL(VHSIC Hardware Description Language)

3.1 ภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language(VHSIC ย่อมาจาก Very High Speed Integrated Circuit) เป็นภาษาคอมพิวเตอร์ระดับสูง(High Level Language) ซึ่งใช้อธิบายการทำงานของระบบเชิงตัวเลข สามารถใช้อธิบายฟังก์ชันการทำงานได้หลายๆ ระดับ ตั้งแต่ระดับผังงานจนกระทั่งจนถึงระดับวงจรเกท ความซับซ้อนของระบบสามารถจะเขียนได้ตั้งแต่ระดับวงจรเกทประกอบกันจนเป็นระบบที่สมบูรณ์ รูปแบบของภาษา VHDL นั้นจะประกอบไปด้วย 2 ส่วนใหญ่ๆ ได้แก่ ส่วนของ Sequential Language และ Concurrent Language การโปรแกรมด้วยภาษา VHDL สามารถจะเขียนได้ทั้ง 2 รูปแบบรวมกัน เพราะในการทำงานของระบบใดๆ ย่อมจะมีการทำงานในแบบ Sequential และ Concurrent อยู่รวมกัน นอกจากนี้ตัวภาษา VHDL ยังสามารถอธิบายถึงการเชื่อมต่อระหว่างระบบย่อยๆ เข้าด้วยกันเพื่อให้เป็นระบบใหญ่ได้ ตัวภาษา VHDL นอกจากจะกำหนดรูปแบบไวยากรณ์ (Syntax) ของตัวภาษาแล้ว ยังมีการตรวจสอบความหมายของตัวภาษาว่าจะทำการจำลองการทำงานได้หรือไม่ เพราะว่าโปรแกรมที่เขียนโดย VHDL ต้องผ่านการจำลองการทำงานเพื่อตรวจสอบดูการทำงาน ฉะนั้นในการ Compile จะมีการตรวจสอบทั้ง Syntax และ Simulation Semantics อย่างไรก็ตามตัวภาษานั้นถึงจะมีความซับซ้อนมากมายในรูปแบบและกฎเกณฑ์ของภาษาแต่ในการเรียนรู้เพียงบางส่วนของตัวภาษาก็สามารถนำมาใช้งานได้โดยไม่จำเป็นจะต้องศึกษารายละเอียดทั้งหมด เนื่องจากตัวภาษา VHDL ออกแบบมาให้ใช้ออกแบบได้ตั้งแต่วงจรที่มีขนาดเล็กจนถึงวงจรที่มีขนาดใหญ่ซับซ้อน

3.2 ประวัติความเป็นมาของภาษา VHDL

ความต้องการภาษานี้เริ่มจากโครงการ VHSIC ของ Department of Defense(DOD) ของสหรัฐอเมริกาเนื่องจากมีบริษัทที่สร้าง VHSIC Chip หลายบริษัทได้ร่วมโครงการที่จะพัฒนาในขณะนั้นๆ หลายๆ บริษัทใช้ภาษา VHDL ซึ่งแตกต่างกัน ในการอธิบายการทำงาน Chip ของตนด้วยเหตุนี้ทำให้เกิดความแตกต่างแต่ละบริษัทไม่สามารถแลกเปลี่ยนเทคโนโลยีให้กันและกันได้ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DOD เกิดปัญหาในการที่จะพัฒนาและซ่อมบำรุงในภายหลัง จึงเกิดความต้องการภาษา VHDL ซึ่งเป็นมาตรฐานในการที่จะอธิบายถึงตัวที่ทำการออกแบบนั้นๆ ดังนั้น DOD จึงมอบให้บริษัท IBM, Texas Instruments และ Intermetics 3 บริษัทแรกร่วมกันพัฒนาและกำหนดมาตรฐานของ VHDL ขึ้นมาในปี 1983 หลังจากนั้น VHDL Version 7.2 ได้ทำการพัฒนาและออกเผยแพร่ต่อสา-ธารณชนในปี 1983 ได้รับความสนใจอย่างมากในอุตสาหกรรมโดยเฉพาะอย่างยิ่งบริษัทที่ทำ VHSIC Chip จากผลสำเร็จนี้ทำให้เกิดมาตรฐาน IEEE ของ VHDL ในปี 1986 ภายหลังจากนั้นก็มีการพัฒนาขยายขีดความสามารถของตัวภาษา VHDL เพิ่มขึ้นจากในวงการอุตสาหกรรม, มหาวิทยาลัย และ DOD ก็มีการปรับปรุงและจดมาตรฐานใหม่ IEEE ปี 1987 อีกครั้งหนึ่งซึ่งเป็นที่รู้จักกันในชื่อของ IEEE STD 1076-1987 หลังจากกันยายน 1988 บริษัทใดๆ ที่ทำการพัฒนา ASIC CHIP ใน DOD ของอเมริกาต้องส่งตัว VHDL Model พร้อมกับ TEST BENCH ตามมาตรฐานที่ได้กำหนดเอาไว้

3.3 ความสามารถของภาษา VHDL (CAPABILITY)

หัวข้อดังต่อไปนี้คือความสามารถหลักๆ ของตัวภาษา VHDL

3.3.1 ตัวภาษา VHDL สามารถใช้เป็นสื่อกลางในการแลกเปลี่ยนสื่อสารระหว่างผู้ผลิต Chip กับผู้ออกแบบ (CAD Tools)

3.3.2 ใช้เป็นสื่อกลางในการแลกเปลี่ยนสื่อสารระหว่าง CAE และ CAD Tools เช่น Source Code ของภาษา VHDL สามารถใช้ Compile โดยใช้ Compile&Simulator ได้หลายตัวแตกต่างกัน

3.3.3 ภาษา VHDL สนับสนุนการออกแบบ Top-Down Design และBottom-Up Design หรือผสมกันทั้ง 2 แบบ

3.3.4 ตัวภาษา VHDL เป็น Generic คือ ไม่อิงเทคโนโลยีอันใดอันหนึ่ง (แต่สามารถอิงเทคโนโลยีใดก็ได้และในขณะเดียวกัน ก็สามารถสนับสนุนหลายๆ เทคโนโลยี)

3.3.5 สนับสนุนการออกแบบทั้งระบบ Synchronous และ Asynchronous

3.3.6 สนับสนุนการออกแบบระบบเชิงตัวเลขในหลายๆ เทคนิค เช่น Finite State Machine Algorithmic หรือ Boolean Equation

3.3.7 ตัวภาษา VHDL สามารถอ่านและทำความเข้าใจได้โดยมนุษย์(Human readable)

3.3.8 ภาษา VHDL เป็นมาตรฐานรับรองโดย IEEE และ ANSI ทำให้ Model ที่ออกแบบโดย VHDL สามารถเคลื่อนย้าย (Portable) ไปยังระบบใดๆ ก็ได้และสามารถนำกลับมาใช้ใหม่ได้ (Reuse)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.9 ภาษา VHDL สนับสนุนรูปแบบการเขียนถึง 3 รูปแบบ ได้แก่ Behavioral Style, Structural Style, Dataflow Style หรือสามารถเขียนรวมกันทั้ง 3 รูปแบบ (Mix Style)

3.3.10 สนับสนุนการออกแบบขนาดใหญ่โดยใช้ความสามารถของ Component, Function Procedure และ Package

3.3.11 ไม่จำเป็นต้องศึกษา Software Simulator เพราะ Simulation Model สามารถเขียนได้โดยใช้ภาษา VHDL เช่นกัน

3.3.12 สามารถเขียน Model ได้ขนาดไม่จำกัด ไม่มีข้อจำกัดในตัวภาษาเรื่องขนาดของ Model (ขึ้นอยู่กับ Software Tools)

3.3.13 สามารถอธิบาย Parameter ที่เกี่ยวกับฟังก์ชันทางด้านเวลา เช่น Propagation Delay, Min-Max Delay, Setup, Holding Time, Spike Detection สามารถอธิบายได้ภายในตัวภาษา

3.3.14 GENERICS ช่วยให้เราสามารถสร้าง Parameter ของตัวที่ทำการออกแบบ

3.3.15 Model ที่สร้างด้วยภาษา VHDL นั้น ไม่เพียงแต่จะอธิบายฟังก์ชันการทำงานเท่านั้นแต่ยังสามารถอธิบายถึงรายละเอียดของตัว Model เช่น Total Area และ Speed ของ Model

3.3.16 ภาษา VHDL เป็นมาตรฐานใช้โดยบริษัทและผู้ออกแบบหลายๆ แห่ง ฉะนั้นจึงเป็นการง่ายที่จะทำความเข้าใจ ถึงแม้ว่าจะมาจากแหล่งต่างๆ

3.3.17 Model ที่สร้างขึ้นสามารถจำลองการทำงานได้ เพราะว่าตัวแปลภาษาได้ตรวจสอบไวยากรณ์ทางด้าน Simulation Semantics ไว้ด้วย

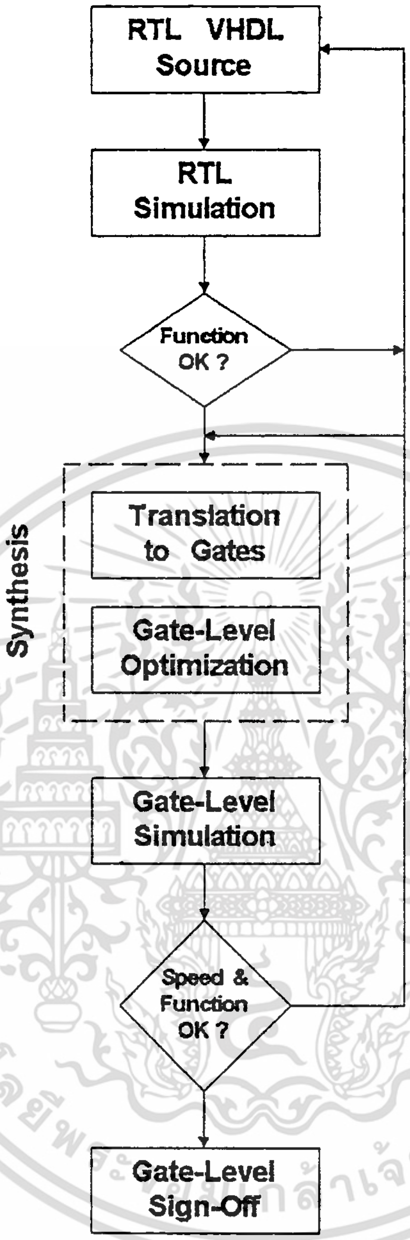
3.3.18 การอธิบาย Model ด้วย Behavioral Style สามารถสังเคราะห์ไปเป็นระดับวงจรเกทได้ ถ้าทำตามกฎของ Synthesis Guideline

3.3.19 มีความสามารถที่ให้เราออกแบบข้อมูลชนิดใหม่ๆ ได้ ทำให้ VHDL Model เป็นการออกแบบในระดับสูงที่ไม่ต้องคำนึงถึงว่าจะสร้างตัว Model นั้นขึ้นมาได้อย่างไร

3.4 กระบวนวิธีในการออกแบบโดยภาษา VHDL

ขั้นตอนในการออกแบบนั้นสามารถแสดงได้ดังโฟลว์ชาร์ตดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ระเบียบวิธีในการออกแบบโดยใช้ Hardware Description Language

3.5 หลักการสร้างโมเดลโดยใช้ภาษา VHDL (General VHDL Modeling Principles)

3.5.1 VHDL เป็นภาษาที่ใช้สำหรับอธิบายการทำงานของฮาร์ดแวร์ในรูปแบบฟอร์มที่อ่านเข้าใจได้ (Human Readable) ซึ่งช่วยในการสร้างและออกแบบวงจรของระบบเชิงตัวเลข (Digital Hardware System, Circuit Board) และอุปกรณ์ต่างๆ อาจใช้อธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบทั้งระบบหรือ อธิบายเพียงบางส่วน ซึ่งอยู่ในรูปของแผนผังอุปกรณ์ (Component Block) จากนั้นก็ทำการจำลองการทำงาน โดยที่ตัวที่ทำการออกแบบนั้นยังไม่ได้สร้างขึ้นจริงหรือเพียงแต่อยู่ในรูปของคำอธิบายเท่านั้น (Textual Format) หลังจากจำลองการทำงานจนได้ตามที่ต้องการจึงนำไปทำการสังเคราะห์เพื่อให้ได้วงจรเกทต่อไป

ประโยชน์จริงๆ ของการใช้ VHDL เป็น Tool แทนการสร้างระบบต้นแบบขึ้นมาจริงแบบเมื่อก่อน ก็คือ เราสามารถอธิบาย Product Idea, Product Proposal, Product Specification เป็นลักษณะในรูปของแฟ้ม Text จากนั้นก็นำไป Compile เพื่อดูผังเวลาการทำงาน จากนั้นก็ทำการ Refine แก้ไขจนกว่าจะได้ Specification ตามต้องการ เมื่อผลิต-ภัณฑ์ได้ผลตามที่ต้องการแล้วจึงนำไปสู่การสังเคราะห์เพื่อให้ได้ผังวงจรในระดับวงจรเกท แล้วนำไปสร้างเป็นระบบต้นแบบจริงต่อไป ซึ่งระบบต้นแบบที่สร้างนั้นทำงานได้จริงเพราะได้ทำการจำลองการทำงานมาเรียบร้อยแล้ว เป็นการลดเวลาและค่าใช้จ่ายในการสร้างระบบต้นแบบได้มาก

เนื่องจากว่าภาษา VHDL เป็นภาษาที่มีประสิทธิภาพสูง เราจึงใช้ VHDL อธิบายฮาร์ดแวร์เพราะว่ามีข้อดี 2 ประการ คือ

3.5.1.1 เข้าใจได้ง่าย (Easy To Understand)

3.5.1.2 สามารถแก้ไขได้ง่าย (Modificable)

การเข้าใจได้ง่ายมีประโยชน์ต่อใครก็ได้ซึ่งมีความจำเป็นที่จะต้องอ่าน Code ที่ได้ออกแบบมาแล้วโดยไม่จำเป็นต้องให้ผู้ออกแบบมาอธิบายให้ฟัง ตัวภาษา VHDL อธิบายการทำงานภายในตัวอยู่แล้ว ส่วนอีกประการหนึ่งก็คือ ความต้องการในการเปลี่ยนแปลง Hardware ที่ได้ออกแบบแล้วก็คือว่า หลังจากทดสอบแล้วพบข้อที่ผิดพลาดซึ่งต้องแก้ไขหรือว่าระหว่างพัฒนามีการเปลี่ยนแปลงความต้องการของระบบหรือต้องการเพิ่มการทำงานบางส่วน

กรณีอื่นๆ ที่ VHDL มีประโยชน์ก็คือ ตัวภาษานั้นสนับสนุนหลักการต่างๆ ให้เขียน แก้ไข และบำรุงรักษาระบบเชิงตัวเลขที่มีความซับซ้อนเป็นไปได้อย่างรวดเร็ว และมีประสิทธิภาพ และหลักการดังที่กล่าวมีดังนี้

- Top Down Design
- Modularity
- Abstraction
- Information Hiding
- Uniformity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะอธิบายประโยชน์ของหลักการต่างๆ ในหัวข้อต่อไป และแสดงให้เห็นว่า VHDL นั้นช่วยในการออกแบบระบบเชิงตัวเลขขนาดใหญ่และซับซ้อนนั้น ให้อ่านเข้าใจได้ง่ายและแก้ไขได้ง่ายอย่างไร

3.5.2 Top Down Design

ในการพัฒนาระบบเชิงตัวเลขขนาดใหญ่ที่มีความซับซ้อน วิศวกร หรือผู้ออกแบบมักจะมองระบบที่จะออกแบบให้อยู่ในรูปของแผนผังงานเสียก่อน ก่อนที่จะย่อยระบบที่จะออกแบบให้ลึกถึงรายละเอียดต่อไป ซึ่ง VHDL นั้นอนุญาตให้

3.5.2.1 อธิบายการทำงานของแต่ละส่วน

3.5.2.2 วิเคราะห์การทำงาน

3.5.2.3 จัดการแก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามที่ต้องการ ก่อนที่จะทำการออกแบบให้ละเอียดลึกลงไปในขั้นตอนต่อไป การแก้ไขในขั้นตอนนี้จะทำให้ลดค่าใช้จ่ายกว่าการไปแก้ไขในช่วงของการพัฒนาในระดับสร้าง Silicon Chip

3.5.3 Modularity

Modularity คือหลักการในการแยกส่วน (Partitioning) ฮาร์ดแวร์ออกเป็นส่วนย่อยเล็กๆ ลงไปซึ่งปกติการทำงานของฮาร์ดแวร์ส่วนใหญ่ต้องประกอบด้วยฮาร์ดแวร์ส่วนย่อยๆ ลงไป ซึ่งวงจรระดับเกททั้งหมดจะอยู่ในรูปๆ เดียว (Flatten Design) หลังจากนั้น จะตัดออกเป็นส่วนย่อยๆ เล็กลงมา เมื่อเราออกแบบโดยใช้ VHDL หน้าที่การทำงานของแต่ละส่วนสามารถอธิบายได้โดย Module ของรหัส (คล้าย Function หรือ Procedure) ซึ่งแสดงการทำงานของส่วนย่อยนั้นอย่างชัดเจน ซึ่งการแยกระบบที่จะออกแบบใหญ่ๆ ออกเป็นส่วนย่อยๆ นี้ทำให้ง่ายต่อการจัดการและง่ายต่อการทำความเข้าใจ

ตัวภาษา VHDL ประกอบขึ้นมาด้วย Language Building Block ซึ่งประกอบไปด้วย 75 Reserved Word และมากกว่า 200 Combination Words

3.5.4 Abstraction

คำนิยามของระบบที่จะออกแบบ จะอธิบายการทำงานของตัวระบบมากกว่าที่จะอธิบายถึงว่าจะพัฒนาตัวระบบนั้นอย่างไร หลักการนี้จะมีความสัมพันธ์อย่างใกล้ชิดกับหลักการ Modularity

อีกวิธีการหนึ่งซึ่งแสดงถึงการอธิบายการทำงานของระบบโดยใช้ VHDL ในหลายๆ ระดับของการนิยาม ROM(Read Only Memory) อธิบายโดยใช้ภาษาระดับสูงแสดงถึง Address ต่างๆ ซึ่งเก็บ DATA ไว้ใน Address นั้นๆ ที่ระดับนี้ไม่ต้องสนใจถึง Address Line, Data Line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ Control Line เราสามารถเพ่งจุดสนใจไปที่ขนาดของ DATA โดยไม่ต้องคิดถึงสัญญาณควบ - คุมต่างๆ มากมายภายใน เพราะว่าส่วนนั้นจะถูกจัดการเองในระดับต่ำลงมา ในระดับล่างลงมาเราสามารถอธิบายการทำงานของสัญญาณแต่ละเส้นภายใน ROM ในการจัดการสัญญาณภายในทุกเส้นภายในการที่จะอ่านข้อมูลหรือโปรแกรมข้อมูลใน ROM ถ้าต้องการเปลี่ยนค่าข้อมูลภายใน ROM เราควรขึ้นมาแก้ไขในระดับที่สูงขึ้นมา จะทำให้ง่ายกว่าในการที่จะควบคุมสัญญาณภายใน ซึ่งเราจะเห็นว่าในแต่ละระดับมีความเหมาะสม แตกต่างกันไป และตรงจุดนี้เองทำให้ระบบที่เราออกแบบง่ายต่อการแก้ไขโดยการใช้ประโยชน์ของ Abstraction

3.5.5 Information Hiding

เมื่อเราทำการเขียนรหัสต้นแบบของภาษา VHDL ขึ้นมาเพื่ออธิบายการทำงานของฮาร์ดแวร์ตัวหนึ่ง บางครั้งเราอาจต้องการที่ซ่อนรายละเอียดการพัฒนา Module นั้นๆ โดยไม่ต้องการให้ส่วน Module อื่นๆ รู้การทำงานภายใน Information Hiding มีประโยชน์ก็คือ ทำให้ระบบที่ออกแบบนั้น สามารถจัดการได้ง่ายและสามารถอ่านและทำความเข้าใจได้ง่ายกว่า หลักการนี้จะใช้สนับสนุนการ Abstraction คือ จะสนใจรายละเอียดในการใช้งานมากกว่าจะสนใจว่าระบบนั้นจะถูกสร้างขึ้นมาอย่างไรมีวงจรอย่างไรบ้างเป็นต้น การซ่อนรายละเอียดภายใน Module ทำให้ความสนใจของผู้ออกแบบสนใจไปในส่วนที่สำคัญมากกว่า ในส่วนที่ไม่น่าสนใจจะซ่อนไว้และเข้าถึงไม่ได้ เช่น คำอธิบายของ Nand Gate นั้นจะถูกปิดบังเอาไว้จากคนที่เขียนอธิบาย Flip-Flop คนที่เขียนอธิบายการทำงานของ Flip-Flop ไม่ต้องสนใจเลยว่า Nand Gate จะทำงานอย่างไรจะต่อกันภายในอย่างไร โดย Nand Gate สามารถเขียนขึ้นมาแล้วคอมไพล์เก็บไว้ใน Library ผู้ที่ออกแบบ Flip-Flop ระดับสูงขึ้นมาเพียงแต่ต้องรู้ว่าจะเชื่อมต่อ Input/Output ของ Nand Gate มาใช้งานได้อย่างไร และประโยชน์อีกอย่างของ Information Hiding ก็คือป้องกันข้อมูลภายในในกรณีที่แจกจ่าย VHDL Module ไปยังที่อื่นๆ เช่นส่งไปให้อีกบริษัทใช้พัฒนาร่วมกับ VHDL อื่นๆ โดยเป็นการแจกจ่าย อาจส่งไปแค่ VHDL ที่คอมไพล์แล้วไม่ต้องส่งตัวรหัสต้นแบบไป ทำให้เราป้องกันทรัพย์สินทางปัญญาได้ในอีกระดับหนึ่ง

3.5.6 Uniformility

Uniformility เป็นหลักการอีกอย่างที่ช่วยในการอธิบายระบบด้วยภาษา VHDL นั้นอ่านได้ง่ายขึ้น Uniformility หมายถึงการสร้าง Module ของรหัสในลักษณะคล้ายกัน โดยใช้ตัวภาษา VHDL Building Block ทำให้เกิดการเขียนรหัสต้นแบบที่ดีอย่างเช่น มีการใช้ย่อหน้า มีการใช้ Comment อธิบายเป็นต้น ทำให้การพัฒนา Module อ่านและทำการเข้าใจง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ระดับของการอธิบายระบบ (Level of Abstraction)

การออกแบบโดยใช้ภาษา VHDL Description ก็มีหลายระดับแต่ละระดับก็เหมาะสมและมีข้อเสียต่างกันไปซึ่งรูปแบบของภาษา VHDL มี 3 ระดับดังนี้

3.6.1 Behavioral Description

3.6.2 Dataflow Description

3.6.3 Structural Description

3.6.1 Behavioral Description

เป็นระดับที่เป็นนามธรรมที่สุด โดยภาษา VHDL นั้นมีรูปแบบนี้เป็นการอธิบายการทำงานของระบบคล้ายๆ กับโปรแกรมคอมพิวเตอร์ คือ มี Procedural Form และ Function Form ซึ่งไม่ต้องกล่าวถึงรายละเอียดและการสร้างตัวระบบเลย การใช้ภาษา VHDL รูปแบบนี้เหมาะสำหรับอธิบายระบบที่มีความซับซ้อนมากๆ โดยนักออกแบบไม่ต้องทราบเลยว่าระบบจะประกอบด้วยอุปกรณ์อะไรบ้าง ทำให้ VHDL รูปแบบนี้เหมาะกับผู้ที่ไม่ใช่วิศวกรและผู้ใช้งานทั่วๆ ไปและเป็นการอธิบายตัวระบบที่ดีไปในตัว (Good Documentation)

3.6.2 Dataflow Description

เป็นระดับที่ต่ำกว่า Behavioral คือแสดงการไหลของข้อมูลภายในหน่วยย่อยๆ ของระบบ โดยตัวภาษาจะอธิบายการสื่อสารข้อมูลระหว่างหน่วยย่อยๆ นั้น ผู้ออกแบบจะต้องรู้การทำงานของหน่วยย่อยๆ แต่ละหน่วยและรูปแบบของข้อมูลที่จะส่งไปมา การอธิบายในระดับนี้ไม่เหมาะกับผู้ใช้ทั่วๆ ไป แต่จะเหมาะกับระดับช่างเทคนิค หรือวิศวกรเท่านั้น การเขียนอธิบายแบบนี้จะเสียเวลามากกว่าแบบ Behavioral ก็จริงแต่ข้อดีก็คือสามารถที่จะสังเคราะห์ออกมาได้ดีกว่าเพราะแสดงรายละเอียดของระบบออกมาในระดับผังงานซึ่งมากกว่าแบบ Behavioral ซึ่งไม่แสดงอะไรออกมาเลย แต่อย่างไรก็ตามเราจะใช้รูปแบบ Dataflow ในการอธิบายการทำงานของระบบโดยการไหลข้อมูลระหว่าง Register และ Bus

3.6.3 Structural Description

เป็นการอธิบายฮาร์ดแวร์ในระดับต่ำสุด โดยอธิบายถึงการเชื่อมต่ออุปกรณ์แต่ละตัวเพื่อที่จะให้ได้เป็นระบบขึ้นมา ระดับนี้เป็นระดับที่สังเคราะห์ได้ง่ายที่สุดเพราะแสดงรายละเอียดของระบบได้มากที่สุด อย่างไรก็ตามภาษา VHDL ในระดับนี้อธิบาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์ได้ชัดเจนที่สุด บางครั้งเรียกว่า Hardware Description at Gate Level ระดับนี้เข้าใจได้ยากที่สุด แต่เหมาะสำหรับการจำลองการทำงานที่ต้องการผังเวลาที่ละเอียดที่สุด

3.7 สรุป

3.7.1 ภาษา VHDL คือภาษาที่ออกแบบมาเฉพาะเพื่อที่ใช้อธิบายการทำงานของฮาร์ดแวร์ให้อยู่ในรูปที่สามารถอ่านทำความเข้าใจได้ (Human Readable) สามารถอธิบายได้ถึงการจัดระบบและการทำงานของวงจรระดับเกท, วงจรระดับ Board และอุปกรณ์ต่างๆ

3.7.2 เหตุผลที่ทำให้ภาษา VHDL ใช้ในการออกแบบและจำลองการทำงานของผลิต-ภัณฑ์ตัวหนึ่งซึ่งยังไม่ได้สร้างจริงๆ เพื่อดูการทำงานก่อนลงมือสร้าง หรืออาจใช้เป็นตัวแทนแนวคิดในผลิตภัณฑ์นั้นๆ มีดังนี้

3.7.2.1 ภาษา VHDL อนุญาตให้เราออกแบบ, จำลองการทำงาน และทดสอบระบบ โดยใช้รูปแบบของภาษาระดับสูงจนถึงระดับวงจรเกท

3.7.2.2 ภาษา VHDL ถ้าเราเขียนตามรูปแบบของ VHDL นั้นไปทำการสร้างวง-จรได้โดยใช้ VHDL Synthesis Guide จะทำให้เราสามารถใช้รหัสภาษา VHDL นั้นไปทำการสร้างวงจรได้โดยใช้ VHDL Synthesis Tools

3.7.2.3 เพราะว่าภาษา VHDL เป็นภาษาที่กำหนดเป็นมาตรฐาน IEEE 1076-1987 วิศวกร หรือผู้ออกแบบทั่วๆ ไป สามารถใช้ภาษานี้ในการพัฒนาได้เหมือนกัน ลดปัญหาความเข้ากันไม่ได้ลงไป

3.7.3 VHDL มีคุณสมบัติที่ดีที่ทำให้เราสามารถเขียนและแก้ไขระบบเชิงตัวเลขที่มีขนาดใหญ่และซับซ้อนได้อย่างสะดวกรวดเร็วและมีประสิทธิภาพ ดังนี้คือ

3.7.3.1 Top Down Design วิธีการนี้ ทำให้เราสามารถอธิบายการทำงานของระบบได้ในลักษณะของ Block ใหญ่ๆ จากนั้นทำการวิเคราะห์จำลองการทำงานและแก้ไขให้ได้คุณสมบัติตามที่เราต้องการ ณ. ระดับ Block ก่อนที่จะลงลึกในระดับต่ำต่อไป

3.7.3.2 Modularity วิธีการที่แยกส่วน (หรือการประกอบส่วนย่อยๆ ขึ้นมา) ระบบที่เราออกแบบออกเป็นส่วนย่อยๆ เล็กๆ ออกมา

3.7.3.3 Abstraction เป็นรายละเอียดใน Module ซึ่งอธิบายการทำงานของ Module มากกว่าที่จะอธิบายถึงการพัฒนาและการสร้าง Module นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038377

บทที่ 4
VLSI เบื้องต้น

ในการออกแบบวงจรรวม VLSI จะสามารถแบ่งออกได้เป็น 2 ลักษณะคือ แบ่งตามเทคโนโลยีของการผลิตเป็น Silicon Technology และ GaAs Technology หรือ แบ่งตามชนิดของวงจรจะเป็น Analog IC และ Digital IC ซึ่งวิธีการออกแบบจะมีความแตกต่างกันไป คือ

-Analog IC จะมีจำนวนทรานซิสเตอร์ต่อวงจรไม่มาก ดังนั้นวงจรจึงไม่ค่อยซับซ้อน วงจรAnalog มีทั้ง Passive และ Active Element ซึ่งค่าของ Element เหล่านี้จะต้องมีความแม่นยำ (Accuracy) อย่างยิ่ง จึงจะสามารถ ทำให้ทำงานได้ดีตามที่ต้องการ ด้วยเหตุผลนี้การออกแบบจึงต้องคำนึงถึงค่า Parasitic ที่เกิดขึ้น และต้องคำนึงถึงการเปลี่ยนแปลงของค่า Element ต่างๆ จากสภาวะแวดล้อมด้วย

-Digital IC เป็น IC ที่มีจำนวนทรานซิสเตอร์วงจรสูงมาก วงจรมีความซับซ้อนมาก ขั้นตอนในการออกแบบจึงเป็นลักษณะของ Design Automation

4.1 ขั้นตอนในการออกแบบ (THE DESIGN CYCLE)

จากรูปที่ 4.1.1 แสดงถึงขั้นตอนในการออกแบบ ซึ่งจะเห็นว่า มีการตรวจสอบ และ Simulation เป็นระยะเพื่อที่จะทำให้วงจรที่ออกแบบมีความถูกต้อง และง่ายต่อการแก้ไขมากกว่าที่จะแก้ไข เมื่อผ่านมาหมดทุกขั้นตอนแล้ว ซึ่งค่า เวลา /ค่าใช้จ่าย จะเป็นฟังก์ชันที่ไม่เชิงเส้นดังรูปที่ 4.1.2 กล่าวคือ จะต้องพยายามที่จะตรวจสอบทุกๆ ขั้นตอนของการออกแบบเพื่อที่จะให้ได้ผลลัพธ์ที่สมบูรณ์ที่สุด

