สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเข้าและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่นโค้ด

Digital Data Encoding and Decoding Using Convolutional Codes

T050101

โดย

นาย กุลพงษ์ ประสงค์ดี

นาย ธวัชชัย จันทร์ถาวร

นาย ฤกษ์ชัย พิพัฒน์นาวิน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 50101
วัน,เดือน,ปี 21 มี.ย. 2547

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

ข้อความในเอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่นโค้ด

Digital Data Encoding and Decoding Using Convolutional Codes

โดย

นาย กุลพงษ์ ประสงค์ดี 43015002

นาย ธวัชชัย จันทร์ถาวร 43015015

นาย ฤกษ์ชัย พิพัฒน์นาวิน 43015030

อาจารย์ที่ปรึกษา

อาจารย์ อัครพล ตรีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การเข้าและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่นโค้ด**

Digital Data Encoding and Decoding sing Convolutional Codes

ผู้จัดทำ

1. นายกุลพงษ์ ประสงค์ดี 43015002

2. นายธวัชชัย จันทร์ถาวร 43015015

3. นายฤกษ์ชัย พิพัฒน์นาวิน 43015030

..... อาจารย์ที่ปรึกษา

(อาจารย์ อัครพล ตรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่นโค้ด

Digital Data Encoding and Decoding

Using Convolutional Codes

โดย นายกุลพงษ์ ประสงค์ดี 43015002

นายธวัชชัย จันทร์ถาวร 43015015

นายฤกษ์ชัย พิพัฒน์นาวิน 43015030

อาจารย์ที่ปรึกษา อาจารย์ อัครพล ตรีรัตน์

บทคัดย่อ

โครงงานนี้นำเสนอการทำงานของการเข้ารหัสและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่นโค้ด (Convolutional Codes) ซึ่งใช้ในระบบการสื่อสารข้อมูล โดยการเข้ารหัสดังกล่าวมีข้อดีคือ ช่วยลดความผิดพลาดในการรับ-ส่งข้อมูลในช่องสัญญาณสื่อสารซึ่งมีสัญญาณรบกวนให้น้อยลงได้ ในโครงงานนี้จะใช้การเขียนคำสั่งฮาร์ดแวร์ด้วยภาษาวีเอชดีแอล (VHDL : Very High Speed Integrated Circuit Hardware Description Language) เพื่ออธิบายวงจรทั้งหมดของการเข้ารหัสและถอดรหัส และทำการสร้างให้อยู่ภายในชิพอุปกรณ์เอฟพีจีเอ (FPGA : Field Programmable Gate Array)

Abstract

This project proposes the study and the design of encoding and decoding of digital data stream by using convolutional codes. The proposes of this encoding scheme is to reduce the Bit Error Rate (BER) and increase the transmission efficiency. This project, FPGA (Field Programmable Gate Array) will be used to design and implement of a coder and a decoder circuit by using VHDL language (Very High Speed Integrated Circuit Hardware Description Language) .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	2
1.3 ขอบเขตของปริญญานิพนธ์	2
1.4 เนื้อหาของปริญญานิพนธ์	3
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 พื้นฐานการเข้ารหัส	4
2.1.1 รหัสลิเนียร์บล็อก	6
2.1.2 รหัสวงเวียน	9
2.1.3 คอนโวลูชั่นโค้ด	11
2.2 ภาษาวีเอชดีแอล	24
2.2.1 การออกแบบระบบดิจิตอล	24
2.2.2 ประวัติความเป็นมาของภาษาวีเอชดีแอล	25
2.2.3 องค์ประกอบพื้นฐานต่างๆ ของภาษาวีเอชดีแอล	26
2.2.4 การบรรยายเชิงพฤติกรรม	31
2.2.5 การโปรเซส	31
2.2.6 การกำหนดตัวดำเนินการภายในโปรเซส	31
2.2.7 การกำหนดการกระทำภายในโปรเซส	32
2.2.8 การกระตุ้นและยับยั้งการกระทำของโปรเซส	32
2.2.9 การออกแบบจากบนลงล่าง (Top-Down Design)	32
2.3 เอฟพีจีเอ	34
2.4 การออกแบบวงจรเชิงเลขด้วยชิพอุปกรณ์เอฟพีจีเอ	35
2.5 สถาปัตยกรรมภายในของชิพอุปกรณ์เอฟพีจีเอ	36
2.5.4 แอลอี (LE : Logic Element)	36
2.5.5 แอลเอบี (LAB : Logic Array Block)	39
2.5.6 อีเอบี (EAB : Embedded Array Block)	40
2.5.7 ไอโออี (IOE : Input Output Element)	41
2.6 ปัจจัยที่ทำให้การออกแบบชิพอุปกรณ์เอฟพีจีเอ ทำได้ง่ายและสะดวกรวดเร็ว	42
บทที่ 3 การคำนวณและการสร้าง	43
3.1 ส่วนของการเข้ารหัส (Encoder Part)	44
3.1.1 ส่วนของการส่งค่าอินพุตจากคอมพิวเตอร์	44
3.1.2 ส่วนของวงจรบัพเฟอร์	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ส่วนของวงจรลดค่าความถี่ 45
3.1.4 ส่วนของการเข้ารหัส 45
3.2 ส่วนของการถอดรหัส (Decoder Part) 47
3.2.1 วงจรไวตาบิดิสแตนซ์ (Viterbi_distance) 48
3.2.2 วงจรคอมพิวเมตริกซ์ (Compute_metric) 48
3.2.3 วงจรคอมแพร์เมตริกซ์ (Compare_metric) 49
3.2.4 วงจรซีเลกเมตริกซ์ (Select_metric) 50
3.2.5 วงจรคอมแพร์ (Compare) 51
3.2.6 วงจรไฟน์สมอลเลท (find_smallest) 51
3.2.7 วงจรคอนโทรล (Control) 51
3.2.8 วงจรรีดิวซ์ (Reduce) 52
3.2.9 วงจรเมตริกซ์ (Metric) 52
3.2.10วงจรเทรดแบคเอซีเอส (Traceback_acs) 53
3.2.11วงจรเทรดแบคคอนโทรล (Tracback_control) 53
3.2.12วงจรถอดรหัส (Decoder) 55
3.3 ส่วนของการวัดค่าความผิดพลาดของข้อมูล (Bit Error Rate) 55
3.3.1 วงจรอีเนเบิล (Enable) 55
3.3.2 วงจรสุ่มค่า (Random_bit) 56
3.3.3 วงจรจำลองการผิดพลาดของข้อมูล (Noise_add) 56
3.3.4 วงจรหน่วงเวลา (Delay_time) 57
3.3.5 วงจรเปรียบเทียบสัญญาณ (Comparator) 57
บทที่ 4 การทดลองและผลการทดลอง 59
4.1 ส่วนของการของรหัส 59
4.1.1 วงจรลดค่าความถี่ 59
4.1.2 วงจรบัพเฟอร์ 61
4.1.3 วงจรเข้ารหัส 63
4.2 ส่วนของการถอดรหัส 67
4.2.1 วงจรไวตาบิดิสแตนซ์ 67
4.2.2 วงจรคอมพิวเมตริกซ์ 67
4.2.3 วงจรคอมแพร์เมตริกซ์ 68
4.2.4 วงจรซีเลกเมตริกซ์ 69
4.2.5 วงจรคอมแพร์ 69
4.2.6 วงจรไฟน์สมอลเลท 70
4.2.7 วงจรคอนโทรล 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 วงจรรีดิวซ์ 71
4.2.9 วงจรเมตริกซ์ 72
4.2.10 วงจรเทรดแบคเอซีเอ 73
4.2.11 วงจรเทรดแบคคอนโทรล 73
4.2.12 วงจรถอดรหัส 74
4.3 ส่วนของการวัดค่าความผิดพลาดของข้อมูล 75
4.3.1วงจรอีเนเบิล 75
4.3.2 วงจรสุ่มค่า 75
4.3.3 วงจรจำลองการผิดพลาดของข้อมูล 77
4.3.4 วงจรหน่วงเวลา 77
4.3.5 วงจรเปรียบเทียบสัญญาณ (Comparator) 78
4.4 ผลการทดลองของการวัดค่าความผิดพลาดของข้อมูล 79
บทที่ 5 บทวิจารณ์และบทสรุป 83
กิตติกรรมประกาศ
หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 2.1 แผนภาพบล็อกระบบสื่อสารแบบดิจิตอล 4

รูปที่ 2.2 แผนภาพของเครื่องเข้ารหัสสำหรับรหัสลิเนียร์บล็อก 6

รูปที่ 2.3 การตรวจสอบและแก้ไขความผิดพลาด 8

รูปที่ 2.4 เครื่องเข้ารหัสวงเวียน (n-k) ที่เกิดจาก

$$g(x) = 1 + g_1 x + g_2 x^2 + \ldots + g_{n-k-1} x^{n-k-1} + x^{n-k}$$ 10

รูปที่ 2.5 แผนภาพของวงจรคำนวณซินโดรม (n-k) สำหรับรหัสวงเวียน 11

รูปที่ 2.6 การเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยความยาวคอนสเตร็นท์ M
และอัตราการเข้ารหัส k/n 12

รูปที่ 2.7 การเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2 ความยาวคอนสเตร็นท์
M = 3 13

รูปที่ 2.8 ขั้นตอนการเข้ารหัสแบบคอนโวลูชั่นโค้ด อัตราการเข้ารหัส 1/2 ความยาวคอนสเตร็นท์
M = 3 14

รูปที่ 2.9 แผนภาพสเต็จของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2
และ M = 3 15

รูปที่ 2.10 แผนภาพต้นไม้ของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2
และ M = 3 16

รูปที่ 2.11 แผนภาพเทลลิสของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2
และ M = 3 17

รูปที่ 2.12 การถอดรหัสโดยใช้แผนภาพเทลลิส ของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วย
อัตราการเข้ารหัส 1/2 และ M = 3 18

รูปที่ 2.13 เมตริกซ์เส้นทางสองเส้นทางที่มาพบกัน 19

รูปที่ 2.14 การเลือกเส้นทางที่เหลืออยู่

(a) เส้นทางที่เหลือเมื่อเวลา t_2 20

(b) เส้นทางที่เหลือเมื่อเวลา t_3 20

(c) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_4 21

(d) เส้นทางที่เหลือเมื่อเวลา t_4 21

(e) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_5 21

(f) เส้นทางที่เหลือเมื่อเวลา t_5 22

(g) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_6 22

(h) เส้นทางที่เหลือเมื่อเวลา t_6 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 แสดงผลลัพธ์การจำลองการทำงานของการเข้ารหัสแบบคอนโวลูชั่น(อัตราการเข้ารหัสข้อมูล1/2) ด้วยวิธีของไวตาบิดีโค้ดดิ้งบนช่องสัญญาณรบกวนขาว(AWGN:Aditive White Gaussian Noise) ในกรณีที่มีค่าความยาวของคอนสเตร็นท์ต่างกัน 23
รูปที่ 2.16 แสดงขั้นตอนการออกแบบระบบดิจิตอล 24
รูปที่ 2.17 การออกแบบระบบเส้นทางข้อมูล 25
รูปที่ 2.18 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้ 26
รูปที่ 2.19 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม 27
รูปที่ 2.20 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต 28
รูปที่ 2.21 โครงสร้างของบอดี้แพ็กเก็ต 29
รูปที่ 2.22 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ 29
รูปที่ 2.23 ตัวดำเนินการใน วีเอชดีแอล 30
รูปที่ 2.24 รูปแบบของการบรรยายแบบโปรเซส 31
รูปที่ 2.25 ขั้นตอนการออกแบบจากบนลงล่าง 33
รูปที่ 2.26 ผังแสดงการแบ่งกลุ่มของวงจรรวมเอซิก 35
รูปที่ 2.27 แสดงลักษณะของตัว FPGA และการนำไปใช้งาน 36
รูปที่ 2.28 แสดงโครงสร้างของชิพอุปกรณ์เอฟพีจีเอ ตระกูล FLEX 10 K 37
รูปที่ 2.29 แสดงโครงสร้างภายในของแอลอี 37
รูปที่ 2.30 แสดงการใช้งาน LUT เป็นโครงข่ายของลอจิก 38
รูปที่ 2.31 แสดงโครงข่ายของการเชื่อมต่อ 38
รูปที่ 2.32 แสดงโครงสร้างภายในของ แอลเอบี 39
รูปที่ 2.33 แสดงโครงสร้างภายในอีเอบี 40
รูปที่ 2.34 แสดงโครงสร้างภายในของไอโออี 41
รูปที่ 2.35 การโปรแกรมลงในชิพอุปกรณ์เอฟพีจีเอ 42
รูปที่ 3.1 แสดงโปรแกรมการใช้งานส่งค่าบิตข้อมูลออกทางพอร์ตขนาน 44
รูปที่ 3.2 แสดงสัญลักษณ์ของส่วนวงจรบัพเฟอร์ 45
รูปที่ 3.3 แสดงสัญลักษณ์ของวงจรลดค่าความถี่ 45
รูปที่ 3.4 แสดงสัญลักษณ์ของส่วนการเข้ารหัส แบบคอนโวลูชั่น 46
รูปที่ 3.5 แสดงลักษณะการเข้ารหัสแบบคอนโวลูชั่น 46
รูปที่ 3.6 แสดงการทำงานในส่วนของการเข้ารหัส 47
รูปที่ 3.7 แสดงส่วนของการถอดรหัส 47
รูปที่ 3.8 แสดงสัญลักษณ์ของวงจรไวตาบิดิสแตนซ์ ที่ได้จากการคอมไพล์ 48
รูปที่ 3.9 แสดงสัญลักษณ์ของวงจรคอมพิวเมตริกซ์ 48
รูปที่ 3.10 แสดงแผนภาพเทลลิสของการพิจารณาค่าความผิดพลาดในแต่ละเส้นทาง 49
รูปที่ 3.11 แสดงสัญลักษณ์ของวงจรคอมแพร์เมตริกซ์ 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 แสดงแผนภาพเทลลิสที่เส้นทาง 2 เส้นทางมาบรรจบกันทั้ง 4 จุด 50
รูปที่ 3.13 แสดงสัญลักษณ์ของวงจรซีเลกเมตริกซ์ 50
รูปที่ 3.14 แสดงสัญลักษณ์ของวงจรคอมแพร์ 51
รูปที่ 3.15 แสดงสัญลักษณ์ของวงจรไฟน์สมอลเลท 51
รูปที่ 3.16 แสดงสัญลักษณ์ของวงจรคอนโทรล 51
รูปที่ 3.17 แผนภาพเทลลิสแสดงสัญญาณคอนโทรลในแต่ละชุดของสัญญาณ 52
รูปที่ 3.18 แสดงสัญลักษณ์ของวงจรรีดิวซ์ 52
รูปที่ 3.19 แสดงสัญลักษณ์ของวงจรเมตริกซ์ 52
รูปที่ 3.20 แสดงสัญลักษณ์ของวงจรเทรดแบคเอซีเอส 53
รูปที่ 3.21 แสดงสัญลักษณ์ของวงจรเทรดแบคคอนโทรล 53
รูปที่ 3.22 แสดงโครงสร้างของวงจรพรีดีโค้ดเดอร์ซึ่งประกอบด้วยส่วนต่างๆที่ได้กล่าวมาข้างต้น 54
รูปที่ 3.23 แสดงสัญลักษณ์ของวงจรถอดรหัส(Decoder) 55
รูปที่ 3.24 แสดงสัญลักษณ์ของวงจรอีเนเบิล 55
รูปที่ 3.25 แสดงสัญลักษณ์ของวงจรสุ่มค่า 56
รูปที่ 3.26 แสดงสัญลักษณ์ของวงจรจำลองการผิดพลาดของข้อมูล 56
รูปที่ 3.27 แสดงสัญลักษณ์ของวงจรหน่วงเวลา 57
รูปที่ 3.28 แสดงสัญลักษณ์ของวงจรเปรียบเทียบสัญญาณ 57
รูปที่ 3.29 แสดงการรวมของวงจรเข้ารหัสกับวงจรถอดรหัสและวงจรการวัดค่าความผิดพลาดของข้อมูล 58
รูปที่ 4.1 แสดงสัญลักษณ์ของวงจรลดค่าความถี่ที่ได้จากการคอมไพล์ 59
รูปที่ 4.2 แสดงส่วนของอินพุตของวงจรลดค่าความถี่(สัญญาณนาฬิกาจากบอร์ดเอฟพีจีเอ) 59
รูปที่ 4.3 แสดงส่วนของเอาต์พุตของวงจรลดค่าความถี่ (ใช้เปรียบเทียบสัญญาณที่เกิดจากการเข้ารหัส) 60
รูปที่ 4.4 แสดงส่วนของเอาต์พุตของวงจรลดค่าความถี่(ใช้ในการวัดค่าความผิดพลาดของข้อมูล) 60
รูปที่ 4.5 แสดงสัญลักษณ์ของวงจรบัฟเฟอร์ที่ได้จากการคอมไพล์ 61
รูปที่ 4.6 แชนแนล 1 แสดงส่วนของอินพุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่แสดงสัญญาณนาฬิกาที่ความถี่ 1.501 Hz 61
รูปที่ 4.7 แชนแนล 1 แสดงส่วนของเอาท์พุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่แสดงสัญญาณนาฬิกาความถี่ 1.501 Hz 62
รูปที่ 4.8 แชนแนล 1 แสดงส่วนของอินพุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่สัญญาณนาฬิกาที่ความถี่ 64.03 Hz 62
รูปที่ 4.9 แชนแนล 1 แสดงส่วนของเอาท์พุตวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่สัญญาณนาฬิกาที่ความถี่ 64.02 Hz 63
รูปที่ 4.10 แสดงสัญลักษณ์ของวงจรเข้ารหัสที่ได้จากการคอมไพล์ 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 0001110001 64
รูปที่ 4.12 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1 ของการทำงาน
ของวงจรเข้ารหัสเมื่อป้อนอินพุต 0001110001 64
รูปที่ 4.13 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1001001110 65
รูปที่ 4.14 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1ของการทำงาน
ของวงจรเข้ารหัสเมื่อป้อนอินพุต 1001001110 65
รูปที่ 4.15 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1100110011 66
รูปที่ 4.16 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1 ของการทำงาน
ของวงจรเข้ารหัสเมื่อป้อนอินพุต 1100110011 66
รูปที่ 4.17 แสดงสัญลักษณ์ของวงจรไวตาบิดิสแตนซ์ ที่ได้จากการคอมไพล์ 67
รูปที่ 4.18 แสดงการจำลองการทำงานของวงจรไวตาบิดิสแตนซ์เมื่อป้อนอินพุต
(11101111101111101100111) 67
รูปที่ 4.19 แสดงสัญลักษณ์ของวงจรคอมพิวเมตริกซ์ที่ได้จากการคอมไพล์ 67
รูปที่ 4.20 แสดงการจำลองการทำงานของวงจรคอมพิวเมตริกซ์เมื่อป้อนอินพุต
(11101111101111101100111) 68
รูปที่ 4.21 แสดงสัญลักษณ์ของวงจรคอมแพร์เมตริกซ์ที่ได้จากการคอมไพล์ 68
รูปที่ 4.22 แสดงการจำลองการทำงานของวงจรคอมพิวเมตริกซ์เมื่อป้อนอินพุต
(11101111101111101100111) 68
รูปที่ 4.23 แสดงสัญลักษณ์ของวงจรซีเลกเมตริกซ์ที่ได้จากการคอมไพล์ 69
รูปที่ 4.24 แสดงการจำลองการทำงานของวงจรซีเลกเมตริกซ์เมื่อป้อนอินพุต
(11101111101111101100111) 69
รูปที่ 4.25 แสดงสัญลักษณ์ของวงจรคอมแพร์ที่ได้จากการคอมไพล์ 69
รูปที่ 4.26 แสดงการจำลองการทำงานของวงจรคอมแพร์เมื่อป้อนอินพุต
(11101111101111101100111) 70
รูปที่ 4.27 แสดงสัญลักษณ์ของวงจรไฟน์สมอลเลทที่ได้จากการคอมไพล์ 70
รูปที่ 4.28 แสดงการจำลองการทำงานของวงจรไฟน์สมอลเลทเมื่อป้อนอินพุต
(11101111101111101100111) 70
รูปที่ 4.29 แสดงสัญลักษณ์ของวงจรคอนโทรลที่ได้จากการคอมไพล์ 71
รูปที่ 4.30 แสดงการจำลองการทำงานของวงจรคอนโทรลเมื่อป้อนอินพุต
(11101111101111101100111) 71
รูปที่ 4.31 แสดงสัญลักษณ์ของวงจรรีดิวซ์ที่ได้จากการคอมไพล์ 71
รูปที่ 4.32 แสดงการจำลองการทำงานของวงจรรีดิวซ์เมื่อป้อนอินพุต
(11101111101111101100111) 72
รูปที่ 4.33 แสดงสัญลักษณ์ของวงจรเมตริกซ์ที่ได้จากการคอมไพล์ 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.34 แสดงการจำลองการทำงานของวงจรรีดิวซ์เมื่อป้อนอินพุต
(11101111101111011001111) 72
รูปที่ 4.35 แสดงสัญลักษณ์ของวงจรเทรดแบคเอซีเอสที่ได้จากการคอมไพล์ 73
รูปที่ 4.36 แสดงการจำลองการทำงานของวงจรเทรดแบคเอซีเอสเมื่อป้อนอินพุต
(11101111101111011001111) 73
รูปที่ 4.37 แสดงสัญลักษณ์ของวงจรเทรดแบคคอนโทรลที่ได้จากการคอมไพล์ 73
รูปที่ 4.38 แสดงการจำลองการทำงานของวงจรเทรดแบคคอนโทรลเมื่อป้อนอินพุต
(11101111101111011001111) 74
รูปที่ 4.39 แสดงสัญลักษณ์ของวงจรถอดรหัสที่ได้จากการคอมไพล์ 74
รูปที่ 4.40 แสดงการจำลองการทำงานของวงจรถอดรหัสเมื่อป้อนอินพุต
(11101111101111011001111) 74
รูปที่ 4.41 แสดงสัญลักษณ์ของวงจรอีเนเบิลที่ได้จากการคอมไพล์ 75
รูปที่ 4.42 แสดงการจำลองการทำงานของวงจรอีเนเบิลเมื่อป้อนอินพุต “0” 75
รูปที่ 4.43 แสดงการจำลองการทำงานของวงจรอีเนเบิลเมื่อป้อนอินพุต “1” 75
รูปที่ 4.44 แสดงสัญลักษณ์ของวงจรสุ่มค่าที่ได้จากการคอมไพล์ 75
รูปที่ 4.45 แสดงการจำลองการทำงานของวงจรสุ่มค่าเมื่อเทียบกับสัญญาณนาฬิกา 76
รูปที่ 4.46 แชนแนล 1 แสดงข้อมูลที่สุ่มค่า และ แชนแนล 2 แสดงสัญญาณนาฬิกาของวงจรสุ่มค่า 76
รูปที่ 4.47 แสดงสัญลักษณ์ของวงจรจำลองการผิดพลาดของข้อมูลที่ได้จากการคอมไพล์ 77
รูปที่ 4.48 แสดงการจำลองการทำงานของวงจรจำลองการผิดพลาดของข้อมูล 77
รูปที่ 4.49 แสดงสัญลักษณ์ของวงจรหน่วงเวลาที่ได้จากการคอมไพล์ 77
รูปที่ 4.50 แสดงการจำลองการทำงานของวงจรหน่วงเวลาเมื่อป้อนอินพุตบิต 101 78
รูปที่ 4.51 แสดงสัญลักษณ์ของวงจรเปรียบเทียบสัญญาณที่ได้จากการคอมไพล์ 78
รูปที่ 4.52 แสดงการจำลองการทำงานของวงจรเปรียบเทียบสัญญาณเมื่ออินพุต a = 0011 และ
อินพุต b = 0101 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงค่าความจริงของส่วนการเข้ารหัส 46

ตารางที่ 3.2 แสดงค่าความเป็นจริงของระยะแฮมมิ่งเมื่อเทียบกับค่าน้ำหนัก
ในแต่ละเส้นทางที่ทำการเลือก 48

ตารางที่ 3.3 แสดงตำแหน่งของสัญญาณคอนโทรลในแต่ละชุดของสัญญาณ 52

ตารางที่ 4.1 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูล
ที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 0001110001 78

ตารางที่ 4.2 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูล
ที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 1001001110 78

ตารางที่ 4.3 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูล
ที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 1100110011 79

ตารางที่ 4.4 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูล
ที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุตแบบสุ่มค่า 79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์

การสื่อสารข้อมูลในโลกปัจจุบันถือว่าเป็นการสื่อสารในรูปแบบดิจิตอล ทั้งสิ้นเนื่องจากมีประสิทธิภาพมาก ทั้งในด้าน ความคมชัดของเสียง ที่ส่งในรูปแบบของดิจิตอล และยังมีความต้านทานต่อสัญญาณรบกวนสูงเมื่อเทียบกับสัญญาณอนาลอก ฉะนั้น ในปัจจุบันได้มีการพัฒนา เทคนิคใหม่ ๆ เพื่อเพิ่มประสิทธิภาพของระบบการสื่อสารข้อมูลแบบดิจิตอลให้มากขึ้น เพื่อให้ประสบความสำเร็จในการสื่อสารข้อมูลมากที่สุด โดยโครงงานนี้ ได้นำเอาเทคนิคการเข้ารหัส มาเพิ่มประสิทธิภาพในการสื่อสารข้อมูลดิจิตอล

การพัฒนาการทางด้านของการเข้ารหัสสัญญาณแบบดิจิตอลนั้นเริ่มตั้งแต่ปี 1948 โดย คลอดด์ แชลนอน (Claude Shannon) ได้พยายามที่จะหาแนวคิดที่จัดการเกี่ยวกับงานทางด้าน การติดต่อสื่อสารที่มีสัญญาณรบกวน (Noisy communication) โดยพัฒนาเทคนิคการเข้ารหัส-ถอดรหัส ซึ่งเป็นเทคนิคอย่างหนึ่งที่สามารถลด ความผิดพลาดในระบบการสื่อสารข้อมูลแบบดิจิตอล ซึ่งแบ่งการเข้ารหัสได้เป็น 2 รูปแบบ คือ รหัสบล็อก (Block code) และ คอนโวลูชั่นโค้ด (convolutional code)

ในปี 1950 ได้มีการพัฒนาเทคนิคการเข้ารหัสแบบ รหัสบล็อก เพื่อเพิ่มประสิทธิภาพของระบบสื่อสารข้อมูลจนมาถึงประมาณ 10 ปีหลัง ได้มีการคิดค้นวิธีการเข้ารหัสแบบรหัสลิเนียร์บล็อก (Linear block code) หรือ รหัสแฮมมิ่ง (Hamming code) ปี 1950 ซึ่งเป็นการแก้ไขบิตข้อมูลที่ผิดพลาดในข้อมูลที่มีลักษณะบิตเดียว

จนกระทั่ง ปี 1960 ได้มีการคิดค้น เพื่อหาโครงสร้างของการเข้ารหัสรูปแบบใหม่ๆ เพื่อที่จะได้การเข้ารหัสแบบใหม่ที่ดีขึ้น แต่ก็ยังไม่ประสบความสำเร็จ ต่อมาในปี 1961 เมกกิท(Meggit) ได้คิดค้นวิธีการถอดรหัสที่ใช้กับ รหัสวงเวียน (Cyclic code) แต่ได้รับการกลั่นกรองสำหรับการประยุกต์ใช้งานจริง ซึ่งเทคนิค ที่ทราบ คือทำการถอดรหัสและตรวจจับ ความผิดพลาดซึ่งเป็นความคิดของ เมกกิท ซึ่งมี คาซามิ(Kasami) มิเชล(Mitchel) รูดอฟ(Rudolph) เป็นผู้ร่วมทีม อย่างไรก็ตามวิธีการในการตรวจจับความผิดพลาดของ เมกกิท นี้ถือว่าเป็นวิธีที่ให้อัตราการส่งผ่านข้อมูลสูง และมีสมรรถภาพในการแก้ไขข้อผิดพลาดของข้อมูลที่ผิดได้ดี ซึ่งวิธีการเข้ารหัสแบบ รหัสวงเวียน นั้นจะสามารถแก้ไขความผิดพลาดที่เกิดขึ้นแบบ ฉับพลัน(Burst) ได้อีกด้วยโดยใช้เทคนิควิธีการถอดรหัสแบบ ตรวจจับข้อผิดพลาด (Error-trapping) ต่อมาในปี 1959 โบส ชอดฮูรี(Bose Chaudhuri) และ ฮอคเควนแฮม (Hocquenghem) ได้ค้นพบวิธีการเข้ารหัสเพื่อแก้ไขข้อผิดพลาด (Error-correcting code) ที่ชื่อว่า รหัสวงเวียนพิเศษ (BCH:Bose-Chandhuri-Hocquenghem code) ต่อมาได้มีการคิดค้นวิธีการถอดรหัสที่มีความเหมาะสม ซึ่งวิธีการนี้เรียกว่า ปีเตอร์สัน อัลกอริทึม (Peterson algorithm) คิดค้นได้ในปี 1960 และได้มีการพัฒนาวิธีการเข้ารหัสต่อมาในปี 1961 โดย กอเรนสไตน์ (Gorenstein) เซียร์เลอร์(Zierler) ต่อมาได้พัฒนาต่อโดย เบอร์กแคม (Berlekamp) เมสเซ (Massey) เชน (Chien) โฟนี่ (Foney) และ ลิน (Lin) ในปี 1960 รีด (Reed) และ โซโลมอน (Solomon) ได้ค้นพบสิ่งที่สำคัญขึ้น คือนำ รหัสวงเวียนพิเศษ มาใช้ในแบบนอนไบนารี่(Nonbinary) ซึ่งสามารถแก้ไขความผิดพลาดของบิตของข้อมูล ที่เกิดขึ้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉับพลัน ได้ เรียกโค้ดแบบใหม่นี้ว่า รีด-โซโลมอน โค้ด(Reed-Solomon code) ซึ่งใช้การถอดรหัสแบบรหัสวงเวียน ต่อมาในปี 1963 เมสเซ ได้ค้นพบอัลกอริทึมในการถอดรหัสที่มีความเหมาะสมทำให้เกิดประสิทธิภาพที่สูงขึ้น

การพัฒนาวิธีการเข้ารหัสตั้งแต่ปี 1960 ได้มีการพัฒนาต่อมาเรื่อยๆ โดยในปี 1961 การเข้ารหัสแบบ คอนโวลูชั่นโค้ด ได้กำเนิดขึ้นครั้งแรกโดย เอเลียส (Elias) โวเซนคราฟท์ (Wozencraft) ซึ่งใช้วิธีการถอดรหัสแบบซีเควนเชี่ยน ในปี 1963 ได้อัลกอริทึมที่มีชื่อว่า โฟโน อัลกอริทึม (Fano algorithm) เป็นอัลกอริทึมในการถอดรหัสและต่อมาได้ออกรูปแบบการถอดรหัสแบบซีเควนเชี่ยนใหม่เรียกว่า ซีเจ สแต็ก อัลกอริทึม(ZJ stack algorithm) โดยผู้คิดค้นคือ ซีแกนกรูฟ (Zigangrov) ปี 1966 และ เจลลิเนก (Jelinek) ปี 1969 ต่อมาในปี 1967 ไวตาบิ (Viterbi) ได้เสนอวิธีการถอดรหัสคือ ไวตาบิ อัลกอริทึม เป็นวิธีการถอดรหัสที่ดีวิธีหนึ่งของการถอดรหัสแบบ แม็กซิมั่ม ไลรี่ฮูด (Maximum-likelihood) ตั้งแต่นั้นมา โอมารุ (Omura) (ปี 1969) และ โฟนี่ (ปี 1972-1974) ได้พิสูจน์ว่า ไวตาบิ อัลกอริทึม เป็นวิธีการถอดรหัสที่ดีที่สุดของการถอดรหัส ในการเข้ารหัสแบบคอนโวลูชั่นโค้ดที่มี ค่าความยาวของข้อมูล (Constraint length) น้อย และได้มีการพัฒนาคิดค้น วิธีการถอดรหัสแบบ เทสโฮลดีโค้ดดิ้ง (Threshold decoding) โดย เมสเซ ซึ่งมีความแตกต่างจากการถอดรหัสแบบซีเควนเชี่ยน และไวตาบิ

ในปีต่อๆมา ตั้งแต่ปี 1959 เฮเกลบาร์เกอร์ (Hegelbarger) ได้ทำการแนะนำคำจำกัดความของการแก้ไขข้อผิดพลาดที่เกิดแบบฉับพลัน (Burst-error correction) ที่เกิดขึ้นซ้ำ ซึ่งได้มีการพัฒนาเทคนิคต่างๆ สำหรับจัดการให้คอนโวลูชั่นโค้ดมีประสิทธิภาพในการแก้ไขการแก้ไขข้อผิดพลาดที่เกิดแบบฉับพลันได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 เพื่อศึกษาหลักการของการเข้ารหัสและถอดรหัสข้อมูลดิจิตอลแบบ คอนโวลูชั่นโค้ด

1.2.2 เพื่อศึกษาการเขียนภาษา วีเอชดีแอล ในการออกแบบวงจรทางลอจิก

1.2.3 นำทฤษฎีมาประยุกต์เพื่อออกแบบตัวเข้าและถอดรหัสบนชิพอุปกรณ์เอฟพีจีเอในรูปแบบคอนโวลูชั่นโค้ด

1.2.4 เพื่อศึกษาการใช้งานและการทำงานของ ชิพอุปกรณ์เอฟพีจีเอ

1.2.5 เพื่อศึกษาวิธีการออกแบบจากบนลงล่าง (Top-down Design) และโปรแกรมที่ช่วยในการออกแบบวงจรทางลอจิก (Max+plus II)

1.3 ขอบเขตของปริญญานิพนธ์

โครงงานนี้ เป็นการออกแบบการเข้ารหัส-ถอดรหัสข้อมูลดิจิตอล ที่ใช้ในการสื่อสารข้อมูล โดยใช้วิธีการเข้ารหัสแบบคอนโวลูชั่นโค้ดซึ่งมีข้อดีคือ ช่วยลดความผิดพลาดที่จะเกิดขึ้นกับข้อมูลและเพิ่มความน่าเชื่อถือให้กับระบบการสื่อสาร โดยโครงงานนี้จะใช้การเขียนชุดคำสั่งด้วยภาษา วีเอชดีแอล เพื่อใช้ในการออกแบบวงจรทางลอจิกภายในชิพอุปกรณ์เอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 เนื้อหาของปริญญานิพนธ์

ในบทที่ 2 จะกล่าวถึงทฤษฎีการเข้ารหัสและถอดรหัสข้อมูลดิจิตอลแบบคอนโวลูชั่น ซึ่งจะบอกถึงรายละเอียดพื้นฐานของคอนโวลูชั่นโค้ด และวิธีการเข้ารหัส-ถอดรหัส ทฤษฎีของภาษา วีเอชดีแอล รวมทั้งตัวอย่างการออกแบบ ในส่วนของการออกแบบโครงสร้าง ทฤษฎีและหลักการของชิพอุปกรณ์เอฟพีจีเอรวมถึงหลักการออกแบบเพื่อโปรแกรมลงบนชิพอุปกรณ์เอฟพีจีเอ และโครงสร้างในส่วนต่างๆ ของบอร์ด เอฟพีจีเอ

ในบทที่ 3 กล่าวถึงแนวความคิดในการออกแบบตัวเข้ารหัส (Encoder) และ ตัวถอดรหัส (Decoder) แบบคอนโวลูชั่นโค้ด และรวมไปถึงโครงสร้างในส่วนต่างๆ ของโครงงาน

ในบทที่ 4 การทดลอง และผลการทดลอง

ในบทที่ 5 กล่าวถึง บทวิจารณ์และสรุปผล

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

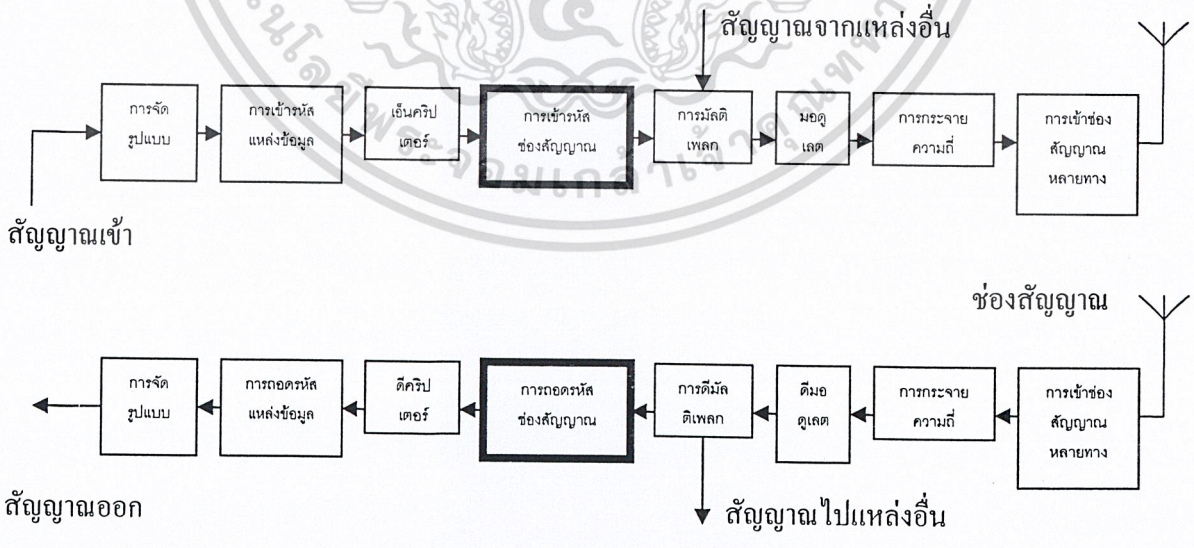
บทที่ 2
ทฤษฎีและหลักการ

2.1 พื้นฐานการเข้ารหัส

สิ่งที่สำคัญสำหรับระบบการสื่อสารก็คือความสามารถในการสื่อสาร และตามปกติถ้าไม่มีสัญญาณรบกวนเราจะส่งสัญญาณอย่างไรก็จะได้รับสัญญาณอย่างนั้น แม้ว่าจะถูกลดกำลังด้วยระยะทางก็ตาม แต่ถ้ามีสัญญาณรบกวนแล้วนั้นจะเพี้ยนไปหรือรับไม่ได้ ถ้าสัญญาณรบกวนมีกำลังมากกว่าสัญญาณ ฉะนั้นต้องออกแบบระบบสื่อสารที่มีการทำงานที่ดีหรืออัตราส่วนกำลังของสัญญาณต่อกำลังของสัญญาณรบกวน (Signal to Noise power Ratio : SNR) สูงสำหรับระบบสื่อสารอนาล็อก ถ้าอัตราส่วนกำลังของสัญญาณต่อกำลังของสัญญาณรบกวนมีค่ามากขึ้นก็แสดงว่าเครื่องมือสื่อสารสามารถขจัดสัญญาณรบกวนได้ดีสำหรับพารามิเตอร์ในการวัดความสามารถในการทำงานของระบบสื่อสารดิจิตอลก็คือ อัตราความผิดพลาด (Error rate) ซึ่งอัตราการส่งสัญลักษณ์ข้อมูล (Symbol rate) จะเท่ากับอัตราการส่งบิต (Bit rate) เมื่อเป็นการส่งข้อมูลแบบรหัสฐานสอง (Binary) ดังนั้นในการออกแบบต้องให้มีอัตราส่วนสัญญาณต่อสัญญาณรบกวน มีค่าสูง และมีค่าอัตราความผิดพลาดน้อย (Bit error rate : BER)

การปรับปรุงสมรรถนะของระบบการสื่อสารสามารถทำได้หลายวิธี

1) การใช้เทคนิคไดเวอซิตี้ เช่น การรวมอัตราส่วนสูงสุด ไดเวอร์ซิตี้ทางความถี่ ไดเวอซิตี้ทางเวลา
2) การเข้ารหัส (Coding) ซึ่งกระบวนการเข้ารหัสสามารถแบ่งออกตามวัตถุประสงค์ได้เป็น 2 ประเภท คือ
   2.1 การเข้ารหัสแหล่งกำเนิด (Source coding)
   2.2 การเข้ารหัสช่องสัญญาณ (Channel coding)



รูปที่ 2.1 แผนภาพบล็อกระบบสื่อสารแบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 แสดงแผนภาพบล็อกระบบการสื่อสารดิจิตอล จากรูปจะเห็นว่าข่าวสารสามารถกำเนิดในรูปของอนาล็อกหรือดิจิตอลก็ได้ถ้าอยู่ในรูปอนาล็อกก็จะผ่านการจัดรูปแบบ (Formatting) โดยการชักตัวอย่าง การควอนไทซ์และแปลงเป็นสัญญาณพีซีเอ็ม สัญญาณข่าวสารที่เป็นดิจิตอลอาจจะผ่านหรือไม่ผ่านเครื่องเอ็นคริปเตอร์ (Endcryptor) ก็ได้แล้วแต่ความสำคัญและลักษณะงานข้อมูล เครื่องเอ็นคริปเตอร์นี้จะทำหน้าที่แปลงข้อมูลให้อยู่ในรูปที่มีความปลอดภัยจากการปลอมแปลงหรือขโมยข้อมูลไปใช้งานได้ดีขึ้น จากนั้นข้อมูลจะผ่านเครื่องเข้ารหัสช่องสัญญาณ โดยเครื่องเข้ารหัสช่องสัญญาณนี้จะแปลงข้อมูลให้ส่งผ่านช่องสัญญาณโดยมีความผิดพลาดน้อย และเป็นการช่วยเพิ่มอัตราข้อมูล ขั้นตอนการเข้ารหัสช่องสัญญาณนี้จะอยู่ระหว่างการเข้ารหัสแหล่งกำเนิด และการมอดูเลต ซึ่งการเข้ารหัสช่องสัญญาณสามารถแบ่งออกได้เป็น 2 ชนิด คือ

