# ปริญญานิพนธ์

## ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต

## 4 BITs ERROR - CORRECTING LABORATORY SET


นายมานพ ประจญกิจชัย

นายรุ่งโรจน์ ดิษฐภู่

นายศิริชัย เอมช้าง

นายสุธี นิ่มเทียน



A021291

เลขหมู่ ... 1522 ก21291

เลขทะเบียน ...

วัน เดือน ปี ... 29 ตค 2539


ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**คณะครุศาสตร์อุตสาหกรรม**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

.....

**หัวข้อปริญญานิพนธ์** ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต

4 BITs ERROR - CORRECTING LABORATORY SET

**ชื่อนักศึกษา**

1. นายมานพ ประจญกิจชัย รหัสประจำตัว 37031215
2. นายรุ่งโรจน์ ดิษฐภู่ รหัสประจำตัว 37031217
3. นายศิริชัย เอมช้าง รหัสประจำตัว 37031221
4. นายสุธี นิ่มเทียน รหัสประจำตัว 37031226

**หลักสูตร** ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** วิศวกรรมโทรคมนาคม

**อาจารย์ผู้ควบคุมปริญญานิพนธ์**

1. อาจารย์กิติพงศ์ มะโน
2. ดร.สุรสิทธิ์ ราตรี
3. อาจารย์วรวิทย์ สมหา

| คณะกรรมการสอบปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| 1. อาจารย์กิติพงศ์ มะโน | |
| 2. อาจารย์วิสุทธิ์ อธิพรธรรม | |
| 3. อาจารย์วรวิทย์ สมหา | |
| 4. อาจารย์โกศล ตราชู | |
| 5. อาจารย์อำพล ทองระอา | |

**วัน/เดือน/ปี ที่สอบ**

**สถานที่สอบ** ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม

ภาควิชารับรองแล้ว

ลงชื่อ.....

(ผศ.ศรีธีระพล เทพหัสดิน ณ อยุธยา)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่.....เดือน.....พ.ศ.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ปริญญานิพนธ์**

**เรื่อง** ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต

4 BITs ERROR - CORRECTING LABORATORY SET


**ผู้จัดทำ**

1. นายมานพ ประจญกิจชัย
2. นายรุ่งโรจน์ ดิษฐภู่
3. นายศิริชัย เอมช้าง
4. นายสุธี นิ่มเทียน


**อาจารย์ที่ปรึกษา**

ลงนาม .....

( อาจารย์ กิติพงค์ มะโน )

ลงนาม .....

( ดร. สุรสิทธิ์ ราตรี )

ลงนาม .....

( อาจารย์ วรวิทย์ สมหา )

**หัวหน้าภาควิชาครุศาสตร์อุตสาหกรรม**

ลงนาม .....

( ผศ.ดร. ธีระพล เทพหัสดิน ณ อยุธยา )


เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปริญญานิพนธ์

เรื่อง ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต

4 BITs ERROR - CORRECTING LABORATORY SET

## จุดประสงค์

1. เพื่อศึกษาและวิเคราะห์สมการ สำหรับการเข้าและถอดรหัสช่องสัญญาณ
2. เพื่อกำหนดสมการทางคณิตศาสตร์ สำหรับการเข้ารหัสและถอดรหัสช่องสัญญาณ
3. เพื่อออกแบบวงจรเข้าและถอดรหัสช่องสัญญาณ จากสมการทางคณิตศาสตร์ได้
4. เพื่อตรวจสอบการทำงานของวงจรที่ได้จากการออกแบบได้อย่างถูกต้อง
5. เพื่อสร้างเป็นชุดทดลองเข้าและถอดรหัสช่องสัญญาณ จากข้อมูลขนาด 4 บิตได้
6. เพื่อนำชุดทดลองที่ได้ ไปใช้เป็นชุดทดลองปฏิบัติการเกี่ยวกับการสื่อสารข้อมูล

## ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถวิเคราะห์และจัดสร้างสมการทางคณิตศาสตร์สำหรับ การเข้ารหัสและถอดรหัสช่องสัญญาณได้
2. สามารถออกแบบวงจรเข้าและถอดรหัสจากสมการทางคณิตศาตร์ได้
3. สามารถจัดสร้างพร้อมทั้งทดสอบวงจรที่ได้ออกแบบไว้ได้
4. สามารถจัดทำเป็นชุดทดลองปฏิบัติการเกี่ยวกับการสื่อสารข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต

นายมานพ ประจญกิจชัย
นายรุ่งโรจน์ ดิษฐภู่
นายศิริชัย เอมช้าง
นายสุธี นิ่มเทียน

อาจารย์ที่ปรึกษา
อาจารย์กิติพงค์ มะโน
อาจารย์สุรสิทธิ์ ราตรี
อาจารย์วรวิทย์ สมหา

ปีการศึกษา 2538

## บทคัดย่อ

ปริญญานิพนธ์นี้เสนอ ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิต ด้วยการออกแบบวงจรเข้ารหัสและวงจรถอดรหัสโดยใช้วิธีทางรหัสระบบเชิงเส้นที่สามารถตรวจจับและแก้ไขความผิดพลาดของข้อมูลได้ และสร้างเป็นชุดทดลองปฏิบัติการเพื่อแสดงถึงกระบวนการเข้ารหัสและถอดรหัสข้อมูล พร้อมทั้งจัดทำใบงานประกอบการทดลองซึ่งผลจากการใช้ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิตนี้ พบว่าสามารถทำการเข้ารหัสและถอดรหัสข้อมูลได้อย่างมีประสิทธิภาพและมีความเชื่อถือได้ตามขอบเขตความสามารถของวิธีทางรหัสระบบเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 4 BITs ERROR - CORRECTING LABORATORY SET

MR. MANOP PRAJONKITCHAI
MR. RUNGROTE DITPHOO
MR. SIRICHAI EMCHANG
MR. SUTEE NIMTIEN

**ADVISOR**

MR. KITIPONG MANO
DR. SURASIT RATREE
MR. VORAWIT SOMHA

**YAER 1995**

**ABSTRACT**

This thesis presents the 4 BITs ERROR-CORRECTING LABORATORY SET. We design the encoding and decoding circuits by using the systematic linear block code algorithms that it can check and correct the data . And then, we create the laboratory set that it presents process of encoding and decoding. In addition, it has the sheet lab. According to the result of lab, it can encode and decode the data efficiently and reliably in the limit of its algorithms ability.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

การที่ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีนั้น คณะผู้จัดทำจึงเรียนมาเพื่อขอขอบพระคุณ ท่านอาจารย์ที่ปรึกษาปริญญานิพนธ์รวมถึง คณาจารย์ในภาควิชาครุศาสตร์วิศวกรรมทุกท่านเป็นอย่างยิ่งที่ได้ให้ทั้งข้อเสนอแนะและแนวทางการแก้ไขปัญหาต่างๆ รวมทั้งได้ให้คำปรึกษา ตลอดจนให้การสนับสนุนด้านเครื่องมือและอุปกรณ์ ที่ใช้ในการทดลองโครงงานในครั้งนี้ นอกจากนี้ขอขอบพระคุณ คุณ ชาญฤทธิ์ ดวงสุวรรณ ที่ให้ความช่วยเหลือทางด้านโปรแกรมรับ-ส่งข้อมูล ของไอซีไมโครคอนโทรลเลอร์ เบอร์ 87C51 ตลอดจนเพื่อนๆ ทุกคนที่ได้ให้ทั้งคำแนะนำและความช่วยเหลือรวมถึงการให้กำลังใจในการทำปริญญานิพนธ์ในครั้งนี้ ท้ายที่สุดขอขอบพระคุณเพื่อนร่วมงานทุกคนที่ได้ให้ความร่วมมือ ร่วมใจ ในการทำงานจนกระทั่ง ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| เรื่อง | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญภาพ | VII |
| สารบัญตาราง | XI |
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | 4 |
| 2.1 ทฤษฎีรหัส | 4 |
| 2.1.1 ลักษณะของการเข้ารหัส ( Coding Classification ) | 4 |
| 2.1.2 ชนิดของรหัส ( Types of Codes ) | 4 |
| 2.2 ทฤษฎีชุดรหัสเชิงเส้น ( Linear Block Code Theory ) | 5 |
| 2.2.1 นิยาม ( Definition ) | 5 |
| 2.2.2 หลักการเข้ารหัสและถอดรหัส ( Principle of Encoding and Decoding ) | 7 |
| 2.2.3 ขอบเขตของรหัสเชิงเส้น ( Bounds of Linear Block Code ) | 12 |
| 2.3 หลักการสื่อสารข้อมูล ( Principle of Data Communication ) | 14 |
| 2.3.1 โหมดการสื่อสาร ( Communication Modes ) | 15 |
| 2.3.2 เทคนิคของการส่งผ่านข้อมูล ( Data Transmission Techniques ) | 16 |
| 2.3.3 การส่งผ่านข้อมูลแบบอนุกรม ( Series Types Data Transmission ) | 17 |
| 2.3.4 การแปลงข้อมูลการติดต่อสื่อสาร ( Data Communication Transfer ) | 20 |
| 2.3.5 ลักษณะของช่องสัญญาณการสื่อสาร ( Characteristic of Communication Channel ) | 21 |
| 2.4 อุปกรณ์ควบคุมสำหรับการสื่อสาร ( Serial Communication Device ) | 21 |
| 2.4.1 โครงสร้างของ 87C51 ( Structure of 87C51 ) | 21 |
| 2.4.2 สถาปัตยกรรมของ 87C51 ( Architecture of 87C51 ) | 23 |
| 2.5 อุปกรณ์ชนิดโปรแกรมได้ ( Program Array Logic ; PAL ) | 36 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2..5.1 ลักษณะโครงสร้าง ( PLD Hardware ) 36
2.5.2 ลักษณะคำสั่งการใช้งาน ( PLD Software ) 40
2.6 การสอนเชิงปฏิบัติ 45
2.6.1 บทนำ 45
2.6.2 จุดมุ่งหมายของการสอนเชิงปฏิบัติ 45
2.6.3 การดำเนินการสอนเชิงปฏิบัติ 46
บทที่ 3 การออกแบบและการสร้าง 50
3.1 การออกแบบชุดทดลองการปฏิบัติการ ( Laboratory Design ) 50
3.1.1 ส่วนตัวกระข้อมูล ( Operator Section ) 51
3.1.2 ส่วนควบคุมและแสดงผล ( Control and Display Section ) 51
3.2 การออกแบบวิธีการสร้างรหัสระบบเชิงเส้น 51
( Systematic Linear Block Codes Algorithms Design )
3.2.1 การสร้างชุดเข้ารหัส ( Encoder Implementation ) 51
3.2.2 การสร้างชุดถอดรหัส ( Decoder Implementation ) 56
3.3 การออกแบบสร้างระบบวงจรชุดรหัสขนาด (8,4) บิต 67
( Circuit Design and Build for (8,4) Systematic Linear Block Codes )
3.3.1 การออกแบบชุดวงจรเข้ารหัสระบบขนาด (8,4) บิต 68
((8,4) Systematic Encoder Circuit Design )
3.3.2 การออกแบบชุดวงจรถอดรหัสระบบขนาด (8,4) บิต 69
((8,4) Systematic Decoder Circuit Design )
3.4 การปรับปรุงและพัฒนาระบบเชิงเส้น 71
3.4.1 การปรับปรุงวงจรชุดเข้ารหัสเชิงเส้นขนาด (8,4) บิต 73
3.4.2 การพัฒนารหัสระบบเชิงเส้นด้วยการเพิ่มบิตตรวจสอบ 74
3.5 การออกแบบชุดทดลองปฏิบัติการ 103
3.6 การออกแบบใบงานการทดลอง 107
บทที่ 4 การทดลองและผลการทดลอง 109
4.1 การทดลองวงจรส่วนประกอบต่างๆ ของชุดทดลอง 109

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

4.1.1 การทดลองวงจรชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 109
4.1.2 การทดลองชุดวงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต 115
4.1.3 การทดลองวงจรเข้ารหัสและถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต 134
4.2 การทดลองส่งข้อมูลผ่านชุดรับ-ส่งข้อมูลแบบอนุกรม 202
4.2.1 การทดลองระบบรับ-ส่งข้อมูลกับชุดวงจรเข้ารหัส
และถอดรหัสภายนอก 203
4.2.2 การทดลองระบบรับ-ส่งข้อมูลกับชุดวงจรเข้ารหัสและถอดรหัสภายใน 204
บทที่ 5 บทสรุป วิจารณ์ และแนวทางในการพัฒนา 206
5.1 บทสรุป 206
5.2 ปัญหาและการแก้ไข 207
5.2.1 ปัญหา 207
5.2.2 การแก้ไข 207
5.3 แนวทางในการพัฒนา 208
บรรณานุกรม
ภาคผนวก ก.
ภาคผนวก ข.
ภาคผนวก ค.
ภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ


| | หน้า |
|---|---|
| รูปที่ 1.1 ระบบการสื่อสารที่เสนอโดยแชนนอน | 1 |
| รูปที่ 1.2 ระบบการสื่อสารที่เสนอโดยฟาโน | 2 |
| รูปที่ 2.1 การเพิ่มรหัสแก้ไขจากการเข้ารหัสแบบรหัสเชิงเส้น | 8 |
| รูปที่ 2.2 ส่วนประกอบของการสื่อสารข้อมูล | 14 |
| รูปที่ 2.3 รูปแบบของการส่งข้อมูล | 16 |
| รูปที่ 2.4 ตัวอย่างของการส่งสัญญาณแบบอนุกรมและการส่งสัญญาณแบบขนาน | 17 |
| รูปที่ 2.5 รูปแบบชุดข้อมูลสำหรับการส่งสัญญาณแบบไม่เข้าจังหวะ | 18 |
| รูปที่ 2.6 ตัวอย่างรูปแบบการส่งสัญญาณแบบไม่เข้าจังหวะ | 19 |
| รูปที่ 2.7 การแปลงรูปแบบข้อมูลจากแบบอนุกรมไปเป็นแบบขนาน | 20 |
| รูปที่ 2.8 การแปลงรูปแบบข้อมูลจากแบบขนานไปเป็นแบบอนุกรม | 20 |
| รูปที่ 2.9 โครงสร้างภายในของ MCS-51 | 22 |
| รูปที่ 2.10 สถาปัตยกรรมภายในของ 87C51 | 23 |
| รูปที่ 2.11 แผนผังขาของ 87C51 แบบ DIP | 24 |
| รูปที่ 2.12 โครงสร้างของพอร์ต 0 | 25 |
| รูปที่ 2.13 โครงสร้างของพอร์ต 1 | 26 |
| รูปที่ 2.14 โครงสร้างของพอร์ต 2 | 27 |
| รูปที่ 2.15 โครงสร้างของพอร์ต 3 | 28 |
| รูปที่ 2.16 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซต 87C51 | 30 |
| รูปที่ 2.17 วงจรออสซิลเลเตอร์ภายใน 87C51 | 32 |
| รูปที่ 2.18 87C51 ที่ทำงานโดยสัญญาณที่มาจากภายนอก | 32 |
| รูปที่ 2.19 ลำดับสถานะการทำงานใน MCS-51 | 35 |
| รูปที่ 2.20 การอธิบายพื้นฐานภายในโครงสร้าง PAL | 37 |
| รูปที่ 2.21 ตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL | 38 |
| รูปที่ 2.22 โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต | 39 |
| รูปที่ 2.23 การเปลี่ยนแปลงลอจิกโดยใช้ PAL | 39 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 2.25 ตัวอย่างวงจรลอจิกพื้นฐาน Output = $I_1\overline{I}_2 + \overline{I}_1 I_2$ 40
รูปที่ 2.26 ผังการสร้าง JEDEC FILE 43
รูปที่ 2.27 ผังการโปรแกรมอุปกรณ์ PLD 44
รูปที่ 2.28 ตัวอย่างใบงาน 48
รูปที่ 3.1 ระบบชุดทดลองปฏิบัติการ การเข้ารหัสและถอดรหัสช่องสัญญาณ 50
รูปที่ 3.2 แบบจำลองชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 56
รูปที่ 3.3 แบบจำลองชุดสร้างเวกเตอร์ซีนโดรม 60
รูปที่ 3.4 แบบจำลองการทำงานของชุดตรวจสอบแก้ไขรหัสที่ผิดไป 1 บิต 62
รูปที่ 3.5 ชุดวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 68
รูปที่ 3.6 วงจรสร้างเวกเตอร์ซีนโดรม 69
รูปที่ 3.7 วงจรตรวจจับความผิดพลาด 70
รูปที่ 3.8 วงจรแก้ไขความผิดพลาด 72
รูปที่ 3.9 วงจรชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 73
รูปที่ 3.10 วงจรชุดถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต 74
รูปที่ 3.11 วงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 76
รูปที่ 3.12 วงจรกำเนิดเวกเตอร์ซีนโดรมของชุดถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต 80
รูปที่ 3.13 วงจรแก้ไขความผิดพลาดของเวกเตอร์คำรหัสของชุดถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต 94
รูปที่ 3.14 ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิต 103
รูปที่ 3.15 ผังกระบวนการชุดรับ-ส่งข้อมูลการเข้ารหัสช่องสัญญาณ 106
รูปที่ 4.1 ผังขั้นตอนการใช้งานของชุดโปรแกรมสำเร็จรูปออร์แคด 110
รูปที่ 4.2 วงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 111
รูปที่ 4.3 ผลจากการจำลองการทำงานด้วยข้อมูล 1011 112
รูปที่ 4.4 ภาพวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 114
รูปที่ 4.5 ภาพการทดสอบวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต ด้วยข้อมูล 1011 115
รูปที่ 4.6 วงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต 116
รูปที่ 4.7 ผลของเวกเตอร์ซีนโดรมที่ได้จากคำรหัส 10110010 117

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.8 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 10110010 118
รูปที่ 4.9 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 10110010 119
รูปที่ 4.10 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 10100010 120
รูปที่ 4.11 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 10100010 121
รูปที่ 4.12 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 10100010 122
รูปที่ 4.13 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 10101010 124
รูปที่ 4.14 วงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต 129
รูปที่ 4.15 ภาพวงจรถอดรหัสที่ประกอบบนแผงประกอบวงจร 130
รูปที่ 4.16 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10110010 131
รูปที่ 4.17 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10100010 132
รูปที่ 4.18 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10101010 133
รูปที่ 4.19 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10011110 134
รูปที่ 4.20 วงจรเข้ารหัสขนาด (12,4) บิต 135
รูปที่ 4.21 ผลการจำลองการทำงานด้วยข้อมูล 1011 136
รูปที่ 4.22 วงจรถอดรหัสขนาด (12,4) บิต 138
รูปที่ 4.23 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 101111010111 148
รูปที่ 4.24 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 101111010111 149
รูปที่ 4.25 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 101111010111 150
รูปที่ 4.26 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 101011010111 151
รูปที่ 4.27 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 101011010111 152
รูปที่ 4.28 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 101011010111 153
รูปที่ 4.29 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 011111010111 155
รูปที่ 4.30 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 011111010111 156
รูปที่ 4.31 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 011111010111 157
รูปที่ 4.32 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 011101010111 162
รูปที่ 4.33 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 011101010111 163
รูปที่ 4.34 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 011101010111 164

เอกสารนี้มีการสงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.35 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 110010100011 175
รูปที่ 4.36 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 110010100011 176
รูปที่ 4.37 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 110010100011 177
รูปที่ 4.38 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 100011000000 199
รูปที่ 4.39 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 100011000000 200
รูปที่ 4.40 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 100011000000 201
รูปที่ 4.41 ภาพการทดลองระบบรับ-ส่งข้อมูล ที่ต่อร่วมกับวงจรชุดเข้ารหัสและถอดรหัสภายนอก 204
รูปที่ 4.42 ภาพการทดลองระบบรับ-ส่งข้อมูล ที่ต่อร่วมกับวงจรชุดเข้ารหัสและถอดรหัสภายใน 205

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


หน้า

ตารางที่ 2.1 การบวกและการคูณของรหัสสองสัญลักษณ์ 5

ตารางที่ 3.1 รูปแบบเวกเตอร์คำรหัสของชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต 54

ตารางที่ 3.2 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 1 บิต 61

ตารางที่ 3.3 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 2 บิต 63

ตารางที่ 3.4 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 3 บิต 64

ตารางที่ 3.5 รูปแบบเวกเตอร์คำรหัสของชุดเข้ารหัสระบบเชิงเส้นขนาด (12,4) บิต 77

ตารางที่ 3.6 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 1 บิต 81

ตารางที่ 3.7 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 2 บิต 81

ตารางที่ 3.8 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 3 บิต 84

ตารางที่ 3.9 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 4 บิต 89

ตารางที่ 3.10 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 5 บิต 92

ตารางที่ 4.1 ผลของเวกเตอร์คำรหัสที่ได้จากการจำลองการทำงาน ของข้อมูลทั้ง 16 รูปแบบ 113

ตารางที่ 4.2 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 1 บิต 123

ตารางที่ 4.3 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 2 บิต 125

ตารางที่ 4.4 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 3 บิต 126

ตารางที่ 4.5 ผลของเวกเตอร์คำรหัสที่ได้จากการจำลองการทำงาน ของข้อมูลทั้ง 16 รูปแบบ 137

ตารางที่ 4.6 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 1 บิต 154

ตารางที่ 4.7 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 2 บิต 158

ตารางที่ 4.8 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 3 บิต 165

ตารางที่ 4.9 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 4 บิต 178

ตารางที่ 4.10 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 5 บิต 202

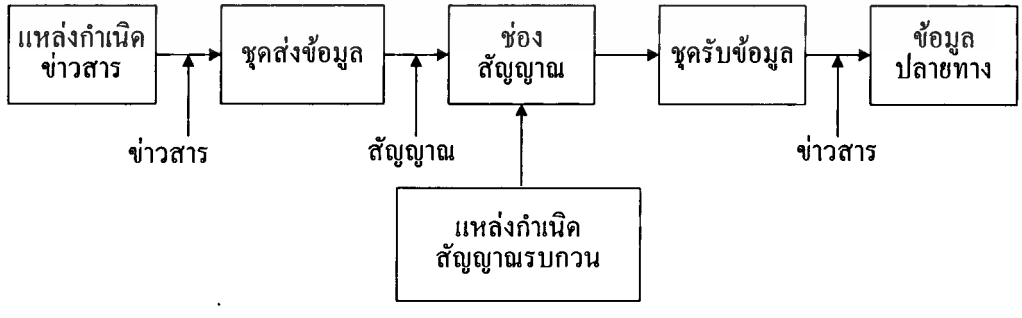

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่1

# บทนำ

นับตั้งแต่สมัยโบราณการสื่อความหมายของมนุษย์กระทำโดย ผู้ส่งทำการส่งข่าวสารที่ต้องการผ่านตัวกลางไปยังผู้รับแล้วผู้รับจะแปลความและตีความ ฉะนั้นการสื่อสารจึงนิยามได้ว่า เป็นการส่งผ่านข่าวสารจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง นั่นก็หมายความว่า ตัวข่าวสารจะต้องอาศัยสื่อกลางในการเดินทาง ที่เรียกว่า ช่องสัญญาณ ( Channel ) และในช่องสัญญาณจะมีสัญญาณรบกวน ( Noise Signal ) อยู่ 2 ประเภท คือ สัญญาณรบกวนตามธรรมชาติและที่มนุษย์สร้างขึ้น ( Natural and Man-Made Noise ) โดยสัญญาณรบกวนนี้จะทำให้ข่าวสารข้อมูลมีการผิดเพี้ยนไป เรียกว่า ความผิดพลาด ( Error ) อันทำให้การติดต่อสื่อสารไม่บรรลุวัตถุประสงค์ที่ต้องการ ด้วยเหตุนี้จึงได้มีการคิดวิธีการที่จะควบคุมความผิดพลาดของข่าวสารในระหว่างการส่งผ่านช่องสัญญาณ ซึ่งถือได้ว่าเป็นแนวคิดพื้นฐานที่จะทำให้เกิดวิธีการเข้ารหัส ( Encoding or Coding ) และวิธีการถอดรหัส ( Decoding ) ข้อมูลข่าวสารขึ้น

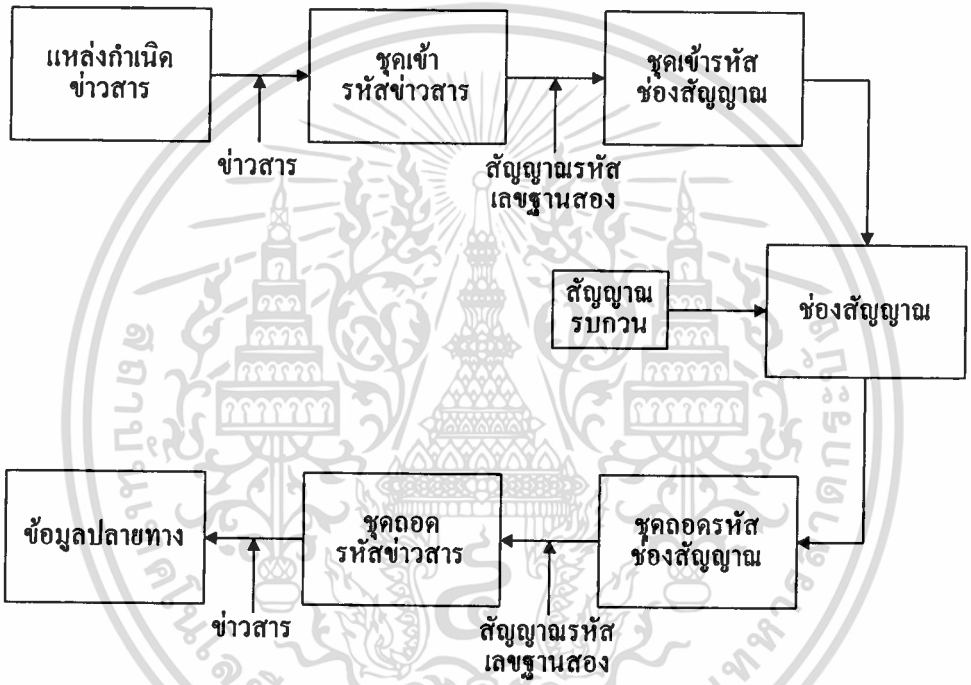
หลังสงครามโลกครั้งที่ 2 เทคโนโลยีทางการสื่อสารได้พัฒนาให้ไปอย่างรวดเร็วและทฤษฎีข่าวสาร ( Information Theory ) ก็ถือได้ว่าเป็นวิทยาการแขนงหนึ่งของการสื่อสาร ซึ่งได้พัฒนาขึ้น ในช่วงปี ค.ศ. 1940-1950 ด้วยกลุ่มนักวิจัยที่นำโดย คลอดแชนนอน ( Claud Shannon ) โดยมีสาระสำคัญ คือ การจัดการกับสัญญาณข่าวสารด้วยการเข้ารหัส และในปี ค.ศ. 1948 แชนนอน เป็นคนแรกที่เสนอรูปแบบ ( Model ) ของระบบสื่อสารได้ตรงกับนิยามการสื่อสารดังรูปที่ 1.1




**รูปที่ 1.1 ระบบการสื่อสารที่เสนอโดยแชนนอน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาในปี ค.ศ.1961 ฟาโน ( Fano ) ได้เสนอรูปแบบของระบบสื่อสาร โดยดัดแปลงมาจากรูปแบบของแชนนอนดังรูปที่1.2 ซึ่งเรียกว่า รูปแบบของระบบสื่อสารข้อมูลจากรูปแบบของการสื่อสารดังกล่าว จึงทำให้เกิดวิวัฒนาการของ ทฤษฎีรหัส ( Coding Theory ) ขึ้นเมื่อบทความเกี่ยวกับทฤษฎีการแก้ไขรหัสที่ผิดพลาดของแชนนอนได้ถูกตีพิมพ์และเผยแพร่ในปี ค.ศ.1948 และจากงานวิจัยของแชนนอนได้สร้างความสนใจแก่นักวิจัยในการที่จะแก้ไขปัญหาการนำชุดเข้ารหัสและชุดถอดรหัสไปควบคุมสัญญาณรบกวนในระบบสื่อสาร




**รูปที่ 1.2 ระบบการสื่อสารที่เสนอโดยฟาโน**

ดังนั้นปริญญานิพนธ์ฉบับนี้ จึงกล่าวถึง การออกแบบชุดวงจรเข้ารหัสและวงจรถอดรหัสรวมถึงการสร้างชุดทดลองปฏิบัติการ เพื่อให้ผู้ที่ศึกษาได้เข้าใจถึงวิธีการและผลที่ได้อย่างชัดเจน ซึ่งรายละเอียดต่างๆ มีดังนี้

บทที่ 1 บทนำ กล่าวถึง ความหมายความสำคัญและวิวัฒนาการของทฤษฎีรหัสรวมถึงสาระสำคัญของปริญญานิพนธ์ฉบับนี้

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึง รายละเอียดของทฤษฎีและหลักการที่เกี่ยวข้องกับปริญญานิพนธ์ฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบและการสร้างวงจร กล่าวถึง การออกแบบระบบ วิธีการออกแบบวงจรจากพีชคณิตของทฤษฎีรหัส และลักษณะของวงจรตลอดจนการสร้างจนกระทั่งถึงการปรับปรุงวงจร

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึง วิธีการทดลองในแต่ละส่วน ผลการทดลองที่ได้รวมถึงการทดสอบขีดความสามารถของระบบ และการสรุปผลที่ได้

บทที่ 5 สรุป อภิปราย และ ข้อเสนอแนะ กล่าวถึง ปัญหาอุปสรรค ผลที่ได้เปรียบเทียบกับวัตถุประสงค์และแนวทางในการแก้ไข เพื่อพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ทฤษฎีและหลักการ

## 2.1 ทฤษฎีรหัส ( Coding Theory )

### 2.1.1 ลักษณะของการเข้ารหัส ( Coding Classification )

จากรูปแบบการสื่อสารข้อมูลของฟาโนที่เสนอไว้ในปี ค.ศ.1961 ดังรูปที่ 1.3 จะเห็นว่าการเข้ารหัสจะมีอยู่ 2 ลักษณะ คือการเข้ารหัสแหล่งกำเนิด ( Source Encoding ) และการเข้ารหัสช่องสัญญาณ ( Channel Encoding ) ซึ่งจะมีความแตกต่างกันตรงที่การเข้ารหัสแหล่งกำเนิดหรือการเข้ารหัสข่าวสารจะมีหน้าที่ในการแปลงข่าวสารที่รับเข้ามาในแต่ละรูปแบบให้อยู่ในรูปของสัญญาณรหัสเลขฐานสอง ( Binary Signal ) อันประกอบด้วย 0 กับ 1 สำหรับข้อดีของการเข้ารหัสแบบนี้อยู่ที่ทำให้การส่งข่าวสารเป็นไปอย่างรวดเร็วและมีประสิทธิภาพสูง นอกจากนี้ ถ้าสามารถเลือกวิธีของการเข้ารหัสได้อย่างเหมาะสมแล้ว จะทำให้ได้ความยาวโดยเฉลี่ยของสัญญาณรหัสเลขฐานสองที่สั้นที่สุด นั่นคือ การอัดข่าวสารได้เป็นจำนวนมาก แต่ข้อเสีย ก็คือ สัญญาณรหัสเลขฐานสองที่ได้ยังมีการจัดวางที่ยังไม่เป็นระบบระเบียบพอที่จะนำมาใช้ในการส่งเพื่อติดต่อสื่อสารได้ เพราะจะเกิดการผิดพลาดของข้อมูลเนื่องจากสัญญาณรบกวนในระหว่างการส่งได้ ฉะนั้นจึงต้องมีการเข้ารหัสช่องสัญญาณ ทั้งนี้ก็เพื่อที่จะจัดการกับสัญญาณรหัสเลขฐานสองให้มีความซับซ้อนอย่างเป็นระบบมากขึ้นและเหมาะสมสำหรับการส่งผ่านข้อมูลในช่องสัญญาณที่มีอยู่ เพื่อให้สัญญาณที่ได้จากเครื่องถอดรหัสช่องสัญญาณ ( Channel Decoder ) มีการผิดพลาดน้อยที่สุด

### 2.1.2 ชนิดของรหัส ( Type of Codes )

รหัสที่ใช้ควบคุมความผิดพลาดนั้นแบ่งออกเป็น 2 ชนิด คือรหัสกลุ่ม ( Block Code ) และรหัสลำดับหรือรหัสการประสาน ( Sequence or Convolutional Codes ) โดยระบบของรหัสกลุ่มนั้น หนึ่งกลุ่มจะมีอยู่ n บิต ซึ่งจะขึ้นอยู่กับกลุ่มของข้อมูล k บิต ในหนึ่งหน่วยเวลาใดๆ และกลุ่มของรหัสแก้ไขจะมี (n-k) บิต ในส่วนของรหัสแก้ไขนั้นจะถูกนำมาใช้ในการตรวจสอบและแก้ไขข่าวสารข้อมูลในกลุ่มนั้นๆ แต่ระบบของรหัสลำดับจะเรียกชุดข่าวสารว่าลำดับข่าวสาร ( Information Sequence ) ในลำดับรหัส ( Code Sequence ) หนึ่งๆ จะมีอยู่หลายชุดรหัสที่เรียงกันไปเป็นลำดับโดยหนึ่งชุดรหัสจะมีอยู่ n บิต ซึ่งจะได้จากลำดับข่าวสารครั้งละหนึ่งบิตจากจำนวน k บิตและ N-1 ลำดับข่าวสาร รหัสที่ใช้ในการแก้ไขความผิดพลาดจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทรกเข้าระหว่างลำดับของข่าวสาร ซึ่งนอกจากจะใช้ตรวจลำดับข่าวสารที่อยู่ข้างหน้าแล้วยังใช้ตรวจสอบลำดับข่าวสารในลำดับรหัสอื่นๆ ด้วย