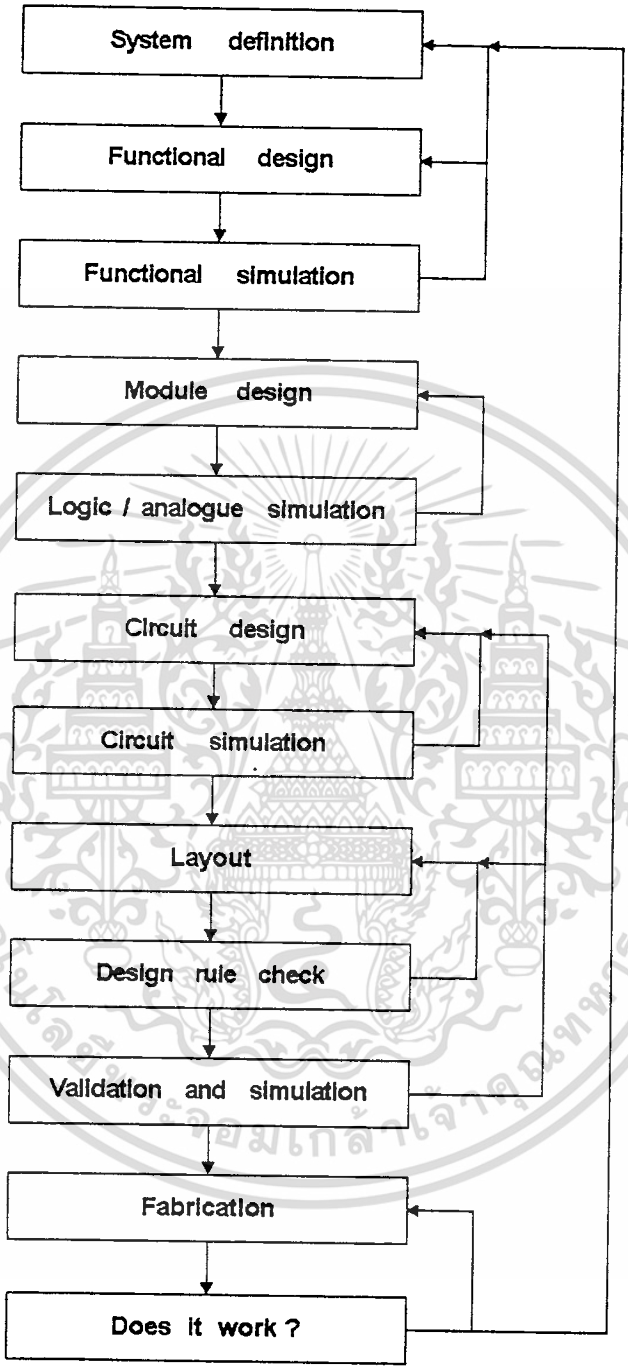
4.2 หน้าที่ในการออกแบบ (THE DESIGN TASKS)

จากรูปที่ 4.1.1 สามารถอธิบายได้ดังนี้

System Definition เป็นส่วนที่บางครั้งง่ายที่สุด หรือบางครั้งก็เป็นส่วนที่ยากที่สุดของการออกแบบ กล่าวคือเป็นการอธิบายถึงพฤติกรรมของระบบต่ออินพุทที่เป็นไปได้ทั้งหมด

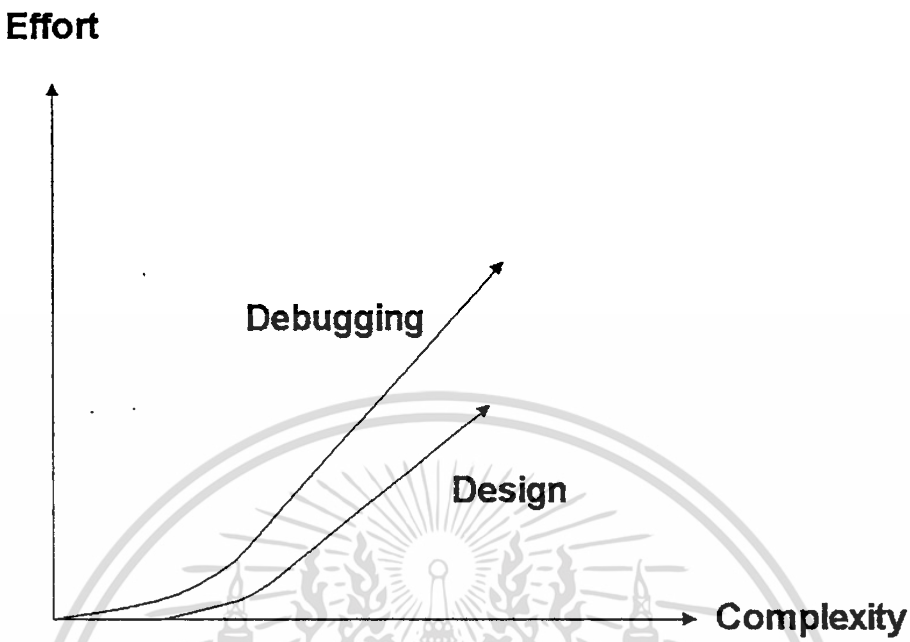
Functional Design เป็นการเปลี่ยนข้อมูลจาก System Definition (เป็นข้อมูลที่คนเขียนอยู่ในรูปเอกสาร) ให้อยู่ในเทอมของตัวแปรต่างๆ และค่าที่ Quantize ระหว่างอินพุทและเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1.1 แสดงขั้นตอนวงรอบของการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1.2 ค่าความสัมพันธ์ระหว่าง Effort กับ Complexity

Module Design เป็นการออกแบบในระดับ Circuit Module เช่น Transducers, Logic Blocks, Primitive Gates, Gain Blocks ซึ่งส่วนใหญ่สามารถนำมาจาก Libraries โดยที่ในระดับนี้สามารถที่ตัดสินใจเลือกว่าจะใช้ Technology ใดในการสร้าง

Circuit Design ในขั้นตอนนี้จะเป็นการออกแบบวงจรที่ไม่มีใน Library และจะต้องออกแบบตามลักษณะ Technology ที่เลือกด้วย

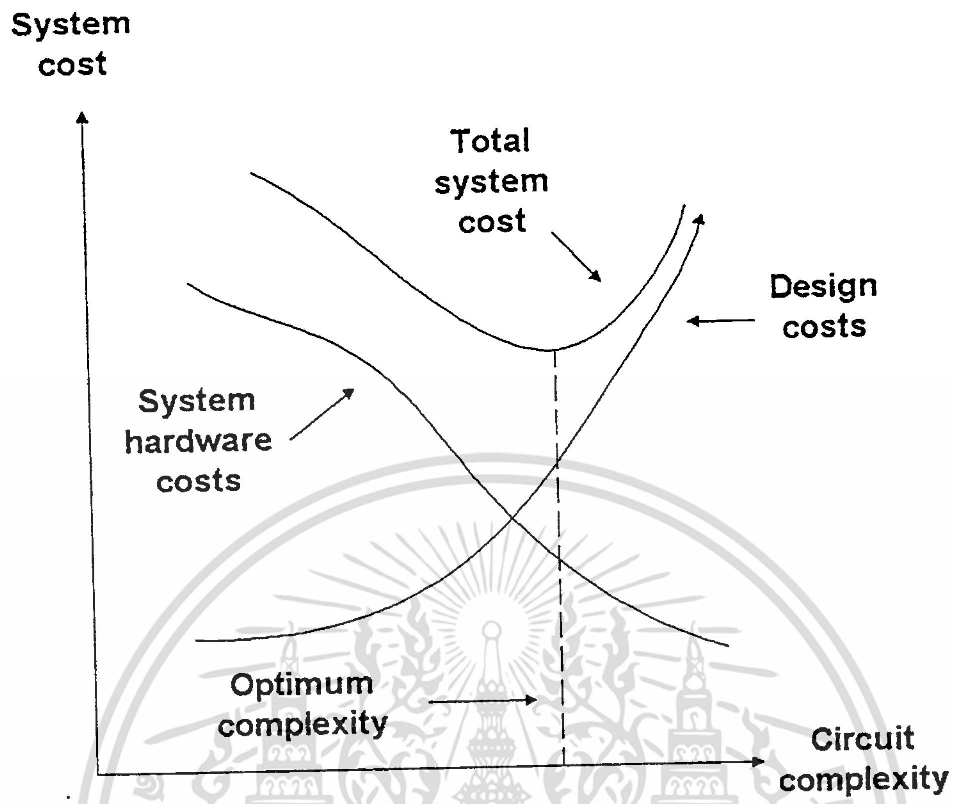
Lay out เป็นขั้นตอนที่จะทำ IC Masks ซึ่งสามารถใช้ Computer ช่วยในการออกแบบได้

Fabrication เป็นขั้นตอนที่เอา IC Mask มาใช้ในการสร้าง IC จริงๆ โดยที่เมื่อขนาดความซับซ้อนของระบบเพิ่มขึ้น ค่าราคาของการ Fabrication จะมีค่าน้อยลงดังรูปที่ 4.2.1

4.3 ขั้นตอนการสร้างไอซี (IC PROCESS)

วงจรชนิดต่างๆ เช่น Analog, Digital, Low-Noise ฯลฯ นั้นต้องใช้อุปกรณ์ที่มีความแตกต่างกันเพื่อที่จะให้ได้วงจรที่ดีสุด ซึ่งจะไม่มีวิธีไหนที่ใช้เป็นมาตรฐานสำหรับไอซีทุกชนิด เช่น วงจรดิจิตอลขนาดใหญ่โดยทั่วๆ ไปจะใช้ MOS Technology ในการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.1 ความสัมพันธ์ระหว่างราคาของระบบกับความซับซ้อนของระบบ

ในอนาคตจะมีความพยายามที่จะสร้างระบบลงบน Chip เดียว และพยายามทำระบบนี้เป็น Process นี้จะประกอบด้วย

4.3.1 Active Devices เพื่อที่จะสร้างอุปกรณ์ Switching และวงจรขยาย

4.3.2 Passive Components เช่น ตัวต้านทานและ Transmission Lines

4.3.3 Interconnections เป็นหน้าสัมผัสที่มีความต้านทานต่ำใช้เชื่อมส่วนต่างๆ ซึ่งความแตกต่างระหว่าง Transmission Lines กับ Connection นั้นยังไม่สามารถบ่งบอกได้ชัดเจน

4.3.4 Isolation เป็นส่วนที่แยกอุปกรณ์ออกจากกัน เช่น ออกไซด์, รอยต่อพีเอ็น, Wells, Grooves ซึ่งการเลือก Isolation แต่ละแบบนั้น จะขึ้นอยู่กับระบบและความหนาแน่นของ Package เป็นหลัก

4.4 กฎของการออกแบบ (DESIGN RULES)

ผู้ออกแบบจะต้องคำนึงถึง การทำให้ไอซีที่ผลิตมีคุณภาพสูงที่สุด ซึ่งมีวิธีการที่สำคัญ 3 ประการดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1 หาความเร็วที่เหมาะสม หาการตอบสนองความถี่หรือข้อกำหนดอื่นใดทางไฟฟ้า เพราะว่าเมื่อกำหนดไปแล้ว จะมีผลทำให้ขนาดของ Chip ลดลงไปได้

4.4.2 พยายามลดพื้นที่ของแต่ละ Chip ให้ได้พื้นที่ที่น้อยที่สุด ซึ่งจะทำให้ราคาลดลง และเมื่อมีแผ่น Wafer ที่มีเส้นผ่านศูนย์กลางเท่ากัน Chip ที่เล็กกว่าจะสามารถสร้างได้มากกว่า

4.4.3 ค่า Geometric Tolerances ของแต่ละรูปทรง จะต้องกำหนดให้เป็นค่าที่สูงที่สุด เพื่อที่จะคงไว้ซึ่งข้อกำหนดของวงจรที่ออกแบบ

กฎในการออกแบบโดยทั่วๆ ไปนั้น จะต้องมีการเชื่อมต่อกันระหว่าง Process Engineer กับ Chip Designer เพื่อที่จะหาจุดที่เหมาะสมที่สุดระหว่างข้อกำหนดที่มีความสับสนอยู่ระหว่างกัน

4.5 คอมพิวเตอร์ช่วยในการออกแบบ (COMPUTER AIDS)

เมื่อระบบมีความซับซ้อนมากขึ้นเช่น วงจรดิจิตอลขนาดใหญ่ การใช้คอมพิวเตอร์เข้ามาช่วยในการออกแบบเป็นสิ่งที่มีความจำเป็นที่ขาดเสียมิได้ ซึ่งเครื่องมือที่ใช้ในการออกแบบสามารถประยุกต์ใช้ได้ในหลายขั้นตอนของรูปที่ 4.1.1 แต่ทว่าการติดต่อระหว่างขั้นตอนยังเป็นที่สับสนอยู่ ดังนั้นจึงต้องมีการกำหนด EDIF (Electronic Data Interchange Format) ขึ้นมาเป็นโปรโตคอลมาตรฐานที่ใช้ในการสื่อสาร ได้ในทุกระดับของการออกแบบ

ถึงแม้ว่าเราจะมีระบบผู้เชี่ยวชาญ และ Silicon Compilers ซึ่งระบบที่เราใช้อยู่ทุกวันนี้ เป็นเพียงเครื่องมือที่จะทำให้ข้อกำหนดเบื้องต้นจากมนุษย์แปลงไปสู่วงจรและก็เหมือนกับเครื่องมืออื่นๆ ทั่วไปตรงที่ถ้ามีข้อผิดพลาดเกิดขึ้นในการออกแบบ ก็จะต้องใช้มนุษย์เป็นผู้แก้ไขเช่นกัน

การออกแบบโดยใช้คอมพิวเตอร์ (Level of Abstraction) มี 3 ระดับคือ

1. Behavioral Level ในระดับนี้จะสนใจเพียงพฤติกรรมของระบบที่เราออกแบบเท่านั้น พฤติกรรมของระบบอาจกำหนดด้วย Boolean Equations, ตารางค่าความจริง หรือเป็น Algorithms ที่เขียนด้วย Computer Language เช่น LISP, Ada หรือ Hardware Description Languages (HDLS) เช่น VHDL , Verilog, ELLA

2. Structural Level ในระดับนี้จะมองระบบว่าประกอบด้วย Module หรือ Macrocell ดังนั้นจะสนใจว่า Module ดังกล่าวมีการเชื่อมต่อการทำงานสัมพันธ์กันอย่างไร จึงจะได้พฤติกรรมตามที่ต้องการ เช่น ALU I/O Driver เป็นต้น

ใน Module นี้ หากพิจารณาถึงโครงสร้างภายในจะแบ่งได้อีกเป็น 3 ระดับคือ

2.1 Gate Level

2.2 Switch Level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Circuit Level

3. Physical Level เป็นระดับล่างสุดของการออกแบบ เป็นการกำหนดรูปร่างลักษณะของ Lay Out ของวงจร ซึ่งจะใช้เป็นข้อมูลในการทำ Photomask ของ Process การผลิตไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การออกแบบในเบื้องต้น

ในระบบการควบคุมอุณหภูมิแบบพีไอ (PI : Proportional Integral Control) นั้น โดยตัวของระบบเองจะมีคุณสมบัติเป็นวงจรกรองความถี่ต่ำ (Low Pass Filter) ที่พยายามจะทำให้ Steady State Error เป็น 0 ซึ่งปัญหาของเราก็คือ ทำอย่างไรเราจึงจะสามารถให้ระบบการควบคุมแบบพีไอ สามารถปรับตัวเองให้เหมาะสมกับการทำงานในสภาวะต่างๆ ได้อย่างมีประสิทธิภาพ ซึ่งคำตอบของเราจะมีพื้นฐานมาจากการตอบสนองความถี่ของระบบพีไอ

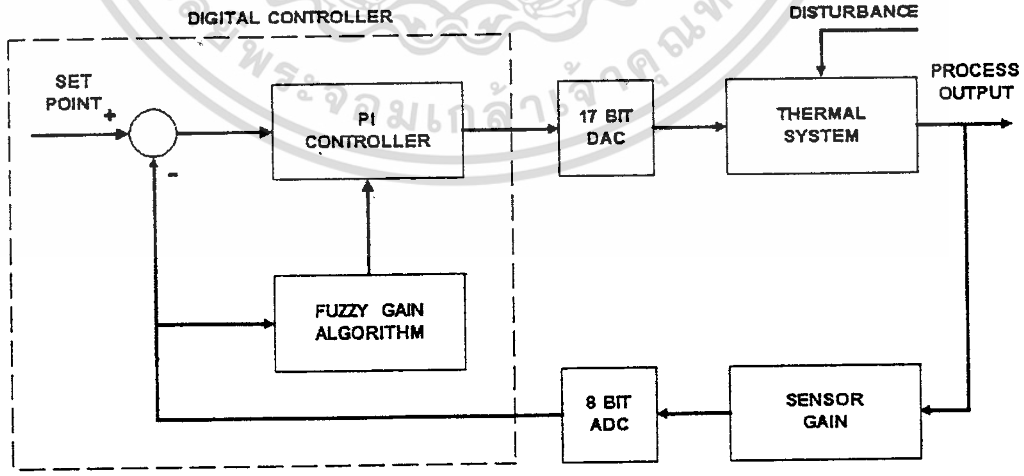
ระบบพีไอเป็นระบบที่ฟังก์ชันถ่ายโอนเป็น

$$T(s) = K_p + K_i/s$$

$$= [K_p s + K_i]/s$$

คุณสมบัติต่างๆ ของระบบอาทิเช่น ผลตอบสนองของระบบ Overshoot, Rise Time, Settling Time ขึ้นอยู่กับอัตราส่วนของสัมประสิทธิ์ของ K_p/K_i ดังนั้นการเปลี่ยนแปลงค่า K_p และ K_i ก็จะมีผลต่อเสถียรภาพของระบบด้วย ฉะนั้นในการปรับค่า K_i และ K_p จึงต้องคำนึงถึงหลักใหญ่ๆ อยู่ 2 ประการ คือ

1) ต้องไม่ทำให้ระบบเกิด Oscillate
2) ต้องทำให้ระบบมีประสิทธิภาพสูงสุด



รูปที่ 5.1 แสดงระบบควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5.1 จะเห็นว่า ถ้าเราจะหาฟังก์ชันถ่ายโอนแบบปิด (Close Loop Transfer Function) ของระบบรวม เป็นไปได้ค่อนข้างลำบาก เนื่องจากเราไม่สามารถทราบถึงฟังก์ชันถ่ายโอนของวงจร DAC (Digital to Analog Converter), ADC (Analog to Digital Converter), Thermal System, Sensor, Fuzzy Algorithm ดังนั้นในการกำหนดค่า K_p, K_i ที่จะรักษาเสถียรภาพของระบบนั้น จึงต้องเป็นการทำแบบลองผิดลองถูกกับวงจร ซึ่งเราสามารถทำได้โดย ใช้การ Simulation ในคอมพิวเตอร์ เพื่อหาค่าที่เหมาะสมต่อไป แต่ในที่นี้ก็จะมีการกำหนดค่า K_p, K_i ขึ้นมาก่อน เพื่อให้เป็นค่าที่ใช้ลองผิดลองถูก ดังจะกล่าวถึงในหัวข้อต่อไป

5.1 FUZZIFICATION OF INPUT

เราจะนำ Input Temperature (ความละเอียดในการวัดอยู่ที่ 0.1 องศาเซลเซียส) ที่ได้มาหาค่า Degree of Membership โดยเราจะทำการแปลง Input Temperature เป็นอินพุทของ Fuzzy Logic System ได้ 2 ค่า คือ

1. Error Temperature E(t) = Set Temp - T(t)
2. Change of Error Temperature $\Delta E = E(t) - E(t-1)$

โดยที่

Set Temp = อุณหภูมิที่ตั้งไว้

T(t) = อุณหภูมิในขณะนั้น

E(t) = ค่า Error ในปัจจุบัน

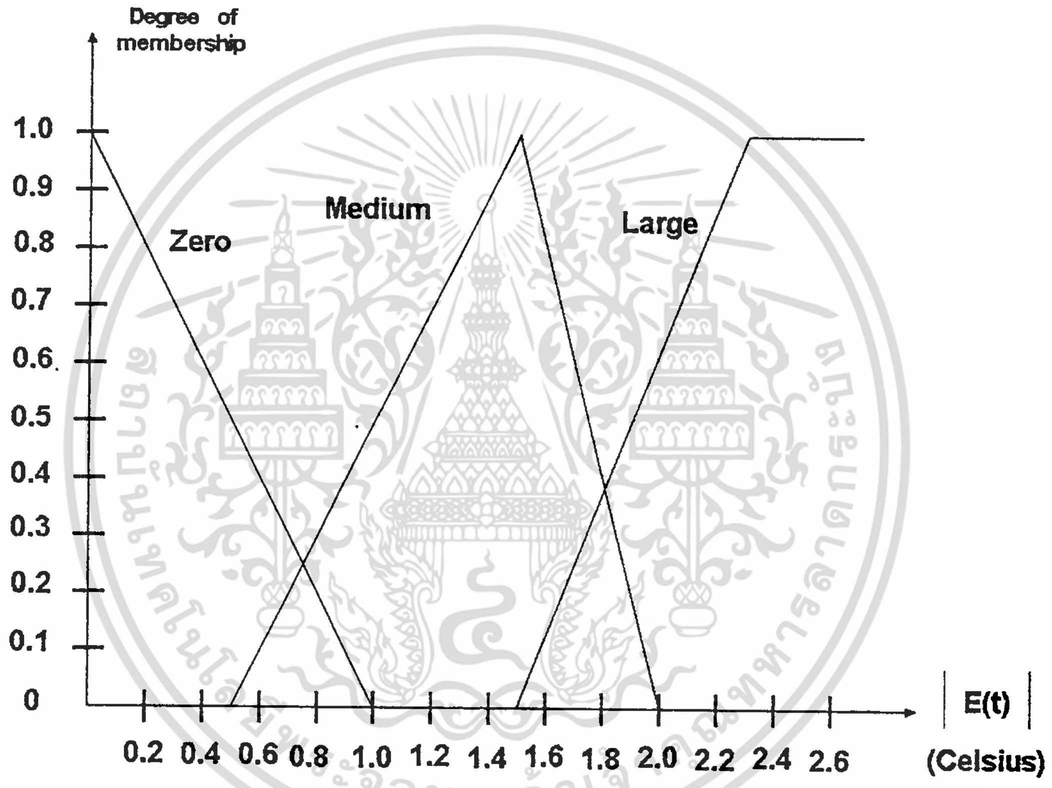
E(t-1) = ค่า Error ก่อนหน้านั้น

ΔE = ค่า Change Error Temperature

จากตัวแปร E(t) และ ΔE ดังกล่าวข้างต้น เราจะใช้เป็น Fuzzy Input ให้กับส่วน Fuzzification เพื่อหาค่าสัมประสิทธิ์ K_p, K_i ให้เหมาะสม ตามการทำงานในแต่ละสภาวะ ซึ่งค่า E(t) และ ΔE นั้น มีโอกาสที่จะเป็นไปได้ทั้งค่าบวกและค่าลบ ซึ่งถ้าพิจารณาดูให้ดีจะเห็นว่า ค่า E(t) และ ΔE ที่เป็นลบ จะมีค่าสัมประสิทธิ์ K_p, K_i เหมือนกับค่า E(t) และ ΔE ค่าเท่ากันกับที่เป็นบวก เพราะว่าค่าสัมประสิทธิ์ K_p, K_i นี้จะเป็นตัวกำหนดความเร็วในการเข้าสู่สภาวะ Steady State เท่านั้น ในการ Fuzzification เราจึงไม่ต้องสนใจเครื่องหมายของ E(t) และ ΔE ทำให้ตัวแปรที่จะใช้หาค่า Membership Function มีสองตัวคือ $|E(t)|$ และ $|\Delta E|$

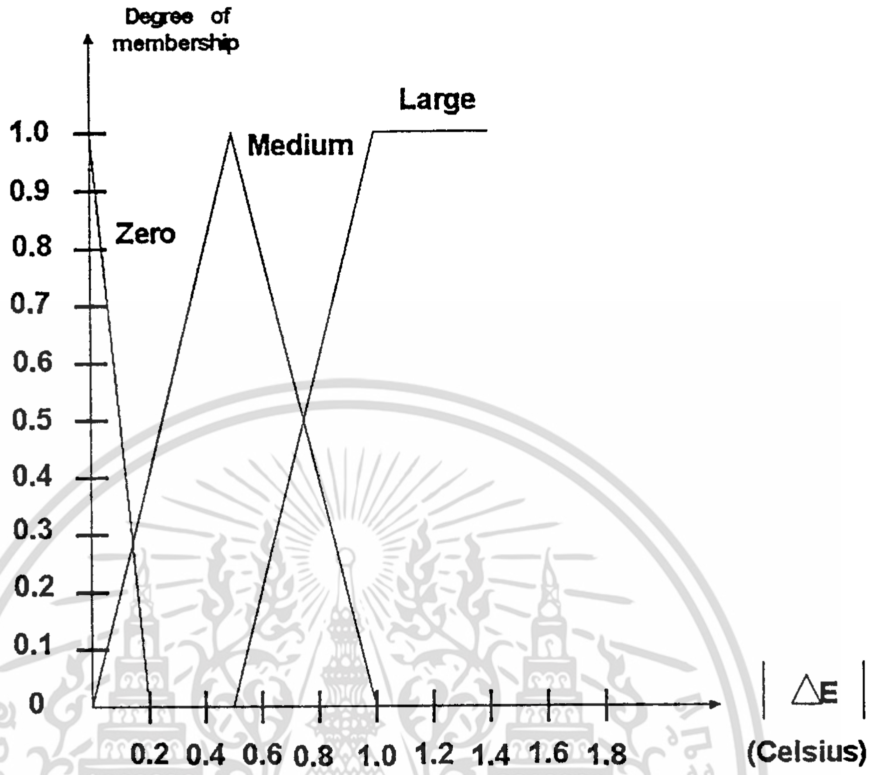
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจากตัวแปรอินพุททั้งสองตัวนี้ |E(t)| และ |ΔE| จะต้องนำมาเปรียบเทียบกราฟความเป็นสมาชิกที่กำหนดขึ้นมาก่อน ดังรูปที่ 5.1.1 (ก) และ 5.1.1 (ข) ซึ่งกราฟความเป็นสมาชิกนี้จะกำหนดขึ้นมาโดยอาศัยความต้องการและความรู้ของมนุษย์ในการเลือกค่าอุณหภูมิ แล้วเราต้องค่อยๆ ปรับกราฟนี้จนสอดคล้องกับความเป็นจริงมากที่สุด โดยที่รูปที่ 5.1.1 (ก) จะเป็นกราฟความเป็นสมาชิกของ Error Temperature และรูปที่ 5.1.1 (ข) จะเป็นกราฟความเป็นสมาชิกของ Change of Error Temperature



รูปที่ 5.1.1 (ก) ค่า Membership ของ |E(t)|

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1.1 (ข) ค่า Membership ของ $|\Delta E|$

Rule Evaluation

ตารางที่ 5.1.1 แสดง Fuzzy IF-THEN Rule

	$\|E(t)\|$	$\|\Delta E\|$	K_p	K_i	STATUS
1	ZERO	ZERO	20	12	A
2	ZERO	MEDIUM	20	12	A
3	ZERO	LARGE	16	10	B
4	MEDIUM	ZERO	16	10	B
5	MEDIUM	MEDIUM	20	10	C
6	MEDIUM	LARGE	20	10	C
7	LARGE	Any Value	24	8	D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตั้งกฎเหล่านั้นขึ้นมาจะอาศัยหลักเกณฑ์ดังนี้ คือ

1. มี $|E(t)|$ และ $|\Delta E|$ เท่านั้นที่เป็นอินพุท
2. เอาท์พุททั้งหมดเป็น Integer
3. ค่า K_i, K_p จะเปลี่ยนไปพร้อมๆ กัน
4. เมื่อ Error Signal มาก K_p/K_i จะต้องมากเพื่อจะเพิ่มสัญญาณควบคุม
5. เมื่อ Error Signal เริ่มลดลง K_p/K_i จะลดลงเพื่อลด Overshoot
6. เมื่อ Error Signal น้อย K_p/K_i จะมีค่ามากเพื่อเพิ่มความไว (Sensitivity)

การคำนวณค่าตามกฎที่ตั้งไว้ โดยอาศัยค่าความเป็นสมาชิกของตัวแปร ที่ได้จากการเปรียบเทียบค่า Fuzzy Input กับกราฟความเป็นสมาชิกเป็นตัวกำหนดความสำคัญของกฎแต่ละกฎ โดยอาศัยเทคนิคของค่ากฎ Smallest ระหว่างตัวแปร Fuzzy Input ทั้ง 2 ตัว ในส่วนเงื่อนไขคือ Error และ Change of Error ตาม Fuzzy IF-THEN Rule ดังตารางที่ 5.1.1

จากตารางที่ 5.1.1 นั้นค่าผลลัพธ์ที่เกิดจากแต่ละเงื่อนไขของแต่ละตัว มีค่าตามตารางที่ 5.1.2

ตารางที่ 5.1.2 ตารางแสดงค่าสภาวะ

STATUS (TEMPERATURE)	ASSIGNED VALUE
A	1
B	2
C	3
D	4

5.2 DEFUZZIFICATION OF OUTPUT

เป็นขั้นตอนที่นำเอาผลลัพธ์ที่ได้จากขั้นตอน Rules Evaluation เพื่อหาค่าของ Fuzzy Output ซึ่งเมื่อทำการเปรียบเทียบตามกฎต่างๆ แล้ว จะเห็นได้ว่า ถ้ามีการคำนวณตามกฎการควบคุม แล้วนำเพียงกฎใดกฎหนึ่งมาใช้จะทำให้ผลการควบคุมระบบผิดพลาดได้ จึงได้นำค่าเอาพุทของกฎต่างๆ มาหาค่าเอาท์พุทที่จะใช้ในการควบคุมโดยใช้การคำนวณแบบ Center of Gravity คำนวณผลลัพธ์ออกมา โดยค่าของจุดศูนย์ถ่วงสำหรับตัวแปรทางเอาท์พุทได้แสดงดังตารางที่ 5.1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2.1 ตารางแสดงค่าเอาท์พุทจากการคำนวณโดย Fuzzy Rules

|ΔE| |E(t)|	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	≥1.0
0	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.1	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.2	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.3	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.4	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.5	1	1	1	1	1	1	1.5	1.5	1.5	1.5	2
0.6	1.5	1.75	2	2	2	2	2.25	2.25	2.25	2.25	2.5
0.7	1.5	1.75	2	2	2	2	2.25	2.25	2.25	2.25	2.5
0.8	1.5	1.75	2	2	2	2	2.25	2.25	2.25	2.25	2.5
0.9	1.5	1.75	2	2	2	2	2.25	2.25	2.25	2.25	2.5
1.0	2	2.5	3	3	3	3	3	3	3	3	3
1.1	2	2.5	3	3	3	3	3	3	3	3	3
1.2	2	2.5	3	3	3	3	3	3	3	3	3
1.3	2	2.5	3	3	3	3	3	3	3	3	3
1.4	2	2.5	3	3	3	3	3	3	3	3	3
1.5	2	2.5	3	3	3	3	3	3	3	3	3
1.6	3	3.25	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
1.7	3	3.25	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
1.8	3	3.25	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
1.9	3	3.25	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
≥2.0	4	4	4	4	4	4	4	4	4	4	4

หลังจากการที่ได้ทำการคำนวณตาม Rule Evaluation แล้วทำการหาค่า Center of Gravity แล้วสามารถเขียนตารางของ Fuzzy Output สำหรับค่า Error และ Change of Error ต่างๆ ได้ตามตารางที่ 5.2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ผลลัพธ์จากการทำ Defuzzification แล้วก็จะได้ค่าของการคำนวณออกมาในรูปของ Fuzzy Output ซึ่งค่าของ Fuzzy Output นี้ยังไม่สามารถไปใช้ในการควบคุมอุณหภูมิได้โดยตรงจึงต้องนำมาผ่านขบวนการที่จะทำการเปลี่ยนให้ค่าของ Fuzzy Output นั้นสามารถนำไปใช้ในการควบคุมเครื่องทำน้ำอุ่นได้ซึ่งวิธีการก็คือ จะนำค่า Fuzzy Output ไปเปรียบเทียบกับตารางที่ 5.2.2 ซึ่งเป็นขบวนการกลับกันกับการหาค่า Fuzzy Input เพื่อหาสถานะของค่า K_p และ K_i เมื่อได้แล้ว ทำการเปลี่ยนค่า Fuzzy Output ให้เป็นค่าสัมประสิทธิ์ K_p, K_i ซึ่งจะได้สัมประสิทธิ์ของระบบพีไอที่ Error Temperature และ Change of Error Temperature ต่างๆ ดังแสดงในตารางที่ 5.2.3