1) การเข้ารหัสรูปคลื่น (Waveform coding) เป็นการสร้างรูปคลื่นใหม่ขึ้นมาเพื่อปรับปรุงคุณสมบัติให้ดีกว่ารูปคลื่นเดิม เช่น การส่งสัญญาณตั้งฉากกัน (Orthogonal) สัญญาณตรงข้ามกัน (Antipodal) สัญญาณตั้งฉากกันสองขั้ว (Biorthogonal)

2) การเข้ารหัสแบบการจัดลำดับโครงสร้างของข้อมูล (Structured Sequences) เป็นการเพิ่มบิตเข้าไปรวมกับบิตของข้อมูล เพื่อช่วยในการตรวจสอบความผิดพลาดของบิตข้อมูลอันเนื่องจากสัญญาณรบกวนจากช่องสัญญาณ ตัวอย่างของการเข้ารหัสแบบนี้ เช่น การเข้ารหัสลิเนียร์บล็อก (Linear block code) การเข้ารหัสแบบวงเวียน(Cyclic code) และการเข้ารหัสแบบคอนโวลูชั่นโค้ด (Convolutional code)

ข้อมูลที่ผ่านการเข้ารหัสช่องสัญญาณแล้วจะถูกมัลติเพล็ก จากนั้นสัญญาณจะถูกส่งไปยังการมอดูเลตตามระบบดิจิตอลมอดูเลตแล้วจะผ่านช่องสัญญาณ ซึ่งจะถูกรบกวนด้วยสัญญาณรบกวนเกาส์เซียน ช่องสัญญาณจะเป็นตัวกลางใดๆก็ได้แล้วแต่ความต้องการของงาน เช่น อากาศ สายไขว้คู่และเส้นใยแก้ว เมื่อข้อมูลมาถึงปลายทางก็จะถูกดีมอดูเลตและดีมัลติเพล็ก จากนั้นจะถูกถอดรหัสช่องสัญญาณ (Channel decoding) สัญญาณที่ได้ผ่านเครื่องดีคริปเตอร์ (Decryptor) ถ้าใช้เครื่องเอ็นคริปเตอร์ที่ต้นทาง แล้วจะผ่านการถอดรหัสแหล่งกำเนิดเมื่อได้สัญญาณดิจิตอล แล้วจะถูกแปลงกลับเป็นอนาล็อกหรือคงอยู่ในรูปดิจิตอลแล้วแต่ว่าสัญญาณเดิมที่ส่งเป็นอนาล็อกหรือดิจิตอล

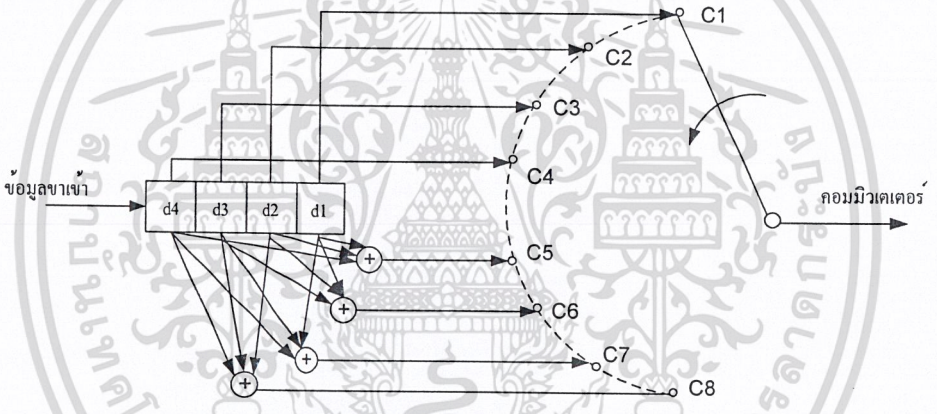
จุดประสงค์ของการเข้ารหัส เพื่อเป็นการเพิ่มความน่าเชื่อถือในการรับส่งข้อมูลให้มากขึ้น ดังนั้นรหัสที่สร้างจึงต้องมีการตรวจสอบความผิดพลาดของข้อมูลที่ได้รับเข้ามาว่ามีความผิดพลาดหรือไม่ ถ้าไม่มีความผิดพลาดก็นำข้อมูลที่ได้รับไปประมวลผลในขั้นตอนต่อไป หากเกิดความผิดพลาดขึ้นก็อาจให้ด้านส่งทำการส่งข้อมูลเดิมมาใหม่ ในการตรวจสอบความผิดพลาดที่เครื่องรับนั้นมีวิธีง่ายๆที่สามารถจะตรวจสอบได้ว่าข้อมูลที่รับเข้ามาเกิดความผิดพลาดหรือไม่ แต่ไม่สามารถจะบอกได้ว่าผิดพลาดที่ตำแหน่งใด ซึ่งจะเรียกรหัสตรวจสอบความผิดพลาดนี้ว่า รหัสตรวจคู่ (Parity check code) หลักการของรหัสตรวจคู่ คือ จะทำการเติมบิตต่อท้ายข่าวสาร (information bit) ที่ทางด้านส่งเพื่อให้ผลบวกเลขฐานสองของบิตทั้งหมด (รวมทั้งบิตที่เพิ่มด้วย) เป็น 0 หรือ 1 แล้วแต่ว่าเป็นภาวะคู่หรือภาวะคี่ (Even / Odd Parity) บิตที่ทำการเพิ่มเพื่อตรวจสอบความผิดพลาดนี้เรียกว่า บิตตรวจคู่ (Parity check bit) ซึ่งวิธีการดังกล่าวสามารถที่จะทำการตรวจสอบความผิดพลาดของข้อมูลได้ง่ายๆ โดยการคำนวณผลบวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขฐานสองของจำนวนบิตทั้งหมดที่ได้รับว่าเป็นเลขคู่ 0 หรือ เลขคี่ 1 ตามที่ตกลงกันหรือไม่ ถ้าไม่ก็แสดงว่าเกิดความผิดพลาดของข้อมูล สมมติบิตที่ได้รับเป็นชนิดภาวะคู่ และบิตที่ผิดพลาดคือบิตที่ 3 เป็น 1 แทนที่จะเป็น 0 ผลบวกเลขฐานสองของบิตทั้งหมดจะได้ 1 ก็จะทราบได้ทันทีว่าข้อมูลเกิดผิดพลาด แต่ไม่สามารถตรวจสอบได้ว่าผิดที่บิตไหน และสังเกตว่ากรณีที่เกิดผิด 2 บิตพร้อมกันก็ไม่สามารถตรวจสอบได้ว่าเกิดบิตผิด ฉะนั้นรหัสตรวจคู่จึงควรใช้ในกรณีที่ไม่ค่อยมีบิตผิดไม่เกินกว่า 1 บิต

2.1.1 รหัสลิเนียร์บล็อก

เป็นรหัสที่ประกอบด้วยเวกเตอร์ที่มีขนาดความยาวจำกัด ซึ่งจะถูกเรียกว่า คำรหัส (Code words) ความยาวของคำรหัสคือ จำนวนตัวประกอบทั้งหมดที่มีอยู่ในเวกเตอร์ซึ่งจะมีค่าเท่ากับ n โดยตัวประกอบแต่ละตัวจะเรียกว่า บิต ซึ่งคำรหัสมีจะเลือกจากจำนวนตัวเลขทั้งหมด q หมายเลขถ้าหากตัวประกอบในเวกเตอร์ประกอบด้วยค่าเพียงสองค่า คือ 1 และ 0 รหัสนี้ถูกเรียกว่า รหัสไบนารี่ หาก $q > 2$ รหัสนี้จะถูกเรียกว่า นอนไบนารี่



รูปที่ 2.2 แผนภาพของเครื่องเข้ารหัสสำหรับรหัสลิเนียร์บล็อก

1) ค่าพารามิเตอร์ของรหัสลิเนียร์บล็อก

ในการเข้ารหัสของข้อมูลซึ่งมีขนาดความยาวเท่ากับ k บิต หลังจากผ่านการเข้ารหัสจะได้ความยาวของคำรหัสมีค่าเท่ากับ n บิต ซึ่งขนาด n - k บิต เป็นบิตที่เพิ่มรวมเข้าไปกับข้อมูล จะถูกเรียกว่าพาริตี้บิตหรือบิตตรวจสอบ โดยรหัสที่ได้จะถูกเรียกว่าเป็น รหัส (n,k) ซึ่งอัตราส่วนของจำนวนบิตของข้อมูลกับความยาวของคำรหัส $k/n = R_c$ เรียกว่า อัตราการเข้ารหัส (Code rate)

ตัวพารามิเตอร์มีค่าหนึ่งที่มีความสำคัญของการเข้ารหัส คือ น้ำหนักแฮมมิ่ง (Hamming weight) หมายถึง จำนวนบิตของรหัสซึ่งมีค่าไม่เท่ากับศูนย์ เช่น รหัส 101000 จะมีน้ำหนักเท่ากับ 2 หากมีรหัสลิเนียร์บล็อก 2 ค่า คือ

$$C_i = 10011010$$

$$C_j = 11110000$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการหาค่าความแตกต่างระหว่างคำรหัสทั้งสองนี้จะวัดได้จากจำนวนบิตหรือตำแหน่งของบิตที่มีค่าแตกต่างกัน ซึ่งจะเรียกค่าที่ได้ว่าระยะแฮมมิ่ง (Hamming distance) เขียนได้เป็น d_{ij} โดยที่ $i \neq j$ จากตัวอย่างข้างต้นจะพบว่า ค่าระยะแฮมมิ่งมีค่าเท่ากับ 4 โดยที่ค่าของระยะแฮมมิ่งมีค่าระหว่าง $0 < d_{ij} \leq n$ ค่าที่ต่ำสุดของ d_{ij} สำหรับคำรหัสใดๆ จะเรียกว่า ระยะความแตกต่างต่ำสุด (Minimum distance) ของรหัส เขียนได้เป็น d_{min}

2) ความสามารถในการตรวจสอบและแก้ไขความผิดพลาดของข้อมูลของรหัสลิเนียร์บล็อก

รหัสลิเนียร์บล็อกสามารตรวจสอบและแก้ไขความผิดพลาดของข้อมูลได้โดยการตรวจสอบกับซินโดรม($S = RH^T$) ว่าถ้า $S = 0$ ก็แสดงว่ารหัสที่ได้รับนั้นไม่ผิดพลาดและถ้า $S \neq 0$ ก็แสดงว่ารหัสที่ได้รับผิดพลาดซึ่งพบว่ารหัสลิเนียร์บล็อก สามารถตรวจสอบและแก้ไขบิตที่ผิดพลาดได้ในการศึกษาความสามารถในการตรวจสอบและแก้ไขความผิดพลาดของข้อมูลของรหัสลิเนียร์บล็อกนั้นจะต้องใช้สิ่งเหล่านี้วัดความสามารถดังกล่าว เช่น น้ำหนักแฮมมิ่ง (Hamming weight) ของค่ารหัส และค่าระยะแฮมมิ่ง (Hamming distance) ระหว่างคำรหัส 2 คำ

ในรูปที่ 2.3 แสดงถึงค่าระยะแฮมมิ่งของรหัสสองรหัส คือ U และ V โดยจุดดำที่อยู่บนเส้นหมายถึงการที่ค่าของรหัสถูกรบกวน จากรูป 2.3 ก แสดงถึงด้านรับ รับสัญญาณที่เวกเตอร์ r_1 ซึ่งมีระยะห่างจาก U เท่ากับ 1 และระยะห่างจาก V เท่ากับ 4 การแก้ไขความผิดพลาดทำได้โดยอาศัยความน่าจะเป็นที่จะเกิดขึ้นได้มากที่สุด นั่นคือ ในกรณีนี้จะทำการเลือกสัญญาณ U เมื่อได้รับ r_1 นั่นคือ r_1 เกิดจากการที่เกิดความผิดพลาดขึ้น 1 บิต จากการส่งสัญญาณ U ในระหว่างการส่งสัญญาณ แต่ทางด้านรับก็สามารถที่จะแก้ไขความผิดพลาดนี้ได้ เช่นเดียวกันหากเกิดความผิดพลาดขึ้น 2 บิต เนื่องจากการส่งสัญญาณ U ซึ่งมีผลทำให้ด้านรับรับสัญญาณ r_2 ซึ่งจะมีระยะห่างจาก U เท่ากับ 2 และระยะห่างจาก V เท่ากับ 3 ดังแสดงในรูปที่ 2.3 ข การถอดรหัสยังคงเลือกว่าสัญญาณ r_2 ที่ได้รับเกิดจากการส่งสัญญาณ U นั่นเอง ในกรณีที่เกิดความผิดพลาดขึ้น 3 บิต ของการส่งสัญญาณ U ซึ่งด้านรับรับได้เวกเตอร์ r_3 ซึ่งจะมีระยะห่างจาก U เท่ากับ 3 และห่างจาก V เท่ากับ 2 ซึ่งแสดงดังรูป 2.3 ค ในกรณีนี้ตัวถอดรหัสจะทำการเลือกสัญญาณ V ว่าเป็นสัญญาณที่ส่งมาจากด้านส่ง นั่นคือเกิดความผิดพลาดในการถอดรหัส

จากรูป 2.3 สามารถสรุปได้ว่า การตรวจสอบความผิดพลาดจะสามารถตรวจสอบได้ในกรณีที่มีความผิดพลาดเกิดขึ้น 1 , 2 , 3 และ 4 บิต ซึ่งแสดงเป็นจุดสีดำ ในกรณีที่ความผิดพลาดเกิดขึ้น 5 บิตเป็นผลทำให้ด้านรับจะรับสัญญาณเวกเตอร์ V จากการส่งสัญญาณเวกเตอร์ U ซึ่งเมื่อเกิดเหตุการณ์นี้ขึ้นด้านรับจะไม่สามารถตรวจสอบความผิดพลาดที่เกิดขึ้นได้

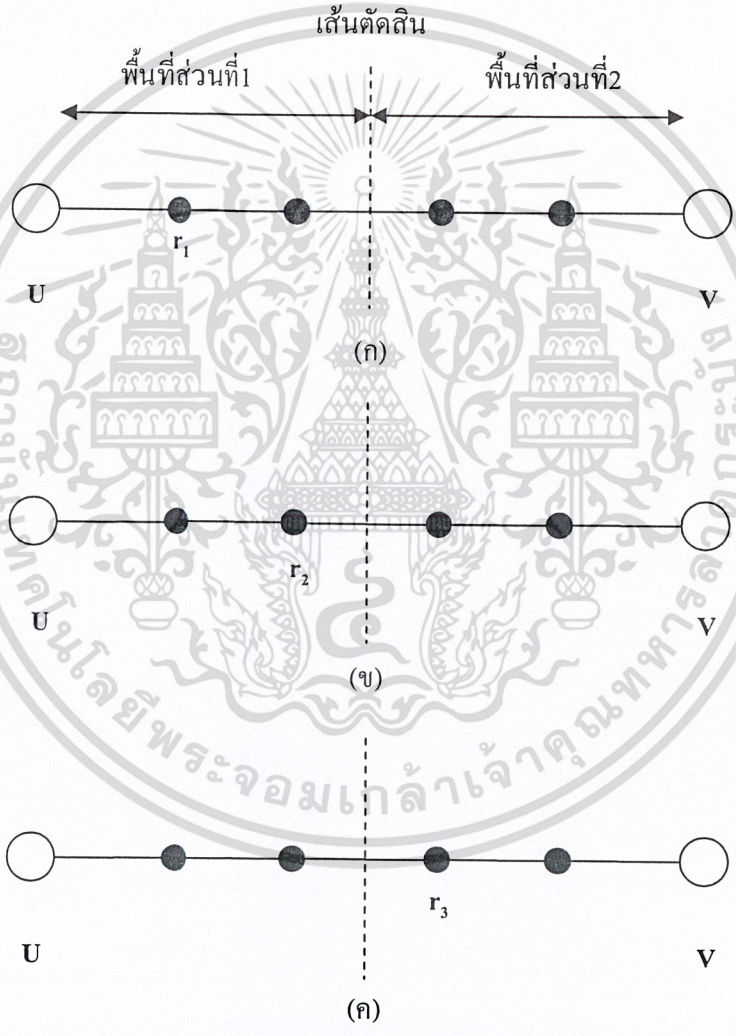
จะเห็นได้ว่าความสามารถในการตรวจสอบและแก้ไขความผิดพลาดของการเข้ารหัส จะมีความสัมพันธ์กับระยะความแตกต่างต่ำสุดระหว่างรหัสเวกเตอร์ทั้งสอง จากรูป 2.3 หากเวกเตอร์ด้านรับ r อยู่ในพื้นที่ส่วนที่ 1 จะทำการเลือกสัญญาณ U เป็นสัญญาณด้านส่ง หากเวกเตอร์ด้านรับ r อยู่ในพื้นที่ส่วนที่ 2 จะทำการเลือกสัญญาณ V เป็นสัญญาณด้านส่ง จะพบว่าเวกเตอร์ U และ V จะมีระยะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแตกต่างต่ำสุด (d_{min}) เท่ากับ 5 ซึ่งจะสามารถแก้ไขความผิดพลาดได้ 2 ตำแหน่ง ดังนั้น ความสามารถในการแก้ไขความผิดพลาดของรหัสจะมีค่าเท่ากับ

$$t = \left[\frac{d\ \min - 1}{2} \right] \qquad (2.1)$$

โดย [x] หมายถึง ค่าจำนวนเต็มที่มากที่สุดแต่มีค่าเกิน x



รูปที่ 2.3 การตรวจสอบและแก้ไขความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 รหัสวงเวียน

รหัสวงเวียนเป็นซับคลาส (Subclass) หนึ่งของรหัสลิเนียร์บล็อก รหัสลิเนียร์บล็อกมีขั้นตอนในการเลือกเมตริกซ์ตัวกำเนิดสำหรับรหัสที่แก้ไขบิตที่ผิดเพียง 1 บิตได้ง่าย แต่ขั้นตอนนี้ไม่สะดวกที่จะนำมาใช้ในการสร้างรหัสที่แก้ไขบิตที่ผิดมากกว่า 1 บิต รหัสวงเวียนจึงถูกนำมาใช้อย่างกว้างขวางเพราะมีโครงสร้างทางคณิตศาสตร์ที่เกื้ออำนวยให้สามารถออกแบบรหัสที่แก้ไขบิตที่ผิดมากกว่า 1 บิตได้ และการเข้ารหัส และการเข้ารหัสและการคำนวณซินโดรมก็สามารถสร้างขึ้นมาได้ง่ายๆ โดยใช้ตัวชิพรีจิสเตอร์ (Shift register) ได้

รหัสวงเวียนสร้างคำรหัสโดยเลื่อนบิตข้อมูลผ่านตัวชิพรีจิสเตอร์ เช่น

$$c = (c_0, c_1, c_2, ..., c_{n-1})$$

เป็นรหัสเวกเตอร์ของรหัสวงเวียน ถ้าเลื่อนรหัสเวกเตอร์ c เชิงวงเวียนไปทางขวา i ตำแหน่ง รหัสเวกเตอร์ที่ได้ก็ยังคงเป็นรหัสเวกเตอร์ของรหัสวงเวียนอีก ดังนี้

$$c^{(i)} = (c_{n-i}, c_{n-i+1}, ..., c_0, c_1, ..., c_{n-i-1})$$

คุณสมบัติของรหัสวงเวียนข้อนี้ ทำให้เราสามารถเขียนรหัสวงเวียนในรูปของพหุนาม (Polynomial) ที่มีกำลัง (Degree) n-1 ได้

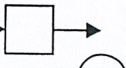
$$c(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{n-1} x^{n-1} \qquad (2.2)$$

สัมประสิทธิ์ของพหุนามจึงเป็น '0' หรือ '1' และเป็นไปตามกฎของผลบวกและผลคูณของเลขฐานสอง ดังนี้

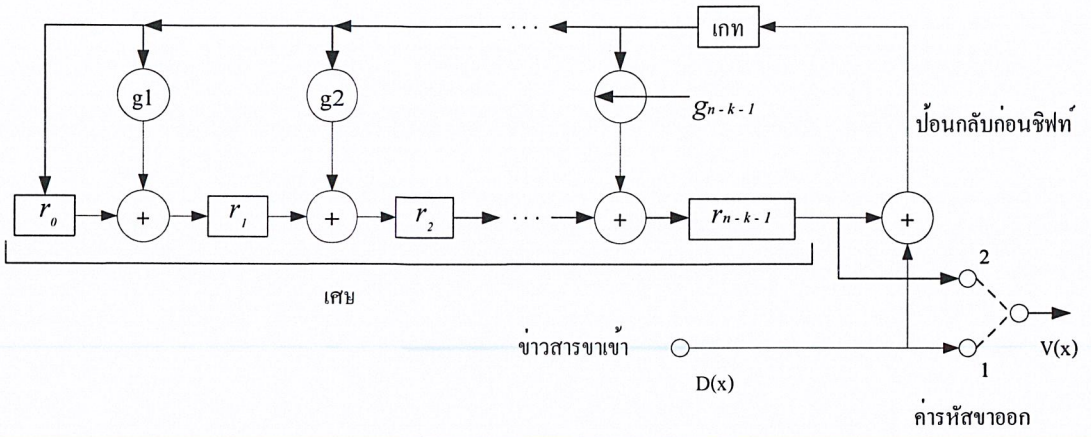
0	+ 0	= 0	0.0	= 0
0	+ 1	= 1	0.1	= 0
1	+ 0	= 1	1.0	= 0
1	+ 1	= 0	1.1	= 1

1) การเข้ารหัสโดยใช้ (n-k) บิตชิพรีจิสเตอร์

ข้อดีของรหัสวงเวียนก็คือ วงจรเข้ารหัสและถอดรหัสสามารถใช้ชิ้นส่วนอิเล็กทรอนิกส์ง่ายๆ เช่น รีจิสเตอร์และตัวบวกเลขฐาน 2 (modulo 2 adder)

ถ้าเราให้  แทนฟลิปฟลอป (Flip flop) ที่อยู่ในชิพรีจิสเตอร์ แทนตัวบวกเลขฐาน 2 และ แทนวงจรปิดเมื่อ $g_i = 1$ และวงจรเปิดเมื่อ $g_i = 0$ ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 เครื่องเข้ารหัสวงเวียน (n-k) ที่เกิดจาก $g(x) = 1 + g_1 x + g_2 x^2 + \ldots + g_{n-k-1} x^{n-k-1} + x^{n-k}$

2) การถอดรหัสวงเวียน

สมมติว่ารหัสเวกเตอร์ V ถูกส่งผ่านช่องสื่อสารที่มีเสียงรบกวน เวกเตอร์ที่รับได้ R อาจจะไม่ตรงกับรหัสเวกเตอร์ หน้าที่ของเครื่องถอดรหัสวงเวียนคือ หาเวกเตอร์ที่ส่งจากเวกเตอร์ที่รับได้

ก่อนอื่นเครื่องถอดรหัสจะทดสอบเวกเตอร์ที่รับได้ว่าเป็นรหัสเวกเตอร์ หรือไม่โดยการคำนวณซินโดรมของคำที่รับได้ ถ้าซินโดรมเท่ากับศูนย์ เวกเตอร์ที่รับได้จะหารลงตัวโดยพหุนามตัวกำเนิด ของเวกเตอร์ที่รับได้คือ รหัสเวกเตอร์นั่นเอง เครื่องถอดรหัสจะยอมรับเวกเตอร์ที่รับได้เป็นรหัสเวกเตอร์ที่ส่งมา ถ้าซินโดรมไม่เท่ากับศูนย์แสดงว่าเกิดความผิดพลาดในการส่งซินโดรม S(x) ของเวกเตอร์ที่รับได้ R(x) คือเศษที่เกิดจากการหาร R(x) โดย g(x) ดังนี้

$$\frac{R(x)}{g(x)} = P(x) + \frac{S(x)}{g(x)} \tag{2.3}$$

ในที่นี้ P(x) คือ ผลลัพธ์จากการหาร

ซินโดรม S(x) เป็น พหุนามที่มีกำลัง n-k-1 หรือน้อยกว่า ให้ E(x) เป็น แพตเทิร์น ความผิดพลาดที่เกิดจากช่องสื่อสาร จะได้

$$R(x) = V(x) + E(x)$$

และ

$$\frac{R(x)}{g(x)} = \frac{V(x)}{g(x)} + \frac{E(x)}{g(x)} \tag{2.4}$$

แต่ V(x) = D(x) g(x) ในที่นี้ D(x) คือ พหุนามข้อมูล ฉะนั้นจากสมการ (2.3)และ (2.4) จะได้

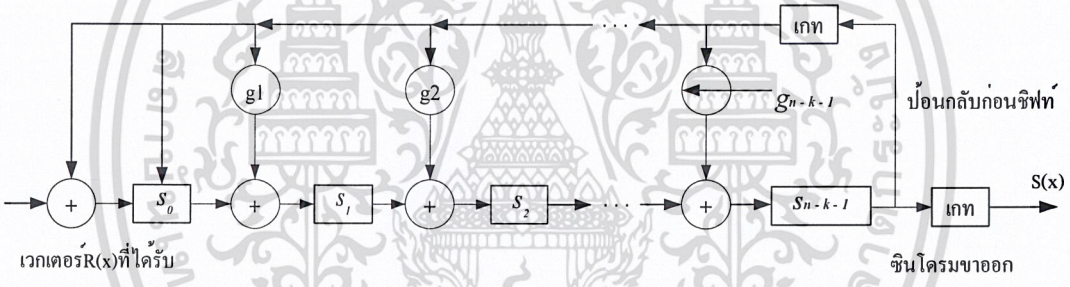
$$E(x) = [P(x) + D(x)]g(x) + S(x) \tag{2.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S(x) = \mathrm{Re}\, m \frac{E(x)}{g(x)} \tag{2.6}$$

ฉะนั้นซินโดรม R(x) เท่ากับเศษที่เกิดจากการหาร แพตเทิร์นความผิดพลาดโดยพหุนามตัวกำเนิด และซินโดรมจะมีข่าวสารเกี่ยวกับ แพตเทิร์น ความผิดพลาดที่สามารถใช้ในการแก้ไขความผิดพลาดวงจรการหารที่ใช้ในการคำนวณซินโดรมนั้นแสดง ในรูปที่ 2.5 ซึ่งคล้ายกับวงจรการเข้ารหัสในรูปที่ 2.4

รหัสเวกเตอร์ที่รับได้ R ได้จากคำรหัสใดคำรหัสหนึ่งในจำนวน 2^k คำและความผิดพลาด E เช่น ถ้า R = 0110010 ซึ่งอาจมาจาก C = 1110010 และ E = 1000000 หรือ C = 1101000 และ E = 1011010 หรือ คำรหัสที่เหลือและความผิดพลาดตัวที่เหมาะสมก็ได้ อย่างไรก็ตาม แพตเทิร์นความผิดพลาดที่มีโอกาสมากที่สุดคือ แพตเทิร์น ความผิดพลาดที่มีน้ำหนักน้อยที่สุด ดังนั้นในที่นี้การถอดรหัสที่ถูกควรเป็น C = 1110010 และ E = 1000000 จึงเป็นการสะดวกในการสร้างตารางถอดรหัส (Decoding table) เพื่อเรียงซินโดรมที่ตรงกับความผิดพลาดที่จะแก้ไขทั้งหมดไว้



รูปที่ 2.5 แผนภาพของวงจรคำนวณซินโดรม (n-k) สำหรับรหัสวงเวียน

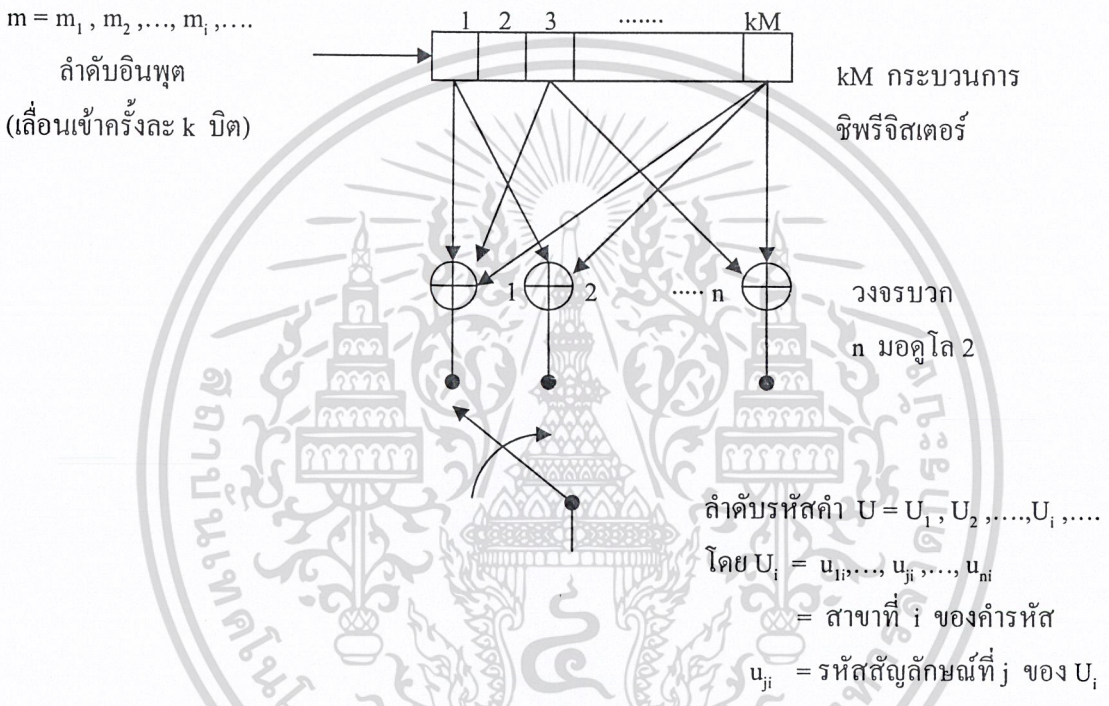
2.1.3 คอนโวลูชั่นโค้ด

การเข้ารหัสช่องสัญญาณแบ่งเป็น 2 ชนิด คือ แบบลิเนียร์บล็อกและแบบคอนโวลูชั่นโค้ดโดยรหัสลิเนียร์บล็อก ตัวแปรที่สำคัญคือ จำนวนบิตของข้อมูลอินพุต k และจำนวนของบิตคำรหัส n โดยอัตราส่วนของตัวแปรทั้งสองเรียก อัตราการเข้ารหัส ในส่วนของคอนโวลูชั่นโค้ดจะอธิบายอยู่ในรูปของตัวแปร 3 ตัว คือ n k และ M โดยอัตราส่วนของ k / n ยังคงเรียกว่า อัตราการเข้ารหัส ส่วนตัวแปร M จะเรียกว่า ความยาวคอนสเตร็นท์ (Constraint length) ซึ่งหมายถึง จำนวนของขั้นตอนกระบวนการในการเข้ารหัสของชิพรีจิสเตอร์ สิ่งที่แตกต่างกันระหว่างรหัสลิเนียร์บล็อกและคอนโวลูชั่นโค้ดที่สำคัญ คือ การเข้ารหัสแบบคอนโวลูชั่นโค้ดจะมีหน่วยความจำ โดยที่จำนวน n บิตเอาท์พุตของคอนโวลูชั่นโค้ดไม่เพียงขึ้นอยู่กับอินพุต k เท่านั้น แต่ยังขึ้นอยู่กับอินพุตก่อนหน้านั้น M – 1 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) หลักการเข้ารหัสแบบคอนโวลูชั่นโค้ด

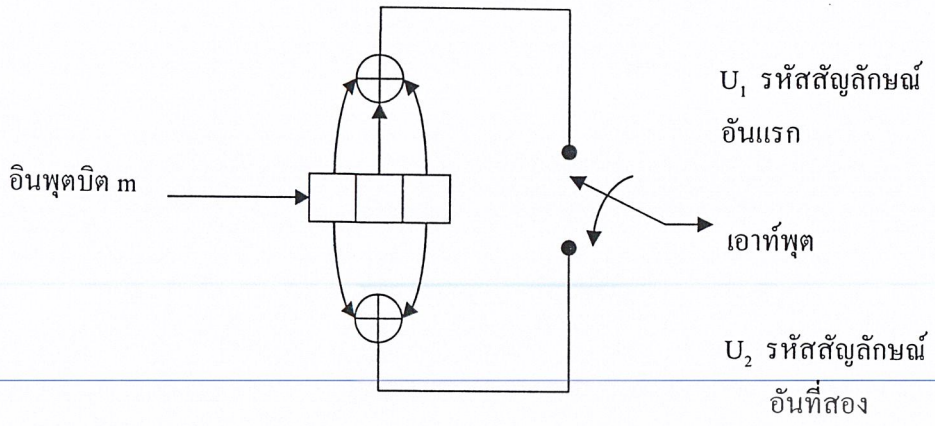
การเข้ารหัสโดยทั่วไปของคอนโวลูชั่นโค้ดแสดงได้ดังรูป 2.6 โดยมีชิพรีจิสเตอร์ kM ขั้น และมีวงจรบวกกันแบบมอดูโล 2 (modulo – 2) n ตัว โดยที่ M เป็นความยาวคอนสเตร็นท์โดยแต่ละครั้งที่อินพุต k บิตถูกเลื่อนเข้ามายังรีจิสเตอร์ k ขั้น บิตที่อยู่ในชิพรีจิสเตอร์จะถูกเลื่อนไปทางขวา และเอาท์พุตของวงจรบวกจะถูกเลือกเพื่อเป็นรหัสสัญลักษณ์ (Code symbols) สัญญาณที่ได้จะผ่านวงจรมอดูเลตเพื่อเปลี่ยนเป็นรูปคลื่นเพื่อที่จะทำการส่งออกไปตามช่องสัญญาณ



รูปที่ 2.6 การเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยความยาวคอนสเตร็นท์ M และอัตราการเข้ารหัส k / n

ในที่นี้จะทำการพิจารณาที่อินพุตบิตถูกชิพเข้าไปทีละ 1 บิต และเอาท์พุตที่ได้แต่ละข่าวสารจะมีความยาว n บิต ดังนั้นอัตราการเข้ารหัสมีค่า 1/n โดย n คือ รหัสสัญลักษณ์ (Code symbol) ที่เกิด ณ. เวลา t_i ซึ่งประกอบด้วยคำสาขา (Branch word) ที่ i ; $U_i = u_{1i},..., u_{ji} ,..., u_{ni}$ โดย u_{ji} (j = 1,2,...,n) คือ รหัสสัญลักษณ์ที่ j ของคำสาขาที่ i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



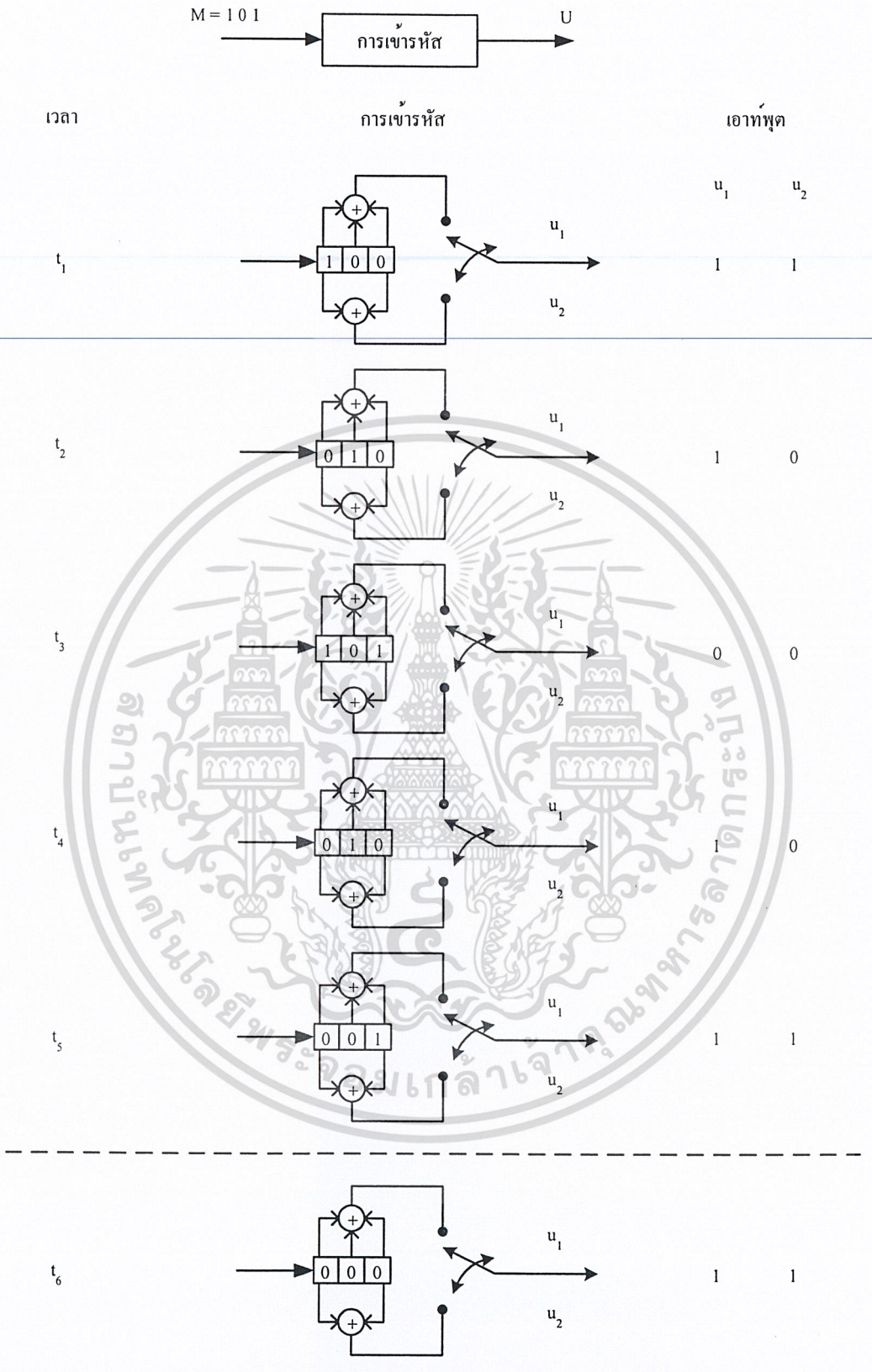
รูปที่ 2.7 การเข้าแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1 / 2 , ความยาวคอนสเตร็นท์

รูปที่ 2.7 แสดงถึงแบบจำลองของการเข้ารหัสแบบคอนโวลูชั่นโค้ด โดยมีค่าคอนสเตร็นท์ M = 3 เป็นรหัส (1,2) คือ อินพุต 1 บิต ให้เอาท์พุต 2 บิต โดยแต่ละช่วงเวลาอินพุตแต่ละบิตจะถูกชิพเข้าไปในรีจิสเตอร์ ซึ่งเอาท์พุตได้จากสวิตซ์สุ่มเลือกจากวงจรบวกมอดูโล 2 แต่ละตัว (เช่น ในตอนแรกเลือกจากวงจรบวกตัวบน แล้วจึงเลือกจากวงจรบวกตัวล่าง) จะสังเกตเห็นว่านอกจากอินพุตแล้ว การเชื่อมต่อระหว่างตำแหน่งของรีจิสเตอร์กับวงจรบวกด้วยเวกเตอร์ โดยเวกเตอร์ดังกล่าวจะมีขนาดเท่ากับค่าคอนสเตร็นท์ โดยจะมีค่าเป็น 1 หากมีการเชื่อมต่อ และจะมีค่าเป็น 0 เมื่อไม่มีการเชื่อมต่อโดยจากรูปที่ 2.7 สามารถเขียนการเชื่อมต่อเป็นเวกเตอร์ g_1 สำหรับการเชื่อมต่อด้านบน และ g_2 สำหรับการเชื่อมต่อด้านล่าง โดย

$$g_1 = 1\ 1\ 1$$
$$g_2 = 1\ 0\ 1$$

หากสมมติว่าข้อมูลอินพุต m = 1 0 1 ถูกทำการเข้ารหัสแบบคอนโวลูชั่นโค้ด โดยตัวเข้ารหัสแสดงได้ดังรูป 2.7 โดยอินพุตบิต 3 บิต ถูกชิพเข้าไปในเวลา t_1 , t_2 , และ t_3 ซึ่งผลที่ได้แสดงได้ดังรูปที่ 2.8 ส่วนเวลา t_4 และ t_5 เป็นการชิพบิต 0 เข้าไป เพื่อให้บิตข้อมูลบิตสุดท้ายถูกชิพไปจนถึงรีจิสเตอร์ตัวสุดท้าย ส่วนเวลา t_6 เป็นการทำให้รีจิสเตอร์มีค่ากลับเป็น 0 ทั้งหมด โดยเอาท์พุตที่ได้จะมีค่าเท่ากับ 1 1 1 0 0 0 1 0 1 1 โดยสัญลักษณ์ที่อยู่ซ้ายมือสุดจะถูกส่งออกไปก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลำดับเอาท์พุต : 11 10 00 10 11