ฉะนั้นถ้าพิจารณาถึงการรับสัญญาณเลขฐานสองเข้ามาเพื่อเข้ารหัสและถอดรหัสช่องสัญญาณแล้วจะแบ่งออกได้ 2 แบบ คือ การเข้ารหัสแบบกลุ่ม ( Block Encoding ) เป็นลักษณะของการนำข้อมูลเข้ามาจนครบ 1 ชุดก่อน แล้วจึงเข้ารหัสพร้อมกัน จากนั้นจึงส่งออกไป และการเข้ารหัสแบบลำดับ หรือการเข้ารหัสการประสาน ( Sequence or Convolutional Encoding ) เป็นลักษณะของการนำข้อมูลเข้ามาพร้อมกับทำการเข้ารหัสทีละบิตโดยจะรับข้อมูลมากี่ชุดก็ตาม จากการเปรียบเทียบการเข้ารหัสทั้ง 2 แบบนั้น จะเห็นได้ว่าการเข้ารหัสแบบลำดับนั้นจะมีความยุ่งยากมากกว่าการเข้ารหัสแบบกลุ่ม ซึ่งเท่าที่ผ่านมาความสามารถในการแก้ไขความผิดพลาดของรหัสข้อมูลโดยการเข้ารหัสแบบลำดับมีประสิทธิภาพดีกว่าแบบกลุ่ม แต่ในปัจจุบันแนวโน้มของการพัฒนาวิธีการเข้ารหัสและถอดรหัสให้มีความสามารถตรวจสอบและแก้ไขความผิดพลาดของรหัสอย่างมีประสิทธิภาพนั้น รหัสกลุ่มจะได้รับความสนใจและมีการพัฒนามากขึ้นเป็นลำดับ

## 2.2 ทฤษฎีชุดรหัสเชิงเส้น ( Linear Block Code Theory )

### 2.2.1 นิยาม ( Definition )

**1. กาลัวส์ฟิลด์ ( Galois Field Arithmetic )**

โดยปกติแล้ว การบวกและการคูณของจำนวนเลขที่นับได้ ( Finite Number ) นั้นจะสามารถกระทำได้ตามกฎเกณฑ์ธรรมดาของคณิตศาสตร์แต่ในกรณีของการบวกและการคูณของรหัสตัวเลขสำหรับการติดต่อสื่อสารซึ่งมีสัญลักษณ์เป็น 0 กับ 1 จะทำได้ตามกฎเกณฑ์ต่อไปนี้

| การบวก | การคูณ |
|---|---|
| 0 + 0 = 0 | 0 . 0 = 0 |
| 0 + 1 = 1 | 0 . 1 = 0 |
| 1 + 0 = 1 | 1 . 0 = 0 |
| 1 + 1 = 0 | 1 . 1 = 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1 การบวกและการคูณของรหัสสองสัญลักษณ์**

การบวกและการคูณดังกล่าว เรียกว่า การบวกและการคูณแบบโมดูโล-2 (Modulo-2) โดยถือว่า 2 เท่ากับ 0 ดังนั้น 1+1 = 0 และ 1 = -1 ฉะนั้นสัญลักษณ์ 0 กับ 1 พร้อมด้วยการบวกและการคูณแบบโมดูโล-2 นี้รวมกันเข้าเป็นฟิลด์ ( Field ) ซึ่งเรียกว่า ไบนารีฟิลด์ ( Binary Field ) เขียนได้เป็น GF(2)

**2. เวกเตอร์สเปซ ( Vector Space )**

เมื่อพิจารณาถึงลำดับของสัญลักษณ์เลขฐานสองที่ประกอบด้วย 0 กับ 1 จะได้

$$\underline{v} = (v_1, v_2, v_3, ..., v_n) \tag{2.1}$$

เมื่อ $v_i$ เป็นรหัสบิตที่ i มีค่าเป็น 0 หรือ 1 จำนวน n บิต ดังนั้นลำดับของรหัสเวกเตอร์ $\underline{v}$ สามารถเขียนได้ถึง $2^n$ ลำดับของรหัสที่แตกต่างกัน ในการบวกของลำดับของเวกเตอร์สองชุด โดยที่แต่ละชุดจะมี n ตัว หรือเรียกว่า n-ทูเปิ้ลส์ ( n-tuples ) ทำได้ดังนี้

$$\begin{aligned} \underline{v} &= (v_1, v_2, v_3, ..., v_n) \\ \underline{u} &= (u_1, u_2, u_3, ..., u_n) \\ \underline{v} + \underline{u} &= (v_1+u_1, v_2+u_2, v_3+u_3, ..., v_n+u_n) \end{aligned} \tag{2.2}$$

โดยการบวกระหว่าง $v_i$ และ $u_i$ เป็นไปตามกฎโมดูโล-2 ดังนั้นลำดับสัญญาณ $\underline{v} + \underline{u}$ ก็ยังคงเป็นลำดับของรหัสที่มีเลขฐานสองอยู่ n ตัว และการคูณเลขฐานสองของ n-ทูเปิ้ลส์ด้วยสกาลาร์ ( Scalar ) ใน GF (2) คือ

$$\sigma (v_1, v_2, ..., v_n) = (\sigma v_1, \sigma v_2, ..., \sigma v_n) \tag{2.3}$$

ฉะนั้นเซตของเลขฐานสองจำนวน n-ทูเปิ้ลส์นี้ เรียกว่า เวกเตอร์สเปซของ GF (2) ซึ่งแทนด้วย$\underline{v_n}$ ถ้าให้ s เป็นเซตย่อย ( Subset ) ของ $\underline{v_n}$ ซึ่งจะเรียกว่า ซับสเปซ ( Subspace ) ต่อเมื่อ

- มีเวกเตอร์ศูนย์ ( Zero Vector ) อยู่ในเซตย่อย
- ผลรวมของสองเวกเตอร์ในเซตย่อย ยังคงอยู่ในเซตย่อย

ดังนั้นถ้าหาก $k < n$ และ $v_1, v_2, ..., v_k$ เป็น k เวกเตอร์อิสระเชิงเส้นของ $\underline{v_n}$ และการรวมกันอย่างเชิงเส้นของเวกเตอร์ $v_1, v_2, ... ,v_k$ จะได้

$$\underline{u} = c_1v_1 + c_2v_2 + ...+c_kv_k \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง $\underline{u}$ เป็นซับสเปซ k มิติของเซตย่อย ใน $\underline{v}_n$ โดย $c_i$ มีค่าเป็น 0 หรือ 1 ที่ได้จาก GF(2) ซึ่งเรียกว่า เป็นสัมประสิทธิ์ของ $v_i$ และจะมีการรวมอย่างเชิงเส้นของ $v_1, v_2, ..., v_k$ ที่ไม่เหมือนกันอยู่ $2^k$ ชุด ดังนั้นเซตย่อยจะมี $2^k$ เวกเตอร์ ที่ต่างกันและเป็นซับสเปซ k มิติของ $\underline{v}_n$

ถ้าหากข่าวสารที่ส่งจากต้นกำเนิดอยู่ในลักษณะรหัสเลขฐานสอง คือ เป็นชุดลำดับของรหัสเลขฐานสอง ( Sequence of Binary Codes ) กระบวนการเข้ารหัสจะประกอบด้วย 2 ขั้นตอน คือ การแบ่งชุดลำดับของสัญญาณออกเป็นกลุ่มข่าวสาร ( Message Block ) ต่างๆ แต่ละกลุ่มจะประกอบด้วยรหัสข่าวสาร k บิต และการเข้ารหัสแต่ละกลุ่มให้เป็นชุดลำดับรหัสที่ยาว n บิต โดยที่ $(n > k)$ ซึ่งชุดของสัญญาณใหม่นี้เรียกว่า คำรหัส ( Code Word ) หรือ เวกเตอร์คำรหัส ( Code Word Vector ) เขียนแทนด้วยอักษร "$\underline{c}$" เนื่องจากแต่ละชุดข่าวสารจะประกอบด้วยรหัส k บิต ดังนั้นจะมีรหัสข่าวสารที่แตกต่างกัน $2^k$ ชุด หรือจะมี $2^k$ คำรหัสที่แตกต่างกันโดยชุดของ $2^k$ คำรหัสนี้เรียกว่า ชุดรหัสหรือกลุ่มรหัส ( Block Code ) บางครั้งอาจเรียกว่า เวกเตอร์รหัส ( Code Vector ) เพราะเป็น n-ทูเปิ้ลส์จากเวกเตอร์สเปซ $\underline{v}_n$ ของทุกๆ n-ทูเปิ้ลส์

**นิยามที่ 2.1** ชุดของ $2^k$ n-ทูเปิ้ลส์ จะเรียกว่ารหัสเชิงเส้น ถ้าและเพียงแต่ถ้า $2^k$ n-ทูเปิ้ลส์เป็นซับสเปซของเวกเตอร์สเปซ $\underline{v}_n$ ของทุกๆ n-ทูเปิ้ลส์

### 2.2.2 หลักการเข้ารหัสและถอดรหัส ( Principle of Encoding and Decoding )

**1.เมตริกซ์ตัวกำเนิด ( Generator Matrix )**

ถ้า k*n เป็นอะเรย์ ( Array ) ที่มี k แถว ( Row ) และ n หลัก ( Column ) จะได้ว่า

$$G = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{bmatrix} \qquad (2.5)$$

โดยส่วนของ $g_{ij}$ จะเป็น 0 หรือ 1 เท่านั้น อะเรย์ G นี้ เรียกว่า k*n เมตริกซ์ของ GF(2) แต่ละแถวจะมีรหัสเลขฐานสอง n บิต และแต่ละหลักจะมีรหัสเลขฐานสอง k บิต ( เมื่อ $k<n$ ) ถ้าทั้ง k แถวต่างก็มี n-ทูเปิ้ลส์ที่เป็นอิสระเชิงเส้นใน $\underline{v}_n$ ดังนั้นการรวมอย่างเชิงเส้นของแถวต่างๆ ของเมตริกซ์ G จะให้ซับสเปซ k มิติของ $\underline{v}_n$ ซึ่งเรียกว่า โรว์สเปซ ( Row Space ) ของเมตริกซ์ G ถ้าจัดให้จำนวน k คำรหัสเป็นอิสระต่อกันจะได้เมตริกซ์ k*n

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{1,1} & g_{1,2} & g_{1,3} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & g_{2,3} & \cdots & g_{2,n} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k,1} & g_{k,2} & g_{k,3} & \cdots & g_{k,n} \end{bmatrix} \quad (2.6)$$

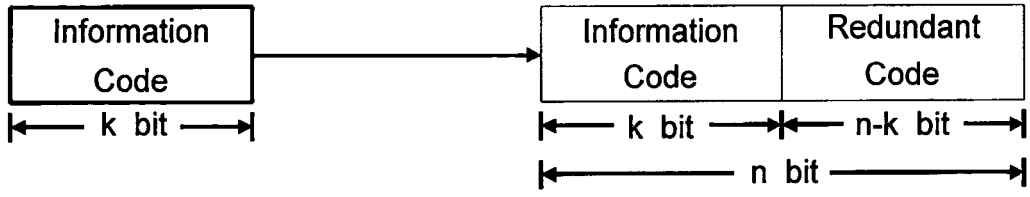
เมื่อ $g_i = ( g_{i,1}, g_{i,2}, g_{i,3}, ..., g_{i,n} )$ และ $i = 1,2,...,k$ กำหนดให้ $\underline{d} = ( d_1 ,d_2,...,d_k )$ คือ ชุดของข่าวสารข้อมูล ( Information Data ) ที่เป็นรหัสเลขฐานสอง ดังนั้นเวกเตอร์คำรหัสที่ได้ คือ

$$\underline{c} = \underline{d}\, G \quad (2.7)$$

$$\underline{c} = ( d_1, d_2, ..., d_k ) \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} \quad (2.8)$$

$$\underline{c} = d_1 g_1 + d_2 g_2 + ... + d_k g_k \quad (2.9)$$

ดังนั้นเวกเตอร์คำรหัสที่สอดคล้องกับชุดรหัสข่าวสาร ( $d_1,d_2,...,d_k$ ) เกิดจากการรวมกันแบบเชิงเส้นของแถวในเมตริกซ์ G และกลุ่มแถวต่างๆ ของเมตริกซ์ G จะเป็นตัวผลิตรหัสเชิงเส้น ซึ่งเรียกเมตริกซ์ G ว่า เมตริกซ์ตัวกำเนิดของรหัส และรหัสเชิงเส้นที่กล่าวนี้ เรียกว่า ชุดรหัสเชิงเส้นขนาด (n,k) บิต โดยในแต่ละชุดจะมีข่าวสารอยู่ k บิต ที่ถูกเข้ารหัสเป็นเวกเตอร์คำรหัสอยู่ n บิต และถูกส่งผ่านออกไปในช่องส่งที่มีสัญญาณรบกวนด้วยอัตราส่วน $R = k/n$ เรียกว่า ความเร็วรหัส ( Code Rate ) เมื่อในเวกเตอร์คำรหัสขนาด n บิต มีรหัสของข่าวสารอยู่ k บิต ก็หมายความว่า ได้มีการเพิ่มรหัสเข้าไปในแต่ละคำรหัสด้วยขนาด (n-k) บิต ซึ่งรหัสที่เพิ่มเข้าไปนี้ เรียกว่า รหัสแก้ไข ( Redundant Code ) โดยจะอยู่ต่อท้ายรหัสข่าวสาร k บิต ดังรูปที่ 2.1




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น สงวนลิขสิทธิ์ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงต้นฉบับทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.1 การเพิ่มรหัสแก้ไขจากการเข้ารหัสแบบชุดรหัสเชิงเส้น**

รหัสที่ได้นี้เรียกว่า รหัสระบบ ( Systematic Code ) โดยที่รหัสแก้ไขนี้จะเป็นตัวแก้รหัสที่ผิดที่เกิดขึ้นในระหว่างการส่งผ่านช่องสัญญาณที่มีสัญญาณรบกวนปะปนอยู่ หรือกล่าวได้ว่า รหัสแก้ไขมีความสามารถที่ช่วยป้องกันและแก้ไขข่าวสารข้อมูลจากการผิดพลาดได้ในระดับหนึ่ง

สำหรับระบบรหัสเชิงเส้น (n,k) บิตแล้ว รูปแบบของเมตริกซ์ตัวกำเนิดขนาด k*n คือ

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,(n-k)} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{2,1} & p_{2,2} & \cdots & p_{2,(n-k)} \\ 0 & 0 & 1 & 0 & \cdots & 0 & p_{3,1} & p_{3,2} & \cdots & p_{3,(n-k)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{k,1} & p_{k,2} & \cdots & p_{k,(n-k)} \end{bmatrix} \quad (2.10)$$

โดยที่ $p_{ij} = 0$ หรือ 1 ให้ Ik เป็นเมตริกซ์เอกลักษณ์ ( Identity Matrix ) ขนาด k*k และให้ p เป็นเมตริกซ์ขนาด k * (n-k) ที่มีส่วนประกอบเป็น $p_{ij}$ ดังนั้นเมตริกซ์ตัวกำเนิดของรหัสระบบเขียนใหม่ได้เป็น

$$G = [\, Ik : Pk*(n-k) \,] \quad (2.11)$$

ถ้าพิจารณาถึงชุดของข่าวสาร $\underline{d} = ( d_1, d_2, ..., d_k )$ แล้ว เมื่อใช้เมตริกซ์ตัวกำเนิดของสมการที่ (2.10) จะได้ชุดคำรหัสเป็น

$$\underline{c} = ( c_1, c_2, c_3, ..., c_n )$$

$$\underline{c} = ( d_1, d_2, ..., d_k ) G \quad (2.12)$$

$$\underline{c} = ( d_1, d_2, ..., d_k ) \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{1,1} & p_{1,2} & \cdots & p_{1,(n-k)} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{2,1} & p_{2,2} & \cdots & p_{2,(n-k)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{k,1} & p_{k,2} & \cdots & p_{k,(n-k)} \end{bmatrix}$$

จากการคูณของเมตริกซ์จะได้

$$\underline{c} = ( c_1, c_2, ..., c_k, c_{n-k+1}, ..., c_n ) \quad (2.13)$$

$$\underline{c} = ( d_1, d_2, ..., d_k, \gamma_1, \gamma_2, ..., \gamma_{n-k} ) \quad (2.14)$$

และ

$$c_{k+j} = p_{1j} d_1 + p_{2j} d_2 + ... + p_{kj} d_k \quad (2.15)$$

สำหรับ j = 1,2,...,n-k จากสมการที่ (2.13) และ (2.14) จะพบว่า รหัส k บิตแรกของเวกเตอร์คำรหัส คือ รหัสข่าวสารที่ต้องการจะส่ง ส่วน (n-k) บิตหลังเป็นฟังก์ชันเชิงเส้นของรหัส

เอกสารนี้มีลิขสิทธิ์ ... ไม่อนุญาตให้ทำซ้ำ ดัดแปลง ... เพื่อการค้า

ข่าวสารซึ่งเรียกว่า รหัสแก้ไข (n-k) บิต ของเวกเตอร์คำรหัสหรือรหัสตรวจสอบ ( Parity Check Code ) ของเวกเตอร์คำรหัส ดังนั้นสมการที่ (2.15) จึงเรียกว่า สมการรหัสตรวจสอบ

**2. เมตริกซ์ตรวจสอบ ( Parity Check Matrix )**

ถ้าพิจารณาถึงอินเนอร์โพรดัคท์ ( Inner Product ) ของสองเวกเตอร์ใดๆ โดยให้ $\underline{v} = (v_1, v_2, ..., v_n)$ กับ $\underline{u} = (u_1, u_2, ..., u_n)$ ทำการอินเนอร์โพรดัคท์กันจะได้

$$\underline{v}.\underline{u} = v_1u_1 + v_2u_2 + ... + v_nu_n \tag{2.16}$$

โดยการบวกและการคูณที่ใช้หลักการของโมดูโล-2 ถ้า $\underline{v}.\underline{u} = 0$ แสดงว่า $\underline{v}$ และ $\underline{u}$ จะตั้งฉากกัน ( Orthogonal ) สำหรับเมตริกซ์ตัวกำเนิดขนาด k*n แถว ที่เป็นอิสระอย่างเชิงเส้น ซึ่งจะมีเมตริกซ์ H ขนาด (n-k)*n

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{(n-k)} \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ \vdots & \vdots & & \vdots \\ h_{(n-k),1} & h_{(n-k),2} & \cdots & h_{(n-k),n} \end{bmatrix} \tag{2.17}$$

เมตริกซ์ H มี $h_j = (h_{j1}, h_{j2}, ..., h_{jn})$ ดังนั้น (n-k) แถวที่เป็นอิสระอย่างเชิงเส้นของเวกเตอร์ g ในโรว์สเปซของเมตริกซ์ตัวกำเนิดจะตั้งฉากกับทุกแถวของเมตริกซ์ H นั่นคือ อินเนอร์โพรดัคท์ ถ้า $g_i$ เป็นเวกเตอร์ในโรว์สเปซของเมตริกซ์ตัวกำเนิดซึ่งจะให้อินเนอร์โพรดัคท์เป็น

$$g_i h_j = 0 \tag{2.18}$$

โดยที่ $1 \le i \le k$ และ $1 \le j \le n-k$ แล้วให้ $\underline{u}$ คือ เวกเตอร์ใดๆ ในโรว์สเปซของเมตริกซ์ตัวกำเนิดกับเวกเตอร์ใดๆ ในโรว์สเปซของเมตริกซ์ H ต่างก็ตั้งฉากซึ่งกันและกัน ดังนั้นโรว์สเปซของเมตริกซ์ตัวกำเนิด เรียกว่า นูลล์สเปซ ( Null Space ) ของเมตริกซ์ H หรือกลับกันเมตริกซ์ H เป็นนูลล์สเปซของเมตริกซ์ตัวกำเนิดและถ้ากำหนดให้ $\underline{c} = (c_1, c_2, ..., c_n)$ เป็นเวกเตอร์ในโรว์สเปซของตัวกำเนิดจะได้

$$\underline{c}\, H^T = 0 \tag{2.19}$$

หรือ

$$\underline{c}\, H^T = c_1h_{i1} + c_2h_{i2} + ... + c_nh_{in} = 0 \tag{2.20}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $i = 1,2,...,n-k$ จึงสรุปได้ว่า $\underline{c}$ เป็นเวกเตอร์คำรหัสที่ได้จากเมตริกซ์ตัวกำเนิด ถ้าและเพียงแต่ถ้า $\underline{c}\,H^T = 0$ จะเรียกเมตริกซ์ H นี้ว่า เมตริกซ์ตรวจสอบ ( Parity Check Matrix ) ถ้าเมตริกซ์ตัวกำเนิดของรหัสระบบเชิงเส้นได้จากสมการที่ (2.10) เมตริกซ์ตรวจสอบของรหัสระบบเชิงเส้นคือ

$$H = \begin{bmatrix} p_{1,1} & p_{2,1} & \cdots & p_{k,1} & 1 & 0 & \cdots & 0 \\ p_{1,2} & p_{2,2} & \cdots & p_{k,2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ p_{1,(n-k)} & p_{2,(n-k)} & \cdots & p_{k,(n-k)} & 0 & 0 & \cdots & 1 \end{bmatrix} \qquad (2.21)$$

$$H = [\, P^T_{(n-k)*k} : I_{n-k} \,] \qquad (2.22)$$

โดย $P^T$ เป็นทรานซ์โพส ( Transpose ) ของเมตริกซ์ P

เมื่อกำหนดให้ $\underline{c} = (c_1, c_2, ..., c_n)$ เป็นเวกเตอร์คำรหัสที่ได้จากสมการที่ (2.7) และได้ส่งผ่านช่องสัญญาณที่มีสัญญาณรบกวนแล้วทางด้านรับ จะได้รับสัญญาณเป็นเวกเตอร์ $\underline{r}$ ( Receiver Codes ) ซึ่งเป็นเวกเตอร์รวมระหว่างเวกเตอร์คำรหัสและเวกเตอร์ของรหัสบิตผิด $\underline{e}$ นั่นคือ $\underline{r} \neq \underline{c}$ จะได้

$$\underline{r} = \underline{c} + \underline{e} \qquad (2.23)$$

หรือ

$$\underline{e} = \underline{r} + \underline{c} \qquad (2.24)$$

$$\underline{e} = (e_1, e_2, ..., e_n) \qquad (2.25)$$

และเวกเตอร์ $\underline{e}$ คือ n-ทูเปิ้ลส์ของรูปแบบรหัสผิด เมื่อ $e_i = 1$ และ $0 \leq i \leq n$ ฉะนั้นชุดถอดรหัสจะต้องนำเอาเวกเตอร์ $\underline{c}$ คืนมาจากเวกเตอร์ $\underline{r}$ โดยชุดถอดรหัสจะหารหัสบิตที่ผิดได้จากการคำนวนหาเวกเตอร์ซีนโดรม ( Vector Syndrome ; $\underline{s}$ ) ซึ่งเป็นเวกเตอร์ขนาด (n-k) ดังนั้นเวกเตอร์ซีนโดรม คือ

$$\underline{s} = (s_1, s_2, ..., s_{n-k}) = \underline{r}\,H^T \qquad (2.26)$$

จากสมการที่ (2.19) ถ้าเวกเตอร์ $\underline{r}$ ที่รับเข้ามาเป็นคำรหัสเดียวกันกับชุดเข้ารหัสแล้ว เวกเตอร์ซีนโดรมจะเป็นศูนย์ ดังนั้นสมการ (2.26) เขียนใหม่ได้เป็น

$$\underline{s} = [\,\underline{c} + \underline{e}\,]H^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

$$\underline{s} = \underline{c}H^T + \underline{e}H^T \quad ; \quad \underline{c}H^T = 0$$

$$\underline{s} = \underline{e}H^T \quad (2.27)$$

จากสมการ (2.27) จะเห็นว่าเวกเตอร์ซีนโดรมที่เกิดจากเวกเตอร์ $\underline{r}$ ที่รับเข้ามาจะเป็นศูนย์ ถ้าเวกเตอร์ $\underline{r}$ เป็นคำรหัสที่ส่งมา แต่ถ้าเกิดความผิดพลาดระหว่างการส่งเวกเตอร์ซีนโดรมของเวกเตอร์ $\underline{r}$ ที่รับเข้ามาจะไม่เป็นศูนย์ ซึ่งค่าของเวกเตอร์ซีนโดรมและค่าเวกเตอร์รหัสบิตผิดจะมีความสัมพันธ์กันในลักษณะที่เวกเตอร์ซีนโดรมจำนวน (n-k) ทูเปิ้ลส์จะมีอยู่ $2^{n-k}$ ของ (n-k) ทูเปิ้ลส์ที่แตกต่างกันในขณะที่รูปแบบบิตผิด ( Coset Leaders ) จะมีอยู่ $2^{n-k}$ รูปแบบ ดังนั้นจะทำให้เกิดความสอดคล้องแบบหนึ่งต่อหนึ่งระหว่างรูปแบบบิตผิดหนึ่งกับจำนวน (n-k) ทูเปิ้ลส์ของเวกเตอร์ซีนโดรมตัวหนึ่ง หรือกล่าวได้ว่า จะมีความสอดคล้องแบบหนึ่งต่อหนึ่งระหว่างหนึ่งรูปแบบบิตผิดกับหนึ่งเวกเตอร์ซีนโดรมทำให้สามารถสร้างตารางของการถอดรหัสที่ประกอบด้วย $2^{n-k}$ รูปแบบบิตที่ผิดกับเวกเตอร์ซีนโดรมที่เกี่ยวข้องได้ เพราะฉะนั้นในการถอดรหัสเวกเตอร์ที่รับเข้ามาจะแบ่งเป็น 3 ขั้นตอน คือ

1. คำนวนหาเวกเตอร์ซีนโดรม จากสมการที่ (2.26)

2. กำหนดรูปแบบบิตที่ผิดของคำรหัส $\underline{e}_i$ จำนวน $2^{n-k}$ รูปแบบแล้วนำไปหาค่าเวกเตอร์ซีนโดรมที่สอดคล้องแบบหนึ่งต่อหนึ่งจากสมการ (2.27)

3.ทำการแก้ไขบิตที่ผิดจนได้คำรหัส $\underline{c}_i$ ซึ่งประมาณได้ว่าเป็นคำรหัสที่ส่งมา จากสมการที่ (2.28)

$$\underline{c}_i = \underline{r} + \underline{e}_i \quad (2.28)$$

### 2.2.3 ขอบเขตของรหัสเชิงเส้น ( Bounds of Linear Block Code )

**นิยามที่ 2.2** น้ำหนักแฮมมิ่ง ( Hamming Weight ) สำหรับรหัสเวกเตอร์ $\underline{c}$ n-ทูเปิ้ลส์ แทนด้วย $w(\underline{c})$ หมายถึง จำนวนบิตของรหัสเวกเตอร์ $\underline{c}$ ที่ไม่เป็นศูนย์

**นิยามที่ 2.3** ให้เวกเตอร์ $\underline{u}$ และ $\underline{v}$ เป็นรหัสเวกเตอร์ขนาด n-ทูเปิ้ลส์ ระยะทางแฮมมิ่ง ( Hamming Distance ) ระหว่างเวกเตอร์ $\underline{u}$ และ $\underline{v}$ เขียนได้เป็น $d(\underline{u}, \underline{v})$ หมายถึง จำนวนบิตรหัสที่เป็น 1 ของผลต่างระหว่างเวกเตอร์ทั้งสอง

จากคำจำกัดความของการบวกแบบโมดูโล-2 จะเขียนได้ใหม่ได้ว่า

$$d(\underline{u}, \underline{v}) = w(\underline{u}, \underline{v}) \quad (2.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นก็หมายความว่า ระยะห่างระหว่างเวกเตอร์ทั้งสอง คือ น้ำหนักผลรวมของเวกเตอร์ทั้งสองในรหัสเชิงเส้นการหาระยะห่างของแต่ละคู่คำรหัสนั้น ระยะห่างที่ต่ำที่สุดเขียนได้เป็น $d_{min}$

**ทฤษฎีบท** กล่าวว่า รหัสเชิงเส้นที่มีระยะทางน้อยที่สุด ( Minimum Distance ; $d_{min}$ ) สามารถแก้ไขความผิดพลาดได้ $(d_{min}-1)/2$ บิตและตรวจสอบความผิดพลาดได้ $d_{min}-1$ บิตในแต่ละคำรหัส โดยที่ค่า $[(d_{min}-1)/2]$ จะเป็นเลขจำนวนเต็ม ( Integer ) ที่มีค่าน้อยกว่า $(d_{min}-1)/2$

**พิสูจน์**

ให้ $\underline{r}$ เป็นเวกเตอร์คำรหัสที่รับได้ $\underline{c}$ เป็นเวกเตอร์คำรหัสที่ส่งให้ $\underline{c}'$ เป็นเวกเตอร์คำรหัสอื่นๆ ระยะทางแฮมมิ่งระหว่างคู่ของเวกเตอร์คำรหัส $\underline{c}$ และ $\underline{c}'$ หรือ $d(\underline{c},\underline{c}')$ กับระยะทาง $d(\underline{c},\underline{r})$ และ $d(\underline{c}',\underline{r})$ จะมีความสัมพันธ์กัน คือ

$$d(\underline{c},\underline{r}) + d(\underline{c}',\underline{r}) \geq d(\underline{c},\underline{c}') \tag{2.30}$$

ถ้า t' บิต เกิดผิดระยะทางแฮมมิ่งระหว่างเวกเตอร์คำรหัส $\underline{c}$ และ $\underline{r}$ คือ $d(\underline{c},\underline{r}) = t'$ และให้ $d_{min}$ เป็นระยะทางน้อยที่สุด จะได้

$$d(\underline{c},\underline{c}') \geq d_{min} \tag{2.31}$$

$$d(\underline{c},\underline{r}) = t' \tag{2.32}$$

$$d(\underline{c}',\underline{r}) \geq d_{min}-t' \tag{2.33}$$

ชุดถอดรหัสจะทำการถอดรหัสให้เป็น $\underline{c}$ เหมือนกับที่ส่งมา ถ้า $d(\underline{c},\underline{r})$ มีค่าน้อยกว่า $d(\underline{c}',\underline{r})$ เมื่อ $d(\underline{c},\underline{r}) < d(\underline{c}',\underline{r})$ จำนวนบิตที่ผิด t' ย่อมน้อยกว่า ระยะทางน้อยที่สุดของผลบวกของ $d(\underline{c},\underline{r})$ และ $d(\underline{c}',\underline{r})$ ดังนี้

$$t' < d_{min}/2 \tag{2.34}$$

เพื่อให้ $t' < d_{min}/2$ เสมอ จึงกำหนดให้ $(d_{min}-1)/2$ เป็นตัวเลขจำนวนเต็มที่มีค่าน้อยกว่า $(d_{min}-1)/2$ เขียนได้ว่า

$$t' \leq [(d_{min}-1)/2] \tag{2.35}$$

ฉะนั้นรหัสเชิงเส้นที่มีระยะทาง $d_{min}$ สามารถแก้ไขบิตที่ผิดได้ t บิต ถ้า $t' \leq \lfloor (d_{min}-1)/2 \rfloor$ โดยรหัสเชิงเส้นขนาด (n,k) บิต จะมีเวกเตอร์ซินโดรม m มิติ ( $m = n-k$ ) จึงมี $2^{m-1}$ เวกเตอร์ซินโดรมที่ไม่ซ้ำกัน และสามารถแก้ไขรูปแบบที่ผิดเพียง 1 บิต ( Single Error Pattern ) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

$2^{m-1}$ รูปแบบ เพราะว่ารหัสเชิงเส้นขนาด (n,k) บิต มีรูปแบบที่ผิดเพียง 1 บิต ได้ n รูปแบบ ดังนั้นรูปแบบที่ผิดทั้งหมดสามารถแก้ไขได้ถ้า

$$2^{m-1} \geq n \qquad (2.36)$$

เมื่อพิจารณาการแก้ไขรหัสผิดจะเห็นว่า ชุดถอดรหัสไม่สามารถจะแก้ไขทุกรูปแบบของรหัสที่ผิดไป $l$ บิตได้เมื่อ $l \geq t+1$ และความสามารถในการตรวจจับบิตที่ผิดของเวกเตอร์คำรหัสนั้น ตัวถอดรหัสจะตรวจจับเวกเตอร์คำรหัสที่มีบิตผิดไม่ได้ ถ้าเวกเตอร์คำรหัสนั้นมี $l$ เป็นจำนวนบิตที่ผิด และ $l > d_{min}$

## 2.3 หลักการสื่อสารข้อมูล ( Principle of Data Communication )

การสื่อสารข้อมูล คือ กระบวนการของการใช้ร่วมหรือแลกเปลี่ยนข้อมูลข่าวสารที่ถูกแปลงรหัส ( Encoded Information ) ระหว่างอุปกรณ์สองตัวหรือมากกว่านั้น ในการสื่อสารข้อมูลตัวส่งและตัวรับเป็นอุปกรณ์หรือตัวเครื่อง และข้อมูลข่าวสารที่ถูกแปลงรหัส หมายถึงข้อมูลข่าวสารที่ถูกส่งผ่านไป ในลักษณะของการส่งผ่านข้อมูลแบบอนุกรมของสัญญาณทางไฟฟ้าผ่านตัวกลางที่ใช้ในการส่งผ่านข้อมูล

**รูปที่ 2.2 ส่วนประกอบของการสื่อสารข้อมูล**

ส่วนประกอบของการสื่อสารข้อมูล ประกอบด้วย

1.DTE ( Data Terminal Equipment ) เป็นอุปกรณ์ต้นทางหรือปลายทางของข้อมูล ทำหน้าที่รับหรือส่งข้อมูลในข่ายคอมพิวเตอร์ โดยอีกนัยหนึ่งคือผู้ใช้ข้อมูล ( Data User )

2.DCE ( Data Communication Equipment ) เป็นอุปกรณ์การสื่อสารข้อมูล ทำหน้าที่ในการเชื่อมต่อและควบคุมการส่งผ่านข้อมูลระหว่างตัว DTE ทั้งสองตัวผ่านตัวกลางในการสื่อสารหรือช่องผ่านสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ม ๔๔๓ ช 2538

ห้องสมุด
คณะครุศาสตร์อุตสาหกรรม ศจ.