ตารางที่ 5.2.2 แสดงค่าความสัมพันธ์ของ Fuzzy Output กับค่าสัมประสิทธิ์ K_p, K_i

FUZZY OUTPUT	STATUS	(K_p, K_i)
0.0-1.4	A	(20,12)
1.5-2.4	B	(16,10)
2.5-3.4	C	(20,10)
3.5-4.0	D	(24,8)

จากขั้นตอนต่างๆ ที่กล่าวมาข้างต้น จะขอแสดงการหาค่า Fuzzy Output เพื่อเป็นตัวอย่าง 1 ค่า โดยในตอนแรกเราต้องสมมติค่าต่างๆ ดังนี้

Set Temperature = 32.7 C

T(t) = 31.0 C

T(t-1) = 32.9 C

นำค่าเหล่านี้เข้าสู่ขั้นตอน Fuzzification แปลงเป็น Fuzzy Input จะได้ค่า

$$
\begin{aligned}
|\text{Error Temperature } (E(t))| &= |\text{Set} - T(t)| \\
&= 32.7 - 31.0 \\
&= 1.7 \\
|\text{Change of Error Temperature}| &= |E(t) - E(t-1)| \\
&= |1.7-(-0.2)| \\
&= 1.9
\end{aligned}
$$

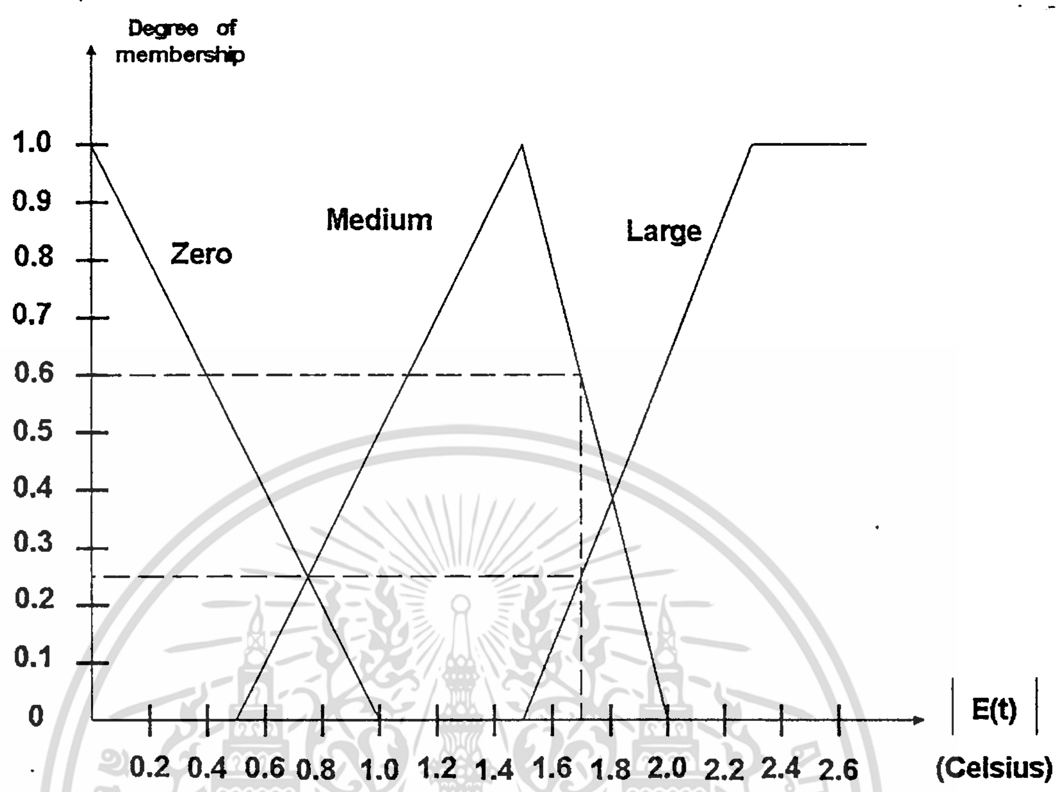
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2.3 แสดง Fuzzy Output ที่นำไปใช้งานจริง

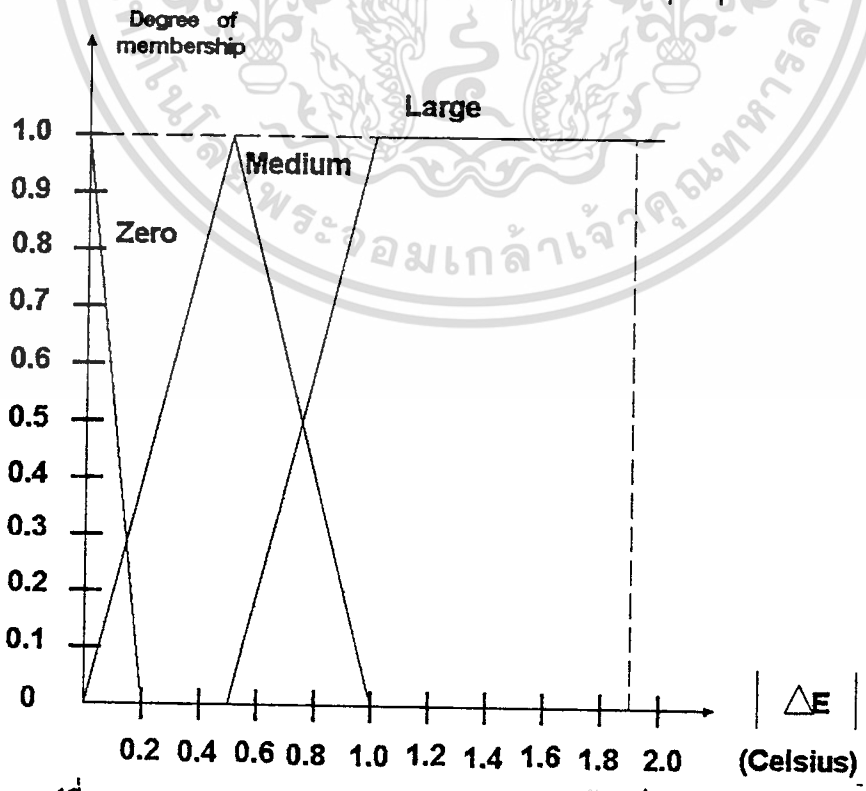
$\lvert\Delta E\rvert$ / $\lvert E(t)\rvert$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	≥1.0
0	A	A	A	A	A	A	B	B	B	B	B
0.1	A	A	A	A	A	A	B	B	B	B	B
0.2	A	A	A	A	A	A	B	B	B	B	B
0.3	A	A	A	A	A	A	B	B	B	B	B
0.4	A	A	A	A	A	A	B	B	B	B	B
0.5	A	A	A	A	A	A	B	B	B	B	B
0.6	B	B	B	B	B	B	B	B	B	B	C
0.7	B	B	B	B	B	B	B	B	B	B	C
0.8	B	B	B	B	B	B	B	B	B	B	C
0.9	B	B	B	B	B	B	B	B	B	B	C
1.0	B	C	C	C	C	C	C	C	C	C	C
1.1	B	C	C	C	C	C	C	C	C	C	C
1.2	B	C	C	C	C	C	C	C	C	C	C
1.3	B	C	C	C	C	C	C	C	C	C	C
1.4	B	C	C	C	C	C	C	C	C	C	C
1.5	B	C	C	C	C	C	C	C	C	C	C
1.6	C	C	D	D	D	D	D	D	D	D	D
1.7	C	C	D	D	D	D	D	D	D	D	D
1.8	C	C	D	D	D	D	D	D	D	D	D
1.9	C	C	D	D	D	D	D	D	D	D	D
≥2.0	D	D	D	D	D	D	D	D	D	D	D

นำค่า $\lvert E(t)\rvert$ และ $\lvert\Delta E\rvert$ ไปหาค่า Degree of Membership จากกราฟรูปที่ 5.1.1 (ก) และ 5.1.1(ข) ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2.1 (ก) แสดงการหาค่า Membership Function ที่ |E(t)| = 1.7



รูปที่ 5.2.1 (ข) แสดงการหาค่า Membership Function ที่ |△E | = 1.9

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงการหาค่า Degree of Membership ของ E(t) และ ΔE ที่ $|E(t)| = 1.7$ และ $|\Delta E| = 1.9$ ตามลำดับ

จากกราฟจะเห็นว่าค่า $|E(t)|$ เท่ากับ 1 จะไปตัดกราฟ Medium และ Large ที่ค่า Membership เท่ากับ 0.25 สำหรับ Large และเท่ากับ 0.6 ที่ Medium ส่วนค่า $|\Delta E| = 1.9$ จะไปตัดกราฟ Large ที่ Membership Function เท่ากับ 1 เมื่อได้ค่าเหล่านี้แล้ว เราจะใช้กฎ Smallest หาค่าตัวแทนกลุ่ม คือ

$$\mu(|E(t)|,|\Delta E|) = \min[\mu(|E(t)|), \mu(|\Delta E|)]$$
$$= 0.25$$

หลังจากนั้นเราจะไปดูตารางที่ 5.1.1 เพื่อหากฎของ Fuzzy Rule ที่สอดคล้องกับเงื่อนไขทางอินพุทที่หามาได้ จะเห็นว่าจะสอดคล้องกฎที่ 6 และ 7 ซึ่งมีค่าน้ำหนักเป็น 3 และ 4 ตามลำดับ แล้วนำค่านี้มาคำนวณแบบ Center of Gravity โดยมีสูตร

$$In = \frac{\sum_{i=1}^{n}(\mu n \times Un)}{\sum_{i=1}^{n}\mu n}$$

โดยที่ In = Result of Centroid Method

μn = Degree of Membership

Un = Status from IF-THEN Rules

$$In = \frac{(0.25 \times 4) + (0.25 \times 3)}{(0.25 + 0.25)}$$
$$= 3.5$$

ซึ่งเป็นค่าที่ตรงกับตารางที่ 5.2.1 และเมื่อนำค่าที่คำนวณได้นี้ไปเปรียบเทียบกับตารางที่ 5.2.2 จะได้ค่าของ Fuzzy Output คือ D ($K_p = 24, K_i = 8$) ซึ่งตรงกันค่าในตารางที่ 5.2.3 ที่ $|E(t)| = 1.7$ และ $|\Delta E| \geq 1.0$ ส่วนการหาค่า Fuzzy Output อื่นๆ ที่อุณหภูมิอื่นๆ ก็จะได้ในทำนองเดียวกัน

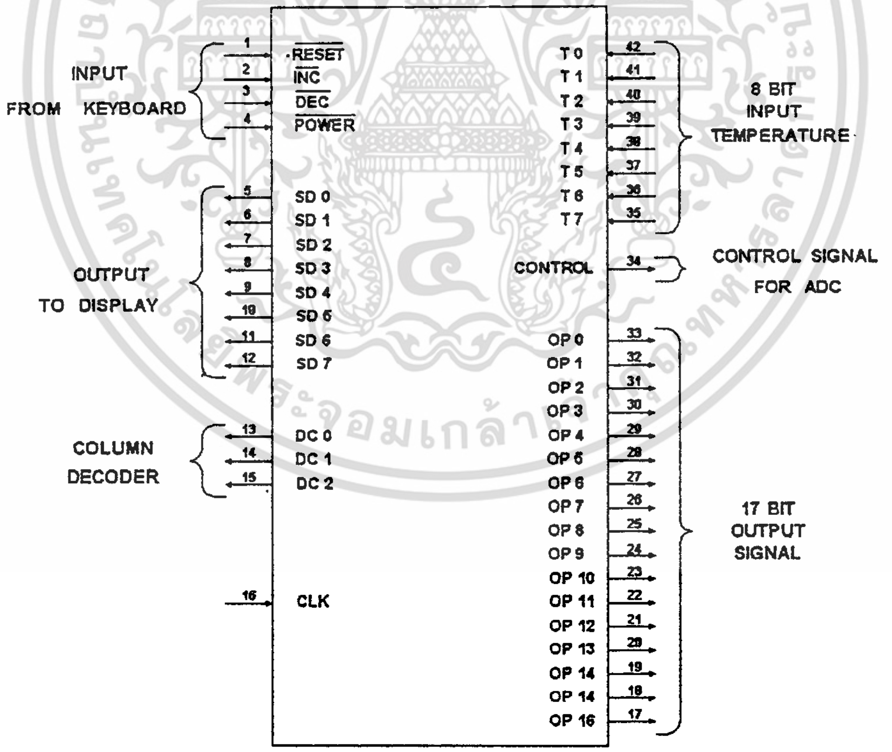
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
การออกแบบทางด้าน HARDWARE

หลังจากที่ได้ระเบียบวิธีในการคำนวณแล้ว สิ่งที่ต้องทำต่อไปก็คือ การออกแบบ Hardware ซึ่งวงจรที่ต้องการนี้จะต้องมีคุณสมบัติเบื้องต้นดังนี้ คือ

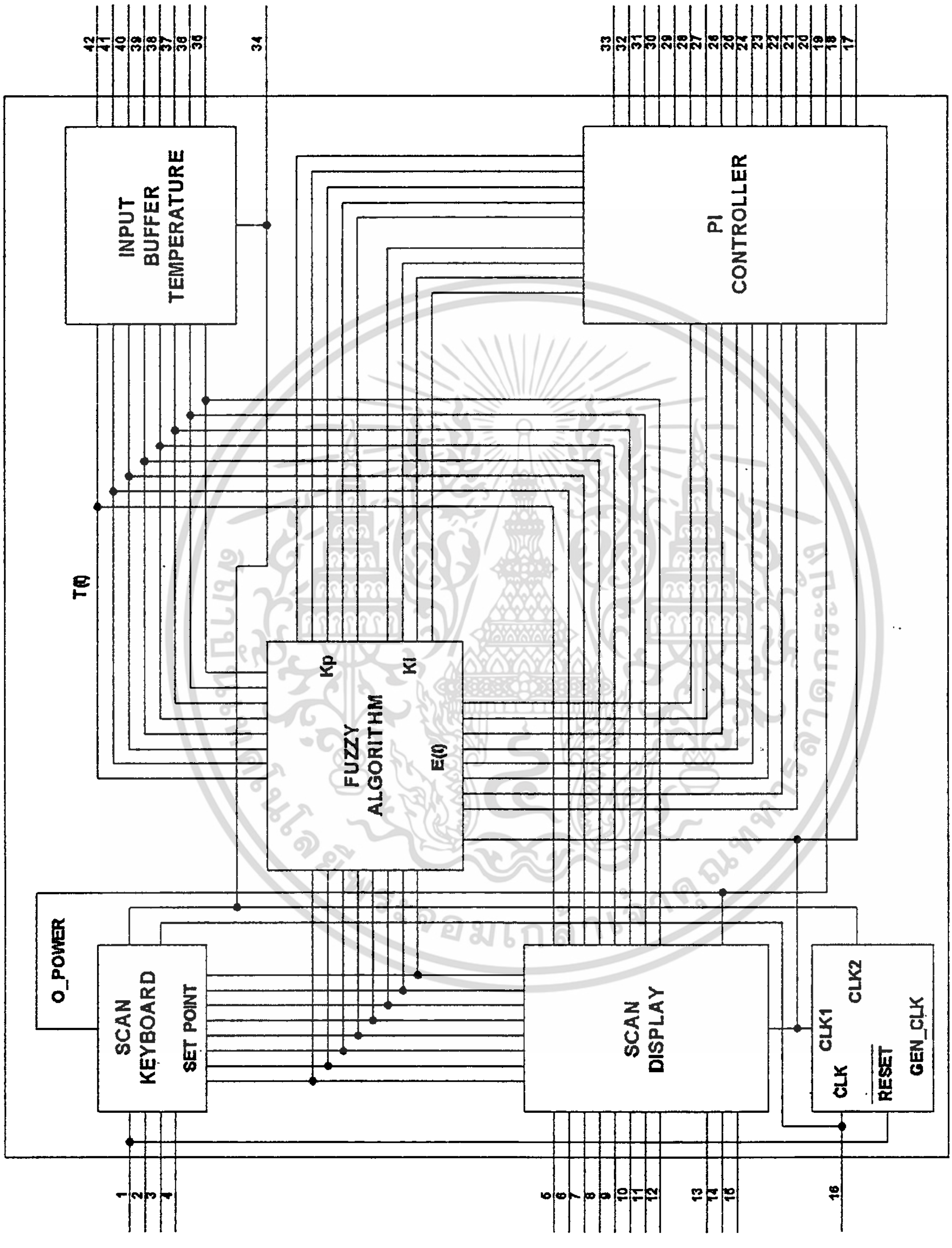
1. ใช้การคำนวณแบบดิจิตอล
2. รับอินพุทจากมนุษย์ในการตั้งค่าอุณหภูมิ
3. รับอินพุทจากเซนเซอร์อุณหภูมิที่เป็นข้อมูลดิจิตอลแล้ว
4. มีส่วนแสดงผลในการแสดง Output Temperature ปัจจุบัน และอุณหภูมิที่ตั้งไว้
5. ค่าเอาท์พุทที่ออกมาจะเป็นสัญญาณดิจิตอล

จากข้อกำหนดเบื้องต้นนี้ทำให้สามารถออกแบบระบบได้ดังรูปที่ 6.1



รูปที่ 6.1 แสดงวงจรที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

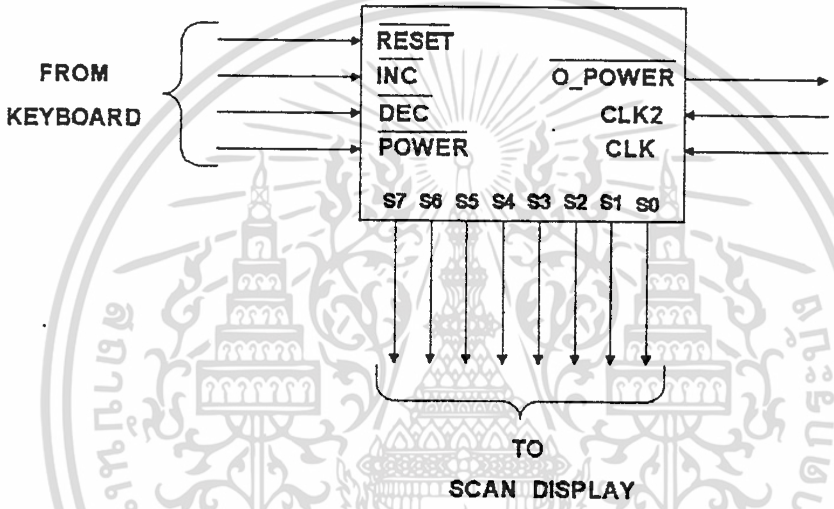


รูปที่ 6.2 แสดงรายละเอียดภายในของระบบที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่วงจรที่ต้องการในรูป 6.1 นี้ ยังไม่มีรายละเอียดเพียงพอที่จะนำไปสร้างโดยภาษา VHDL ได้ ฉะนั้น จึงต้องมีการออกแบบลงไปในรายละเอียดภายใน ซึ่งจะทำให้ได้วงจรดังรูปที่ 6.2

6.1 ส่วนสแกนคีย์บอร์ด (SCAN KEYBOARD)



รูปที่ 6.3 รูปส่วนสแกนคีย์บอร์ด

ในส่วนนี้จะรับค่าคีย์บอร์ดอยู่ 4 แบบ คือ

-INC เป็นสวิทซ์ที่ใช้เพิ่มค่าอุณหภูมิ

-DEC เป็นสวิทซ์ที่ใช้ลดค่าอุณหภูมิ

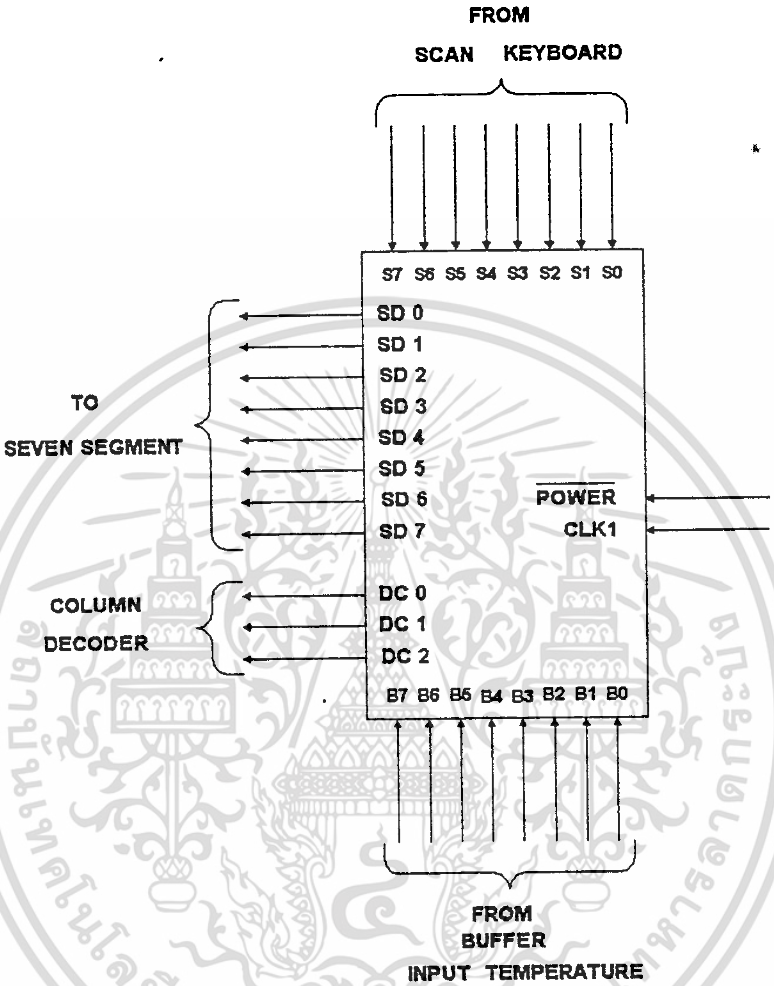
-POWER ใช้เปิด-ปิดระบบ

-RESET ใช้รีเซตระบบ

ซึ่งค่าเอาท์พุทของส่วนนี้จะส่งออกไปเป็นเลขฐาน 2 ขนาด 8 บิต และสามารถตั้งอุณหภูมิได้ในช่วง 20-40 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ส่วนสแกนดิสเพลย์ (SCAN DISPLAY)

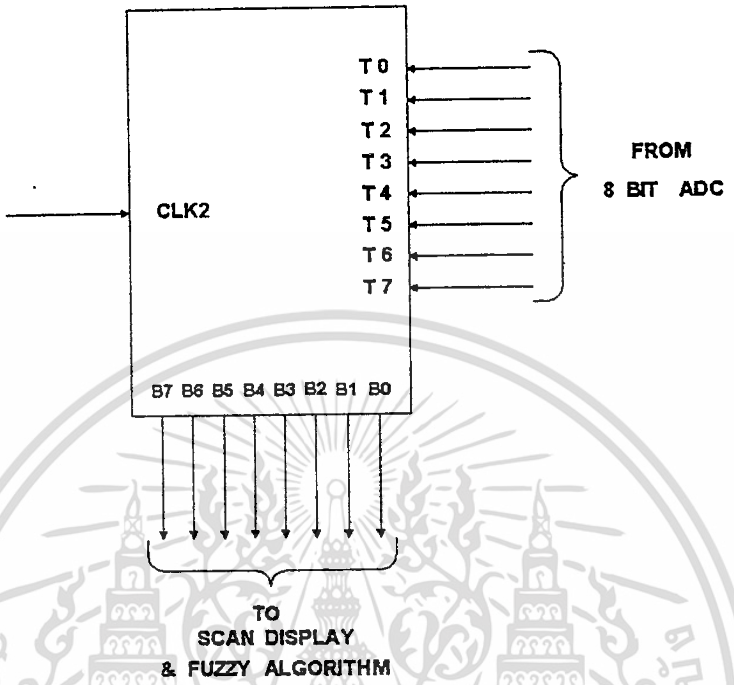


รูปที่ 6.4 ส่วน Scan display

ในส่วนนี้จะรับค่าอินพุทจากส่วนสแกนคีย์บอร์ด และส่วน Buffer Temperature เพื่อนำมาแปลงเป็นเลข BCD แล้วนำไปแสดงที่ Seven Segment อีกทีหนึ่ง และสัญญาณ Decode ขนาด 3 บิตนี้ใช้สำหรับเลือก Column ของ Seven Segment ที่จะใช้แสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 ส่วน BUFFER TEMPERATURE



รูปที่ 6.5 ส่วน Buffer Temperature

เป็นส่วนที่จะรับค่าอุณหภูมิจากภายนอกที่เป็นเลขฐานสองขนาด 8 บิต เข้ามาโดยมีสัญญาณควบคุมขนาด 1 บิต เพื่อใช้ควบคุมสัญญาณที่จะส่งมาจากวงจร ADC (Analog to Digital Converter) ให้ Synchronize กัน แล้วค่าอุณหภูมิที่ได้นี้จะส่งไปให้ส่วน Display และ Fuzzy Algorithm พร้อมๆ กัน

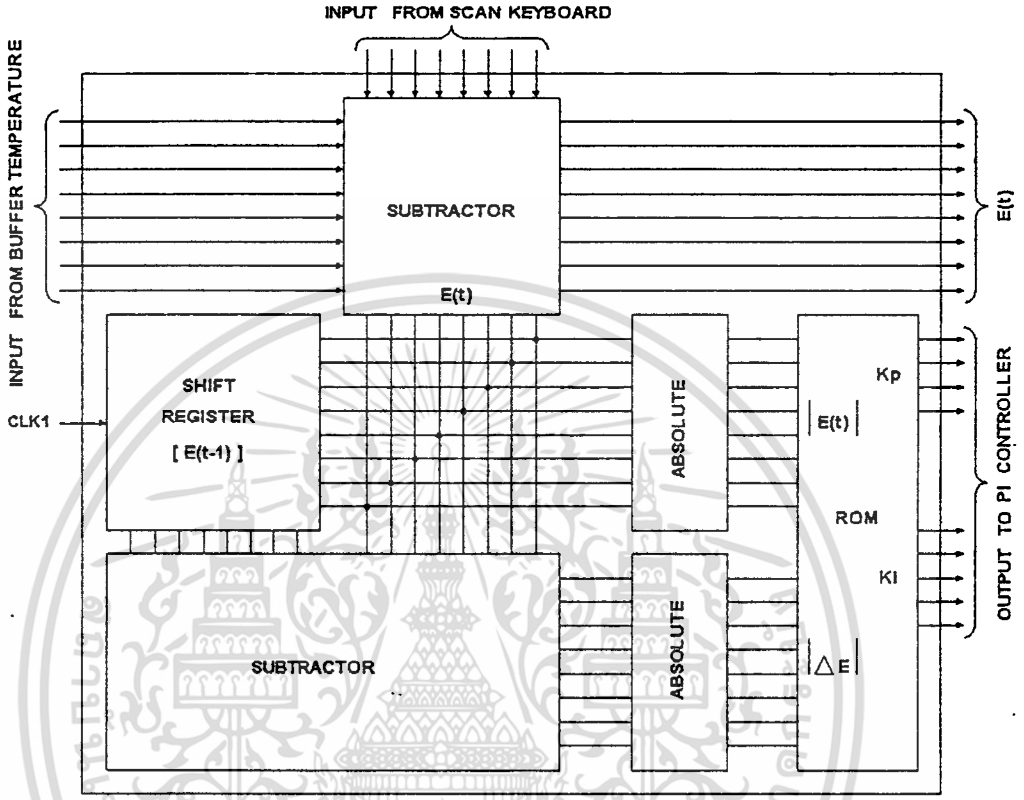
6.4 ส่วน Fuzzy Algorithm

สามารถแบ่งส่วนนี้เป็นส่วนย่อยๆ ได้ดังรูปที่ 6.6 จะเห็นว่าอินพุท คือ ค่า T(t) และ Set Temp ก็จะเข้ามาคำนวณหาค่า E(t) แล้วนำค่านี้ออกไปคำนวณค่า ΔE อีกทีโดยที่ E(t-1) ได้มาจาก Shift Register ซึ่งค่า ΔE และ E(t) ต้องนำไปผ่านวงจร Absolute เพื่อที่จะเลือกเฉพาะค่าที่เป็นบวก แล้วนำค่า $|\Delta E|$ และ $|E(t)|$ ไปหาค่า K_p และ K_i ที่เหมาะตาม Algorithm Fuzzy โดยที่จะใช้เทคนิค Look Up Table ในการหาค่า

ค่าเอาท์พุทจากส่วนนี้คือค่า K_p , K_i และ E(t) จะถูกส่งออกไปในรูปของเลขฐานสองขนาด 5, 4 และ 8 บิต ตามลำดับ ก็เพราะว่าค่า K_p สูงสุดคือ 24 และ K_i มีค่าสูงสุดเท่ากับ 12 จึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแทนด้วยเลขฐานสองดังกล่าวข้างต้น ส่วนค่า E(t) จะเป็นค่าที่ใช้ในการคำนวณในลำดับถัดไป จึงต้องส่งออกไปด้วย



รูปที่ 6.6 แสดงรายละเอียดของส่วน Fuzzy Algorithm

6.5 ส่วนคำนวณแบบพีไอ (PI CONTROLLER)

ในส่วนนี้ เราจะต้องทำการคำนวณค่าฟังก์ชันถ่ายโอนของพีไอ ซึ่งเป็นตัวแปรในโดเมน S ฉะนั้นจะต้องมีการเปลี่ยนไปเป็นโดเมน Z เพื่อที่จะทำให้สามารถทำการคำนวณแบบดิจิตอลได้ โดยในการเปลี่ยนตัวแปรนี้ จะใช้การส่งแบบ Bilinear Transform ซึ่งจะให้ผลการแปลงมีค่าใกล้เคียงกับฟังก์ชันถ่ายโอนเดิมที่ความถี่ไม่สูงมากนัก

หลังจากที่ได้ฟังก์ชันถ่ายโอนในโดเมน Z จากการแปลงแบบ Bilinear มาแล้ว ก็จะสามารถเขียนโครงสร้างของระบบควบคุมแบบ PI ได้หลายอย่าง เช่น แบบโดยตรง 1, แบบโดยตรง 2 , แบบสลับเปลี่ยน (Transpose Structure) เป็นต้น ทั้งนี้การเลือกโครงสร้างในแต่ละแบบจะมีข้อดีและเสียแตกต่างกันไป ซึ่งในที่นี้ได้เลือกใช้โครงสร้างแบบโดยตรง 2 เพราะว่าไม่มีการป้อนกลับจากเอาท์พุทเข้ามาบวกเพิ่ม ทำให้โอกาสเกิด Overflow น้อยกว่าแบบอื่นๆ ซึ่งสามารถแสดงได้โดยใช้สมการดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T(S) = Kp + \frac{Ki}{S}$$