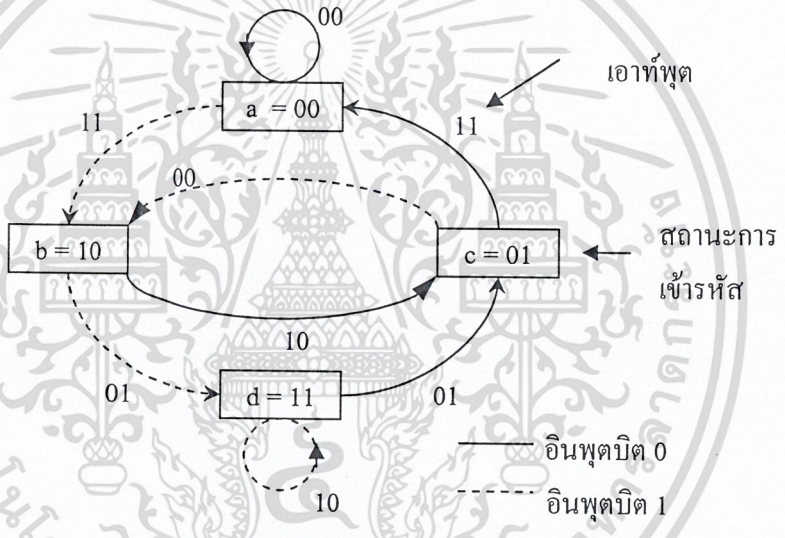
รูปที่ 2.8 ขั้นตอนการเข้ารหัสแบบคอนโวลูชั่นโค้ด อัตราการเข้ารหัส ½ ความยาวคอนสเตร์นท์ M = 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการอธิบายการเข้ารหัสแบบคอนโวลูชั่นโค้ดโดยทั่วไปสามารถอธิบายได้ 3 วิธี คือ แผนภาพต้นไม้ (Tree diagram) แผนภาพ เทลลิส (Trellis diagram) และแผนภาพสเต็จ (State diagram) ซึ่งจะกล่าวในหัวข้อต่อไป

2) แผนภาพสเต็จ

ในการเข้ารหัสแบบคอนโวลูชั่นโค้ด จะเกิดสถานะของรีจิสเตอร์ซึ่งมีผลกับสัญญาณเอาท์พุต $M-1$ สถานะ โดยเมื่อนำสถานะดังกล่าวมาเขียนเป็นแผนภาพบล็อก และเชื่อมต่อแต่ละสถานะเข้าด้วยกันจะได้แผนภาพสเต็จ โดยเส้นทางที่เชื่อมต่อระหว่างสถานะต่างๆจะหมายถึง เอาท์พุตของคำสาขา ซึ่งเป็นผลมากจากการเปลี่ยนจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง ซึ่งสามารถแสดงได้ดังรูป 2.9 ซึ่งสถานะในรีจิสเตอร์ประกอบด้วย a = 00, b = 10, c = 01 และ d = 11 ในการเปลี่ยนสถานะนั้นสามารถทำได้จากอินพุตที่เข้ามา ในกรณีที่เป็นเส้นปะ หมายถึง อินพุต 0 ส่วนที่เป็นเส้นทึบ หมายถึง อินพุต 1

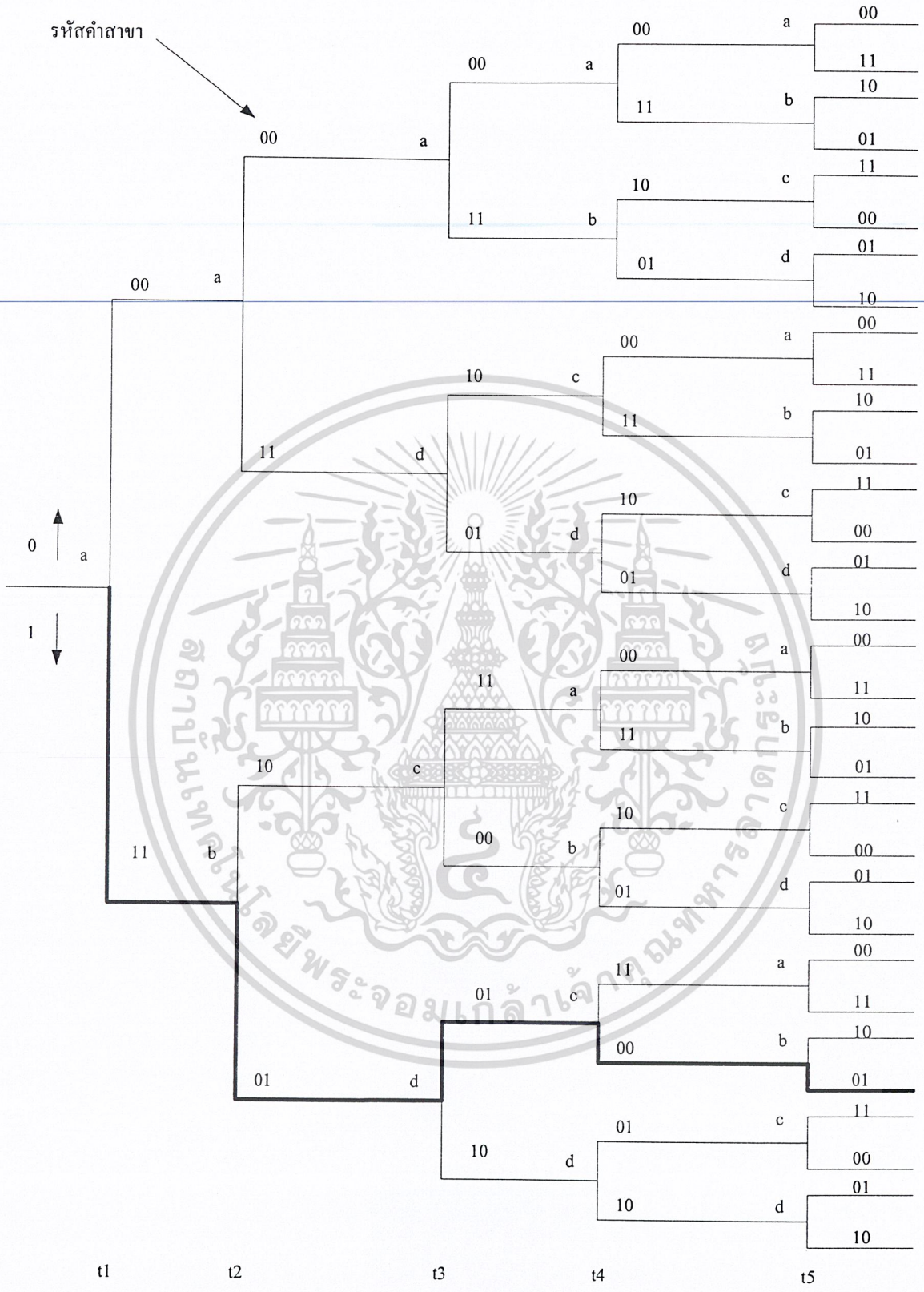


รูปที่ 2.9 แผนภาพสเต็จของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2 และ M = 3

3) แผนภาพต้นไม้

ถึงแม้ว่าแผนภาพสเต็จจะอธิบายถึงการเข้ารหัสแบบคอนโวลูชั่นโค้ดได้ดี แต่เนื่องจากแผนภาพสเต็จไม่สามารถอธิบายสถานะในแต่ละเวลาที่เปลี่ยนไป ดังนั้นจึงมีการนำเสนอแผนภาพต้นไม้ซึ่งแผนภาพต้นไม้แสดงดังรูป 2.10 โดยเมื่ออินพุตมีค่าเป็น 0 สาขาของคำจะอยู่ที่สาขาทางขวามือด้านบน แต่เมื่ออินพุตมีค่าเป็น 1 สาขาของคำจะอยู่ที่สาขาทางขวามือด้านล่าง โดยสมมติว่าในสภาวะเริ่มต้นมีค่าเป็น 0 หมด เมื่ออินพุตมีค่าเป็น 0 เข้ามา เอาท์พุตคำสาขาที่ได้มีค่าเป็น 00 แต่หากอินพุตมีค่าเป็น 1 เอาท์พุตคำสาขาจะมีค่าเป็น 11 จากรูป 2.10 แสดงอินพุตบิตมีค่า 1 1 0 1 1 โดยเอาท์พุตที่ได้แสดงได้ดังเส้นทึบ ซึ่งทำให้ลำดับรหัสคำมีค่าเป็น 1 1 0 1 0 1 0 0 0 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แผนภาพต้นไม้ของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2 และ M = 3

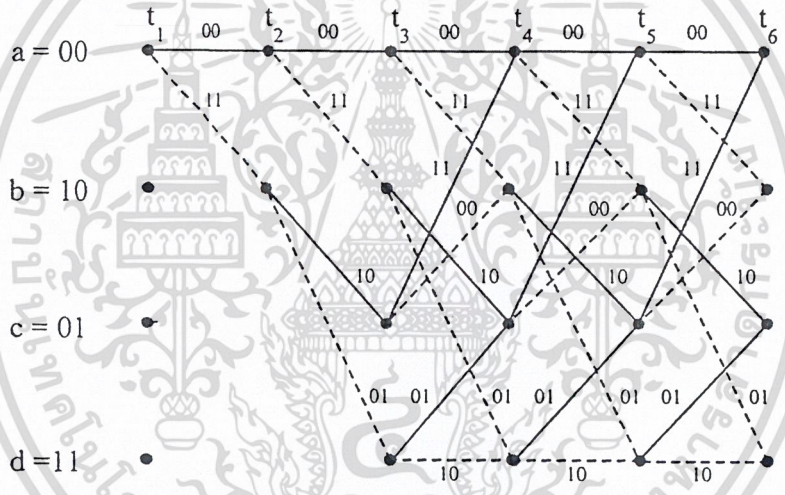
เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อย่างไรก็ตามแผนภาพต้นไม้นั้นมีปัญหาตรงที่ไม่สามารถอธิบายลำดับที่มีความยาวมากๆ โดยจำนวนของสาขาจะมีค่าเพิ่มขึ้นด้วยฟังก์ชัน 2^n โดย n คือจำนวนบิตของลำดับอินพุต

4) แผนภาพเทลลิส

จากรูป 2.10 จะพบว่าเมื่อเวลา t_4 โครงสร้างจะซ้ำแบบเดิม นั่นคือ โครงสร้างจะซ้ำเดิมหลังจากสาขาที่ 3 (โดยทั่วไปเป็นหลังจากสาขาที่ M โดย M เป็นความยาวคอนสเตร็นท์) โดยแต่ละโหนดของแผนภาพต้นไม้จะสอดคล้องกับสถานะต่างๆ 4 สถานะในชิฟรีจิสเตอร์ คือ a = 00, b = 10, c = 01 และ d = 11 โดยที่เวลา t_1 สาขาแรกจะทำให้เกิดโหนด 2 โหนด คือ a และ b และแต่ละสาขาต่อไปก็จะเกิดโหนด 2 โหนดเช่นเดียวกัน โดยที่สาขาที่ 2 ณ. เวลา t_2 ทำให้เกิดโหนด a, b, c, และ d และสาขาที่ 3 จะมีทั้งหมด 8 โหนด โดยเป็นโหนด a, b, c, และ d อย่างละ 2 โหนด จะเห็นได้ว่าสาขาที่เกิดขึ้นจะมีสถานะซ้ำแบบเดิมทั้งครึ่งบนและครึ่งล่างของแผนภาพต้นไม้จะมีค่าเหมือนกัน ดังนั้นเมื่อนำมาเขียนแผนภาพใหม่ ซึ่งแสดงได้รูป 2.11 จะเรียกแผนภาพนี้ว่า แผนภาพเทลลิส



รูปที่ 2.11 แผนภาพเทลลิสของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2 และ M = 3

เช่นเดียวกันกับแผนภาพสเต็จ เส้นทึบหมายถึง เอาท์พุตที่เกิดจากอินพุตบิตที่มีค่า 0 และเส้นประหมายถึง เอาท์พุตที่เกิดจากอินพุตบิตที่มีค่า 1 โดย เทลลิส จะมีโหนดทั้งหมด 2^{M-1} โหนด แต่ละโหนดของเทลลิส หมายถึง สถานต่างๆ เช่น a = 00, b = 10, c = 01 และ d = 11 จากรูปที่ 2.11 แผนภาพเทลลิส จะมีโครงสร้างที่ซ้ำแบบเดิมหลังจากผ่านไป 3 สาขา นั่นคือ ที่เวลา t_4 เอาท์พุตในแต่ละคำสาขา คือ การเปลี่ยนสถานะจากสถานะหนึ่งเป็นอีกสถานะหนึ่งซึ่งแสดงได้ตามสาขาของ เทลลิส ตามรูป 2.11

5) การถอดรหัสการประสานด้วยวิธีไวตาบิ

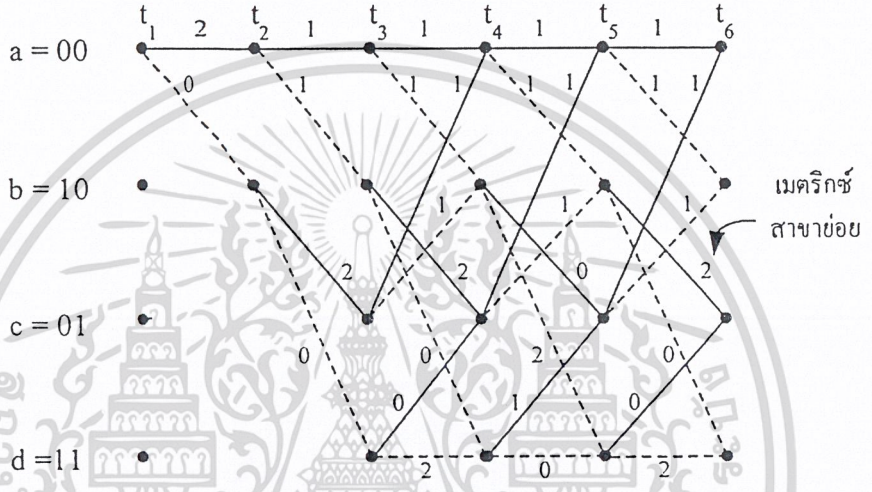
การถอดรหัสด้วยวิธีไวตาบิ จะทำการเปรียบเทียบเส้นทางที่แตกต่างกัน 2 เส้นทางที่ให้สถานะของรีจิสเตอร์เดียวกัน โดยเส้นทางที่ดีที่สุดจะถูกเลือกและจะถูกเรียกว่า เส้นทางที่เหลืออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50101

(Surviving path) จากรูปที่ 2.7 สามารถจะนำมาเขียนการถอดรหัสแผนภาพเทลลิส ได้ดังรูป 2.12 โดยหมายเลขที่เขียนบนสาขา เทลลิส ในแต่ละเวลา t_i จะเป็นระยะห่างแฮมมิ่งระหว่างรหัสสัญลักษณ์ที่รับได้กับคำสาขาที่ได้จากการเข้ารหัสเทลลิส โดยรูป 2.12 m เป็นลำดับบิตอินพุต U เป็นลำดับคำรหัสที่ส่งจากด้านส่ง และ Z แทนลำดับที่รับได้

ข้อมูลอินพุต	m:	1	1	0	1	1	...
ลำดับคำรหัสที่ส่ง	U:	11	01	01	00	01	...
ลำดับที่รับได้	Z:	11	01	01	10	01	...



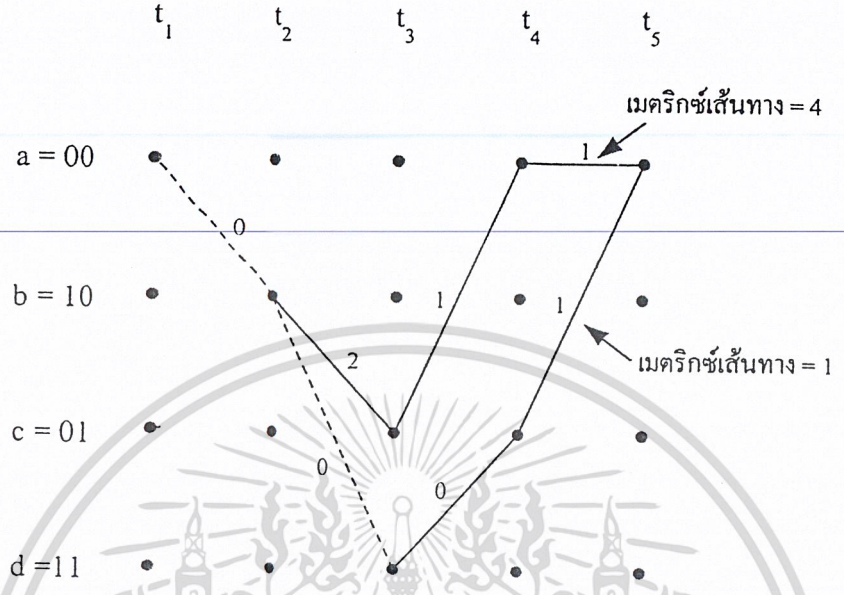
รูปที่ 2.12 การถอดรหัสโดยใช้แผนภาพเทลลิส ของการเข้ารหัสแบบคอนโวลูชั่นโค้ดด้วยอัตราการเข้ารหัส 1/2 และ M = 3

จากรูปที่ 2.12 ลำดับที่รับได้ Z ณ.เวลา t_i มีค่าเป็น 11 เมื่อเปรียบเทียบกับรูปที่ 2.11 ซึ่งเปลี่ยนจากสถานะ 00 → 00 จะได้เอาท์พุตคำสาขา 00 ซึ่งทำให้มีความแตกต่างระยะแฮมมิ่งเท่ากับ 2 และการเปลี่ยนจากสถานะ 00 → 10 ณ เวลา t_i ได้เอาท์พุตคำสาขา 11 ซึ่งจะสอดคล้องกับรหัสที่รับได้ นั่นคือ มีค่าความแตกต่างระยะแฮมมิ่งเท่ากับ 0 ซึ่งจะทำการเปรียบเทียบอย่างนี้ไปเรื่อยในแต่ละช่วงเวลา t_i ในกระบวนการถอดรหัสโดยใช้เมตริกซ์ของความแตกต่างระยะแฮมมิ่งตลอดเส้นทาง เทลลิสนี้ เพื่อจะหาเส้นทางที่น่าจะเป็นไปได้มากที่สุด หรือเส้นทางที่ให้ค่าความแตกต่างของระยะแฮมมิ่งต่ำสุด

พื้นฐานการถอดรหัสด้วยวิธีไวตาบิ คือ เมื่อมีเส้นทางสองเส้นทางของเทลลิส ที่มาพบกันที่สถานเดียวกันจะมีเส้นทางหนึ่งจะไม่ถูกนำมาพิจารณาอีก ในรูปที่ 2.13 แสดงสองเส้นทางที่มาพบกันที่เวลา t_5 ณ สถานะ 00 ซึ่งจะต้องทำการพิจารณาเมตริกซ์ความแตกต่างของระยะแฮมมิ่งโดยรวมระหว่างสองเส้นทาง ซึ่งจะเป็นผลรวมของเมตริกซ์ความแตกต่างของระยะแฮมมิ่งในแต่ละช่วงเวลา t_i จะพบว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.13 ในเส้นทางบนจะให้ผลรวมของเมตริกซ์ความแตกต่างของระยะแฮมมิ่งเท่ากับ 4 และเส้นทางล่างได้ผลรวมของเมตริกซ์ความแตกต่างของระยะแฮมมิ่งเท่ากับ 1 แสดงว่าเส้นทางด้านบนจะไม่ถูกนำมาพิจารณาอีก เนื่องจากให้ค่าเมตริกซ์ความแตกต่างของระยะแฮมมิ่งมากกว่าเส้นทางด้านล่าง



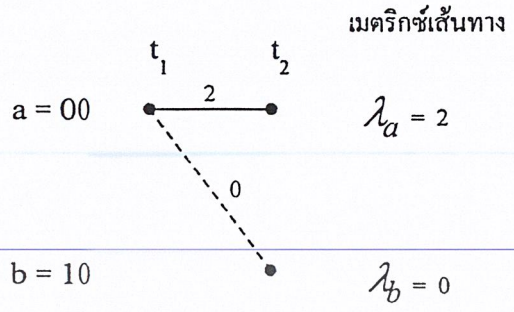
รูปที่ 2.13 เมตริกซ์เส้นทางสองเส้นทางที่มาพบกัน

ในกระบวนการถอดรหัสด้วยวิธีไวตาบิ จะทำการคำนวณเมตริกซ์ระหว่างสองเส้นทางที่มาถึงสถานะเดียวกัน จากนั้นจะทำการกำจัดเส้นทางหนึ่งทิ้งไป โดยในการคำนวณดังกล่าวจะทำในแต่ละโหนด ณ.เวลา t_i จากนั้นจึงจะถอดรหัส ณ.เวลาถัดไปซึ่งจะทำกระบวนการดังกล่าวซ้ำอีก ซึ่งรูปที่ 2.14 แสดงขั้นตอนการถอดรหัสด้วยวิธีไวตาบิ โดยสมมติให้ข้อมูลอินพุตเป็น m รหัสคำเป็น U และลำดับที่รับได้ Z เช่นเดียวกับรูปที่ 2.12 เมื่อเวลา t_1 ด้านรับรับได้ 11 จากรูปที่ 2.14ก มีเปลี่ยนจากสถานะ 00 → 00 และ 00 → 10 ซึ่งจะมีเมตริกซ์สาขาเท่ากับ 2 และ 0 ตามลำดับต่อมาที่เวลา t_2 ในแต่ละสถานะจะมีสาขาอย่างละ 2 สาขาซึ่งแสดงได้ดังรูป 2.14ข โดยผลรวมของเมตริกซ์ของแต่ละสาขามีค่าเป็น λ_a, λ_b, λ_c และ λ_d ที่เวลา t_3 ในแต่ละสถานะยังคงมีอีก 2 สาขา ซึ่งแสดงได้ดังรูป 2.14ค เป็นผลทำให้แต่ละโหนดจะมีเส้นทางที่มาถึง 2 เส้นทาง จากที่กล่าวไว้ในตอนต้น เส้นทางที่ให้ค่าผลเมตริกซ์เส้นทางมากที่สุดจะถูกกำจัดออกไป เส้นทางที่เหลืออยู่แสดงได้ดังรูป 2.14ง ซึ่งเมื่อถึงกระบวนการนี้จะพบว่าที่เวลาระหว่าง t_1 และ t_2 ตัวถอดรหัสจะเลือกเส้นทาง 00 → 10 เนื่องจากการเปลี่ยนสถานะดังกล่าวเกิดจากอินพุตที่มีค่าบิตเป็น 1 ดังนั้นตัวถอดรหัสจึงถอดรหัสได้บิตแรก คือ 1 จะพบว่าการถอดรหัสบิตแรกนั้นจะทำได้เมื่อเวลาผ่านไปช่วงเวลาหนึ่ง

ในแต่ละกระบวนการถอดรหัสแต่ละโหนดจะมีเส้นทางที่มาถึง 2 เส้นทาง ซึ่งรูปที่ 2.14จ แสดงเส้นทางที่มาถึงโหนดเมื่อเวลา t_5 ซึ่งเส้นทางหนึ่งจะถูกกำจัด เส้นทางที่เหลือแสดงได้ดังรูปที่ 2.14ฉ จะเห็นได้ว่ายังไม่สามารถที่จะถอดรหัสบิตที่สองได้ รูปที่ 2.14ช และรูปที่ 2.14ซ แสดงเส้นทางที่มาถึง

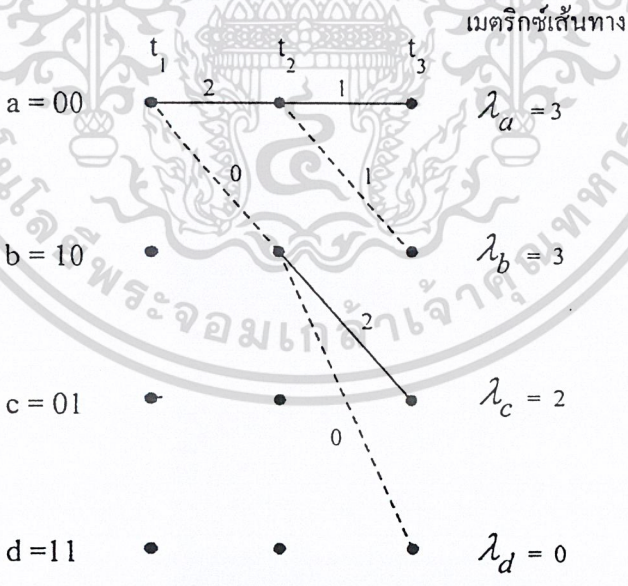
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหนดและเส้นทางที่เหลือเมื่อเวลา t_6 พบว่าสามารถทำการถอดรหัสบิตที่สองได้ ซึ่งมีค่าเป็น 1 ซึ่งจะทำการถอดรหัสในลักษณะเช่นนี้ไปจนกว่าจะถอดรหัสได้ครบ



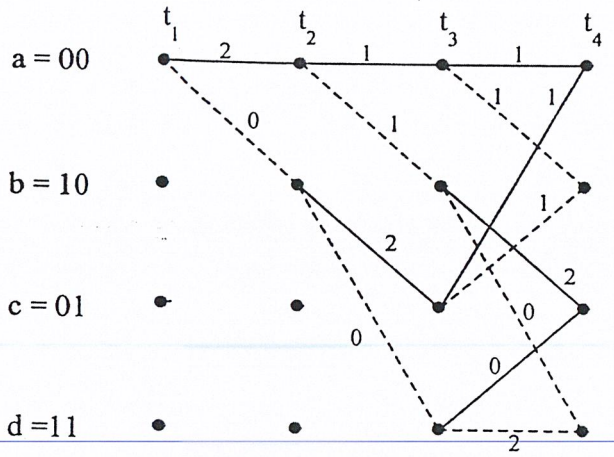
รูปที่ 2.14 การเลือกเส้นทางที่เหลืออยู่

(ก) เส้นทางที่เหลือเมื่อเวลา t_2

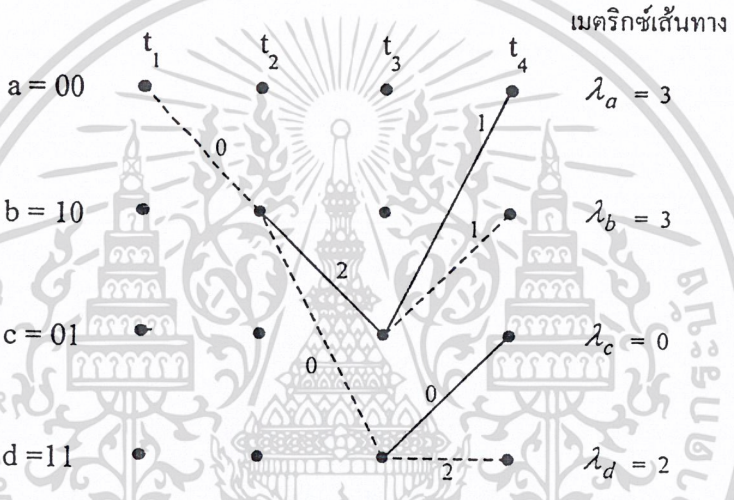


รูปที่ 2.14 (ข) เส้นทางที่เหลือเมื่อเวลา t_3

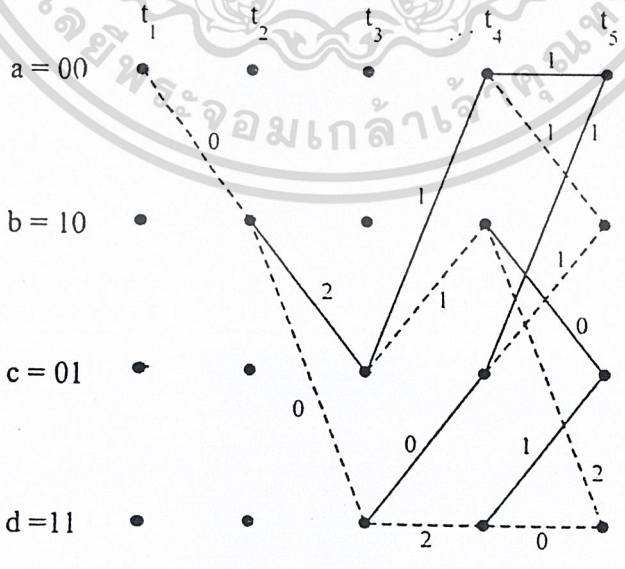
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 (ค) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_4

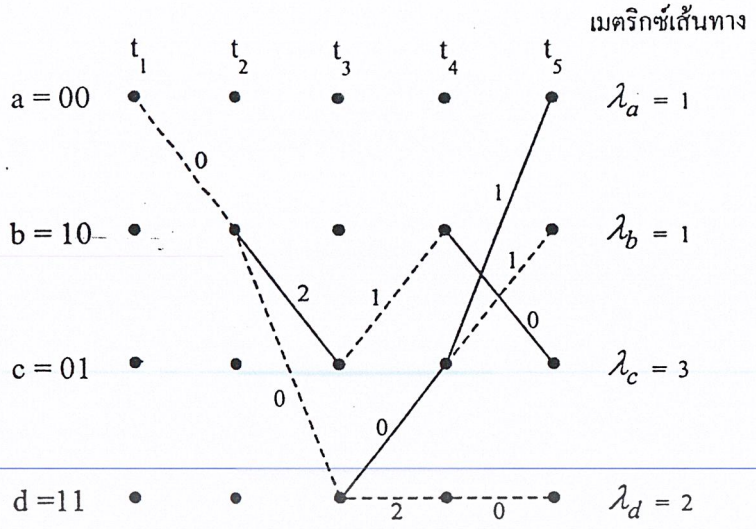


รูปที่ 2.14 (ง) เส้นทางที่เหลือเมื่อเวลา t_4

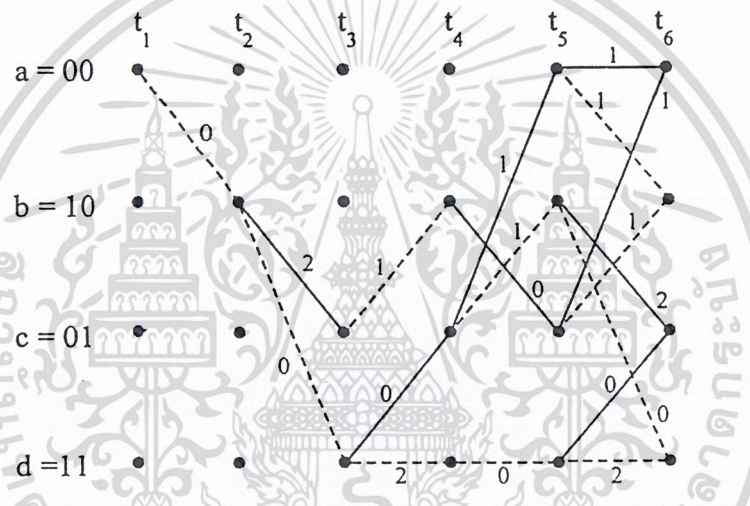


รูปที่ 2.14 (จ) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_5

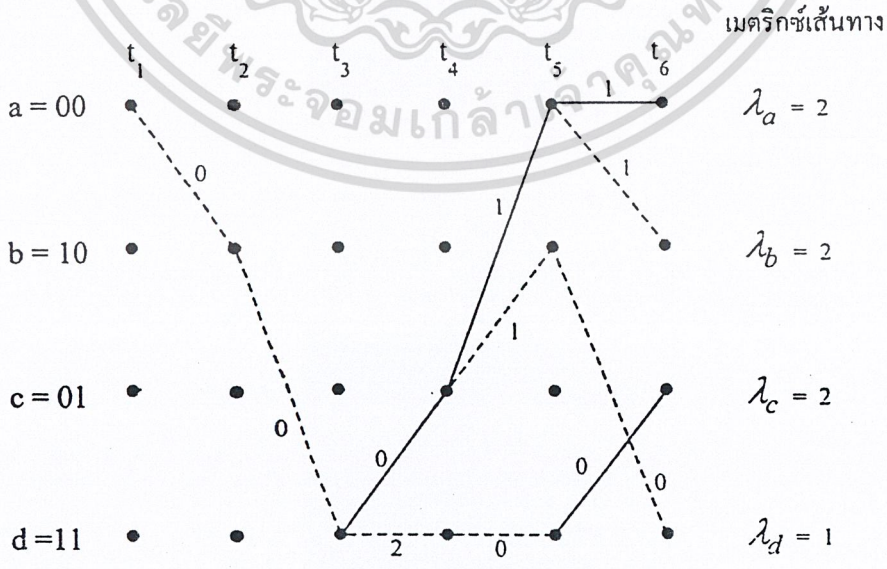
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 (ฉ) เส้นทางที่เหลือเมื่อเวลา t_5

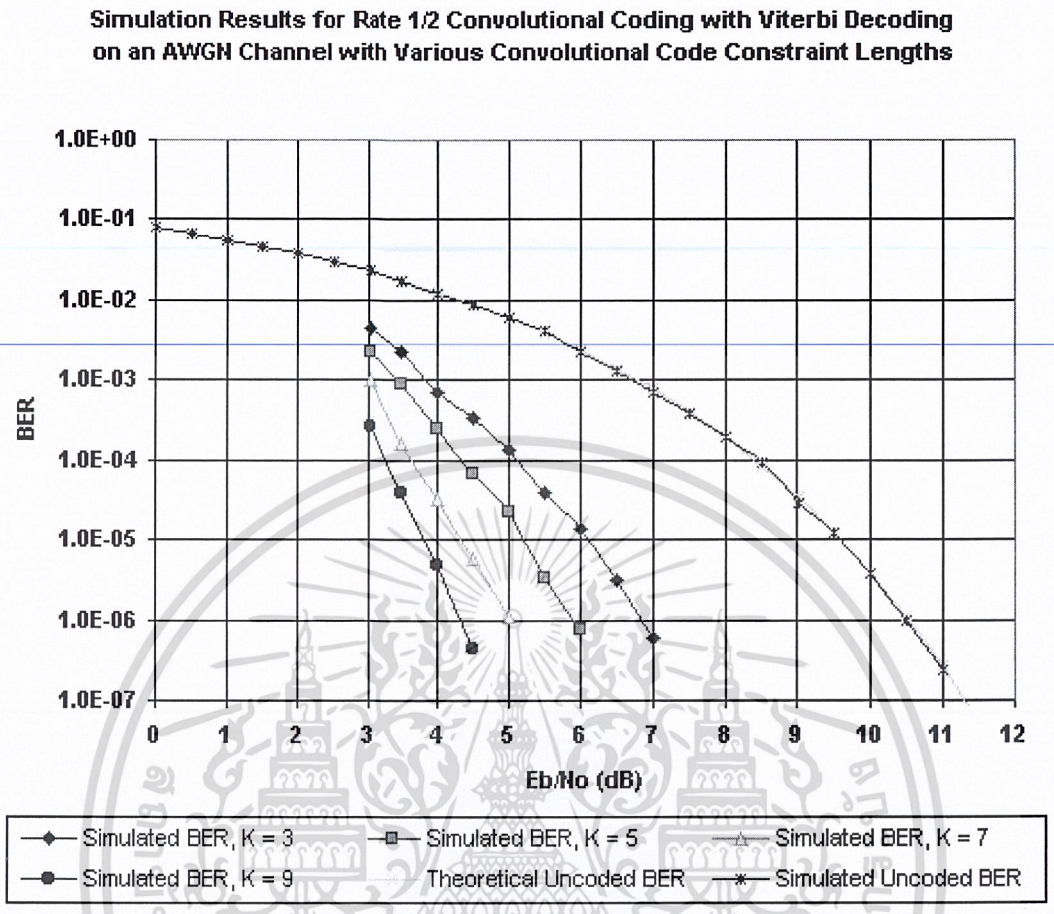


รูปที่ 2.14 (ช) การเปรียบเทียบเมตริกซ์เมื่อเวลา t_6



รูปที่ 2.14 (ซ) เส้นทางที่เหลือเมื่อเวลา t_6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงผลลัพธ์การจำลองการทำงานของการเข้ารหัสแบบคอนโวลูชั่น(อัตราการเข้ารหัสข้อมูล 1/2) ด้วยวิธีของไวตาบิดีโค้ดดิ้งบนช่องสัญญาณรบกวนขาว (AWGN : Aditive White Gaussian Noise) ในกรณีที่มีค่าความยาวของคอนสเตร็นท์ต่างกัน

จากกราฟพบว่าที่สภาวะของช่องสัญญาณรบกวนขาวค่าของความน่าจะเป็นในการเกิดความผิดพลาดของข้อมูลในลักษณะของข้อมูลที่มีการเข้ารหัสจะมีค่าน้อยกว่าลักษณะของข้อมูลที่ไม่มีการเข้ารหัสและเมื่อพิจารณาภายใต้เงื่อนไขของการเข้ารหัสที่อัตราการเข้ารหัสข้อมูลที่เท่ากันพบว่า หากมีการใช้ค่าของความยาวคอนสเตร็นท์ที่มีค่ามากขึ้นยิ่งจะทำให้ความน่าจะเป็นในการเกิดความผิดพลาดของข้อมูลลดน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

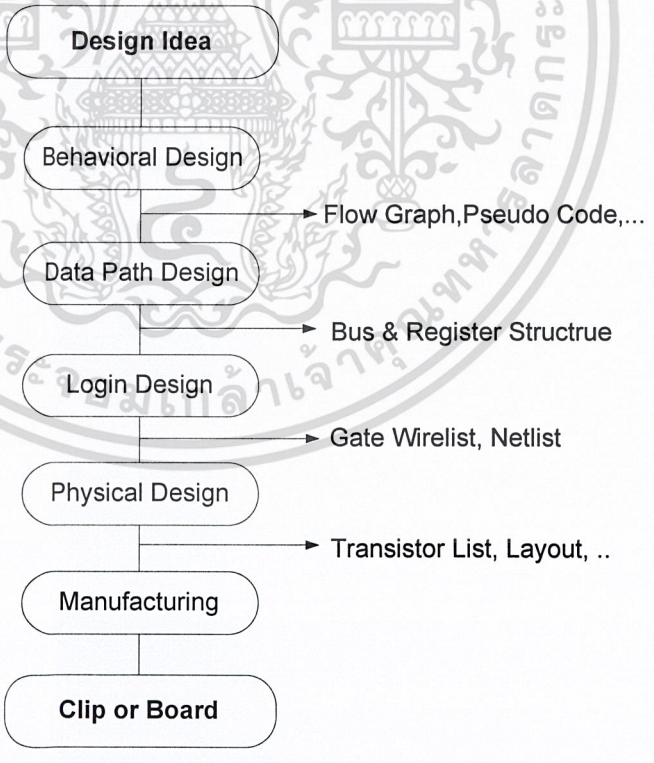
2.2 ภาษาวีเอชดีแอล

ความซับซ้อนและขนาดของระบบดิจิตอลในปัจจุบันได้เพิ่มมากขึ้นทุกขณะ ส่งผลให้มีการนำคอมพิวเตอร์เพื่อช่วยในการออกแบบ มาใช้ในกระบวนการออกแบบฮาร์ดแวร์เพิ่มขึ้นเช่นกัน อีกทั้งอุปกรณ์และวิธีการออกแบบใหม่ๆ ก็ถูกพัฒนาขึ้นมาเพื่อช่วยอำนวยความสะดวกให้กับนักออกแบบมากขึ้นด้วย สำหรับภาษาบรรยายอุปกรณ์ฮาร์ดแวร์ เอชดีแอล (HDL: Hardware Description Language) ก็เป็นเครื่องมืออย่างหนึ่งที่ได้รับการพัฒนามาอย่างต่อเนื่องเพื่อช่วยในการปรับปรุงขบวนการออกแบบระบบดิจิตอลเป็นไปอย่างมีประสิทธิภาพ

2.2.1 การออกแบบระบบดิจิตอล

ในการออกแบบระบบดิจิตอล เริ่มตั้งแต่การกำเนิดแนวความคิดเบื้องต้นจนกระทั่งได้ออกมาเป็นอุปกรณ์ฮาร์ดแวร์ที่ใช้งานได้จะต้องผ่านขั้นตนต่างๆ มากมาย และในแต่ละขั้นตอนผู้ออกแบบจะต้องตรวจสอบผลลัพธ์ในแต่ละขั้นก่อนเข้าสู่กระบวนการออกแบบในขั้นต่อไป

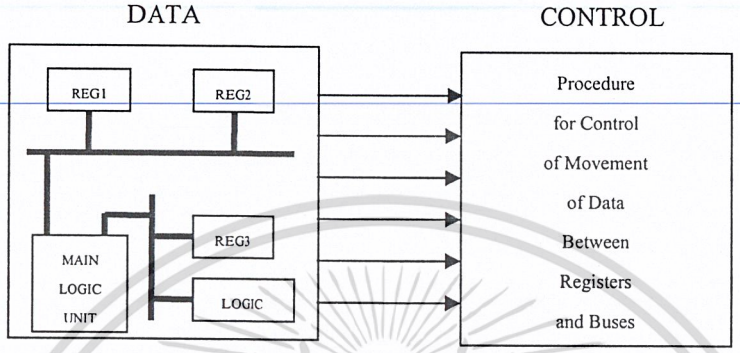
รูปที่ 2.16 แสดงขั้นตอนปกติที่ใช้ในการออกแบบระบบดิจิตอลทั่วไป ขั้นแรกผู้ออกแบบจะกำหนดแนวความคิดในการออกแบบ แล้วทำการพัฒนาให้สามารถนำไปใช้ได้อย่างสมบูรณ์ ซึ่งภายในขั้นตอนนี้ผู้ออกแบบจำเป็นต้องสร้างรูปแบบระบบเชิงพฤติกรรมขึ้นมาตรวจสอบซึ่งอาจจะเป็นผังงาน แสดงแบบหรือรหัสคำสั่งเทียม(Pseudo code) ก็ได้