3.Medium เป็นตัวกลางที่ใช้ในการสื่อสารข้อมูล หรืออาจจะเรียกว่า ช่องสัญญาณสื่อสาร ( Communication Channel )

**2.3.1 โหมดการสื่อสาร ( Communication Modes )**

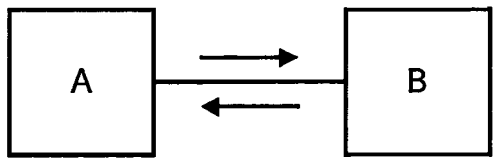
การสื่อสารที่จำแนกตามรูปแบบของการส่งข้อมูล แบ่งออกเป็น 3 แบบ คือ การส่งข้อมูลแบบทางเดียว ( Simplex ), การส่งข้อมูลแบบกึ่งสองทาง ( Half-Duplex ) และการส่งข้อมูลแบบสองทาง ( Full-Duplex ) ซึ่งมีลักษณะดังนี้

**1. การส่งข้อมูลแบบทางเดียว** ข้อมูลถูกส่งได้ไปในทางเดียวเท่านั้น เช่น การส่งข้อมูลจากเทอร์มินัล A ไปยังเทอร์มินัล B โดยเทอร์มินัล A จะเป็นเครื่องส่งและเทอร์มินัล B จะเป็นเครื่องรับเท่านั้น ดังในรูปที่ 2.3 (ก)

**2. การส่งข้อมูลแบบกึ่งสองทาง** ข้อมูลถูกส่งได้ทั้งสองทาง แต่จะต้องผลัดกันส่งและผลัดกันรับ จะส่งพร้อมกันไม่ได้ เทอร์มินัล A ส่งข้อมูลไปให้เทอร์มินัล B ได้และเทอร์มินัล B ก็ส่งข้อมูลตอบกลับมาให้เทอร์มินัล A เช่นกัน ดังในรูปที่ 2.3 (ข)

**3. การส่งข้อมูลแบบสองทาง** ข้อมูลสามารถรับ-ส่ง ได้ในเวลาเดียวกัน เทอร์มินัล A ส่งข้อมูลไปให้เทอร์มินัล B ซึ่งเป็นเวลาเดียวกันที่เทอร์มินัล B ส่งข้อมูลไปให้เทอร์มินัล A โดยที่เทอร์มินัล A และ B ทำงานเป็นอิสระต่อกัน ดังในรูปที่ 2.3 (ค)

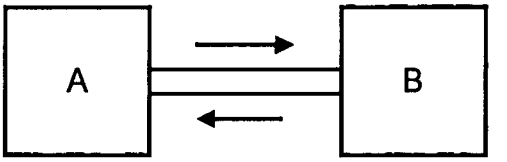
**(ก) แบบทางเดียว**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**(ข) แบบกึ่งสองทาง**

1522
021291

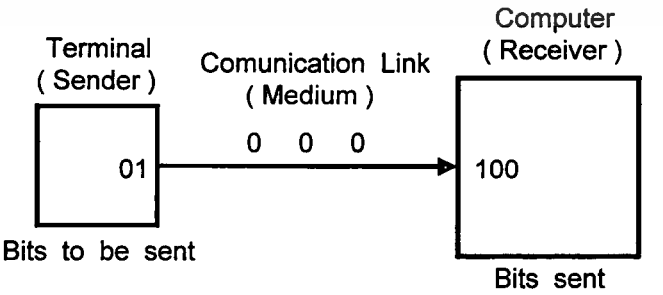



**(ค) แบบสองทาง**

**รูปที่ 2.3 รูปแบบของการส่งข้อมูล**

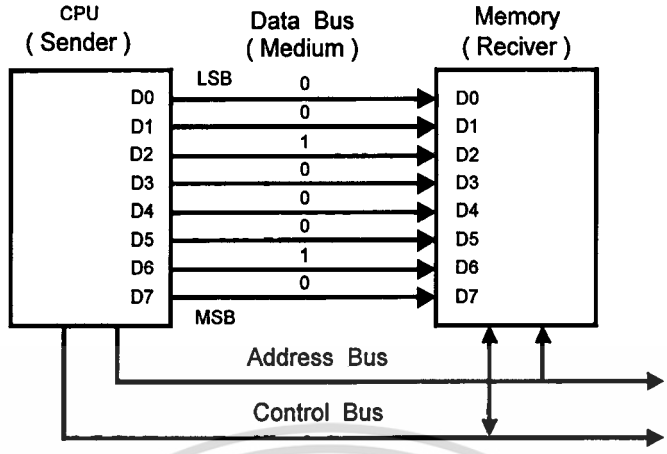
### 2.3.2 เทคนิคของการส่งผ่านข้อมูล ( Data Transmission Techniques )

ในการส่งผ่านข้อมูล ถ้าจำแนกตามลักษณะของการส่งข้อมูลจะแบ่งได้เป็น 2 แบบ คือ การส่งผ่านข้อมูลแบบขนาน และการส่งผ่านข้อมูลแบบอนุกรม การส่งผ่านข้อมูลโดยใช้สายคู่เดียวเป็นการส่งผ่านข้อมูลแบบอนุกรม ( Serial ) และการส่งผ่านข้อมูลโดยใช้สายหลายคู่เป็นการส่งผ่านข้อมูลแบบขนาน ( Parallel ) ในการส่งผ่านข้อมูลแบบอนุกรม ข้อมูลรหัสเลขฐานสองจะถูกส่งออกไปครั้งละ 1 บิต ที่เวลาหนึ่ง ส่วนในการส่งผ่านข้อมูลแบบขนานข้อมูลแต่ละบิตจะมีสายส่งเฉพาะและทุกบิตของข้อมูลที่แต่ละสายส่งจะถูกส่งออกในเวลาเดียวกันดังผังในรูปที่ 2.4 (ก) ที่แสดงตัวอย่างการส่งผ่านข้อมูลแบบอนุกรม จากเทอร์มินัลไปยังคอมพิวเตอร์ รหัสเลขฐานสอง 01000100 จะถูกส่งผ่านเส้นทางข้อมูลภายในระบบคอมพิวเตอร์ เป็นตัวอย่างหนึ่งของการส่งผ่านข้อมูลแบบขนาน ดังผังในรูป 2.4 (ข)




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**(ก) การส่งสัญญาณแบบอนุกรม**




**(ข) การส่งสัญญาณแบบขนาน**

**รูปที่ 2.4 ตัวอย่างของการส่งสัญญาณแบบอนุกรม และการส่งสัญญาณแบบขนาน**

การส่งผ่านข้อมูลแบบขนานย่อมเร็วกว่าการส่งผ่านข้อมูลแบบอนุกรม เพราะว่าทุกบิตถูกส่งออกไปพร้อมกัน ดังนั้นในการส่งผ่านข้อมูลแบบขนาน จะถูกนำมาใช้สำหรับการรับ-ส่งข้อมูลในระยะใกล้ๆ เช่น ในการส่งผ่านข้อมูลระหว่างหน่วยประมวลผล ( CPU ) กับหน่วยความจำ( Memory ) หรือระหว่างหน่วยประมวลผลกับ I/O Chips เป็นต้น อย่างไรก็ตาม การใช้สายหลายๆ เส้นทำให้สับสน ราคาสูง และรับสัญญาณรบกวนได้ง่าย ถ้านำการส่งผ่านข้อมูลแบบขนานมาใช้กับการส่งผ่านข้อมูลระยะไกล เช่น ใช้ในการส่งผ่านข้อมูลระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ หรือคอมพิวเตอร์กับอุปกรณ์รอบนอก เป็นต้น แม้ว่าในบางระบบจะใช้การส่งผ่านข้อมูลแบบขนานสำหรับการส่งผ่านข้อมูลไปยังอุปกรณ์รอบนอก แต่ส่วนใหญ่จะใช้การส่งผ่านข้อมูลแบบอนุกรม ซึ่งมี I/O Chips ทำการแปลงข้อมูลแบบขนานให้เป็นข้อมูลแบบอนุกรมก่อนที่จะส่งออกไป และ I/O Chips เหล่านี้ สามารถรับข้อมูลแบบอนุกรมและแปลงกลับมาเป็นแบบขนานได้เช่นกัน

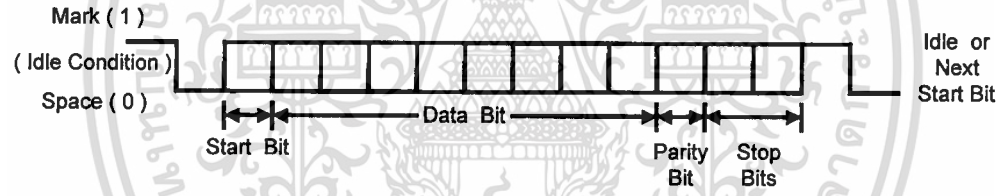
### 2.3.3 การส่งผ่านข้อมูลแบบอนุกรม ( Series Types Data Transmission )

การส่งข้อมูลแบบอนุกรมนั้นแบ่งออกเป็น 2 วิธีคือ การส่งข้อมูลแบบไม่เข้าจังหวะ (Asynchronous Transmission) และการส่งข้อมูลแบบเข้าจังหวะ (Synchronous Transmission)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อทำให้ตัวส่งและตัวรับสามารถทำงานได้สอดคล้องกันทั้งคู่ จึงจะต้องใช้วิธีการส่งข้อมูลแบบเดียวกัน คือ ตัวรับต้องสามารถตรวจจับการเริ่มต้นและการสิ้นสุดของชุดข้อมูลได้สำหรับการส่งข้อมูลแบบไม่เข้าจังหวะ ส่วนแบบส่งข้อมูลแบบเข้าจังหวะนั้นจะดูที่ชุดของการเริ่มต้นและการสิ้นสุดของชุดข้อมูลสัญญาณ

**1. การส่งสัญญาณแบบไม่เข้าจังหวะ** คำว่า "อะซิงโครนัส" หมายความว่า ที่เวลาหนึ่งข้อมูล 1 ชุด จะสามารถถูกส่งออกไปได้การส่งข้อมูลแบบไม่เข้าจังหวะจะใช้กับการส่งข้อมูลที่มีความเร็วต่ำ ( น้อยกว่า 19,200 Bits Per Second (BPS) ) และใช้กับอุปกรณ์ราคาไม่แพงนักจึงมีผลทำให้วิธีนี้เป็นวิธีที่ได้รับความนิยมในการส่งข้อมูล เพราะว่าวงจรง่ายต่อการออกแบบและสามารถลดต้นทุนในการสร้างได้ด้วย วิธีนี้ยังทำให้มีการต่อเนื่องของข้อมูลไม่ตายตัว คือ เวลาระหว่างชุดข้อมูลไม่จำเป็นต้องเท่ากันและรูปแบบชุดข้อมูลของการส่งข้อมูลแบบไม่เข้าจังหวะ แสดงในรูปที่ 2.5




**รูปที่ 2.5 รูปแบบชุดข้อมูลสำหรับการส่งสัญญาณแบบไม่เข้าจังหวะ**

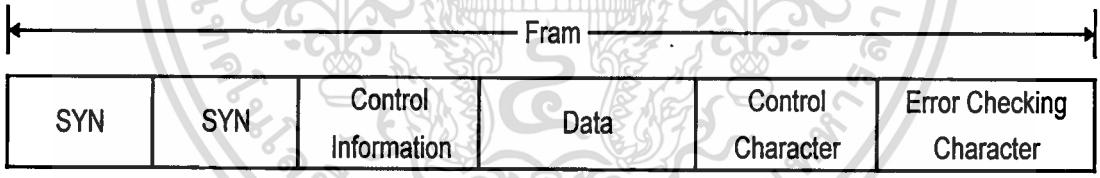
จากรูปที่ 2.5 แสดงรูปแบบของการส่งข้อมูลแบบไม่เข้าจังหวะ จะเห็นว่าในการส่งชุดข้อมูล 1 ชุด แบบไม่เข้าจังหวะจะประกอบด้วย 4 ส่วนด้วยกัน คือ บิตเริ่มต้น ( Start Bit ), บิตข้อมูล ( Data Bit ), บิตตรวจสอบ ( Parity Bit ) และบิตหยุด ( Stop Bit ) ( ซึ่งอาจมีอยู่ 1,1.5 หรือ 2 บิต ) แม้ว่าบิตตรวจสอบจะเป็นบิตตัวเลือกว่ามีหรือไม่มีก็ได้ แต่ระบบส่วนใหญ่จะใช้กันและแม้ว่าการส่งสัญญาณแบบไม่เข้าจังหวะ จะง่ายต่อการออกแบบสร้างและใช้งาน แต่เป็นแบบวิธีการส่งข้อมูลที่ไม่ค่อยมีประสิทธิภาพ

**2. การส่งสัญญาณแบบเข้าจังหวะ** การส่งสัญญาณแบบนี้ไม่ต้องการบิตเริ่มต้นและบิตหยุดในแต่ละเฟรมของข้อมูลโดยการส่งข้อมูลแต่ละครั้งจะส่งเป็นชุดขนาดใหญ่เพื่อที่จะให้เข้า

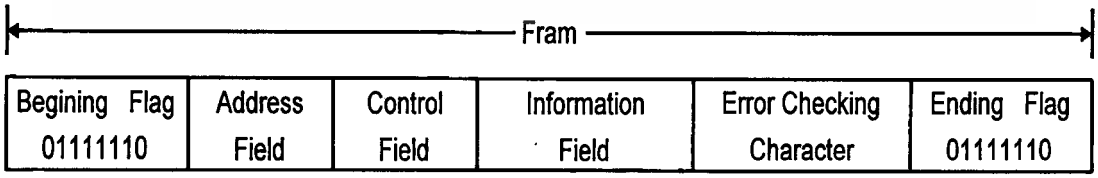
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะกับสัญสัญญาณนาฬิกาของตัวรับ ในการส่งข้อมูลแบบเข้าจังหวะตัวรับสามารถรู้ถึงการเริ่มต้นเฟรมใหม่ของข้อมูลแต่ละเฟรมและแต่ละบิตภายในเฟรม ดังในรูปที่ 2.6 เป็นรูปแบบของการส่งข้อมูลแบบเข้าจังหวะทั้ง 2 แบบที่แตกต่างกัน คือ แบบ Character-Oriented Frame และแบบ Bit-Oriented Frame เฟรมแรกของการส่งข้อมูลแบบ Character-Oriented จะเริ่มต้นด้วยอักขระพิเศษหนึ่งตัวหรือมากกว่านั้น หรือเรียกว่า Synchronization ; ( SYN ) Character อักขระ SYN จะมีรูปแบบทางไบนารี่ที่เป็นเอกลักษณ์ หรือเฉพาะอักขระ SYN จะตามด้วยข่าวสารควบคุม ( Control Information ) ข้อมูลอักขระควบคุมและสุดท้าย คือ อักขระตรวจสอบความผิดพลาด

ส่วนแบบ Bit-Oriented นั้น เฟรมหนึ่งจะประกอบด้วยรูปแบบบิตพิเศษที่ตอนเริ่มต้นและสิ้นสุดของเฟรม รูปแบบเหล่านี้จะประกอบด้วยความยาว 8 บิต ซึ่งเรียกว่า แฟล็ก ( Flags ) แฟล็กเริ่มต้นของเฟรมและสิ้นสุดเฟรมนั้นจะมีค่าเหมือนกัน คือ 01111110 ดังในรูปที่2.6 (ข) ซึ่งได้แสดงแฟล็กเริ่มต้น (Opening Flag) แล้วตามด้วยข้อมูลแสดงที่อยู่ (Address Information) ข้อมูลควบคุม ( Control Information ), ข้อมูลการตรวจสอบข้อผิดพลาด ( Error Checking Information ) และสุดท้าย คือ แฟล็กสิ้นสุด ( Ending Flag )




**(ก) Character Oriented**



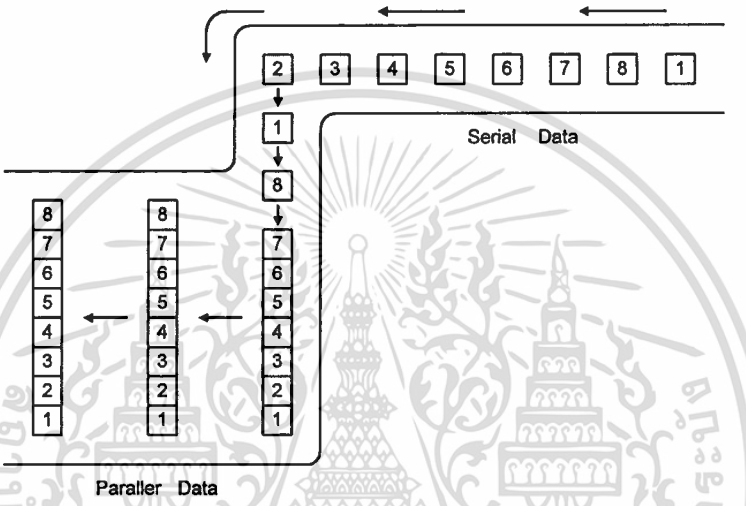

**(ข) Bit Oriented**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

**รูปที่ 2.6 ตัวอย่างรูปแบบการส่งสัญญาณแบบไม่เข้าจังหวะ**

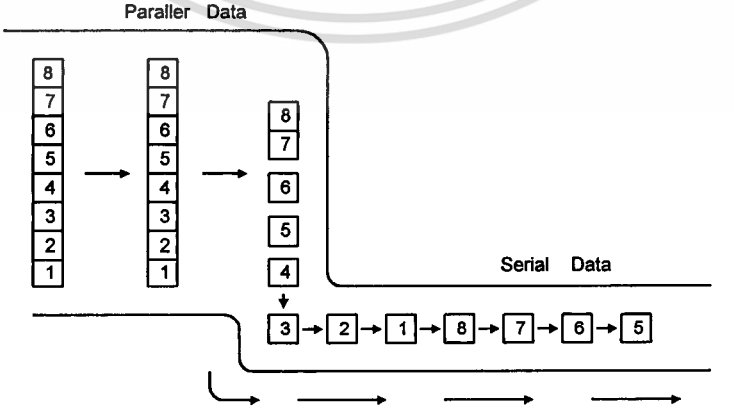
### 2.3.4 การแปลงข้อมูลการติดต่อสื่อสาร ( Data Communication Transfer )

หลักการทำงานของการแปลงรูปแบบข้อมูลโดยอาศัยชิฟต์รีจิสเตอร์ มีหลักการดังนี้ คือ ข้อมูลที่ส่งเข้ามาจะเป็นอนุกรม โดยส่งมาทีละบิตเมื่อเข้ามาถึงรีจิสเตอร์ บิตแต่ละบิตที่เข้ามาจะถูกเรียงในรีจิสเตอร์จนครบจำนวนบิตที่ต้องการ แล้วรีจิสเตอร์จะส่งข้อมูลทั้งชุดออกไป ดังรูปที่ 2.7




**รูปที่ 2.7 การแปลงรูปแบบข้อมูลจากแบบอนุกรมไปเป็นแบบขนาน**

เมื่อต้องการจะแปลงรูปแบบข้อมูลกลับไปเป็นแบบอนุกรมอีก ก็ทำตามกระบวนการที่ตรงกันข้าม ดังในรูปที่ 2.8




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งนี้ ...

**รูปที่ 2.8 การแปลงรูปแบบข้อมูลจากแบบขนานไปเป็นแบบอนุกรม**

**2.3.5ลักษณะช่องสัญญาณการสื่อสาร ( Characteristic of Communication Channel )**

ช่องสัญญาณ คือ ทางเดินสำหรับการไหลของสัญญาณทางไฟฟ้า ซึ่งอาจจะเป็นโลหะเส้นเดียว, สายคู่, โคแอคเชียลเคเบิ้ล ในบางการใช้งานของช่องสัญญาณถูกกำหนดด้วยแถบความถี่ ( Frequency Band ) และคุณสมบัติทางไฟฟ้าของช่องสัญญาณจะเป็นตัวจำกัด อัตราเร็ว ( Rate ) ของข้อมูลที่จะสามารถส่งได้ นอกจากนี้ ควรทราบเกี่ยวกับคำจำกัดความของคำว่าอัตราบิต ( Bit Rate ) และอัตราบอด ( Baud Rate ) ด้วยซึ่งคำว่า "อัตราบิต" คือ จำนวนไบนารี่ดิจิตที่ถูกส่งออกไป ภายใน 1 วินาที ปกติอัตราบิตจะมีค่าเป็น 300, 600, 1200, 2400, 4800,และ 9600 บิตต่อวินาที ( BPS ) ส่วนคำว่า "อัตราบอด" คือ อัตราเร็วบอดที่เป็นตัวกำหนดจำนวนสัญญาณ ( Signaling Elements ) ต่อหน่วยเวลา

## 2.4 อุปกรณ์ควบคุมสำหรับการสื่อสารแบบอนุกรม ( Serial Communication Device )

อุปกรณ์สนับสนุน ( Chip Support ) ที่จำเป็นในการส่งข้อมูลแบบอนุกรม คือไมโครคอนโทรลเลอร์แบบชิ้นเดียว ( Sigle Chip Microcontroller )ซึ่งจัดเป็นไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม ( Integrated Circuit ) เพียงชิ้นเดียวเหมาะสำหรับงานควบคุมอุปกรณ์อื่นๆ แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนรายการคำสั่ง ( Program ) ควบคุมการทำงานได้ตามต้องการ สำหรับไมโครคอนโทรลเลอร์แบบชิ้นเดียวตระกูล 51 หรือ MCS-51 จะมีอยู่หลายเบอร์ด้วยกันซึ่งในปริญญานิพนธ์นี้จะใช้เบอร์ 87C51 ที่มีโครงสร้างและรหัสคำสั่งเหมือนกันกับเบอร์ 8051 จะต่างกันตรงที่เบอร์ 8051 จะมีหน่วยความจำแบบรอม ( ROM ) ขนาด 4 กิโลไบต์ ที่ไม่สามารถเปลี่ยนแปลงคำสั่งควบคุมตัว 8051 ได้ แต่เบอร์ 87C51 จะมีหน่วยความจำแบบอีพรอม ( EPROM ) ขนาด 4 กิโลไบต์ที่สามารถเปลี่ยนแปลงแก้ไขและพัฒนาชุดคำสั่งควบคุมตัว 87C51 ได้

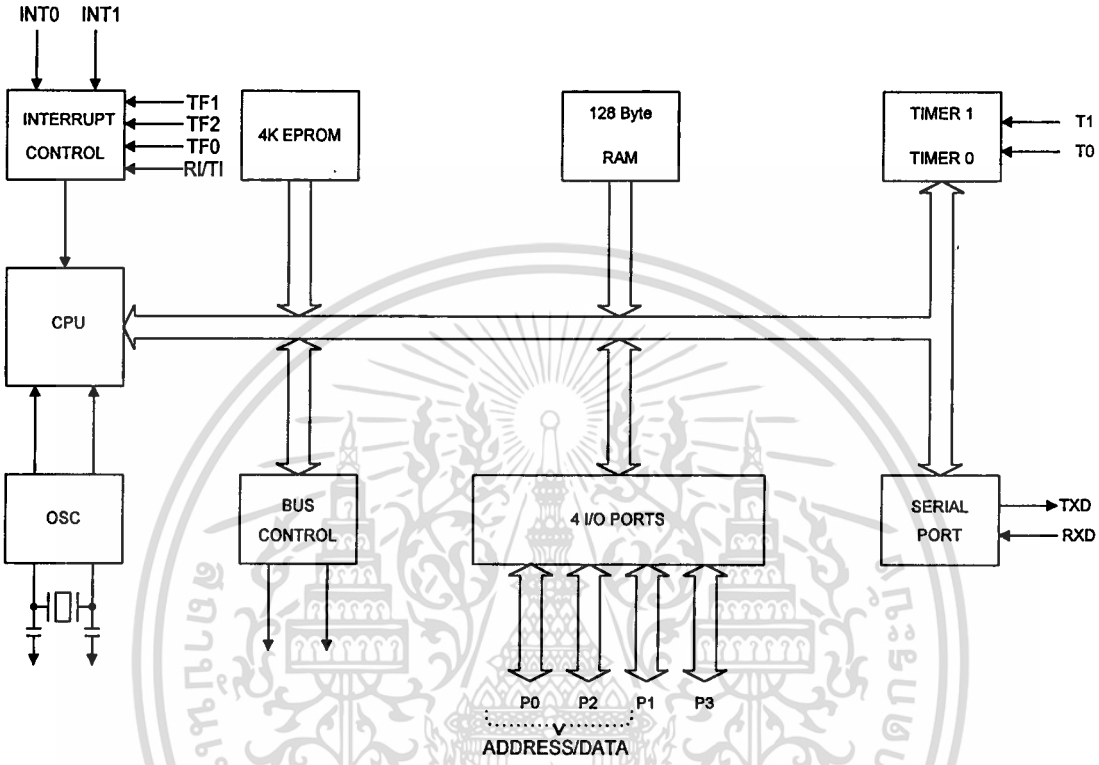
### 2.4.1 โครงสร้างของ 87C51 ( Structure of 87C51 )

ภายในจะประกอบด้วยเกต ( Gate ) ต่างๆ ที่ออกแบบมาให้ทำหน้าที่ต่างกัน เช่น วงจรชุดถอดรหัสคำสั่ง ( Instruction Decoder ) และวงจรสร้างสัญญาณนาฬิกา ( Clock Signal Generator ) เป็นต้น ดังนั้นโครงสร้างภายในของ 87C51 จะประกอบด้วยส่วนย่อยดังนี้

จากรูปที่ 2.9 จะเห็นได้ว่า 87C51 เปรียบเสมือนไมโครคอมพิวเตอร์ จึงประกอบด้วย 3 ส่วนหลักๆ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 คือ หน่วยประมวลผล ( Central Processing Unit ) ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่า หน่วยควบคุม ( Control Unit )




**รูปที่ 2.9 โครงสร้างภายในของ MCS-51**

ส่วนที่ 2 คือ หน่วยความจำ ( Memory Unit ) เป็นส่วนที่มีไว้สำหรับจดจำข้อมูลโดยสามารถที่จะนำข้อมูลเดิมออกไปใช้งาน หรือนำข้อมูลใหม่มาบันทึกลงในหน่วยความจำนี้ได้

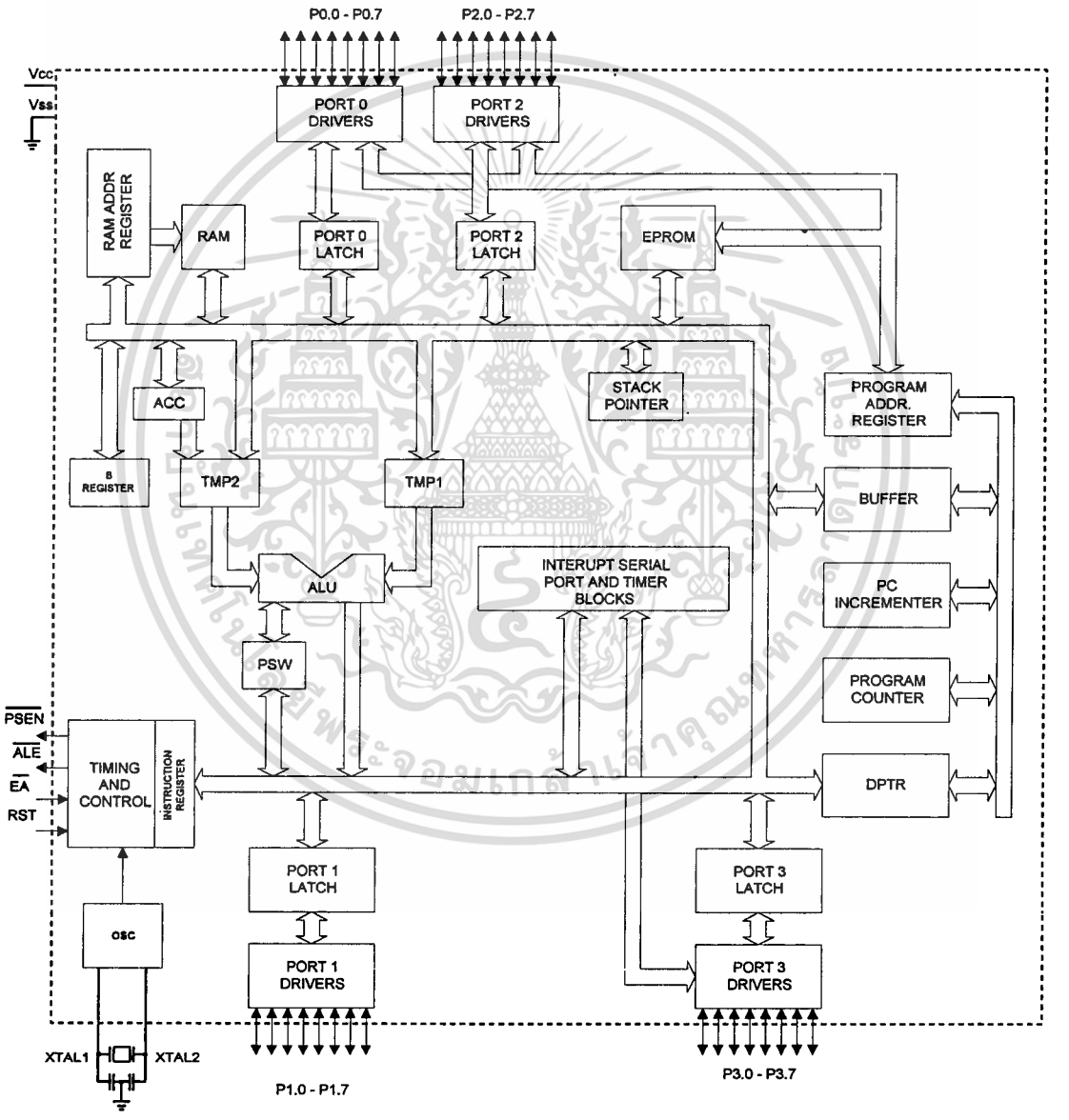
ส่วนที่ 3 คือ อุปกรณ์อินพุตและเอาต์พุต ( Input/Output Device ) เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือออกจาก 87C51 ซึ่งจะทำให้ 87C51 ติดต่อกับภายนอกได้ ดังรูปที่ 2.9 อุปกรณ์อินพุตและเอาต์พุต ได้แก่ 4 I/O Port, Timer 0, Timer 1, Serial Port โดยที่

- 4 I/O Port จะเป็นจุดที่ใช้สำหรับรับ-ส่งข้อมูล ซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัว 87C51 ซึ่งมีพอร์ตทั้งหมดอยู่ 4 พอร์ต โดยแต่ละพอร์ตจะรับ-ส่ง ข้อมูลได้ 8 บิต จะมีพอร์ต P0, P1, P2 และ P3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ การนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอกของ 87C51 หรือ สัญญาณนาฬิกาภายใน 87C51 ก็ได้ ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดยหน่วยประมวลผล

### 2.4.2 สถาปัตยกรรมของ 87C51 ( Architecture of 87C51 )

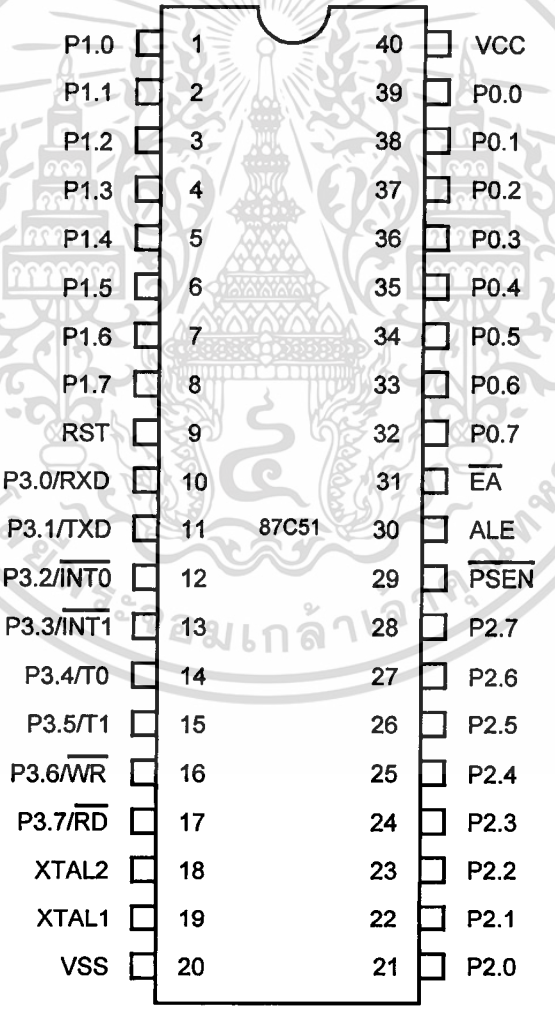



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.10 สถาปัตยกรรมภายในของ 87C51**

- Serial Port หรือ พอร์ตอนุกรม หน่วยประมวลผล จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกจาก 87C51 เรียงไปครั้งละบิตออกจากขา TXD และในการรับข้อมูลก็จะรับเข้ามาครั้งละบิต ทางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิต เพื่อให้หน่วยประมวลผลอ่านไปใช้งานต่อ

ในรูปที่ 2.10 เป็นสถาปัตยกรรมภายในของ 87C51ซึ่งจะอธิบายถึงส่วนย่อยของภายใน 87C51 เพียงชิปเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 87C51 ที่มีอยู่ 40 ขา ดังรูปที่ 2.11




**รูปที่ 2.11 แผนผังขาของ 87C51 แบบ DIP**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

87C51 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package ( DIP ) ซึ่งแต่ละข้างของ 87C51 มีขาอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขานั้นจะใช้งานต่างๆ กันดังนี้ คือ

**Vcc**

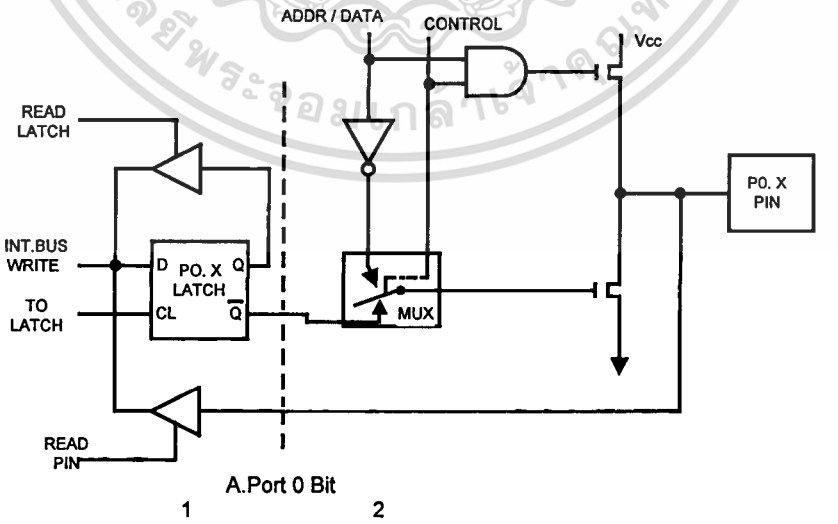
ขา 40 เป็นขาที่ป้อนไฟเลี้ยง +5โวลต์ เข้าไปเพื่อให้วงจรรวมทำงาน ระดับโวลต์เตจของลอจิก 0 และ 1 ของ 87C51 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