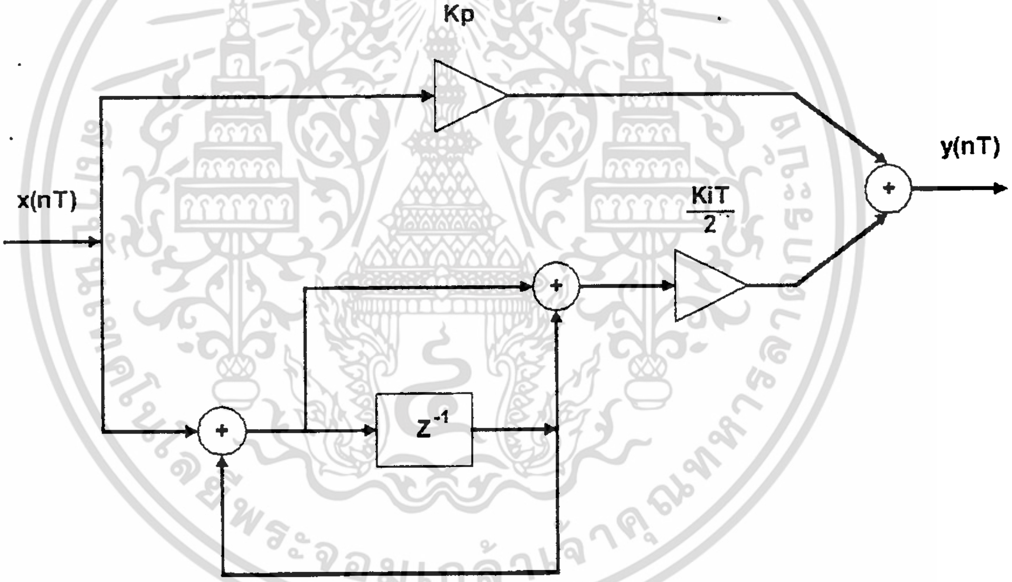
$$S = \frac{2}{T}\frac{(1-Z^{-1})}{(1+Z^{-1})}$$

$$\therefore T(Z) = \frac{Y(Z)}{X(Z)} = \frac{W(Z)}{X(Z)} \times \frac{Y(Z)}{W(Z)} = Kp + Ki\frac{T}{2}\frac{(Z+1)}{(Z-1)}$$

$$\therefore w(nT) = x(nT) + w(nT-T)$$

$$\therefore y(nT) = (Kp + \frac{KiT}{2})w(nT) + (\frac{KiT}{2} - Kp)w(nT-T)$$

จากสมการข้างต้นนี้ สามารถนำมาเขียนโครงสร้างได้ดังรูปที่ 6.7

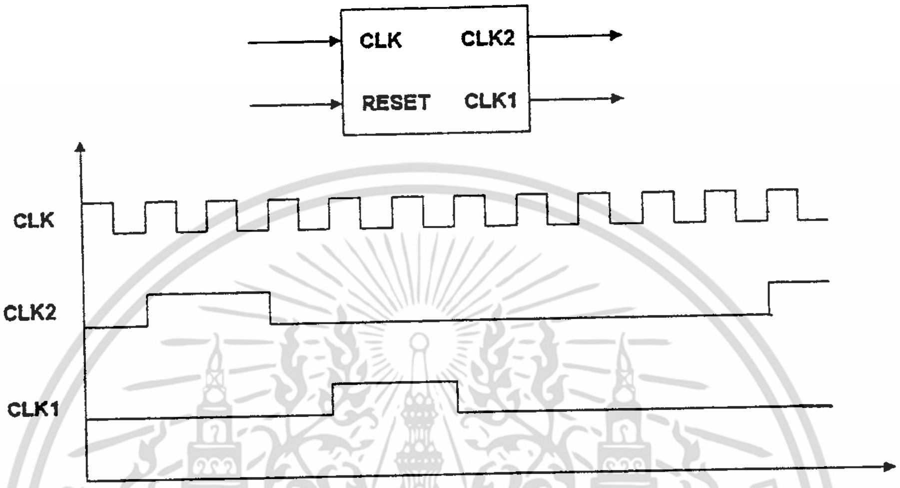


รูปที่ 6.7 แสดงโครงสร้างแบบโดยตรง 2

โดยที่ x(nT) คือ E(t) ที่ถูกสุ่มขึ้นมาด้วยความถี่เท่ากับส่วนกลับของคาบเวลา เนื่องจากเราใช้การคำนวณแบบดิจิตอล จึงต้องมีการสุ่มสัญญาณ ซึ่งค่าความถี่ในการสุ่มนี้จะต้องมากพอที่จะรักษารูปร่างของสัญญาณไว้ไม่ให้ผิดเพี้ยนไป แต่ในที่นี้สัญญาณ E(t) เป็นค่า Error ของอุณหภูมิ ซึ่งมีการเปลี่ยนแปลงที่ค่อนข้างช้าฉะนั้นเราจึงไม่จำเป็นต้องใช้ความถี่ในการสุ่มสูงมาก โดยในที่นี้จะใช้ค่าความถี่ในการสุ่มสัญญาณเท่ากับ 1 kHz และความถี่ของสัญญาณนาฬิกามีค่าเท่ากับ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 ส่วนสร้างสัญญาณนาฬิกา 2 เฟส



รูปที่ 6.8 แสดงวงจรและสัญญาณนาฬิกาของวงจรสร้างนาฬิกา 2 เฟส

ในส่วนนี้จะทำการสร้างสัญญาณนาฬิกา 2 เฟส เพื่อทำให้ระบบทำงานสอดคล้องกันได้ โดยที่มีลักษณะของสัญญาณดังรูปที่ 6.8

6.7 การทำงานโดยรวม

เมื่อได้ทราบถึงการทำงานในแต่ละส่วนแล้วก็จะทำให้สามารถเข้าใจ การทำงานโดยรวมได้ไม่ยาก กล่าวคือ เมื่อมีอินพุตจาก Keyboard เข้ามา ส่วน Scan Keyboard ก็จะรับข้อมูลเข้ามาในช่วงขอบขาลงของ CLK2 ในขณะเดียวกันส่วน Buffer Temp ก็จะรับค่าอุณหภูมิจากสิ่งแวดล้อมภายนอกเข้ามาที่ขอบขาลงของ CLK2 เช่นกัน โดยที่สัญญาณ CLK2 จะเป็นสัญญาณที่ใช้ในการควบคุมวงจร ADC ภายนอก ค่าที่ได้จาก Sensor และจาก keyboard นี้จะส่งไปที่ส่วน Fuzzy Algorithm ในช่วงขอบขาขึ้นของ CLK1 เพื่อหาค่า K_p และ K_i ที่เหมาะสม หลังจากนั้นค่า K_p และ K_i นี้จะใช้ในการคำนวณหาค่าเอาท์พุท ตามการคำนวณแบบดิจิตอลแล้วส่งออกไปที่ เอาท์พุทในช่วงขอบขาลงของ CLK1 ทั้งนี้ก็เพื่อที่จะให้การทำงานของแต่ละภาคสอดคล้องกัน และสัมพันธ์กับค่าความถี่การสุ่มสัญญาณ

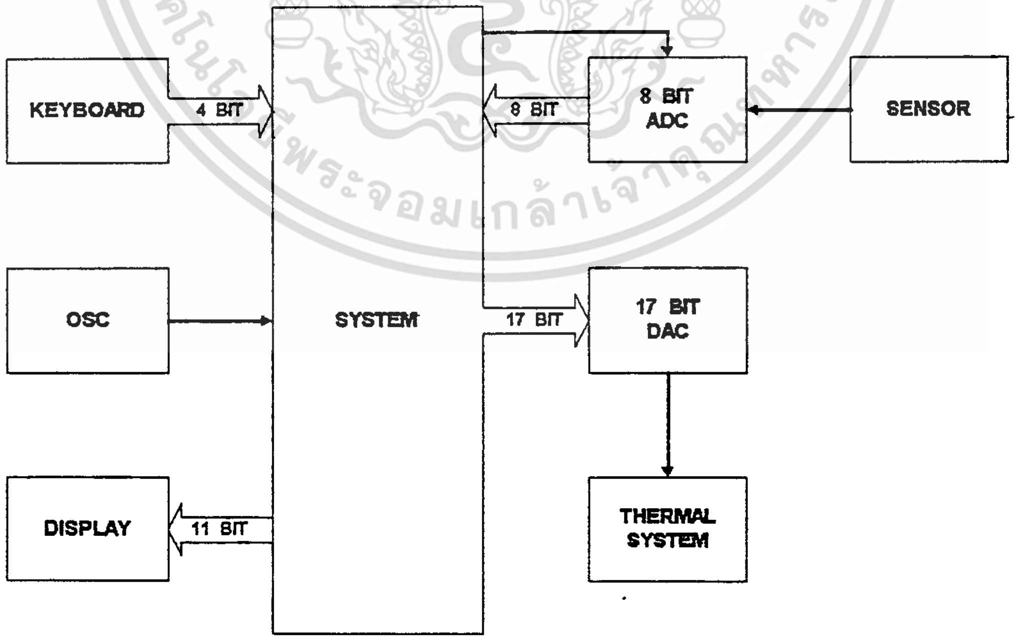
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
การเชื่อมต่อกับอุปกรณ์ภายนอก

ในบทนี้จะกล่าวถึงลักษณะของการเชื่อมต่อ ระบบที่ได้ออกแบบไว้กับอุปกรณ์ภายนอก โดยในการเชื่อมต่อกับอุปกณ์ภายนอกนี้ สามารถแบ่งอุปกรณ์ภายนอกที่จะนำมาเชื่อมต่อกับระบบของเราได้ 2 กลุ่ม คือ

1. อุปกรณ์ภายนอกที่ต้องต่อตามแบบที่กำหนดเท่านั้น ได้แก่ อุปกรณ์ในส่วนที่ใช้แสดงผล
2. อุปกรณ์ภายนอกที่รูปแบบในการเชื่อมต่อไม่ได้กำหนดตายตัวสามาเชื่อมต่อได้หลายลักษณะได้แก่ สวิชต์, วงจร ADC, วงจร DAC

ดังที่ได้กล่าวมาแล้วว่าระบบควบคุมอุณหภูมินี้สามารถใ ช้ควบคุมอุณหภูมิได้ ทั้งแบบที่เป็นเครื่องทำความเย็นหรือทำเป็นเครื่องทำความร้อนได้ ทั้งนี้ขึ้นอยู่กับการเชื่อมต่อกับอุปกรณ์ภายนอก ซึ่งก็หมายความว่า ถ้าจะนำไปประยุกต์ใช้กับเครื่องทำความเย็น เอาท์พุทที่ออกมาจากระบบนั้นจะต้องนำไปเชื่อมต่อกับอุปกรณ์ทำความเย็นให้เหมาะสม และในทางกลับกัน ถ้าจะประยุกต์ใช้กับระบบที่เป็นเครื่องทำความร้อนนั้น เอาท์พุทของระบบจะต้องนำไปเชื่อมต่อกับระบบทำความร้อนให้เหมาะสม จากที่กล่าวมาข้างต้นนี้ พอที่จะเขียนรูปการเชื่อมต่ออย่างคร่าวๆ ได้ดังรูปที่ 7.1

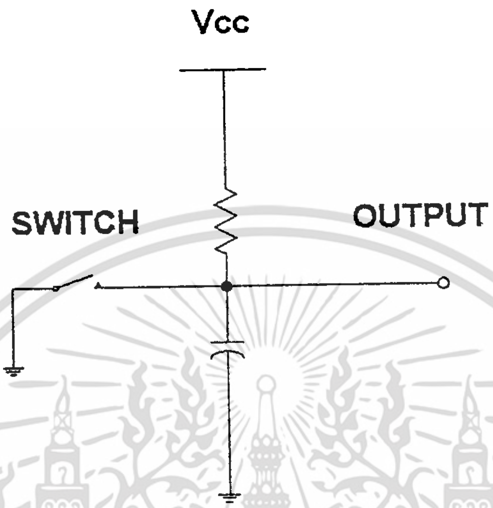


รูปที่ 7.1 แสดงถึงอุปกรณ์ภายนอกที่ระบบต้องการ

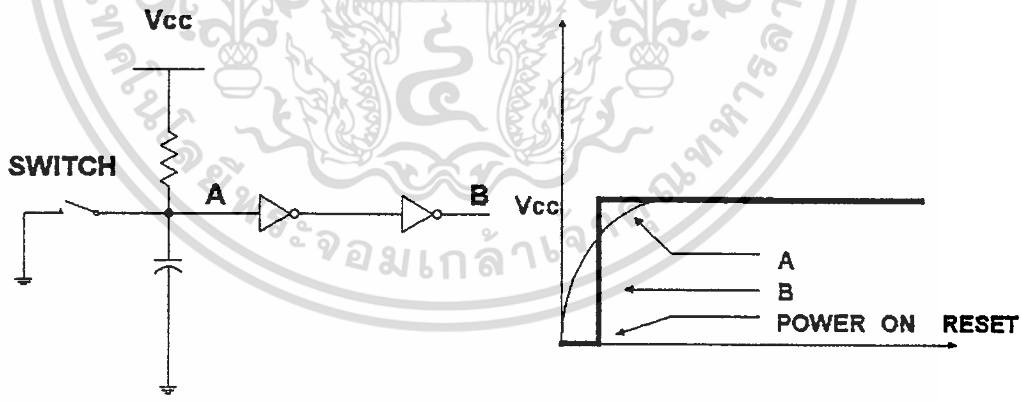
เอกสารนี้เป็นเอกสารที่ใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1 ส่วนคีย์บอร์ด

ในส่วนนี้จะต้องต่อสวิชต์เข้ากับระบบโดยที่ระบบจะรับค่าคีย์บอร์ดที่ระดับลอจิก " 0 " ดังแสดงในรูปที่ 7.1.1



รูปที่ 7.1.1 (ก) แสดงตัวอย่างการเชื่อมต่อกับคีย์บอร์ด



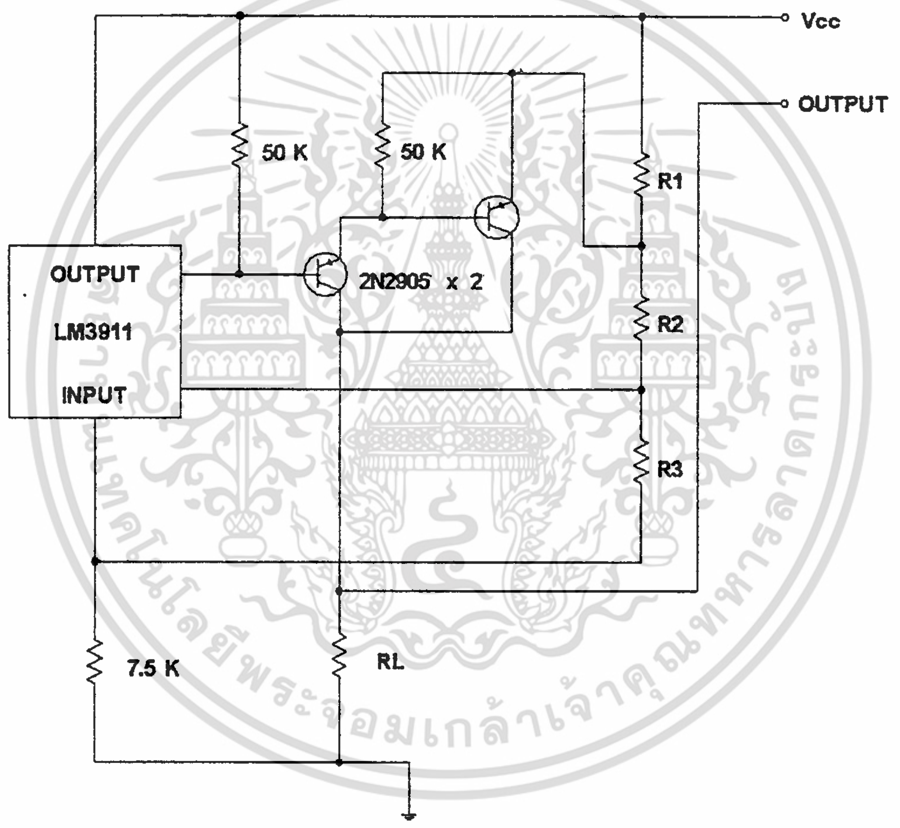
รูปที่ 7.1.1 (ข) แสดงการทำงานของ Power On Reset

จากรูปที่ 7.1.1 (ก) และ (ข) จะเห็นว่ามีตัวเก็บประจุต่ออยู่เพื่อที่จะลดสัญญาณรบกวนอันเนื่องมาจากหน้าสัมผัสของคีย์บอร์ด และคีย์บอร์ดของขา Reset จะมีความพิเศษตรงที่ว่าต้องทำ Power On Reset กล่าวคือ เมื่อเริ่มจ่ายไฟแก่ระบบ ระบบก็จะถูก Reset 1 ครั้ง ดังรูปที่ 7.1.1(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ส่วนเซนเซอร์อุณหภูมิ

เนื่องจากตอนที่ออกแบบระบบไว้ในเบื้องต้น ได้กำหนดไว้ให้อุณหภูมิที่ใช้อยู่ในหน่วยเซลเซียส ดังนั้นเพื่อความเหมาะสม จึงควรใช้เซนเซอร์อุณหภูมิที่เหมาะกับอุณหภูมิในระบบเซลเซียส และเซนเซอร์อุณหภูมิที่จะใช้นี้ ก็ควรจะมีความละเอียดมากกว่าหรือเท่ากับความละเอียดของอินพุทของระบบ ซึ่งในที่นี้ ระบบควบคุมอุณหภูมินี้ใช้ความละเอียดของอุณหภูมิไว้ที่ 0.1 องศาเซลเซียส เซนเซอร์อุณหภูมิที่จะเลือกใช้ก็ควรจะวัดอุณหภูมิได้ละเอียดทางเท่าที่ระบบต้องการ



รูปที่ 7.2.1 แสดงตัวอย่างเซนเซอร์อุณหภูมิโดยใช้ IC3911

โดยที่ตัวแปรต่างๆ มีค่าดังนี้

$$R1 = \frac{(Vz)(10mV)(\Delta T)}{\frac{Vo}{RL}(Vz - 0.01To)}$$

$$R2 = \frac{0.01To - IoR1}{Io}$$

$$R3 = \frac{Vz}{Io} - R1 - R2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vz = Shunt regulator voltage

ΔT = Temperature span (Kelvin)

To = Temperature for zero output (Kelvin)

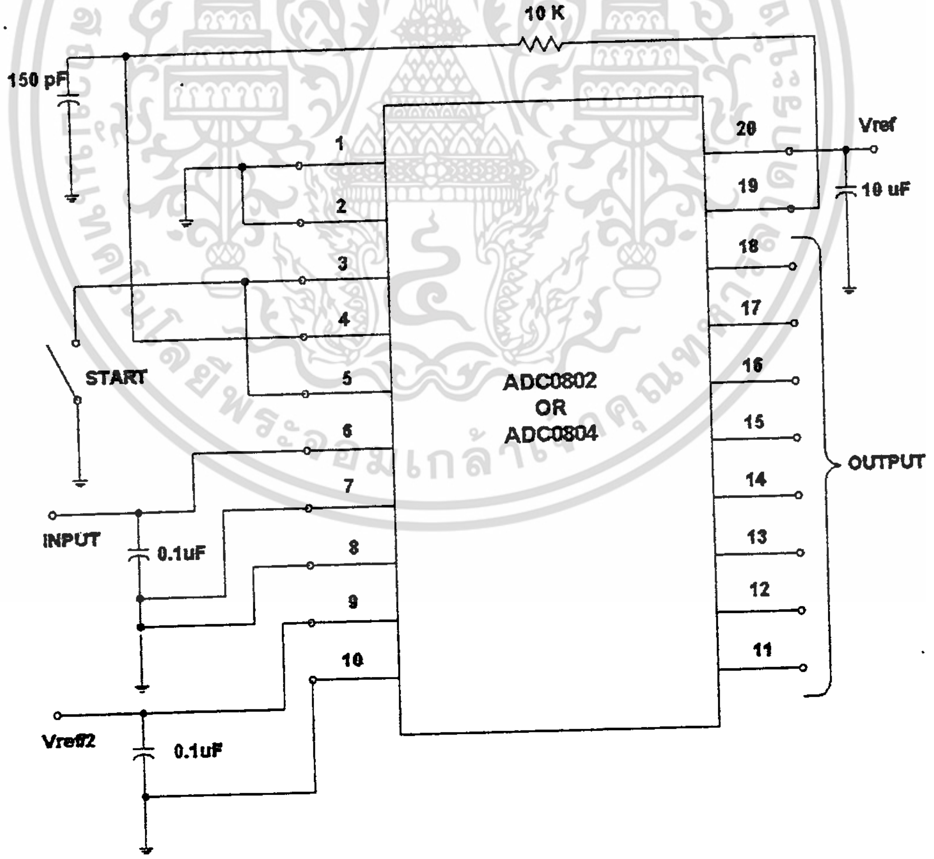
Vo = Full scale output voltage < 10 V

Io = Current through R1,R2,R3 at zero voltage

(Typically 100uA to 1.0mA)

7.3 ส่วน ADC

สัญญาณที่ได้มาจากเซนเซอร์อุณหภูมินั้นเป็นสัญญาณอนาลอก จึงต้องมีการแปลงเป็นสัญญาณดิจิตอลเสียก่อนจึงจะสามารถนำข้อมูลไปคำนวณหาค่าเอาท์พุทที่เหมาะสมได้ ซึ่งวงจร ADC ที่ระบบต้องการก็คือ จะต้อง Enable ได้โดยสายสัญญาณขนาด 1 บิต และ ค่าของข้อมูลต้องเป็นข้อมูลขนานขนาด 8 บิต

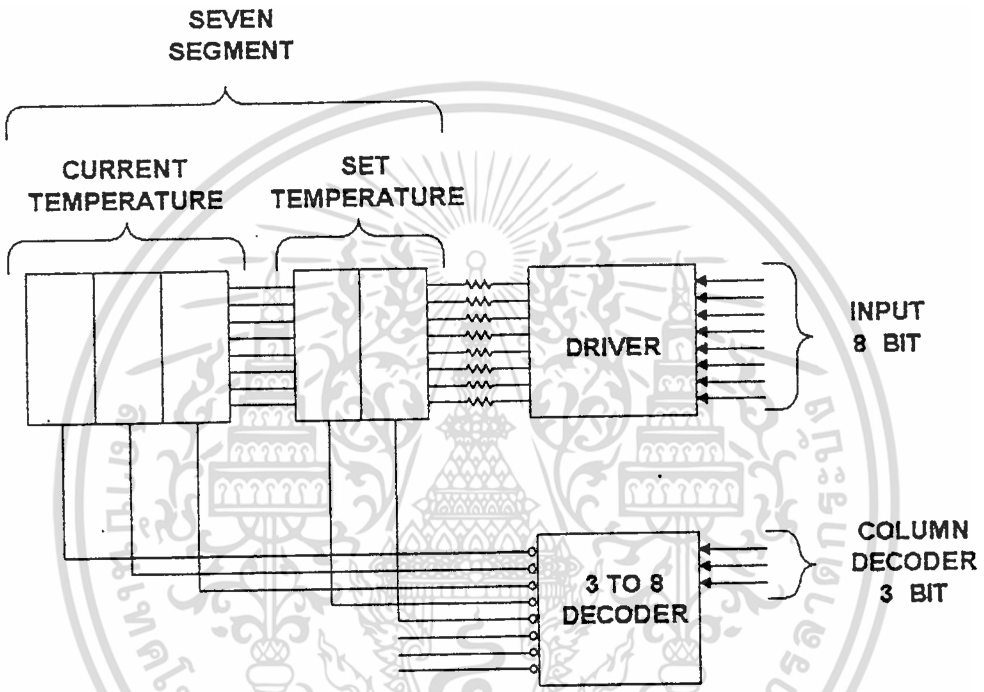


รูปที่ 7.3.1 แสดงตัวอย่างการเชื่อมต่อกับ ADC เบอร์ ADC0802 หรือ ADC0804

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 ส่วนแสดงผล

ในส่วนนี้จะต้องใช้ Seven Segment แสดงผลเท่านั้นเพราะว่าระบบควบคุมอุณหภูมินี้จะส่งค่าข้อมูลออกไปแสดงผลและเวลาในการ Scan Display ให้เหมาะสมกัน การใช้ Seven Segment เท่านั้น และจะไม่สามารถใช้กับ Display ชนิดอื่นได้เช่น LCD โดยที่ลักษณะในการต่อจะแสดงดัง รูปที่ 7.4.1



รูปที่ 7.4.1 แสดงการเชื่อมต่อกับ Seven Segment

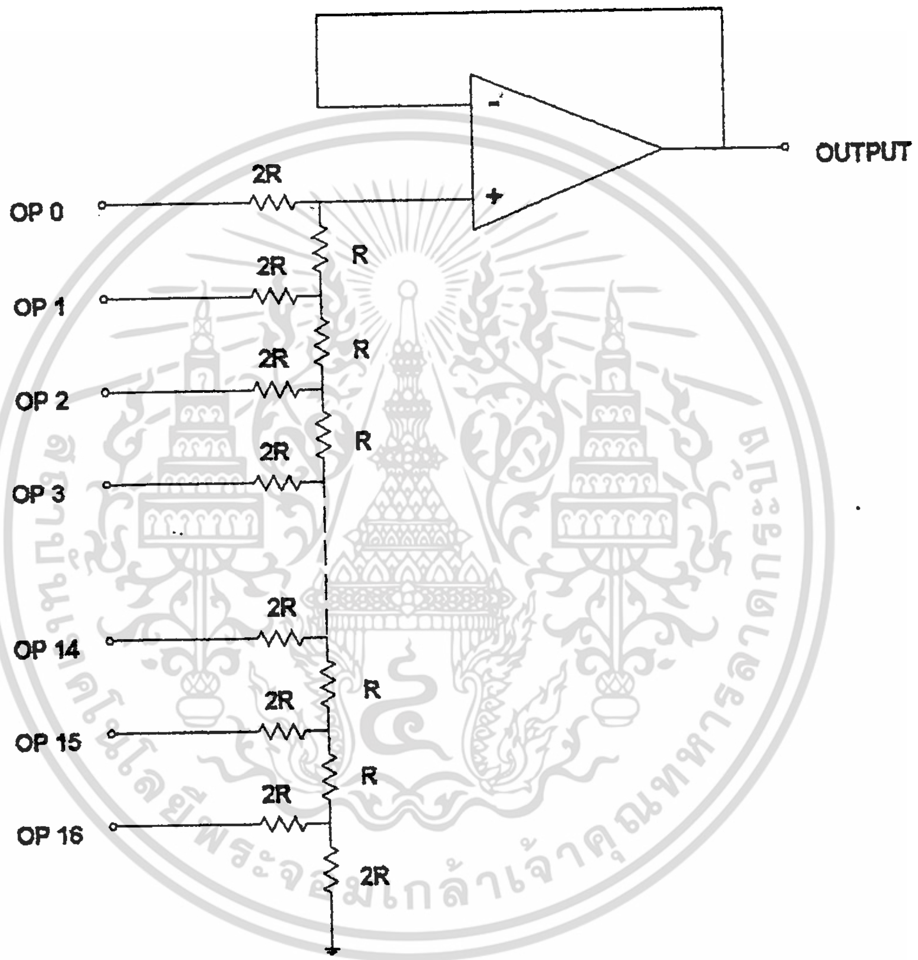
จากรูปจะเห็นว่ามี Seven Segment อยู่ 2 แถว ในแถวแรกที่มี Seven Segment 3 ตัว นั้น จะใช้แสดงค่าอุณหภูมิที่วัดไว้ ซึ่งจะมีทศนิยม 1 ตำแหน่ง ดังนั้นจึงต้องใช้ Seven Segment 3 ตัว ในแถวที่สองจะมี Seven Segment 2 ตัว จะใช้แสดงค่าอุณหภูมิที่ตั้งไว้ ซึ่งค่าที่จะตั้งนี้ อนุญาตให้ตั้งได้ในความละเอียดในหน่วยเซลเซียสเท่านั้น ดังนั้นจึงสามารถใช้ Seven Segment เพียง 2 ตัว

IC 3 TO 8 Decoder (เช่น 74LS138) นั้นใช้เป็นตัวเลือก Column ของ Seven Segment เพราะว่าสัญญาณที่ออกจากระบบวัดอุณหภูมิ จะมีเพียงแค่ 3 บิต ดังนั้นจึงต้องใช้วงจร Decoder เข้าช่วย ส่วน IC Driver (เช่น 74LS241) นั้นเป็น Driver เพื่อจ่ายกระแสให้กับ Seven Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5 ส่วนเอาท์พุท

ในส่วนนี้จะต้องทำการแปลงเอาท์พุทที่เป็นดิจิตอลให้เป็นอนาลอก เพื่อที่จะนำไปควบคุมระบบอุณหภูมิที่ต้องการได้ ในการเชื่อมต่อกับวงจร DAC นั้นสามารถทำได้หลายลักษณะ เช่น อาจจะใช้ความต้านทานต่อเป็น Ladder หรือใช้ไอซีสำเร็จรูปก็ได้ ดังแสดงในตัวอย่าง

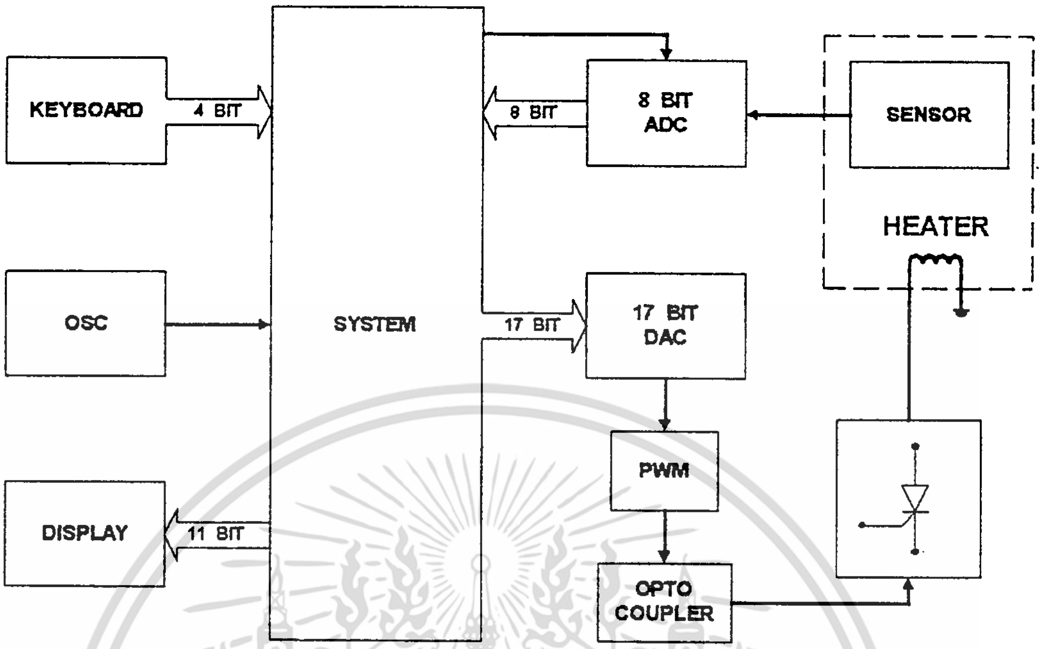


รูปที่ 7.5.1 แสดงตัวอย่างของวงจร DAC

7.6 ตัวอย่างการประยุกต์ใช้งานกับระบบทำความร้อน

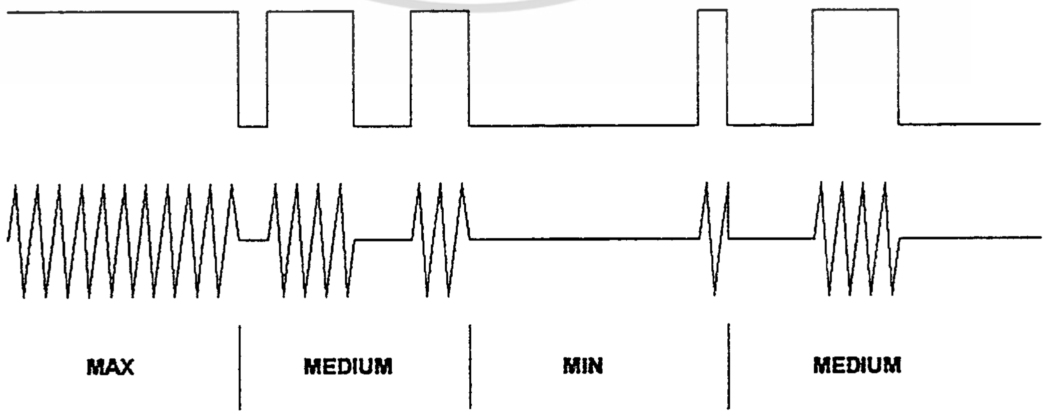
ในหัวข้อนี้จะแสดงตัวอย่างการประยุกต์ใช้กับระบบทำความร้อนซึ่งมีขดลวดความร้อนเป็นตัวสร้างความร้อน และในระบบนี้จะต้องมีการไหลวนของอากาศภายใน ห้องควบคุมอย่างสม่ำเสมอด้วยระบบควบคุมอุณหภูมินี้จึงจะสามารถใช้ได้ผลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6.1 แสดงตัวอย่างการประยุกต์ใช้งานกับระบบทำความร้อน