รูปที่ 2.16 แสดงขั้นตอนการออกแบบระบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อไปเป็นการออแบบระบบเส้นทางของข้อมูล ผู้ออกแบบจะกำหนดส่วนประกอบของรีจิสเตอร์และวงจรลอจิกที่จำเป็นทั้งหมด เพื่อนำมาประกอบเป็นระบบที่สมบูรณ์ โดยแต่ละองค์ประกอบสามารถเชื่อมต่อกันด้วนบัสหนึ่งหรือสองทิศทาง (Unidirectional or Bidirectional Bus) กระบวนการในการควบคุมการเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์และวงจรลอจิกจะขึ้นอยู่กับพฤติกรรมของระบบที่กำหนดไว้ดังรูปที่ 2.17



รูปที่ 2.17 การออกแบบระบบเส้นทางข้อมูล

ขั้นตอนถัดมาเป็นการออกแบบวงจรลอจิกซึ่งจะเกี่ยวข้องกับการนำเกทดิจิตอลพื้นฐาน และฟลิปฟลอป (Flip – Flop) มาประกอบเป็นอุปกรณ์ย่อยต่าง ๆ เช่นรีจิสเตอร์เก็บข้อมูล บัสวงจรลอจิก และส่วนควบคุมฮาร์ดแวร์ ซึ่งผลลัพธ์ที่ได้ในขั้นตอนนี้จะเป็นเครือข่ายของการโยงใยระหว่างเกทและฟลิปฟลอปนั่นเอง

การออกแบบในขั้นตอนถัดไปเป็นการเปลี่ยนเครือข่ายการโยงใยที่ได้จากขั้นตอนที่แล้ว ให้เป็นลำดับของทรานซิสเตอร์ (Transistor List) และ โครงงาน (Layout) ซึ่งขั้นตอนนี้จะเกี่ยวข้องกันโดยตรงกับการจัดวางทรานซิสเตอร์หรือไลบรารีเซลล์เพื่อแทนเกทและฟลิปฟลอปต่าง ๆ

และในขั้นตอนสุดท้ายจะเป็นการส่งระบบที่ออกแบบไว้ไปทำการเจือที่โรงงานเพื่อผลิตออกมาเป็นวงจรรวมในที่สุด

2.2.2 ประวัติความเป็นมาของภาษาวีเอชดีแอล

วีเอชดีแอล ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และสามารถเขียนได้หลายรูปแบบด้วยเหตุผลนี้จึงทำให้ภาษาวีเอชดีแอล เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุด คือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้วิศวกรได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชั่นการทำงานของวงจรอย่างสังเขป โดยไม่คำนึงถึงรายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนี้ วีเอชดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอล ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มีความซับซ้อนได้ทั้งหมด ดังนั้น วีเอชดีแอล จึงเป็นภาษาที่น่าสนใจในการศึกษาและนำไปใช้งานเป็นอย่างยิ่ง สำหรับมาตรฐานของภาษาที่ใช้บรรยายพฤติกรรมวงจร หรือฮาร์ดแวร์ของระบบสำหรับโครงการ วีเอชเอสไอซี ที่ ดีโอดีได้ให้ไว้สามารถสรุปได้ดังนี้

- ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถเข้าใจได้ทั้งมนุษย์และเครื่องคอมพิวเตอร์โดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
- สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้
- ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร

ฉะนั้นภาษาดังกล่าวนี้จึงจัดเป็นภาษาโปรแกรมระดับสูง เช่น เดียวกับภาษาปาสคาล หรือภาษาซี ซึ่งในทางวิศวกรรมภาษาที่ใช้ในการออกแบบฮาร์ดแวร์นี้เรียกว่า ภาษาโปรแกรมระดับสูง

2.2.3 องค์ประกอบพื้นฐานต่างๆ ของภาษาวีเอชดีแอล

ในการเขียนรูปแบบภาษาบรรยายระบบดิจิตอล ในมุมมองของการออกแบบลักษณะบนลงล่าง จะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆ ของรูปแบบภาษาวีเอชดีแอลเสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียน คือ

1) หน่วยการออกแบบเอนทิตี้ (Entity Design Unit)

หน่วยการออกแบบนี้ เป็นส่วนที่ใช้สำหรับติดต่อระหว่างโลกภายนอกกับรูปแบบที่เขียนขึ้นที่เรียกว่า หน่วยการออกแบบเอนทิตี้ ในส่วนนี้ใช้กำหนดจุดเชื่อมต่อ ของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.18 แสดงให้เห็นโครงสร้างอย่างง่ายๆ ของ หน่วยการออกแบบเอนทิตี้

```
ENTITY component_name IS
        Input and output ports
        Physical and other parameter
END [component_name];
```

รูปที่2.18 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้

ส่วนนี้จะขึ้นต้นด้วยคำว่า ENTITY และ IS ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน (component_name) สำหรับการตั้งชื่อนั้นต้องเป็นไปตามกฎเกณฑ์ของภาษาหลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกข้อมูล (input – output) รวมทั้งพารามิเตอร์อื่นๆ ส่วนนี้เรียกว่าส่วนหัว (entity header) และที่สำคัญคือ หน่วยการออกแบบเอนทิตี้จะต้องปิดท้ายด้วยคำว่า END และเครื่องหมายอัฒภาคเสมอ (;)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หน่วยการออกแบบสถาปัตยกรรม (Architecture Design Unit)

คือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบ ในมุมมองของการจำลองการทำงานพฤติกรรมต่างๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออก ตรงช่องทางตลอดจนพารามิเตอร์ต่างๆ ที่กำหนดในหน่วยการออกแบบเอนทิตี้ รูปที่2.19 แสดงให้เห็นถึงโครงสร้างอย่างง่ายๆของหน่วยการออกแบบสถาปัตยกรรม

```
ARCHITECTURE identifier OF component_name IS
[declaration]
BEGIN
        specification of the functionality of the
        component in terms of its input lines and as
        influenced by physical and other parameters
END [identifier];
```

รูปที่2. 19 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรม เริ่มต้นด้วยคำว่า ARCHITECTURE และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า ARCHITECTURE นั้นใช้บรรยายหน่วยการออกแบบเอนทิตี้ใดๆ (OF < entity design unit > IS) ส่วนที่อยู่ระหว่าง ARCHITECTURE และ BEGIN เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (architecture declarative area) ที่เป็นเพียงส่วนเผื่อเลือก (option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่นประเภท (type) ต่างๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ตัวคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์(component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า และไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง PORT) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า BEGIN กับ END ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (concurrent statement) เท่านั้น หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง END และชื่อของสถาปัตยกรรมนั้นๆ ที่เป็นส่วนเผื่อเลือกโดยทั่วไปการเขียนรูปแบบระบบดิจิตอลด้วยภาษาวีเอชดีแอล สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

-ประเภทการไหลของข้อมูล (Dataflow description)

-ประเภทพฤติกรรม (Behavioral description)

-ประเภทโครงสร้าง (Structure description)

-ประเภทผสม (Mixed model description)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) หน่วยการออกแบบแพ็กเก็ต (Package Design Unit)

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อย (subprogram) ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบดิจิตอล สามารถเก็บไว้ในส่วนที่เรียกว่า package ได้ และข้อมูลเหล่านี้สามารถถูกเรียกไปใช้ได้โดย entity design unit, architecture design unit หรือจาก package design unit อื่นๆ ด้วยชุดคำสั่ง USE statement นอกจากนั้นสิ่งที่นิยมทำกันมากคือรูปแบบ (model) มาตรฐานต่างๆ อาทิเช่น standard components (model ของ IC ตระกูล 74xx)จะถูกเก็บไว้ใน แพ็กเก็ต ที่ทุกคนสามารถเข้าถึงและนำไปใช้ได้

โดยปกติแล้ว แพ็กเก็ตจะแบ่งออกเป็น 2 ส่วนคือ การประกาศแพ็กเก็ต (Package declaration) และส่วนของบอดี้แพ็กเก็ต (Package body) เนื่องจาก แพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่นำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษาวีเอชดีแอลสามารถกระทำได้ด้วยชุดคำสั่ง USE

- การประกาศแพ็กเก็ต (Package Declaration)

ส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (มองในแง่ของการนำไปใช้จากภายนอก) ได้แก่ การประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อ (identifier) ของสิ่งที่ประกาศอยู่ภายในแพ็กเก็ต สำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเอง ถ้าสิ่งใดๆ ถูกประกาศในส่วนของบอดี้แพ็กเก็ต แต่ไม่ถูกประกาศในการประกาศแพ็กเก็ตจะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้จากส่วนนอกได้ ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตี้ คือ จุดเชื่อมต่อ ที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็กเก็ตสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วน บอดี้และยังสามารถถูกนำไปใช้จากรูปแบบ (model) ภายนอกได้เช่นใช้สำหรับประกาศ ชนิด(TYPE) หรือ สัญญาณ (gobal) เช่นเดียวกันกับบอดี้แพ็กเก็ตที่ไม่จำเป็นต้องมีการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นจะไม่สามารถนำไปใช้ในรูปแบบ (model) อื่นได้ การเขียนการประกาศแพ็กเก็ต มีกฎเกณฑ์ตามที่แสดงในรูปที่2 .20

```
PACKAGE package_name IS
        package_declarative_part
END package_name ;
```

รูปที่ 2.20 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

- โครงสร้างของแพ็กเก็ต (Package Body)

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปของคำสั่งลำดับ ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ต แล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ต แต่ถูกกำหนดค่าคงที่ต่างๆ อันได้แก่ตัวค่าคงที่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต ฉะนั้น ส่วนบอดี้แพ็กเก็ต จึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเก็ต ไม่มีการประกาศชื่อ ที่เป็น โปรแกรมย่อย หรือ ค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.21

```
PACKAGE BODY package_name IS
    declarative part
END package_name ;
```

รูปที่ 2.21 โครงสร้างของบอดี้แพ็กเก็ต

4) หน่วยการออกแบบโครงแบบ (Configuration Design Unit)

รูปแบบหนึ่งของระบบดิจิตอลไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตี้ได้ เพียงหนึ่งเดียวเท่านั้น แต่ในขณะที่หน่วยการออกแบบเอนทิตี้ หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมี หน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตี้กับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

```
CONFIGURATION identifier OF entity_name IS
    configuration_declarative_part
END;
```

รูปที่ 2.22 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

5) โปรแกรมย่อย

การใช้ฟังก์ชันและโพรซีเจอร์ใน วีเอชดีแอล เปรียบได้กับการใช้โปรแกรมย่อยในการเขียนโปรแกรมภาษาขั้นสูงทั่วๆ ไปค่าที่ถูกส่งกลับหรือถูกเปลี่ยนแปลงโดยโปรแกรมย่อยอาจจะมีหรือไม่มีผลต่อฮาร์ดแวร์โดยตรงก็ได้ เช่นถ้าใช้ฟังก์ชันแทนการกระทำในสมการบูลีนก็จะมีผลต่อวงจรลอจิกจริงๆ ในขณะที่ถ้าใช้โปรแกรมย่อยในการเปลี่ยนชนิดของข้อมูลหรือในการคำนวณค่าการหน่วงเวลาแล้ว ก็จะไม่มีผลต่อโครงสร้างของฮาร์ดแวร์

6) โอเปอร์เรเตอร์

การบรรยายเชิงพฤติกรรมในภาษา วีเอชดีแอล มีตัวดำเนินการหรือโอเปอร์เรเตอร์ทางลอจิกและคณิตศาสตร์เช่นเดียว กับภาษาซอฟต์แวร์ทั่วไปดังรูปที่ 2.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PREDEFIND OPERATORS

LOGICAL OPERATORS : NOT AND OR NAND NORXOR

OPERAND TYPE : BIT BOOLEAN

RESULT TYPE : BIT BOOLEAN

RELATIONAL OPERATORS : = / = < <= > >=

OPERAND TYPE : any type

RESULT TYPE : Boolen

ARITHMETIC OPERATORS : + - * / ** MOD REM ABS

OPERAND TYPE : INTEGER REAL Physical

RESULT TYPE : INTEGER REAL Physical

CONCANTENATION OPERATOR : &

OPERAND TYPE : ARRAY of any type

RESULT TYPE : array of any type

RESULT TYPE : array of any type

รูปที่ 2.23 ตัวดำเนินการใน วีเอชดีแอล

7) เวลาและความพร้อมเพรียง

ในวงจรอิเล็กทรอนิกส์อุปกรณ์ทุกๆ ตัวจะอยู่ในสภาพเตรียมพร้อมเสมอ (Always Active) และจะมีเรื่องของเวลา เข้ามาเกี่ยวข้องในทุกๆ เหตุการณ์ที่เกิดขึ้นเสมอ วีเอชดีแอล เป็นภาษาที่ได้รับการออกแบบมาเพื่อให้สามารถบรรยายรูปแบบและการพ้องกันของเวลา สำหรับการทำงานของอุปกรณ์ได้อย่างถูกต้อง การบรรยายการทำงานที่อยู่ภายในส่วน ของการบรรยายสถาปัตยกรรม จะมีการทำงานที่พร้อมเพรียงกันเสมอ หรือแม้แต่โปรเซสซึ่งมีการทำงานภายในเป็น แบบลำดับคำสั่งก็ตาม ซึ่งหากมีหลายๆ โปรเซสอยู่ภายในโครงสร้างเดียวกัน ทุกๆ โปรเซสก็จะทำงานไปพร้อมๆ กันด้วย

8) สัญญาณและตัวแปร

สัญญาณมีลักษณะเป็นเสมือนตัวกลางฮาร์ดแวร์ ที่ใช้ในการผ่านข้อมูลและมีเรื่องของเวลามาเกี่ยวข้องด้วยการกำหนดค่าให้สัญญาณจะใช้สัญลักษณ์ <= ในการส่งค่าและสามารถใช้คำสั่ง AFTER เพื่อกำหนดช่วงเวลาในการส่งผ่านค่าของสัญญาณ เช่น <= a AFTER 12 NS หมายถึงการกำหนดค่าสัญญาณ a ให้กับ w หลังจากเวลา ผ่านไป 12 นาโนวินาที ในทางตรงข้ามตัวแปรมีลักษณะเป็นเสมือนตัวกลางที่ใช้ในการส่งผ่านข้อมูลและไม่มีเรื่องของเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งตัวแปรจะถูกใช้ในส่วนที่มีการทำงานเป็นแบบลำดับคำสั่งเช่นในฟังก์ชั่น, โพรซีเจอร์และโปรเซสสำหรับกำหนดค่าให้กับตัวแปรจะใช้สัญลักษณ์ :=

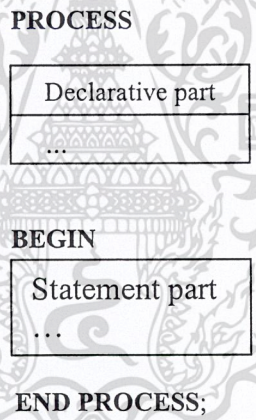
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การบรรยายเชิงพฤติกรรม

การบรรยายลักษณะการทำงานของอุปกรณ์ฮาร์ดแวร์ในเชิงพฤติกรรม เป็นการบรรยายลักษณะการเปลี่ยนแปลงของข้อมูลในรูปแบบของอัลกอริธึม สำหรับการคำนวณผลลัพธ์ที่เกิดขึ้น ซึ่งสืบเนื่องมาจากการเปลี่ยนแปลงสภาวะของข้อมูลที่เข้ามา โดยไม่คำนึงถึงลักษณะโครงสร้าง หรือความสัมพันธ์ของอุปกรณ์ที่อยู่ภายในว่าจะเป็นอย่างไร ในหัวข้อนี้จะแสดงถึงการบรรยายเชิงพฤติกรรมแทนการใช้โมดูลฮาร์ดแวร์รวมถึงข้อกำหนดต่างๆ ที่ควรรู้

2.2.5 การโปรเซส

โปรเซสเป็นรูปแบบพื้นฐานอย่างหนึ่งที่ใช้ในการกำหนดให้กับสัญญาณ โปรเซสจะอยู่ในสถานะที่เตรียมพร้อมอยู่เสมอและจะปฏิบัติคำสั่งพร้อมๆ กันกับโปรเซสอื่นๆ ที่อยู่ในสถาปัตยกรรมบรรยายเดียวกัน โดยโปรเซสจะปฏิบัติงานตามคำสั่งทันทีที่มีเหตุการณ์เกิดขึ้นกับสัญญาณที่อยู่ทางด้านขวามือของสัญลักษณ์กำหนดค่าให้กับสัญญาณ (<=) การบรรยาย โปรเซสจะเริ่มต้นด้วยคำสั่ง PROCESS และ END PROCESS ในรูปที่ 2.24 เป็นการแสดงส่วนประกอบของการบรรยายแบบโปรเซส ซึ่งประกอบด้วยส่วนของการประกาศตัวแปรที่ต้องใช้และส่วนการปฏิบัติ คำสั่งเพื่อให้ได้ผลลัพธ์ที่ต้องการ



รูปที่ 2.24 รูปแบบของการบรรยายแบบโปรเซส

2.2.6 การกำหนดตัวดำเนินการภายในโปรเซส

ตัวดำเนินการภายในโปรเซสมี 3 ชนิดคือ ตัวแปร (Variable) ไฟล์ (File) และตัวคงที่ (Constant) ซึ่งตำดำเนิน การทั้ง 3 ชนิดนี้หากมีการประกาศไว้ในโปรเซสใดก็จะใช้ได้เฉพาะภายในโปรเซสนั้นเท่านั้นสำหรับการติดต่อภายนอก หรือระหว่างโปรเซสสามารถทำได้โดยใช้สัญญาณ (Signal) หรือตัวคงที่ที่ได้ประกาศไว้ในส่วนของ ARCHITECTURE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 การกำหนดการกระทำภายในโปรเซส

การกระทำใดๆ ภายในโปรเซสจะเป็นการปฏิบัติแบบลำดับ (Sequential) เสมอ ซึ่งภายในโปรเซสสามารถใช้ประโยคเงื่อนไขหรือการทำซ้ำได้เช่น IF-ELSE,CASE-WHEN,FOR LOOP และ WHILE LOOP

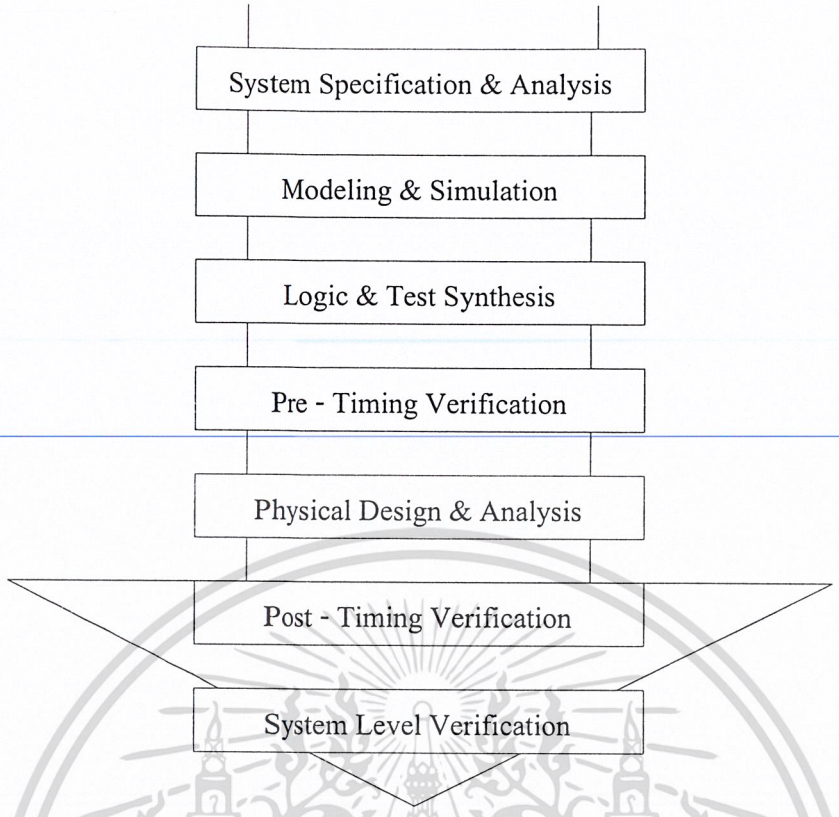
2.2.8 การกระตุ้นและยับยั้งการกระทำของโปรเซส

การกระทำภายในโปรเซสจะอยู่ในสภาพเตรียมพร้อม และมีการปฏิบัติงานอยู่ตลอดเวลาที่มีการเปลี่ยนแปลงของเหตุการณ์ เกิดขึ้น อย่างไรก็ดีเราสามารถกระตุ้นหรือยับยั้งการกระทำภายในโปรเซสได้โดยการกำหนดรายการของสัญญาณที่ต้อง การให้โปรเซสปฏิบัติงานเมื่อมีเหตุการณ์เกิดขึ้นกับสัญญาณที่กำหนดไว้เท่านั้น ส่วนเหตุการณ์ใดๆ ที่เกิดขึ้นกับสัญญาณ ที่ไม่ได้กำหนดไว้ในรายการก็จะไม่ส่งผลให้มีการกระทำภายในโปรเซส ซึ่งรายการของสัญญาณนี้เรียกว่า Sensitivity List และกำหนดไว้ภายในวงเล็บหลังคำสั่ง PROCESS

2.2.9 การออกแบบจากบนลงล่าง (Top-Down Design)

ในการพัฒนาวงจรรวมดิจิตอลขนาดใหญ่ที่มีความซับซ้อน วิศวกรหรือผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของ บล็อกไดอะแกรมก่อนที่ทำวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษาวีเอชดีแอลนั้นอนุญาตให้อธิบายและวิเคราะห์การณ์ ทำงานของแต่ละบล็อก รวมถึงการปรับปรุงการทำงานจากผลที่วิเคราะห์เพื่อให้ได้การทำงานตามต้องการนอกจากนี้ยัง สามารถเพิ่มเติมในรายละเอียดในแต่ละขั้นตอนได้ ซึ่งหลักการนี้สอดคล้องกับหลักการออกแบบจากบนลงล่างนั่นเอง ถ้าทดลองเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่า การออกแบบจากล่างขึ้นบนจะใช้เวลาการออกแบบมากกว่า 90% เนื่องจากเป็นการวางวงจรด้วยอุปกรณ์ต่างๆ (Schematic capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ ก่อนแล้วจึงทำการจำลองการทำงาน และตรวจสอบความถูกต้อง วีเอชดีแอล กับหลักการออกแบบจากบนลงล่างจึงเป็นทางออกให้กับวิศวกรให้สามารถ ออกแบบและพัฒนาวงจรที่มีความซับซ้อนได้มากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.25 ขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.25 แสดงถึงขั้นตอนการออกแบบจากบนลงล่าง ทั้งนี้ในทางปฏิบัติอาจมีข้อแตกต่างไปจากนี้บ้าง เล็กน้อยเนื่องจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้หลายเทคโนโลยี สำหรับรายละเอียดของขั้นตอน การออกแบบจากบนลงล่างในแต่ละขั้นตอนมีดังนี้

1) ความต้องการของระบบและการวิเคราะห์ คือ การสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2) รูปแบบและการจำลองการทำงาน คือ การเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษา วีเอชดีแอล หรือ ภาษา เอชดีแอล อื่นๆ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3) ลอจิกและการทดสอบการสังเคราะห์ คือ หลังจากที่ได้หลักการขั้นต้นพร้อมแนวความคิดที่ผ่านการตรวจสอบแล้วหลักการนี้จะถูกเพิ่มเติมรายละเอียดลงมาเป็นลำดับขั้นที่สอง จนกระทั้งอยู่ในระดับที่จะนำไปผลิตวงจรจริง หรือสังเคราะห์ในขั้นตอนนี้เองเทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้น และระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่จะเขียนขึ้นให้อยู่ในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้นหรือไม่ก็อยู่ในรูปของโครงข่ายการเชื่อมต่อ ที่สมารถนำไปผลิตอุปกรณ์อื่นได้

4) การตรวจสอบเวลาก่อนการออกแบบ คือ หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือโครงข่ายการเชื่อมต่อแล้ว ข้อมูลนี้จะถูกใช้สำหรับจำลองการทำงานในเรื่องความถูกต้องของฟังก์ชัน พร้อมกับนำข้อมูลที่เกี่ยวข้องกับเวลาเข้ามาประกอบการพิจารณาด้วย ซึ่งตามปกติแล้วอุปกรณ์ ทางอิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของการแพร่กระจาย (Propagation Delay Time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาทีก็ตาม แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชันต่างๆ จำนวน 10, 000 เกท ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจทำให้การทำงานของวงจรรวมทั้งหมดผิดพลาดไป หรือไม่สามารถทำ งานในย่านความถี่สัญญาณนาฬิกาที่สูงได้

5) การออกแบบทางกายภาพและการวิเคราะห์ คือ ขั้นตอนการผลิตเป็นวงจรจริง (Technology and device mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจ จะอยู่ในรูปของแผงวงจรไฟฟ้าที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้นหรืออยู่รูปของวงจรรวมเอซิก (ASIC)

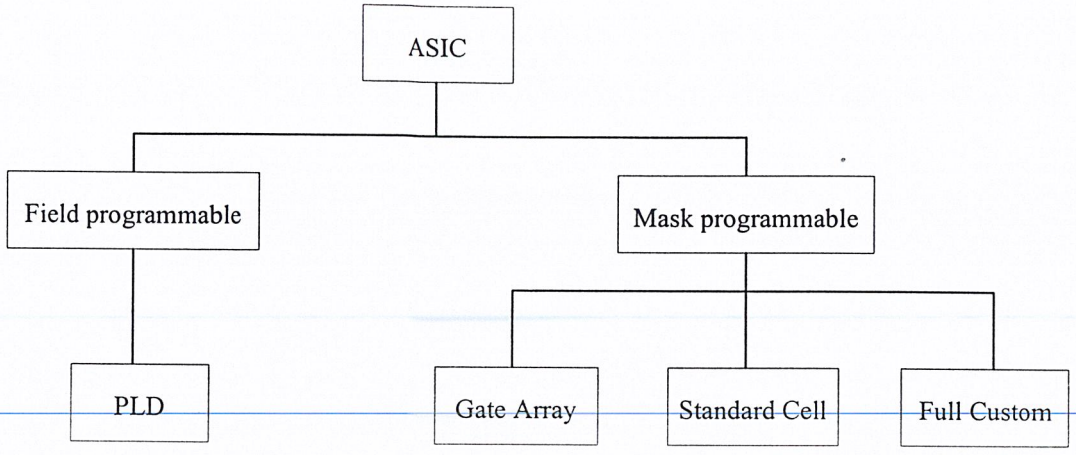
6) การตรวจสอบเวลาหลังการออกแบบ คือ การทำการตรวจสอบการทำงานด้วยตัวแปรทางด้านเวลาทั้งหมด เพื่อความถูกต้องของวงจรเป็นครั้งสุดท้ายก่อนนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เนื่องจากในขั้นตอนนี้ วงจรที่ออกแบบ จะประกอบด้วยจุดต่อทางอินพุทและเอาท์พุท ซึ่งเป็นจุดต่อสำหรับการรับและส่งสัญญาณกับภายนอก

7) การตรวจสอบระบบ คือ การนำวงจรที่ออกแบบไว้ประกอบเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบที่สมบูรณ์ แล้วทำการทดสอบการทำงานทั้งระบบรวมกับอุปกรณ์อื่นๆ อีกครั้งเพื่อควบคุมคุณภาพของผลิตภัณฑ์

2.3 เอฟพีจีเอ

เทคโนโลยีความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ในปัจจุบัน ทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ต่างๆ ซึ่งทำให้ลดค่าใช้จ่ายต่างๆได้มาก ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพและระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจากเทคโนโลยีไมโครโปรเซสเซอร์ และหน่วยความจำปัจจุบัน ทุกๆครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างระหว่างวงจรรวมและไอซีมาตรฐานมากขึ้น นักออกแบบอุปกรณ์ทางด้านดิจิตอล ได้พิจารณาถึงการผลิตให้มีขนาดมากขึ้นและการผลิตวงจรรวมเอซิก (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวมจะแบ่งตามสร้างออกเป็น 2 กลุ่ม คือ ฟิลด์โปรแกรมเมเบิล (Field programmable) และ แมสโปรแกรมเมเบิล (Mask programmable) ดังแสดงในรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



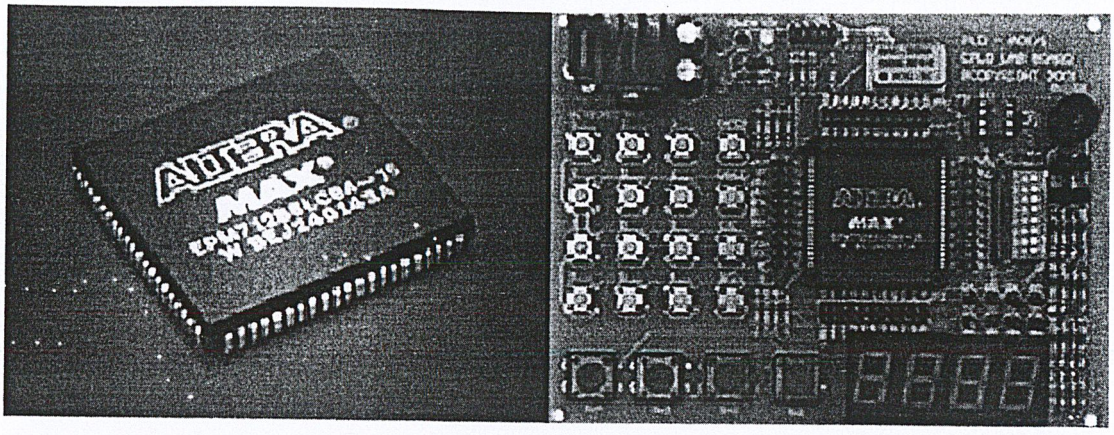
รูปที่ 2.26 ผังแสดงการแบ่งกลุ่มของวงจรรวมเอซิก

2.4 การออกแบบวงจรเชิงเลขด้วย ชิพอุปกรณ์เอฟพีจีเอ

ชิพอุปกรณ์เอฟพีจีเอ เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์เอฟพีจีเอ มีฟังก์ชั่นการทำงานตามที่ออกแบบไว้ ในการทำชิพอุปกรณ์ ซึ่งเป็นวิธีการออกแบบ ไอซี (IC : Integrated Circuit) แบบ เซมิคัสตัม (Semicustom) อีกวิธีหนึ่ง เมื่อเทียบกับการทำ เอซิก แล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือ การทำ ชิพอุปกรณ์เอฟพีจีเอ จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในชิพอุปกรณ์เอฟพีจีเอ จะมีจำนวนเกท (gate) ให้ใช้จำนวนจำกัด และการทำชิพอุปกรณ์เอฟพีจีเอ ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำชิพอุปกรณ์ก็คือ ระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (download) นั้นน้อยกว่าการทำเอซิก มากและการตรวจสอบหรือแก้ไขการออกแบบที่ทำได้สะดวก

การทำชิพอุปกรณ์เอฟพีจีเอ ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตชิพอุปกรณ์เอฟพีจีเอ ได้เพิ่มความสามารถของชิพอุปกรณ์เอฟพีจีเอ โดยเพิ่มจำนวนองค์ประกอบภายใน หรือปรับปรุงโครงสร้างสถาปัตยกรรมภายในและยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ พีพีอาร์ (PPR : Partitioning Placement and Routing) สำหรับอุปกรณ์นั้นๆด้วย ลักษณะของตัว ชิพอุปกรณ์เอฟพีจีเอ และการนำไปใช้งานแสดงดังในรูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 แสดงลักษณะของตัว FPGA และการนำไปใช้งาน

สำหรับตัวชิพอุปกรณ์เอฟพีจีเอนั้นมีโครงสร้างพื้นฐาน เทคโนโลยีที่ใช้สร้าง ตลอดจนเทคนิควิธีการโปรกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นชิพอุปกรณ์เอฟพีจีเอ ของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วน ในการใช้งานนั้นชิพอุปกรณ์เอฟพีจีเอ สามารถนำไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP : Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้น

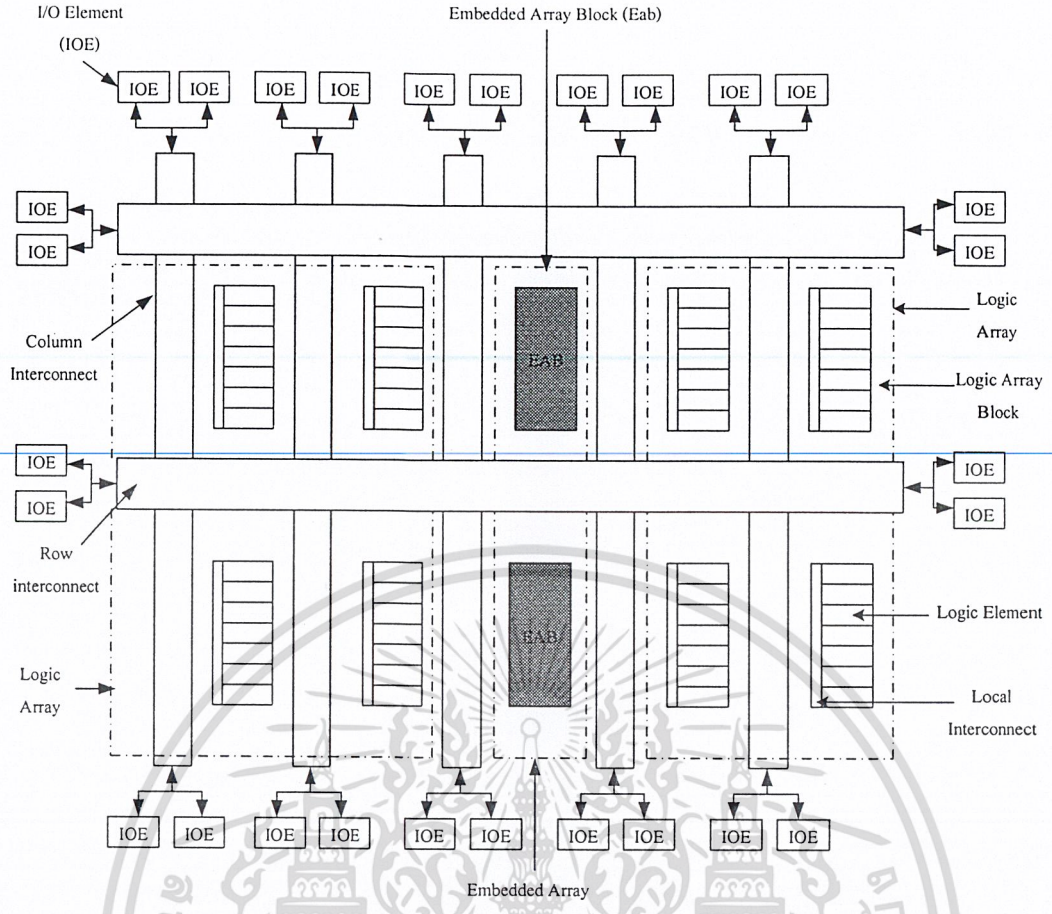
2.5 สถาปัตยกรรมภายในของชิพอุปกรณ์เอฟพีจีเอ

ชิพอุปกรณ์เอฟพีจีเอของบริษัท Altera ตระกูล FLEX 10 K เป็นอุปกรณ์ที่มีความหนาแน่นเกทประมาณตั้งแต่ 250,000-10,000 เกท โดยการจัดโครงสร้าง (Configuration) จะใช้วิธีโหลดโครงสร้างเข้าไปใน SRAM ภายใน ซึ่งหมายความว่าไม่ได้มีการจ่ายไฟเลี้ยงให้ โครงสร้างที่จัดเอาไว้ก็จะหายไป ชิพอุปกรณ์เอฟพีจีเอ ประเภทนี้จะสามารถโปรแกรมซ้ำได้ไม่จำกัดจำนวนครั้ง และการทำงานของลอจิกฟังก์ชั่นจะใช้วิธีการเปิดตารางความจริงดู (Look Up table : LUT) โดยโครงสร้างของชิพอุปกรณ์เอฟพีจีเอตระกูล FLEX 10 K แสดงดังรูปที่ 2.28 โดยในโครงสร้างของชิพอุปกรณ์เอฟพีจีเอตระกูล FLEX 10 K สามารถที่จะแบ่งเป็นส่วนต่างๆ ได้ดังนี้

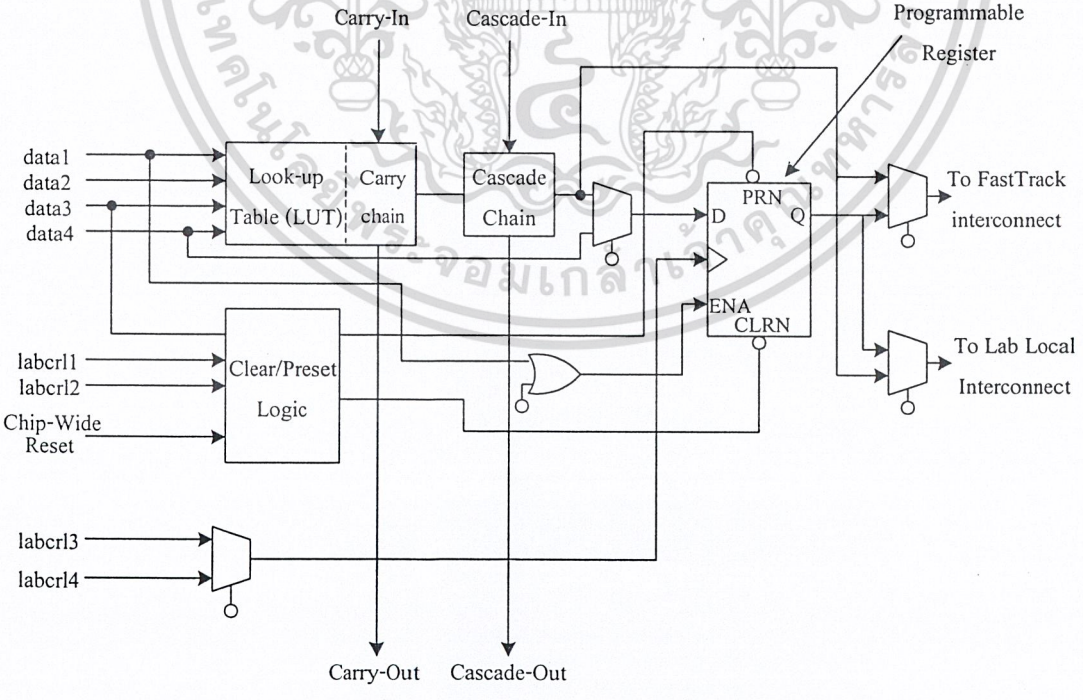
2.5.1 แอลอี (LE:Logic Element)

ในรูปที่ 2.29 แสดงโครงสร้างภายในของแอลอี โดยการกระทำทางบูลีนของลอจิกเกทจะสร้างด้วยวิธีการ LUT โดย LUT คือ 1x16 SRAM ซึ่ง LUT เพียงตัวเดียวสามารถนำมาทำโครงข่ายของลอจิกเกทที่มี 4 อินพุท และ 1 เอาท์พุท โดยโครงข่ายของลอจิกเกทจะถูกแปลงไปเป็นตารางค่าความจริง (Truth Table) ดังแสดงในรูปที่ 2.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

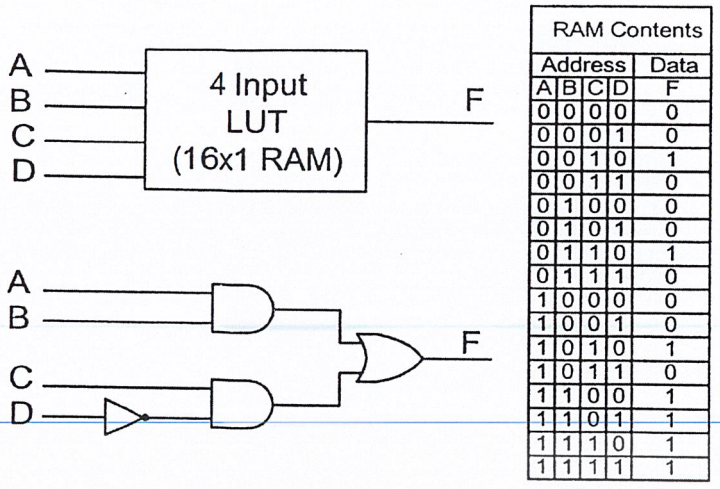


รูปที่ 2.28 แสดงโครงสร้างของชิพอุปกรณ์เอฟพีจีเอ ตระกูล FLEX 10 K



รูปที่ 2.29 แสดงโครงสร้างภายในของแอลอี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