**Vss**

ขา 20 เป็นขาที่ต้องต่อกับกราวด์ ( Ground ) ของแหล่งจ่ายไฟการต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

**Port 0**

เป็นพอร์ตขนานขนาด 8 บิต อยู่ที่ขา 39 ถึง 32 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับดังรูปที่ 2.11 แต่ละขาจะเขียนว่า P0.0,P0.1, ..., P0.7 นั้น P0.7 หมายถึงบิต 7 ของพอร์ต 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด ( Most Significant ) และ P0.0 คือ บิต 0 ของพอร์ต 0 เป็นบิตที่มีนัยสำคัญต่ำสุด ( Least Significant ) พอร์ต 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่งและข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ตรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ต 0 จะถูกแลตช์ ( Latch ) ไว้ที่ขาของพอร์ตโครงสร้างแต่ละบิตของพอร์ต 0 เป็นแบบ Open Drain Bidirectional ดังรูปที่ 2.12




**รูปที่ 2.12 โครงสร้างของพอร์ต 0**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.12 เมื่อเปรียบเทียบกับรูปที่ 2.10 ส่วนที่ 1 ของรูป 2.12 ก็คือ Port 0 แลตช์ในรูปที่ 2.10 และส่วนที่ 2 ของรูป 2.12 ก็คือ Port 0 Driver ของรูปที่ 2.10 นั่นเอง

จากโครงสร้างในรูปที่2.12 เมื่อมีคำสั่งการเขียนข้อมูลมายังพอร์ต 0 ข้อมูลจากเส้นทางข้อมูลภายใน Internal Data Bus จะถูกแลตช์ไว้ที่ ดี-ฟลิบฟลอบ ( D-FlipFlop ) โดยสัญญาณ "Write to Latch" ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ต 0 จะอ่านได้ 2 แบบ คือ การอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ตก็จะมีสัญญาณ "Read Latch" มาเพื่ออ่านข้อมูลจาก ดี-ฟลิบฟลอบ กลับเข้าไปยังเส้นทางข้อมูลภายใน การอ่านข้อมูลอีกแบบ ก็คือ การอ่านสภาวะของสัญญาณที่เข้ามาทางพอร์ต 0 และจะมีสัญญาณ "Read Pin" มาควบคุมการอ่านพอร์ต 0 จะใช้งานหลายอย่างดังนี้

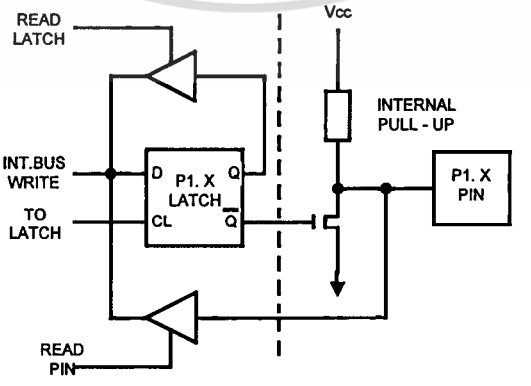
1.ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้ก็คือ 64 กิโลไบต์

2.ใช้รับ-ส่งข้อมูลกับหน่วยความจำข้อมูล ( Data Memory ) หรือใช้รับข้อมูลจากหน่วยความจำรายการคำสั่ง ( Program Memory )

3.ใช้รับ-ส่งข้อมูลผ่านทางพอร์ตโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของหน่วยความจำรายการคำสั่งหรือหน่วยความจำข้อมูลภายนอก

- **Port 1**

เป็นพอร์ตขนานขนาด 8 บิต ในรูปที่ 2.11 คือ ขา P1.0 ถึง P1.7 ( ขา 1-8 ) P1.0 หมายถึง บิต 0 ของพอร์ต 1 ซึ่งเป็นบิตที่มีนัยสำคัญต่ำสุด และบิต P1.7 หมายถึง บิตที่ 7 ของพอร์ต 1 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด โครงสร้างของพอร์ต 1 แต่ละบิตมีดังรูปที่ 2.13




B.Port 2 Bit

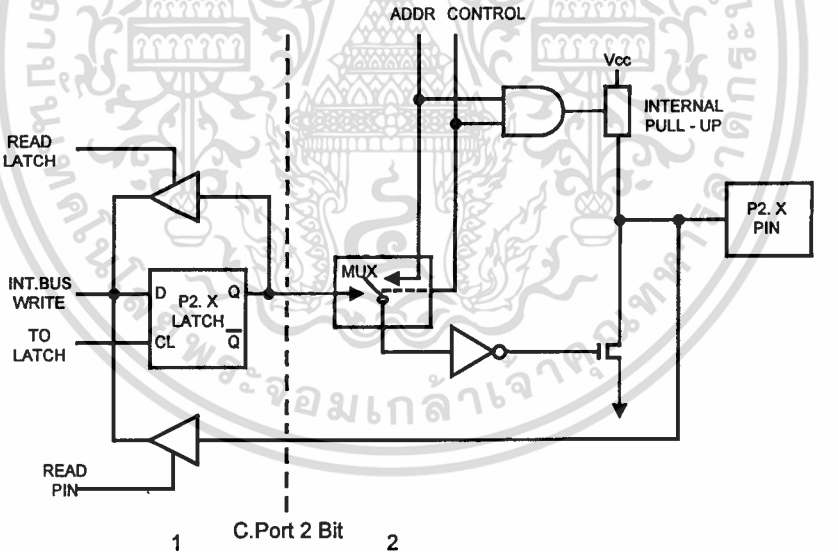
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.13 โครงสร้างของพอร์ต 1**

ส่วนที่ 1 คือ Port 1 แลตช์ ในรูปที่ 2.10 ซึ่งจะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ต 0 ในรูปที่ 2.12 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 2.10 Port 1 Driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ต 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกทางพอร์ต 1 จะถูกแลตช์ไว้แล้วส่งออกไปทางแต่ละขาก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ต 1 จะต้องเขียน 1 ไปยังทุกบิตของพอร์ต 1 เสียก่อน เพื่อให้เฟท ( FET ) อยู่ในสภาวะปิด ( OFF ) ก่อน มิฉะนั้นแล้วถ้าข้อมูล 0 ส่งออกมาค้างอยู่ที่ ดี-ฟลิบฟลอบ จะทำให้เฟทอยู่ในสภาวะเปิด ( ON ) ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงจะเป็น 0 เสมอ

**Port 2**

พอร์ตขนานขนาด 8 บิตคือ ขา P2.0 ถึง P2.7 ( บิต 0 ถึงบิต 7 ของพอร์ต 2) ในรูปที่ 2.11 โครงสร้างของพอร์ต 2 แต่ละบิตจะแสดงดังรูปที่ 2.14




**รูปที่ 2.14 โครงสร้างของพอร์ต 2**

ลักษณะโครงสร้างจะเหมือนกับพอร์ต 0 แตกต่างกันที่ใน Port 2 นั้น ภาค Driver จะใช้งานเพียง 2 ลักษณะ คือ

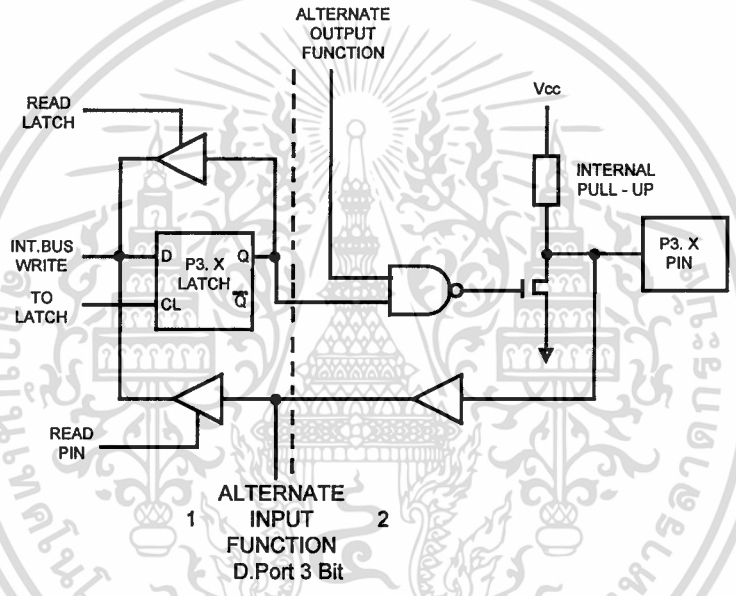
เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิต บนของค่าตำแหน่ง

2. ใช้เป็นพอร์ตรับและส่งข้อมูลกับภายนอก

**Port 3**

คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับดังรูปที่ 2.12 พอร์ตนี้มีโครงสร้างดังรูปที่ 2.15




**รูปที่ 2.15 โครงสร้างของพอร์ต 3**

ส่วนที่ 1 ในรูปที่ 2.15 เป็นส่วนแลตช์ข้อมูลที่เขียนมายังพอร์ต 3 ทางเส้นทางภายใน ( Internal Bus ) เหมือนกับพอร์ตอื่นๆ และพอร์ต 3 จะมี Internal pull up อยู่ทุกบิตแต่พอร์ต3 นี้แต่ละบิตจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ "Alternative Output Function" ที่สร้างจากส่วน Timing and Control สัญญาณ "Alternative Output Function" เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ต 3 ทำงานในฟังก์ชันอื่น และจุด Alternative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่น ตามการทำงานของบิตนั้นแต่ละบิตของพอร์ต 3 มีฟังก์ชันอื่นๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.0/RXD ( Serial Input Port ) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD ( Serial Output Port ) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/$\overline{\text{INT0}}$ ( External Interrupt ) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.3/$\overline{\text{INT1}}$ ( External Interrupt ) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/TO ( Timer/ Couter 0 External Input ) ทำหน้าที่เป็นขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T0 นี้ หรือสัญญาณนาฬิกาก็ได้

P3.5/T1( Timer/ Counter 1 External Input ) เป็นขารับสัญญาณเข้าไปยังวงจร Timer/ Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

P3.6/$\overline{\text{WR}}$ ( External Data Memory Write Strobe ) เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 87C51

P3.7/$\overline{\text{RD}}$ ( External Data Memory Read Strobe ) เป็นขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

**RST**

ขารีเซตขานี้ จะใช้ทำการรีเซตการทำงานของ 87C51 ที่ขารีเซตภายใน 87C51 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ ถ้าป้อนสัญญาณที่สภาวะลอจิก 1 เข้าไปที่ขานี้ จะเป็นการรีเซตการทำงานของ 87C51 ดังนั้นจึงสามารถต่อตัวเก็บประจุ ( Capacitor ) ภายนอกระหว่างขารีเซต กับไฟเลี้ยง +5 โวลต์ เพื่อให้เกิดการรีเซตเมื่อเริ่มป้อนไฟเลี้ยงให้กับ 87C51 ซึ่งเรียกว่า Power on reset การรีเซตจะทำให้ค่ารีจิสเตอร์ต่างๆ เปลี่ยนไปเป็นค่าหนึ่งดังในตารางรูปที่ 2.16

ในตารางรูปที่ 2.16 ช่องทางขวาเป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้าย เมื่อสิ้นสุดการรีเซตในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซตจะมีค่าที่ไม่แน่นอน และพอร์ตจะอยู่ในสภาวะลอจิก 1 ทุกบิตตลอดที่สัญญาณของขารีเซตเป็น High อยู่

เมื่อสัญญาณที่ขารีเซตกลับเป็น 0 ก็จะออกจากการรีเซต 87C51 จะเริ่มทำงานจากคำสั่งที่อยู่ในหน่วยความจำรายการคำสั่งตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์โปรแกรมเคาน์เตอร์ ( Program Counter ; PC ) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H

**ALE**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่า ของสัญญาณนาฬิกาจากออสซิลเลเตอร์สัญญาณนี้จะส่งออกมาตลอดเวลายกเว้นบางครั้งติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 87C51 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอกว่า ขณะนี้สัญญาณนี้

| REGISTER | CONTENT |
|---|---|
| PC | 0000H |
| ACC | 00H |
| B | 00H |
| PSW | 00H |
| SP | 00H |
| DPTR | 0000H |
| P0-P3 | 0FFH |
| IP | 00H |
| IE | 0X000000B |
| TMOD | 00H |
| TCON | 00H |
| T2CON | 00H |
| THO | 00H |
| TLO | 00H |
| TH1 | 00H |
| TH2 | 00H |
| TL1 | 00H |
| TL2 | 00H |
| RCAP2H | 00H |
| RCAP2L | 00H |
| SCON | 00H |
| SBUF | Indeterminate |
| IOCON | 00H |

**รูปที่ 2.16 ค่าของรีจิสเตอร์เมื่อเกิดการรีเซต 87C51**

เป็นลอจิก 1 ( Active ) จะมีการส่งข้อมูลที่เป็น 8 บิตล่าง ของตำแหน่งหน่วยความจำภายนอก 87C51 ที่ต้องการติดต่อออกไปทางพอร์ต 0 และอุปกรณ์ภายนอกจะใช้สัญญาณนี้ ในการแลตช์ข้อมูลไว้เพราะพอร์ต 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ต 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{PSPN}}$

Program Store Enable เป็นขาที่ 29 ในรูปที่ 2.11 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง ( Fetch Instruction ) ที่จะนำไปทำงานมาจากหน่วยความจำ สำหรับโปรแกรมภายนอก 87C51 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำ สำหรับโปรแกรมภายใน 87C51 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา $\overline{\text{PSEN}}$ นี้สามารถต่อไปยังอินพุตของ TTL ชนิด LS ได้ถึง 8 อินพุต

$\overline{\text{EA}}$

External Access ขา 31 ของรูปที่ 2.11 ขานี้เป็นอินพุตที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 2.10 เพื่อควบคุมการสร้างสัญญาณ $\overline{\text{PSEN}}$ ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{\text{EA}}$ นี้ แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงาน ถูกเก็บไว้ภายนอก 87C51 จะต้องสร้างสัญญาณ $\overline{\text{PSEN}}$ ออกไปยังภายนอก เพื่อทำการอ่านคำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา $\overline{\text{EA}}$ เป็น 1 หมายความว่า โปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 87C51 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะทำการอ่านคำสั่งต่างๆ จากอีพรอมภายใน 87C51

**XTAL 1**

ขาที่ 19 ของรูปที่ 2.11 ขานี้จะต่อเข้ากับขาของ วงจรขยายแบบป้อนกลับเฟสสัญญาณ ( Inverting Amplifier ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 2.17 จะเห็นวงจรภายในของออสซิลเลเตอร์ แนนด์เกต ( Nand Gate ) จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นอยู่กับสัญญาณ PD ซึ่งต่อมาจากบิต PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา ( Clock Signal ) จากภายนอกมาเป็นสัญญาณนาฬิกาควบคุมการทำงานของ 87C51 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อคริสตัล ( Crytal ) หรือเซรามิคเรโซเนเตอร์ ดังรูปที่ 2.17 ตัวเก็บประจุควรมีค่าประมาณ 20 pF

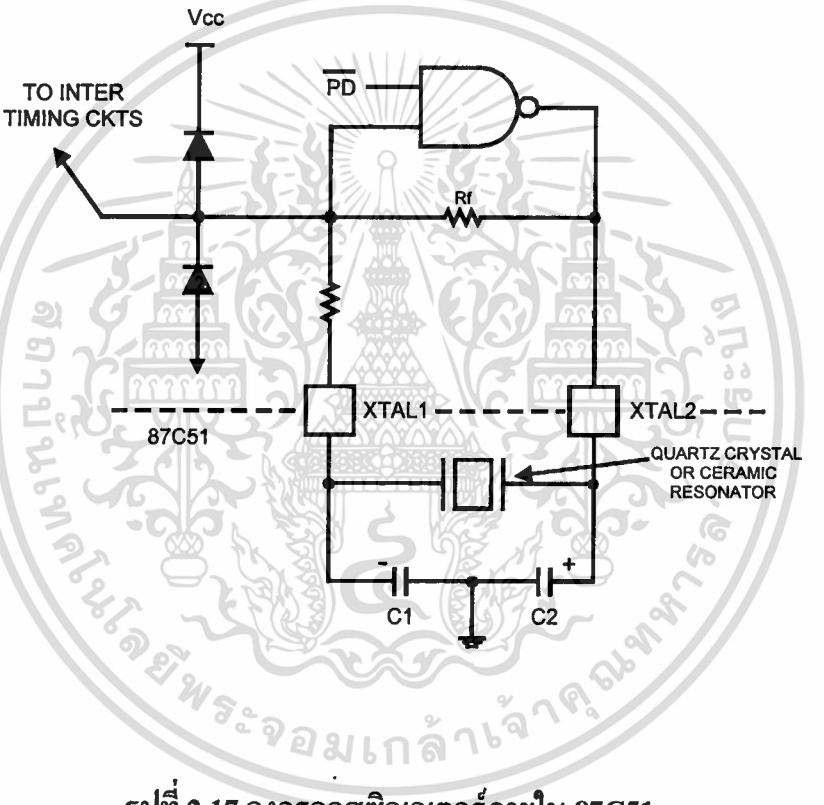
**XTAL 2**

ขาที่ 18 ของรูปที่ 2.11 ขานี้เป็นจุดเอาต์พุตของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ ( อินพุตคือขา XTAL 1 ) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 87C51 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 2.18

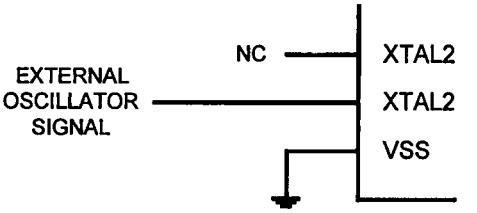
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การทำงานของ 87C51 ( Operating of 87C51 )**

จากรูปที่ 2.10 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 87C51 ซึ่งมีวงจร Power on reset ต่ออยู่จะมีการรีเซตเกิดขึ้น การทำงานภายใน 87C51 จะเริ่มจากชุดโปรแกรมเคาน์เตอร์ ซึ่งเป็นวงจรนับ( Counter Circuit ) ชนิดหนึ่งที่ส่งค่าหน่วยความจำสำหรับโปรแกรมลงไปบนเส้นทางข้อมูล




**รูปที่ 2.17 วงจรออสซิลเลเตอร์ภายใน 87C51**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.18 87C51 ที่ทำงานโดยสัญญาณที่มาจากภายนอก**

( Bus ) หมายเลข 1 ซึ่งมีขนาด 16 บิตค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่รีจีสเตอร์ตำแหน่งโปรแกม ( Program Address Register ) ที่เป็นวงจรแลตช์ข้อมูล ซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บนเส้นทางข้อมูลขนาด 16 บิตหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซตค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้เป็นอีพรอมภายใน หรือรอม,อีพรอมภายนอก 87C51 โดยทำการป้อนสภาวะลอจิกเข้าไปที่ 87C51 ทางขา $\overline{EA}$ ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส แล้วสร้างสัญญาณควบคุมต่อไป ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{EA}$ จะเป็นการเลือกใช้อีพรอมภายใน 87C51 โดยที่วงจร Timing and Control จะสร้างสัญญาณไปยังอีพรอมภายใน ให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมายังเส้นทางข้อมูลหมายเลข 2 ข้อมูลจากอีพรอมจะถูกส่งลงไปยังเส้นทางข้อมูลหมายเลข 3 ที่เรียกว่าเส้นทางข้อมูลภายใน ( Internal Data Bus ) แล้วนำไปเก็บไว้ที่รีจีสเตอร์คำสั่ง ( Instuction Register ; IR ( เป็นวงจรแลตช์ ) ) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสและควบคุมการทำงานส่วนอื่นๆ ต่อไป แล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไรในกรณีที่เลือก รอม หรือ อีพรอม ภายนอก 87C51 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา $\overline{EA}$ จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนเส้นทางข้อมูลหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยังเส้นทางข้อมูลภายใน แล้วไปเก็บที่รีจีสเตอร์คำสั่ง เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจากอีพรอมภายในการทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำ แล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ในรีจีสเตอร์คำสั่ง เรียกว่า เป็นช่วงการเฟตช์ ( Fetch Cycle ) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า ช่วงการเอ็กซีคิ้ว ( Execut Cycle ) เช่น ถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์แอคคิวมูเลเตอร์ ( Accumulator Register ; ACC ) กับข้อมูลจากหน่วยความจำ ข้อมูลภายในแรม ( RAM ) ที่ตำแหน่ง 23H วงจร Timing and Control ก็ส่งสัญญาณให้รีจีสเตอร์คำสั่งส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยังเส้นทางข้อมูลภายในแล้วนำข้อมูลไปเก็บไว้ที่แรมแอดเดรสรีจีสเตอร์ ( RAM Address Register ) เพื่อใช้ชี้ตำแหน่งหน่วยความจำแรม จากนั้น Timing and Control จะสั่งให้แรมส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายังเส้นทางข้อมูลภายในแล้วนำข้อมูลไปเก็บไว้ที่ TMP1 ( วงจรแลตช์ ) ขณะเดียวกันนั้นวงจร Timing and Control ก็จะส่งสัญญาณไปยังรีจีสเตอร์แอคคิวมูเลเตอร์ ให้ส่งข้อมูลมายังTMP2 ( วงจรแลตช์ ) วงจรหน่วยคำนวณ ( Arithematic and Logic Unit ; ALU ) ซึ่งโครงสร้างเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรทำการคำนวณทางคณิตศาสตร์และยังสามารถทำงานทางลอจิก โดยจะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกัน ผลลัพธ์ที่ได้จะส่งผ่านเส้นทางข้อมูลภายในกลับไปเก็บยังรีจิสเตอร์แอคคิวมูเลเตอร์ PSW ( Program Status Word ) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานในหน่วยคำนวณ เช่น ผลลัพธ์การบวกมีค่าเกิน 8 บิต ก็จะทำให้บิตหนึ่งใน PSW ถูกเซตเป็น 1

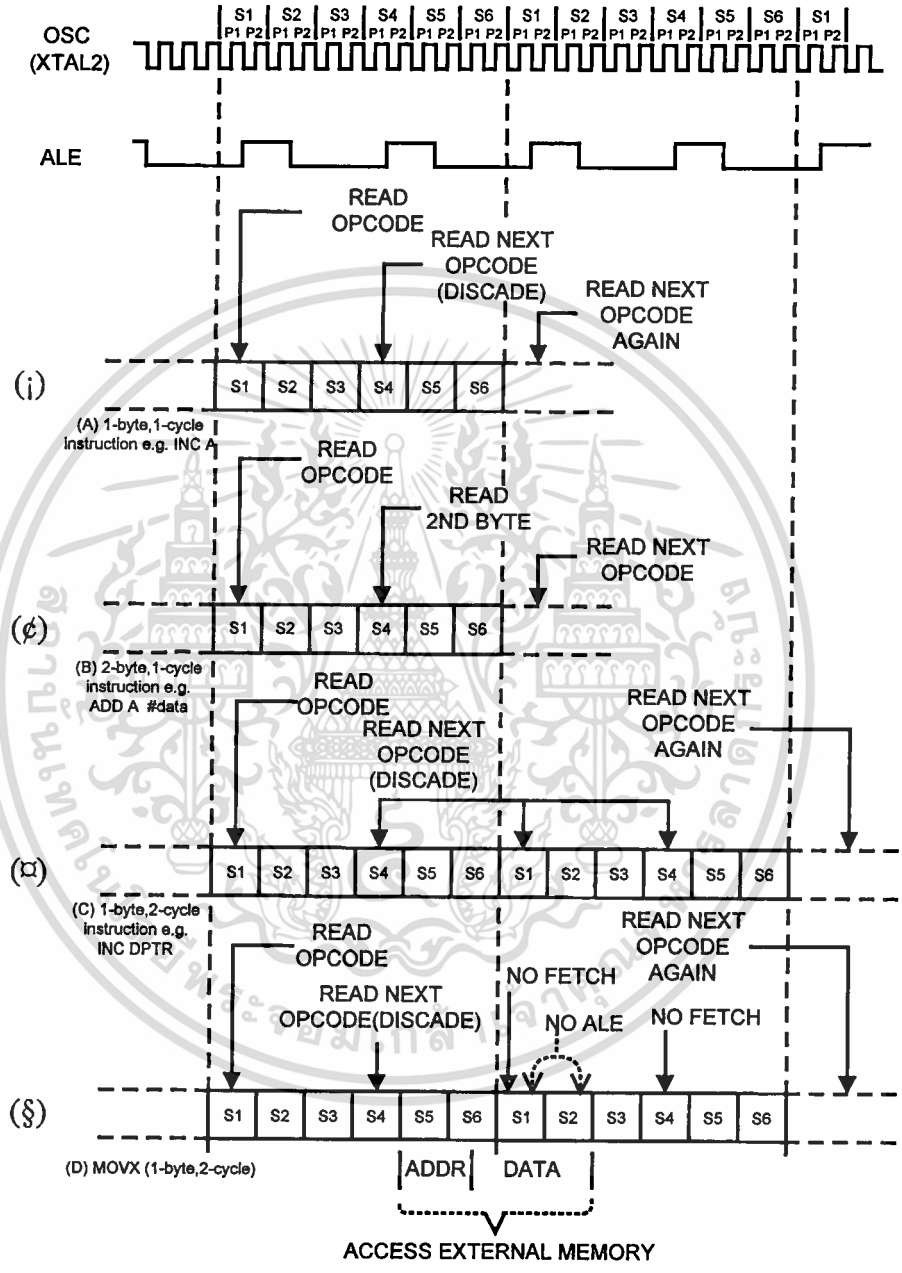
การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจาก วงจรออสซิลเลเตอร์ ทำให้การทำงานต่างๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังในรูปที่ 2.19

คำสั่งแต่ละคำสั่งของ 87C51 จะใช้เวลาทำงาน 1,2 หรือ3 ไซเคิลของเครื่อง ( Machine Cycle ) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไซเคิลของเครื่องจะใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของ 87C51 จะใช้เวลาการทำงาน 12,24 หรือ 36 ไซเคิลของสัญญาณนาฬิกานั่นเอง แต่ละไซเคิลของเครื่องจะถูกแบ่งออกเป็น 6 สเตท ( State ) คือ S1, S2, S3, S4, S5 และ S6 แต่ละสเตทจะประกอบด้วย 2 ไซเคิลของสัญญาณนาฬิกา ในไซเคิลแรกจะเรียกว่าเฟส 1 (P1) และไซเคิลที่ 2 เรียกเฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาลงของสัญญาณนาฬิกาถึงขอบขาลงของสัญญาณนาฬิกาที่อยู่ถัดไป ดังในรูปที่ 2.19 เมื่อ 87C51 ทำงานเสร็จ 1 ไซเคิลของเครื่อง ก็จะเริ่มทำงาน สเตท 1 เฟส 1 ( S1P1 ) ของไซเคิลต่อไป ใน 1 ไซเคิลของเครื่องวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไซเคิล เพื่อเฟตช์คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE

คำสั่งประเภท 1 ไบต์ 1 ไซเคิลของเครื่องได้แก่ คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่านเข้าไปที่เวลาขอบขาขึ้นของสัญญาณ ALE แรก แล้วนำไปเก็บที่รีจิสเตอร์คำสั่งเพื่อให้วงจร Timing and Control ถอดรหัส แล้วเข้าอยู่กับการ เอ็กซีคิ้ว ขณะเดียวกันก็จะเริ่มต้นการเฟตช์คำสั่งที่อยู่ในหน่วยความจำตำแหน่งถัดไปเข้ามาและคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่งเพียง 1 ไบต์ ดังนั้นคำสั่งที่ถูกอ่านมาไบต์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป ( Discard ) ดังในรูปที่ 2.19 (ก)

คำสั่งประเภท 2 ไบต์ และใช้เวลา 1 ไซเคิลของเครื่องได้แก่ คำสั่ง ADD A, # data ในหนึ่งไซเคิลของเครื่องนี้จะมีการอ่านคำสั่งเข้ามา 2 ไบต์เหมือนกับคำสั่งประเภท1ไบต์ 1 ไซเคิลของเครื่อง แต่แตกต่างกันที่ไบต์ที่ 2 จะถูกนำมาใช้งานด้วยไม่ได้ถูกทิ้งไป ดังในรูปที่ 2.19 (ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.19 ลำดับสถานะการทำงานใน MCS-51**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของคำสั่ง ADD A, # 33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบต์ คือ 24 33 เมื่ออ่านคำสั่งไบต์แรก คือ 24 เข้าไปไว้ที่รีจีสเตอร์คำสั่งแล้ว Timing and Control จะถอดรหัส พบว่าเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยังรีจีสเตอร์แอคคิวมูเลเตอร์ ให้นำข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูกอ่านเข้ามาที่รีจีสเตอร์คำสั่งแล้ว Timing and Control จะสั่งให้เอาข้อมูลไบต์ที่ 2 ส่งลงไปยังเส้นทางข้อมูลภายในไปเก็บยัง TMP1 จากนั้นวงจรหน่วยคำนวณจะนำข้อมูลใน TMP1 และ TMP2 มาบวกกัน ผลลัพธ์ที่ได้จะส่งออกจากวงจรหน่วยคำนวณไปยังเส้นทางข้อมูลภายใน แล้วไปเก็บไว้ที่รีจีสเตอร์แอคคิวมูเลเตอร์

คำสั่งประเภท 1, 2 หรือ 3 ไบต์ ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่อง เช่น คำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป 4 ครั้งทุกๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไซเคิลของเครื่อง ถ้าเป็นคำสั่งประเภท 1, 2 หรือ 3 ไบต์ วงจร Timing and Control จะเอาคำสั่ง 1, 2 หรือ 3 ไบต์แรกเท่านั้นไปทำงาน ส่วนคำสั่งที่เหลือทิ้งไปดังในรูปที่ 2.19 (ค) คำสั่ง 1 ไบต์ ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่อง ที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งจะใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำข้อมูลภายนอก การทำงานของคำสั่งนี้จะมีการเฟตช์คำสั่งเข้าไป 2 ไบต์ ในไซเคิลของเครื่องแรก แต่ในไซเคิลของเครื่องที่ 2 จะไม่มีการเฟตช์คำสั่งเข้าไป แต่จะเป็นช่วงเวลาของการอ่านหรือเขียนข้อมูลกับหน่วยความจำข้อมูลภายนอก สัญญาณ ALE ซึ่งปกติจะเปลี่ยนเป็น 1 ที่ S1P2 ก็จะไม่เปลี่ยนเป็น 1 ในไซเคิลของเครื่องที่ 2 โดยจะเป็น 0 อยู่จนกว่าจะถึงเวลา S4P2 ของไซเคิลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อทำการอ่านหรือเขียนข้อมูลกับหน่วยความจำข้อมูลภายนอก

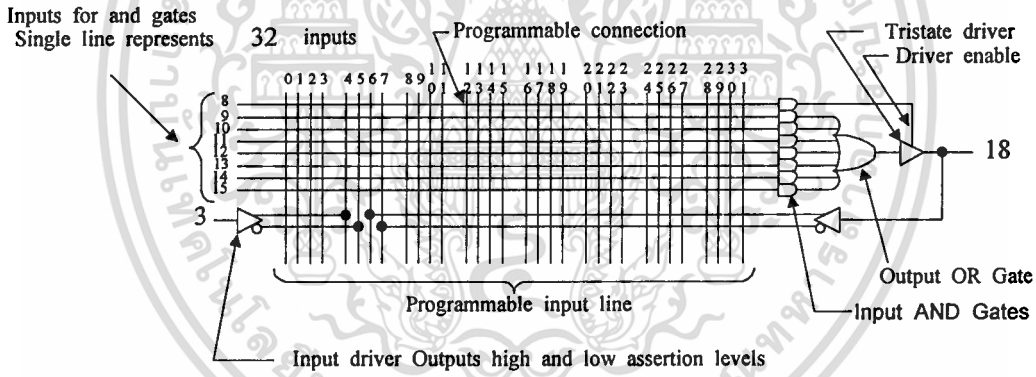
## 2.5 อุปกรณ์ชนิดโปรแกรมได้ ( Program Array Logic ; PAL )

PAL ( Programable Array Logic ) จัดเป็นอุปกรณ์ที่โปรแกรมได้ ( Programable Logic Device; PLD ) ที่บริษัท MMI ( Monolithic Memories Inc. ) เป็นต้นกำเนิดในการผลิต โดย John Birkner ในปี ค.ศ.1976 โครงสร้างภายในของ PALจะทำการโปรแกรมทางด้านแอนด์เกต แต่จะคงส่วนทางด้านออร์เกต มีลักษณะที่เป็นทั้งแบบทีทีแอล ( TTL ) และ ซีมอส ( CMOS ) สำหรับเอาต์พุต ( Output ) ที่มาจาก PAL จะมีทั้งแบบแอคตีฟลอจิก 1 ( Active High ) , แอคตีฟลอจิก 0 ( Active Low ) , แบบเติมเต็ม ( Complementary ), แบบมีการป้อนกลับ และแบบมีรีจิสเตอร์ ( Register ) รูปที่ 2.3