จากรูปจะเห็นว่าเมื่อเอาท์พุทที่เป็นค่าดิจิตอลผ่านส่วนวงจร DAC แล้วกลายเป็นสัญญาณอนาลอกแล้วนั้น ก็จะไปเชื่อมต่อกับวงจร Pulse Width Modulation ที่จะทำหน้าที่ควบคุมเวลาในการ " ON " และ " OFF " ของขดลวดความร้อน โดยที่จะต้องทำการแยกกราวด์ โดยใช้ Opto Isolator ก่อนเพื่อที่จะลดสัญญาณรบกวนที่จะเข้ามาจากสาย AC Line เมื่อผ่านส่วน Opto Isolator แล้ว ก็จะส่งสัญญาณไปควบคุม Triac ให้เป็นตัวควบคุมพลังงานที่จะจ่ายให้กับขดลวดความร้อน ซึ่งการควบคุมพลังงานที่จ่ายให้กับขดลวดความร้อนนี้สามารถอธิบายได้ดังรูปที่ 7.6.2



รูปที่ 7.6.2 แสดงการควบคุม Power ที่ขดลวดความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 7.6.2 สามารถอธิบายได้ว่า เมื่อมีสัญญาณ " ON " มาก จะทำให้ Power ที่ขดลวดความร้อนมาก แต่ถ้าช่วง " ON " น้อย ค่า Power ที่ขดลวดความร้อนก็จะน้อยไปด้วย และด้วยวิธีการดังกล่าวนี้ จึงสามารถที่จะควบคุมอุณหภูมิของระบบได้

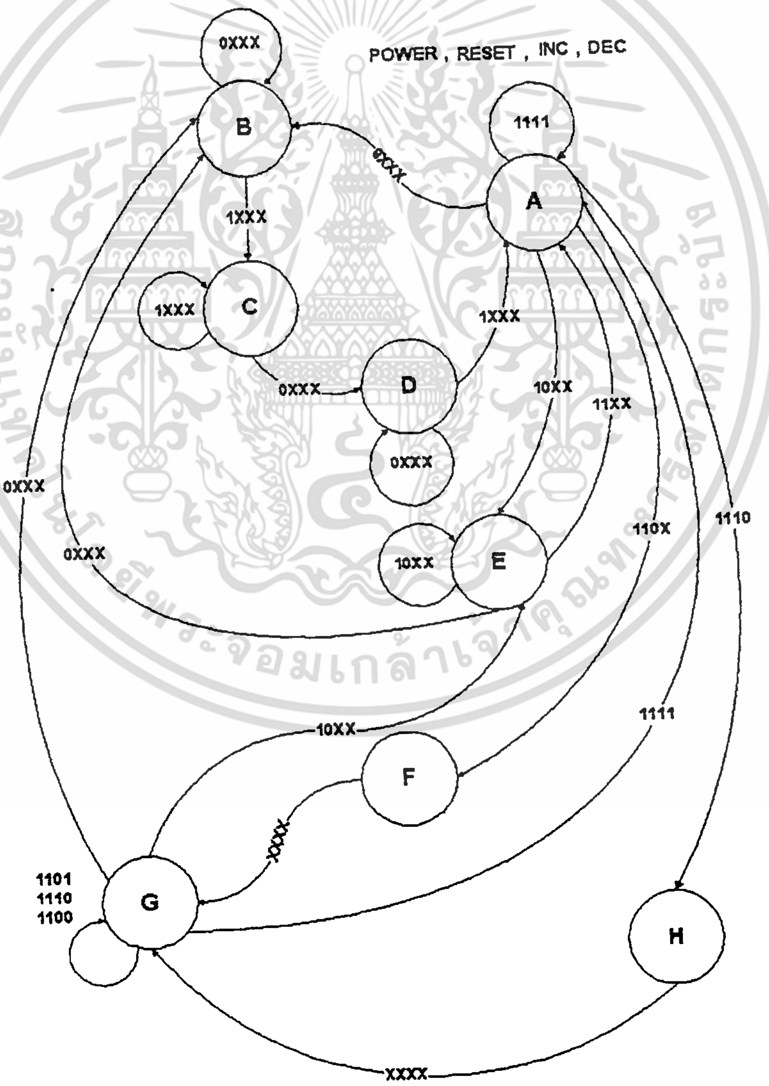
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8
โปรแกรมและการทดสอบ

ในบทนี้จะแสดงโปรแกรม คำอธิบายของแต่ละโปรแกรมและผลการ Simulation ของแต่ละส่วนย่อยๆ ในระบบที่ได้ทำการออกแบบไว้ดังนี้

8.1 โปรแกรมส่วนสแกนคีย์บอร์ด



รูปที่ 8.1 แสดง State Machine ของส่วนสแกนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้จะรับค่าคีย์บอร์ดอยู่ 4 แบบ คือ

-INC เป็นสวิทซ์ที่ใช้เพิ่มค่าอุณหภูมิ

-DEC เป็นสวิทซ์ที่ใช้ลดค่าอุณหภูมิ

-POWER ใช้เปิด-ปิดระบบ

-RESET ใช้รีเซตระบบ

ซึ่งค่าเอาท์พุทของส่วนนี้จะส่งออกไปเป็นเลขฐาน 2 ขนาด 8 บิต และสามารถตั้งอุณหภูมิได้ในช่วง 20-40 องศาเซลเซียส

สามารถอธิบายรูปที่8.1 ได้ดังนี้

สถานะ A เป็นสถานะที่รอรับการกดคีย์

สถานะ B เป็นสถานะที่ระบบถูกปิดและเช็คการปล่อยคีย์ POWER

สถานะ C เป็นสถานะที่ระบบถูกปิดและรอการกดคีย์ POWER

สถานะ D เป็นสถานะที่ระบบถูกเปิดและเช็คการปล่อยคีย์ POWER

สถานะ E เป็นสถานะที่ระบบถูกรีเซ็ตและรอการปล่อยคีย์ RESET

สถานะ F เป็นสถานะที่ระบบเพิ่มค่าอุณหภูมิ 1 องศาเซลเซียสแต่ต้องไม่เกิน 40 องศาเซลเซียส

สถานะ G เป็นสถานะที่ระบบเช็คการปล่อยคีย์ INC หรือ DEC

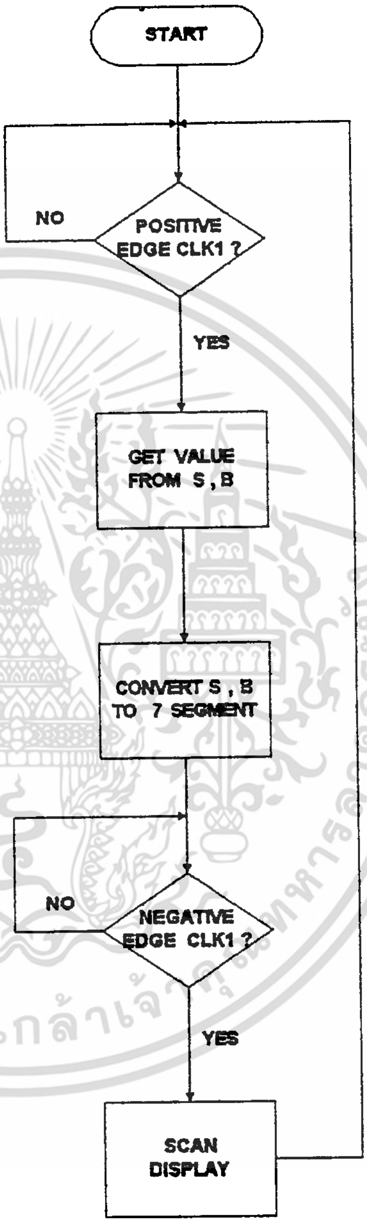
สถานะ H เป็นสถานะที่ระบบลดค่าอุณหภูมิ 1 องศาเซลเซียสแต่ต้องไม่เกิน 20 องศาเซลเซียส

8.2 โปรแกรมส่วนสแกนดิสเพลย์

ในส่วนนี้จะรับค่าอินพุทจากส่วนสแกนคีย์บอร์ด และส่วน Buffer Temperature เพื่อนำมาแปลงเป็นเลข BCD แล้วนำไปแสดงที่ Seven Segment อีกทีหนึ่ง และสัญญาณ Decode ขนาด 3 บิตนี้ใช้สำหรับเลือก Column ของ Seven Segment ที่จะใช้แสดงผล

โปรแกรมนี้มีหลักการทำงานดังต่อไปนี้ เมื่อพบขอบขาขึ้นของ CLK1 วงจรส่วนนี้จะนำค่าอุณหภูมิจากส่วนสแกนคีย์บอร์ดและ Input Buffer Temperature มาเปลี่ยนเป็นข้อมูลชนิด Seven Segment แบบคอมมอนคาโทด หลังจากนั้นจะทำการสแกนดิสเพลย์ที่ขอบขาลงของ CLK1 กล่าวคือ จะทำการส่งข้อมูลออกไปทีละ 1 Column โดยที่จะต้องมีการส่งสัญญาณเลือก Column ออกไปให้สอดคล้องกันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

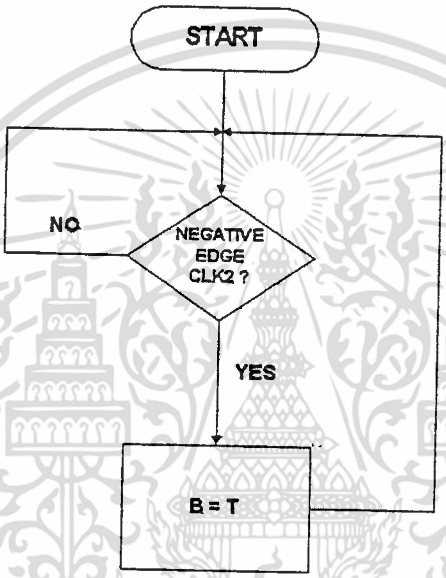


รูปที่ 8.2 แสดงการทำงานของส่วนสแกนดิสเพลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 โปรแกรมส่วน BUFFER TEMPERATURE

เป็นส่วนที่จะรับค่าอุณหภูมิจากภายนอกที่เป็นเลขฐานสองขนาด 8 บิต เข้ามาโดยมีสัญญาณควบคุมขนาด 1 บิต เพื่อใช้ควบคุมสัญญาณที่จะส่งมาจากวงจร ADC (Analog to Digital Converter) ให้ Synchronize กัน แล้วค่าอุณหภูมิที่ได้นี้จะส่งไปให้ส่วน Display และ Fuzzy Algorithm พร้อมๆ กัน



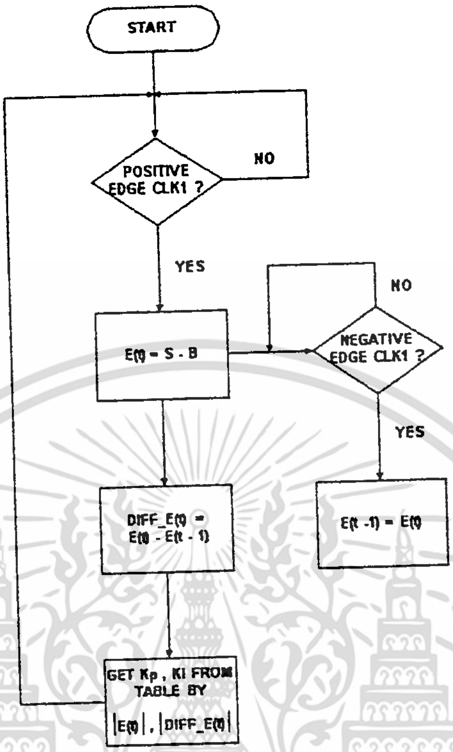
รูปที่ 8.3 แสดงการทำงานของส่วน Buffer Temperature

8.4 โปรแกรมส่วน FUZZY ALGORITHM

จะเห็นว่าอินพุท คือ ค่า T(t) และ Set Temp ก็จะเข้ามาคำนวณหาค่า E(t) แล้วนำค่านี้ออกไปคำนวณค่า ΔE อีกทีโดยที่ E(t-1) ได้มาจาก Shift Register ซึ่งค่า ΔE และ E(t) ต้องนำไปผ่านวงจร Absolute เพื่อที่จะเลือกเฉพาะค่าที่เป็นบวก แล้วนำค่า $|\Delta E|$ และ $|E(t)|$ ไปหาค่า K_p และ K_i ที่เหมาะตาม Algorithm Fuzzy โดยที่จะใช้เทคนิค Look Up Table ในการหาค่า

ค่าเอาท์พุทจากส่วนนี้คือค่า K_p , K_i และ E(t) จะถูกส่งออกไปในรูปของเลขฐานสองขนาด 5, 4 และ 8 บิต ตามลำดับ ก็เพราะว่าค่า K_p สูงสุดคือ 24 และ K_i มีค่าสูงสุดเท่ากับ 12 จึงสามารถแทนด้วยเลขฐานสองดังกล่าวข้างต้น ส่วนค่า E(t) จะเป็นค่าที่ใช้ในการคำนวณในลำดับถัดไป จึงต้องส่งออกไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

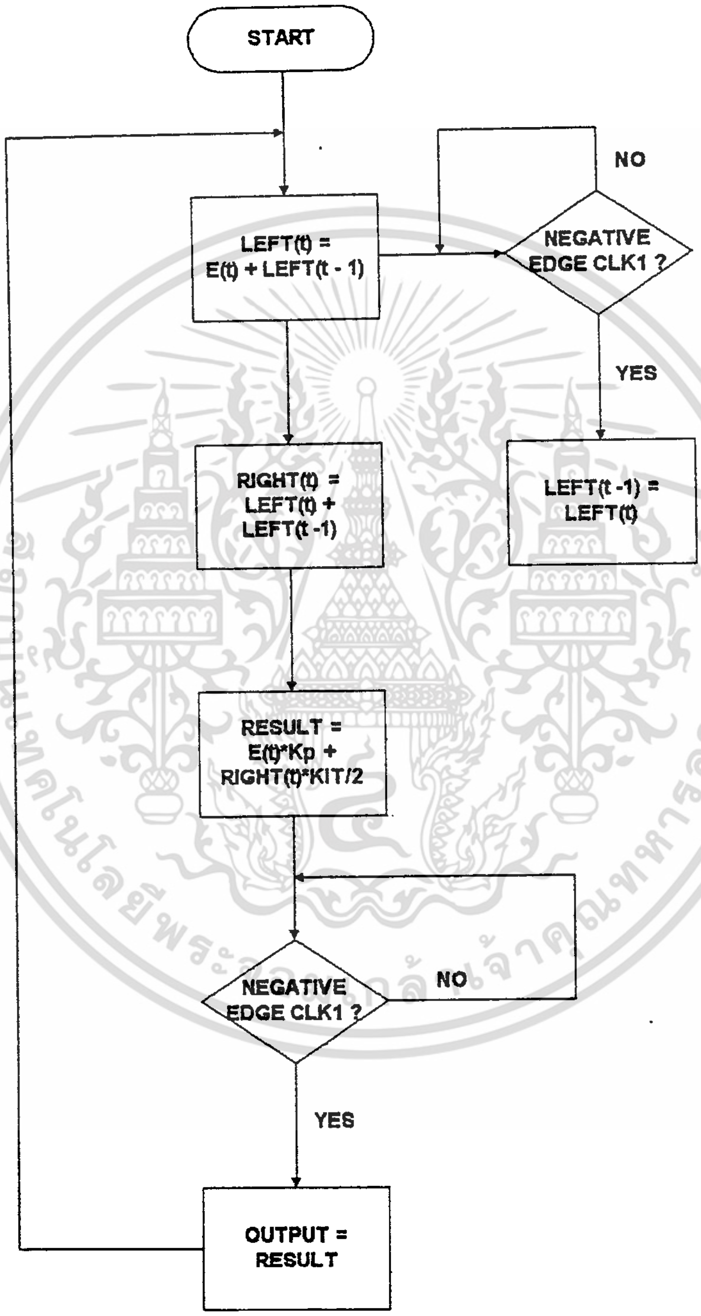


รูปที่ 8.4 แสดงการทำงานของส่วน FUZZY ALGORITHM

8.5 โปรแกรมส่วนคำนวณแบบ PI

จากรูปที่ 6.7 จะเห็นว่าต้องมีตัวแปรที่ใช้ในการคำนวณอยู่ 3 ตัว โดยในที่นี้จะตั้งชื่อตัวแปรตัวที่ 1 ว่า LEFT ซึ่งเมื่อผ่านตัว Delay จะได้ตัวแปรที่ 2 คือ LEFT-1 ส่วนตัวแปรที่ 3 ซึ่งเกิดจากผลรวมของ LEFT และ LEFT-1 เรียกว่า RIGHT ซึ่ง CLK ที่จะใช้ในการทำงานของวงจรส่วนนี้คือ CLK1 และสามารถอธิบายการทำงานได้ดังนี้ เมื่อมีการเปลี่ยนแปลงของสัญญาณจะเริ่มมีการคำนวณให้ค่า LEFT = ERROR + LEFT-1 หลังจากนั้นจะทำการคำนวณค่า RIGHT ซึ่งเท่ากับ LEFT + LEFT-1 โดยที่ค่า LEFT-1 มาจากการเก็บค่า LEFT ก่อนหน้านี้โดยมีขอบขาลงของ CLK1 เป็นตัวกำหนด ค่าเอาท์พุตจะมีค่าเท่ากับ (ERROR*Kp) + (RIGHT*Ki*T/2) ซึ่งค่า T มีค่าเท่ากับส่วนกลับของความถี่ที่ใช้ในการ Sampling ค่าเอาท์พุตที่ได้นี้จะถูกส่งออกไปเมื่อเกิดขอบขาลงของ CLK1 เพื่อที่ระยะห่างของการเปลี่ยนแปลงสัญญาณเอาต์พุตจะได้มีค่าเท่ากับคาบเวลาในการ Sampling

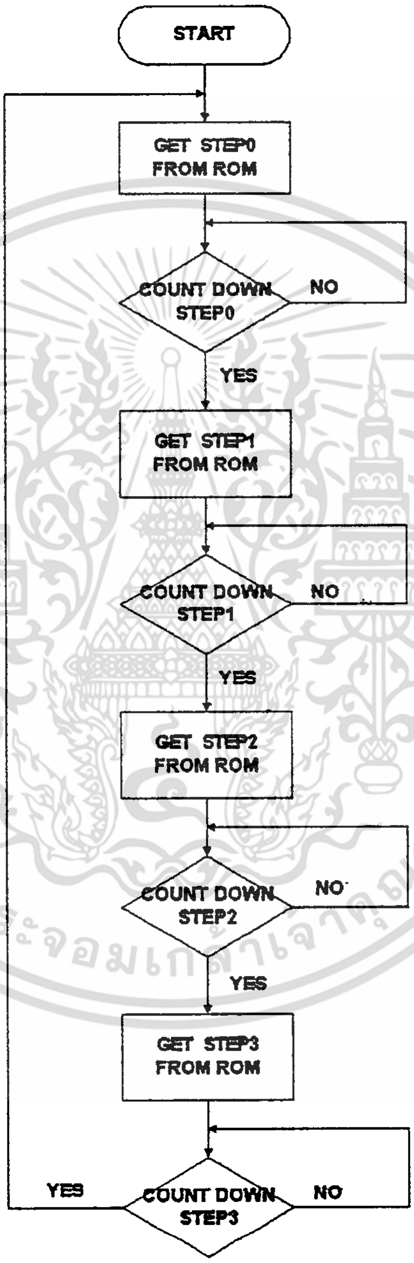
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.5 แสดงการทำงานของส่วนคำนวณแบบพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

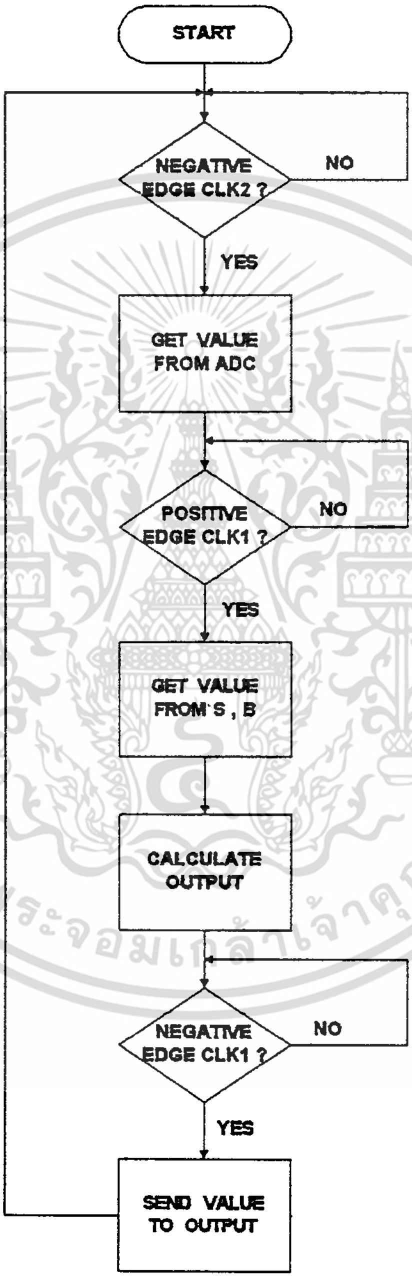
8.6 ส่วนสร้างสัญญาณนาฬิกา 2 เฟส



รูปที่ 8.6 แสดงการทำงานของส่วนสร้างสัญญาณนาฬิกา 2 เฟส

เอกสารนี้เป็นเอกสาร ... นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.7 โปรแกรมโดยรวมทั้งหมด

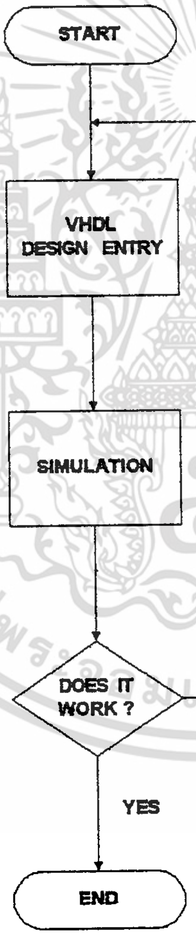


รูปที่ 8.7 แสดงการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสาร ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.8 การทดสอบโปรแกรม (SIMULATION)

เมื่อเขียน Source Code ของภาษา VHDL ได้แล้ว จะต้องนำมา Compile ให้ผ่านก่อน จึงจะสามารถทำการ Simulation โดยโปรแกรม Simulator ได้ และในการ Simulation โปรแกรมต่างๆ ข้างต้น จำเป็นที่จะต้องมี Input Pattern ที่เหมาะสม ซึ่งวิธีที่นิยมใช้โดยทั่วๆ ไป คือวิธีที่จะเขียนโปรแกรมวีเอชดีแอลขึ้นมาอีกโปรแกรมหนึ่ง เพื่อสร้าง Input Pattern ให้กับโปรแกรมที่เราต้องการทดสอบ ซึ่งโปรแกรมนี้เรียกว่า Test Bench จะสามารถสรุปเป็นโฟลว์ชาร์ตได้ดังรูป

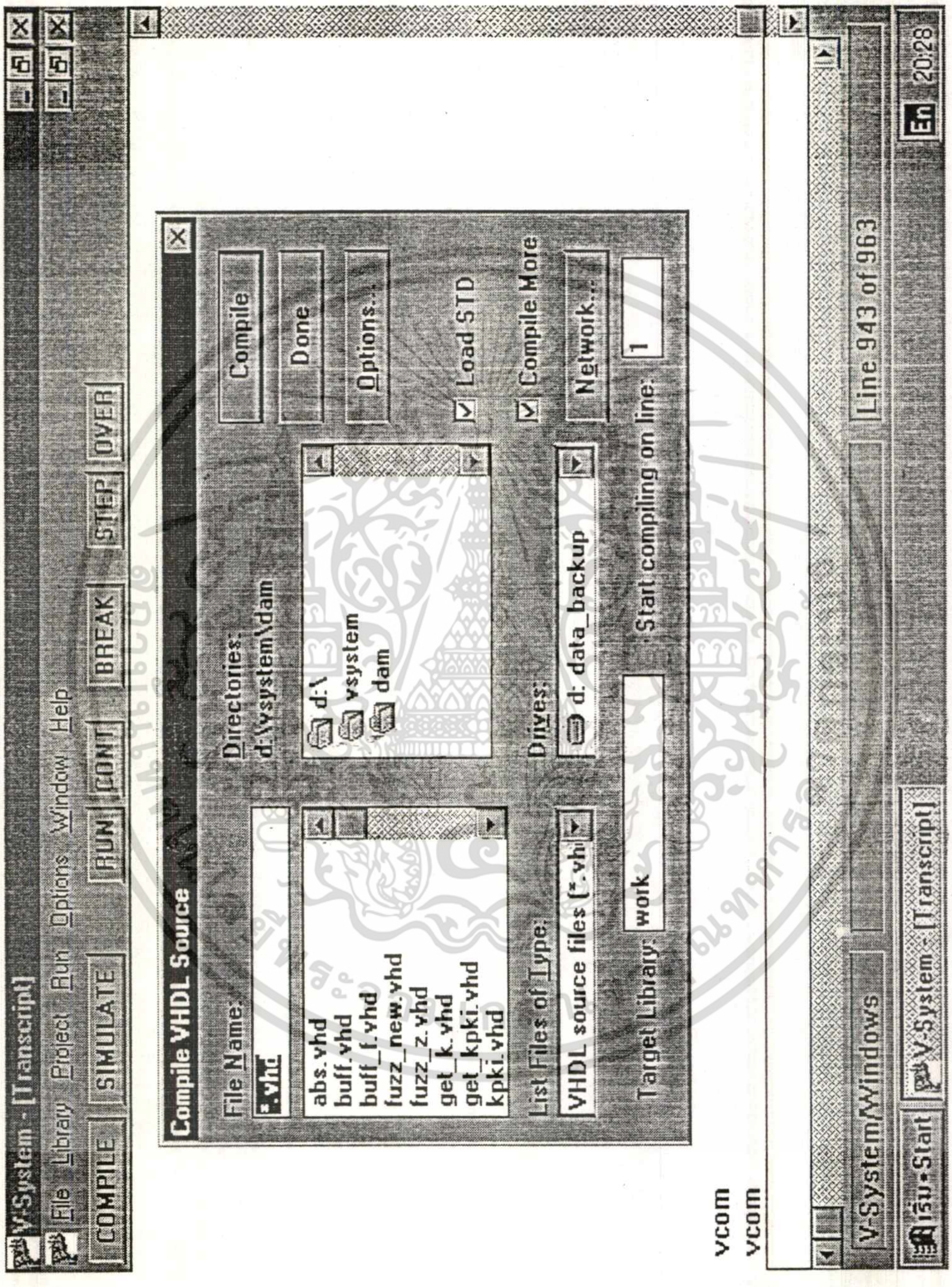


รูปที่ 8.8.1 แสดงขั้นตอนการเขียนโปรแกรมภาษาวีเอชดีแอล

หมายเหตุ

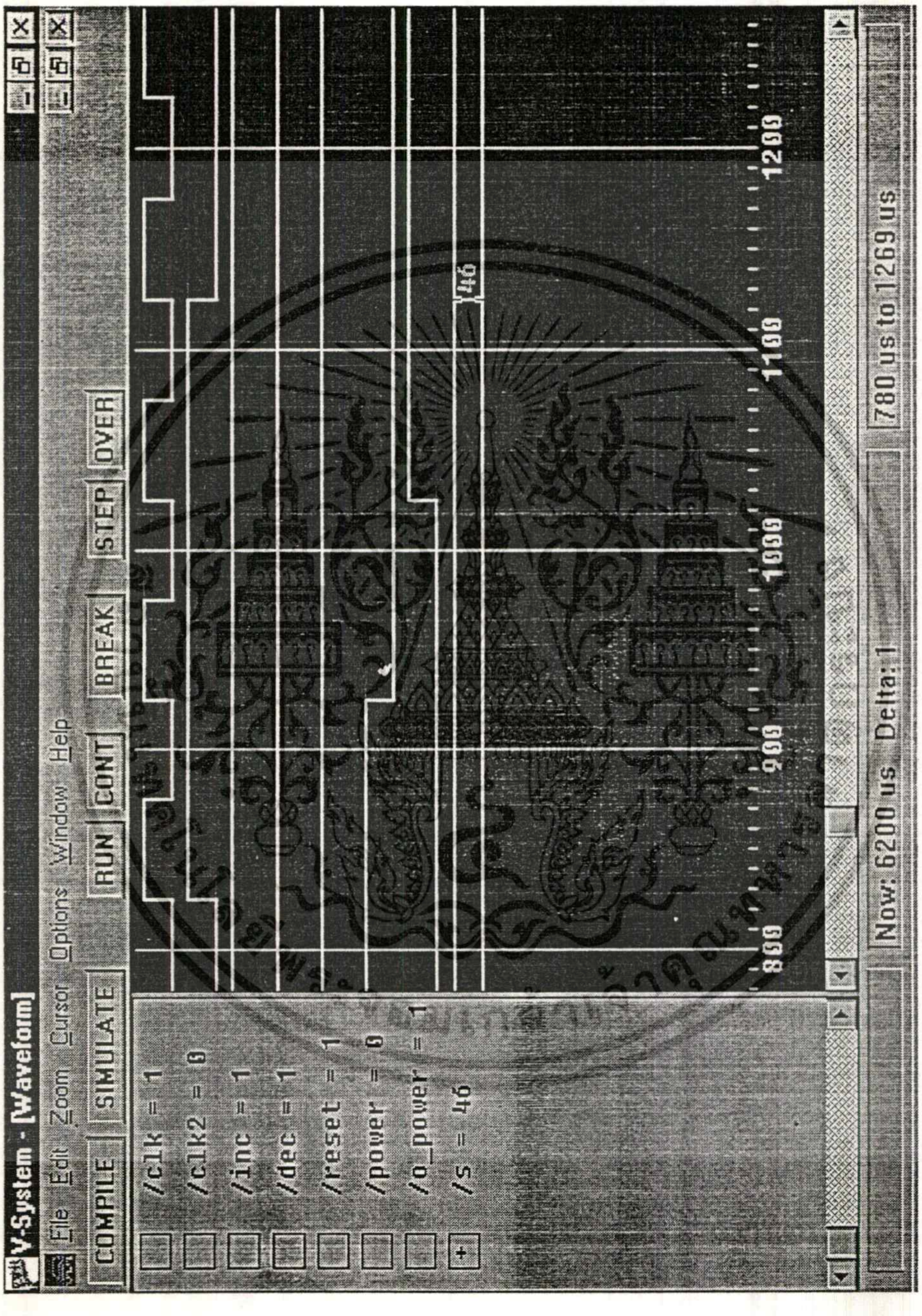
1. โปรแกรมที่ใช้ในการ Compile คือ โปรแกรม V- System
2. โปรแกรมที่ใช้ในการ Simulation คือ โปรแกรม V- System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