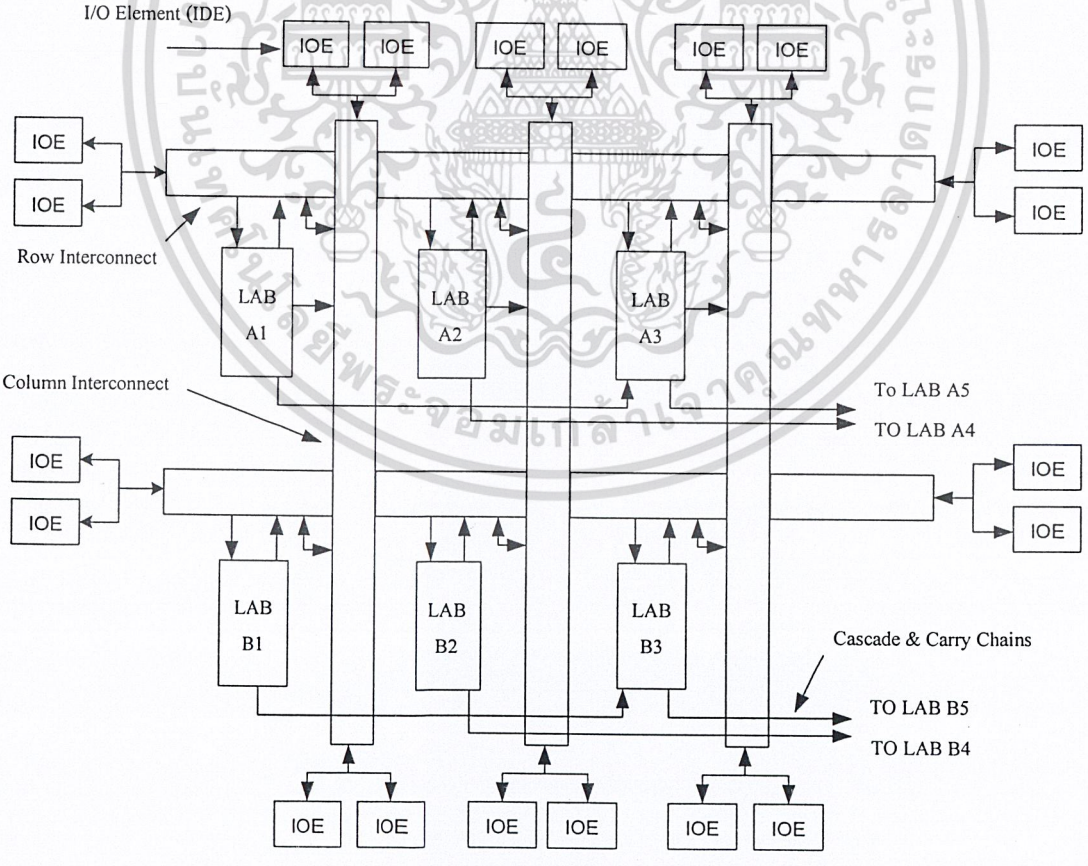


RAM Contents

Address				Data
A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

รูปที่ 2.30 แสดงการใช้งาน LUT เป็นโครงข่ายของลอจิก

ถ้าโครงข่ายของลอจิกเกทมีความซับซ้อนขึ้นจะต้องใช้ LUT ของแต่ละ LE เป็นจำนวนหลายตัว โดยเอาท์พุทของ LUT จะส่งต่อไปยังฟลิปฟลอปและต่อไปยังโครงข่ายการเชื่อมต่อ (Interconnection Network) ดังแสดงในรูปที่ 2.34

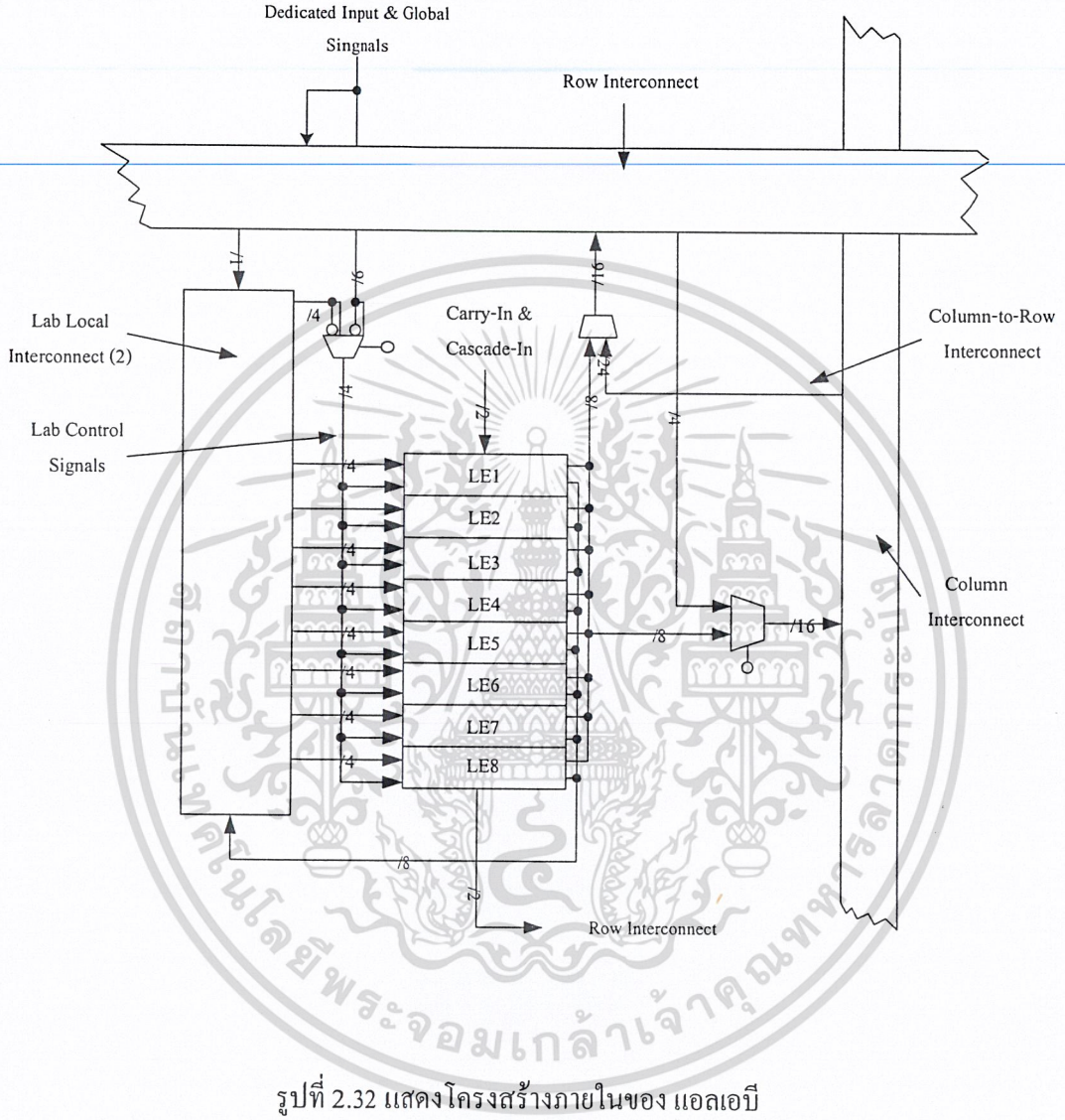


รูปที่ 2.31 แสดงโครงข่ายของการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 แอลเอบี (LAB : Logic Array Block)

แอลเอบี 1 ตัว จะประกอบไปด้วย 8 แอลอี ดังแสดงในรูปที่ 2.32

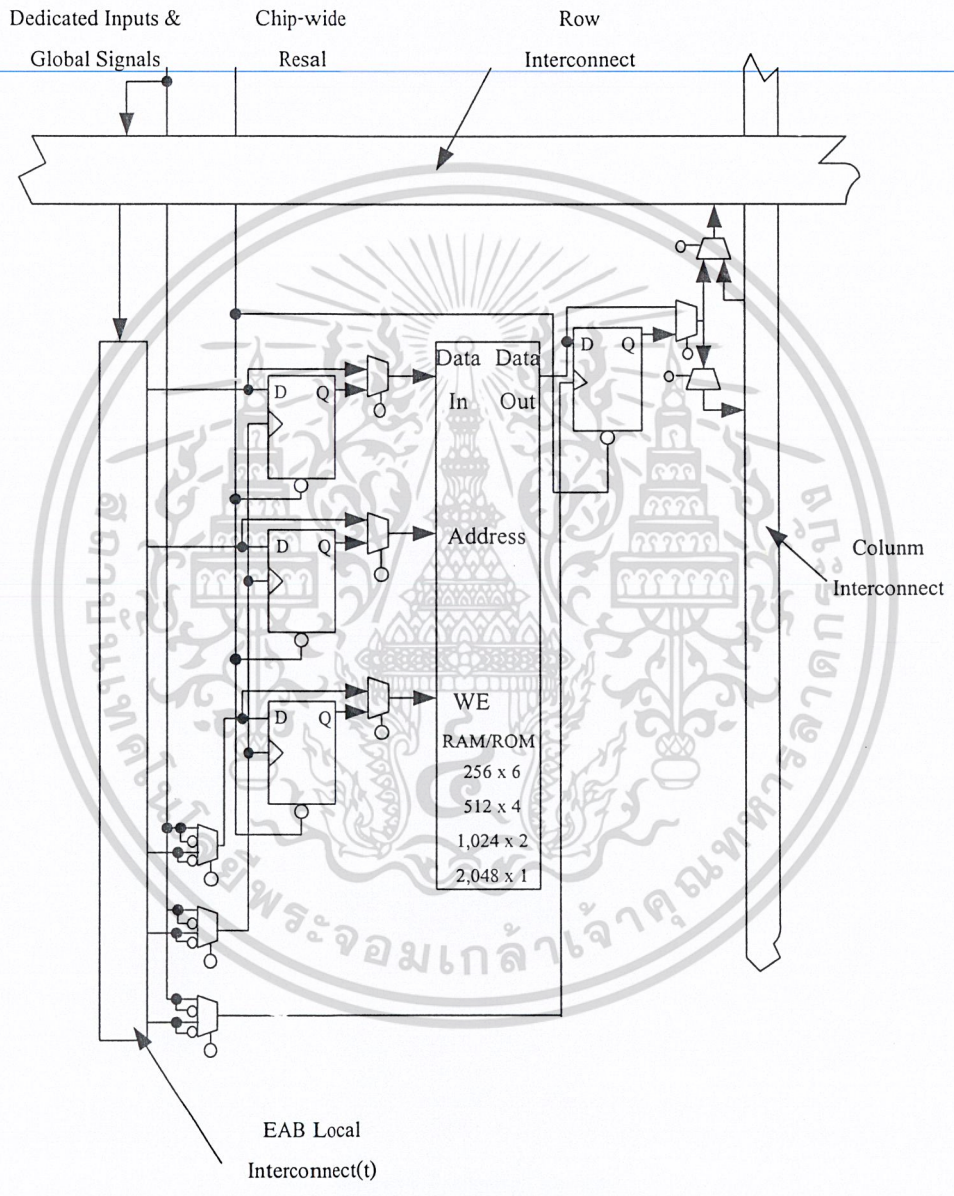


รูปที่ 2.32 แสดงโครงสร้างภายในของ แอลเอบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 อีเอบี (EAB : Embedded Array Block)

สถาปัตยกรรมโดยทั่วไปของ FLEX 10 K จะมีลักษณะของแอลเอบี ที่มีการจัดเรียงแบบเมตริกซ์ และ อีเอบี ซึ่งมีการเชื่อมต่อผ่านทางแถวและคอลัมน์ โดยในแต่ละแถวจะมี 1 อีเอบี จะมีขนาด 2048 บิต และสามารถกำหนดความกว้าง (Width) ความลึก (Depth) ของ อีเอบี ได้โดยไม่ส่งผลต่อความเร็ว

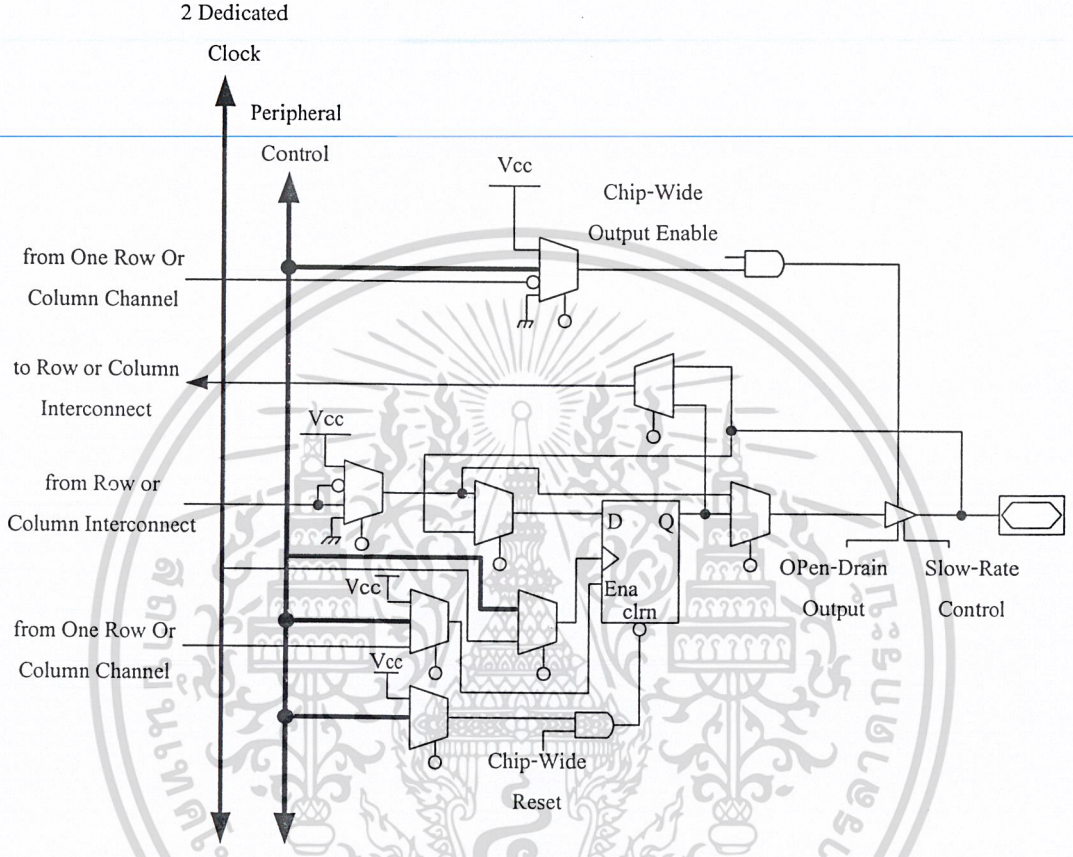


รูปที่ 2.33 แสดงโครงสร้างภายในอีเอบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 ไอโออี (IOE : Input Output Element)

ไอโออี จะถูกต่ออยู่กับขา I/O โดยจะประกอบด้วยส่วนของวงจรที่เป็น Tri State และส่วนที่เป็นฟลิปฟลอป ซึ่งเป็น option ดังแสดงในรูปที่ 2.34



รูปที่ 2.34 แสดงโครงสร้างภายในของไอโออี

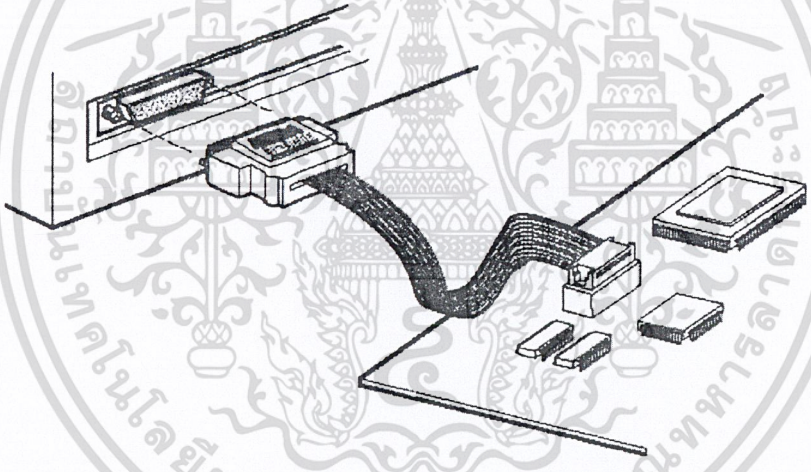
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ปัจจัยที่ทำให้การออกแบบชิพอุปกรณ์เอฟพีจีเอ ทำได้ง่ายและสะดวกรวดเร็ว

1. ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพเพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึงภาษาแอสแซมบี้ (Assembly) ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2. มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจรหรือเอชดีแอล เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มันจากนั้นตัวซอฟต์แวร์จะทำการสังเคราะห์ ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3. การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายดาวน์โหลดทางพอร์ตของคอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก ดังรูปที่ 2.35 และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด



รูปที่ 2.35 การโปรแกรมลงในชิพอุปกรณ์เอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

จากทฤษฎีการเข้าและถอดรหัสแบบคอนโวลูชั่นโค้ด สามารถแบ่งการออกแบบได้เป็น 2 ส่วนใหญ่ๆ ดังนี้

1. ส่วนของการเข้ารหัส ซึ่งจะทำการเข้ารหัสสัญญาณดิจิตอลจากสัญญาณเลขฐานสอง แบบธรรมดาให้เป็นสัญญาณเลขฐานสองที่ผ่านการเข้ารหัสแบบคอนโวลูชั่นโค้ด

2. ส่วนของการถอดรหัส ซึ่งจะทำการถอดรหัสสัญญาณดิจิตอลจากการเข้ารหัสแบบคอนโวลูชั่นโค้ดให้กลับมาเป็นสัญญาณดิจิตอลแบบรหัสเลขฐานสอง แบบธรรมดาตามเดิม

3. ส่วนของการวัดอัตราความผิดพลาด (BER : Bit Error Rate) ซึ่งเป็นส่วนของการวัดอัตราความผิดพลาดของข้อมูลในส่วนที่ไม่ได้ทำการเข้ารหัสและในส่วนข้อมูลที่ทำการเข้ารหัส

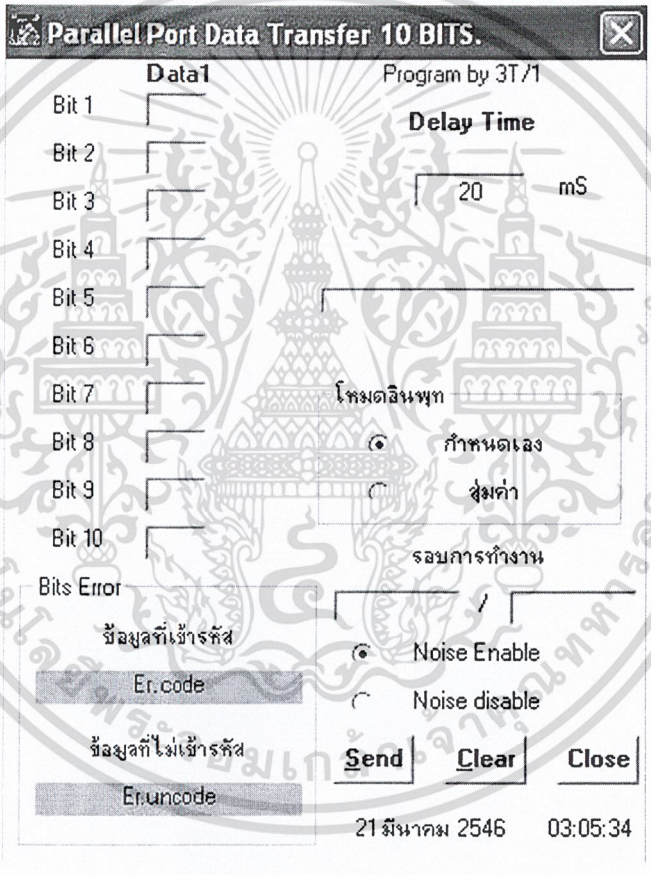
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบ

3.1 ส่วนของการเข้ารหัส (Encoder Part)

ประกอบไปด้วยส่วนของการส่งค่าอินพุตจากคอมพิวเตอร์ วงจรบัพเฟอร์ (Buffer) วงจรสร้างสัญญาณควบคุม (Control Signal) วงจรลดค่าความถี่ของสัญญาณนาฬิกา (Clock Divider) และส่วนของการเข้ารหัส

3.1.1 ส่วนของการเชื่อมต่อกับคอมพิวเตอร์ โดยการทดสอบในส่วนของการเข้ารหัส เราจำเป็นที่จะต้องป้อนอินพุทให้แก่ตัวเข้ารหัส และพิจารณาโค้ดที่ได้เทียบกับทฤษฎีว่ามีการเข้ารหัสถูกต้องหรือไม่ โดยเราจะทำการเขียนโปรแกรมโดยใช้ภาษาวิชวลเบสิก ซึ่งจะทำการเขียนโปรแกรมให้มีการระบุค่าของอินพุตในการทดสอบครั้งละ 10 บิต

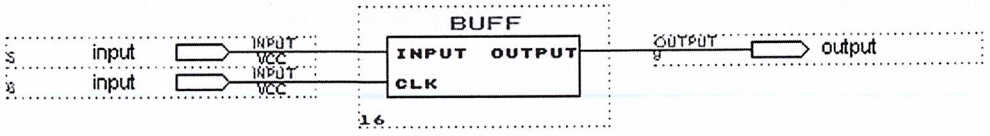


รูปที่ 3.1 แสดงโปรแกรมการใช้งานส่งค่าบิตข้อมูลออกทางพอร์ตขนาน

จากรูปที่ 3.1 การทำงานของโปรแกรมจะทำโดยกำหนดค่าของข้อมูลที่ต้องการส่ง โดยกำหนดข้อมูลได้ 10 บิตหรือกำหนดข้อมูลให้ออกไปแบบสุ่ม และสามารถตั้งค่าหน่วงเวลาตามต้องการได้ มีหน้าที่เป็นทั้งตัวที่ส่งข้อมูลและเป็นตัวนับค่าความผิดพลาดของบิตข้อมูล (Bit Error Rate) รวมถึงเป็นตัวกำหนดการทำงานในส่วนของการสุ่มข้อมูล (ในส่วนของวงจรภายในชิพอุปกรณ์เอฟพีจีเอ) เพื่อเป็นการจำลองว่าข้อมูลมีการผิดพลาด

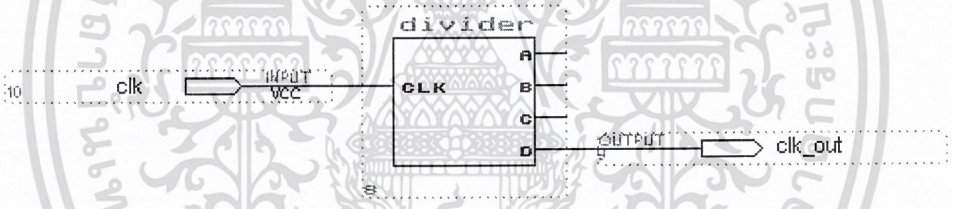
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ส่วนของวงจรบัฟเฟอร์ (Buffer) จะทำหน้าที่ประสานจังหวะของอินพุตที่มาจากคอมพิวเตอร์กับสัญญาณนาฬิกาของระบบให้มีความสัมพันธ์กันโดยอินพุต 1 บิต จะถูกควบคุมด้วยสัญญาณนาฬิกา 1 ไซเคิล (Cycle) มีลักษณะดังรูปที่ 3.2



รูปที่ 3.2 แสดงสัญลักษณ์ของส่วนวงจรบัฟเฟอร์

3.1.3 ส่วนของวงจรลดค่าความถี่ (Divider) ทำหน้าที่ลดค่าความถี่ของสัญญาณนาฬิกาที่ได้จากตัวอุปกรณ์บอร์ดเอฟพีจีเอให้มีความถี่ลดลง เพื่อให้สามารถวัดค่าของสัญญาณที่เกิดขึ้นในระบบได้ (เนื่องจากถ้าไม่ทำการลดค่าความถี่จะไม่สามารถสังเกตสัญญาณได้ชัดเจน) จะมีลักษณะดังรูปที่ 3.3

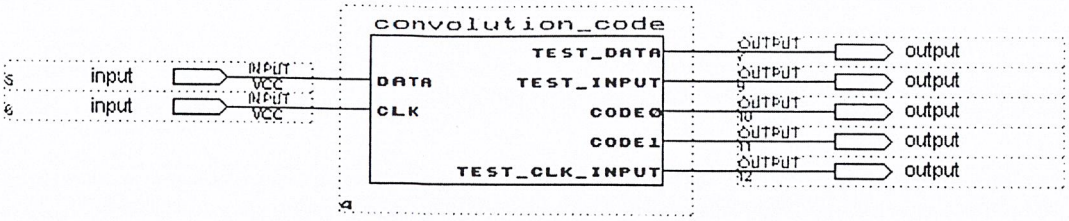


รูปที่ 3.3 แสดงสัญลักษณ์ของวงจรลดค่าความถี่

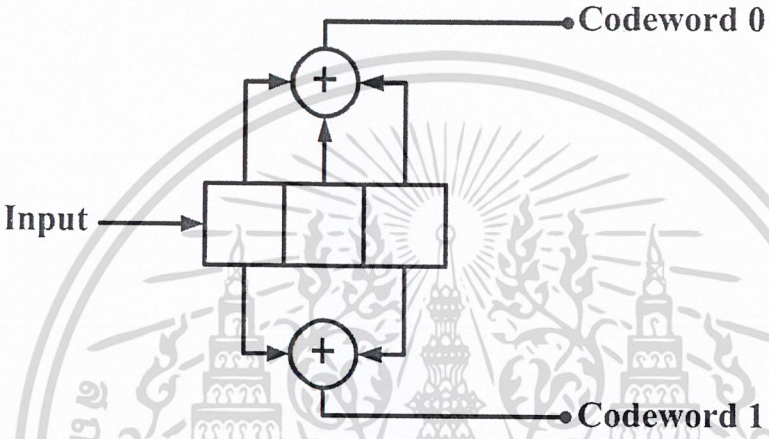
3.1.4 ส่วนของการเข้ารหัส คือส่วนที่จะทำการประมวลผลสัญญาณอินพุตที่เข้ามาให้ได้ผลลัพธ์คือสัญญาณเอาต์พุตที่ได้จากการเข้ารหัส โดยใช้หลักการการเข้ารหัสแบบคอนโวลูชั่นโค้ด

ในการออกแบบในปริญญานิพนธ์นี้จะออกแบบโดยใช้ค่าคอนสเตร็นท์ เท่ากับ 3 และในส่วนของการบวกกันแบบมอดูโล -2 สองชุด โดยชุดแรกจะเป็นการบวกกันระหว่างค่าประจำตำแหน่งคอนสเตร็นท์ตำแหน่งที่ 1, 2 และ 3 ส่วนชุดที่สอง จะเป็นการบวกกันระหว่างค่าประจำตำแหน่งคอนสเตร็นท์ตำแหน่งที่ 1 และ 3 ซึ่งจะทำให้ได้เอาต์พุต คือการเข้ารหัสแบบคอนโวลูชั่นโค้ด จะมีลักษณะดังแสดงในรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงสัญลักษณ์ของส่วนการเข้ารหัส แบบคอนโวลูชั่น



รูปที่ 3.5 แสดงลักษณะการเข้ารหัสแบบคอนโวลูชั่น

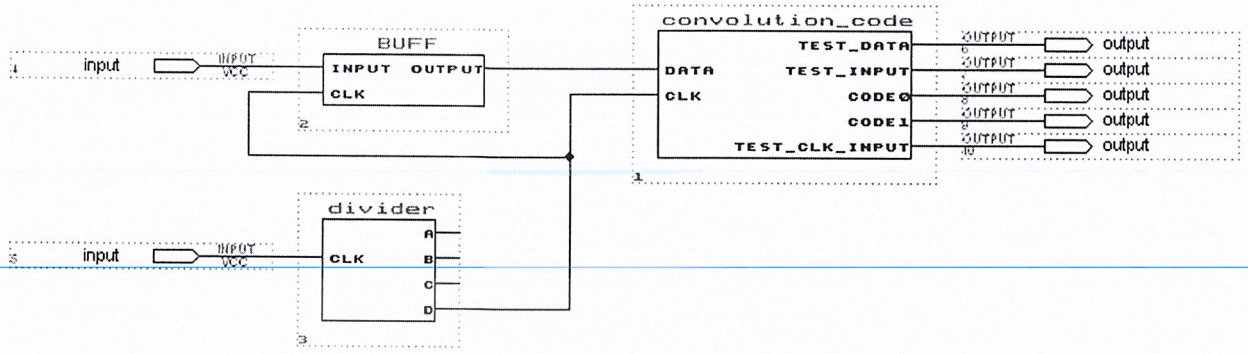
สามารถเขียนตารางความจริงได้ดังนี้

สเตจ	อินพุต	
	0	1
	ค่าภายในคอนสเตร็นท์/เอาต์พุต	ค่าภายในคอนสเตร็นท์/เอาต์พุต
00	000/00	100/11
10	010/10	110/01
01	001/11	101/00
11	011/01	111/10

ตารางที่ 3.1 แสดงค่าความจริงของส่วนการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

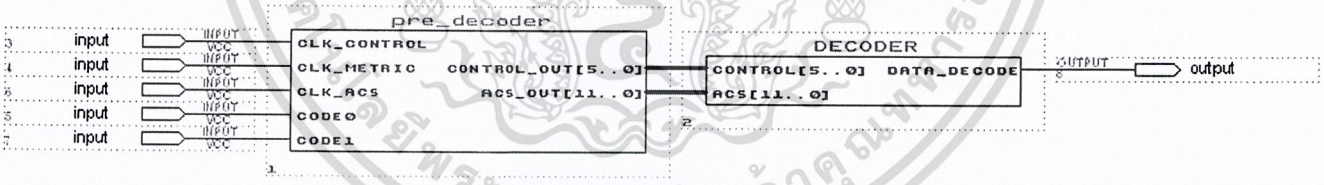
ทำการรวมส่วนประกอบต่างๆ ตามบล็อกไดอะแกรมการทำงานของการเข้ารหัส ทำการจำลองการทำงานและทดสอบ



รูปที่ 3.6 แสดงการทำงานในส่วนของการเข้ารหัส

3.2 ส่วนของการถอดรหัส(Decoder Part)

ส่วนของการถอดรหัสมี 2 ส่วน คือ ส่วนของพรีดีโค้ดเดอร์ (Pre_decoder) และส่วนของดีโค้ดเดอร์ (Decdoer)

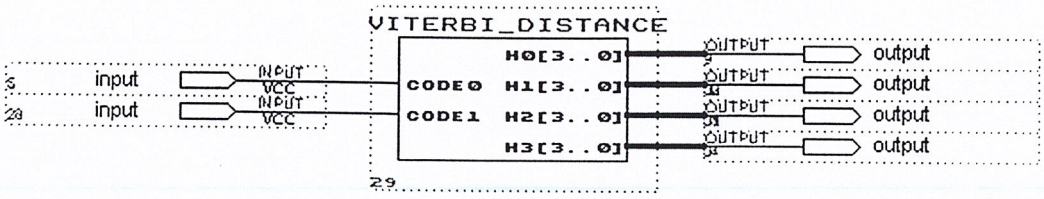


รูปที่ 3.7 แสดงส่วนของการถอดรหัส

จากรูปที่ 3.7 ในส่วนของพรีดีโค้ดเดอร์ ทำหน้าที่หาค่าความผิดพลาดของข้อมูลในแต่ละเส้นทางและจัดเตรียมข้อมูลที่ได้ผ่านการเข้ารหัสมาแล้วให้พร้อม เพื่อส่งข้อมูลที่หาค่าความผิดพลาดแล้วให้ดีโค้ดเดอร์เพื่อทำการถอดรหัสข้อมูลเดิมกลับมา โดยส่วนโครงสร้างภายในของพรีดีโค้ดเดอร์ประกอบไปด้วยส่วนต่างๆตามรายละเอียด ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 วงจรไวตาบิดิสแตนซ์ (Viterbi_distance)



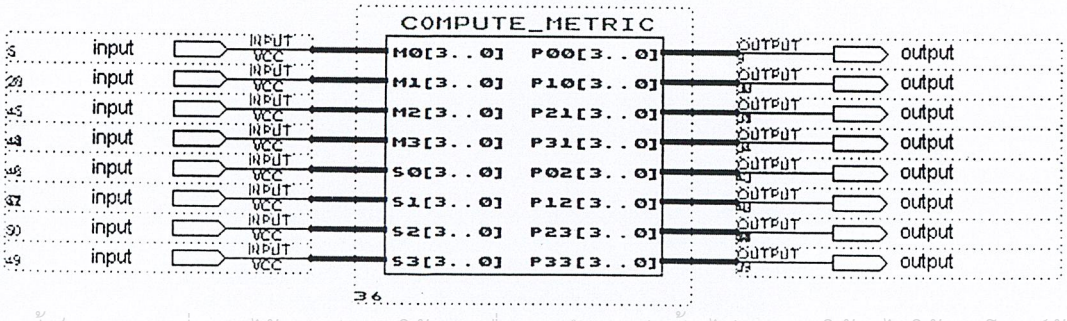
รูปที่ 3.8 แสดงสัญลักษณ์ของวงจรไวตาบิดิสแตนซ์ ที่ได้จากการคอมไพล์

จากรูปที่ 3.8 ทำหน้าที่หาระยะแฮมมิ่งดิสแตนซ์ (Hamming Distance) ของสัญญาณที่รับเข้ามาเพื่อใช้ในการหาค่าความผิดพลาดเมื่อเทียบกับค่าน้ำหนักในแต่ละเส้นทางที่ทำการเลือกโดยมีค่าตามตารางค่าความเป็นจริงดังนี้

input	output			
	h0	h1	h2	h3
00	0000	0001	0001	0010
01	0001	0000	0010	0001
10	0001	0010	0000	0001
11	0010	0001	0001	0000

ตารางที่ 3.2 แสดงค่าความเป็นจริงของระยะแฮมมิ่งเมื่อเทียบกับค่าน้ำหนักในแต่ละเส้นทางที่ทำการเลือก

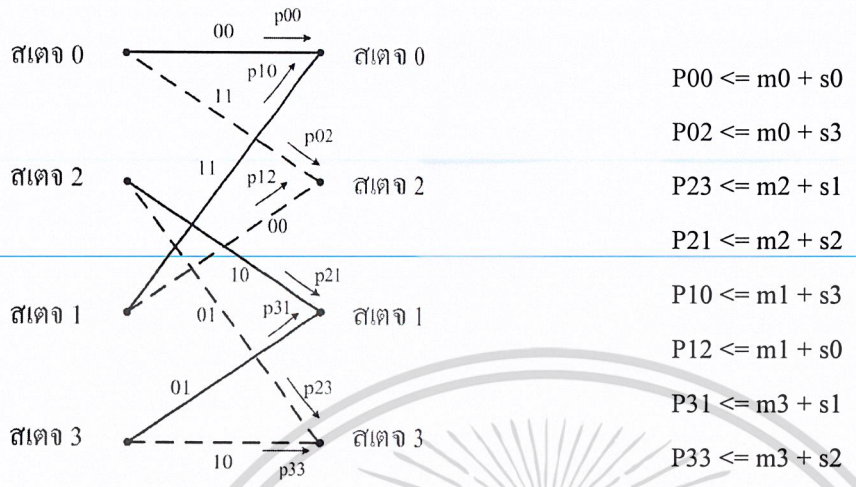
3.2.2 วงจรคอมพิวเมตริกซ์ (Compute_metric)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

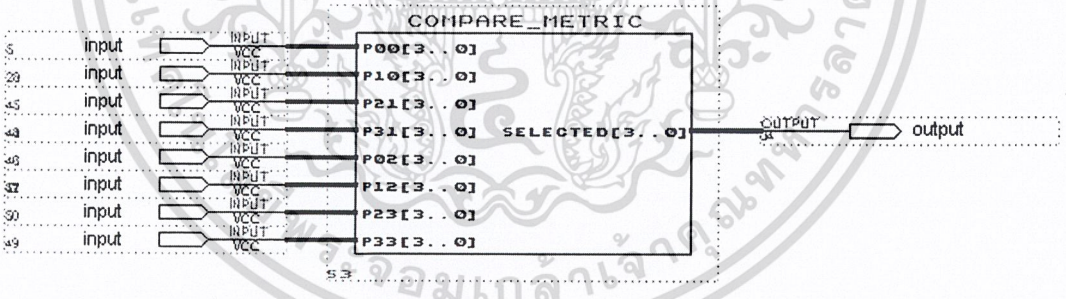
รูปที่ 3.9 แสดงสัญลักษณ์ของวงจรคอมพิวเมตริกซ์

จากรูปที่ 3.9 วงจรนี้จะทำหน้าที่หาค่าสะสมของค่าความผิดพลาดที่เกิดจากการรวมกันของค่าที่ได้จากวงจรหาระยะแฮมมิ่ง รวมกับค่าประจำเส้นทางต่างๆโดยค่าต่างๆ จะกำหนดโดยแผนภาพเทลลิส



รูปที่ 3.10 แสดงแผนภาพเทลลิสของการพิจารณาค่าความผิดพลาดในแต่ละเส้นทาง

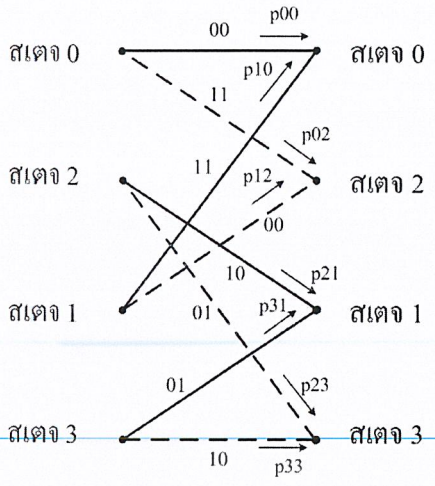
3.2.3 วงจรคอมแพร์เมตริกซ์ (Compare_metric)



รูปที่ 3.11 แสดงสัญลักษณ์ของวงจรคอมแพร์เมตริกซ์

จากรูปที่ 3.11 ในส่วนของวงจรคอมแพร์เมตริกซ์ จะเป็นส่วนที่เปรียบเทียบ เส้นทาง 2 เส้นทาง ที่มาบรรจบกัน ทั้ง 4 จุด ของแผนภาพเทลลิส โดยจะหาเส้นทางที่มีค่าของความผิดพลาด น้อยที่สุด เพื่อเลือกใช้เป็นเส้นทางหลักในการเลือกพิจารณาเส้นทางต่อไปที่เป็นข้อมูลจริงที่ถอดรหัสได้โดยจะเปรียบเทียบกันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงแผนภาพเทลลิสที่เส้นทาง 2 เส้น ทางมาบรรจบกันทั้ง 4 จุด

โดยมีการจับคู่การเปรียบเทียบกันระหว่าง 2 เส้นทางที่มาบรรจบกัน (ทฤษฎีไวตาบิคือ 2 เส้นทางที่มาบรรจบกัน เมื่อเส้นทางใดเส้นทางหนึ่งถูกเลือกแล้วอีกเส้นทางหนึ่งจะไม่ถูกนำมาพิจารณาอีก) โดยจากแผนภาพเทลลิสจะมีการจับคู่เปรียบเทียบดังนี้

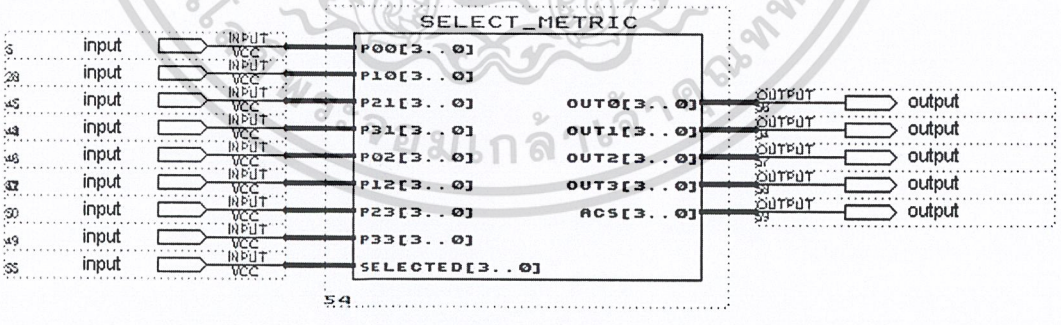
p00 เปรียบเทียบกับ p10

p02 เปรียบเทียบกับ p12

p21 เปรียบเทียบกับ p31

p23 เปรียบเทียบกับ p33

3.2.4 วงจรซีเลกเมตริกซ์ (Select_metric)

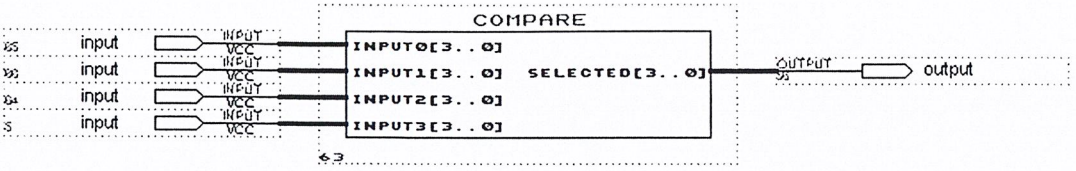


รูปที่ 3.13 แสดงสัญลักษณ์ของวงจรซีเลกเมตริกซ์

จากรูปที่ 3.13 ในส่วนของการเลือกเส้นทาง เมื่อทำการเปรียบเทียบแล้วจะทำการส่งสัญญาณแจ้งมายังวงจรซีเลกเมตริกซ์นี้เพื่อแจ้งเส้นทางที่ถูกเลือก โดยนอกจากจะให้ค่าของเส้นทางที่ถูกเลือกแล้วยังให้ค่าบิตที่แสดงสถานะของการเลือกคือ “0” กับ “1” โดย “0” คือ เส้นทางที่เลือกมาจากเส้นทางบน และ “1” คือเส้นทางที่เลือกมาจากเส้นทางล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