### 2.5.1 ลักษณะโครงสร้าง ( PLD Hardware )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

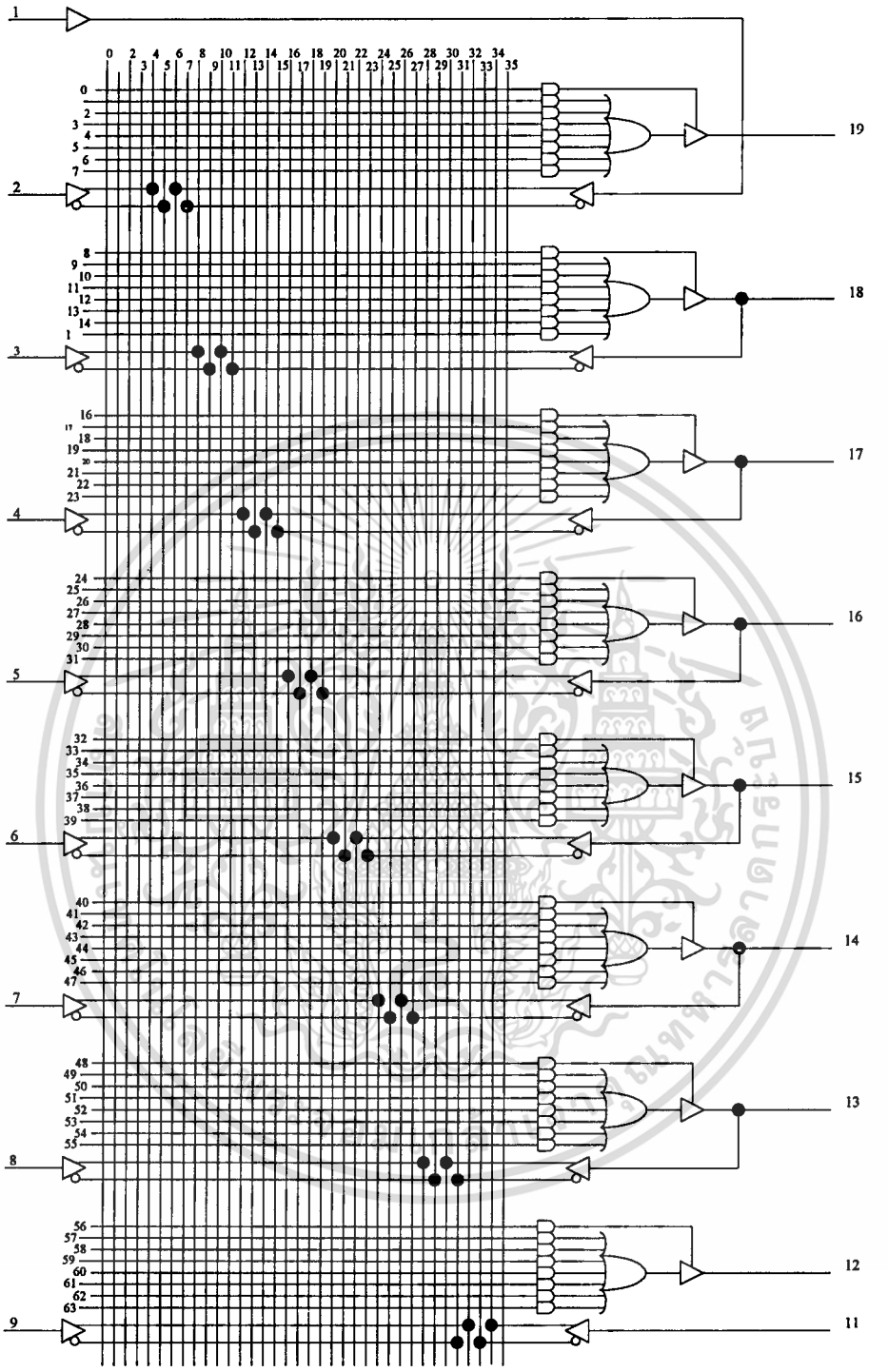
จากรูปที่ 2.20 จะเป็น PAL ที่แสดงค่าเอาต์พุตแบบมีรีจิสเตอร์ที่อธิบายรูปพื้นฐานภายในโครงสร้าง PAL โดยที่ภายในจะมีการต่อถึงกันในแต่ละเส้น เพื่อทำการโปรแกรมระหว่างเส้นในแนวดิ่ง 32 เส้น ซึ่งเราเรียกว่าเส้นอินพุต ( Input Line ) และเส้นที่อยู่ในแนวระดับของอินพุตจะต่อเข้ากับแอนด์เกตและเส้นที่ต่อเข้ากับแอนด์เกตทั้งหมดจะต่อเข้ากับออร์เกต สำหรับขาอินพุตที่ใช้จะไปต่อกับวงจรภายนอกเรียกว่าเป็นตัวขับ ( Driver ) โดยที่ขาอินพุตนี้จะถูกกำหนดให้เป็นได้ทั้งระดับ 1 และระดับ 0 และกรณีขาเอาต์พุต จากรูปจะถูกใช้ให้เป็นได้ทั้งอินพุตและเอาต์พุต เพราะตัวขับเอาต์พุตเป็นแบบ 3 สถานะ ( Tristate ) โดยการโปรแกรมผ่านแอนด์เกตของทางด้านอินพุตของ PAL ถ้าตัวขับเอาต์พุตเปิดทาง ( Enable ) แล้วทางด้านเอาต์พุตจะทำหน้าที่เป็นเอาต์พุต ถ้าตัวขับเอาต์พุตปิดทาง ( Disable ) แล้วขาเอาต์พุตจะทำหน้าที่เป็นอินพุตผ่านทางตัวขับทางด้านอินพุต




**รูปที่ 2.20 การอธิบายพื้นฐานภายในโครงสร้าง PAL**

จากรูปที่ 2.21 เป็นตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL ประกอบด้วย 8 อินพุต 8 เอาต์พุตจะประกอบด้วยอินพุตและเอาต์พุต โดยที่มีขา 2 เป็นอินพุตเหมือนกับการอธิบายในรูปที่ 2.20 รวมทั้งขา1 ก็จะเป็นอินพุตด้วย และให้ผลเอาต์พุตผ่านทางขา 19 และในกรณีของขา 9 และ 11 ก็จะแสดงเป็นอินพุต และส่งเอาต์พุตผ่าน ทางขา 12 สรุปแล้ว PAL นี้มีจำนวนของอินพุตทั้งหมด10 ตัวเป็นการติดต่อจากภายนอกให้ทำหน้าที่เป็นได้ทั้งอินพุตและเอาต์พุต 8 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



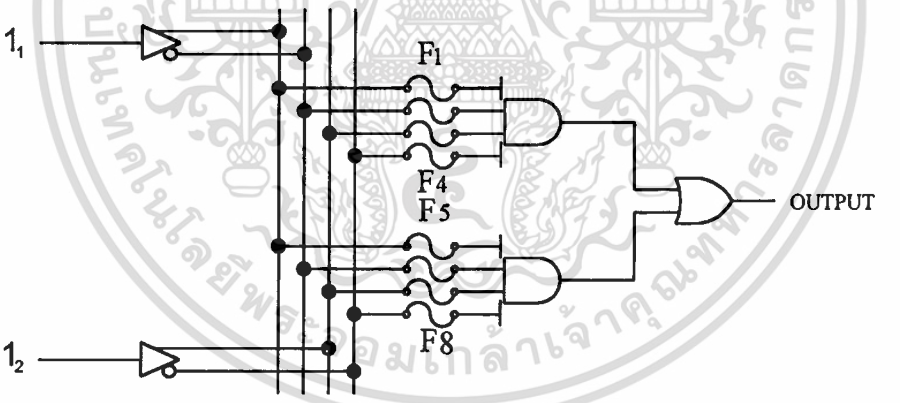
**รูปที่ 2.21 ตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

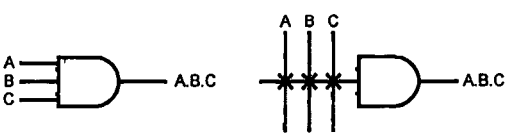
PAL จะอยู่ในตระกูลที่ใช้รูปแบบผลบวกของผลคูณ ( Sum of Product ) ซึ่งจะใช้งานโดยแถวแอนด์ ( And Array ) ก็สามารถโปรแกรมได้ ซึ่งกลุ่มของเอาต์พุตจะต่อไปยังออร์เกต ตั้งแต่ได้มีการใช้รูปแบบผลบวกของผลคูณจึงทำให้สามารถนำสมการบลูลีนมาทำการเปลี่ยนรูปให้สามารถใช้งานในงาน PAL ได้ วงจรภายในของ PAL จะใช้งานอย่างจำกัดตามจำนวนของแถวแอนด์-แถวออร์ อุปกรณ์ PAL จะมีจำนวนของลอจิกที่แตกต่างกันเพื่อจะให้ผู้ใช้สามารถเลือกใช้ตามขนาดใหญ่-เล็กของวงจรที่ออกแบบมา รูปที่ 2.22 แสดงโครงสร้างพื้นฐานของ PAL สำหรับ 2 อินพุต 1 เอาต์พุตสมการลอจิก

$$\text{Output} = (I_1 + \bar{f}_1)(\bar{I}_1 + \bar{f}_2)(I_2 + \bar{f}_3)(\bar{I}_2 + \bar{f}_4) + (I_1 + \bar{f}_5)(\bar{I}_1 + \bar{f}_6)(I_2 + \bar{f}_7)(\bar{I}_2 + \bar{f}_8)$$

โดยที่ f แสดงสถานะของฟิวส์ที่ใช้เชื่อมต่อใน PAL แอนด์อาเรย์ฟิวส์ที่ยังไม่ได้ระเบิดจะแสดงด้วยลอจิก 1 ดังนั้นจึงได้ว่า ฟิวส์ระเบิด f = 0 และ ฟิวส์ยังไม่ถูกระเบิด f = 1

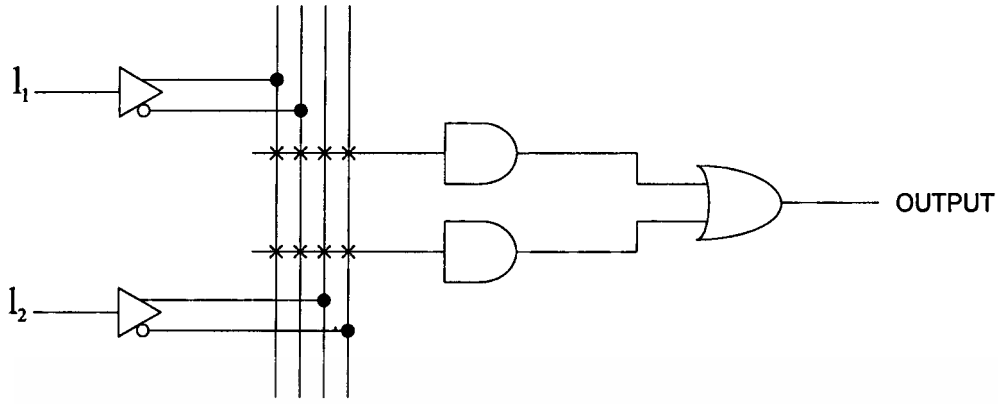



**รูปที่ 2.22 โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.23 การเปลี่ยนแปลงลอจิกโดยใช้ PAL**

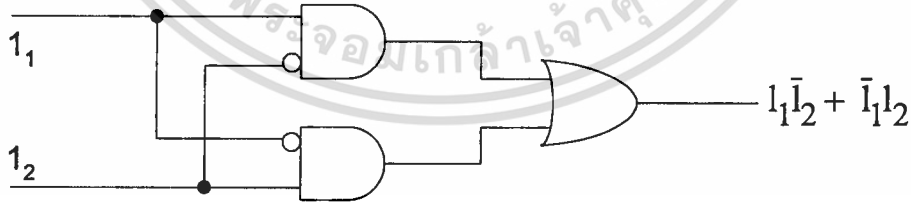



**รูปที่ 2.24 สัญลักษณ์โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต**

การโปรแกรมเลือกวงจรเป็นการส่งสัญญาณกระแสไฟฟ้าเข้าไปเพื่อตัดฟิวส์ที่ไม่ต้องการต่อเข้ากับแอนด์เกตออกให้เหลือเฉพาะสัญญาณที่ต้องการใช้

จากตัวอย่างของ PAL พิจารณาถึงส่วนของผลรวมของเอาต์พุต ที่ได้รูปวงจรลอจิกพื้นฐานที่นำมาประกอบการอธิบาย สามารถเขียนให้อยู่ในรูปของวงจรเมื่อนำมาสร้างโดยใช้ PAL จะได้ดังรูปที่ 2.25

$$\text{Output} = I_1\bar{I}_2 + \bar{I}_1 I_2$$




**รูปที่ 2.25 ตัวอย่างวงจรลอจิกพื้นฐาน** $\text{Output} = 1_1\bar{1}_2 + \bar{1}_1 1_2$

### 2.5.2 ลักษณะคำสั่งใช้งาน ( PLD Software )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม OrCAD/PLD เป็นโปรแกรมที่ใช้สำหรับการออกแบบเพื่อแปลงค่าของ

ลอจิกที่มีการกระทำทางลอจิก การกระทำทางคณิตศาสตร์ หรือ ทฤษฎีของบลูลีน จะทำการแปลงจากโปรแกรม PLD ออกมาเป็น JEDED ไฟล์ การเขียนโปรแกรม PLD ของ OrCAD/PLD จะมีรูปแบบการเขียนที่เข้าใจง่ายสามารถออกแบบวงจรลอจิกได้ตามต้องการ และยังมีฟังก์ชันที่ใช้ในการทำงานของวงจรดิจิตอล ซีเควนเชียล, วงจรนับ และมีโปรแกรมที่ใช้ในการตรวจสอบเวกเตอร์เพื่อตรวจสอบโปรแกรมว่าได้ผลตามที่ต้องการหรือไม่อย่างไร ลักษณะการรายงานผลมีทั้งทางจอภาพและเครื่องพิมพ์ สิ่งที่สำคัญคือ OrCAD/PLD สามารถโปรแกรมอุปกรณ์ PLD ได้ทุกเบอร์ ซึ่งสะดวกแก่การใช้งานมาก

1. สัญลักษณ์ของตัวกระทำ

& แทนการ AND กัน เช่น A & B

# แทนการ OR กัน เช่น A # B

' แทนการ NOT กัน เช่น A'

## แทนการ XOR กัน เช่น A##B

&' แทนการ NAND กัน เช่น (A&B)'

#' แทนการ NOR กัน เช่น (A#B)'

##' แทนการ XNOR กัน เช่น (A##B)'

= แทนการกำหนดค่า เช่น F = A#B

= = แทนการกำหนดค่าในเงื่อนไข เช่น F = = 1

/= แทนเครื่องหมาย

< น้อยกว่า

>= มากกว่าหรือเท่ากับ

-> มากกว่า

+ แทนการบวก เช่น F = A+B

- แทนการลบ เช่น F = A-B

/ แทนการหาร เช่น F = A/B

\ แทนการหารแบบเอาเศษ F= A\B

⇒ แทนการส่งสถานะปัจจุบันไปสถานะถัดไป เช่น ABC DEF

~ แทนการกำหนดช่วง เช่น A[0 1] หมายถึง A0,A1

การกำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In = ขาอินพุต เช่น In : (A,B) หรือ In : A[0 1]

Io = ขาที่เป็นทั้งอินพุตและเอาต์พุต เช่น Io : (A,B)

OUT = ขาเป็นเอาต์พุตเพียงอย่างเดียว เช่น OUT : (A,B)

Clock = ขาสัญญาณนาฬิกา เช่น Clock : clk

Enable = ขาที่ทำหน้าที่เป็นตัวเปิด ปิดการทำงานของเกต

2. คำสั่งที่ใช้ในการเขียนโปรแกรม PLD

Table เป็นตัวกำหนดค่าตารางที่ต้องการจะให้ได้เอาต์พุตตามที่ต้องการ

Map ใช้กำหนดค่าในรูปแบบของวงจรซีเควนเชียล

Procedure เป็นคำสั่งกำหนดการทำงานแบบให้กระโดดไปหา Label หรือ Function ต่างๆ ที่ได้สร้างขึ้น

Display เป็นตัวบอกว่าจะให้แสดงค่าใดออกหน้าจอ

Test เป็นตัวบอกถึงสัญญาณกระตุ้นว่า ต้องการทดสอบค่าใดบ้างเป็นจำลองใน Orcad/VST

* Display clock มีหลายแบบ

( clock )d จะแสดงเป็นเลขฐาน 10

( clock )b จะแสดงเป็นเลขฐาน 2

( clcok )c จะแสดงเป็นรูปสัญญาณนาฬิกา

( clock )o จะแสดงเป็นเลขฐาน 8

* คำสั่งที่ต้องการสั่งให้เป็น A=0 สามารถใช้คำสั่ง Clear A

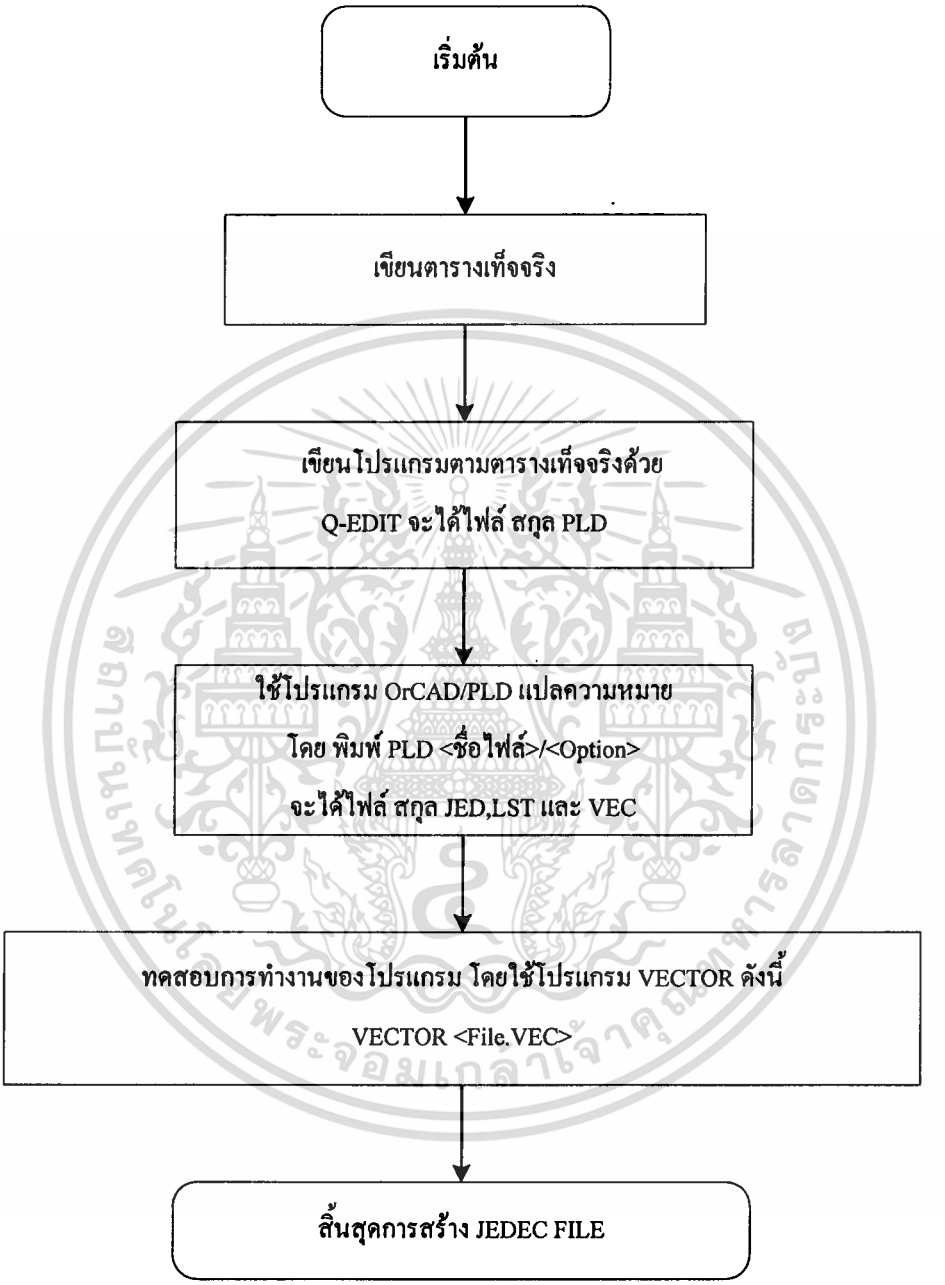
* คำสั่งที่ต้องการสั่งให้เป็น A=1 สามารถใช้คำสั่ง Set A

3. ขั้นตอนการออกแบบแบ่งได้ดังนี้

3.1) สร้าง JEDEC FILE ซึ่งมีขั้นตอนดังผังงานในรูปที่ 2.26

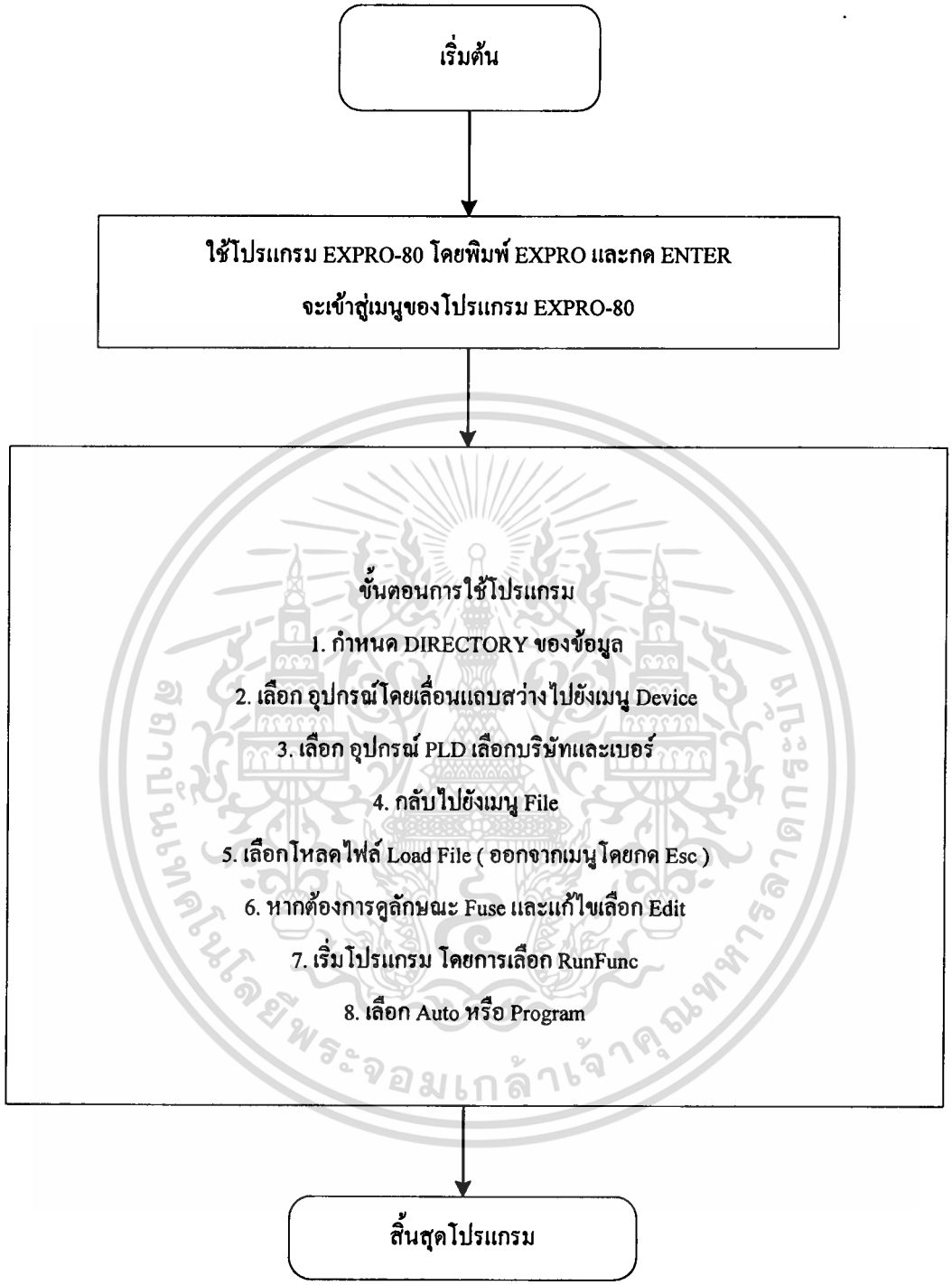
3.2) การโปรแกรมอุปกรณ์ PLD ในเบื้องต้นจะใช้ PAL และ GAL โดยใช้ข้อมูลจาก JEDEC FILE และใช้โปรแกรมของของบริษัท SUNSINE รุ่น EXPRO-80 ซึ่งเป็นโปรแกรมที่ใช้เฉพาะกับเครื่องโปรแกรมของบริษัท SUNSINE มีขั้นตอนดังผังงานในรูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.26 ผังการสร้าง JEDEC FILE**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.27 ผังการโปรแกรมอุปกรณ์ PLD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การสอนเชิงปฏิบัติ

### 2.6.1 บทนำ

หลักการสอนที่สำคัญประการหนึ่ง คือ การสอน โดยให้ผู้เรียนได้ใช้ประสาทสัมผัสหลายทาง นั่นคือ นอกเหนือจากผู้เรียนได้ฟังคำบรรยาย ได้สังเกตการสาธิตแล้ว ผู้สอนต้องจัดกิจกรรมให้ผู้เรียนได้ปฏิบัติ เพื่อให้ผู้เรียนได้ประสบการณ์ตรง

**1. ความสำคัญของการสอนเชิงปฏิบัติ**

การสอนเชิงปฏิบัติมีความสำคัญต่อกระบวนการเรียนการสอน เพราะเป็นการสอนที่ส่งเสริมให้ผู้เรียนนำความรู้ไปประยุกต์ใช้ให้ถูกวิธีด้วยการกระทำ เป็นการขยายและเพิ่มพูนประสบการณ์ นอกเหนือจากการเรียนให้ผู้เรียนสามารถเชื่อมโยงทฤษฎีต่างๆ ไปสัมพันธ์กับปัญหาใช้แก้ปัญหาได้และเป็นการเปิดโอกาสให้ผู้เรียนได้แสวงหาข้อเท็จจริงใหม่ อันนำไปสู่ความรู้ใหม่ๆ เป็นการส่งเสริมความสามารถการเรียนรู้ด้วยตนเองของผู้เรียนซึ่งเป็นจุดมุ่งหมายของการจัดการศึกษาในปัจจุบัน ที่เน้นให้ผู้เรียนมีความสามารถแสวงหาความรู้ด้วยตนเอง

**2. หลักการสอนเชิงปฏิบัติ**

การสอนเชิงปฏิบัติควรมีหลักการที่สำคัญดังนี้

1. เป็นการสอนที่ยึดผู้เรียนเป็นศูนย์กลาง ( Child Center )
2. เป็นการสอนที่ให้ผู้เรียนเรียนรู้ด้วยการกระทำ ( Learning by Doing )
3. เป็นการสอนที่ต้องให้ผู้เรียนได้ปฏิบัติจริง
4. เป็นการสอนที่เน้นทางด้านทักษะพิสัย ( Psychomoter Domain )
5. เป็นการสอนที่ผู้เรียนเรียนรู้ด้วยตนเอง โดยผู้สอนเป็นเพียงผู้คอยช่วยเหลือแนะนำ

### 2.6.2 จุดมุ่งหมายของการสอนเชิงปฏิบัติ

การสอนเชิงปฏิบัติควรมีจุดมุ่งหมายหลัก เพื่อเน้นทางด้านทักษะมุ่งให้ผู้เรียนได้เห็น เข้าใจ สัมผัส และปฏิบัติได้ในสิ่งที่เรียน ฉะนั้นการสอนเชิงปฏิบัติจึงได้นำมาใช้โดยมีจุดมุ่งหมายดังนี้

1. เพื่อพัฒนาความสามารถของผู้เรียน ในการรวบรวมและสัมพันธ์แนวคิด หลักการและความรู้ต่างๆ เข้าด้วยกัน
2. เพื่อประยุกต์หลักการเข้ากับสถานการณ์จริงในการปฏิบัติวิชาชีพนั้นๆ
3. เพื่อให้เห็นถึงปัญหาและพิจารณาทางเลือกในการดำเนินงานปฏิบัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อให้รู้จักวิเคราะห์ ผลของการปฏิบัติและหาทางแก้ไขปัญหา โดยอาศัยความคิดที่มีจุดหมาย ( Aflective Thinking )

5. เพื่อฝึกและพัฒนาความสามารถในการสังเกตรวบรวมและตีความข้อมูลต่างๆ ที่ได้จากการปฏิบัติ

6. เพื่อเข้าถึงทักษะอาชีพปฏิบัติ

7. เพื่อฝึกความสามารถ ในการวางแผนงาน การรายงาน การสรุปผล และการทำงานอย่างมีระบบและกระบวนการ

8. เพื่อส่งเสริมและพัฒนาทัศนคติที่ดีต่องานอาชีพ

การสอนเชิงปฏิบัติ เป็นการสอนโดยให้ผู้เรียนลงมือกระทำและเรียนรู้การแก้ไขปัญหาด้วยตนเอง โดยผู้สอนจะเป็นผู้คอยแนะนำเท่านั้น ซึ่งจะเป็นการกระทำในห้องปฏิบัติการ ( Laboratory ) การปฏิบัติอาจจะปฏิบัติคนเดียวหรือปฏิบัติเป็นกลุ่ม โดยแบ่งงานให้เป็นส่วนๆ แล้วนำผลของการปฏิบัติมาประกอบสัมพันธ์กันเป็นชิ้นงานออกมา

**2.6.3 การดำเนินการสอนเชิงปฏิบัติ**

**1. การเตรียมการสอน**

1. ผู้สอนกำหนดบทเรียน ศึกษาบทเรียนให้เข้าใจ วางโครงการสอนด้วยการลำดับกิจกรรมการสอนให้ชัดเจน และวางวัตถุประสงค์ให้นักเรียนทดลอง เพื่อให้เกิดการเรียนรู้ในประเด็นใด

2. เตรียมเครื่องมือ อุปกรณ์ วัสดุต่างๆ ให้พร้อมเพียงพอกับจำนวนนักเรียน พร้อมทั้งต้องตรวจสอบความสมบูรณ์ของเครื่องมือ

3.ผู้สอนต้องทดลองใช้ให้ได้ผลก่อน เพื่อเป็นแนวทางในการอธิบายและช่วยแก้ปัญหาในขณะปฏิบัติงาน

4. เตรียมใบงานและแบบบันทึกผลการปฏิบัติงาน ( Job Sheet ) ซึ่งลักษณะใบงานส่วนมากจะประกอบด้วย

4.1 วัตถุประสงค์ในการปฏิบัติ

4.2 ทฤษฎีที่จะนำมาปฏิบัติ

4.3 เครื่องมืออุปกรณ์ที่ใช้ในการปฏิบัติ

4.4 ขั้นดำเนินการปฏิบัติ

4.5 ข้อสังเกตสิ่งที่เกิดขึ้นในขณะปฏิบัติ ซึ่งนอกเหนือไปจากทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลสรุปการปฏิบัติ

จากส่วนประกอบของแบบบันทึกผลการปฏิบัติงาน จะเขียนเป็นแบบใบงานได้ ดังรูปที่ 2.28

**ชื่อสถานศึกษา**

แผนกวิชา.....วิชา.....หน่วยการสอน.....

ใบงานที่.....ชื่อ.....ชั้น.....

---

วัตถุประสงค์

1. .....
2. .....
3. .....

ลักษณะทั่วไป

.....

.....

.....

เครื่องมือและอุปกรณ์

1. .....
2. .....
3. .....
4. .....

ภาพประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการปฏิบัติการ

1. .....
2. .....
3. .....
4. .....

ผลการทดลอง

.....

.....

.....

คำถาม

1. .....
2. .....
3. .....

สรุปการทดลอง

.....

.....

.....

.....