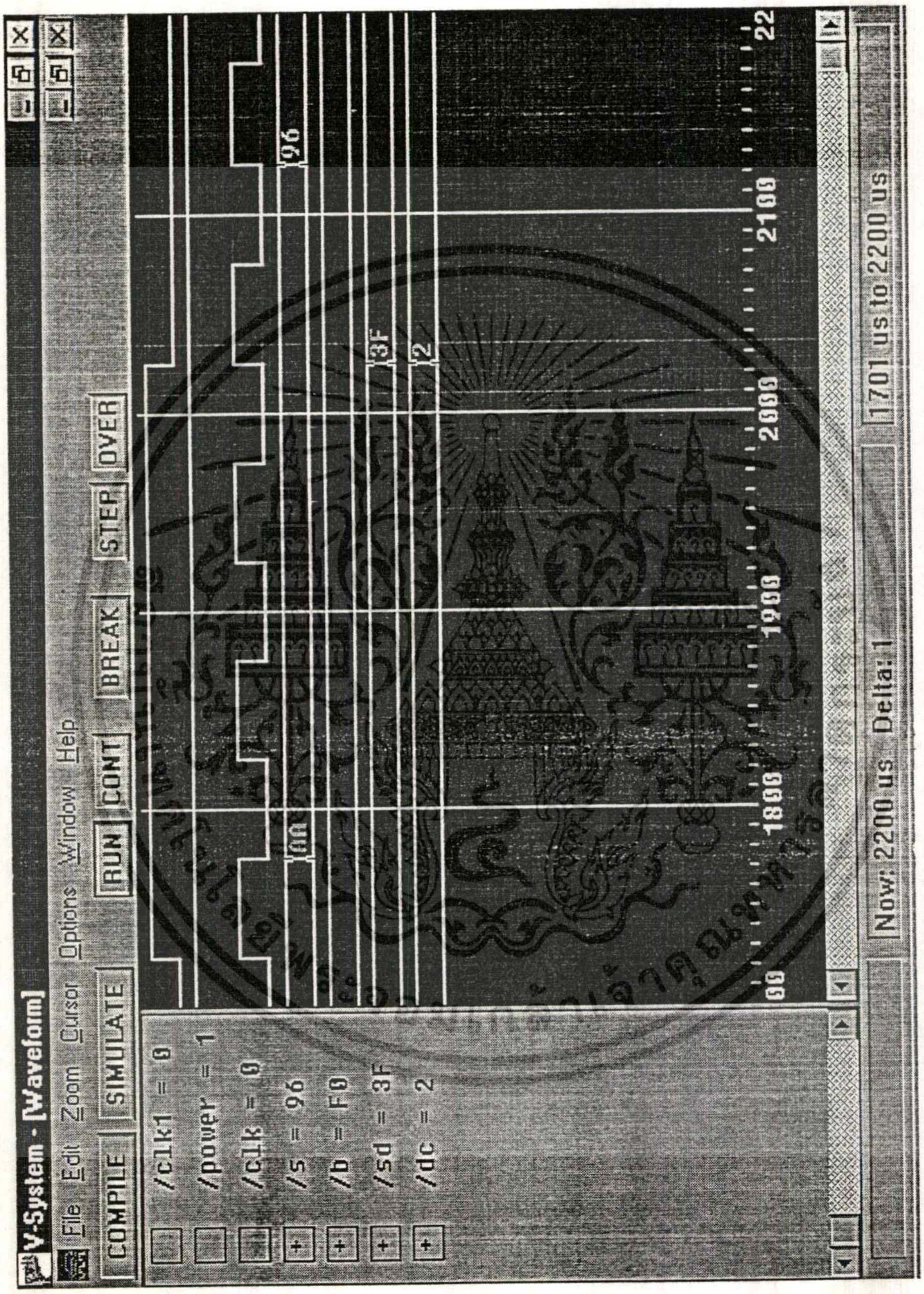
รูปที่ 8.8.2 แสดงการ Compile โดยใช้โปรแกรม V - System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



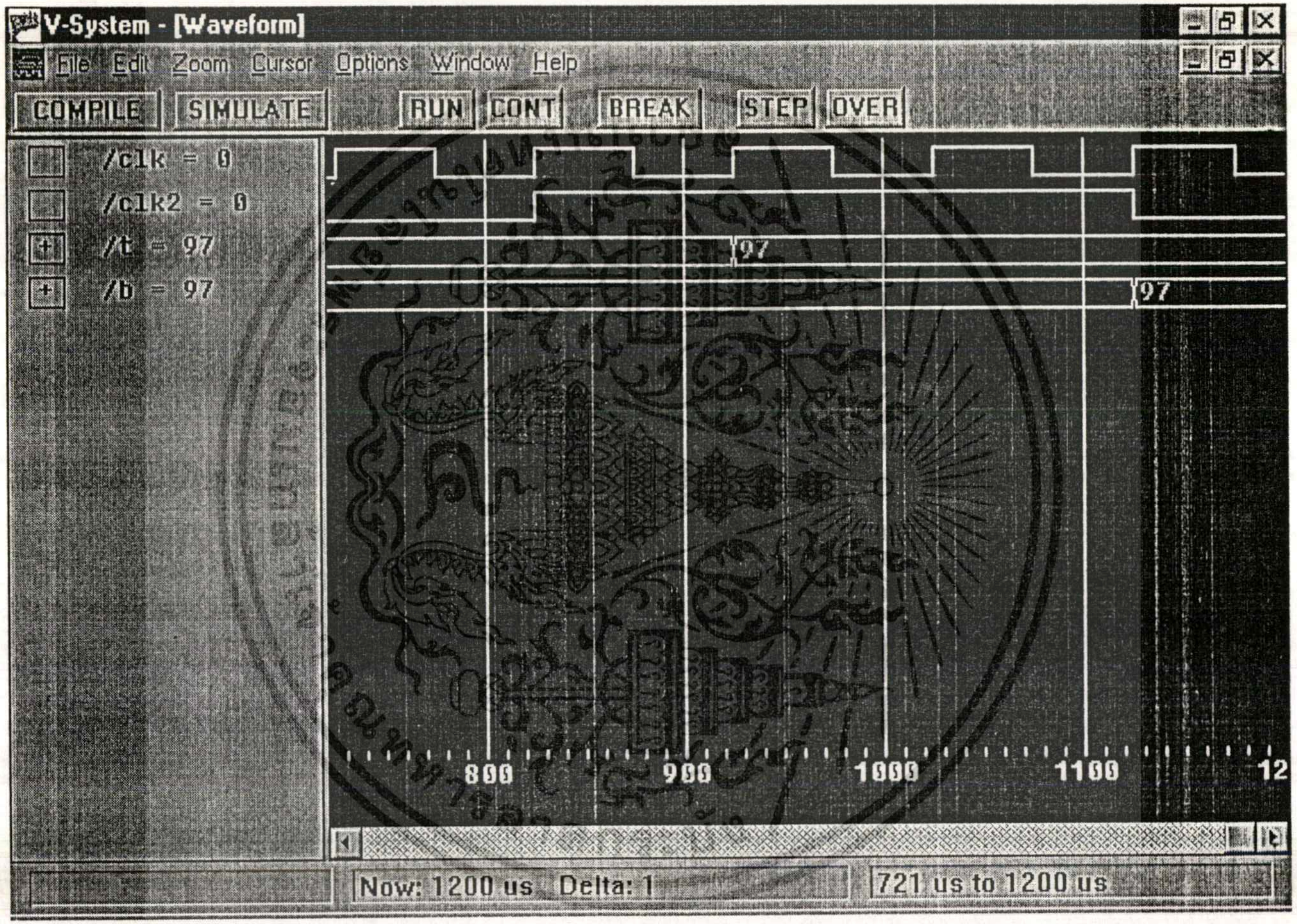
รูปที่ 8.8.3 แสดง Timing Diagram ของส่วนสแกนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



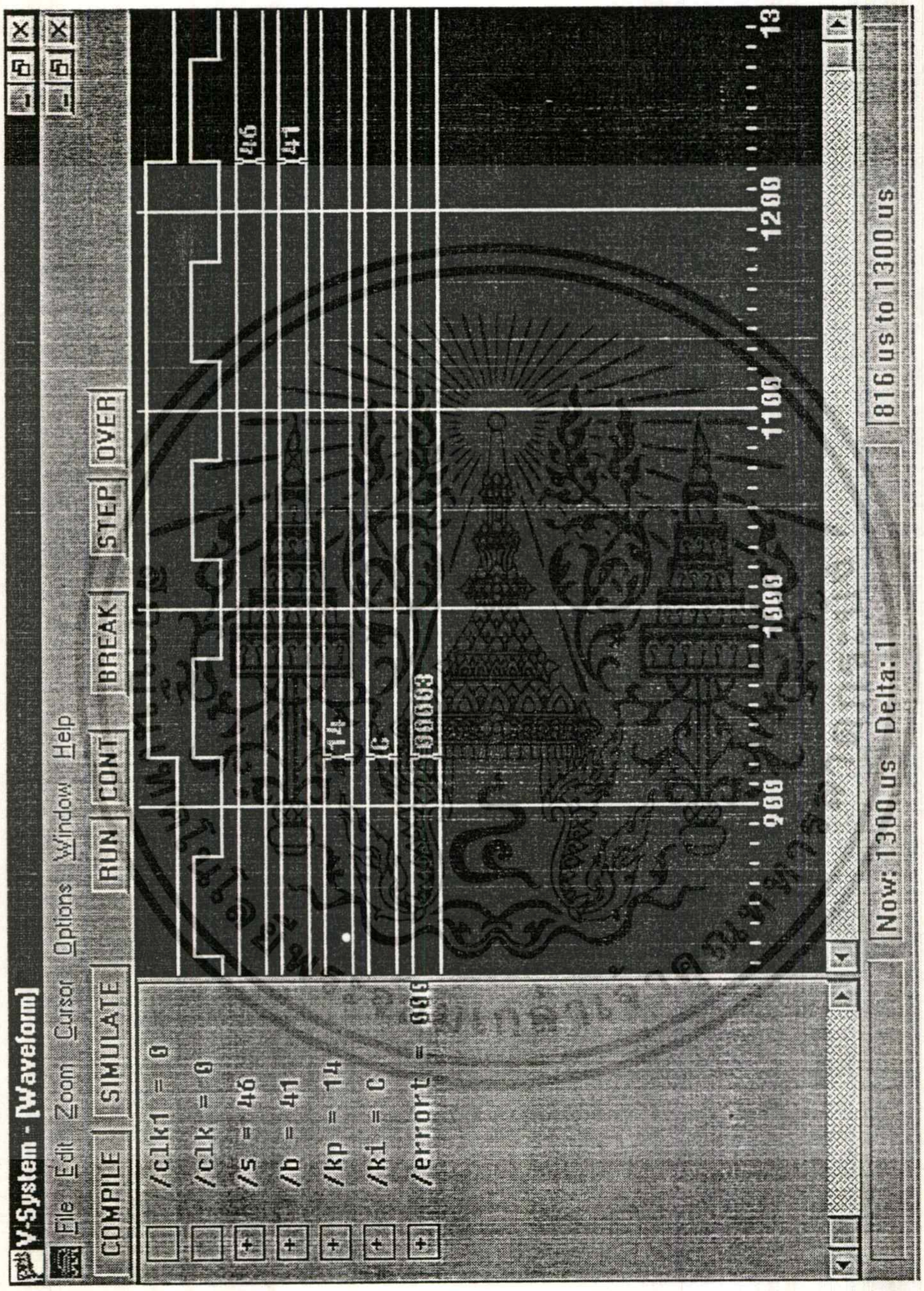
รูปที่ 8.8.4 แสดง Timing Diagram ของส่วนสแกนดิสเพลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



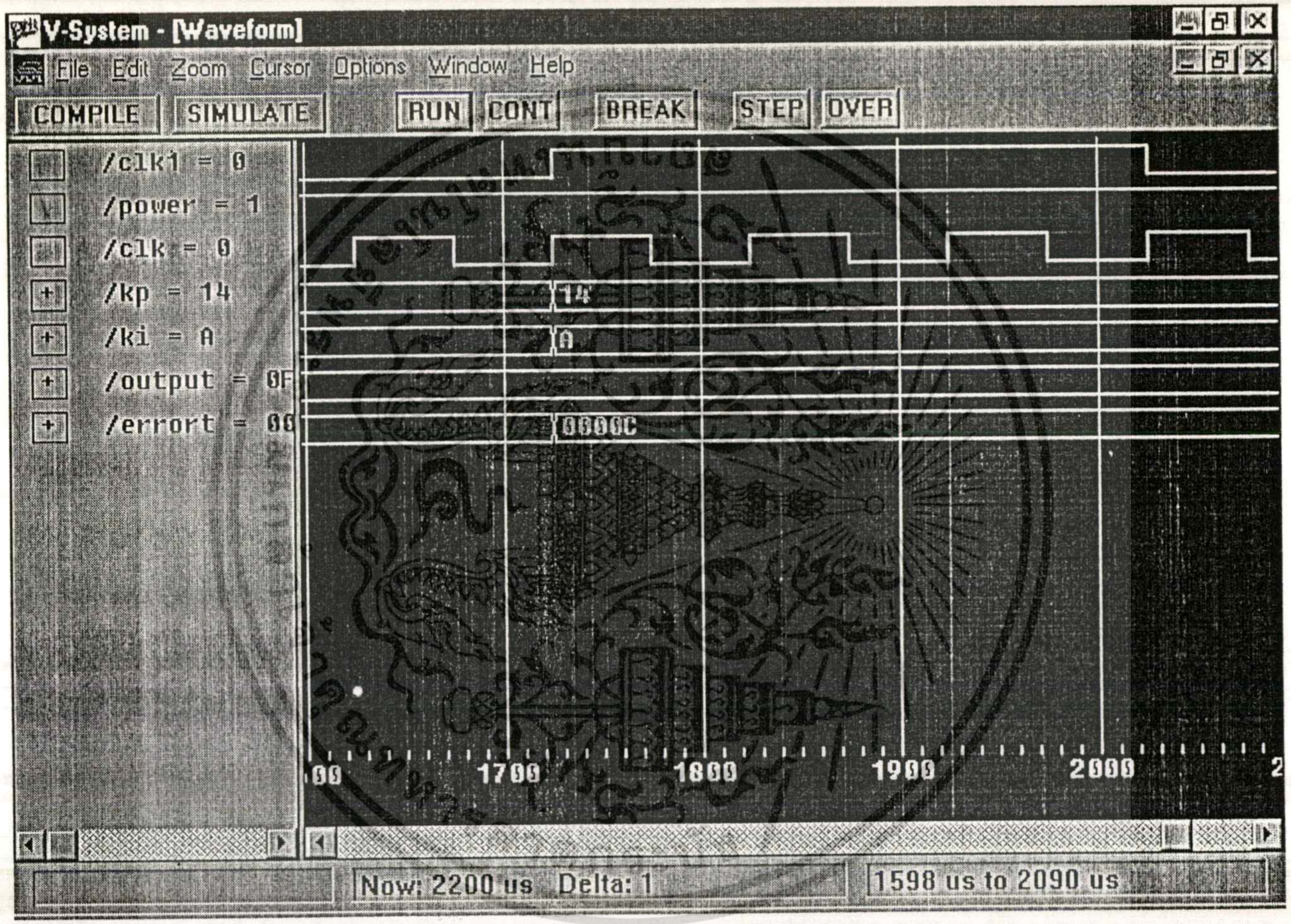
รูปที่ 8.8.5 แสดง Timing Diagram ของส่วน BUFFER TEMPERATURE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



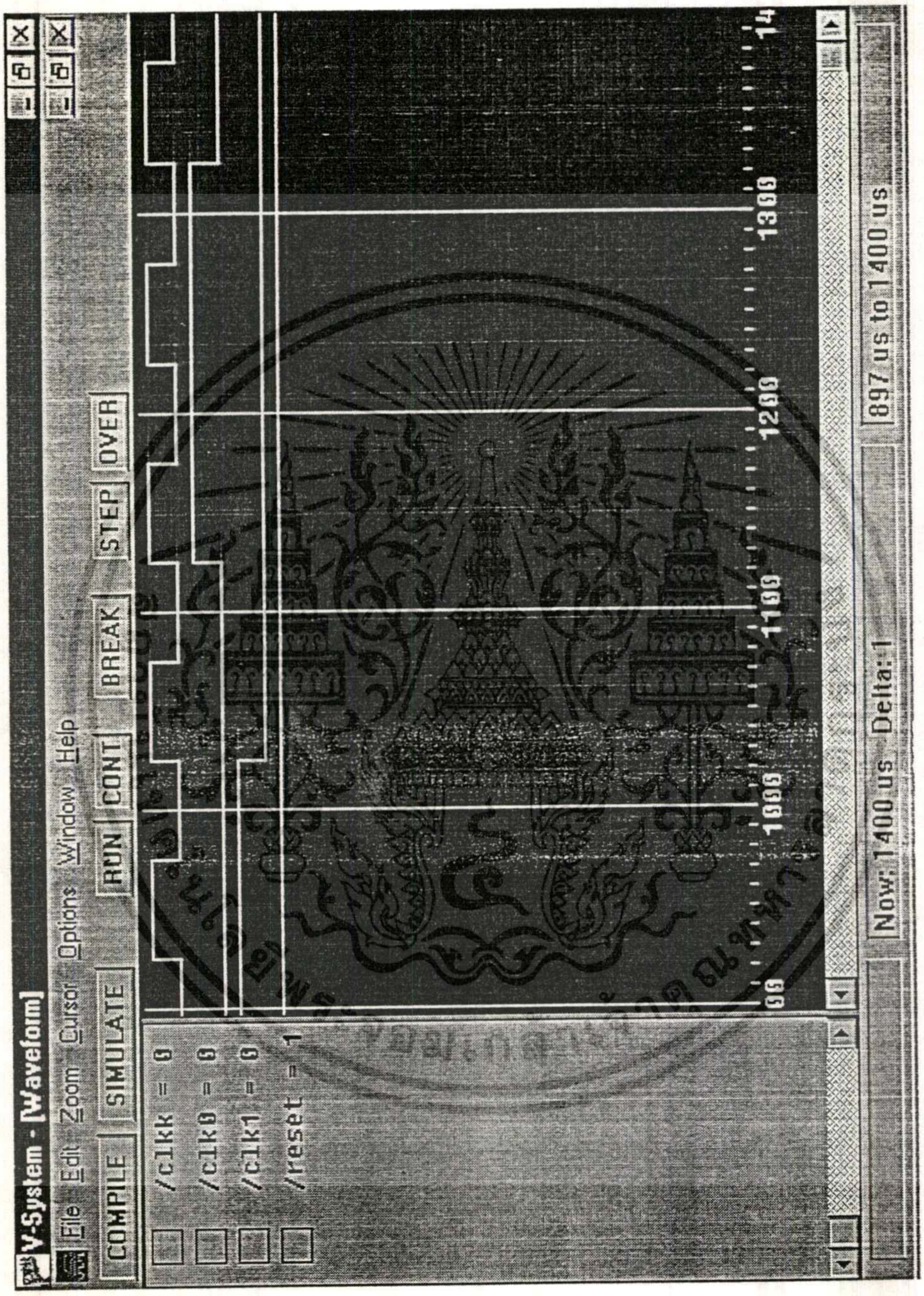
รูปที่ 8.8.6 แสดง Timing Diagram ของส่วน FUZZY ALGORITHM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



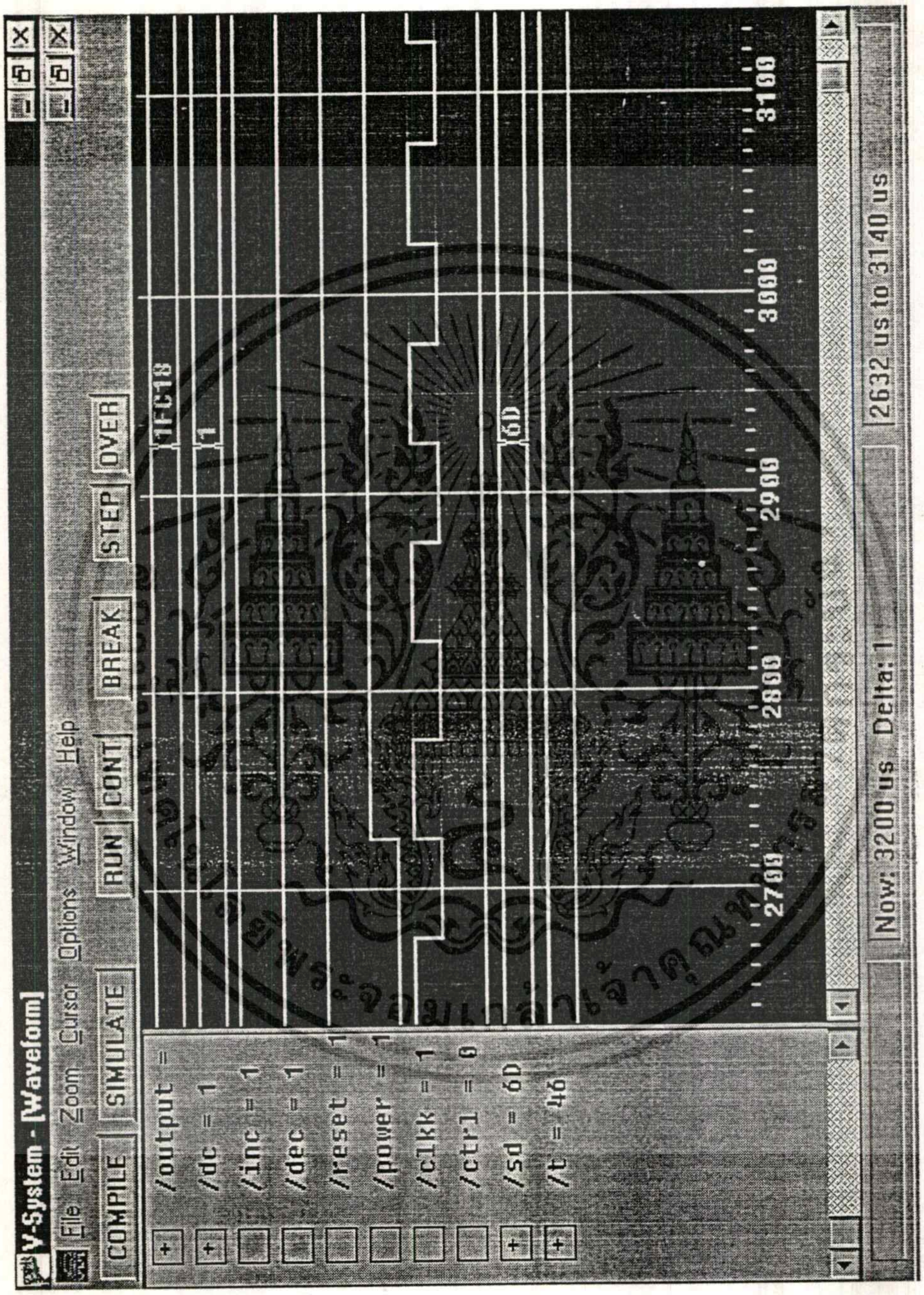
รูปที่ 8.8.7 แสดง Timing Diagram ของส่วน PI CONTROLLER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.8.8 แสดง Timing Diagram ของส่วนสร้างนาฬิกา 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.8.9 แสดง Timing Diagram ของระบบโดยรวมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

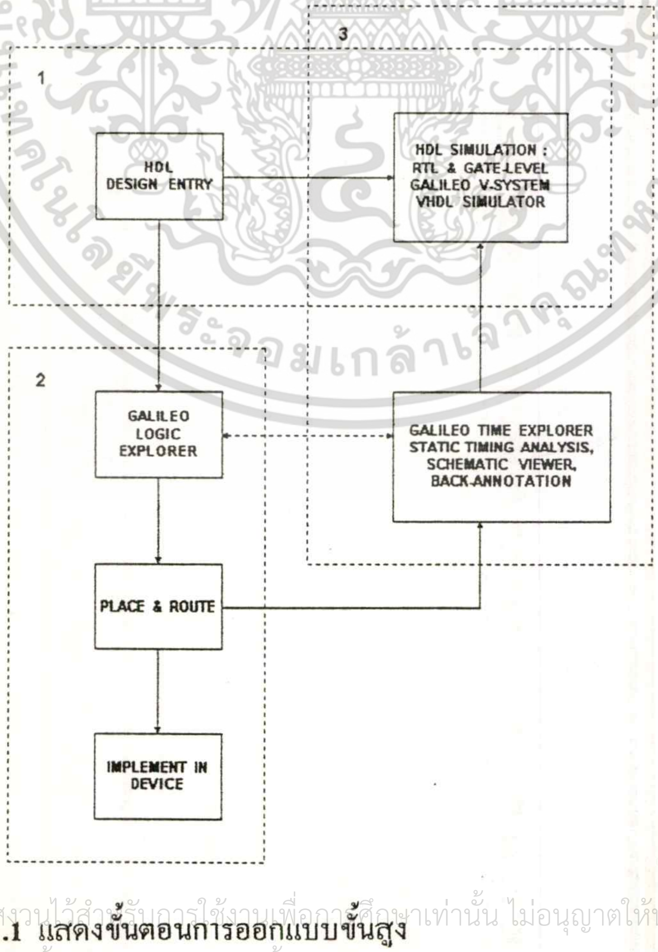
บทที่ 9
การสังเคราะห์วงจร

ในบทนี้จะกล่าวถึงการสังเคราะห์วงจรให้อยู่ในระดับเกททั้งนี้จะใช้เทคโนโลยีของ FPGA เบอร์ XC4000 เป็นหลัก แต่ทั้งนี้มิได้หมายความว่า การสังเคราะห์วงจรจะใช้เทคโนโลยีอื่นไม่ได้ ในการเลือกเทคโนโลยีนั้น จะต้องเลือกให้เหมาะสมกับงานที่ต้องการใช้

9.1 ตัวอย่างของการออกแบบในขั้นสูง

การออกแบบในขั้นสูงนั้น สามารถแบ่งออกได้เป็น 3 ขั้นตอน คือ

1. การกำหนดฟังก์ชันและการ SIMULATION ที่ไม่ขึ้นกับเทคโนโลยีใด
2. การสร้างสถาปัตยกรรมแบบที่ต้องการ
3. การตรวจสอบในระดับเกท



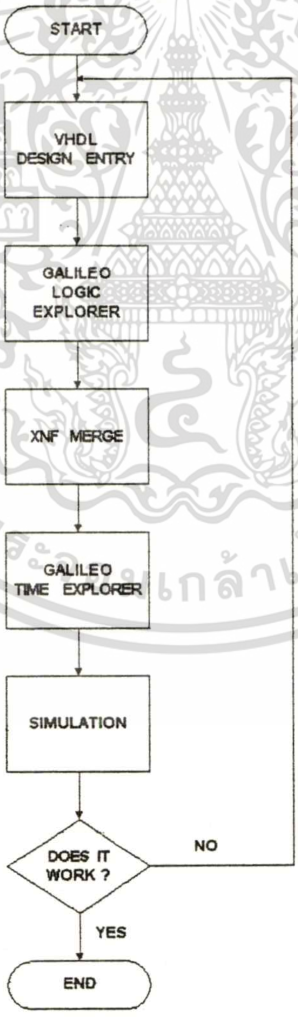
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 9.1.1 แสดงขั้นตอนการออกแบบขั้นสูง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.2 การสังเคราะห์วงจร

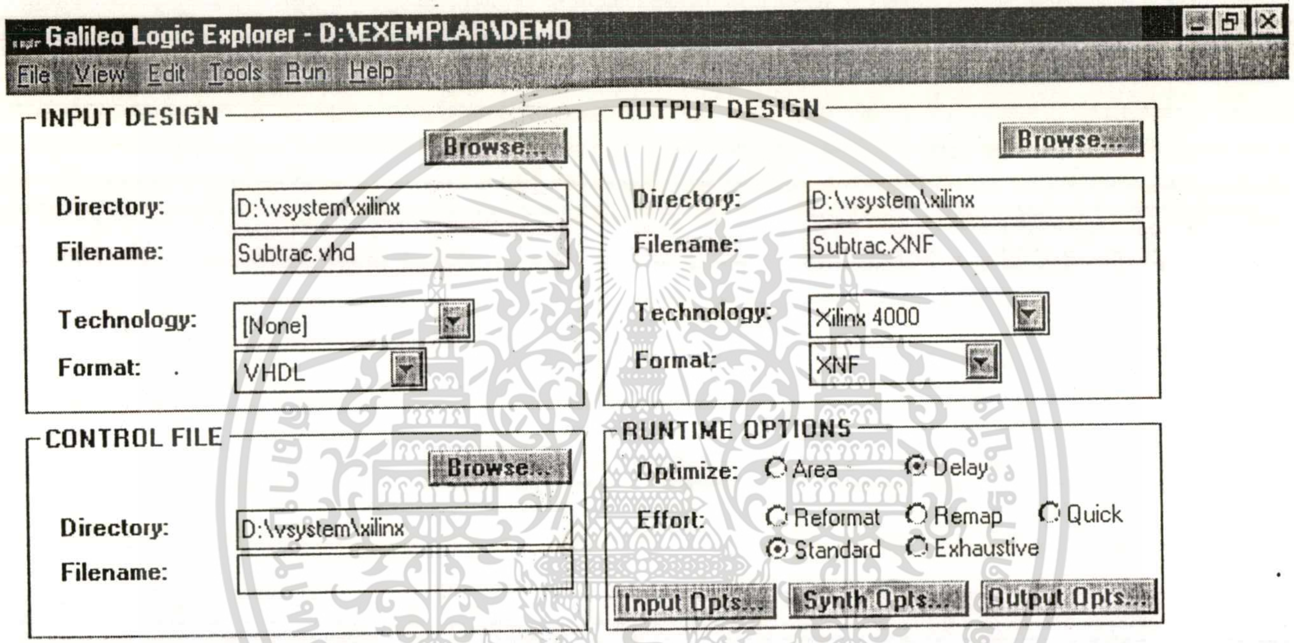
ในการสังเคราะห์วงจรนี้จะใช้โปรแกรมกาลิเลโอ (Logic Explorer) ให้ได้วงจรในระดับเกทแล้ว ก็จะได้ไฟล์ที่มีนามสกุลเป็น .XNF หลังจากนั้นจะนำไฟล์เหล่านี้มารวมกันตามที่ Top Entity ได้บรรยายเอาไว้ด้วยโปรแกรม XNF Merge หลังจากนั้นนำไฟล์ที่ได้ไปผ่านกระบวนการ Place and Route ผลลัพธ์ที่ได้จะนำกลับไปใช้โปรแกรมกาลิเลโอ (Time Explorer) เพื่อให้ได้ Source Code เป็นภาษา VHDL ที่มีคุณลักษณะตามเทคโนโลยีที่เราได้ออกแบบไว้กลับคืนมา ในขั้นตอนสุดท้ายจะนำไฟล์นี้ไปทำการ Simulation ซ้ำอีกครั้ง เพื่อตรวจสอบว่าได้ทำงานตามฟังก์ชันที่กำหนดไว้หรือไม่