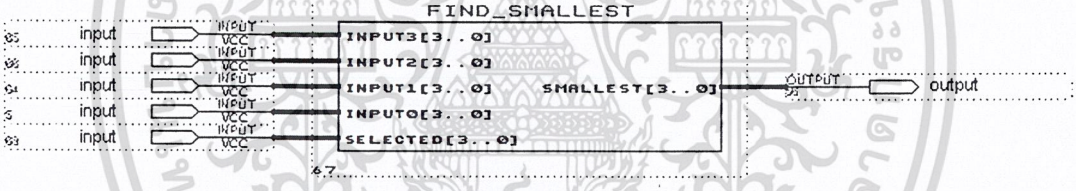
3.2.5 วงจรคอมแพร์ (Compare)



รูปที่ 3.14 แสดงสัญลักษณ์ของวงจรคอมแพร์

จากรูปที่ 3.14 ในส่วนของวงจรคอมแพร์นั้น ค่าที่ได้จากวงจรเลือกเส้นทาง จะมีทั้งหมด 4 ค่า และทำการเปรียบเทียบหาค่าเส้นทางที่มีค่าน้อยที่สุดเพียงเส้นทางเดียวและจะทำการส่งสัญญาณแจ้งไปยังส่วนของวงจรไฟน์สมอลเลท เพื่อใช้เป็นส่วนของการควบคุมการเลือกสัญญาณที่มีค่าความผิดพลาดที่น้อยที่สุด

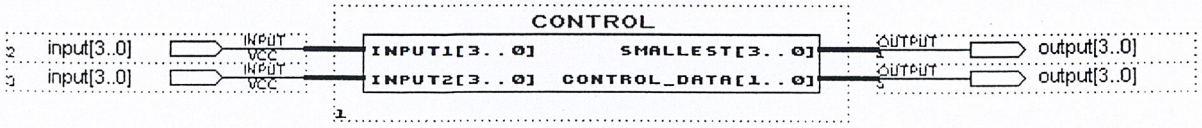
3.2.6 วงจรไฟน์สมอลเลท (find_smallest)



รูปที่ 3.15 แสดงสัญลักษณ์ของวงจรไฟน์สมอลเลท

จากรูปที่ 3.15 เมื่อรับค่าที่เปรียบเทียบกันแล้ว 4 ค่าจากวงจรซีเลกเมตริกซ์ โดยวงจรคอมแพร์จะเป็นตัวกำหนดการเลือกค่าของอินพุตทั้ง 4 ค่า เพื่อเลือกค่าเส้นทางที่มีค่าความผิดพลาดน้อยที่สุด ส่วนของค่าเส้นทางที่มีค่าความผิดพลาดน้อยที่สุดที่เลือกนั้นจะเป็นอินพุตในส่วนของการถอดรหัสต่อไป

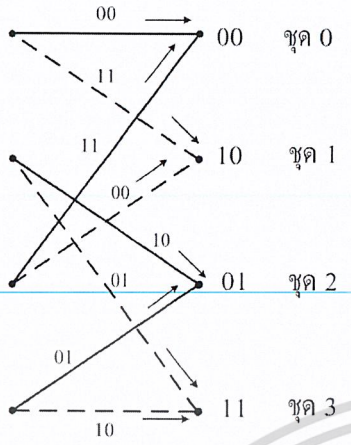
3.2.7 วงจรคอนโทรล (Control)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 แสดงสัญลักษณ์ของวงจรคอนโทรล

จากรูปที่ 3.16 เป็นส่วนที่สร้างสัญญาณควบคุมที่จะบอกว่าค่าของเส้นทางที่เราทำการเลือกนั้นมาจากชุดใด ภายใน 4 ชุดของแผนภาพเทลลิสที่ได้จากการเปรียบเทียบ

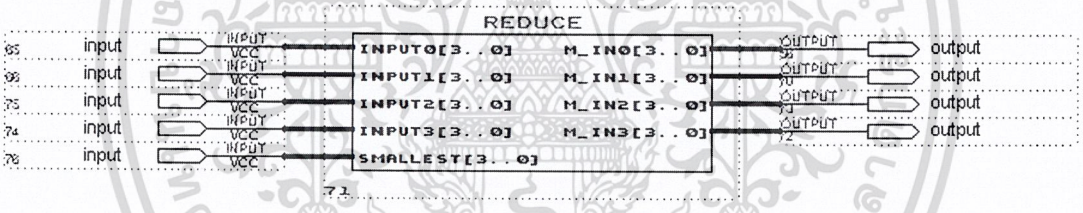


รูปที่ 3.17 แผนภาพเทลลิสแสดงสัญญาณคอนโทรลในแต่ละชุดของสัญญาณ

สัญญาณคอนโทรล	ชุดของสัญญาณ
00	0
01	1
10	2
11	3

ตารางที่ 3.3 แสดงตำแหน่งของสัญญาณคอนโทรลในแต่ละชุดของสัญญาณ

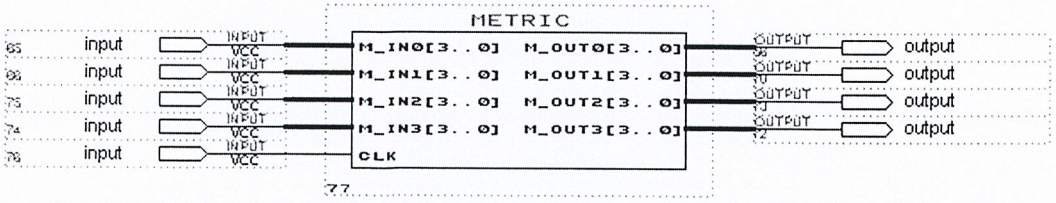
3.2.8 วงจรรีดิวซ์ (Reduce)



รูปที่ 3.18 แสดงสัญลักษณ์ของวงจรรีดิวซ์

จากรูปที่ 3.18 เมื่อทำการเลือกเส้นทางแล้ว ค่าของความผิดพลาดที่สะสมมาต้องถูกลบออกโดยจะลบทุกๆเส้นทาง ออกจากค่าเส้นทางที่น้อยที่สุด

3.2.9 วงจรเมตริกซ์ (Metric)

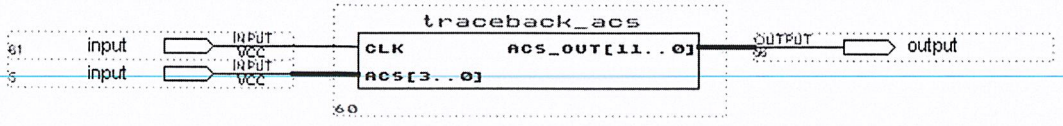


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 แสดงสัญลักษณ์ของวงจรเมตริกซ์

จากรูปที่ 3.19 ทำหน้าที่รับค่าจากวงจรลดค่าสะสมแล้วส่งค่าที่รับได้ต่อไปยังส่วนของวงจรคำนวณหาค่าสะสมของความผิดพลาด เพื่อใช้ในการหาค่าในส่วนต่อไป

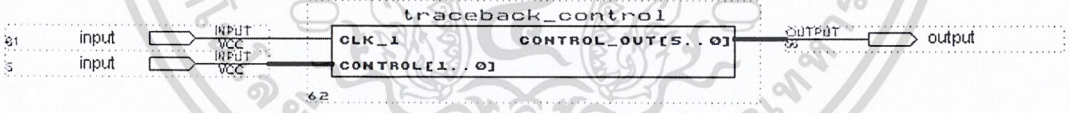
3.2.10 วงจรเทรดแบคเอซีเอส (Traceback_acs)



รูปที่ 3.20 แสดงสัญลักษณ์ของวงจรเทรดแบคเอซีเอส

จากรูปที่ 3.20 วงจรเทรดแบคเอซีเอส ทำหน้าที่เก็บค่าของความผิดพลาดในแต่ละเส้นทางแต่ละจุดของแผนภาพเทลลิส โดยจะต้องทำการเก็บค่าอย่างน้อย 11 ค่า ใน หน่วยความจำ โดยคำนวณจากสมการ 5×(K – 1) โดยจะทำการพิจารณาย้อนหลังไป 3 ชุดก่อนที่จะทำการเลือกค่าของสัญญาณที่ถอดรหัสออกมาได้จริง (K คือค่าความยาวคอนสเตร็นท์ของการเข้ารหัส)

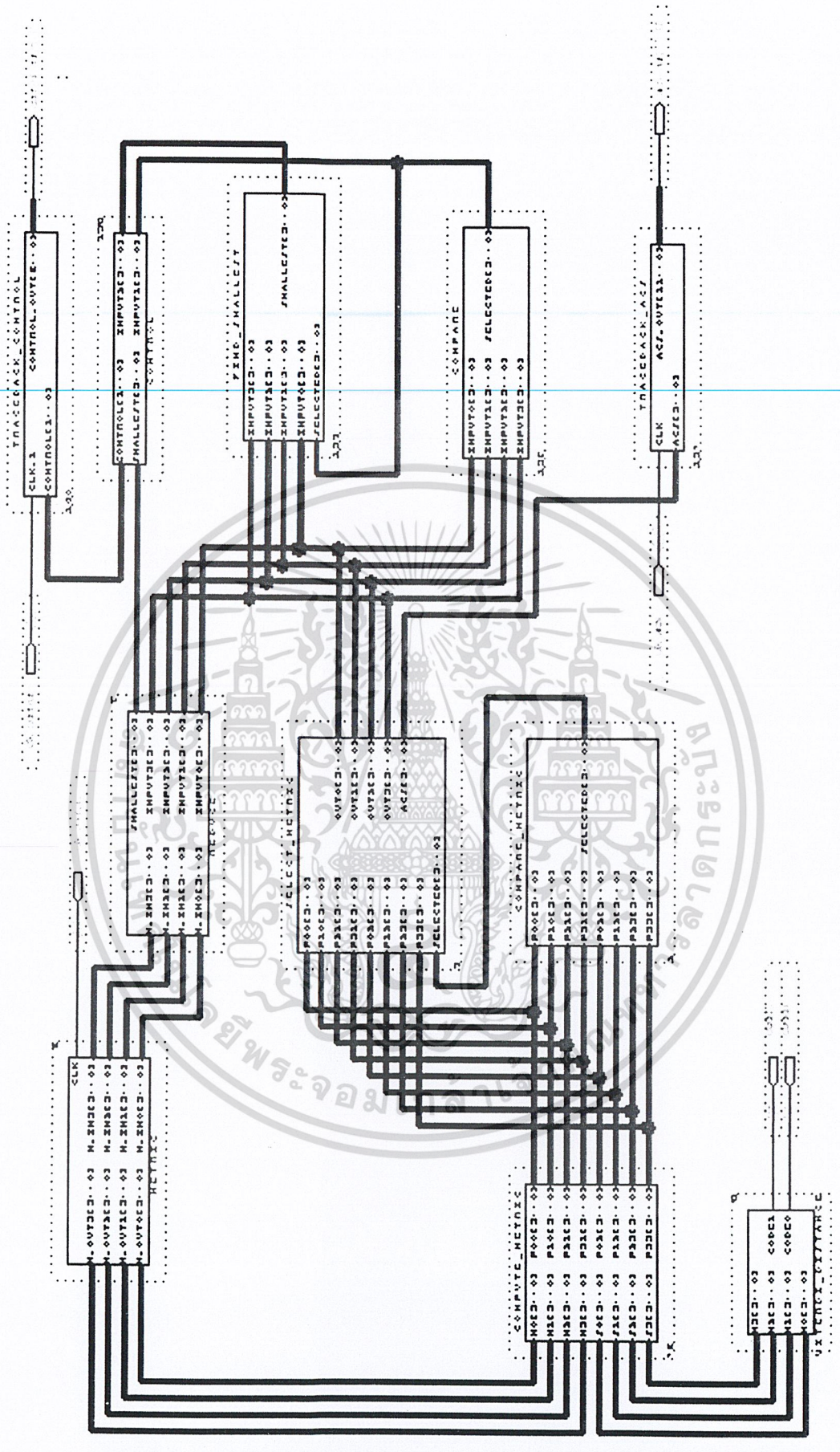
3.2.11 วงจรเทรดแบคคอนโทรล (Tracback_control)



รูปที่ 3.21 แสดงสัญลักษณ์ของวงจรเทรดแบคคอนโทรล

จากรูปที่ 3.21 ทำงานคล้ายวงจรเทรดแบคเอซีเอสและทำหน้าที่ชี้ค่าเส้นทางที่มีค่าน้อยที่สุด โดยจะทำการพิจารณาค่าเส้นทางย้อนหลังไป 3 ชุดก่อนที่จะทำการเลือกค่าของสัญญาณที่ถอดรหัสออกมาได้จริง โดยลักษณะการส่งข้อมูลจะเป็นดังนี้คือถ้ามีข้อมูล 1,2,3,4,5 วงจรเทรดแบคคอนโทรลจะทำการส่งข้อมูลแบบย้อนหลังออกไปชุดละ 3 ค่า คือ 3,2,1 และ 4,3,2 และ 5,4,3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

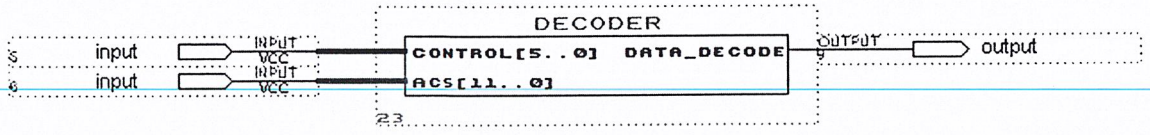


รูปที่ 3.22 แสดงโครงสร้างของวงจรพรีดีโค้ดเดอร์ซึ่งประกอบด้วยส่วนต่างๆที่ได้กล่าวมาข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของดีโค้ดเดอร์ ทำหน้าที่ถอดรหัสข้อมูลที่รับมาจากพรีดีโค้ดเดอร์ ซึ่งข้อมูลที่ถอดรหัสออกมานั้นจะเป็นข้อมูลชุดเดิมก่อนที่จะทำการเข้ารหัสในตอนแรก โดยในส่วนของดีโค้ดเดอร์นี้ประกอบไปด้วยวงจรถอดรหัสเพียงวงจรเดียว

3.2.12 วงจรถอดรหัส (Decoder)



รูปที่ 3.23 แสดงสัญลักษณ์ของวงจรถอดรหัส(Decoder)

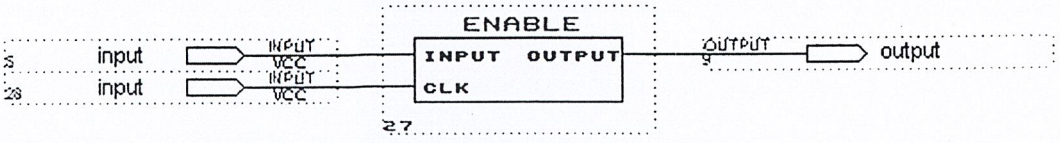
จากรูปที่ 3.23 เป็นส่วนของการถอดรหัสสัญญาณโดยจะพิจารณาอินพุต 2 ส่วน คือ เทรดแบคเอซีเอส และ เทรดแบคคอนโทรล โดย เทรดแบคเอซีเอสจะมีจำนวนข้อมูลอยู่ 12 บิต ส่วนเทรดแบคคอนโทรล จะมีจำนวนข้อมูลอยู่ 6 บิต ซึ่งในการออกแบบจะพิจารณาโดยใช้ในส่วนของ 4 บิตแรกของเทรดแบคเอซีเอส และ 2 บิตแรกของเทรดแบคคอนโทรล โดยมีเงื่อนไขดังนี้

- ถ้ามีค่าเทรดแบคคอนโทรล “00” ก็จะทำการพิจารณาเทรดแบคเอซีเอสในตำแหน่งที่ 4 (xxxA)
- ถ้ามีค่าเทรดแบคคอนโทรล “01” ก็จะทำการพิจารณาเทรดแบคเอซีเอสในตำแหน่งที่ 2 (xAxx)
- ถ้ามีค่าเทรดแบคคอนโทรล “10” ก็จะทำการพิจารณาเทรดแบคเอซีเอสในตำแหน่งที่ 3 (xxAx)
- ถ้ามีค่าเทรดแบคคอนโทรล “11” ก็จะทำการพิจารณาเทรดแบคเอซีเอสในตำแหน่งที่ 1 (Axxx)

- ถ้าเทรดแบคเอซีเอสมีค่า “0” สัญญาณที่ถอดรหัสได้คือข้อมูลบิต “0”
- ถ้าเทรดแบคเอซีเอสมีค่า “1” สัญญาณที่ถอดรหัสได้คือข้อมูลบิต “1”

3.3 ส่วนของการวัดค่าความผิดพลาดของข้อมูล (Bit Error Rate)

3.3.1 วงจรอีเนเบิล (Enable)

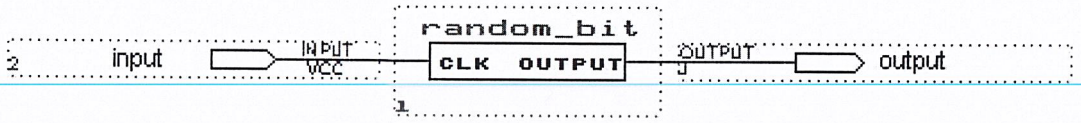


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.24 แสดงสัญลักษณ์ของวงจรอีเนเบิล

จากรูปที่ 3.24 ส่วนของวงจรอีเนเบิลทำหน้าที่ควบคุมการจำลองความผิดพลาดของสัญญาณ โดยสัญญาณที่ใช้ในการควบคุมเป็นสัญญาณที่มาจากคอมพิวเตอร์

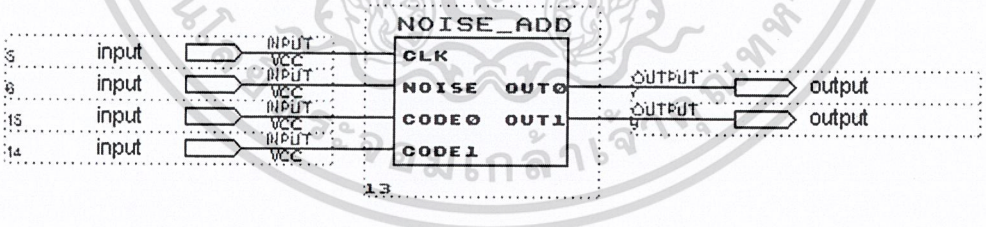
3.3.2 วงจรสุ่มค่า (Random_bit)



รูปที่ 3.25 แสดงสัญลักษณ์ของวงจรสุ่มค่า

จากรูปที่ 3.25 ส่วนของวงจรสุ่มค่าทำหน้าที่จำลองการเกิดข้อมูลแบบสุ่มเพื่อใช้ในการจำลองการเกิดความผิดพลาดของข้อมูล โดยใช้บวกเข้าไปในส่วนของข้อมูลที่ทำการเข้ารหัสและส่วนของข้อมูลที่ไม่ได้ทำการเข้ารหัสและทำการเปรียบเทียบกับข้อมูลที่ส่งมาจริงเพื่อทำการวัดค่าจำนวนความผิดพลาดของข้อมูล

3.3.3 วงจรจำลองการผิดพลาดของข้อมูล (Noise_add)

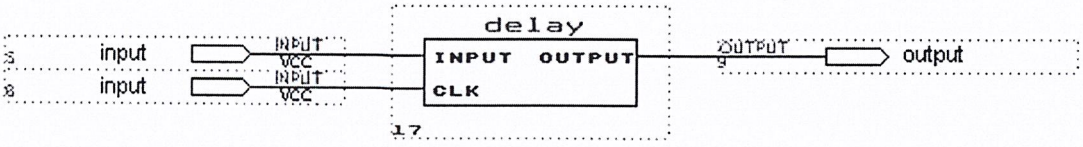


รูปที่ 3.26 แสดงสัญลักษณ์ของวงจรจำลองการผิดพลาดของข้อมูล

จากรูปที่ 3.26 ส่วนของวงจรจำลองการผิดพลาดของข้อมูลทำหน้าที่ บวกค่าที่ได้จากการสุ่มค่าของข้อมูลกับส่วนของข้อมูลที่ทำการเข้ารหัสและส่วนของข้อมูลที่ไม่ได้ทำการเข้ารหัสและทำการเปรียบเทียบกับข้อมูลที่ส่งมาจริงเพื่อทำการวัดค่าจำนวนความผิดพลาดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

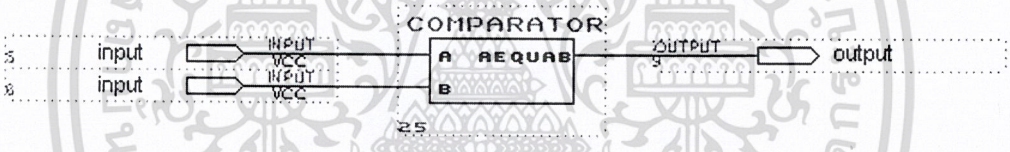
3.3.4 วงจรหน่วงเวลา (Delay_time)



รูปที่ 3.27 แสดงสัญลักษณ์ของวงจรหน่วงเวลา

จากรูปที่ 3.27 ส่วนของวงจรหน่วงเวลาทำหน้าที่ หน่วงเวลาในส่วนของข้อมูลที่ส่งมาจริงเพื่อให้มีช่วงเวลาในการเกิดสัญญาณตรงกับสัญญาณที่ถอดรหัสได้(เนื่องจากสัญญาณที่เข้ารหัสมานั้นจะต้องเสียเวลาในการถอดรหัสในช่วงแรกไป 4 ไซเคิล)

3.3.5 วงจรเปรียบเทียบสัญญาณ (Comparator)

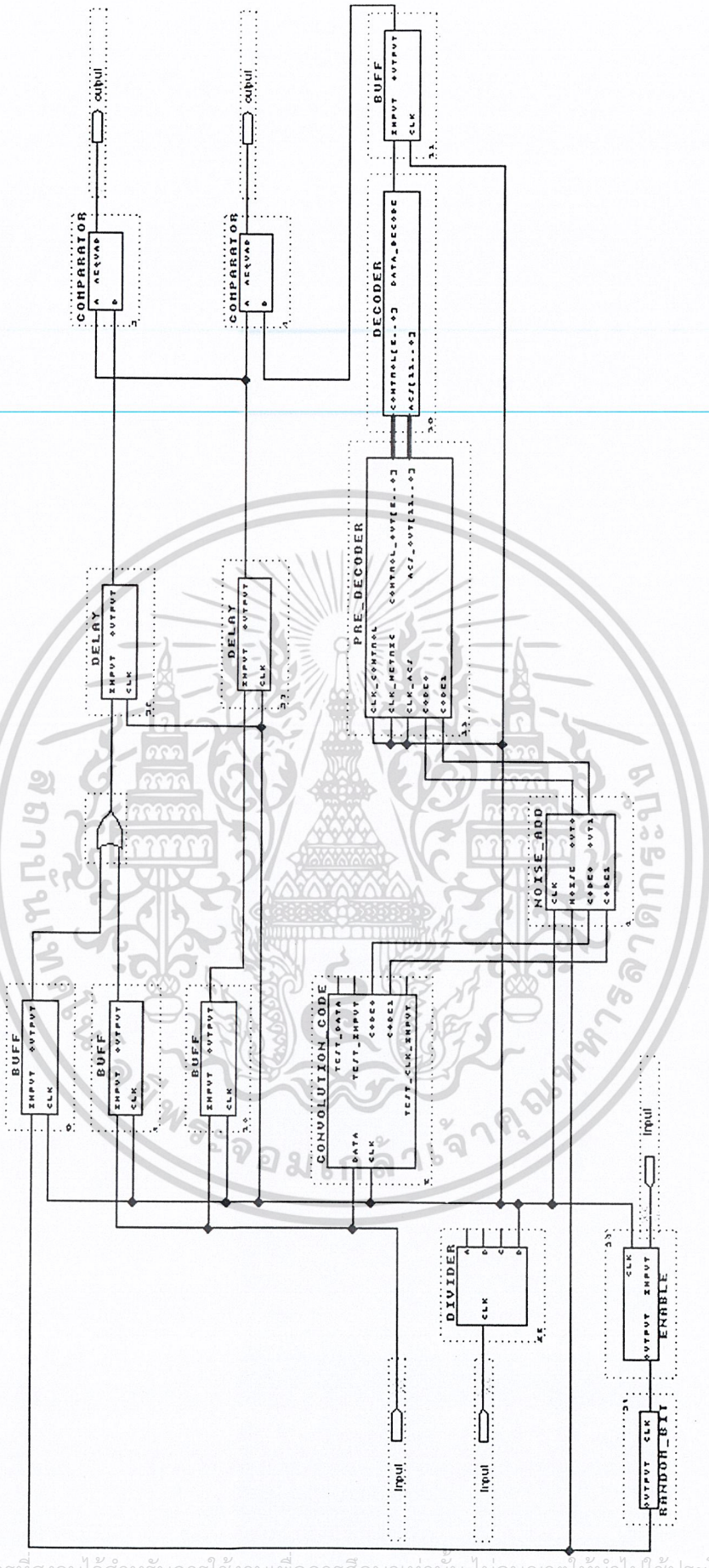


รูปที่ 3.28 แสดงสัญลักษณ์ของวงจรเปรียบเทียบสัญญาณ

จากรูปที่ 3.28 ส่วนของวงจรเปรียบเทียบทำหน้าที่ เปรียบเทียบส่วนของข้อมูลจริงกับส่วนของข้อมูลที่ทำการเข้ารหัสและส่วนของข้อมูลที่ไม่ได้ทำการเข้ารหัสเมื่อเกิดสภาวะของข้อมูลที่ผิดพลาด โดยจะทำการส่งค่าที่เปรียบเทียบได้กลับไปยังคอมพิวเตอร์เพื่อใช้ในการนับจำนวนค่าที่ผิดพลาด

จากทั้ง 3 ส่วนที่กล่าวมาข้างต้นจะทำรวมกันเป็นวงจรได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 แสดงการรวมของวงจรเข้ารหัสกับวงจรถอดรหัสและวงจรการวัดค่าความผิดพลาดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

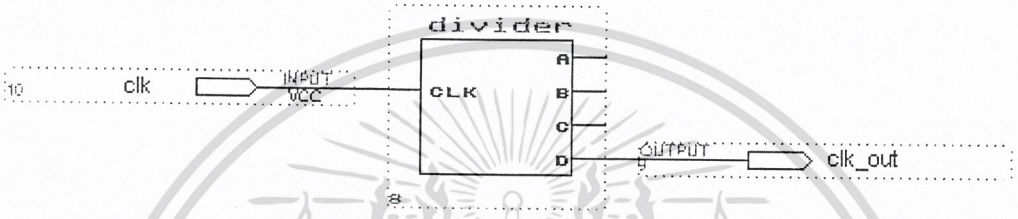
บทที่ 4

การทดลองและผลการทดลอง

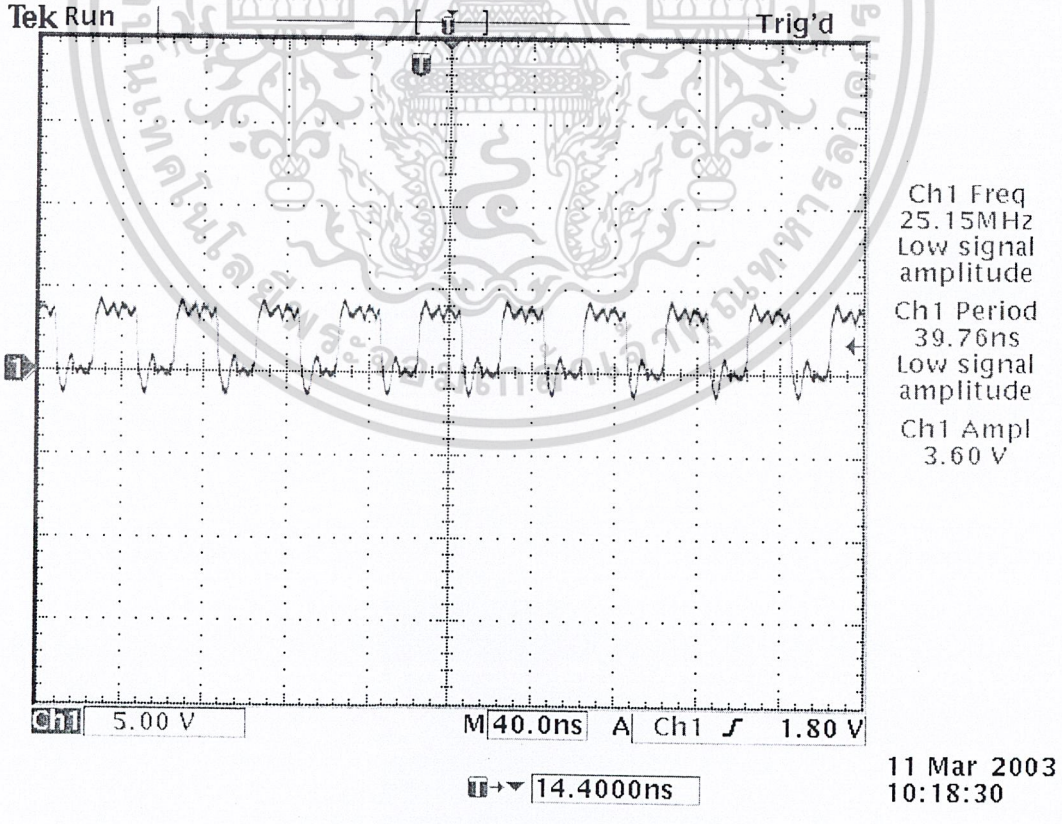
จากการออกแบบส่วนของตัวเข้ารหัสแบบคอนโวลูชั่นโค้ด ทำการออกแบบแต่ละส่วนให้ทำงานตามที่ต้องการ โดยใช้ภาษาวีเอชดีแอลในการเขียนชุดคำสั่ง รวมถึงใช้การออกแบบในแบบกราฟฟิก ผ่านทางโปรแกรม MAX+PLUS II แล้วทำการโปรแกรมลงบนชิพอุปกรณ์เอฟพีจีเอให้ได้ผลตามที่ออกแบบ

4.1 ส่วนของการเข้ารหัส

4.1.1 วงจรลดค่าความถี่ สามารถสังเคราะห์ได้ โดยมีสัญลักษณ์ดังนี้

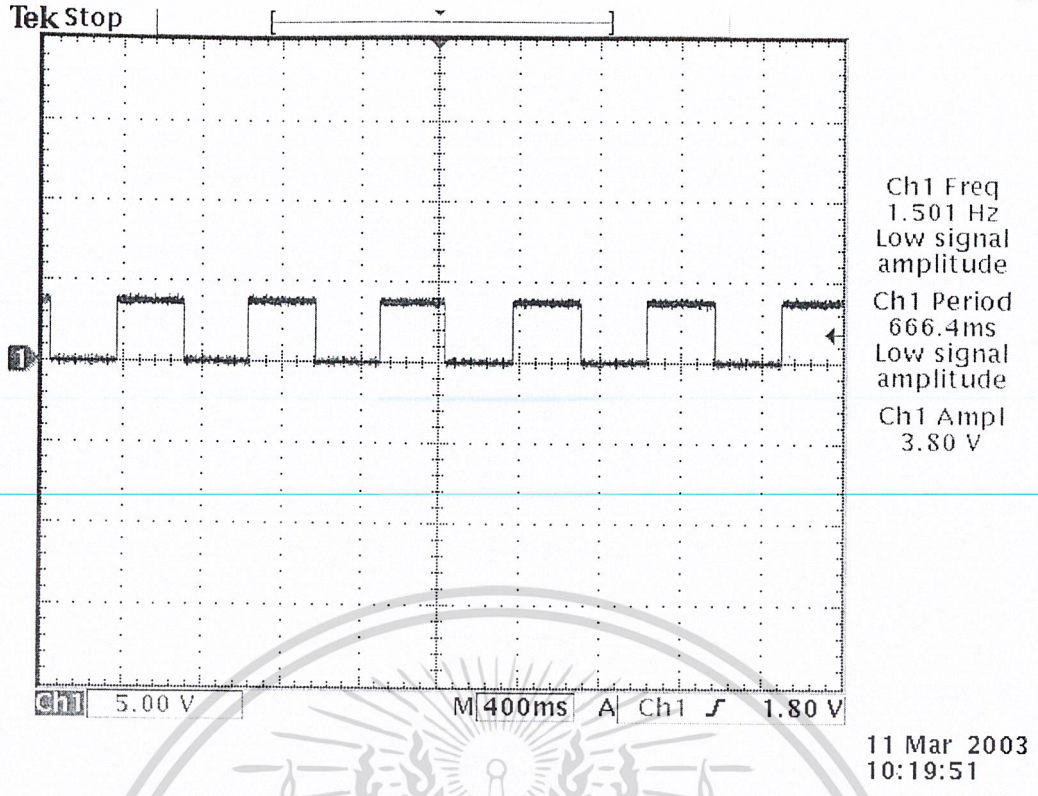


รูปที่ 4.1 แสดงสัญลักษณ์ของวงจรลดค่าความถี่ที่ได้จากการคอมไพล์

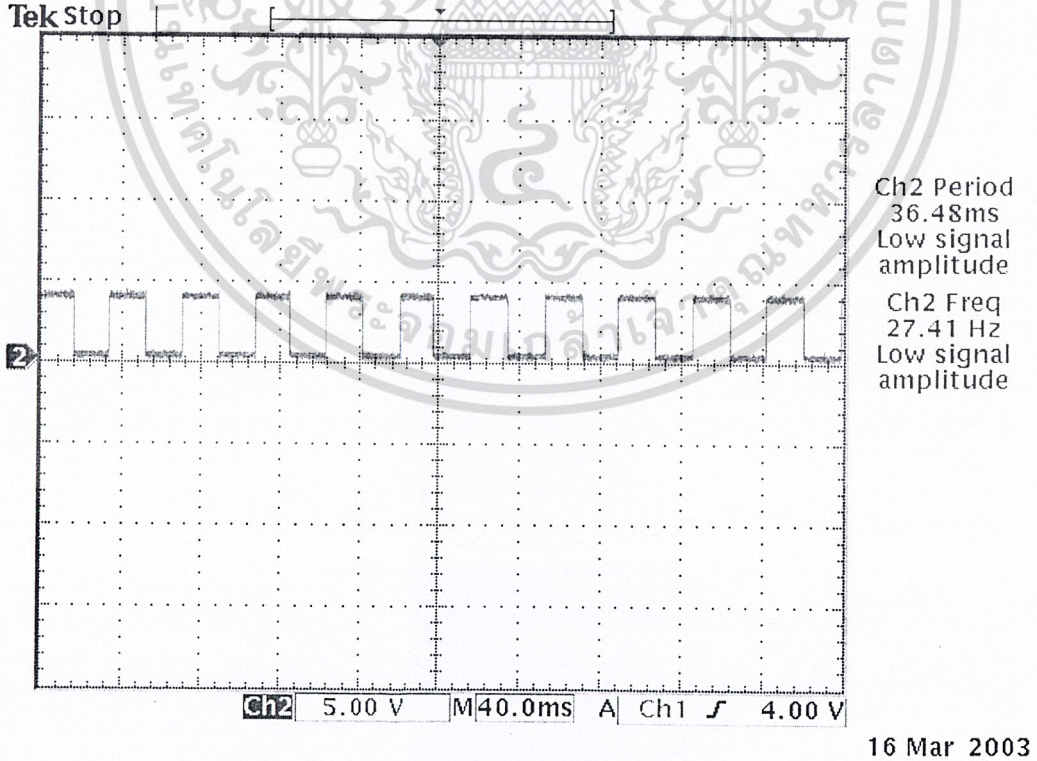


รูปที่ 4.2 แสดงส่วนของอินพุตของวงจรลดค่าความถี่(สัญญาณนาฬิกาจากบอร์ดเอฟพีจีเอ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงส่วนของเอาต์พุตของวงจรลดค่าความถี่(ใช้เปรียบเทียบสัญญาณที่เกิดจากการเข้ารหัส)

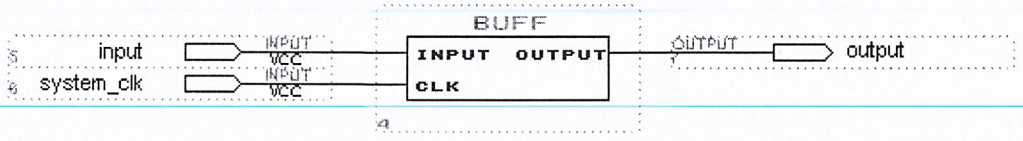


รูปที่ 4.4 แสดงส่วนของเอาต์พุตของวงจรลดค่าความถี่(ใช้ในการวัดค่าความผิดพลาดของข้อมูล)

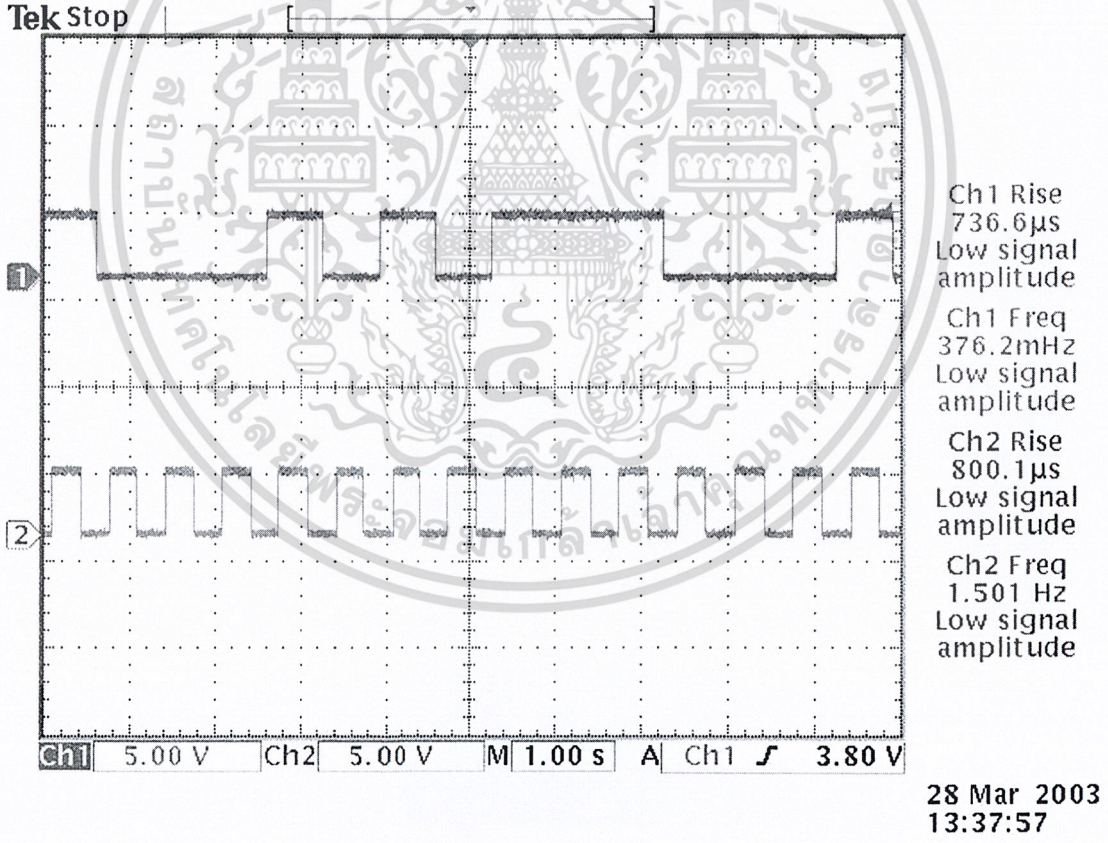
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองในส่วนของวงจรลดค่าความถี่ที่ต้องใช้สัญญาณนาฬิกา 2 ความถี่ เพื่อใช้ในการเปรียบเทียบข้อมูลที่เข้ารหัสแล้วและใช้ในการวัดค่าความผิดพลาดของข้อมูล