แหล่งค้นคว้า

1. .....
2. .....
3. .....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.28 ตัวอย่างใบงาน**

5. เตรียมห้องปฏิบัติงาน โดยติดต่อกับหน่วยงานที่เกี่ยวข้อง

**2. การดำเนินการสอน**

1. ขั้นนำเข้าสู่บทเรียน ผู้สอนอธิบายแนะนำสิ่งที่จะปฏิบัติ อธิบายใบงาน อธิบายอุปกรณ์การใช้วัสดุ ข้อควรระวังในการทดลอง ปัญหาและข้อสังเกต บางกรณีผู้สอนอาจจะสาธิตการปฏิบัติเพื่อให้เห็นเป็นแนวทาง

2. ขั้นปฏิบัติงาน นักเรียนลงมือปฏิบัติด้วยตนเอง ผู้สอนคอยแนะนำให้การปฏิบัติดำเนินการไปตามขั้นตอนที่ต่อเนื่อง คอยชี้แจงเพื่อแก้ปัญหา ชี้แนะข้อสังเกตต่างๆ ตั้งคำถามให้นักเรียนคิดแก้ปัญหา ในขณะที่นักเรียนปฏิบัติผู้สอนต้องเอาใจใส่ตรวจสอบทุกกลุ่มให้ทั่วถึง

3. ขั้นสรุปผล ทำได้หลายวิธีขึ้นอยู่กับลักษณะการปฏิบัติงาน ถ้าเป็นงานที่ต้องปฏิบัติให้เสร็จภายในเวลา อาจจะให้นักเรียนช่วยกันสรุปผลการปฏิบัติท้ายชั่วโมง และบันทึกเป็นรายงานเป็นทฤษฎี ถ้าเป็นชิ้นงานใหญ่ต้องใช้เวลาปฏิบัติหลายครั้งติดต่อกัน อาจจะจัดให้นักเรียนอภิปรายผลการปฏิบัติงาน และแสดงผลการปฏิบัติงานนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

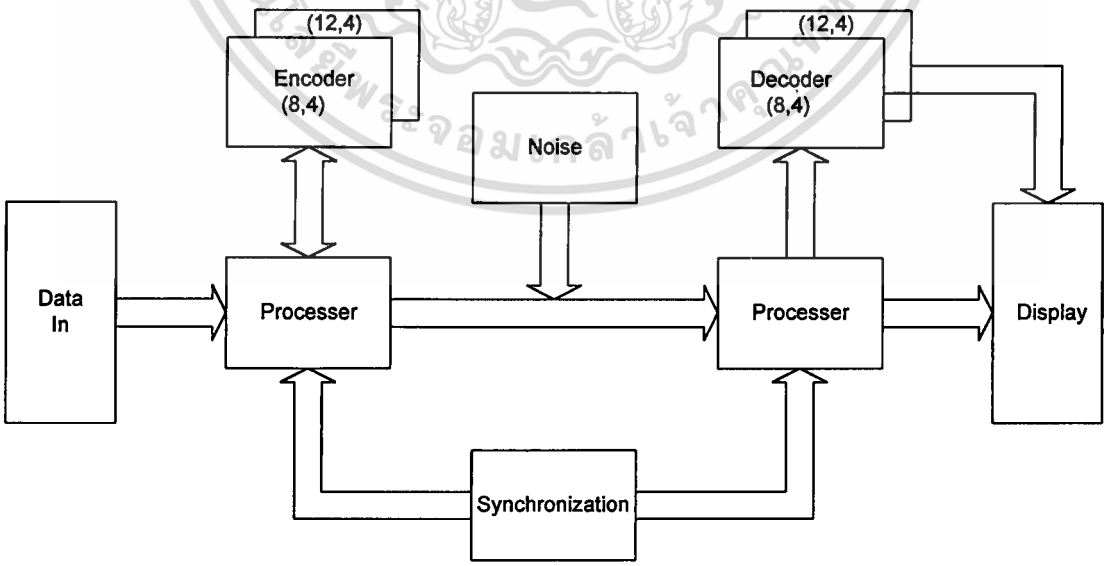
# บทที่ 3

# การออกแบบและการสร้าง

จากเนื้อหาที่กล่าวถึงทฤษฎีและหลักการที่เกี่ยวข้องประกอบกับจุดประสงค์ของปริญญานิพนธ์ฉบับนี้ที่เน้นถึงการทดสอบวิธีการออกแบบ( Design Algorithms ) วงจรเข้ารหัสและวงจรถอดรหัสช่องสัญญาณจากสมการทางคณิตศาสตร์ ในบทนี้จะได้กล่าวถึงแนวความคิดในการวางระบบชุดทดลอง วิธีการออกแบบวงจรเข้ารหัสและถอดรหัสรวมถึงการปรับปรุงแก้ไข เพื่อสามารถที่จะนำไปใช้เป็นแนวทางพื้นฐานในการพัฒนาคิดค้นวิธีการเข้ารหัสและถอดรหัสช่องสัญญาณที่มีประสิทธิภาพต่อไป

## 3.1 การออกแบบระบบชุดทดลองการปฏิบัติการ ( Laboratory Design )

ในการออกแบบชุดทดลองปฏิบัติการเพื่อแสดงการทดสอบวงจรเข้ารหัสและวงจรถอดรหัสช่องสัญญาณนั้น จำเป็นจะต้องพิจารณาถึงการทำงานร่วมระหว่างวงจรเข้ารหัสและวงจรถอดรหัส เพื่อที่จะได้ตรวจสอบการทำงานของวงจรที่ได้ออกแบบไว้ และต้องพิจารณาถึงความเป็นชุดทดลองปฏิบัติการที่สามารถทำให้ผู้ที่ทำการทดลองเกิดการเรียนรู้ตามหลักการสอนเชิงปฏิบัติ ดังนั้นปริญญานิพนธ์นี้ จึงได้ออกแบบระบบชุดทดลองปฏิบัติการ การเข้ารหัสและถอดรหัสช่องสัญญาณ ดังรูปที่ 3.1




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 3.1 ระบบชุดทดลองปฏิบัติการ การเข้ารหัสและถอดรหัสช่องสัญญาณ**

จากรูปที่ 3.1 จะเห็นได้ว่า เป็นระบบที่ใช้ในการสื่อสารข้อมูลโดยแบบทางเดียว ซึ่งในที่นี้จะแบ่งได้เป็น 2 ส่วน คือ

-ส่วนตัวกระทำข้อมูล

-ส่วนควบคุมและแสดงผล

### 3.1.1 ส่วนตัวกระทำข้อมูล ( Operator Section )

ส่วนตัวกระทำข้อมูลตามรูปที่ 3.1 นั้น จะทำหน้าที่จัดการกับข้อมูลที่ได้รับและแบ่งได้เป็น 2 ส่วนโดยในส่วนแรก คือ ส่วนของวงจรเข้ารหัสจะแบ่งออกเป็น 2 ชุด ได้แก่ ชุดเข้ารหัสภายในที่ใช้กับปริญญานิพนธ์นี้ และชุดเข้ารหัสภายนอกที่ผู้ทดลองสามารถประกอบขึ้นเองจากภายนอก และที่ชุดเข้ารหัสนี้จะจัดการกับข้อมูลรหัสเลขฐานสองที่รับเข้ามาในลักษณะที่เพิ่มความซับซ้อนอย่างมีระบบให้กับชุดข้อมูลจนได้เป็นเวกเตอร์คำรหัส และยังสามารถป้อนสัญญาณรบกวนให้กับเวกเตอร์คำรหัสเพื่อใช้ในการทดสอบด้วย จากนั้นจึงส่งเวกเตอร์คำรหัสที่ได้ไปยังส่วนควบคุม และส่วนที่สองเป็นส่วนของวงจรถอดรหัสก็จะแบ่งออกเป็น 2 ชุดเช่นกัน โดยจะทำหน้าที่จัดการกับเวกเตอร์คำรหัสที่รับเข้ามา ซึ่งอาจจะมีการผิดพลาดจากสัญญาณรบกวนจนได้เป็นเวกเตอร์คำรหัสชุดเดิมที่ได้จากชุดเข้ารหัส

### 3.1.2 ส่วนควบคุมและแสดงผล ( Control and Display Section )

ในส่วนนี้จะทำหน้าที่ควบคุมการรับ-ส่งเวกเตอร์คำรหัสระหว่างชุดเข้ารหัสกับชุดถอดรหัสให้มีการทำงานที่สอดคล้องและสัมพันธ์กัน เพราะว่าการส่งเวกเตอร์คำรหัสจากชุดเข้ารหัสไปยังชุดถอดรหัสจะเป็นการส่งข้อมูลแบบอนุกรม รวมทั้งจะแสดงผลที่ได้จากการเข้ารหัสและผลที่ได้จากการถอดรหัสในแต่ละส่วน

## 3.2 การออกแบบวิธีการสร้างรหัสระบบเชิงเส้น ( Systematic Linear Block Codes Algorithms Design )

สำหรับการออกแบบวิธีการสร้างชุดรหัสระบบเชิงเส้นนั้น จะต้องทำการออกแบบทั้งชุดเข้ารหัสและชุดถอดรหัสช่องสัญญาณจากสมการทางคณิตศาสตร์ให้มีความสัมพันธ์กันเพื่อที่จะได้นำมาใช้ทำงานร่วมกันในการเข้ารหัสและถอดรหัสข้อมูลที่ต้องการ ทั้งนี้ในการออกแบบวิธีการสร้างชุดเข้ารหัสและชุดถอดรหัสช่องสัญญาณนั้น จะต้องอาศัยทฤษฎีของรหัสระบบเชิงเส้นที่ได้กล่าวไว้ในบทที่ 2 หัวข้อ 2.2 ซึ่งจะแยกพิจารณาได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 การสร้างชุดเข้ารหัส ( Encoder Implementation )

จากลักษณะการสื่อสารข้อมูลแบบจุดต่อจุด จะสามารถใช้กรรมวิธีเข้ารหัสผ่านความสัมพันธ์ของสมการ

$$\underline{c} = \underline{d}\,G \tag{3.1}$$

หรือ

$$\underline{c} = \sum_{i=1}^{k} digi \quad ; i = 1,2,...,k \tag{3.2}$$

โดยที่ $\underline{d}$ คือ ลำดับข้อมูลขนาด k-ทูเปิ้ลส์ ที่จะถูกนำมาเข้ารหัส

$\underline{c}$ คือ ลำดับคำรหัส n-ทูเปิ้ลส์ ที่ได้จากการเข้ารหัส

และ G คือ เมตริกซ์ตัวกำเนิดขนาดเท่ากับ $[ I_k : P_{k*(n-k)} ]$ ที่กำหนดขึ้นจากรูปแบบข้อมูล

ในสมการที่ (3.1) เป็นสมการเข้ารหัสที่ใช้สำหรับกำหนดชุดเข้ารหัสและลำดับคำรหัสที่ได้จะขึ้นอยู่กับการกระทำของลำดับข้อมูลกับเมตริกซ์ตัวกำเนิดด้วยการคูณ โดยเมตริกซ์ตัวกำเนิดจะประกอบด้วย เบซิสเวกเตอร์ ( Basis Vectors ) ที่ปรากฏอยู่ในแต่ละแถวของเมตริกซ์ตัวกำเนิด ซึ่งจะแสดงได้ตามสมการที่ (3.3) - (3.5)

$$\underline{c} = ( d_1, d_2, ..., d_k ) \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} \tag{3.3}$$

$$( c_1, c_2, ..., c_k, c_{n-k}, ..., c_n ) = ( d_1, d_2, ..., d_k ) [ I_k : P_{k*(n-k)} ] \tag{3.4}$$

$$( c_1, c_2, ..., c_k, c_{n-k}, ..., c_n ) = ( d_1, d_2, ..., d_k ) \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{0,0} & p_{0,1} & \cdots & p_{0,(n-k)} \\ 0 & 1 & \cdots & 0 & p_{1,0} & p_{1,1} & \cdots & p_{1,(n-k)} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & p_{k,0} & p_{k,1} & \cdots & p_{k,(n-k)} \end{bmatrix} \tag{3.5}$$

สำหรับปริญญานิพนธ์ฉบับนี้ได้กำหนดให้

- ลำดับข้อมูล ; $\underline{d}$ มีขนาด 4 บิต ซึ่งจะมีรูปแบบข้อมูลทั้งหมด 16 รูปแบบ
- เมตริกซ์ตัวกำเนิด ; G มีค่าเป็น

$$G = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{3.6}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลำดับคำรหัสที่ได้จากการกระทำ ; $\underline{c}$ มีขนาด 8 บิต ซึ่งมีรูปแบบทั้งหมด 16 รูปแบบ

จากสมการที่ (3.2) จะสามารถนำมาใช้คำนวณเพื่อหาลำดับของคำรหัสจากลำดับข้อมูล โดยกำหนดให้มีค่าเป็น 0111 จะได้ว่า

$$\underline{c} = \sum_{i=1}^{k} digi = d_1g_1 + d_2g_2 + d_3g_3 + ... + d_kg_k \quad (3.7)$$

$$\underline{c} = (d_1, d_2, d_3, d_4) \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \quad (3.8)$$

แทนค่า

$$\underline{c} = (0111) \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\underline{c} = (00000000) + (01001101) + (00100111) + (00011011)$$

$$\underline{c} = 01110001$$

เวกเตอร์คำรหัส ; $\underline{c}$ ที่ได้จะมีขนาด n = 8 บิต ซึ่งจะประกอบด้วยรหัสข้อมูลอยู่ k = 4 บิต และรหัสแก้ไข (n-k) = 4 บิต และรูปแบบของการจัดเรียงของรหัสข้อมูลและรหัสแก้ไข ในลักษณะนี้เรียกว่า รหัสระบบ( Systematic Code ) เนื่องจากในหนึ่งเวกเตอร์คำรหัสจะประกอบด้วยรหัสข้อมูลและรหัสแก้ไข ดังนั้นในส่วนของรหัสแก้ไข จะคำนวณได้จากสมการที่ (3.9)

$$\underline{\gamma} = \underline{d}\,[P_{k*(n-k)}] \quad (3.9)$$

$$(\gamma_1, \gamma_2, ..., \gamma_k) = (d_1, d_2, ..., d_k)\,[P_{k*(n-k)}] \quad (3.10)$$

$$\underline{\gamma} = (d_1, d_2, d_3, d_4) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad (3.11)$$

$$\therefore \quad (\gamma_1, \gamma_2, \gamma_3, \gamma_4) = (d_1+d_2+d_4,\ d_1+d_2+d_3,\ d_1+d_3+d_4,\ d_2+d_3+d_4) \quad (3.12)$$

เมื่อแทนค่าจะได้

$$\underline{\gamma} = (0111) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\underline{\gamma} = (1+0+1,\ 1+1+0,\ 0+1+1,\ 1+1+1)$$

$$\underline{\gamma} = (0001)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉะนั้นในหนึ่งเวกเตอร์คำรหัสจะประกอบด้วย

$$\underline{c} = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) \quad (3.13)$$

โดยที่

$$c_1 = d_1$$

$$c_2 = d_2$$

$$c_3 = d_3$$

$$c_4 = d_4$$

$$c_5 = d_1+d_2+d_4 = \gamma_1$$

$$c_6 = d_1+d_2+d_3 = \gamma_2$$

$$c_7 = d_1+d_3+d_4 = \gamma_3$$

$$c_8 = d_2+d_3+d_4 = \gamma_4$$

$\therefore$ จะได้

$$\underline{c} = (d_1, d_2, d_3, d_4, \gamma_1, \gamma_2, \gamma_3, \gamma_4) \quad (3.14)$$

และจากกระบวนการดังกล่าวจะทำให้ได้เวกเตอร์คำรหัสทั้งหมดดังตารางที่ 3.1

| ลำดับที่ | ลำดับข้อมูล | ลำดับคำรหัส |
|---|---|---|
| 1 | 0000 | 00000000 |
| 2 | 0001 | 00011011 |
| 3 | 0010 | 00100111 |
| 4 | 0011 | 00111100 |
| 5 | 0100 | 01001101 |
| 6 | 0101 | 01010110 |
| 7 | 0110 | 01101010 |
| 8 | 0111 | 01110001 |
| 9 | 1000 | 10001110 |
| 10 | 1001 | 10010101 |

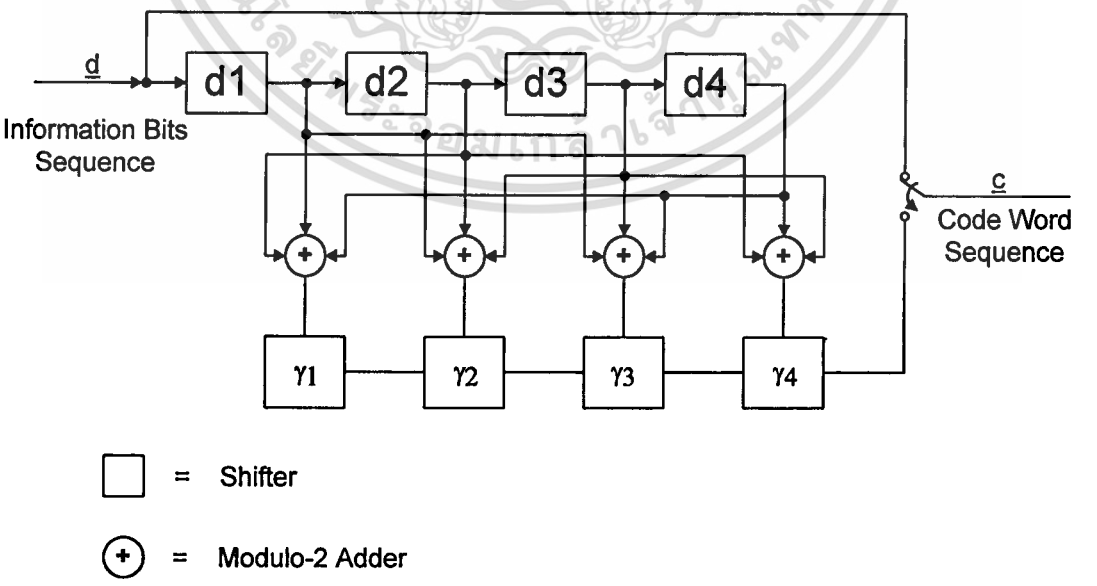
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.1 รูปแบบเวกเตอร์คำรหัสของชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

| ลำดับที่ | ลำดับข้อมูล | ลำดับคำรหัส |
|---|---|---|
| 11 | 1010 | 10101001 |
| 12 | 1011 | 10110010 |
| 13 | 1100 | 11000011 |
| 14 | 1101 | 11011000 |
| 15 | 1110 | 11100100 |
| 16 | 1111 | 11111111 |

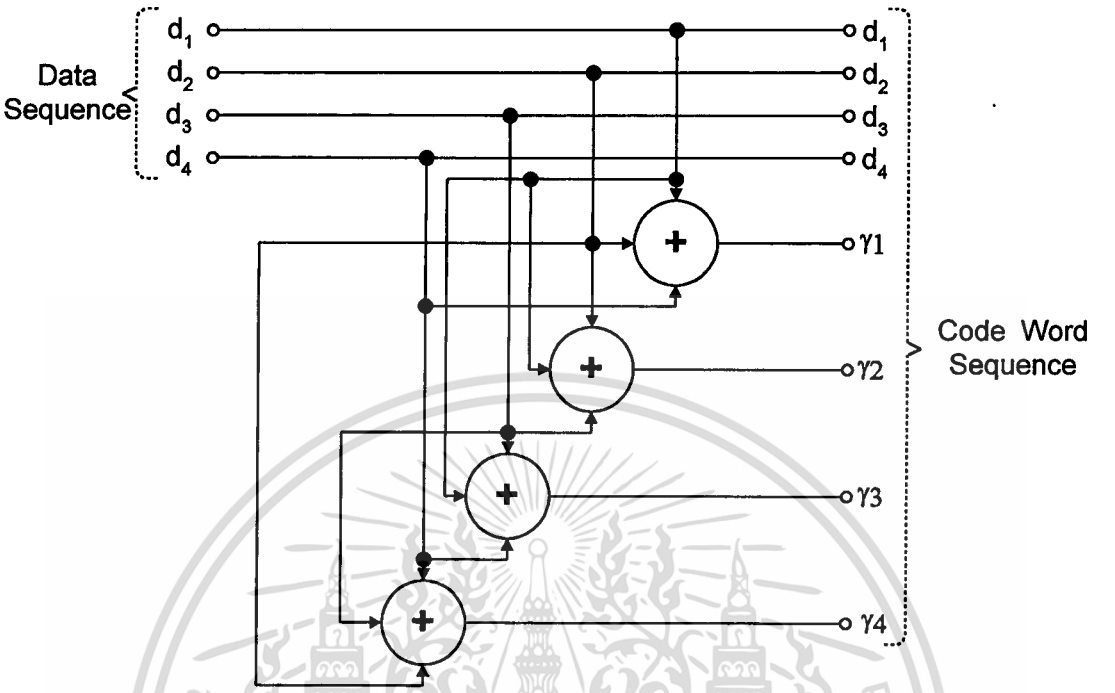
**ตารางที่ 3.1 (ต่อ)**

ดังนั้นชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต จะสามารถจำลองการทำงานแทนกรรมวิธีการคิดทางคณิตศาสตร์ได้ใน 2 ลักษณะตามแบบการส่งผ่านข้อมูลคือ การจำลองในลักษณะของการส่งผ่านข้อมูลแบบอนุกรม ซึ่งจะใช้หลักการเลื่อนข้อมูลและการบวกระหว่างรหัสข้อมูลแบบโมดูโล-2 ดังรูปที่ 3.2 (ก) กับการจำลองในลักษณะของการส่งผ่านข้อมูลแบบขนาน ดังรูปที่ 3.2 (ข)




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

**(ก) ลักษณะการส่งผ่านข้อมูลแบบอนุกรม**




(ข) ลักษณะการส่งผ่านข้อมูลแบบขนาน

**รูปที่ 3.2 แบบจำลองชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

### 3.2.2 การสร้างชุดถอดรหัส ( Decoder Implementation )

ในชุดถอดรหัสจะมีเมตริกซ์ที่ใช้เป็นเครื่องมือสำหรับการถอดรหัสระบบเชิงเส้นขนาด (n,k) บิต ที่เรียกว่า เมตริกซ์ตรวจสอบ ( Parity Check Matrix ; H ) ซึ่งจะประกอบด้วย (n-k) บิต รหัสเวกเตอร์แบบลิเนียร์อินดีเพ็นเด็น ( Lineary Independent Code Vector ) นั่นก็คือ เวกเตอร์ใดๆ ในโรว์สเปซของเมตริกซ์ตัวกำเนิดจะตั้งฉาก ( Orthogonal ) กับแถวเวกเตอร์ในเมตริกซ์ตรวจสอบ ฉะนั้นเมตริกซ์ตรวจสอบจะเขียนได้เป็น

$$H = [\, P^{T}_{(n-k)*k} : I_{n-k} \,] \qquad (3.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยคุณสมบัติของเมตริกซ์ตรวจสอบนี้จะใช้ตรวจสอบเวกเตอร์คำรหัสที่ได้รับว่าเกิดจากเมตริกซ์ตัวกำเนิดที่ตั้งฉากกับเมตริกซ์ตรวจสอบนี้หรือไม่ ถ้าเวกเตอร์คำรหัสนั้นเกิดจากเมตริกซ์ตัวกำเนิดดังกล่าว ผลที่ได้จากการกระทำของเวกเตอร์คำรหัสกับทรานซ์โพสเมตริกซ์ตรวจสอบจะเป็นศูนย์ ดังสมการที่ (3.16)

$$\underline{c}\, H^T = 0 \tag{3.16}$$

จากความหมายของเมตริกซ์ตรวจสอบที่กล่าวมาข้างต้น จะสามารถเขียนเมตริกซ์ตรวจสอบได้เป็น

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.17}$$

และทรานซ์โพสของเมตริกซ์ตรวจสอบ คือ

$$H^T = \left[\frac{P_{k*(n-k)}}{I_{(n-k)}}\right] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

สำหรับปริญญานิพนธ์นี้จะกำหนดให้ $\underline{c}$ เป็นเวกเตอร์คำรหัสที่ถูกส่งจากชุดเข้ารหัส และ $\underline{r}$ เป็นเวกเตอร์คำรหัสที่รับเข้ามาเพื่อนำไปถอดรหัสโดยเวกเตอร์ $\underline{r}$ จะมีค่าเป็น $\underline{r} = \underline{c}+\underline{e}$ โดยที่ $\underline{e}$ เป็นเวกเตอร์แสดงตำแหน่งบิตที่ผิดของเวกเตอร์คำรหัส ซึ่งถ้า $\underline{r} \neq \underline{c}$ ก็จะได้ว่า

$$\underline{e} = \underline{r} + \underline{c} \tag{3.19}$$

$$\underline{e} = (e_0, e_1, \ldots, e_n) \tag{3.20}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อนำเวกเตอร์คำรหัสที่ได้รับมาคูณกับทรานซ์โพสเมตริกซ์ตรวจสอบจะทำให้ได้เวกเตอร์ชุดใหม่กำหนดให้เป็นเวกเตอร์ $\underline{s}$ ที่มีขนาด (n-k) บิต นั่นคือ

$$\underline{s} = \underline{r}\,H^T \tag{3.21}$$

$$\underline{s} = s_1, s_2, \ldots, s_{n-k} = (r_1, r_2, \ldots, r_n) \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix} \tag{3.22}$$

เวกเตอร์ $\underline{s}$ นี้เรียกว่า เวกเตอร์ซีนโดรม ( Vector Syndrome ) หรือซีนโดรมความผิดพลาด ( Error Syndrome ) ของเวกเตอร์ $\underline{r}$ ซึ่งถ้า $\underline{r} = \underline{c}$ เวกเตอร์ซีนโดรม ; $\underline{s}$ มีค่าเท่ากับศูนย์ แต่ถ้า $\underline{r} \neq \underline{c}$ จะทำให้เวกเตอร์ซีนโดรมไม่เท่ากับศูนย์แต่จะมีค่าเป็น

$$\underline{s} = [\underline{c}+\underline{e}]\,H^T \tag{3.23}$$

$$\underline{s} = \underline{c}H^T + \underline{e}H^T \quad ; \underline{c}H^T = 0$$

ดังนั้น

$$\underline{s} = \underline{e}H^T \tag{3.24}$$

จากสมการที่ (3.24) แสดงให้เห็นว่า เวกเตอร์ซีนโดรมมีส่วนเกี่ยวข้องกับความผิดพลาดของเวกเตอร์คำรหัสที่รับเข้ามา ซึ่งจะสามารถนำเอาเวกเตอร์ซีนโดรมนี้ไปใช้ในการตรวจสอบ ( Detect ) และ แก้ไข ( Correct ) ความผิดพลาดของเวกเตอร์ $\underline{r}$ ที่รับเข้ามาได้ ถ้าหากเวกเตอร์คำรหัสที่รับได้มีค่าเป็น $\underline{r} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8)$ จะทำให้สามารถคำนวณหาค่าของเวกเตอร์ซีนโดรมได้ดังนี้

$$\underline{s} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

$$(s_1, s_2, s_3, s_4) = (r_1+r_2+r_4+r_5,\ r_1+r_2+r_3+r_6,\ r_1+r_3+r_4+r_7,\ r_2+r_3+r_4+r_8)$$

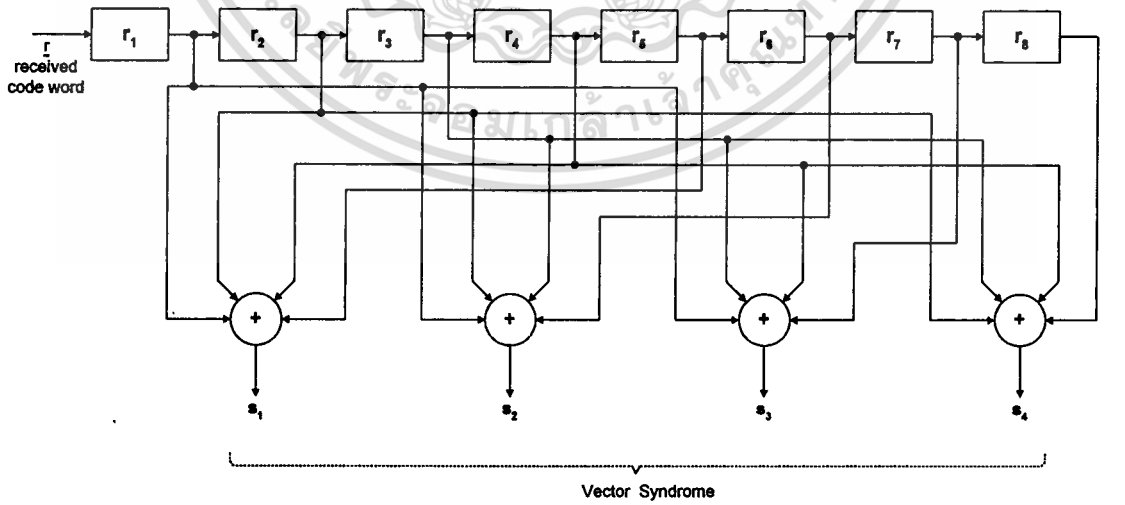
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อนำเวกเตอร์คำรหัสจากชุดเข้ารหัส $\underline{c}$ ที่มีค่าเป็น 01110001 ส่งผ่านช่องสัญญาณไปยังชุดถอดรหัสซึ่งจะรับเป็นเวกเตอร์ $\underline{r}$ ถ้าหากเวกเตอร์ $\underline{r}$ ที่รับได้มีค่าเท่ากับเวกเตอร์ $\underline{c}$ ผลจากการคำนวณของเวกเตอร์ซีนโดรมจะมีค่าเป็นศูนย์ แต่ถ้า $\underline{r} \neq \underline{c}$ นั่นคือ เกิดการผิดพลาดขึ้นจะทำให้เกิดค่าของเวกเตอร์ซีนโดรมหลังจากการคำนวณมีค่าไม่เท่ากับศูนย์ ถ้าสมมติว่าเวกเตอร์ $\underline{r}$ ที่รับเข้ามาเป็น 0111000 จะได้ค่าเวกเตอร์ซีนโดรมเป็น

$$\underline{s} = (01110000)\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

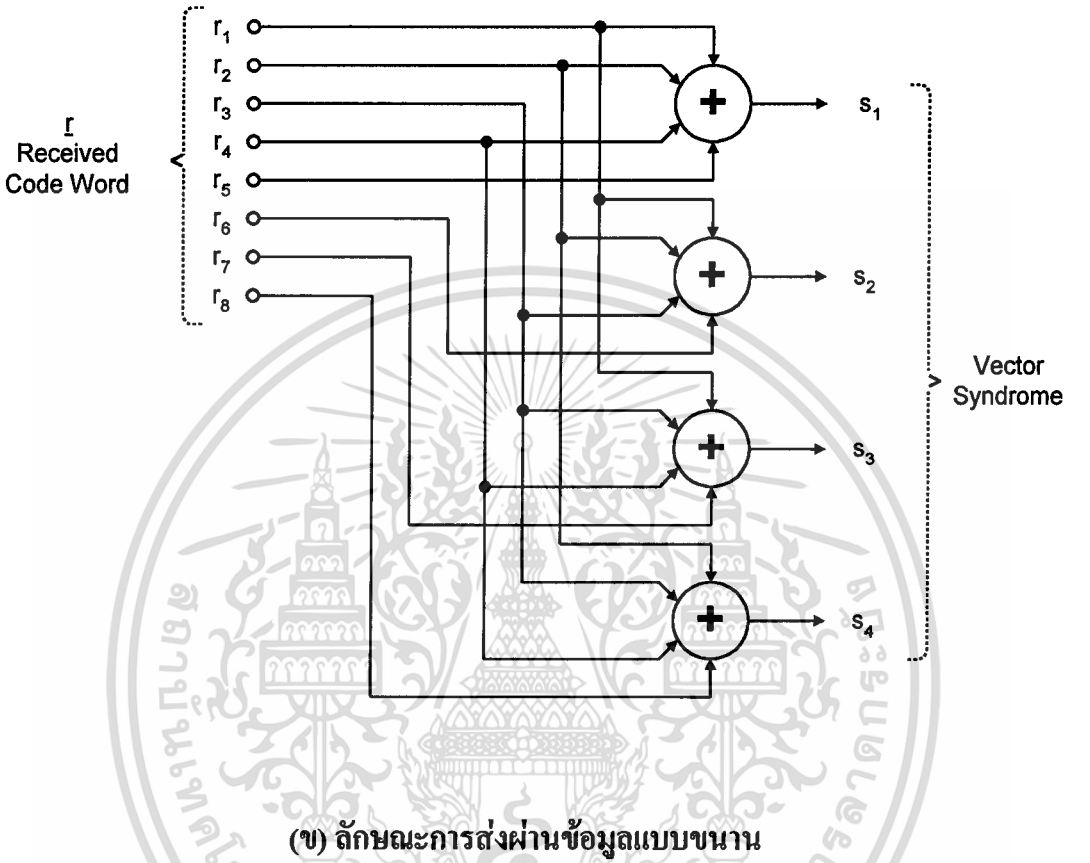
$$\underline{s} = (0001)$$

จากลักษณะการคำนวณเพื่อหาค่าของเวกเตอร์ซีนโดรมที่ผ่านมา ทำให้สามารถจำลองรูปแบบของลักษณะการคำนวณของเวกเตอร์ซีนโดรมได้ 2 แบบ ตามรูปที่ 3.3




**(ก) ลักษณะการส่งผ่านข้อมูลแบบอนุกรม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(ข) ลักษณะการส่งผ่านข้อมูลแบบขนาน

**รูปที่ 3.3** แบบจำลองชุดสร้างเวกเตอร์ซีนโดรม

จากวิธีการดังกล่าว ทำให้เราสามารถกำหนดรูปแบบของตำแหน่งบิตที่ผิด หรือเรียกว่า โคเซตลีดเดอร์ ( Coset Leader ) จำนวน 1 บิตได้ทั้งหมด 8 รูปแบบด้วยกัน และเมื่อนำแต่ละรูปแบบไปคูณกับทรานซ์โพสของเมตริกซ์ตรวจสอบซึ่งได้จากสมการ (3.24) และจะแสดงค่าของเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิดจำนวน 1 บิต ได้ดังตารางที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 1 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | 0000000x | 0001 |
| 2 | 000000x0 | 0010 |
| 3 | 00000x00 | 0100 |
| 4 | 0000x000 | 1000 |
| 5 | 000x0000 | 1011 |
| 6 | 00x00000 | 0111 |
| 7 | 0x000000 | 1101 |
| 8 | x0000000 | 1110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

**ตารางที่ 3.2 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 1 บิต**

ดังนั้นถ้าหากว่าเวกเตอร์คำรหัสที่รับเข้ามามีการผิดพลาดไป 1 บิต ไม่ว่าตำแหน่งใดๆ ก็จะสามารถแจ้งค่าเวกเตอร์ซีนโดรมได้ และถ้าหากนำเอาค่าเวกเตอร์ซีนโดรมที่ได้ไปคูณกับเมตริกซ์ตรวจสอบก็จะได้ เวกเตอร์ของรูปแบบบิตที่ผิด ซึ่งเมื่อนำไปบวกกับเวกเตอร์คำรหัสที่รับเข้ามาตามกฎการบวกแบบโมดูโล-2 ก็จะทำได้เวกเตอร์คำรหัสที่ถูกต้อง ดังสมการที่ (3.26)

$$\underline{c} = \underline{r} + \underline{e} \tag{3.26}$$

และจากค่าเวกเตอร์ซีนโดรม 0001 จะมีรูปแบบของตำแหน่งบิตที่ผิดพลาดคือ 00000001 เมื่อนำไปแก้ไขตามสมการที่ (3.26) จะได้ค่าเวกเตอร์คำรหัสที่ประมาณว่าเป็นเวกเตอร์คำรหัสที่ได้จากชุดเข้ารหัสช่องสัญญาณ

$$\underline{e} = \underline{s}H$$

$$\underline{e} = (0001)\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

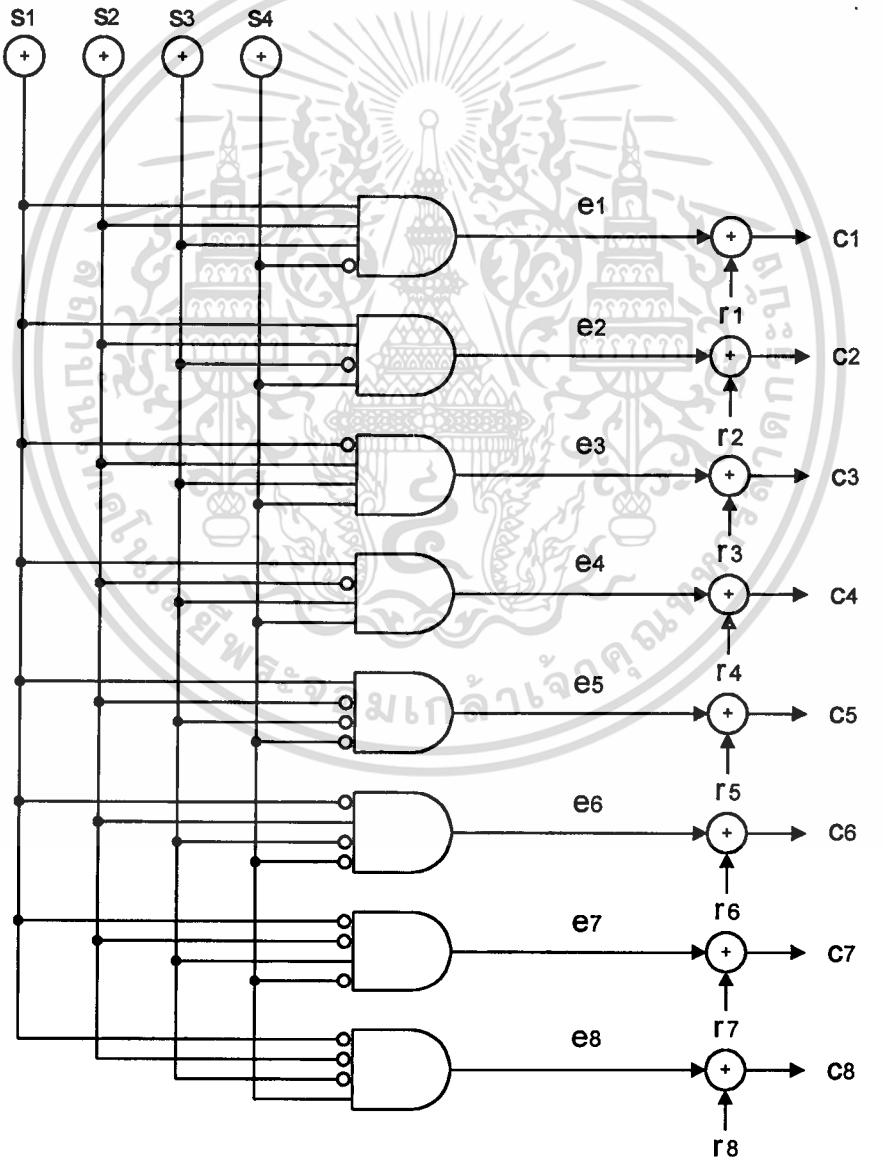
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\underline{e} = 00000001$$

$$\underline{c} = 01110000 + 00000001$$

$$\underline{c} = 01110001$$

และเมื่อนำเอากระบวนการคำนวณทางคณิตศาสตร์ดังกล่าว มาออกแบบให้เป็นแบบจำลอง เพื่อแสดงตำแหน่งที่ผิดพลาดไป 1 บิต ของเวกเตอร์คำรหัส 8 บิต รวมทั้งการแก้ไขให้เป็นเวกเตอร์คำรหัสที่ประมาณว่าถูกต้องนั้น จะเป็นดังรูปที่ 3.4