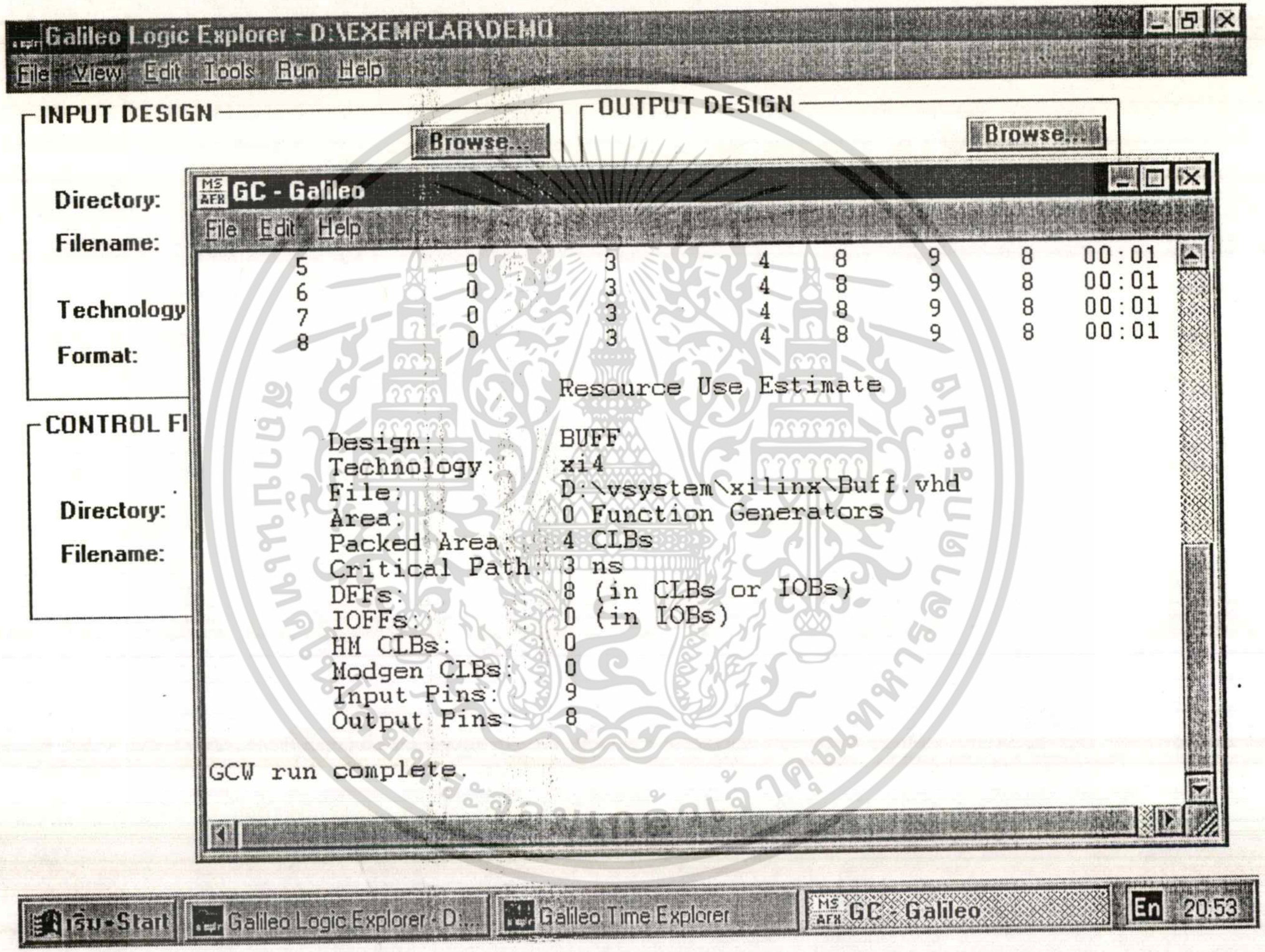
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 9.2.1 แสดงขั้นตอนการสังเคราะห์วงจร

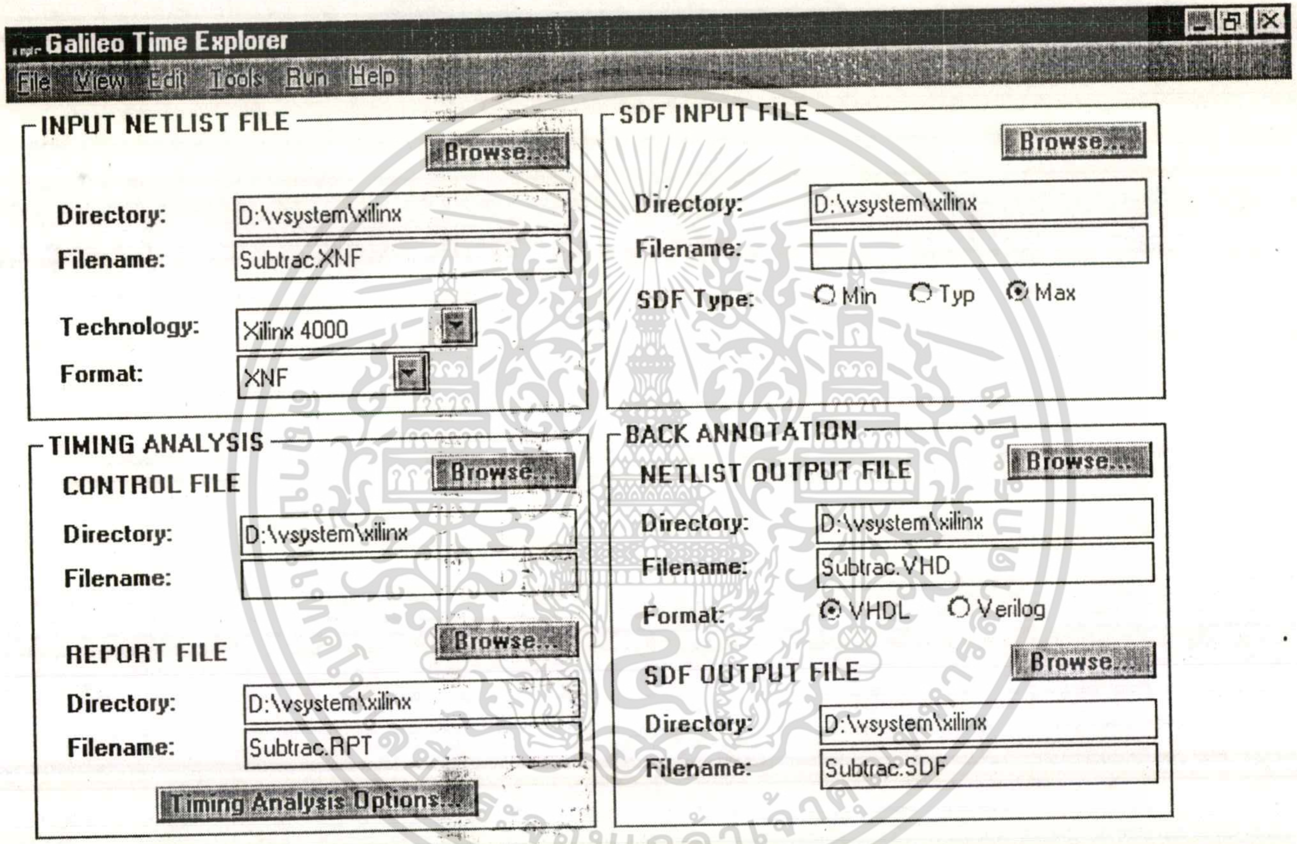
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



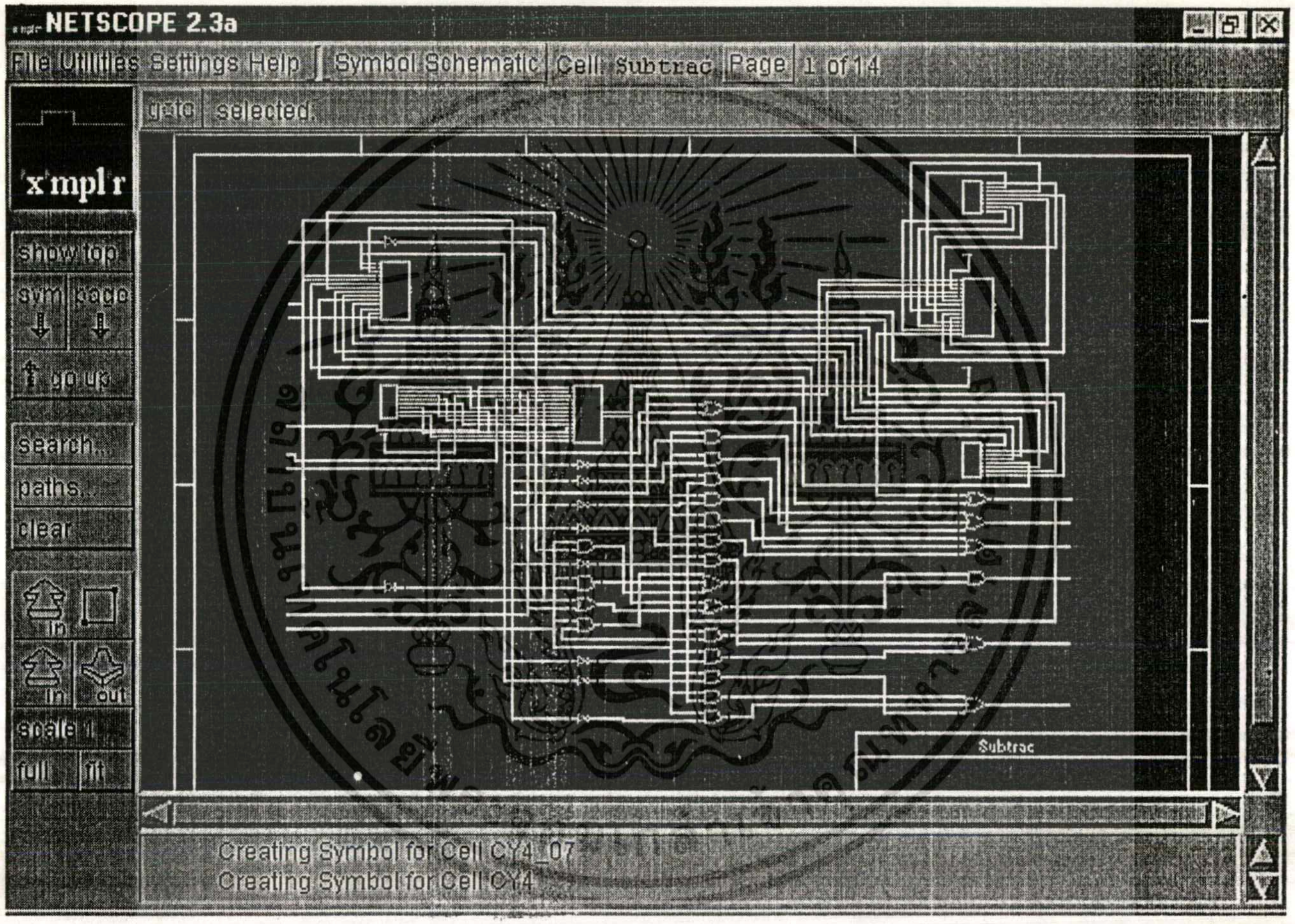
รูปที่ 9.2.2 แสดงเมนูของโปรแกรมกาลิเลโอ Logic Explorer



รูปที่ 9.2.3 แสดงรายงานผลการสังเคราะห์วงจรของโปรแกรมกาลิเลโอ Logic Explorer



รูปที่ 9.2.4 แสดงเมนูของโปรแกรมกาลิเลโอ Time Explorer



รูปที่ 9.2.5 แสดงรูปวงจรที่ได้จากการใช้โปรแกรมกาลิเลโอสังเคราะห์ขึ้นมา

บทที่ 10
สรุปและวิจารณ์

ในบทนี้จะสรุปการทำงานออกเป็นส่วนๆ ดังนี้คือ

10.1 ส่วน FUZZY LOGIC

จากการศึกษาในเทอมนี้ ทำให้สามารถสรุปถึงการออกแบบระบบโดยใช้ FUZZY LOGIC เข้ามาช่วยปรับปรุงคุณภาพได้ดังนี้คือ

10.1.1 หาต้นแบบระบบจากมนุษย์

10.1.2 สร้าง IF/THEN Rule เลียนแบบระบบตามข้อ 10.1.1

10.1.3 ทำ Fuzzification

10.1.4 ทำ Defuzzification

เนื่องจากการที่ออกแบบระบบ Fuzzy Logic ให้มีความใกล้เคียงการตัดสินใจของมนุษย์มากก็จะทำให้มี IF/THEN Rule มาก มีความซับซ้อนในการคำนวณสูง ซึ่งการเลือกว่าจะให้ระบบมีความซับซ้อนขนาดไหน จะขึ้นอยู่กับค่า Error ของระบบที่สามารถยอมรับได้ แล้วจึงออกแบบให้สอดคล้องกับสิ่งที่กำหนดขึ้นมา ก็จะได้ระบบ Fuzzy ที่เหมาะสมตามต้องการ

10.2 ส่วน PI CONTROL

เนื่องจากฟังก์ชันถ่ายโอนของระบบ PI อยู่ในโดเมน S แต่ในระบบควบคุมอุณหภูมินี้จะใช้การคำนวณแบบดิจิตอล จึงต้องหาฟังก์ชันถ่ายโอนของระบบ PI ในโดเมน Z ซึ่งจำเป็นต้องใช้การส่งจากโดเมน S ไปโดเมน Z ทำให้ค่าที่ส่งมาจากโดเมน S ไม่ครบ แต่ทว่าการส่งแบบที่ใช้ในระบบนี้ได้ใช้วิธีการแบบ Bilinear ซึ่งนับว่ามีข้อดีมากที่สุด เมื่อเปรียบเทียบกับวิธีอื่นๆ แต่ก็ยังเป็นวิธีที่มีข้อเสียอยู่บ้าง กล่าวคือ ที่ความถี่สูงๆ จะเกิดปรากฏการณ์ Wrapping

10.3 ส่วน VHDL

จากการศึกษาและทดลองทำการออกแบบระบบดิจิตอลด้วย VHDL นั้น พบว่า ขั้นตอนในการออกแบบมีดังนี้คือ

10.3.1 แนวความคิดในการออกแบบ

10.3.2 การออกแบบตัวบรรยายภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.3.3 การออกแบบทางวงจรตรรกะและการสังเคราะห์

10.3.4 การออกแบบทางกายภาพและการสร้างเครื่องต้นแบบ

10.3.5 การจำลองและการทดสอบ

ในแต่ละขั้นตอนจะให้ผลลัพธ์ที่เป็นข้อมูลให้กับขั้นต่อไป เพราะฉะนั้นการออกแบบไม่สามารถทำข้ามขั้นตอนได้ ในขั้นตอนการออกแบบตัวบรรยายภาษา VHDL นั้น เป็นขั้นตอนที่มีความยืดหยุ่นสูงพอสมควร มีรูปแบบต่างๆ กันในการออกแบบ ซึ่งสามารถแบ่งได้ดังนี้

10.3.2.1 รูปแบบ Top-Down Design รูปแบบนี้มีลักษณะคล้ายกับการออกแบบ Software โดยการแบ่งตัวระบบออกเป็นโมเดล ซึ่งจะทำการออกแบบระบบทั้งหมดก่อน โดยการมองโมเดลหนึ่งๆ เป็น Black Block หรือ Component ที่ไม่ทราบการทำงานภายใน แต่ทราบว่ามี I/O เป็นอย่างไร จากนั้นจึงทำการออกแบบส่วนของโมเดล

10.3.2.2 รูปแบบ Bottom-Up Design รูปแบบนี้จะกลับกันกับแบบ Top-Down Design เพราะจะออกแบบทีละโมเดลก่อน จากนั้นนำโมเดลทั้งหมดมาประกอบเข้าด้วยกัน ผลลัพธ์จากการออกแบบ จะเป็นรูปแบบ Flatten Design

10.3.2.3 รูปแบบ Flatten Design รูปแบบนี้จะไม่มีการแบ่งแยกการทำงานออกเป็นโมเดลจะมองระบบทั้งหมด แล้วทำการออกแบบ รูปแบบนี้เหมาะสำหรับการออกแบบระบบที่ไม่มีการทำงานซ้ำซ้อนกัน และเป็นระบบที่ไม่มีความซับซ้อนมากนัก

10.4 สรุปการทำงาน

10.4.1 ได้ทำการออกแบบขั้นตอน, วิธีการทางฟัซซี่ลอจิกให้เหมาะสมกับระบบ

10.4.2 ได้ทำการออกแบบโครงสร้างระบบควบคุมเชิงตัวเลขของระบบพีไอคอนโทรลเลอร์โดยใช้โครงสร้างที่เหมาะสม ให้สอดคล้องกับระบบที่เราต้องการ

10.4.3 ได้ทำการออกแบบระบบโดยรวมให้มีความเหมาะสมเพื่อที่จะให้ทำงานได้สอดคล้องกันตามที่ต้องการ

10.4.4 ได้เขียนโปรแกรมภาษาวีเอชดีแอล เพื่ออธิบายฟังก์ชันที่ได้กำหนดไว้จากระบบที่ต้องการข้างต้น

10.4.5 ได้ทำการสังเคราะห์วงจรตามขั้นตอนที่ได้กล่าวไว้แล้ว

10.4.6 ได้ทำการตรวจสอบวงจรที่สังเคราะห์ขึ้นมาตามขั้นตอนดังกล่าวข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.5 วิเคราะห์ผลงานและปัญหาในการทำงาน

10.5.1 ภาษาวีเอชดีแอลเป็นภาษาที่มี Solfware ที่สนับสนุนเป็นจำนวนมาก ทำให้ในการเริ่มต้นศึกษาภาษาวีเอชดีแอลเกิดปัญหาบ้าง เช่น ในช่วงแรกทางกลุ่มจะใช้โปรแกรม View Logic แต่พบว่า โปรแกรม V-System (ที่ใช้ในปริญญานิพนธ์ฉบับนี้) มีความสะดวกมากกว่าทั้งในเรื่องขนาดของโปรแกรม วิธีใช้โปรแกรม และการที่โปรแกรม V-System นี้สนับสนุนโปรแกรมกาลิเลโอในการสังเคราะห์วงจร ดังนั้นจึงตัดสินใจมาใช้โปรแกรม V-System

10.5.2 โปรแกรมที่ใช้ในปริญญานิพนธ์ฉบับนี้ ส่วนใหญ่เป็นโปรแกรมที่หาได้ยาก และจำเป็นต้องใช้ Hard Lock ทำให้ในการทำงานต้องเสียเวลาไปส่วนหนึ่งในการรอตัวโปรแกรม

10.5.3 การเขียนโปรแกรมภาษาวีเอชดีแอลนี้ จะมีลักษณะการเขียนได้หลายรูปแบบ แต่มิใช่ว่าทุกรูปแบบจะสามารถสังเคราะห์เป็นวงจรที่ต้องการได้ ดังนั้นการศึกษาโครงสร้างภาษานี้ให้แตกฉาน จึงเป็นสิ่งจำเป็นมาก ซึ่งผู้ที่จะศึกษาภาษาวีเอชดีแอลควรจะทำมิฉะนั้นจะไม่ได้ผลลัพธ์ตามที่ต้องการ

10.5.4 ในการสังเคราะห์วงจรมักใช้เวลาค่อนข้างนาน เช่น วงจรคูณในปริญญานิพนธ์ฉบับนี้ ใช้เวลาสังเคราะห์วงจรประมาณ 3 ชั่วโมง แต่ไม่ได้หมายความว่าผลลัพธ์ที่ได้จะเป็นผลลัพธ์ที่ถูกต้อง เมื่อต้องแก้ไขโปรแกรมส่วนที่เป็นภาษาวีเอชดีแอล แล้วเริ่มขั้นตอนต่างๆ ซ้ำอีกครั้งหนึ่ง จึงทำให้เสียเวลาเป็นอย่างมาก

10.5.5 โปรแกรม XNF Merge และ Place and Route มีเพียงชุดเดียวที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จึงทำให้การใช้โปรแกรมนี้เป็นไปอย่างล่าช้า

10.5.6 ขั้นตอนของการออกแบบระบบในปริญญานิพนธ์ ฉบับนี้ จะต้องมีการทำซ้ำๆ กันในแต่ละขั้นตอนต่างๆ จนกว่าจะได้รับผลลัพธ์ที่ถูกต้อง ซึ่งจากปัญหาและอุปสรรคที่ได้กล่าวไว้ข้างต้น จึงมีผลให้การทำงานโดยรวมต้องล่าช้าไปมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ไพรสิทธิ์ ศรีวัฒนานุกูลกิจ และวิบูลย์ ปิยวัฒนาเมธา, " การนำ FUZZY LOGIC มาประยุกต์ใช้งานในเครื่องทำน้ำอุ่น ", วิทยานิพนธ์ระดับปริญญาตรี, ภาควิชาอิเล็กทรอนิกส์, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2536

2. ศ.ดร.วัลลภ สุระกำพลธร, " การประมวลผลสัญญาณเชิงเลข ", บริษัทไดนาพรินท์จำกัด, พ.ศ. 2533

3. สุรเชษฐ์ ศรีพลกรัง, " การออกแบบวงจรรวมดิจิตอลโดยใช้วีเอชดีแอล ", วิทยานิพนธ์ระดับปริญญาตรี, ภาควิชาคอมพิวเตอร์, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2537

4. Andrew Brown, " VLSI Circuits and System in Silicon ", McGraw-Hill Book Company (UK), 1991.

5. Douglas L. Perry, San Ramon, " VHDL ", McGraw-Hill, 1991.

6. Jayaram Bhasker, " A VHDL Primer ", Prentice-Hall, 1992.

7. Mohammad Jamshidi, Nader Vadiee and Timothy J. Ross, " Fuzzy Logic and Control Software and Hardware Applications ", Prentice-Hall, 1993.

8. Motorola, "Optoelectronics Device Data ", fifth printing, 1989.

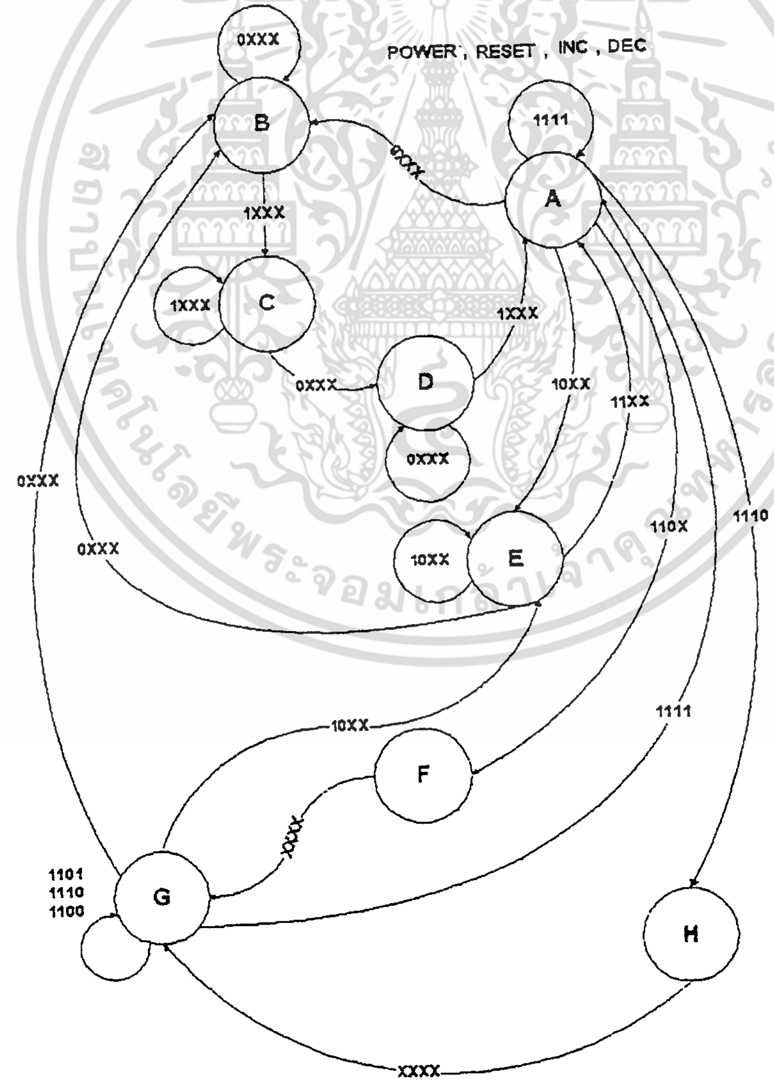
9. Steve Carlson, " Introduction to HDL-Base Design Using VHDL ", Synopsys Inc.,1991.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก
โปรแกรมภาษาวีเอชดีแอล

ในส่วนนี้จะแสดงโปรแกรม คำอธิบายของแต่ละโปรแกรม และโฟลว์ชาร์ตของแต่ละส่วนย่อยๆ ในระบบที่ได้ทำการออกแบบไว้ดังนี้

ผ.1 โปรแกรมส่วนสแกนคีย์บอร์ด



รูปที่ ผ.1 แสดง State Machine ของส่วนสแกนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถอธิบายรูปที่ ผ.1 ได้ดังนี้

สถานะ A เป็นสถานะที่รอรับการกดคีย์

สถานะ B เป็นสถานะที่ระบบถูกปิดและเช็คการปล่อยคีย์ POWER

สถานะ C เป็นสถานะที่ระบบถูกปิดและรอการกดคีย์ POWER

สถานะ D เป็นสถานะที่ระบบถูกเปิดและเช็คการปล่อยคีย์ POWER

สถานะ E เป็นสถานะที่ระบบถูกรีเซ็ตและรอการปล่อยคีย์ RESET

สถานะ F เป็นสถานะที่ระบบเพิ่มค่าอุณหภูมิ 1 องศาเซลเซียสแต่ต้องไม่เกิน 40 องศาเซลเซียส

สถานะ G เป็นสถานะที่ระบบเช็คการปล่อยคีย์ INC หรือ DEC

สถานะ H เป็นสถานะที่ระบบลดค่าอุณหภูมิ 1 องศาเซลเซียสแต่ต้องไม่ต่ำกว่า 20 องศาเซลเซียส

โปรแกรมส่วนสแกนคีย์บอร์ด

```
package  spac_rom  is
subtype  srom_word is bit_vector(7 downto 0);
type  srom_table is array(20 to 40) of srom_word;
constant  s_rom : srom_table := srom_table'(
        "00010100",   -- temp = 20
        "00011110",   -- temp = 21
        "00101000",   -- temp = 22
        "00110010",   -- temp = 23
        "00111100",   -- temp = 24
        "01000110",   -- temp = 25
        "01010000",   -- temp = 26
        "01011010",   -- temp = 27
        "01100100",   -- temp = 28
        "01101110",   -- temp = 29
        "01111000",   -- temp = 30
        "10000010",   -- temp = 31
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        "10001100",    -- temp = 32
        "10010110",    -- temp = 33
        "10100000",    -- temp = 34
        "10101010",    -- temp = 35
        "10110100",    -- temp = 36
        "10111110",    -- temp = 37
        "11001000",    -- temp = 38
        "11010010",    -- temp = 39
        "11011100");   -- temp = 40
end  spac_rom;
use  work.spac_rom.all;
entity  newsck  is
  port(clk,clk2,inc,dec,reset,power : in bit;
      o_power : out bit;
      s : out bit_vector(7 downto 0));
end  newsck;
architecture  newsck of newsck is
type  state_type is (a,b,c,d,e,f,g,h);
signal  current_state,next_state : state_type;
signal  count : integer range 20 to 40;
begin
first : process(clk,clk2)
  begin
    if (clk'event)and(clk = '1') then
      current_state <= next_state;
    elsif (clk2'event)and(clk2 = '0') then
      s <= s_rom(count);
    end if;
end process;
secound : process(current_state,power,reset,inc,dec)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
variable  count_var : integer range 20 to 40;
 begin
     case current_state is
       when a =>
      if (power = '0') then
        next_state <= b;
      elsif (power = '1')and(reset = '0') then
        next_state <= e;
      elsif (power = '1')and(reset = '1')and(inc = '0') then
        next_state <= f;
      elsif (power = '1')and(reset = '1')and(inc = '1')and(dec = '0') then
        next_state <= h;
      elsif (power = '1')and(reset = '1')and(inc = '1')and(dec = '1') then
        next_state <= a;
      end if;
     when b =>
       o_power <= '0';
      if (power = '0') then
        next_state <= b;
      elsif (power = '1') then
        next_state <= c;
      end if;
     when c =>
      if (power = '1') then
        next_state <= c;
      elsif (power = '0') then
        next_state <= d;
      end if;
     when d =>
       o_power <= '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

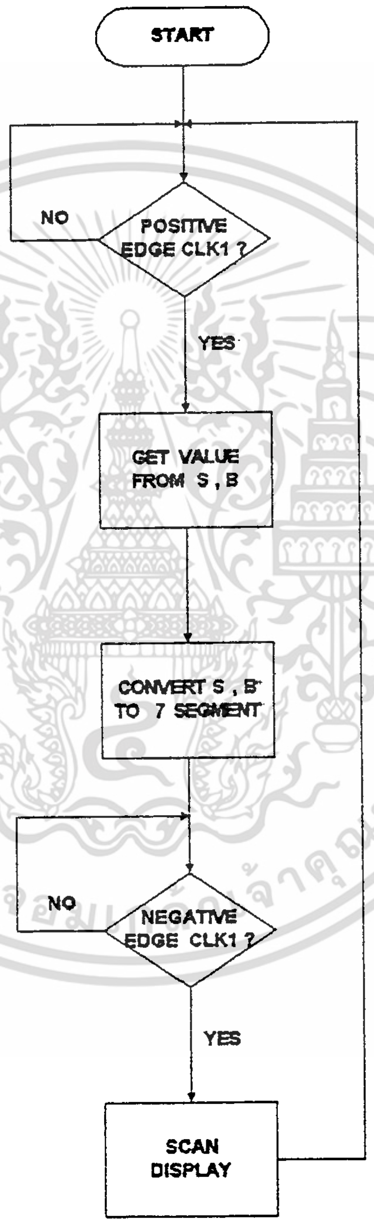
```
  count_var := 25;
  count <= 25;
 if (power = '0') then
   next_state <= d;
 elsif (power = '1') then
   next_state <= a;
 end if;
 when e =>
 count_var := 25;
 count <= 25;
 if (power = '0') then
   next_state <= b;
 elsif (power = '1')and(reset = '0') then
   next_state <= e;
 elsif (power = '1')and(reset ='1') then
   next_state <= a;
 end if;
when f =>
  if (count_var = 40) then
  else
    count_var := count_var + 1;
    count <= count_var;
  end if;
  next_state <= g;
when g =>
  if (power = '0') then
    next_state <= b;
  elsif (power = '1')and(reset = '0') then
    next_state <= e;
  elsif (power = '1')and(reset = '1') then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (inc = '1')and(dec = '1') then
              next_state <= a;
            else
              next_state <= g;
            end if;
          end if;
        when h =>
          if (count_var = 20) then
          else
            count_var := count_var - 1;
            count <= count_var;
          end if;
          next_state <= g;
      end case;
  end process;
  end newsck;
  configuration   scank_con of newsck    is
    for   newsck
    end for;
  end scank_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ.2 โปรแกรมส่วนสแกนดิสเพลย์



รูปที่ ผ.2 แสดงการทำงานของส่วนสแกนดิสเพลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนสแกนดิสเพลย์

```
entity scandi is
   port(clk1,power : in bit;
       s,b : in bit_vector(7 downto 0);
       sd : out bit_vector(7 downto 0);
       dc : out bit_vector(2 downto 0));
end scandi;
architecture scandi of scandi is
  type display is
         record c0 : bit_vector(7 downto 0);
                c1 : bit_vector(7 downto 0);
                c2 : bit_vector(7 downto 0);
         end record;
  type state_type is (cc0,cc1,cc2,cc3,cc4);
  signal current_state,next_state : state_type;
  signal s_show,b_show : display;
         function bcdto7(op : integer range 0 to 20)
         return bit_vector is
         variable sev_7segment : bit_vector(7 downto 0);
         begin
                case op is
                when 0 =>
                sev_7segment := ('0','0','1','1','1','1','1','1');
                when 1 =>
                sev_7segment := ('0','0','0','0','0','1','1','0');
                when 2 =>
                sev_7segment := ('0','1','0','1','1','0','1','1');
                when 3 =>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            sev_7segment := ('0','1','0','0','1','1','1','1');
            when   4 =>
            sev_7segment := ('0','1','1','0','0','1','1','0');
            when   5 =>
            sev_7segment := ('0','1','1','0','1','1','0','1');
            when   6 =>
            sev_7segment := ('0','1','1','1','1','1','0','1');
            when   7 =>
            sev_7segment := ('0','0','0','0','0','1','1','1');
            when   8 =>
            sev_7segment := ('0','1','1','1','1','1','1','1');
            when   9 =>
            sev_7segment := ('0','1','1','0','1','1','1','1');
            when   others =>
            sev_7segment := ('0','0','0','0','0','0','0','0');
            end case;
            return sev_7segment;
end         bcdto7;
function hexto7_3(sb : bit_vector)
return      display  is
variable sev_23segment : display;
variable buf : bit_vector(7 downto 0);
variable high,mid,low,carry,oh,om,ol : integer range 0 to 20;
begin
            ol := 0;
            om := 0;
            oh := 0;
            for i in 0 to 7 loop
              case i is
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    when  0 =>
          high := 0;mid := 0;low := 1;
    when  1 =>
          high := 0;mid := 0;low := 2;
    when  2 =>
          high := 0;mid := 0;low := 4;
    when  3 =>
          high := 0;mid := 0;low := 8;
    when  4 =>
          high := 0;mid := 1;low := 6;
    when  5 =>
          high := 0;mid := 3;low := 2;
    when  6 =>
          high := 0;mid := 6;low := 4;
    when  7 =>
          high := 1;mid := 2;low := 8;
    when  others =>
          high := 0;mid := 0;low := 0;
   end case;
if (sb(i) = '1') then
   ol := ol+low;
        if (ol >= 10) then
            ol := ol-10;
            carry := 1;
        else
            carry := 0;
        end if;
   om := om+mid+carry;
        if (om >= 10) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            om := om-10;
            carry := 1;
          else
            carry := 0;
          end if;
      oh := oh+high+carry;
    end if;
    end loop;
    om := om+8;
    if (om >= 10) then
      om := om-10;
      oh := oh+2;
    else
      oh := oh+1;
    end if;
    buf := bcdto7(oh);
    sev_23segment.c0 := buf;
    buf := bcdto7(om);
      sev_23segment.c1 := buf;
    buf := bcdto7(ol);
      sev_23segment.c2 := buf;
    return sev_23segment;
end     hexto7_3;
begin
        combine_show : process(clk1,power,current_state,s_show,b_show)
        begin
            if (clk1'event)and(clk1 = '1') then
                s_show <= hexto7_3(s);
                b_show <= hexto7_3(b);
```