4.1.2 วงจรบัฟเฟอร์ สามารถสังเคราะห์ได้ โดยมีสัญลักษณ์ดังนี้

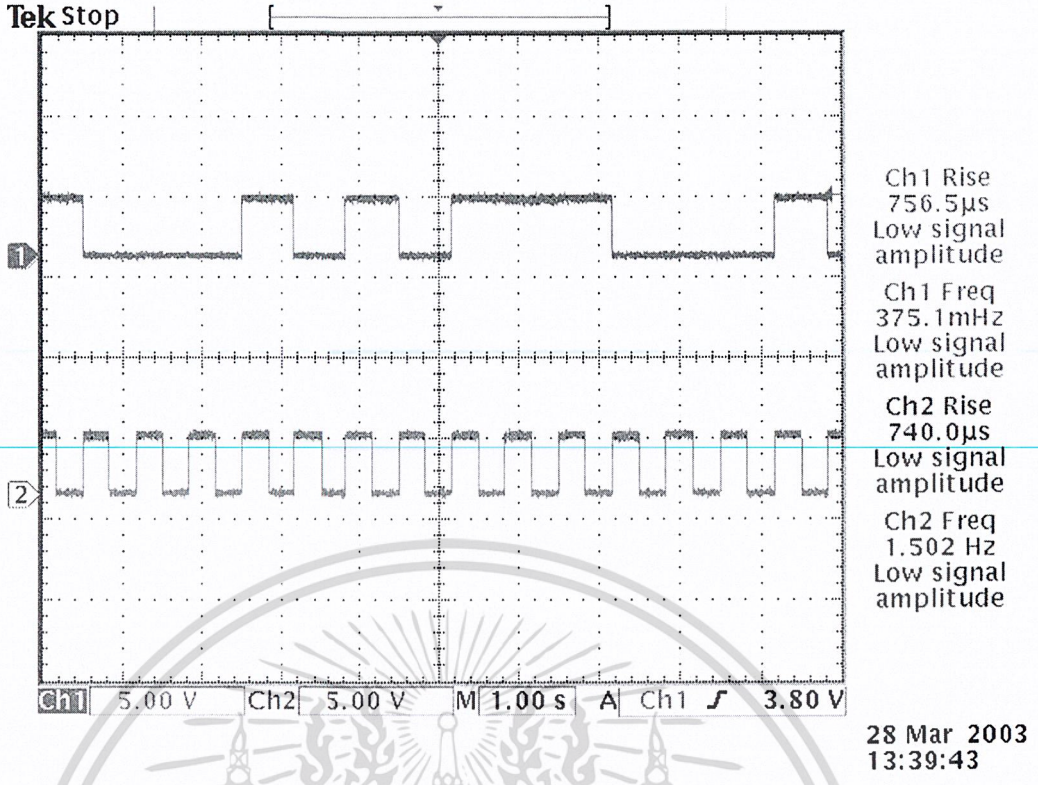


รูปที่ 4.5 แสดงสัญลักษณ์ของวงจรบัฟเฟอร์ที่ได้จากการคอมไพล์

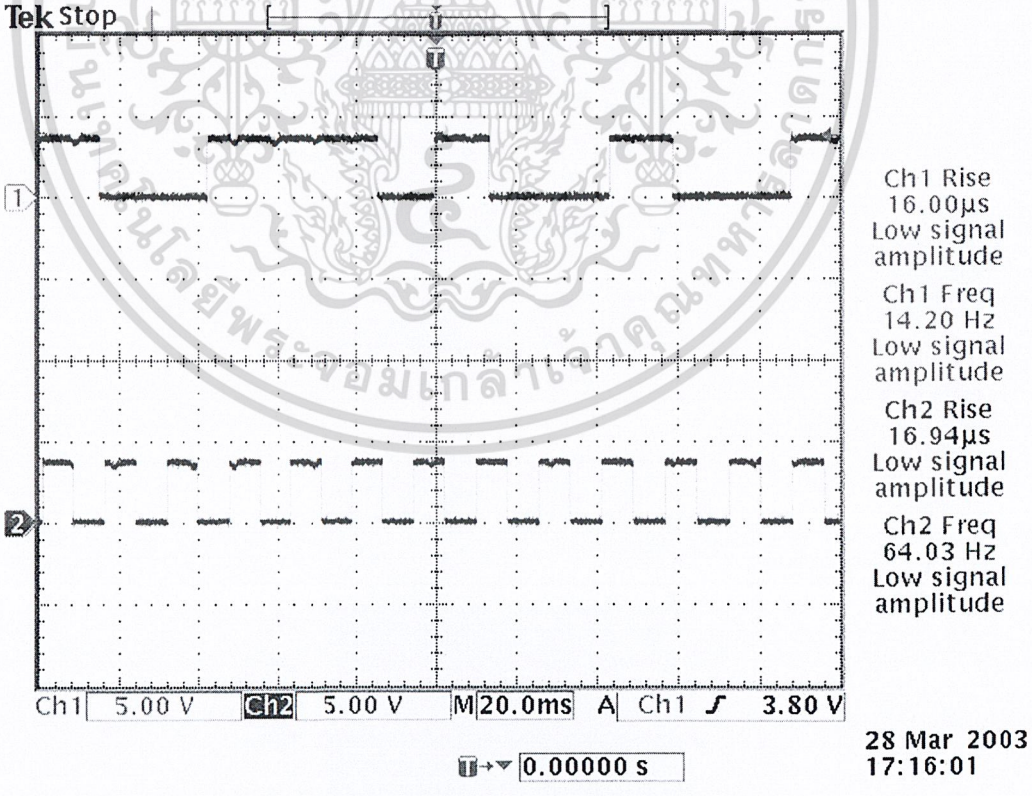


รูปที่ 4.6 แชนแนล 1 แสดงส่วนของอินพุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่แสดงสัญญาณนาฬิกาที่ความถี่ 1.501 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

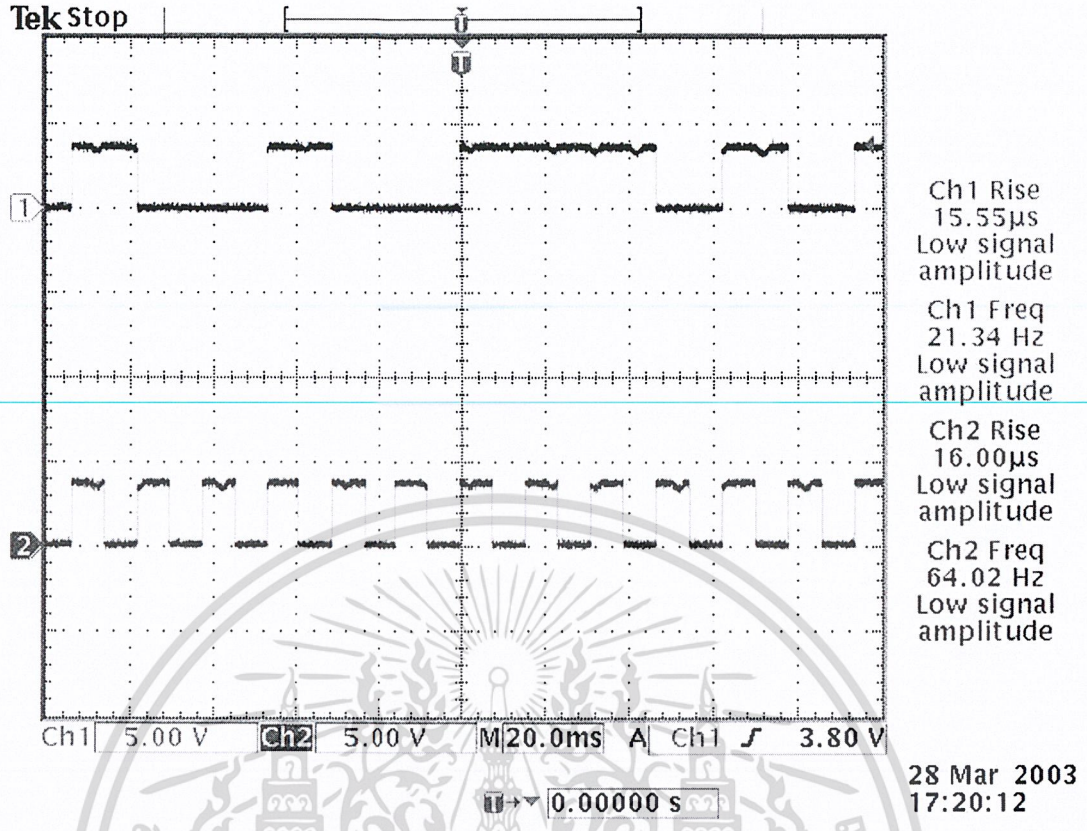


รูปที่ 4.7 แชนแนล 1 แสดงส่วนของเอาท์พุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่แสดงสัญญาณนาฬิกาความถี่ 1.501 Hz



รูปที่ 4.8 แชนแนล 1 แสดงส่วนของอินพุตของวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่สัญญาณนาฬิกาที่ความถี่ 64.03 Hz

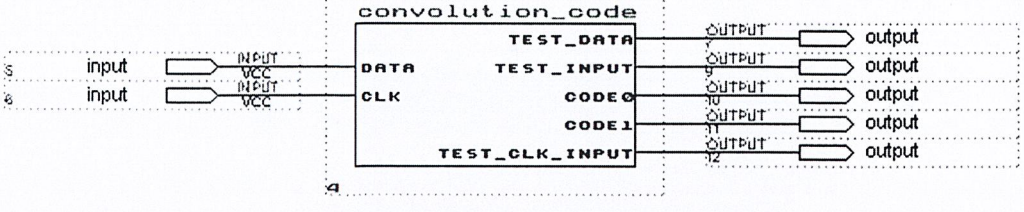
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แชนแนล 1 แสดงส่วนของเอาท์พุตวงจรบัฟเฟอร์เทียบกับแชนแนล 2 ที่สัญญาณนาฬิกาที่ความถี่ 64.02 Hz

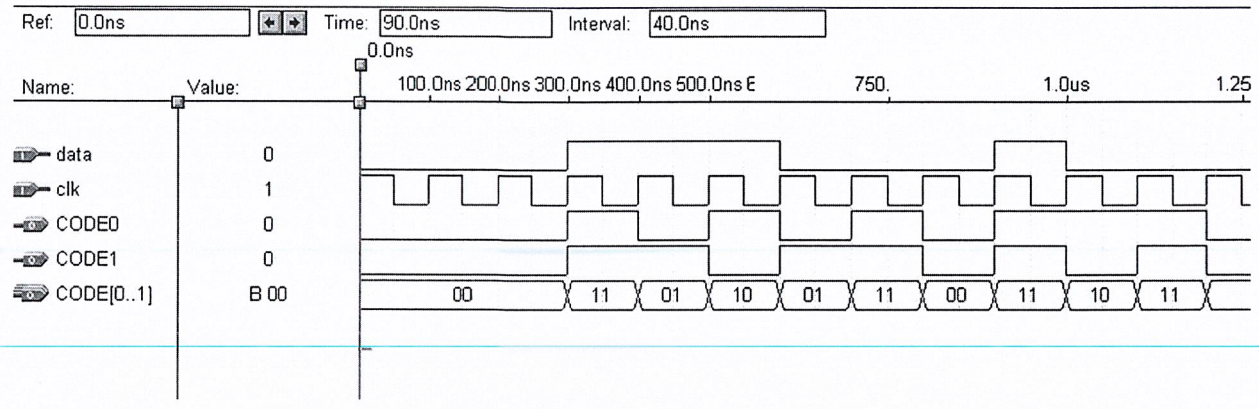
4.1.3 วงจรเข้ารหัส สามารถสังเคราะห์ได้ โดยมีสัญลักษณ์ดังนี้

ในส่วนของการเข้ารหัสจะทำการเปรียบเทียบระหว่างค่าที่ได้จากการจำลองการทำงานตามทฤษฎีกับค่าที่ได้จากการออกแบบ

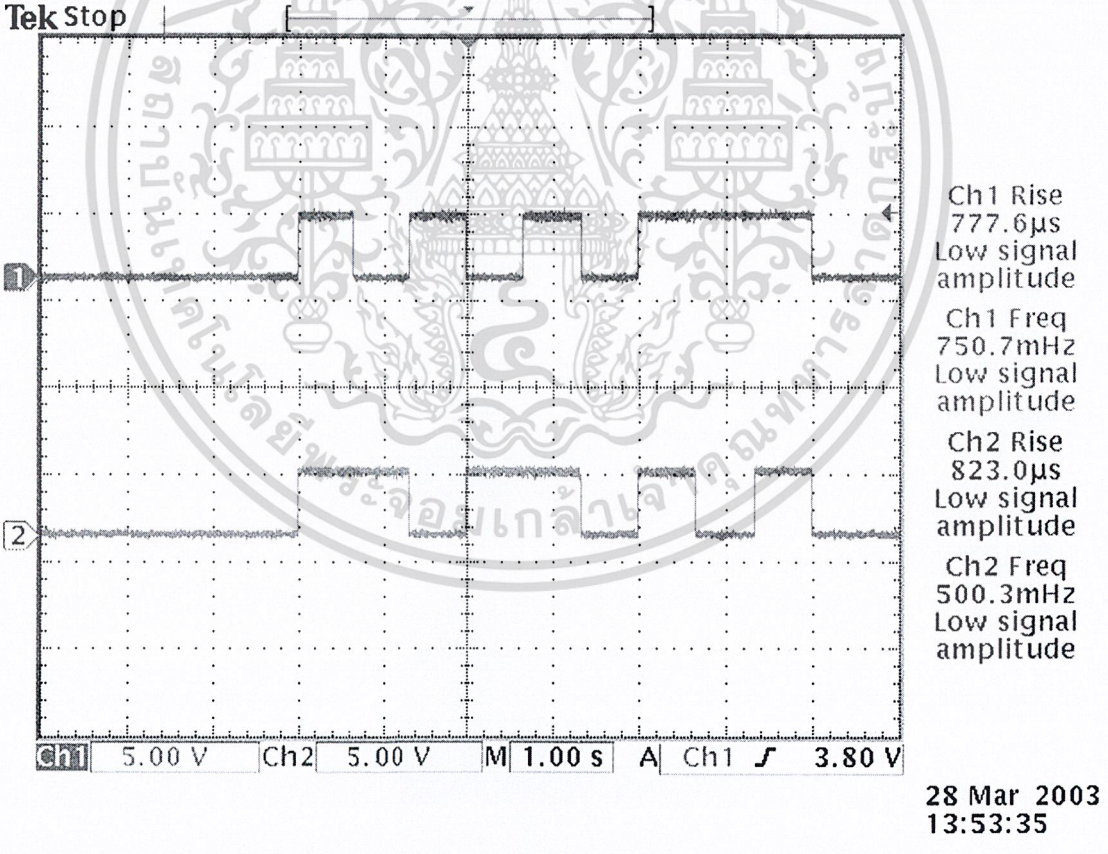


รูปที่ 4.10 แสดงสัญลักษณ์ของวงจรเข้ารหัสที่ได้จากการคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

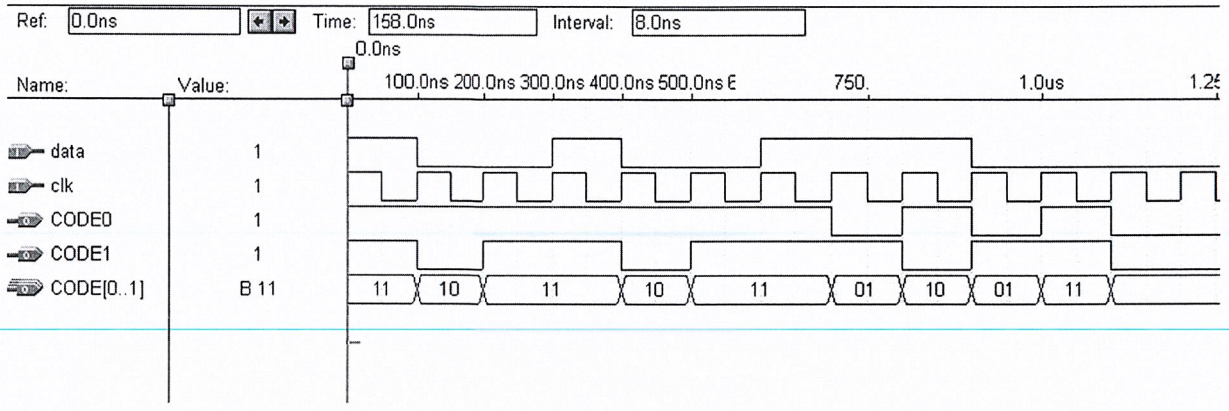


รูปที่ 4.11 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 0001110001

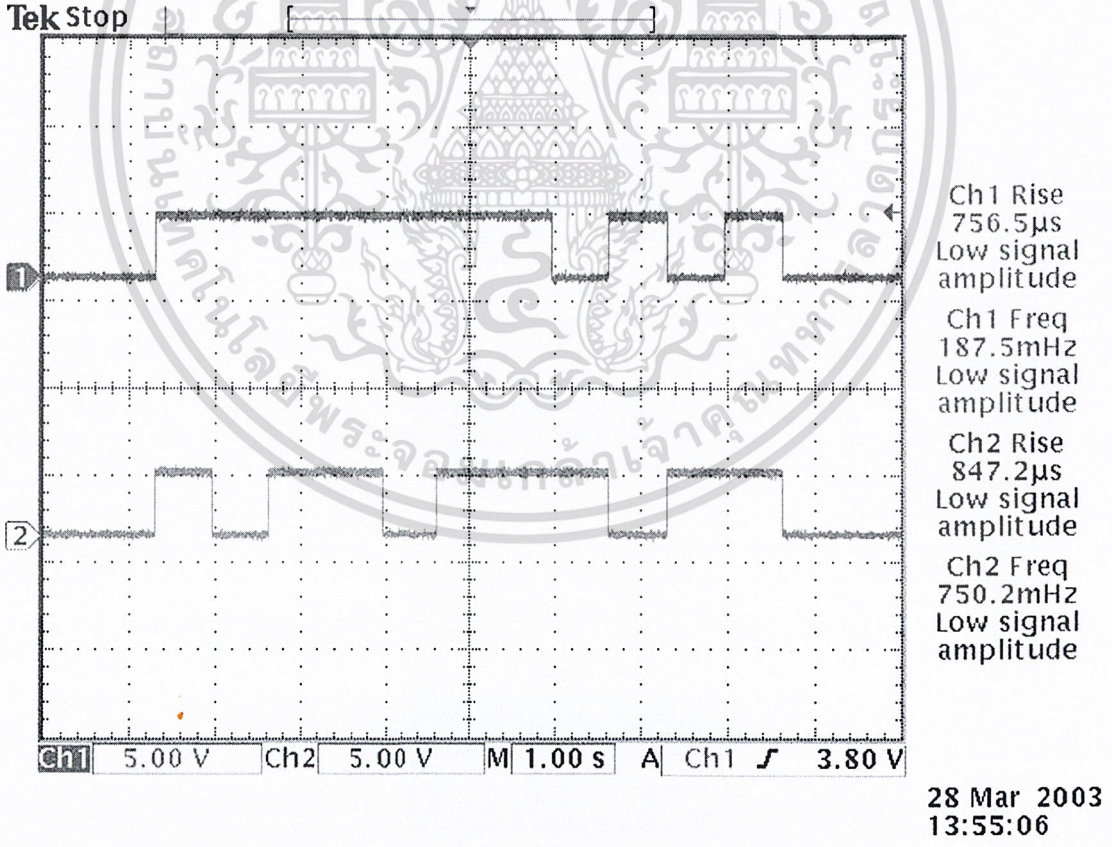


รูปที่ 4.12 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1ของการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 0001110001

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ทำซ้ำ ดัดแปลง หรือนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น โดยไม่ได้รับอนุญาตจากมหาวิทยาลัย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

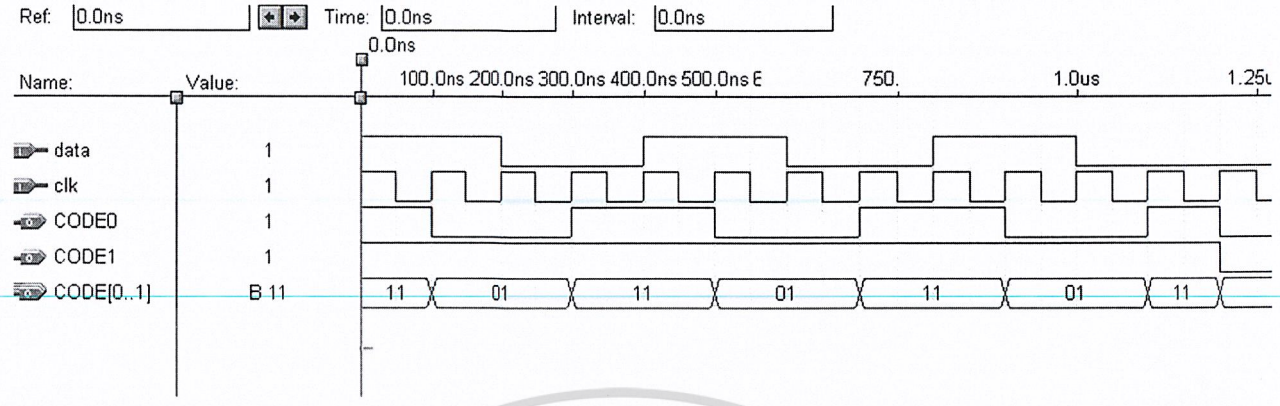


รูปที่ 4.13 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1001001110

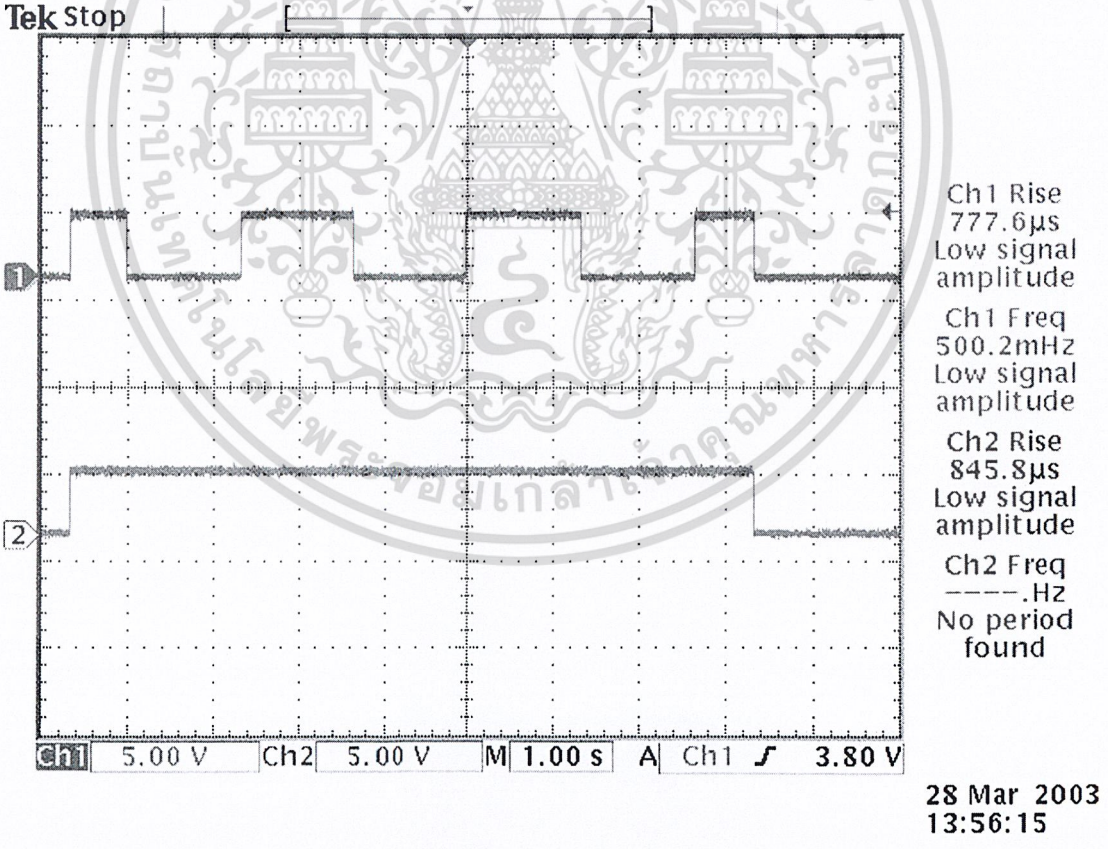


รูปที่ 4.14 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1ของการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1001001110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงการจำลองการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1100110011

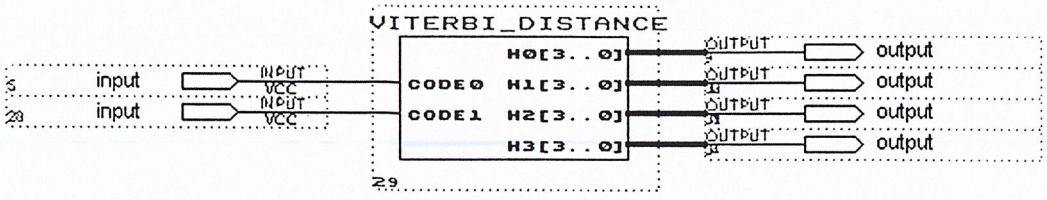


รูปที่ 4.16 แชนแนล 1 แสดงโค้ดเวิร์ด 0 และ แชนแนล 2 แสดงโค้ดเวิร์ด 1 ของการทำงานของวงจรเข้ารหัสเมื่อป้อนอินพุต 1100110011

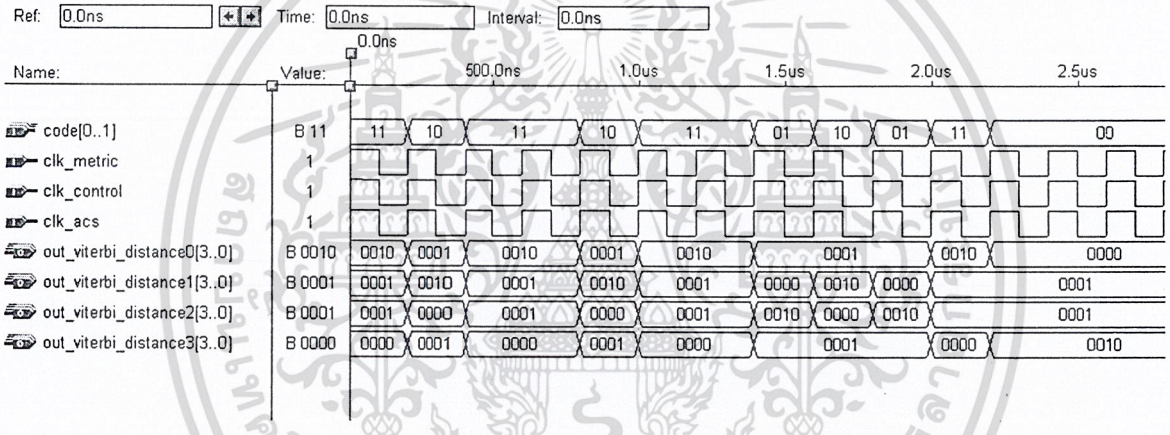
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของการถอดรหัส

4.2.1 วงจรไวตาบิดิสแตนซ์

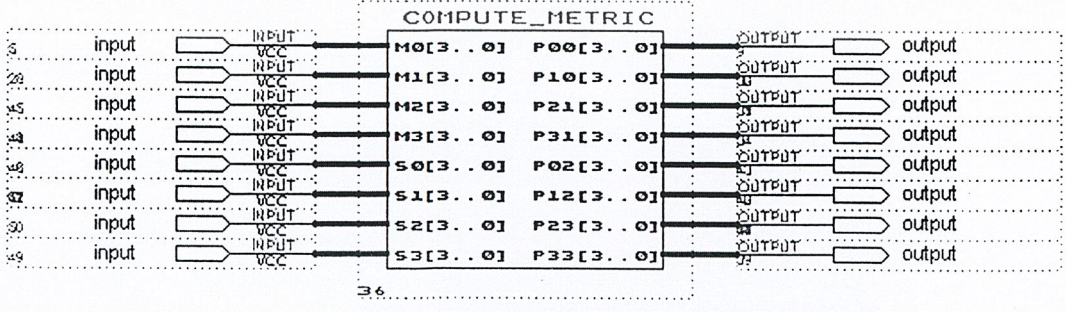


รูปที่ 4.17 แสดงสัญลักษณ์ของวงจรไวตาบิดิสแตนซ์ ที่ได้จากการคอมไพล์



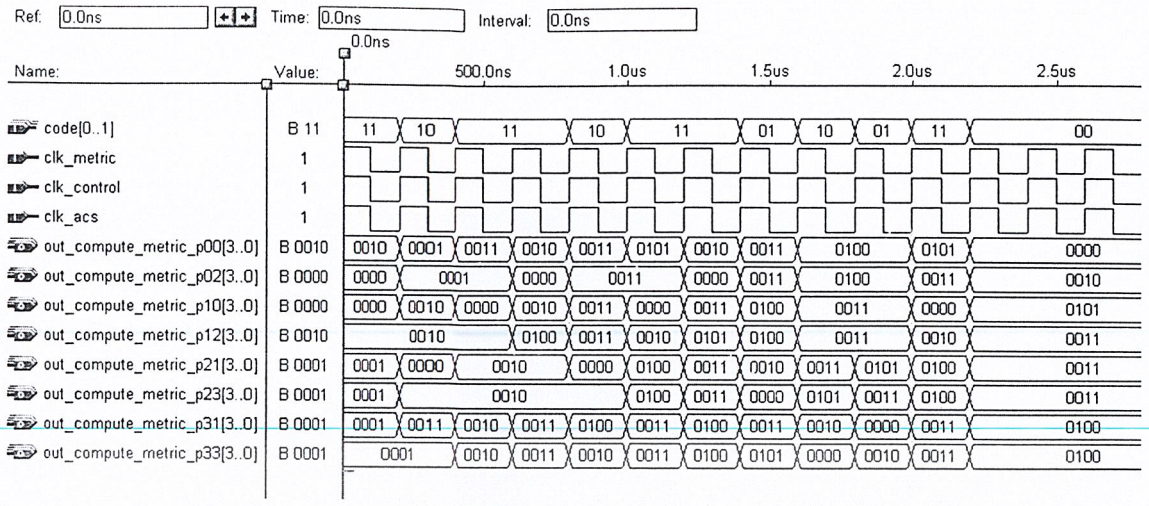
รูปที่ 4.18 แสดงการจำลองการทำงานของวงจรไวตาบิดิสแตนซ์เมื่อป้อนอินพุต (111011111011101100111)

4.2.2 วงจรคอมพิวเมตริกซ์



รูปที่ 4.19 แสดงสัญลักษณ์ของวงจรคอมพิวเมตริกซ์ที่ได้จากการคอมไพล์

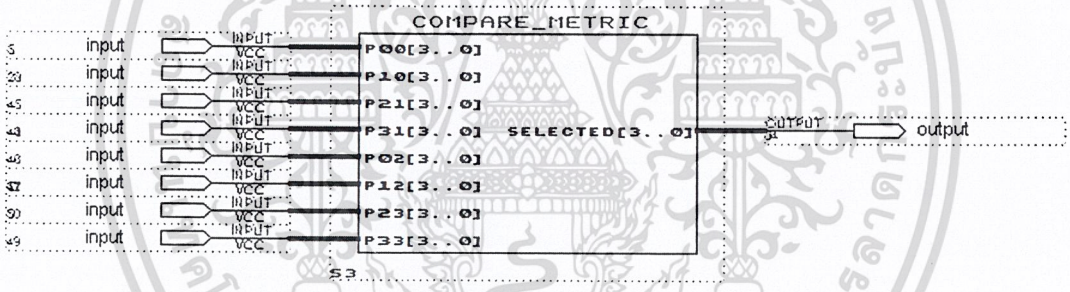
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



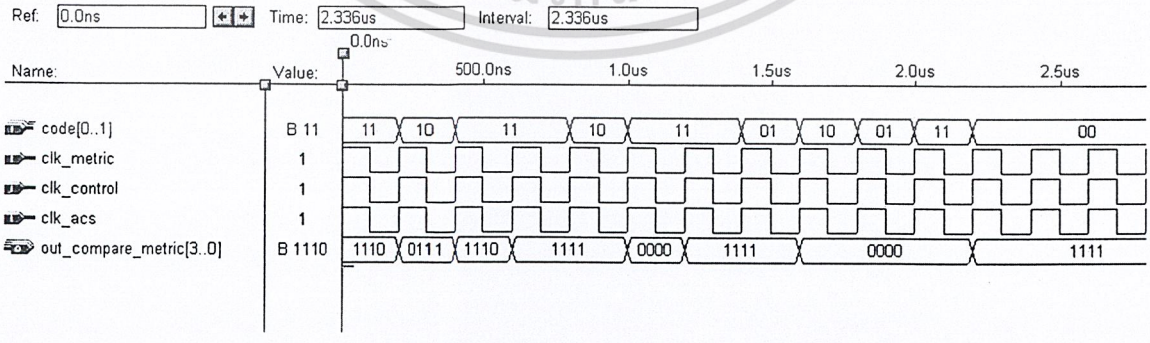
รูปที่ 4.20 แสดงการจำลองการทำงานของวงจรคอมพิวเมตริกซ์เมื่อป้อนอินพุต

(11101111101111011100111)

4.2.3 วงจรคอมแพร์เมตริกซ์



รูปที่ 4.21 แสดงสัญลักษณ์ของวงจรคอมแพร์เมตริกซ์ที่ได้จากการคอมไพล์

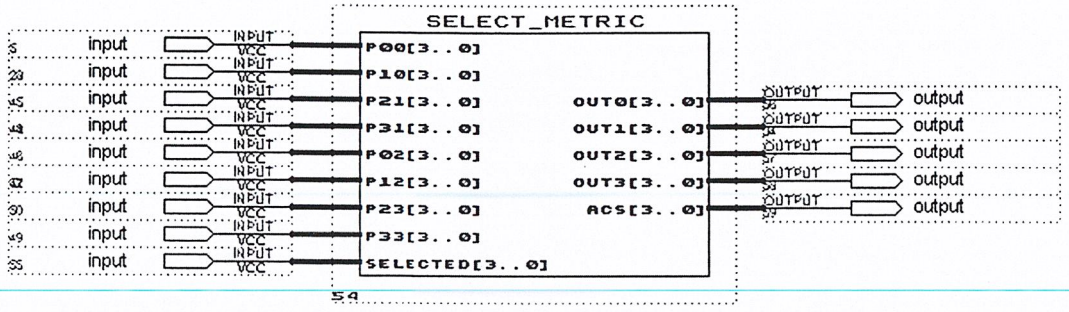


รูปที่ 4.22 แสดงการจำลองการทำงานของวงจรคอมพิวเมตริกซ์เมื่อป้อนอินพุต

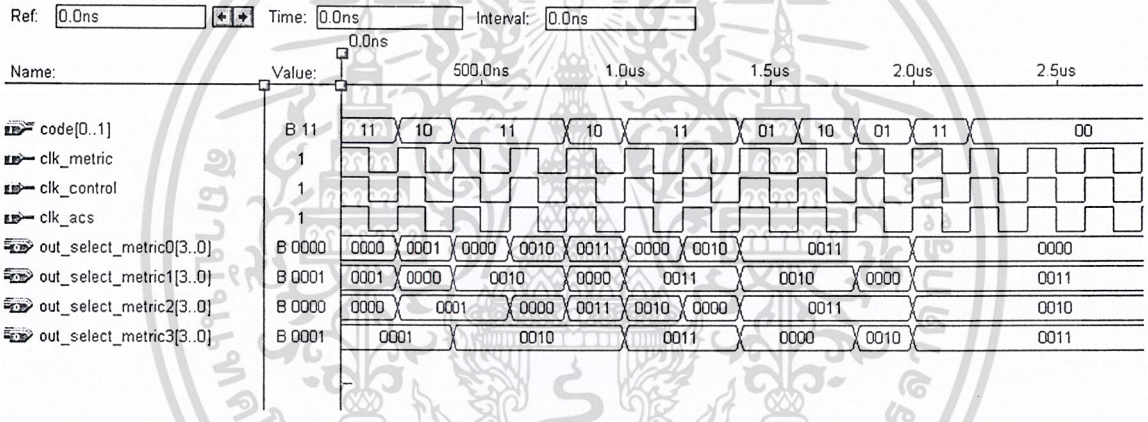
(11101111101111011100111)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 วงจรซีเลกเมตริกซ์

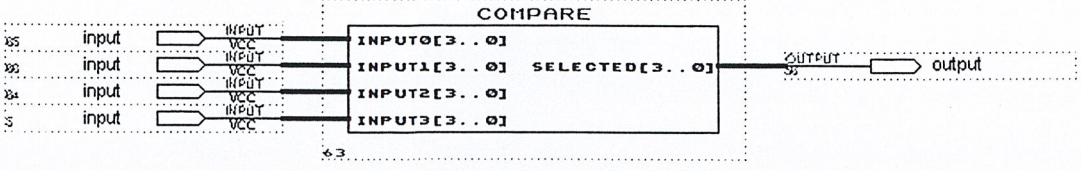


รูปที่ 4.23 แสดงสัญลักษณ์ของวงจรซีเลกเมตริกซ์ที่ได้จากการคอมไพล์



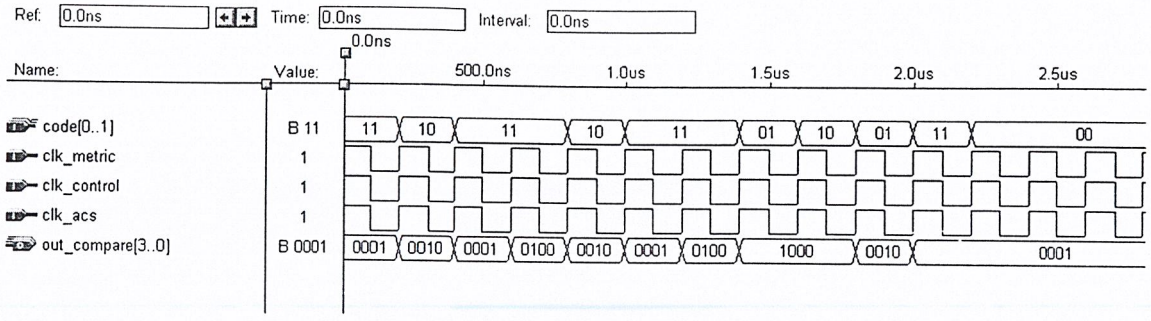
รูปที่ 4.24 แสดงการจำลองการทำงานของวงจรซีเลกเมตริกซ์เมื่อป้อนอินพุต (111011111011101100111)

4.2.5 วงจรคอมแพร์



รูปที่ 4.25 แสดงสัญลักษณ์ของวงจรคอมแพร์ที่ได้จากการคอมไพล์

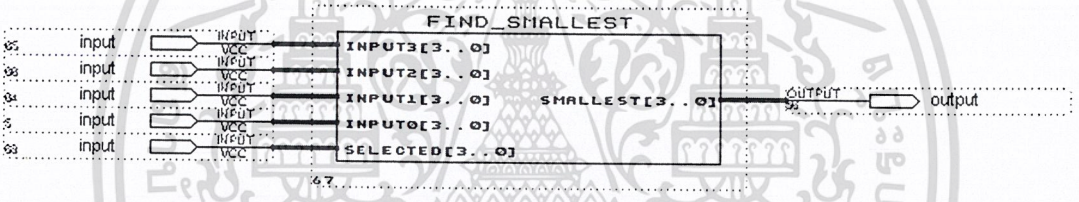
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



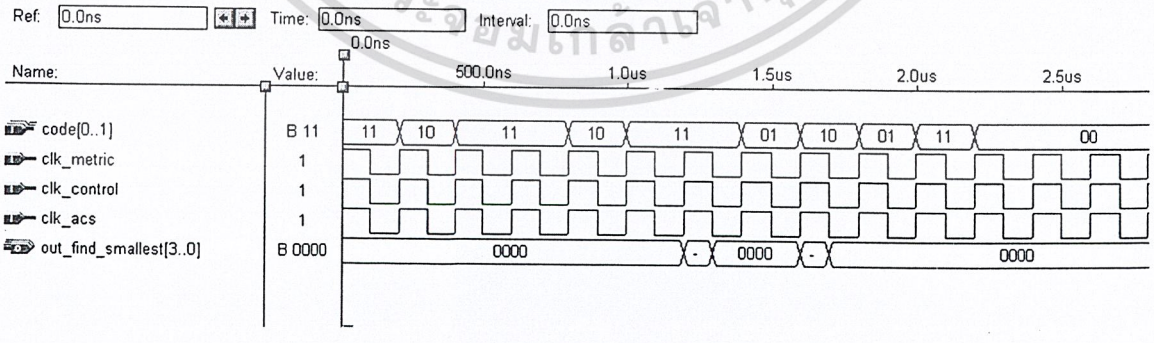
รูปที่ 4.26 แสดงการจำลองการทำงานของวงจรคอมแพร์เมื่อป้อนอินพุต

(11101111101111011001111)

4.2.6 วงจรไฟน์สมอลเลท



รูปที่ 4.27 แสดงสัญลักษณ์ของวงจรไฟน์สมอลเลทที่ได้จากการคอมไพล์

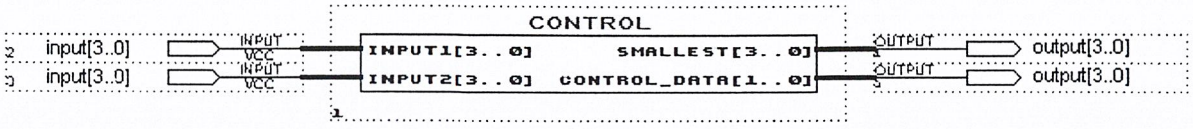


รูปที่ 4.28 แสดงการจำลองการทำงานของวงจรไฟน์สมอลเลทเมื่อป้อนอินพุต

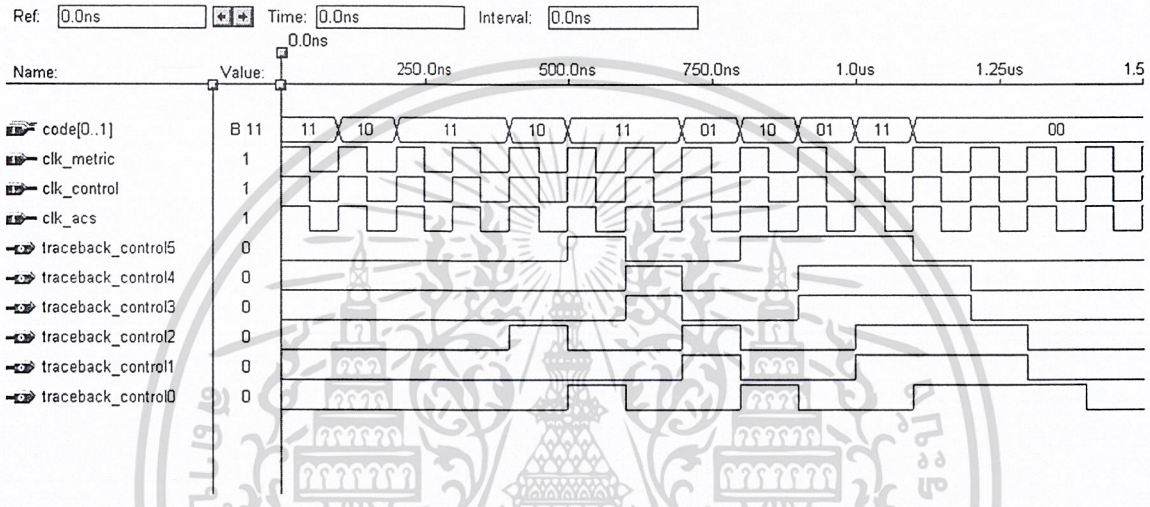
(11101111101111011001111)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 วงจรคอนโทรล