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 3.4 แบบจำลองการทำงานของชุดตรวจสอบแก้ไขรหัสที่ผิดไป 1 บิต**

สำหรับรูปแบบของตำแหน่งบิตที่ผิดจำนวน 2 บิตและ 3 บิตของเวกเตอร์คำรหัส 8 บิต นั้น ก็จะทำการกำหนดรูปแบบตำแหน่งบิตที่ผิดในแต่ละรูปแบบแล้ว คำนวณหาค่าเวกเตอร์ซีนโดรม ซึ่งจะได้ดังตารางที่ 3.3-3.4

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx000000 | 0011 |
| 2 | x0x00000 | 1001 |
| 3 | x00x0000 | 0101 |
| 4 | x000x000 | 0110 |
| 5 | x0000x00 | 1010 |
| 6 | x00000x0 | 1100 |
| 7 | x000000x | 1111 |
| 8 | 0xx00000 | 1010 |
| 9 | 0x0x0000 | 0110 |
| 10 | 0x00x000 | 0101 |
| 11 | 0x000x00 | 1001 |
| 12 | 0x0000x0 | 1111 |
| 13 | 0x00000x | 1100 |
| 14 | 00xx0000 | 1100 |
| 15 | 00x0x000 | 1111 |
| 16 | 00x00x00 | 0011 |
| 17 | 00x000x0 | 0101 |
| 18 | 00x0000x | 0110 |
| 19 | 000xx000 | 0011 |

x : ตำแหน่งบิตที่ผิดมีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.3 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 2 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 20 | 000x0x00 | 1111 |
| 21 | 000x00x0 | 1001 |
| 22 | 000x000x | 1010 |
| 23 | 0000xx00 | 1100 |
| 24 | 0000x0x0 | 1010 |
| 25 | 0000x00x | 1001 |
| 26 | 00000xx0 | 0110 |
| 27 | 00000x0x | 0101 |
| 28 | 000000xx | 0011 |

**ตารางที่ 3.3 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xxx00000 | 0100 |
| 2 | xx0x0000 | 1000 |
| 3 | xx00x000 | 1011 |
| 4 | xx000x00 | 0111 |
| 5 | xx0000x0 | 0001 |
| 6 | xx00000x | 0010 |
| 7 | x0xx0000 | 0010 |

x : ตำแหน่งบิตที่ผิดมีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.4 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 3 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 8 | x0x0x000 | 0001 |
| 9 | x0x00x00 | 1101 |
| 10 | x0x000x0 | 1011 |
| 11 | x0x0000x | 1000 |
| 12 | x00xx000 | 1101 |
| 13 | x00x0x00 | 0001 |
| 14 | x00x00x0 | 0111 |
| 15 | x00x000x | 0100 |
| 16 | x000xx00 | 0010 |
| 17 | x000x0x0 | 0100 |
| 18 | x000x00x | 0111 |
| 19 | x0000xx0 | 1000 |
| 20 | x0000x0x | 1011 |
| 21 | x00000xx | 1101 |
| 22 | 0xxx0000 | 0001 |
| 23 | 0xx0x000 | 0010 |
| 24 | 0xx00x00 | 1110 |
| 25 | 0xx000x0 | 1000 |
| 26 | 0xx0000x | 1011 |
| 27 | 0x0xx000 | 1110 |
| 28 | 0x0x0x00 | 0010 |
| 29 | 0x0x00x0 | 0100 |
| 30 | 0x0x000x | 0111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.4 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 31 | 0x00xx00 | 0001 |
| 32 | 0x00x0x0 | 0111 |
| 33 | 0x00x00x | 0100 |
| 34 | 0x000xx0 | 1011 |
| 35 | 0x000x0x | 1000 |
| 36 | 0x0000xx | 1110 |
| 37 | 00xxx000 | 0100 |
| 38 | 00xx0x00 | 1000 |
| 39 | 00xx00x0 | 1110 |
| 40 | 00xx000x | 1101 |
| 41 | 00x0xx00 | 1011 |
| 42 | 00x0x0x0 | 1101 |
| 43 | 00x0x00x | 1110 |
| 44 | 00x00xx0 | 0001 |
| 45 | 00x00x0x | 0010 |
| 46 | 00x000xx | 0100 |
| 47 | 000xxx00 | 0111 |
| 48 | 000xx0x0 | 0001 |
| 49 | 000xx00x | 0010 |
| 50 | 000x0xx0 | 1101 |
| 51 | 000x0x0x | 1110 |
| 52 | 000x00xx | 1000 |
| 53 | 0000xxx0 | 1110 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.4 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 54 | 0000xx0x | 1101 |
| 55 | 0000x0xx | 1011 |
| 56 | 00000xxx | 0111 |

**ตารางที่ 3.4 (ต่อ)**

จากเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิดจำนวน 2 บิต และ 3 บิต จะสังเกตได้ว่า ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 2 บิต จะแตกต่างไปจากรูปแบบบิตที่ผิด 1 บิตกับ 3 บิตตรงที่ค่าเวกเตอร์ซีนโดรมจะมีลอจิก1เป็นจำนวนคู่ แต่รูปแบบบิตที่ผิด 1 บิตกับ 3 บิต ค่าของเวกเตอร์ซีนโดรมจะซ้ำกัน ซึ่งค่าเวกเตอร์ซีนโดรมของทั้งรูปแบบบิตที่ผิด 2 บิตกับ 3 บิต ก็จะเกิดการซ้ำกันเองอีก และด้วยเหตุนี้ จึงทำให้รูปแบบที่มีตำแหน่งบิตที่ผิด 2 บิตกับ 3 บิต ไม่สามารถจะทำการแก้ไขให้ถูกต้องได้ และนี่ก็เป็นข้อจำกัดของรหัสระบบเชิงเส้นขนาด (8,4) บิต แต่ก็ยังให้ความสามารถของชุดถอดรหัสในการตรวจจับได้ว่า มีการผิดพลาดของเวกเตอร์คำรหัสที่รับเข้ามาจำนวน 2 บิต และ 3 บิต ซึ่งเป็นไปตามเงื่อนไขของรหัสแฮมมิ่ง ( Hamming Codes )

## 3.3 การออกแบบสร้างวงจรชุดรหัสระบบเชิงเส้นขนาด (8,4) บิต ( Circuit Design and Build for (8,4) Systematic Linear Block Codes )

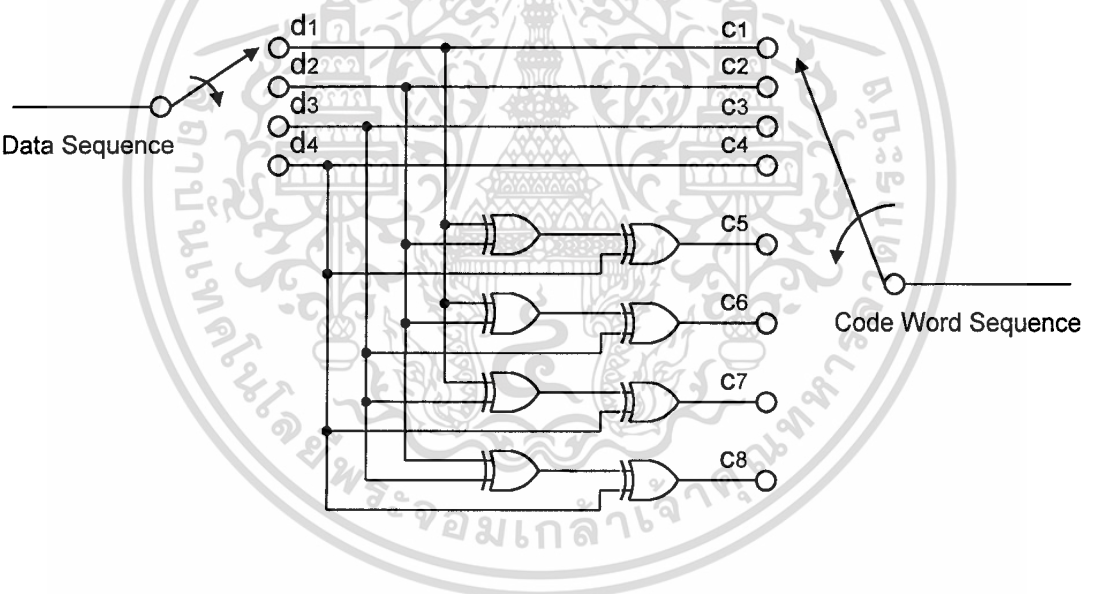
ในการออกแบบสร้างวงจรชุดรหัสระบบเชิงเส้นนั้น จะเป็นการนำเอาลักษณะการคำนวณทางคณิตศาสตร์จากวิธีการของทฤษฎีรหัสมาออกแบบ โดยการเปลี่ยนให้อยู่ในรูปของวงจรที่มีความสามารถในการกระทำได้เหมือนกับการคำนวณทางคณิตศาสตร์ และวงจรที่จะทำการออกแบบจะแบ่งออกเป็น 2 ชุดคือ

- ชุดวงจรเข้ารหัส
- ชุดวงจรถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.1 การออกแบบชุดวงจรเข้ารหัสระบบขนาด (8,4) บิต ( (8,4) Systematic Encoder Circuit Design )

จากรูปที่ 3.2 เป็นแบบจำลองการกระทำทางคณิตศาสตร์ของการเข้ารหัส ซึ่งจะเห็นได้ว่ามีอยู่ 2 ลักษณะด้วยกันคือ มีทั้งลักษณะที่เป็นการส่งผ่านข้อมูลแบบอนุกรมและแบบขนาน สำหรับในปริญญานิพนธ์นี้ จะเลือกใช้การส่งผ่านข้อมูลแบบขนาน เนื่องจากมีความสะดวกในการออกแบบและการทำงานของวงจรมีความยุ่งยากน้อยกว่า อีกทั้งยังใช้ในเวลาในการกระทำเพื่อเข้ารหัสน้อยกว่าอีกด้วย ดังนั้นจะสามารถเปลี่ยนจากแบบจำลองการกระทำทางคณิตศาสตร์ ในรูปที่ 3.2 (ข) โดยที่แทนการรวมกันของข้อมูลด้วยการบวกแบบโมดูโล-2 ตามทฤษฎีของ GF(2) จะได้เป็นชุดวงจรเข้ารหัสระบบเชิงเส้นดังนี้




**รูปที่ 3.5 ชุดวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

เมื่อพิจารณาจากรูปที่ 3.5 แล้ว จะเห็นว่าในส่วนการป้อนข้อมูลเข้า ( Data Input ) จะมีการป้อนแบบอนุกรมหรือแบบขนานก็ได้ ถ้าหากป้อนข้อมูลเข้าแบบอนุกรมก็จะต้องทำการแปลงข้อมูลให้อยู่ในแบบขนานเสียก่อนแล้วจึงส่งให้กับชุดเข้ารหัส โดยชุดเข้ารหัสจะนำข้อมูลจำนวน 4 บิต มาสร้างให้เป็นเวกเตอร์คำรหัสที่มีขนาด 8 บิต จะสังเกตได้ว่าใน 4 บิตแรกของเวกเตอร์คำรหัสคือ $c_1, c_2, c_3, c_4$ ก็จะเป็นข้อมูลที่ป้อนเข้ามา ส่วน 4 บิตหลังคือ $c_5, c_6, c_7, c_8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

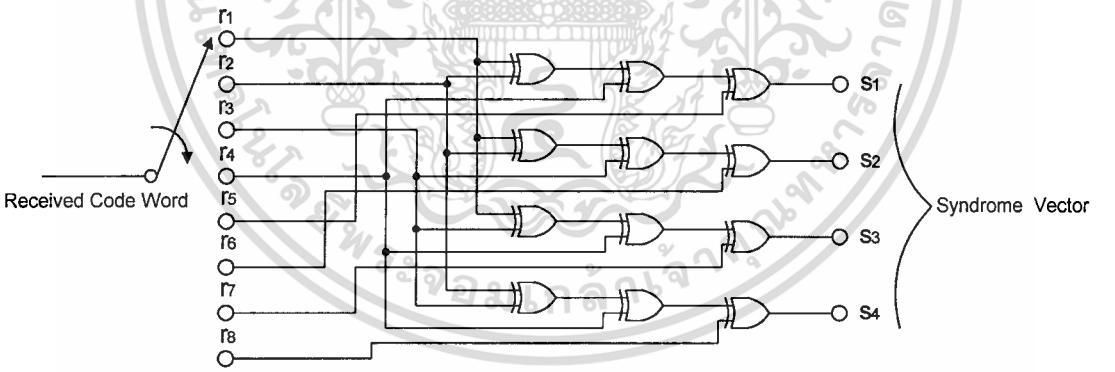
จะเป็นบิตตรวจสอบ ซึ่งได้จากข้อมูลที่ป้อนเข้ามาแล้วถูกนำมากระทำตามกระบวนการคำนวณ และออกแบบดังที่ได้กล่าวมาแล้ว และเวกเตอร์คำรหัสที่ได้ทั้ง 8 บิตนี้ ก็จะถูกส่งออกในรูปแบบขนาน แต่ถ้าต้องการให้มีการส่งผ่านข้อมูลแบบอนุกรมก็ต้องแปลงให้ข้อมูลมีการจัดเรียงแบบอนุกรมเสียก่อน

### 3.3.2 การออกแบบชุดวงจรถอดรหัสระบบขนาด (8,4) บิต ( (8,4) Systematic Decoder Circuit Design )

ในส่วนของชุดวงจรถอดรหัสนั้นก็จะอาศัยแบบจำลองการกระทำทางคณิตศาสตร์ของการถอดรหัสเช่นเดียวกับชุดวงจรเข้ารหัส แต่เนื่องจากความสามารถของชุดถอดรหัสนั้น มีอยู่ 2 ประการ คือ การตรวจจับและการแก้ไขความผิดพลาดของเวกเตอร์คำรหัสที่รับเข้ามา ฉะนั้นในการออกแบบวงจรชุดถอดรหัสจะแบ่งออกเป็น 3 ส่วน คือ

**1. วงจรสร้างเวกเตอร์ซีนโดรม ( Vector Syndrome Generate Circuit )**

ในส่วนนี้จะทำการออกแบบตามแบบจำลองการสร้างเวกเตอร์ซีนโดรมในรูปที่ 3.3(ข) ซึ่งเป็นแบบที่มีการส่งผ่านข้อมูลแบบขนานและวงจรที่ได้จากการออกแบบ แสดงดังรูปที่ 3.6




**รูปที่ 3.6 วงจรสร้างเวกเตอร์ซีนโดรม**

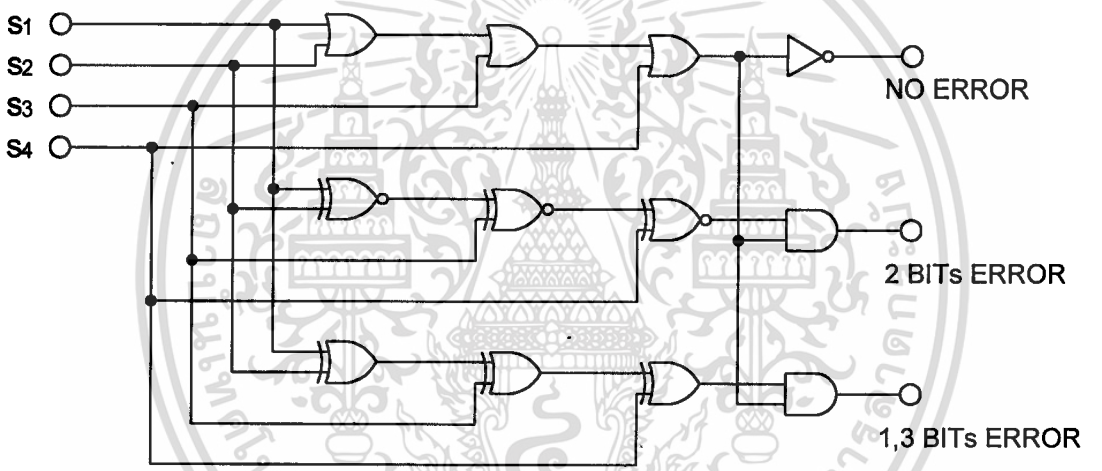
จากรูปที่ 3.6 จะเห็นได้ว่าเวกเตอร์คำรหัสที่รับเข้ามาถ้ามีการส่งผ่านข้อมูลแบบอนุกรมก็จะต้องแปลง เพื่อให้มีการส่งป้อนข้อมูลแบบขนานเสียก่อน ซึ่งเวกเตอร์คำรหัสที่ได้จะเป็น $r_1, r_2, \ldots, r_8$ โดยผลจากการทำงานของวงจรจะได้จากการนำข้อมูลในแต่ละบิตตามที่กำหนดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการรวมกันแบบโมดูโล-2 และผลจากการกระทำจะได้เวกเตอร์ซีนโดรมที่มี 4 บิต และเวกเตอร์ซีนโดรมนี้ จะนำไปใช้ในการออกแบบวงจรในส่วนของการตรวจจับและการแก้ไขความผิดพลาดของเวกเตอร์คำรหัสต่อไป

**2. วงจรตรวจจับความผิดพลาด ( Detection Error Circuit )**

ในส่วนของวงจรตรวจจับนี้ เป็นการนำเวกเตอร์ซีนโดรมที่ได้มาประยุกต์ออกแบบเพื่อใช้ตรวจจับความผิดพลาดของเวกเตอร์คำรหัส โดยจะทำการตรวจจับจำนวนบิตที่ผิดพลาดในแต่ละเวกเตอร์คำรหัสแล้วก็แสดงผล ซึ่งจะแบ่งเป็น 3 ชุด ดังรูปที่ 3.7




**รูปที่ 3.7 วงจรตรวจจับความผิดพลาด**

จากวงจรในรูปที่ 3.7 จะสามารถนำเวกเตอร์ซีนโดรมมาใช้ในการตรวจจับเวกเตอร์คำรหัสที่มีบิตผิด โดยในชุดแรกจะทำการตรวจจับเวกเตอร์คำรหัสที่ไม่มีบิตผิด ถ้าเวกเตอร์คำรหัสไม่มีบิตผิดค่าเอาต์พุตจุดนี้ ( No Error ) ที่ได้จากค่าเวกเตอร์ซีนโดรมจะแสดงสถานะเป็นลอจิก 1 แต่ถ้าเวกเตอร์คำรหัสที่รับเข้ามามีบิตผิด ค่าเอาต์พุตจะให้ลอจิก 0 ในชุดที่สองจะเป็นการตรวจจับคำรหัสที่มีบิตผิด 2 บิต ซึ่งจะนำความแตกต่างของเวกเตอร์ซีนโดรมมาใช้ในการออกแบบ ถ้าหากเวกเตอร์คำรหัสที่รับเข้ามามีบิตผิด 2 บิต ค่าเอาต์พุตที่จุดนี้ ( 2 Bits Error ) จะแสดงเป็นลอจิก 1 แต่ถ้าเป็นกรณีอื่นๆ ค่าเอาต์พุตจะแสดงสถานะเป็นลอจิก 0 สำหรับใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดที่สาม จะเป็นชุดที่ตรวจจับเวกเตอร์คำรหัสที่มีบิตผิด 1 บิตและ 3 บิต ด้วยเหตุที่ค่าเวกเตอร์ซีนโดรมของเวกเตอร์คำรหัสที่มีบิตผิด 1 บิตและ 3 บิตนั้น จะมีการซ้ำกันอยู่ ดังนั้นเวกเตอร์คำรหัสที่มีบิตผิด 1 บิตหรือ 3 บิต ค่าเอาต์พุตที่ได้ที่จุดนี้ ( 1,3 Bits Error ) จะเป็นลอจิก 1 ถ้าเป็นกรณีอื่นๆ เอาต์พุตจะเป็นลอจิก 0

**3. วงจรแก้ไขความผิดพลาด ( Correction Error Circuit )**

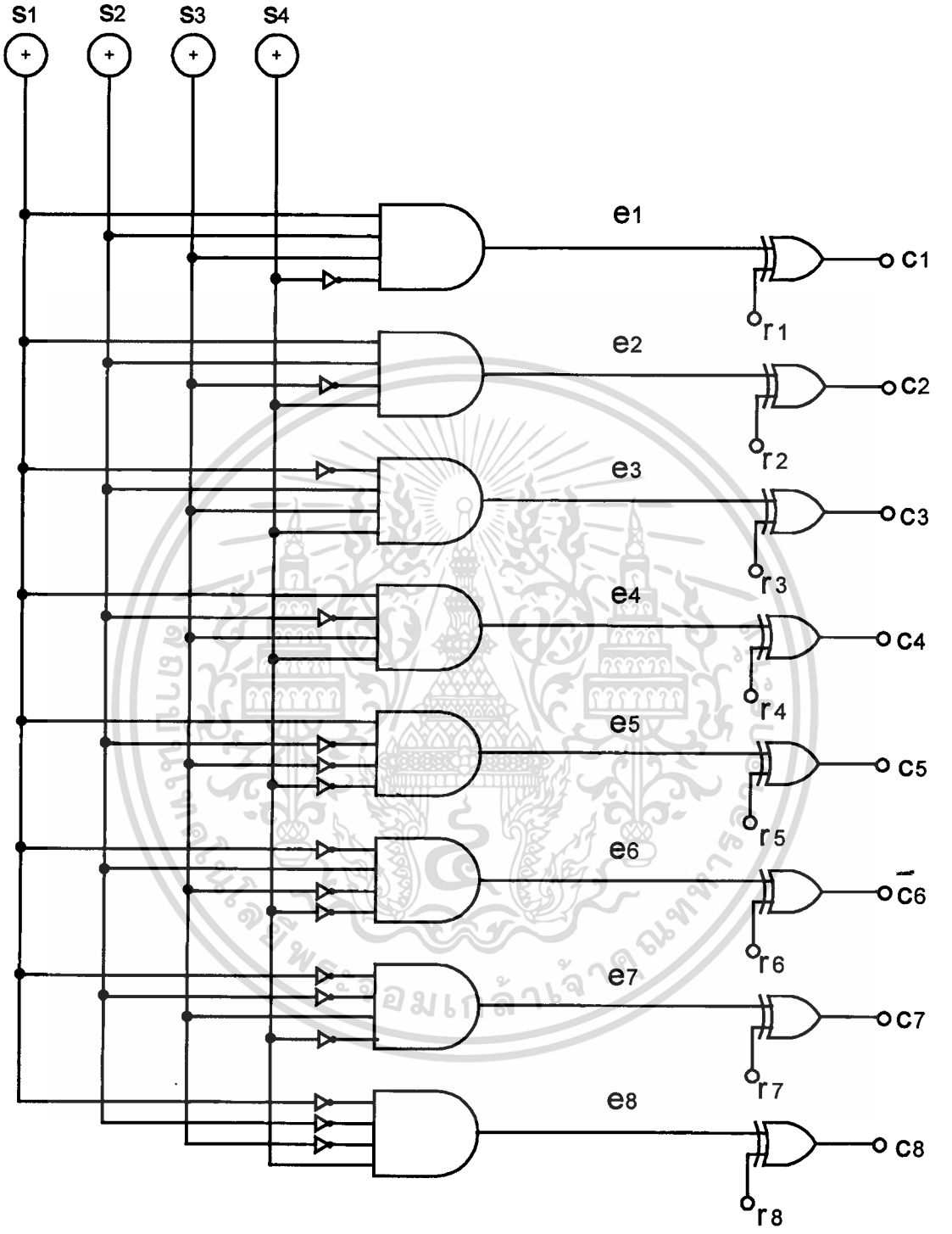
สำหรับวงจรแก้ไขความผิดพลาดของเวกเตอร์คำรหัสของปริญญานิพนธ์นี้ จะสามารถทำการแก้ไขความผิดพลาดของเวกเตอร์คำรหัสได้ 1 บิต ซึ่งจะเป็นไปตามเงื่อนไขของแฮมมิ่ง ดังนั้นจากแบบจำลองการกระทำทางคณิตศาสตร์ตามรูปที่ 3.4 จะสามารถออกแบบวงจรได้ดังรูปที่ 3.8

จากวงจรแก้ไขความผิดพลาดของเวกเตอร์คำรหัสจะเป็นการนำเอาค่าเวกเตอร์ซีนโดรมที่ได้จากวงจรสร้างเวกเตอร์ซีนโดรมมาทำการแก้ไขความผิดพลาดของเวกเตอร์คำรหัสขนาด 1 บิต โดยนำมาแสดงรูปแบบตำแหน่งบิตที่ผิด ซึ่งเมื่อนำไปเปรียบเทียบกับเวกเตอร์คำรหัสที่รับเข้ามาจะได้ผลจากการเปรียบเทียบทางเอาต์พุตของเอ็กคลูซีฟออร์เกตที่ประมาณว่าเป็นเวกเตอร์คำรหัสที่ได้แก้ไขให้ถูกต้องแล้ว

## 3.4 การปรับปรุงและพัฒนาชุดรหัสระบบเชิงเส้น

ในหัวข้อนี้จะได้กล่าวถึงการนำวงจรในแต่ละส่วนที่ได้ทำการออกแบบไปแล้วมาประกอบเป็นวงจรชุดเข้ารหัส และวงจรชุดถอดรหัส โดยจะทำการปรับปรุงวงจรเพื่อให้สามารถนำไปใช้ได้อย่างมีประสิทธิภาพมากขึ้น นอกจากนี้ยังจะได้กล่าวถึงการพัฒนาวิธีการเข้ารหัสและถอดรหัสในลักษณะของการเพิ่มจำนวนบิตตรวจสอบกับการเพิ่มจำนวนบิตข้อมูล ทั้งนี้ก็เพื่อที่จะเพิ่มประสิทธิภาพในการตรวจจับ และแก้ไขความผิดพลาดของข้อมูลให้ได้มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

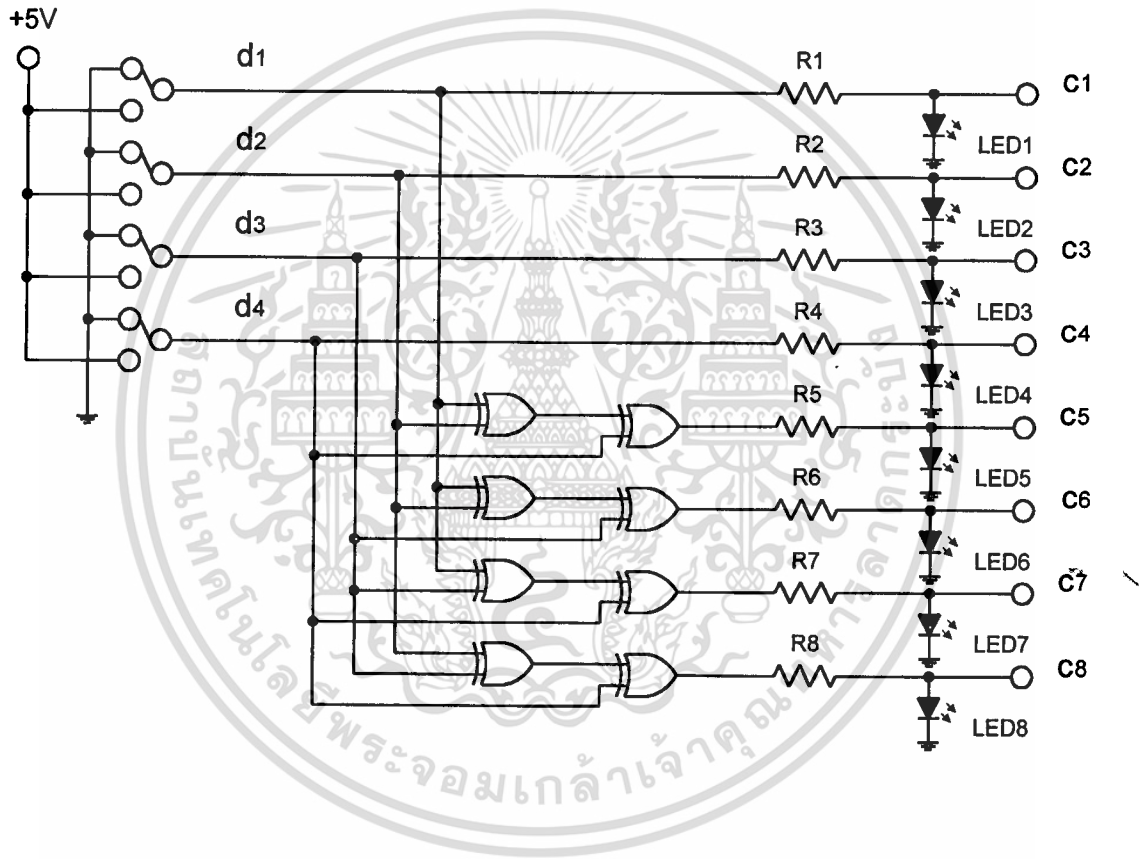



**รูปที่ 3.8 วงจรแก้ไขความผิดพลาด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

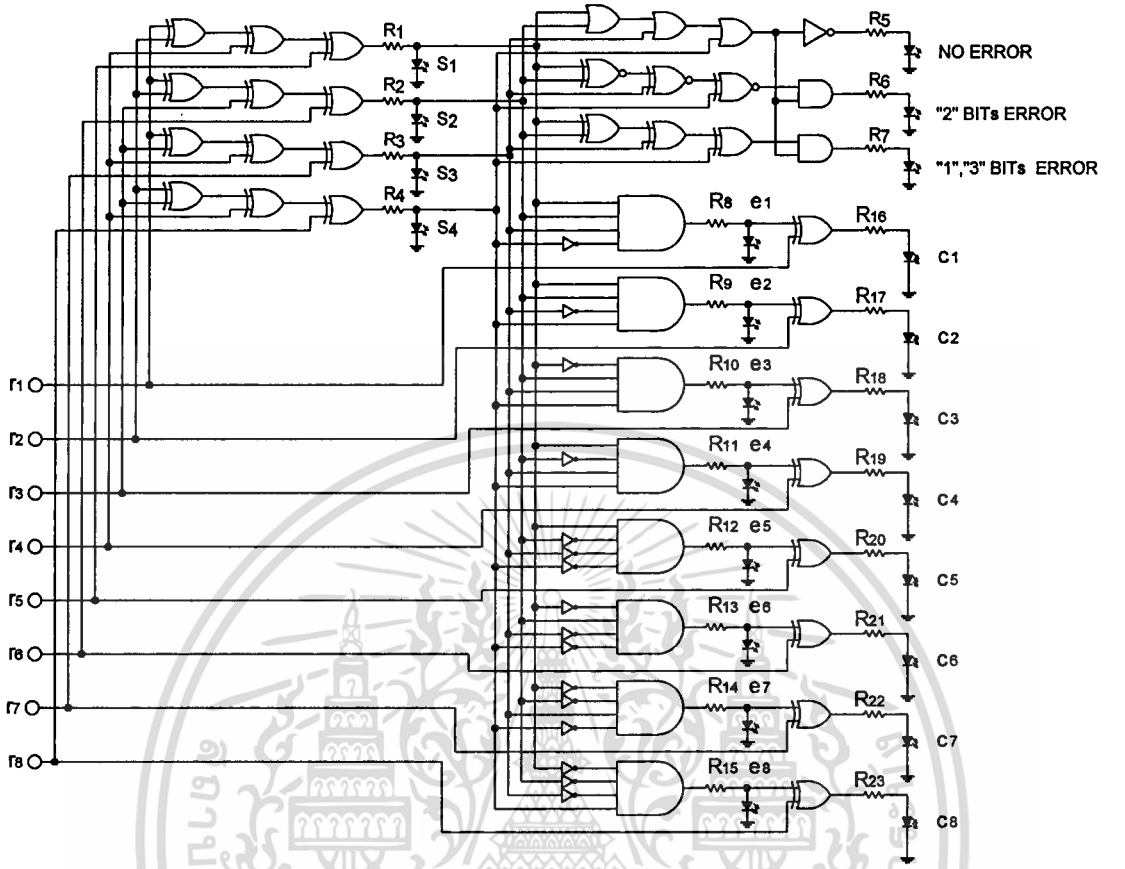
### 3.4.1 การปรับปรุงวงจรชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต

สำหรับในส่วนนี้จะได้นำเอาวงจรที่ออกแบบในหัวข้อ 3.3 มาทำการปรับปรุงเพื่อให้สามารถนำไปใช้ได้อย่างสมบูรณ์มากยิ่งขึ้น โดยจะแบ่งวงจรออกเป็น 2 ภาค คือ ภาคเข้ารหัสกับภาคถอดรหัส ซึ่งจะได้แสดงวงจรที่ปรับปรุงแล้วดังนี้