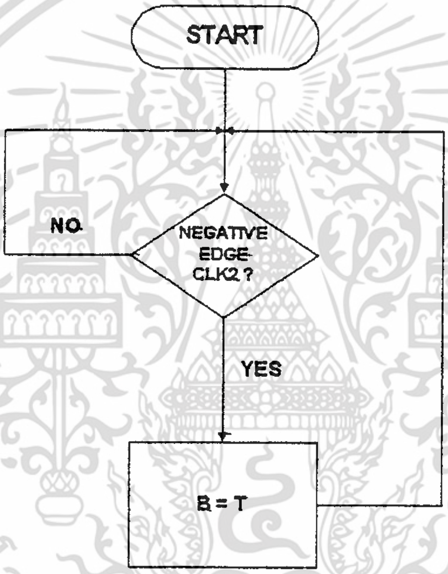
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
current_state <= next_state;
elsif  (clk1'event)and(clk1 = '0') then
if (power = '1') then
 case current_state is
  when cc0 =>
    sd <= b_show.c0;
    dc <= "000";
    next_state <= cc1;
  when cc1 =>
    sd <= b_show.c1;
    dc <= "001";
    next_state <= cc2;
  when cc2 =>
    sd <= b_show.c2;
    dc <= "010";
    next_state <= cc3;
  when cc3 =>
    sd <= s_show.c0;
    dc <= "011";
    next_state <= cc4;
  when cc4 =>
    sd <= s_show.c1;
    dc <= "100";
    next_state <= cc0;
  end case;
else  sd <= "00000000";
end if;
end if;
end process;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end scandi;
configuration   scand_con of scandi is
  for scandi
  end for;
end scand_con;
```

ผ.3 โปรแกรมส่วน BUFFER TEMPERATURE



รูปที่ ผ.3 แสดงการทำงานของส่วน BUFFER TEMPERATURE

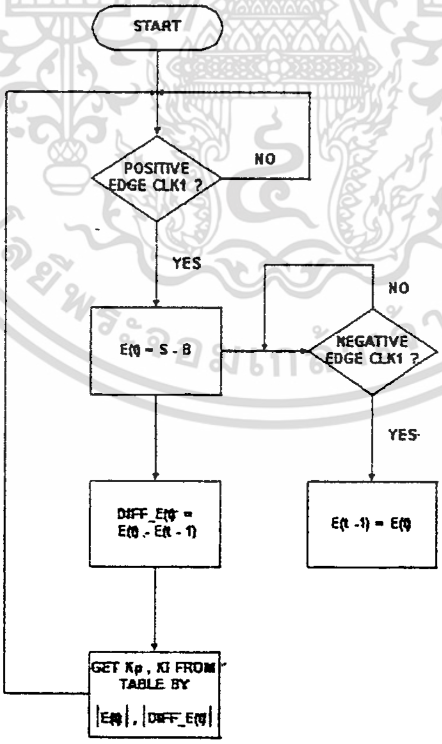
โปรแกรมส่วน BUFFER TEMPERATURE

```
entity   buff  is
   port(clk2 : in bit;
      t : in bit_vector(7 downto 0);
      b : out bit_vector(7 downto 0));
end  buff;
architecture  buff  of  buff  is
   begin
      process(clk2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
          if (clk2'event)and(clk2 = '0') then
           b <= t;
         end if;
      end  process;
end  buff;
configuration  buff_con of buff is
  for  buff
  end  for;
end  buff_con;
```

ผ.4 โปรแกรมส่วน FUZZY ALGORITHM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ ผ.4 แสดงการทำงานของส่วน FUZZY ALGORITHM

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนลบเลข

```
library ieee;
use ieee.exemplar.all;
package pac_sub is
   function add_sub(l,r : bit_vector; add : boolean)
   return bit_vector;
end    pac_sub;
package body pac_sub is
   function add_sub(l,r : bit_vector; add : boolean)
   return bit_vector is
      variable result : bit_vector(l'left+1 downto 0);
      variable trun_result : bit_vector(l'left downto 0);
   begin
         if add then  -- + operation
           result := add2(l,r);
           if (l(l'left) = '0')and(r(r'left) = '0') then  -- check + overflow
             if (result(l'left) = '1') then
               for i in 0 to (l'left-1) loop  -- + overflow
                 trun_result(i) := '1';
               end loop;
               trun_result(l'left) := '0';
             else
               trun_result := extend2(result,l'length);
             end if;
           elsif (l(l'left) = '1')and(r(r'left) = '1') then  -- check - overflow
             if (result(l'left) = '0') then
               for i in 0 to (l'left-1) loop -- - overflow
               trun_result(i) := '0';
               end loop;
               trun_result(l'left) := '1';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
          trun_result := extend2(result,l'length);
        end if;
      else
        trun_result := extend2(result,l'length);
      end if;
  else
    result := sub2(l,r);                    -- - operation
    if  (l(l'left) = '0')and(r(r'left) = '1') then      -- check + overflow
      if (result(l'left) = '1') then
        for i in 0 to (l'left-1) loop  -- + overflow
          trun_result(i) := '1';
        end loop;
        trun_result(l'left) := '0';
      else
        trun_result := extend2(result,l'length);
      end if;
    elsif (l(l'left) = '1')and(r(r'left) = '0') then          -- check - overflow
      if (result(l'left) = '0') then
        for i in 0 to (l'left-1) loop  -- - overflow
          trun_result(i) := '0';
        end loop;
        trun_result(l'left) := '1';
      else
        trun_result := extend2(result,l'length);
      end if;
    else
      trun_result := extend2(result,l'length);
    end if;
  end if;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return trun_result;
          end  add_sub;
end pac_sub;
use      work.pac_sub.all;
entity   subtrac  is
  port(ad_su : in boolean;
     s,b : in bit_vector(16 downto 0);
     result : out bit_vector(16 downto 0));
end      subtrac;
architecture  subtrac  of  subtrac  is
begin
       result <= add_sub(s,b,ad_su);
end  subtrac;
configuration  sub_con of subtrac is
  for subtrac
  end for;
end sub_con;
```

โปรแกรมส่วน SHIFT REGISTER

```
entity  z_delay  is
port(clk1 : in bit;
     data : in bit_vector(16 downto 0);
     data_1 : out bit_vector(16 downto 0));
end z_delay;
architecture  z_delay of z_delay is
begin
process(clk1)
begin
      if (clk1'event)and(clk1 = '0') then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      data_1 <= data;
    end if;
end process;
end z_delay;
configuration z_con of z_delay is
  for z_delay
  end for;
end z_con;
```

โปรแกรมส่วน ABSOLUTE VALUE

```
library  ieee;
use  ieee.exemplar.all;
entity  abs_t  is
    port(non_abs : in bit_vector(16 downto 0);
         abs_v : out bit_vector(16 downto 0));
end  abs_t;
architecture  abs_t of abs_t is
    begin
      process(non_abs)
        begin
          if (non_abs(16) = '1') then
             abs_v <= comp2(non_abs);
          else
             abs_v <= non_abs;
          end if;
      end process;
  end abs_t;
  configuration  abs_con of abs_t is
    for abs_t
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  end for;
end abs_con;
```

โปรแกรมส่วนที่เก็บตาราง Kp, Ki

```
package kp_ki   is
subtype int255 is integer range 0 to 255;
type    two_dimkp is array (0 to 20,0 to 10) of integer range 0 to 24;
type    two_dimki  is array (0 to 20,0 to 10) of integer range 0 to 12;
        constant  table_kp : two_dimkp :=
                ((20,20,20,20,20,20,16,16,16,16,16),
                 (20,20,20,20,20,20,16,16,16,16,16),
                 (20,20,20,20,20,20,16,16,16,16,16),
                 (20,20,20,20,20,20,16,16,16,16,16),
                 (20,20,20,20,20,20,16,16,16,16,16),
                 (20,20,20,20,20,20,16,16,16,16,16),
                 (16,16,16,16,16,16,16,16,16,16,20),
                 (16,16,16,16,16,16,16,16,16,16,20),
                 (16,16,16,16,16,16,16,16,16,16,20),
                 (16,16,16,16,16,16,16,16,16,16,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (16,20,20,20,20,20,20,20,20,20,20),
                 (20,20,24,24,24,24,24,24,24,24,24),
                 (20,20,24,24,24,24,24,24,24,24,24),
                 (20,20,24,24,24,24,24,24,24,24,24),
                 (20,20,24,24,24,24,24,24,24,24,24),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        (24,24,24,24,24,24,24,24,24,24,24));
    constant  table_ki : two_dimki :=
        ((12,12,12,12,12,12,10,10,10,10,10),
        (12,12,12,12,12,12,10,10,10,10,10),
        (12,12,12,12,12,12,10,10,10,10,10),
        (12,12,12,12,12,12,10,10,10,10,10),
        (12,12,12,12,12,12,10,10,10,10,10),
        (12,12,12,12,12,12,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,10,10,10,10,10,10,10,10,10),
        (10,10,8,8,8,8,8,8,8,8,8),
        (10,10,8,8,8,8,8,8,8,8,8),
        (10,10,8,8,8,8,8,8,8,8,8),
        (10,10,8,8,8,8,8,8,8,8,8),
        (8,8,8,8,8,8,8,8,8,8,8));
function send_kp(table_kpp : integer range 0 to 24)
            return  bit_vector;
function send_ki(table_kii : integer range 0 to 12)
            return  bit_vector;
function  hex_decimal(sev_bit : bit_vector)
            return  int255;

end kp_ki;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
package body        kp_ki is
function send_kp(table_kpp : integer range 0 to 24)
        return    bit_vector is
begin
     case table_kpp is
          when 16 =>
          return ('1','0','0','0','0');
          when 20 =>
          return ('1','0','1','0','0');
          when 24 =>
          return ('1','1','0','0','0');
          when others =>
          return ('0','0','0','0','0');
     end case;
end     send_kp;
function send_ki(table_kii : integer range 0 to 12)
return bit_vector is
begin
     case table_kii is
          when 8 =>
          return ('1','0','0','0');
          when 10 =>
          return ('1','0','1','0');
          when 12 =>
          return ('1','1','0','0');
          when others =>
          return ('0','0','0','0');
     end case;
end     send_ki;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function hex_decimal(sev_bit : bit_vector)
return   int255 is
variable result,temp : int255;
begin
        result := 0;
     for i in 0 to 7 loop
       case i is
          when  0  =>
          temp := 1;
          when  1  =>
          temp := 2;
          when  2  =>
          temp := 4;
          when  3  =>
          temp := 8;
          when  4  =>
          temp := 16;
          when  5  =>
          temp := 32;
          when  6  =>
          temp := 64;
          when  7  =>
          temp := 128;
       end  case;
       if (sev_bit(i) = '1') then
          result := result + temp;
       end if;
     end loop;
  return  result;
  end  hex_decimal;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end    kp_ki;
use  work.kp_ki.all;
entity   get_kpki is
port(abs_errort,abs_dif : in bit_vector(16 downto 0);
      kp : out bit_vector(4 downto 0);
      ki : out bit_vector(3 downto 0));
end get_kpki;
architecture   get_kpki of get_kpki is
begin
process(abs_errort,abs_dif)
variable one,two : int255;
begin
   one := hex_decimal(abs_errort);
   two := hex_decimal(abs_dif);
   if (one > 20) then
     one := 20;
   end if;
   if (two > 10) then
     two := 10;
   end if;
   kp <= send_kp(table_kp(one,two));
   ki <= send_ki(table_ki(one,two));
 end process;
 end get_kpki;
 configuration getkpki_con of get_kpki is
   for get_kpki
   end for;
 end getkpki_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วน FUZZY ALGORITHM

```
library  ieee;
use      ieee.exemplar.all;
entity  fuzzy  is
   port(clk1 : in bit;
      s,b : in bit_vector(7 downto 0);
      kp : out bit_vector(4 downto 0);
      ki  : out bit_vector(3 downto 0);
      errort : out bit_vector(16 downto 0));
end  fuzzy;
architecture  fuzzy  of  fuzzy  is
  component  z_delay
    port(clk1 : in bit;
      data : in bit_vector(16 downto 0);
      data_1 : out bit_vector(16 downto 0));
  end component;
  component  subtrac
     port(ad_su : in boolean;
        s,b : in bit_vector(16 downto 0);
        result : out bit_vector(16 downto 0));
  end component;
  component  abs_t
     port(non_abs : in bit_vector(16 downto 0);
        abs_v : out bit_vector(16 downto 0));
  end  component;
  component  get_kpki
     port(abs_errort,abs_dif : in bit_vector(16 downto 0);
        kp : out bit_vector(4 downto 0);
        ki : out bit_vector(3 downto 0));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

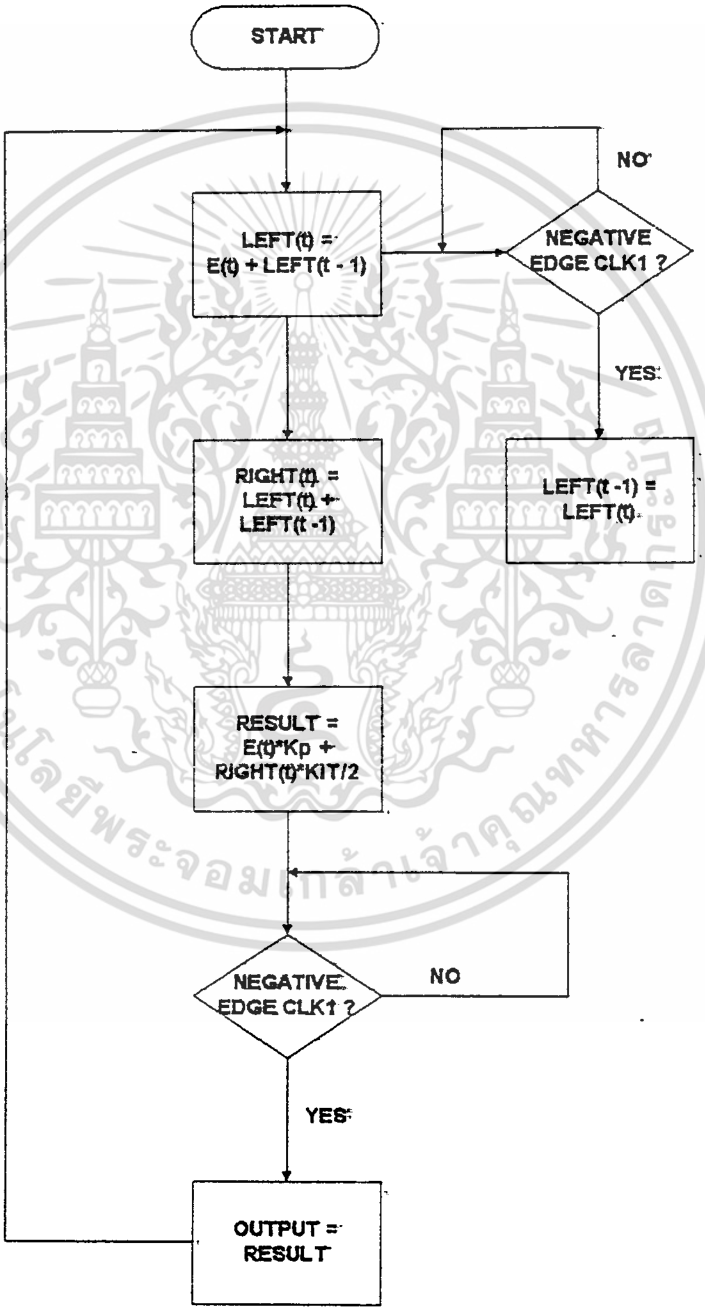
```
   end component;
 signal  ad_suu : boolean;
 signal  errortt,errort_11,ss,bb,abs_errortt,abs_diff,
          diff_err : bit_vector(16 downto 0);
 begin
        errort <= errortt;
      ad_suu <= false;
       sub_1   :  subtrac
 port map (ad_su => ad_suu,
             s  => ss,
             b => bb,
       result  => errortt);
     z_1 : z_delay
 port map (clk1 => clk1,
          data  => errortt,
       data_1 => errort_11);
 sub_2 : subtrac
 port map (ad_su => ad_suu,
          s => errortt,
          b => errort_11,
          result => diff_err);
 abs_1 : abs_t
 port map (non_abs => errortt,
              abs_v => abs_errortt);
 abs_2 : abs_t
 port map (non_abs => diff_err,
          abs_v => abs_diff);
 get_kpki1 : get_kpki
 port map (abs_errort => abs_errortt,
          abs_dif => abs_diff,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
kp => kp,
ki => ki);
sync_input : process(clk1)
begin
if (clk1'event)and(clk1 = '1') then
ss <= extend(s,17);
bb <= extend(b,17);
end if;
end process;
end fuzzy;
configuration fuzz_con of fuzzy is
for fuzzy
for sub_1 : subtrac use configuration work.sub_con;
end for;
for z_1 : z_delay use configuration work.z_con;
end for;
for sub_2 : subtrac use configuration work.sub_con;
end for;
for abs_1 : abs_t use configuration work.abs_con;
end for;
for abs_2 : abs_t use configuration work.abs_con;
end for;
for get_kpki1 : get_kpki use configuration work.getkpki_con;
end for;
end for;
end fuzz_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ.5 โปรแกรมส่วนคำนวณแบบพีไอ



รูปที่ ผ.5 แสดงการทำงานของการคำนวณแบบพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนวงจรคูณ

```
library  ieee;
use  ieee.exemplar.all;
package  multiplier  is
      function  mul_sign(e,f : bit_vector) return bit_vector;
end  multiplier;
package  body  multiplier  is
      function  mul_sign(e,f : bit_vector) return bit_vector is
      constant  max_pos : bit_vector(16 downto 0) :=
              ("01111111111111111");
      constant  min_neg : bit_vector(16 downto 0) :=
              ("10000000000000000");
      variable  mul_result : bit_vector(16 downto 0);
      variable  multiply : bit_vector(33 downto 0);
      variable   cut : bit_vector(16 downto 0);
begin
    multiply := mult2(e,f);
    if ((e(16) = '0')and(f(16) = '0'))or((e(16) = '1')and(f(16) = '1')) then
     for i in 16 to 33 loop
      if (multiply(i) = '1') then
        mul_result := max_pos;
        exit;
      else
        mul_result := extend2(multiply,17);
      end if;
     end loop;
    else
     for i in 16 to 33 loop
      if (multiply(i) = '0') then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          mul_result := min_neg;
          exit;
        else
          mul_result := extend2(multiply,17);
        end if;
        end loop;
      end if;
      return mul_result;
end mul_sign;
end  multiplier;
use   work.multiplier.all;
entity  mul_mul  is
   port(male,female : in bit_vector(16 downto 0);
        son : out bit_vector(16 downto 0));
end  mul_mul;
architecture  mul_mul  of  mul_mul  is
begin
   son <= mul_sign(male,female);
end mul_mul;
configuration  mul_con of mul_mul is
   for mul_mul
   end for;
end mul_con;
```

ส่วนพีไอคอนโทรลเลอร์

```
entity  pi  is
   port(clk1,power : in bit;
        kp : in bit_vector(4 downto 0);
        ki : in bit_vector(3 downto 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      errort : in bit_vector(16 downto 0);
      output : out bit_vector(16 downto 0));
end pi;
architecture pi of pi is
component kpki
  port(kp : in bit_vector(4 downto 0);
      ki : in bit_vector(3 downto 0);
      kp_16,ki_16 : out bit_vector(16 downto 0));
end component;
component z_delay
  port(clk1 : in bit;
      data : in bit_vector(16 downto 0);
      data_1 : out bit_vector(16 downto 0));
end component;
 component subtrac
    port(ad_su : in boolean;
        s,b : in bit_vector(16 downto 0);
        result : out bit_vector(16 downto 0));
 end component;
component mul_mul
  port(male,female : in bit_vector(16 downto 0);
      son : out bit_vector(16 downto 0));
end component;
signal ad_suu : boolean;
signal leftt,left_11,kp_166,ki_166,
        sonn1,sonn2,resultt2,resultt3 : bit_vector(16 downto 0);
begin
        ad_suu <= true;
        add_1 : subtrac
  port map (ad_su => ad_suu,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

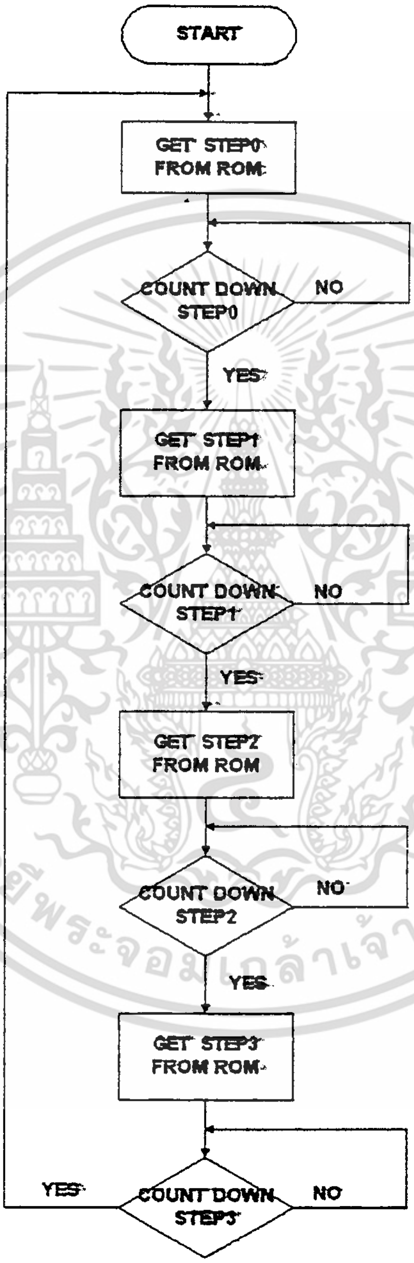
```
        s => errort,
        b => left_11,
        result => leftt);
        z_1 : z_delay
port map (clk1 => clk1,
        data => leftt,
        data_1 => left_11);
        add_2 : subtrac
port map (ad_su => ad_suu,
        s => leftt,
        b => left_11,
        result => resultt2);
        kpki_1 : kpki
port map (kp => kp,
        ki => ki,
        kp_16 => kp_166,
        ki_16 => ki_166);
        mul_1 : mul_mul
port map (male => errort,
        female => kp_166,
        son => sonn1);
        mul_2 : mul_mul
port map (male => ki_166,
        female => resultt2,
        son => sonn2);
        add_3 : subtrac
port map (ad_su => ad_suu,
        s => sonn1,
        b => sonn2,
        result => resultt3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sync_out : process(clk1,power)
  begin
    if (power = '0') then
      output <= "0000000000000000";
    elsif (clk1'event)and(clk1 = '0') then
      output <= resultt3;
    end if;
  end process;
end pi;
configuration pinew_con of pi is
 for pi
  for add_1 : subtrac use configuration work.sub_con;
  end for;
  for z_1 : z_delay use configuration work.z_con;
  end for;
  for add_2 : subtrac use configuration work.sub_con;
  end for;
  for kpki_1 : kpki use configuration work.kpki_con;
  end for;
  for mul_1 : mul_mul use configuration work.mul_con;
  end for;
  for mul_2 : mul_mul use configuration work.mul_con;
  end for;
  for add_3 : subtrac use configuration work.sub_con;
  end for;
 end for;
end pinew_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ.6 ส่วนสร้างนาฬิกา 2 เฟส



รูปที่ ผ.6 แสดงการทำงานของส่วนสร้างสัญญาณนาฬิกา 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนสร้างสัญญาณนาฬิกา 2 เฟส

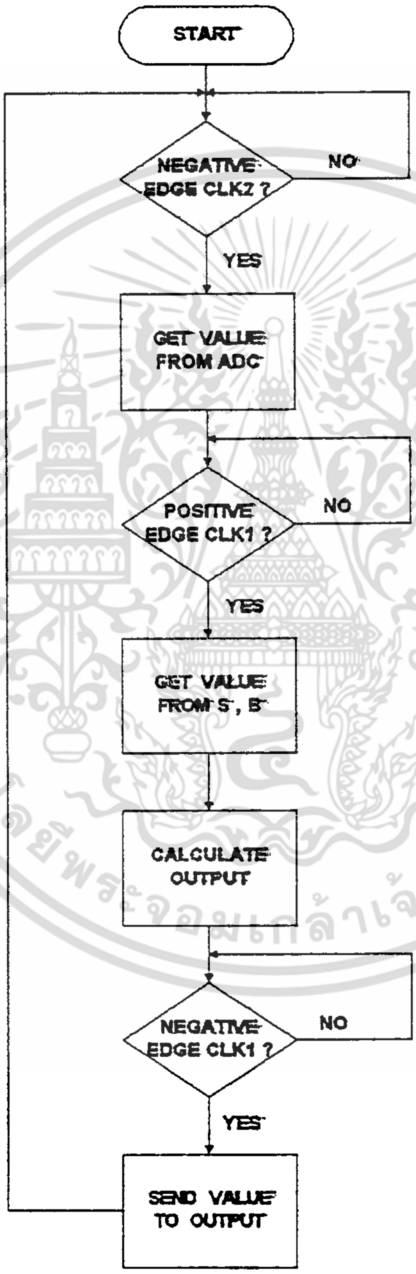
```
package roms is
  subtype rom1_range is integer range 0 to 3;
  type rom1_table is array(0 to 3) of bit_vector(1 downto 0);
  constant rom1 : rom1_table := rom1_table'(
    "10",        -- step 0
    "00",        -- step 1
    "01",        -- step 2
    "00");       -- step 3
  subtype rom2_word is integer range 0 to 5;
  type rom2_table is array(0 to 3) of rom2_word;
  constant rom2 : rom2_table :=
        rom2_table'(1,0,1,4);
  end roms;
use work.roms.all;
entity wave is
  port(reset,clkk : in bit;
       clk : out bit_vector(1 downto 0));
end wave;
architecture wave of wave is
  signal step,next_step : rom1_range;
  signal delay : rom2_word;
begin
  next_step <= rom1_range'low when
         step = rom1_range'high else
              step + 1;
timstep_counter : process
begin
  wait until (clkk'event)and(clkk = '1');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if (reset = '0') then
    step <= 0;
  else
  if (delay = 0) then
    step <= next_step;
  else
    end if;
  end if;
end process;
delay_counter : process
begin
  wait until (clkk'event)and(clkk = '1');
  if (reset = '0') then
    delay <= 0;
  else
  if (delay = 0) then
    delay <=rom2(next_step);
  else
    delay <= delay - 1;
  end if;
  end if;
end process;
clk <= rom1(step);
end  wave;
configuration   wave_con of wave is
  for wave
  end for;
end wave_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ.7 โปรแกรมโดยรวมทั้งหมด



รูปที่ ผ.7 แสดงการทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมรวมทั้งหมด

```
entity  tmp_con is
  port(inc,dec,reset,power,clkk : in bit;
       ctrl : out bit;
       dc : out bit_vector(2 downto 0);
        t : in bit_vector(7 downto 0);
       sd : out bit_vector(7 downto 0);
   output : out bit_vector(16 downto 0));
end tmp_con;

architecture tmp_con of tmp_con is
component  newsck
  port(clk,clk2,inc,dec,reset,power : in bit;
       o_power : out bit;
        s : out bit_vector(7 downto 0));
end  component;
component  wave
  port(reset,clkk : in bit;
       clk : out bit_vector(1 downto 0));
end  component;
component  buff
  port(clk2 : in bit;
        t : in bit_vector(7 downto 0);
       b : out bit_vector(7 downto 0));
end  component;
component  scandi
  port(clk1,power : in bit;
        s,b : in bit_vector(7 downto 0);
        sd : out bit_vector(7 downto 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        dc : out bit_vector(2 downto 0));
end  component;
component  fuzzy
  port(clk1 : in bit;
        s,b : in bit_vector(7 downto 0);
        kp : out bit_vector(4 downto 0);
        ki : out bit_vector(3 downto 0);
        errort : out bit_vector(16 downto 0));
end  component;
component  pi
  port(clk1,power : in bit;
        kp : in bit_vector(4 downto 0);
        ki : in bit_vector(3 downto 0);
        errort : in bit_vector(16 downto 0);
        output : out bit_vector(16 downto 0));
end component;

signal o_powerr,clk1_1,clk2_2 : bit;
signal kp_p : bit_vector(4 downto 0);
signal ki_i : bit_vector(3 downto 0);
signal s_s,b_b : bit_vector(7 downto 0);
signal errort_t : bit_vector(16 downto 0);

begin
  ctrl <= clk2_2;
   u1 : newsck
     port map (clk => clkk,
               clk2 => clk2_2,
                inc => inc,
               dec => dec,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                reset => reset,
              power => power,
          o_power => o_powerr;
                      s => s_s);
u2 : wave
   port map (reset => reset,
                clkk => clkk,
           clk(0) => clk1_1,
           clk(1) => clk2_2);
u3 : scandi
   port map (clk1 => clk1_1,
                power => o_powerr,
                      s  => s_s,
                      b => b_b,
                    sd => sd,
                    dc => dc);
u4 : fuzzy
   port map (clk1 => clk1_1,
                      s => s_s,
                      b => b_b,
                    kp => kp_p,
                    ki => ki_i,
             errort => errort_t);
u5 : buff
   port map (clk2 => clk2_2,
                      t => t,
                      b => b_b);
u6 : pi
   port map (clk1 => clk1_1,
                power => o_powerr,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                kp => kp_p,
                ki => ki_i,
                errort => errort_t,
                output => output);
end tmp_con;

configuration tmpcon_con of tmp_con is
  for tmp_con
    for u1 : newsck use configuration work.scank_con;
    end for;
    for u2 : wave use configuration work.wave_con;
    end for;
    for u3 : scandi use configuration work.scand_con;
    end for;
    for u4 : fuzzy use configuration work.fuzz_con;
    end for;
    for u5 : buff use configuration work.buff_con;
    end for;
    for u6 : pi use configuration work.pinew_con;
    end for;
  end for;
end tmpcon_con;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

แด่พ่อและแม่......................ผู้ให้กำเนิด

แด่ครูบาอาจารย์................ผู้ให้ความรู้

แด่มิตรแท้..........................ผู้ให้ความช่วยเหลือ

แด่ทุกคน............................ผู้ให้กำลังใจ

แด่ตัวเอง............................ที่ทำสำเร็จ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้