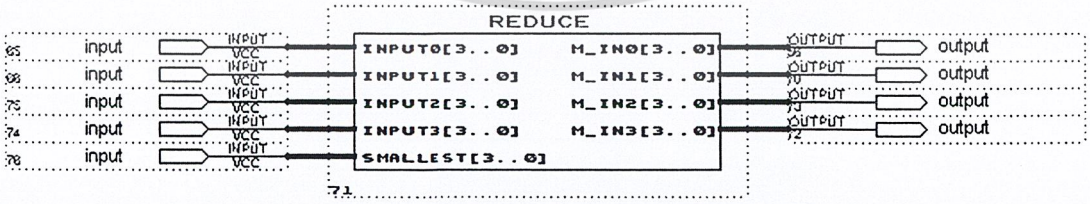


รูปที่ 4.29 แสดงสัญลักษณ์ของวงจรคอนโทรลที่ได้จากการคอมไพล์



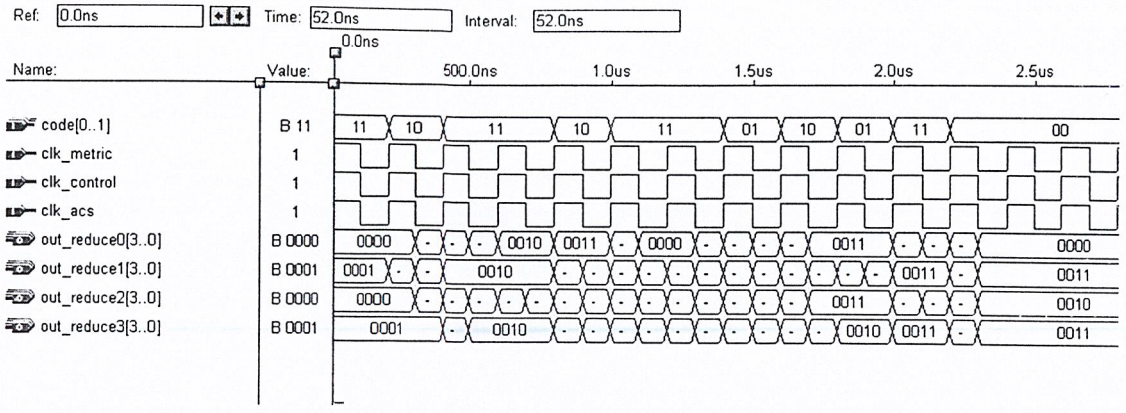
รูปที่ 4.30 แสดงการจำลองการทำงานของวงจรคอนโทรลเมื่อป้อนอินพุต
(111011111101111011100111)

4.2.8 วงจรรีดิวซ์



รูปที่ 4.31 แสดงสัญลักษณ์ของวงจรรีดิวซ์ที่ได้จากการคอมไพล์

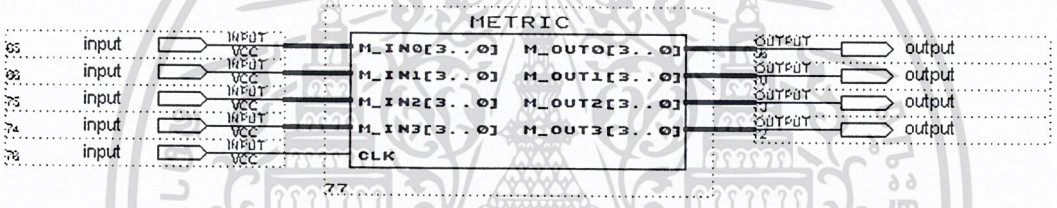
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



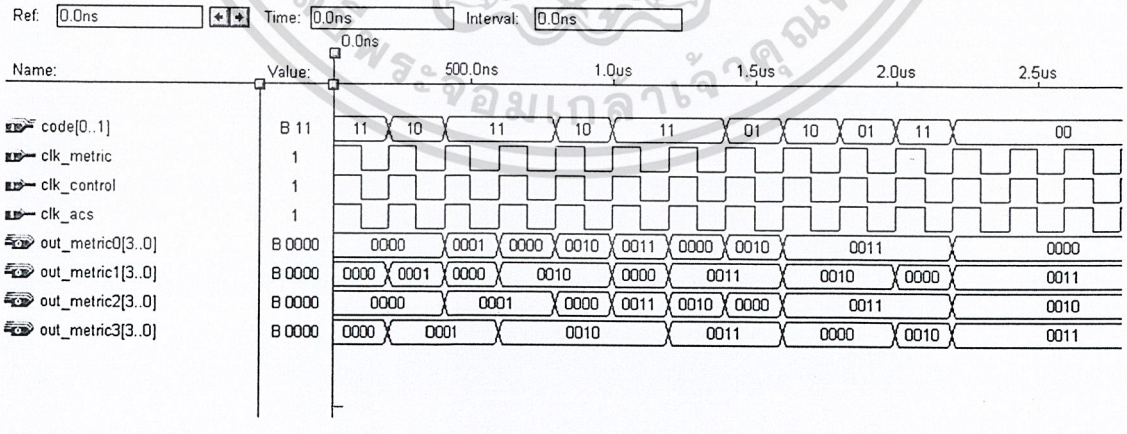
รูปที่ 4.32 แสดงการจำลองการทำงานของวงจรรีดิวซ์เมื่อป้อนอินพุต

(11101111101111101100111)

4.2.9 วงจรเมตริกซ์



รูปที่ 4.33 แสดงสัญลักษณ์ของวงจรเมตริกซ์ที่ได้จากการคอมไพล์

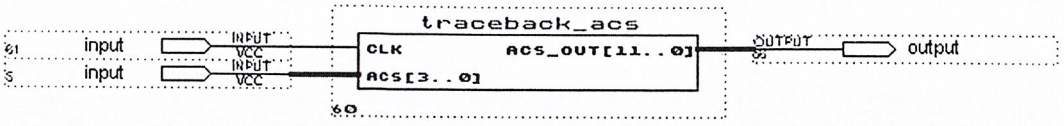


รูปที่ 4.34 แสดงการจำลองการทำงานของวงจรรีดิวซ์เมื่อป้อนอินพุต

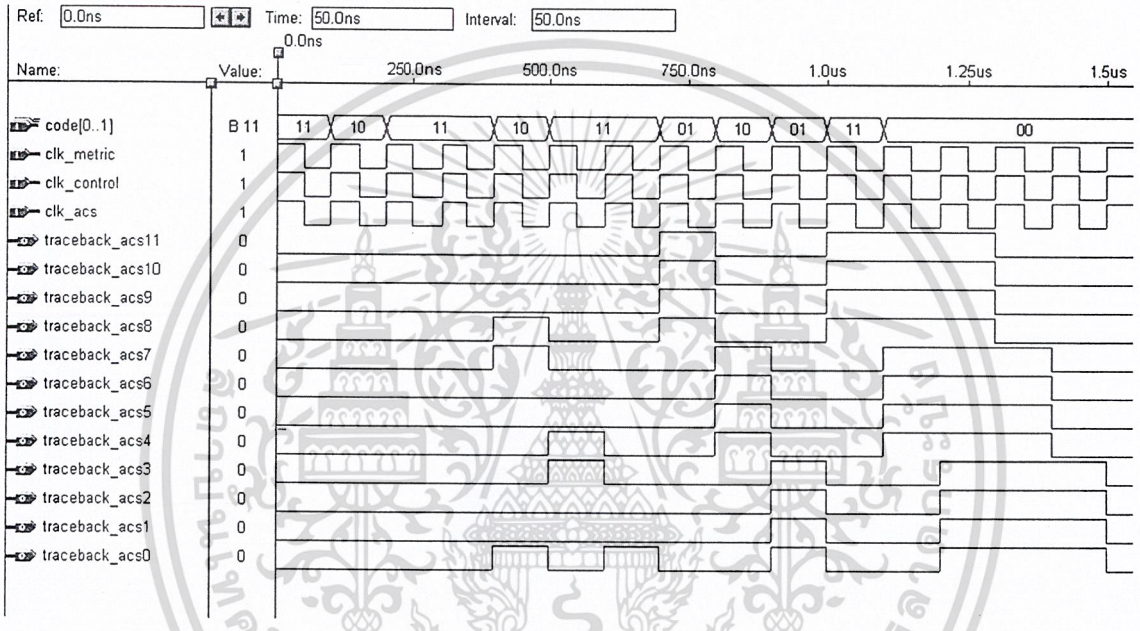
(11101111101111101100111)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.10 วงจรเทรดแบคเอซีเอส

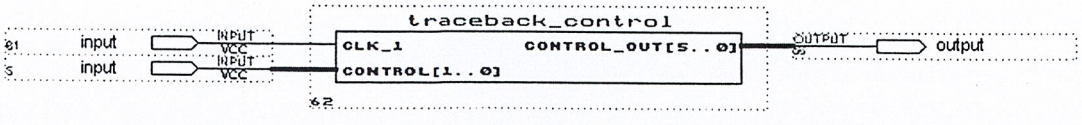


รูปที่ 4.35 แสดงสัญลักษณ์ของวงจรเทรดแบคเอซีเอสที่ได้จากการคอมไพล์



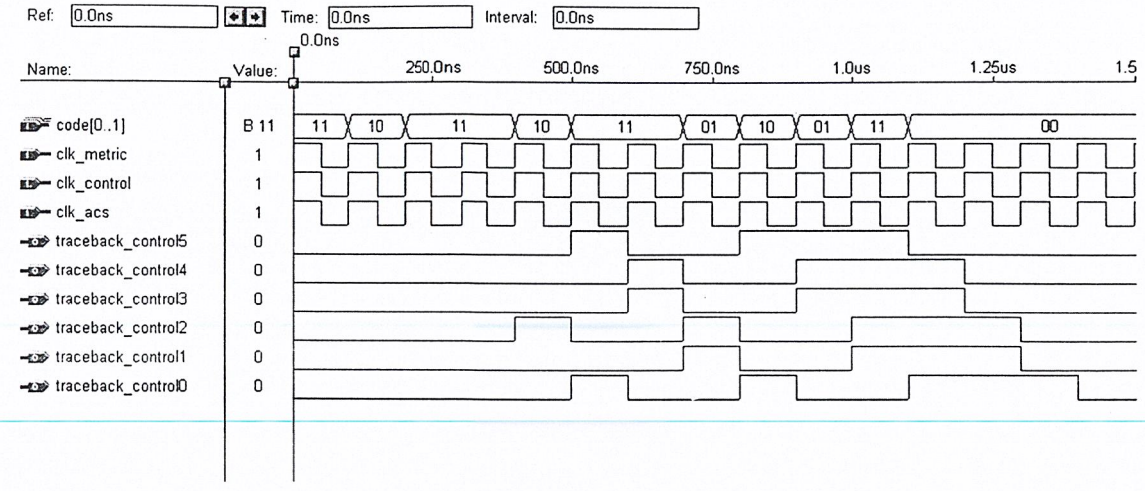
รูปที่ 4.36 แสดงการจำลองการทำงานของวงจรเทรดแบคเอซีเอสมื่อป้อนอินพุต
(111011111010111101100111)

4.2.11 วงจรเทรดแบคคอนโทรล



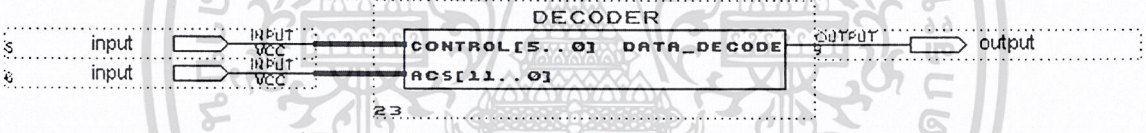
รูปที่ 4.37 แสดงสัญลักษณ์ของวงจรเทรดแบคคอนโทรลที่ได้จากการคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

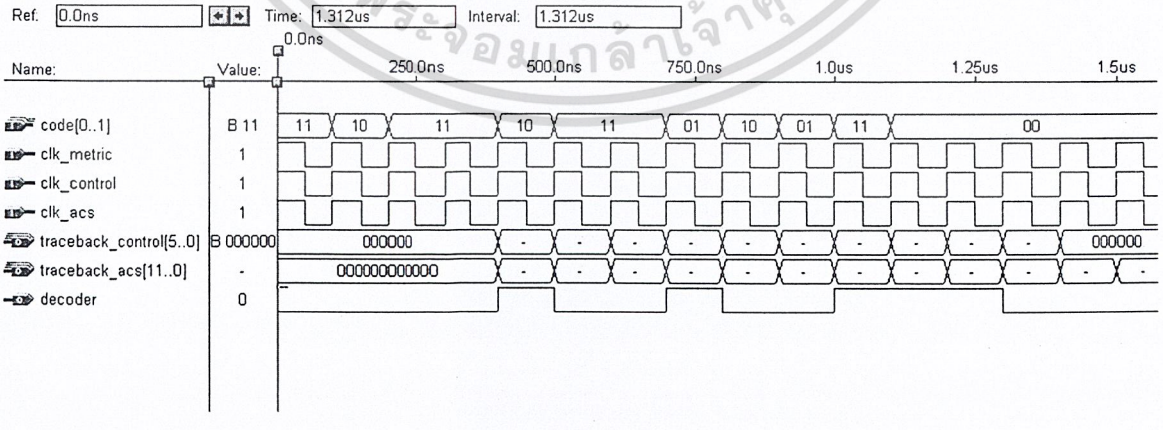


รูปที่ 4.38 แสดงการจำลองการทำงานของวงจรเทรดแบคคอนโทรลเมื่อป้อนอินพุต (111011111011111011100111)

4.2.12 วงจรถอดรหัส



รูปที่ 4.39 แสดงสัญลักษณ์ของวงจรถอดรหัสที่ได้จากการคอมไพล์

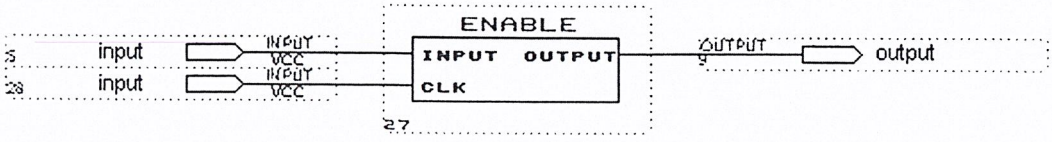


รูปที่ 4.40 แสดงการจำลองการทำงานของวงจรถอดรหัสเมื่อป้อนอินพุต (111011111011111011100111)

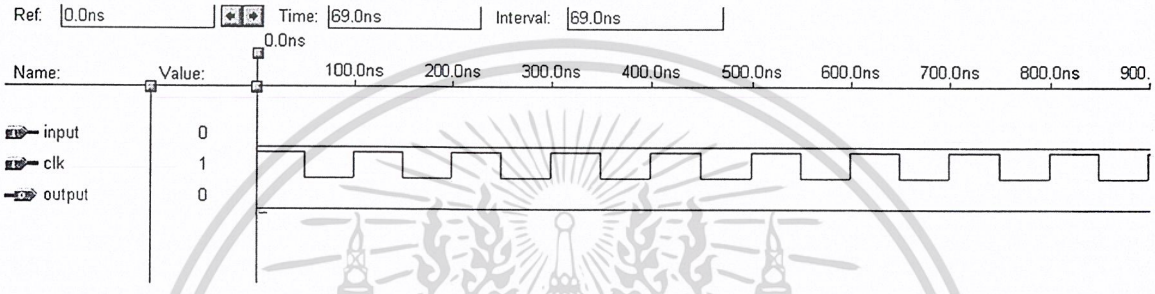
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ส่วนของการวัดค่าความผิดพลาดของข้อมูล

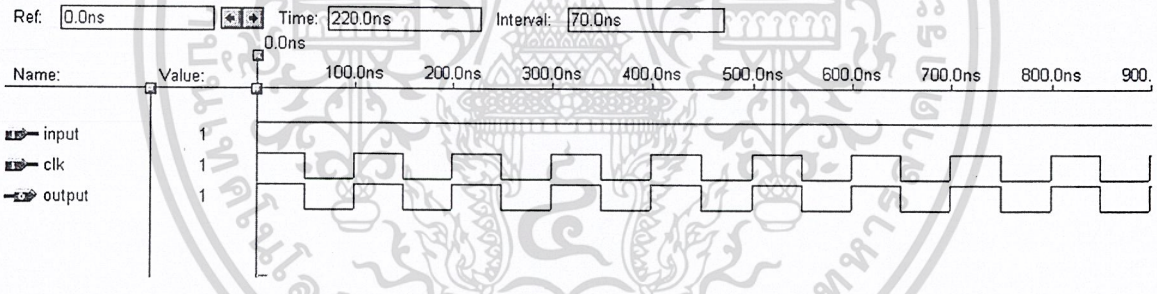
4.3.1 วงจรอีเนเบิล



รูปที่ 4.41 แสดงสัญลักษณ์ของวงจรอีเนเบิลที่ได้จากการคอมไพล์

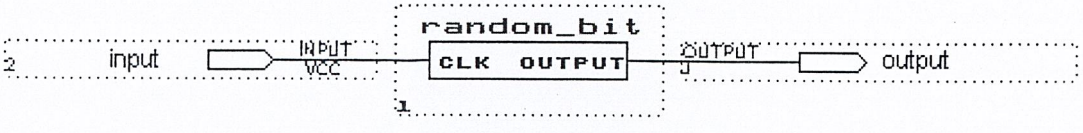


รูปที่ 4.42 แสดงการจำลองการทำงานของวงจรอีเนเบิลเมื่อป้อนอินพุต "0"



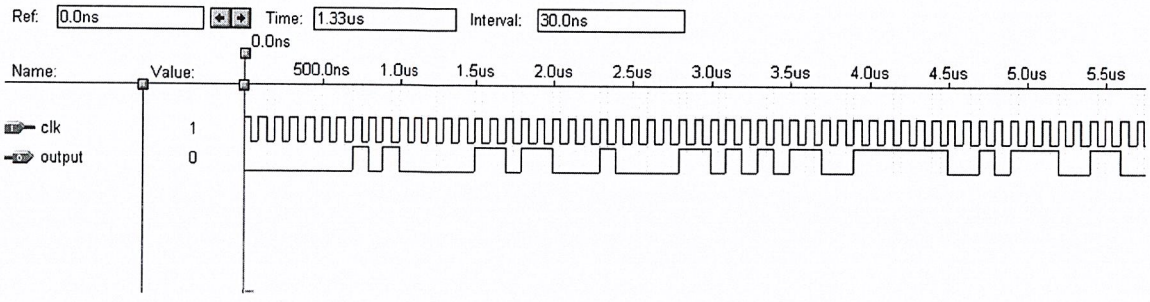
รูปที่ 4.43 แสดงการจำลองการทำงานของวงจรอีเนเบิลเมื่อป้อนอินพุต "1"

4.3.2 วงจรสุ่มค่า

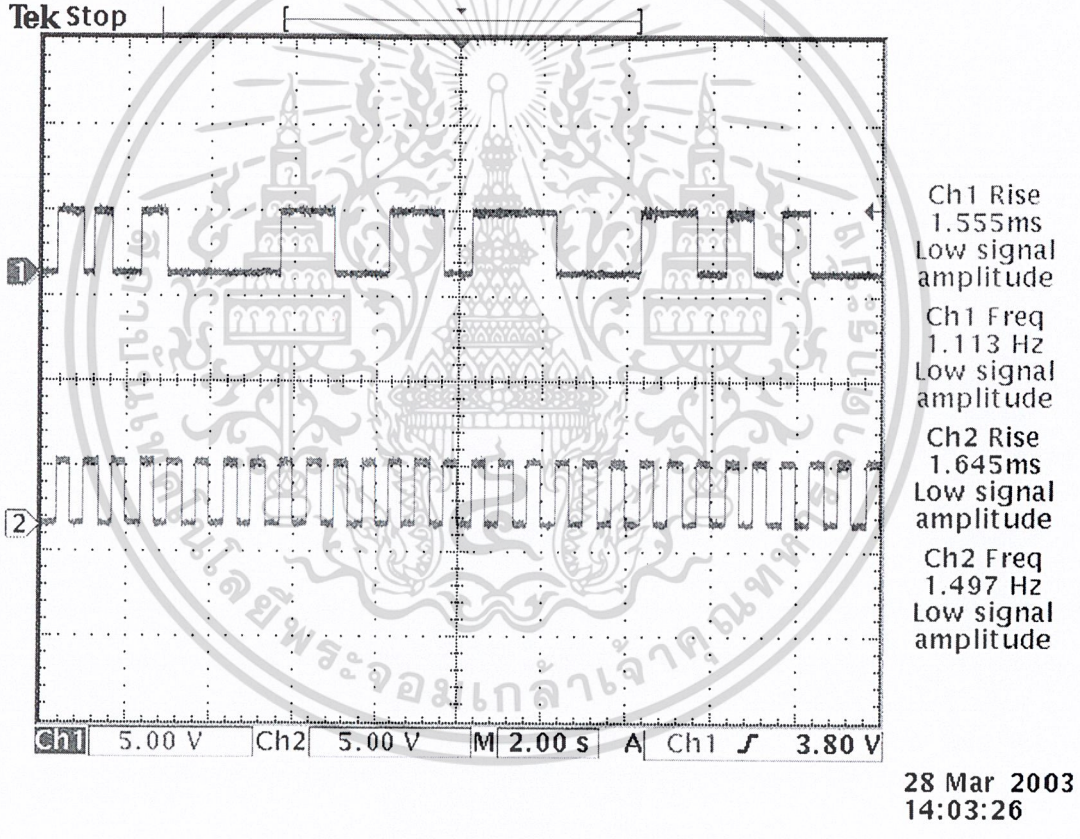


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.44 แสดงสัญลักษณ์ของวงจรสุ่มค่าที่ได้จากการคอมไพล์



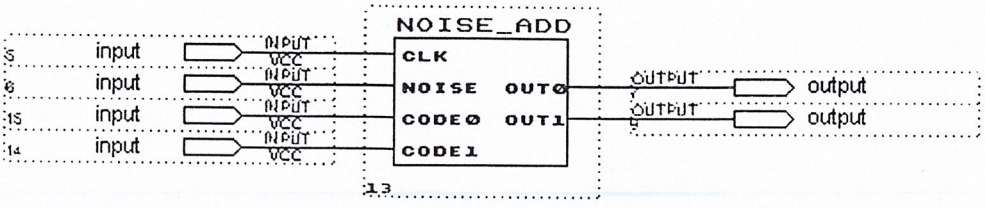
รูปที่ 4.45 แสดงการจำลองการทำงานของวงจรสุ่มค่าเมื่อเทียบกับสัญญาณนาฬิกา



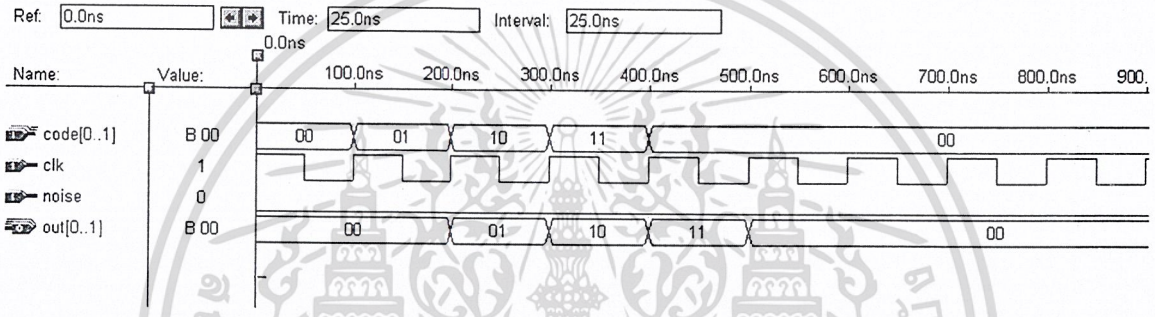
รูปที่ 4.46 แชนแนล 1 แสดงข้อมูลที่สุ่มค่า และ แชนแนล 2 แสดงสัญญาณนาฬิกาของวงจรสุ่มค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 วงจรจำลองการผิดพลาดของข้อมูล

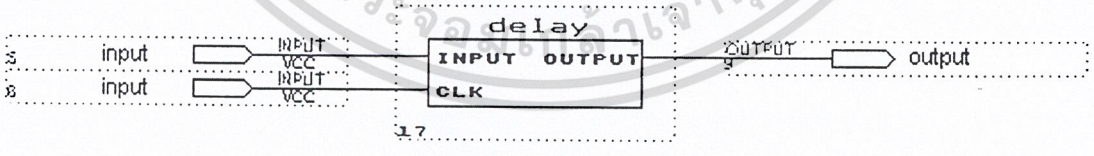


รูปที่ 4.47 แสดงสัญลักษณ์ของวงจรจำลองการผิดพลาดของข้อมูลที่ได้จากการคอมไพล์



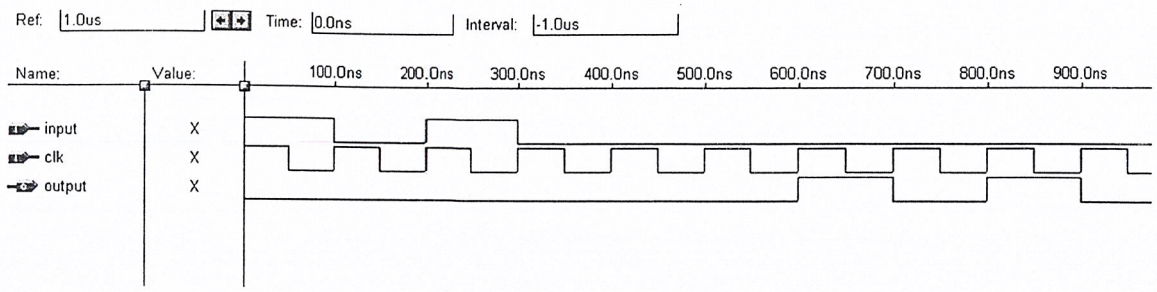
รูปที่ 4.48 แสดงการจำลองการทำงานของวงจรจำลองการผิดพลาดของข้อมูล

4.3.4 วงจรหน่วงเวลา



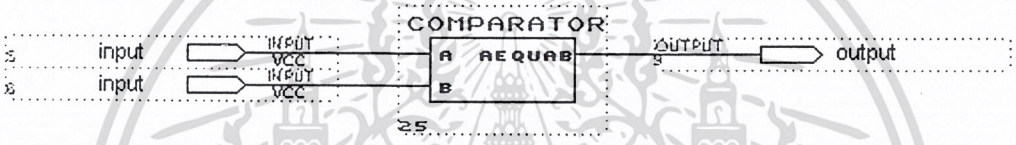
รูปที่ 4.49 แสดงสัญลักษณ์ของวงจรหน่วงเวลาที่ได้จากการคอมไพล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

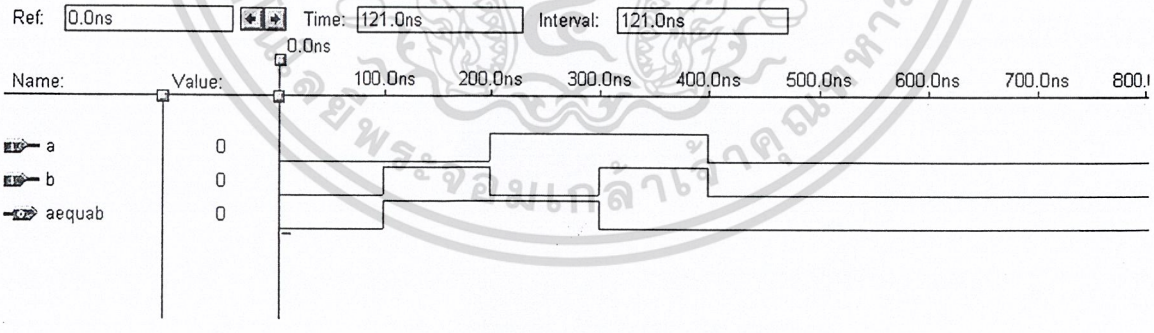


รูปที่ 4.50 แสดงการจำลองการทำงานของวงจรหน่วงเวลาเมื่อป้อนอินพุตบิต 101

4.3.5 วงจรเปรียบเทียบสัญญาณ (Comparator)



รูปที่ 4.51 แสดงสัญลักษณ์ของวงจรเปรียบเทียบสัญญาณที่ได้จากการคอมไพล์



รูปที่ 4.52 แสดงการจำลองการทำงานของวงจรเปรียบเทียบสัญญาณเมื่ออินพุต a = 0011 และ อินพุต b = 0101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดลองของการวัดค่าความผิดพลาดของข้อมูล

เป็นการพิจารณาเปรียบเทียบเมื่อมีการจำลองให้ข้อมูลเกิดความผิดพลาดแล้วข้อมูลที่รับได้เมื่อเปรียบเทียบกับข้อมูลที่ส่งมาจริงมีค่าความผิดพลาดเป็นอย่างไรระหว่างข้อมูลที่ทำการเข้ารหัสและข้อมูลที่ไม่ได้ทำการเข้ารหัส โดยจะแบ่งการทดลองเป็น 4 ชุด โดยชุดที่ 1 – 3 จะส่งข้อมูลในลักษณะวนซ้ำข้อมูลเดิมทุกๆ 10 บิตและในชุดที่ 4 จะเป็นการส่งข้อมูลในลักษณะของการสุ่มค่า

0001110001		
ครั้งที่	จำนวนบิตที่เกิดความผิดพลาด	
	ข้อมูลที่เข้ารหัส	ข้อมูลที่ไม่เข้ารหัส
1	116288	291333
2	115570	291071
3	113941	291806
4	123338	298843
5	123074	297914

ตารางที่ 4.1 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูลที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 0001110001

1001001110		
ครั้งที่	จำนวนบิตที่เกิดความผิดพลาด	
	ข้อมูลที่เข้ารหัส	ข้อมูลที่ไม่เข้ารหัส
1	121590	246821
2	105502	245454
3	172456	246926
4	132052	246491
5	125541	247765

ตารางที่ 4.2 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูลที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 1001001110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

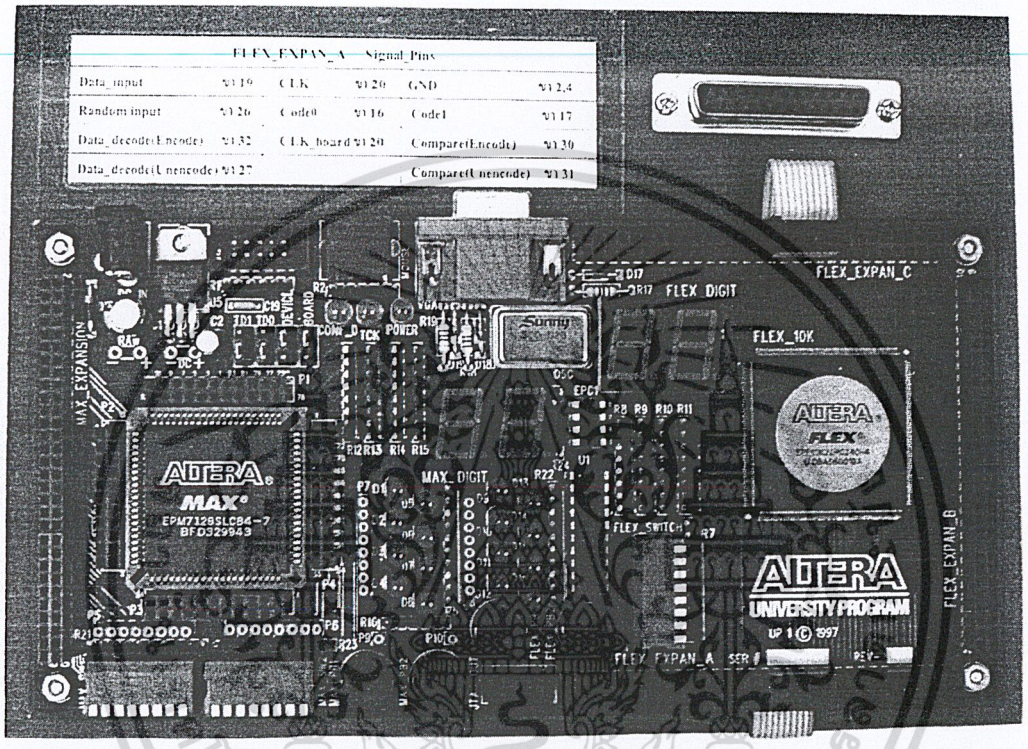
1100110011		
ครั้งที่	จำนวนบิตที่เกิดความผิดพลาด	
	ข้อมูลที่เข้ารหัส	ข้อมูลที่ไม่เข้ารหัส
1	100728	194357
2	92818	193390
3	96184	193150
4	95674	192791
5	95686	192357

ตารางที่ 4.3 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูลที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุต 1100110011

ข้อมูลแบบสุ่มค่า		
ครั้งที่	จำนวนบิตที่เกิดความผิดพลาด	
	ข้อมูลที่เข้ารหัส	ข้อมูลที่ไม่เข้ารหัส
1	159543	245177
2	157451	245046
3	158494	247619
4	162967	246822
5	158443	247608

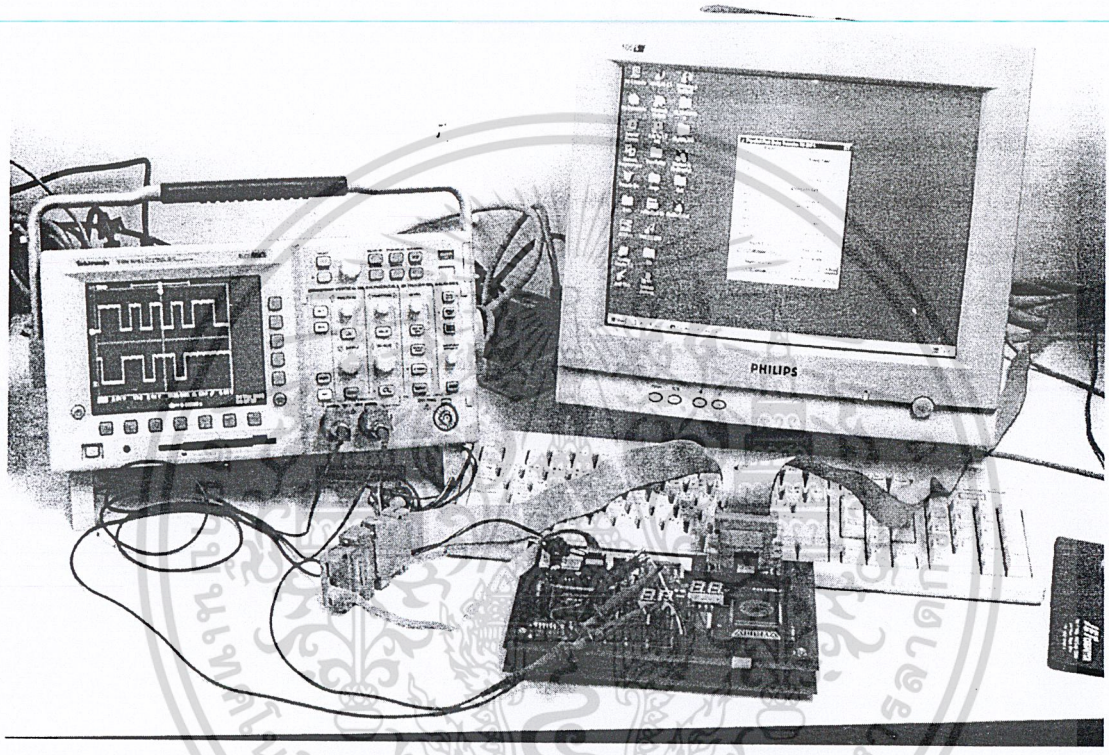
ตารางที่ 4.4 แสดงจำนวนบิตข้อมูลที่เกิดความผิดพลาดเปรียบเทียบกันระหว่างข้อมูลที่เข้ารหัสกับข้อมูลที่ไม่เข้ารหัสเมื่อป้อนอินพุตแบบสุ่มค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.53 แสดงบอร์ดเอฟพีจีเอที่ทำการเชื่อมต่อกับพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.54 แสดงการต่อบอร์ดเอฟพีจีเอในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

จากการออกแบบ ส่วนของการเข้ารหัสแบบคอนโวลูชั่นโค้ด โดยการเขียนโปรแกรมด้วยภาษาวีเอชดีแอลและการออกแบบทางกราฟฟิค แล้วทำการโปรแกรมลงบนชิพอุปกรณ์เอฟพีจีเอ โดยมีการป้อนอินพุต ที่รับมาจากคอมพิวเตอร์ผ่านทางพอร์ตขนานครั้งละ 10 บิต โดยจะทำการป้อนข้อมูลแบบกำหนดเองและแบบสุ่มค่าข้อมูล จะพบได้ว่าส่วนของการเข้ารหัสและถอดรหัสสามารถทำงานได้อย่างถูกต้อง โดยเปรียบเทียบจากผลของการจำลองการทำงาน ผ่านทางโปรแกรม MAX + PLUS II เมื่อเปรียบเทียบแล้วพบว่ามีผลที่ได้ตรงตามการจำลองการทำงาน

ปัญหาที่เกิดขึ้นส่วนใหญ่จะมาจากการวัดค่าสัญญาณเอาต์พุตของบอร์ดเอฟพีจีเอ ซึ่งทำงานที่ความถี่สูง ทำให้เมื่อวัดสัญญาณแล้วไม่สามารถพิจารณาสัญญาณได้จึงต้องใช้ในส่วนของตัวลดความถี่มาเป็นตัวช่วยในการวิเคราะห์สัญญาณที่เกิดขึ้น และข้อผิดพลาดของการถอดรหัสข้อมูลยังมีส่วนของความบกพร่องหากข้อมูลที่ส่งมาทำการถอดรหัสมีลักษณะของความผิดพลาดติดต่อกันมากๆ ทำให้ถอดรหัสได้ไม่ถูกต้องทั้งหมด ซึ่งเป็นข้อจำกัดของการเข้ารหัสแบบคอนโวลูชั่นโค้ด

แนวทางในการพัฒนาในส่วนของการเข้ารหัสและถอดรหัส หากสามารถเข้ารหัสที่อัตราการเข้ารหัสที่ดีกว่านี้ได้จะสามารถทำให้ประสิทธิภาพของการลดค่าความผิดพลาดของข้อมูลยิ่งดีขึ้นแต่ก็จะทำได้ยากในทางปฏิบัติ เนื่องจากด้านของการถอดรหัสจะมีความยุ่งยากมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ประสิทธิ์ ประพิณมงคลการ,“ หลักการระบบสื่อสาร, ” บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน) 2533

[2] บริษัท แอสทรอน ลอจิก รีเสิร์ชแอนด์ดิวิลอปเมนต์ จำกัด, “ เปิดโลก FPGA กับ WIZARD PLD – AO1, ” 2544

[3] ปริญญา เรืองศิริไพศาล, “ การวิเคราะห์สมรรถนะของการเข้ารหัสแบบคอนโวลูชั่นและการไม่เข้ารหัสของระบบไดเร็คซีเควน CDMA บนช่องสัญญาณนาคากามิ, ” วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า 2543

[4] B. Sklar, “ Digital Communications Fundamentals and Applications,” Communication Engineering Services, Tarzana, California and University of California, Los Angeles, 2001

[5] D.L. Perry, “ VHDL, ” New York : McGraw - Hill, 1995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานฉบับนี้สำเร็จลุล่วงได้ดีด้วยได้รับความช่วยเหลือ และชี้แนะจากหลายท่าน ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์อัครพล ตรีรัตน์ ที่ให้คำปรึกษาและความช่วยเหลือด้านข้อมูลและอุปกรณ์ในการทำโครงงาน ขอขอบคุณ คุณศรวัฒน์ ชิวปรีชา ที่ให้คำปรึกษาและให้ความช่วยเหลือในด้านการใช้งานเอฟพีจีเอ และขอขอบคุณ คุณภัทรพงษ์ ผาสุขกิจ ที่ให้ความช่วยเหลือด้านข้อมูลและตำราต่างๆ มาโดยตลอด ผู้เขียนพึงระลึกอยู่เสมอว่ารายงานฉบับนี้จะไม่สามารถสำเร็จลงได้เลย หากขาดความช่วยเหลือจากทุกท่านจึงขอขอบพระคุณมาอย่างสูง

กุลพงษ์ ประสงค์ดี
ธวัชชัย จันทร์ถาวร
ฤกษ์ชัย พิพัฒน์นาวิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้