**รูปที่ 3.9 วงจรชุดเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.10 วงจรชุดถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต**

### 3.4.2 การพัฒนารหัสระบบเชิงเส้นด้วยการเพิ่มจำนวนบิตตรวจสอบ

เนื่องจากรหัสระบบเชิงเส้นขนาด (8,4) บิต มีข้อจำกัดในการแก้ไขบิตผิดพลาดได้เพียงบิตเดียว และสามารถตรวจจับบิตผิดพลาดได้เพียง 2 บิต ซึ่งถ้ามีบิตผิดพลาด 1 บิตหรือ 3 บิต จะแยกการตรวจจับไม่ได้ และข้อจำกัดเหล่านี้มีสาเหตุมาจากรูปแบบของเวกเตอร์ซีนโดรมที่มีเพียง15 รูปแบบ จึงทำให้เกิดการซ้ำกันระหว่างเวกเตอร์คำรหัสที่มีบิตผิดพลาด 1 บิตและ 3 บิต นอกจากนี้ เวกเตอร์ซีนโดรมของเวกเตอร์คำรหัสที่มีบิตผิดพลาด 2 บิต ก็จะเกิดการซ้ำกันเอง ทำให้ไม่สามารถแก้ไขเวกเตอร์คำรหัสที่มีบิตผิดพลาดจำนวน 2 บิตได้ ด้วยเหตุนี้จึงทำให้ต้องมีการพัฒนาเพื่อเพิ่มประสิทธิภาพในการแก้ไขความผิดพลาดของเวกเตอร์คำรหัสให้มากขึ้น และในขั้นตอนการพัฒนาได้อาศัยแนวความคิดที่จะเพิ่มจำนวนบิตแก่คำรหัสจาก 8 บิตเป็น 12 บิต โดยกำหนดให้จำนวนบิตที่เพิ่มขึ้นเป็นจำนวนบิตตรวจสอบโดยเพิ่มจาก 4 บิต เป็น 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งก็จะได้เวกเตอร์คำรหัสที่มีจำนวนบิตข้อมูล 4 บิต จำนวนบิตตรวจสอบ 8 บิต จัดเป็นชุดรหัสระบบเชิงเส้นขนาด (12,4) บิต และผลจากการเพิ่มจำนวนบิตตรวจสอบเข้าไป จะทำให้รูปแบบของเวกเตอร์ซีนโดรมมีจำนวนมากนั้น ก็หมายความว่า สามารถแก้ไขเวกเตอร์คำรหัสที่มีตำแหน่งบิตที่ผิดมากกว่า 1 บิตได้

หลักการออกแบบนั้นจะอาศัยหลักการเดิมตามหัวข้อที่ 3.2 เพียงแต่วิธีในการออกแบบนั้นจะทำการกำหนดข้อมูลที่จะทำการเข้ารหัสให้มี 6 บิต โดยที่ 4 บิตแรกจะเป็นชุดข้อมูลที่ต้องการ ส่วน 2 บิตหลัง คือ บิตที่ 5 และบิตที่ 6 จะเป็นบิตที่ถูกสร้างขึ้นและฝากไปกับข้อมูล 4 บิต ซึ่งจะกำหนดให้เป็นลอจิก 1 ตลอด ทั้งนี้ก็จะได้จำนวนรูปแบบของข้อมูล 16 รูปแบบเท่าเดิมและเมตริกซ์ตัวกำเนิดในที่นี้กำหนดได้ดังสมการที่ (3.27)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \quad (3.27)$$

และ $\underline{d} = d_1, d_2, d_3, d_4, d_5, d_6 \; ; \; d_5, d_6 = 1 \quad (3.28)$

และจากสมการเข้ารหัส $\underline{c} = \underline{d}\,G$ จะได้ค่าแต่ละบิตของเวกเตอร์คำรหัสเป็น

$$c_1 = c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12} \quad (3.29)$$

โดยที่

$c_1 = d_1$

$c_2 = d_2$

$c_3 = d_3$

$c_4 = d_4$

$c_5 = d_5$

$c_6 = d_6$

$c_7 = d_1+d_3+d_4+d_5$

$c_8 = d_2+d_3+d_5+d_6$

$c_9 = d_1+d_3+d_4+d_6$

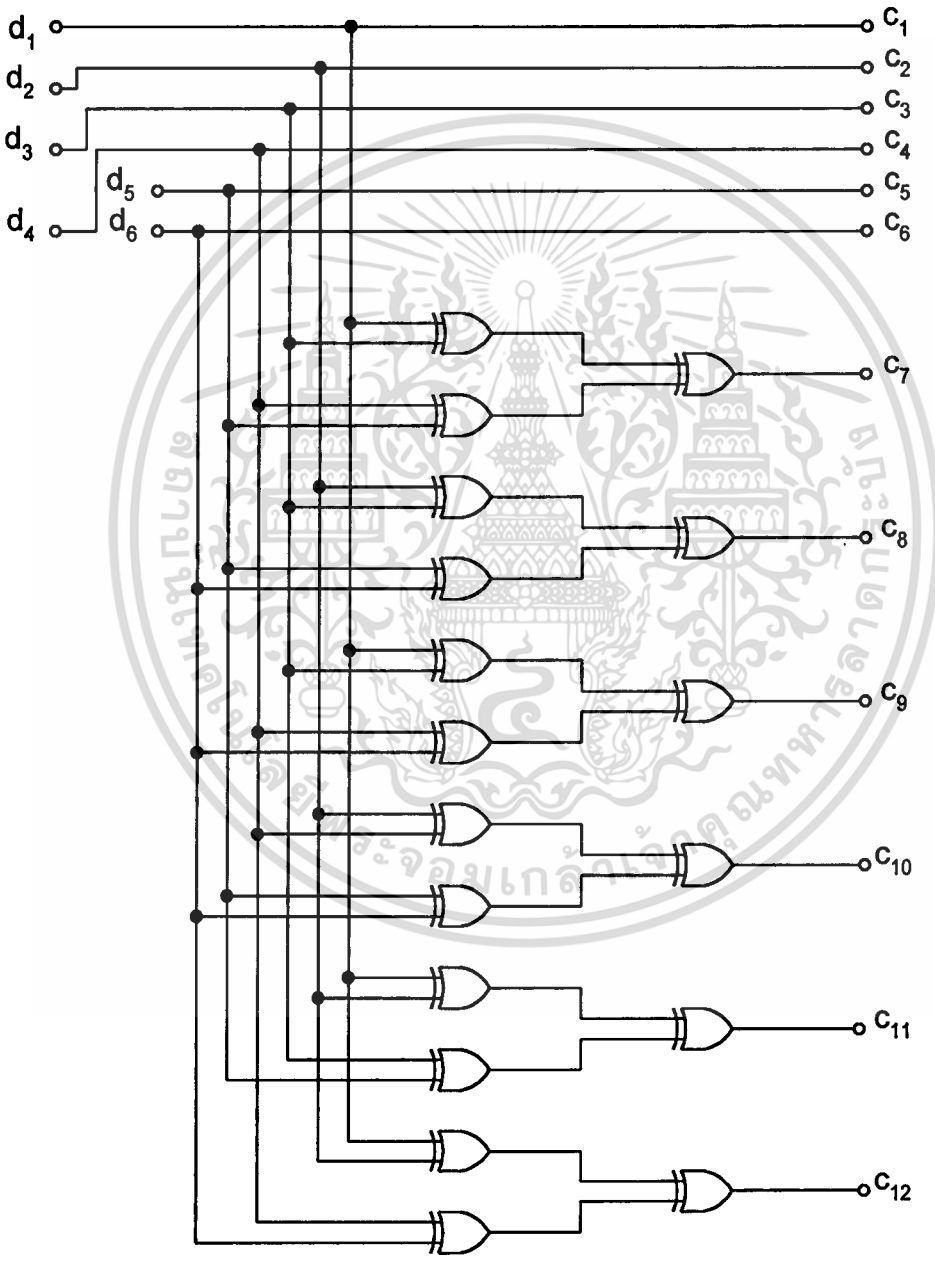
$c_{10} = d_2+d_4+d_5+d_6$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$c_{11} = d_1+d_2+d_3+d_5$$

$$c_{12} = d_1+d_2+d_4+d_6$$

ดังนั้นวงจรเข้ารหัสระบบเชิงเส้นขนาด (12,4) บิต จะสามารถออกแบบได้ดังรูปที่ 3.11




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

**รูปที่ 3.11 วงจรเข้ารหัสระบบเชิงเส้นขนาด (12,4) บิต**

จากวงจรเข้ารหัสที่ได้ในรูปที่ 3.11 จะสร้างเวกเตอร์คำรหัสจากข้อมูลทั้ง 16 รูปแบบ ได้ดังตารางที่ 3.5

| ลำดับที่ | ลำดับข้อมูล | ลำดับคำรหัส |
|---|---|---|
| 1 | 000011 | 000011101011 |
| 2 | 000111 | 000111000110 |
| 3 | 001011 | 001011010001 |
| 4 | 001111 | 001111111100 |
| 5 | 010011 | 010011111100 |
| 6 | 010111 | 010111010001 |
| 7 | 011011 | 011011000110 |
| 8 | 011111 | 011111101011 |
| 9 | 100011 | 100011000000 |
| 10 | 100111 | 100111101101 |
| 11 | 101011 | 101011111010 |
| 12 | 101111 | 101111010111 |
| 13 | 110011 | 110011010111 |
| 14 | 110111 | 110111111010 |
| 15 | 111011 | 111011101101 |
| 16 | 111111 | 111111000000 |

**ตารางที่ 3.5 รูปแบบเวกเตอร์คำรหัสของชุดเข้ารหัสระบบเชิงเส้นขนาด (12,4) บิต**

จากตารางที่ 3.5 จะสังเกตได้ว่า เวกเตอร์คำรหัสที่ได้ทั้ง 16 รูปแบบ จะมีตำแหน่งบิตที่ 5 และ 6 มีค่าเป็นลอจิก 1 ซึ่งเปรียบเสมือนกับการฝากบิตที่ถูกต้องเข้าไปกับทุกๆ เวกเตอร์คำรหัสที่ชุดเข้ารหัส ทั้งนี้ก็เพื่อจะเป็นการเพิ่มรูปแบบของเวกเตอร์ซินโดรมที่ชุดถอดรหัสให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีจำนวนมากขึ้นนั่นเอง ซึ่งจะทำให้ชุดถอดรหัสมีความสามารถในการแก้ไขรูปแบบตำแหน่งบิตที่ผิดของเวกเตอร์คำรหัสได้มากขึ้นด้วย

สำหรับในชุดถอดรหัสนั้นจำเป็นจะต้องออกแบบวงจรกำเนิดเวกเตอร์ซีนโดรมก่อน ซึ่งก็จะอาศัยหลักการของการถอดรหัสระบบเชิงเส้นเช่นกัน ดังนั้นเมตริกซ์ตรวจสอบที่ได้ก็คือ

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.30)$$

และทรานซ์โพสของเมตริกซ์ตรวจสอบคือ

$$H^T = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.31)$$

เนื่องจากเวกเตอร์คำรหัสที่รับเข้ามาคือ

$$\underline{r} = r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12} \quad (3.32)$$

ฉะนั้นจากสมการ การคำนวณหาค่าเวกเตอร์ซีนโดรมคือ $\underline{s} = \underline{r}H^T$ จะทำให้ได้ค่าแต่ละบิตของเวกเตอร์ซีนโดรม ดังนี้

$$\underline{s} = s_1, s_2, s_3, s_4, s_5, s_6$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $s_1 = r_1+r_3+r_4+r_5+r_7$

$s_2 = r_2+r_3+r_5+r_6+r_8$

$s_3 = r_1+r_3+r_4+r_6+r_9$

$s_4 = r_2+r_4+r_5+r_6+r_{10}$

$s_5 = r_1+r_2+r_3+r_5+r_{11}$

$s_6 = r_1+r_2+r_4+r_6+r_{12}$

จากการฝากบิตที่ถูกต้องมากับเวกเตอร์คำรหัสทางชุดเข้ารหัสที่ตำแหน่งบิตที่ 5 และบิตที่ 6 โดยให้มีค่าเป็นลอจิก 1 นั้น เมื่อมาถึงชุดถอดรหัสที่ภาคกำเนิดเวกเตอร์ซีนโดรมก็จะต้องกำหนดให้บิตที่ 5 และบิตที่ 6 ของเวกเตอร์คำรหัสที่รับเข้ามาให้มีลอจิกเป็น 1 ด้วย ซึ่งจะแสดงค่าของเวกเตอร์ซีนโดรมในแต่ละบิตได้ตามสมการ (3.34)

$s_1 = r_1+r_3+r_4+r_5+r_7$

$s_2 = r_2+r_3+r_8$

$s_3 = r_1+r_3+r_4+r_6+r_9$

$s_4 = r_2+r_4+r_{10}$

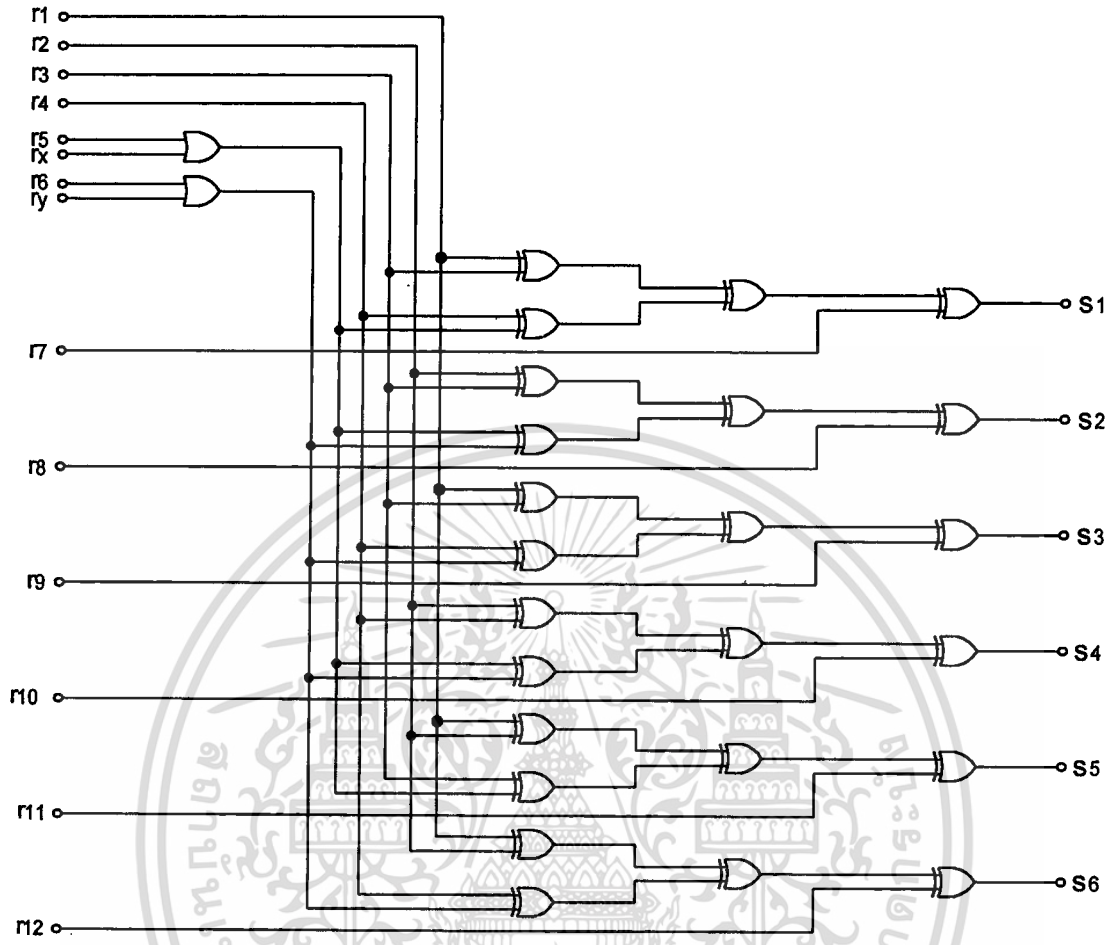
$s_5 = r_1+r_2+r_3+r_5+r_{11}$

$s_6 = r_1+r_2+r_4+r_6+r_{12}$

$\therefore s = (r_1+r_3+r_4+r_5+r_7,\ r_2+r_3+r_8,\ r_1+r_3+r_4+r_6+r_9,\ r_2+r_4+r_{10},\ r_1+r_2+r_3+r_5+r_{11},\ r_1+r_2+r_4+r_6+r_{12})$ (3.34)

ดังนั้นวงจรกำเนิดเวกเตอร์ซีนโดรมจะออกแบบได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.12 วงจรกำเนิดเวกเตอร์ซีนโดรมของชุดถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต**

จากรูปที่ 3.12 จะเห็นได้ว่าขา $r_x$ และ $r_y$ จะถูกกำหนดให้มีค่าเป็นลอจิก 1 เสมอ เพื่อที่จะกำหนดค่า $r_5$ กับ $r_6$ ให้เป็นลอจิก 1 ตลอดเวลา ส่วนค่าเอาต์พุตที่ได้จะเป็นเวกเตอร์ซีนโดรมที่นำไปใช้แก้ไขความผิดพลาดของเวกเตอร์คำรหัสต่อไป

ในการออกแบบส่วนของวงจรแก้ไขความผิดพลาดของเวกเตอร์คำรหัสนั้นจำเป็นจะต้องกำหนดรูปแบบบิตที่ผิดของเวกเตอร์คำรหัส เพื่อจะได้นำไปคำนวณหาค่าของเวกเตอร์ซีนโดรม แล้วจึงนำเวกเตอร์ซีนโดรมที่ได้ไปทำการแก้ไขในตำแหน่งบิตที่ผิดของเวกเตอร์คำรหัสที่รับเข้ามาให้เป็นเวกเตอร์คำรหัสที่ถูกต้องต่อไป ฉะนั้นรูปแบบของเวกเตอร์ซีนโดรมที่ใช้ในการแก้ไขความผิดพลาดของเวกเตอร์คำรหัส เมื่อเทียบกับรูปแบบตำแหน่งบิตที่ผิดของเวกเตอร์คำรหัสจะได้ตามตารางที่ 3.6, 3.7, 3.8, 3.9 และ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 1 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | x00000000000 | 101011 |
| 2 | 0x0000000000 | 010111 |
| 3 | 00x000000000 | 111010 |
| 4 | 000x00000000 | 101101 |
| 5 | 0000x0000000 | 000000 |
| 6 | 00000x000000 | 000000 |
| 7 | 000000x00000 | 100000 |
| 8 | 0000000x0000 | 010000 |
| 9 | 00000000x000 | 001000 |
| 10 | 000000000x00 | 000100 |
| 11 | 0000000000x0 | 000010 |
| 12 | 00000000000x | 000001 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

**ตารางที่ 3.6 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 1 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx0000000000 | 111100 |
| 2 | x0x000000000 | 010001 |
| 3 | x00x00000000 | 000110 |
| 4 | x000x0000000 | 101011 |
| 5 | x0000x000000 | 101011 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ ... ทั้งที่มีการนำไปใช้

**ตารางที่ 3.7 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 2 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 6 | x00000x00000 | 001011 |
| 7 | x000000x0000 | 111011 |
| 8 | x0000000x000 | 100011 |
| 9 | x00000000x00 | 101111 |
| 10 | x000000000x0 | 101001 |
| 11 | x0000000000x | 101010 |
| 12 | 0x00x0000000 | 010111 |
| 13 | 0x000x000000 | 010111 |
| 14 | 0x0000x00000 | 110111 |
| 15 | 0x00000x0000 | 000111 |
| 16 | 0x000000x000 | 011111 |
| 17 | 0x0000000x00 | 010011 |
| 18 | 0x00000000x0 | 010101 |
| 19 | 0x000000000x | 010110 |
| 20 | 00x0x0000000 | 111010 |
| 21 | 00x00x000000 | 111010 |
| 22 | 00x000x00000 | 011010 |
| 23 | 00x00000x000 | 110010 |
| 24 | 00x000000x00 | 111110 |
| 25 | 00x0000000x0 | 111000 |
| 26 | 000xx0000000 | 101101 |
| 27 | 000x0x000000 | 101101 |
| 28 | 000x00x00000 | 001101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.7 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 29 | 000x000x0000 | 111101 |
| 30 | 000x0000x000 | 100101 |
| 31 | 000x0000000x | 101100 |
| 32 | 0000xx000000 | 000000 |
| 33 | 0000x0x00000 | 100000 |
| 34 | 0000x00x0000 | 010000 |
| 35 | 0000x000x000 | 001000 |
| 36 | 0000x0000x00 | 000100 |
| 37 | 0000x00000x0 | 000010 |
| 38 | 0000x000000x | 000001 |
| 39 | 00000xx00000 | 100000 |
| 40 | 00000x0x0000 | 010000 |
| 41 | 00000x00x000 | 001000 |
| 42 | 00000x000x00 | 000100 |
| 43 | 00000x0000x0 | 000010 |
| 44 | 00000x00000x | 000001 |
| 45 | 000000xx0000 | 110000 |
| 46 | 000000x0x000 | 101000 |
| 47 | 000000x00x00 | 100100 |
| 48 | 000000x000x0 | 100010 |
| 49 | 000000x0000x | 100001 |
| 50 | 0000000xx000 | 011000 |
| 51 | 0000000x0x00 | 010100 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.7 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 52 | 0000000x00x0 | 010010 |
| 53 | 00000000xx00 | 001100 |
| 54 | 00000000x0x0 | 001010 |
| 55 | 00000000x00x | 001001 |
| 56 | 000000000x0x | 000101 |
| 57 | 0000000000xx | 000011 |

**ตารางที่ 3.7 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx00x0000000 | 111100 |
| 2 | xx000x000000 | 111100 |
| 3 | xx0000x00000 | 011100 |
| 4 | xx000000x000 | 110100 |
| 5 | x0x0x0000000 | 010001 |
| 6 | x0x00x000000 | 010001 |
| 7 | x0x000x00000 | 110001 |
| 8 | x0x00000x000 | 011001 |
| 9 | x00xx0000000 | 000110 |
| 10 | x00x0x000000 | 000110 |
| 11 | x00x00x00000 | 100110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

**ตารางที่ 3.8 ค่าเวกเตอร์ซีโดรมของรูปแบบตำแหน่งบิตที่ผิด 3 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 12 | x00x0000x000 | 001110 |
| 13 | x000xx000000 | 101011 |
| 14 | x000x0x00000 | 001011 |
| 15 | x000x00x0000 | 111011 |
| 16 | x000x000x000 | 100011 |
| 17 | x000x0000x00 | 101111 |
| 18 | x000x00000x0 | 101001 |
| 19 | x000x000000x | 101010 |
| 20 | x0000xx00000 | 001011 |
| 21 | x0000x0x0000 | 111011 |
| 22 | x0000x00x000 | 100011 |
| 23 | x0000x000x00 | 101111 |
| 24 | x0000x0000x0 | 101001 |
| 25 | x0000x00000x | 101010 |
| 26 | x00000xx0000 | 011011 |
| 27 | x000000x0x00 | 001111 |
| 28 | x000000xx000 | 110011 |
| 29 | x000000x0x00 | 111111 |
| 30 | x000000x00x0 | 111001 |
| 31 | x0000000xx00 | 100111 |
| 32 | 0x00xx000000 | 010111 |
| 33 | 0x00x0x00000 | 110111 |
| 34 | 0x00x00x0000 | 000111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 35 | 0x00x000x000 | 011111 |
| 36 | 0x00x0000x00 | 010011 |
| 37 | 0x00x00000x0 | 010101 |
| 38 | 0x00x000000x | 010110 |
| 39 | 0x000xx00000 | 110111 |
| 40 | 0x000x0x0000 | 000111 |
| 41 | 0x000x00x000 | 011111 |
| 42 | 0x000x000x00 | 010011 |
| 43 | 0x000x0000x0 | 010101 |
| 44 | 0x000x00000x | 010110 |
| 45 | 0x0000x000x0 | 110101 |
| 46 | 0x0000x0000x | 110110 |
| 47 | 0x000000x0x0 | 011101 |
| 48 | 0x000000x00x | 011110 |
| 49 | 00x0xx000000 | 111010 |
| 50 | 00x0x0x00000 | 011010 |
| 51 | 00x0x000x000 | 110010 |
| 52 | 00x0x0000x00 | 111110 |
| 53 | 00x0x00000x0 | 111000 |
| 54 | 00x00xx00000 | 011010 |
| 55 | 00x00x00x000 | 110010 |
| 56 | 00x00x000x00 | 111110 |
| 57 | 00x00x0000x0 | 111000 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 58 | 000xxx000000 | 101101 |
| 59 | 000xx0x00000 | 001101 |
| 60 | 000xx00x0000 | 111101 |
| 61 | 000xx000x000 | 100101 |
| 62 | 000xx000000x | 101100 |
| 63 | 000x0xx00000 | 001101 |
| 64 | 000x0x0x0000 | 111101 |
| 65 | 000x0x00x000 | 100101 |
| 66 | 000x0x00000x | 101100 |
| 67 | 0000xxx00000 | 100000 |
| 68 | 0000xx0x0000 | 010000 |
| 69 | 0000xx00x000 | 001000 |
| 70 | 0000xx000x00 | 000100 |
| 71 | 0000xx0000x0 | 000010 |
| 72 | 0000xx00000x | 000001 |
| 73 | 0000x0xx0000 | 110000 |
| 74 | 0000x0x0x000 | 101000 |
| 75 | 0000x0x00x00 | 100100 |
| 76 | 0000x0x000x0 | 100010 |
| 77 | 0000x0x0000x | 100001 |
| 78 | 0000x00xx000 | 011000 |
| 79 | 0000x00x0x00 | 010100 |
| 80 | 0000x00x00x0 | 010010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 81 | 0000x000xx00 | 001100 |
| 82 | 0000x000x0x0 | 001010 |
| 83 | 0000x000x00x | 001001 |
| 84 | 0000x0000x0x | 000101 |
| 85 | 0000x00000xx | 000011 |
| 86 | 00000xxx0000 | 110000 |
| 87 | 00000xx0x000 | 101000 |
| 88 | 00000xx00x00 | 100100 |
| 89 | 00000xx000x0 | 100010 |
| 90 | 00000xx0000x | 100001 |
| 91 | 00000x0xx000 | 011000 |
| 92 | 00000x0x0x00 | 010100 |
| 93 | 00000x0x00x0 | 010010 |
| 94 | 00000x00xx00 | 001100 |
| 95 | 00000x00x0x0 | 001010 |
| 96 | 00000x00x00x | 001001 |
| 97 | 00000x000x0x | 000101 |
| 98 | 00000x0000xx | 000011 |

**ตารางที่ 3.8 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx00xx000000 | 111100 |
| 2 | xx00x0x00000 | 011100 |
| 3 | xx00x000x000 | 110100 |
| 4 | xx000xx00000 | 110100 |
| 5 | xx000x00x000 | 110100 |
| 6 | x0x0xx000000 | 010001 |
| 7 | x0x0x0x00000 | 110001 |
| 8 | x0x0x000x000 | 011001 |
| 9 | x0x00xx00000 | 110001 |
| 10 | x0x00x00x000 | 011001 |
| 11 | x00xxx000000 | 000110 |
| 12 | x00xx0x00000 | 100110 |
| 13 | x00xx000x000 | 001110 |
| 14 | x00x0xx00000 | 100110 |
| 15 | x00x0x00x000 | 001110 |
| 16 | x000xxx00000 | 001011 |
| 17 | x000xx0x0000 | 111011 |
| 18 | x000xx00x000 | 100011 |
| 19 | x000xx000x00 | 101111 |
| 20 | x000xx0000x0 | 101001 |
| 21 | x000xx00000x | 101010 |
| 22 | x000x0xx0000 | 011011 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมถึงห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.9 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 4 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 23 | x000x0x00x00 | 001111 |
| 24 | x000x00xx000 | 110011 |
| 25 | x000x00x0x00 | 111111 |
| 26 | x000x00x00x0 | 111001 |
| 27 | x000x000xx00 | 100111 |
| 28 | x0000xxx0000 | 011011 |
| 29 | x0000xx00x00 | 001111 |
| 30 | x0000x0xx000 | 110011 |
| 31 | x0000x0x0x00 | 111111 |
| 32 | x0000x0x00x0 | 011001 |
| 33 | x0000x00xx00 | 100111 |
| 34 | 0x00xxx00000 | 110111 |
| 35 | 0x00xx0x0000 | 000111 |
| 36 | 0x00xx00x000 | 011111 |
| 37 | 0x00xx000x00 | 010011 |
| 38 | 0x00xx0000x0 | 010101 |
| 39 | 0x00xx00000x | 010110 |
| 40 | 0x00x0x000x0 | 110101 |
| 41 | 0x00x0x0000x | 110110 |
| 42 | 0x00x000x0x0 | 011101 |
| 43 | 0x00x000x00x | 011110 |
| 44 | 0x000xx000x0 | 110101 |
| 45 | 0x000xx0000x | 110110 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 46 | 0x000x00x0x0 | 011101 |
| 47 | 0x000x00x00x | 011110 |
| 48 | 00x0xxx00000 | 011010 |
| 49 | 00x0xx00x000 | 110010 |
| 50 | 00x0xx000x00 | 111110 |
| 51 | 00x0xx0000x0 | 111000 |
| 52 | 000xxxx00000 | 001101 |
| 53 | 000xxx0x0000 | 111101 |
| 54 | 000xxx00x000 | 100101 |
| 55 | 000xxx00000x | 101100 |
| 56 | 0000xxxx0000 | 110000 |
| 57 | 0000xxx0x000 | 101000 |
| 58 | 0000xxx00x00 | 100100 |
| 59 | 0000xxx000x0 | 100010 |
| 60 | 0000xxx0000x | 100001 |
| 61 | 0000xx0xx000 | 011000 |
| 62 | 0000xx0x0x00 | 010100 |
| 63 | 0000xx0x00x0 | 010010 |
| 64 | 0000xx00xx00 | 001100 |
| 65 | 0000xx00x0x0 | 001010 |
| 66 | 0000xx00x00x | 001001 |
| 67 | 0000xx000x0x | 000101 |
| 68 | 0000xx0000xx | 000011 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 5 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx00xxx00000 | 011100 |
| 2 | xx00xx00x000 | 110100 |
| 3 | x0x0xxx00000 | 110001 |
| 4 | x0x0xx00x000 | 011001 |
| 5 | x00xxxx00000 | 100110 |
| 6 | x00xxx00x000 | 001110 |
| 7 | x000xxxx0000 | 011011 |
| 8 | x000xxx00x00 | 001111 |
| 9 | x000xx0xx000 | 110011 |
| 10 | x000xx0x0x00 | 111111 |
| 11 | x000xx0x00x0 | 111001 |
| 12 | x000xx00xx00 | 100110 |
| 13 | 0x00xxx000x0 | 110101 |
| 14 | 0x00xxx0000x | 110110 |
| 15 | 0x00xx00x0x0 | 011101 |
| 16 | 0x00xx00x00x | 011110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

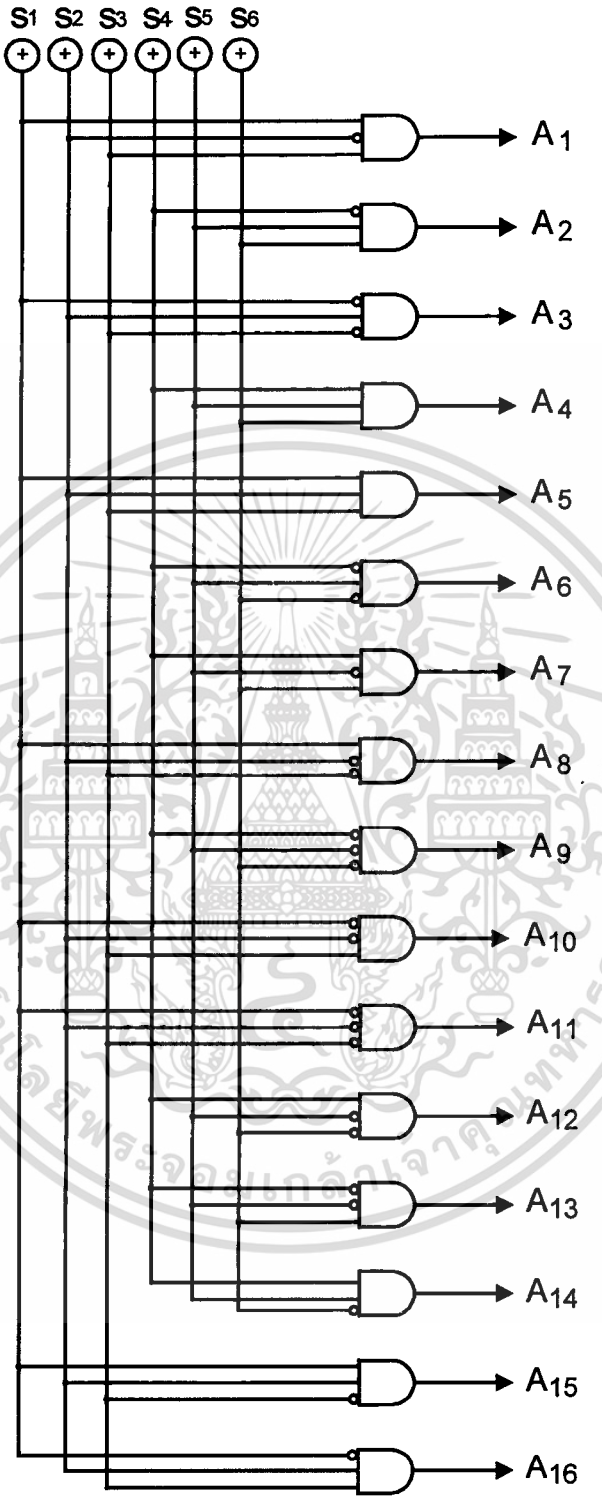
**ตารางที่ 3.10 ค่าเวกเตอร์ซีนโดรมของรูปแบบตำแหน่งบิตที่ผิด 5 บิต**

จากค่าเวกเตอร์ซีนโดรมที่ได้ในตารางที่ 3.6, 3.7, 3.8, 3.9 และ 3.10 จะทำให้สามารถออกแบบวงจรเพื่อแก้ไขความผิดพลาดของเวกเตอร์คำรหัสที่มีรูปแบบ ตามตารางทั้ง 5 ได้ทั้งหมดและสามารถทำการออกแบบวงจรได้ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

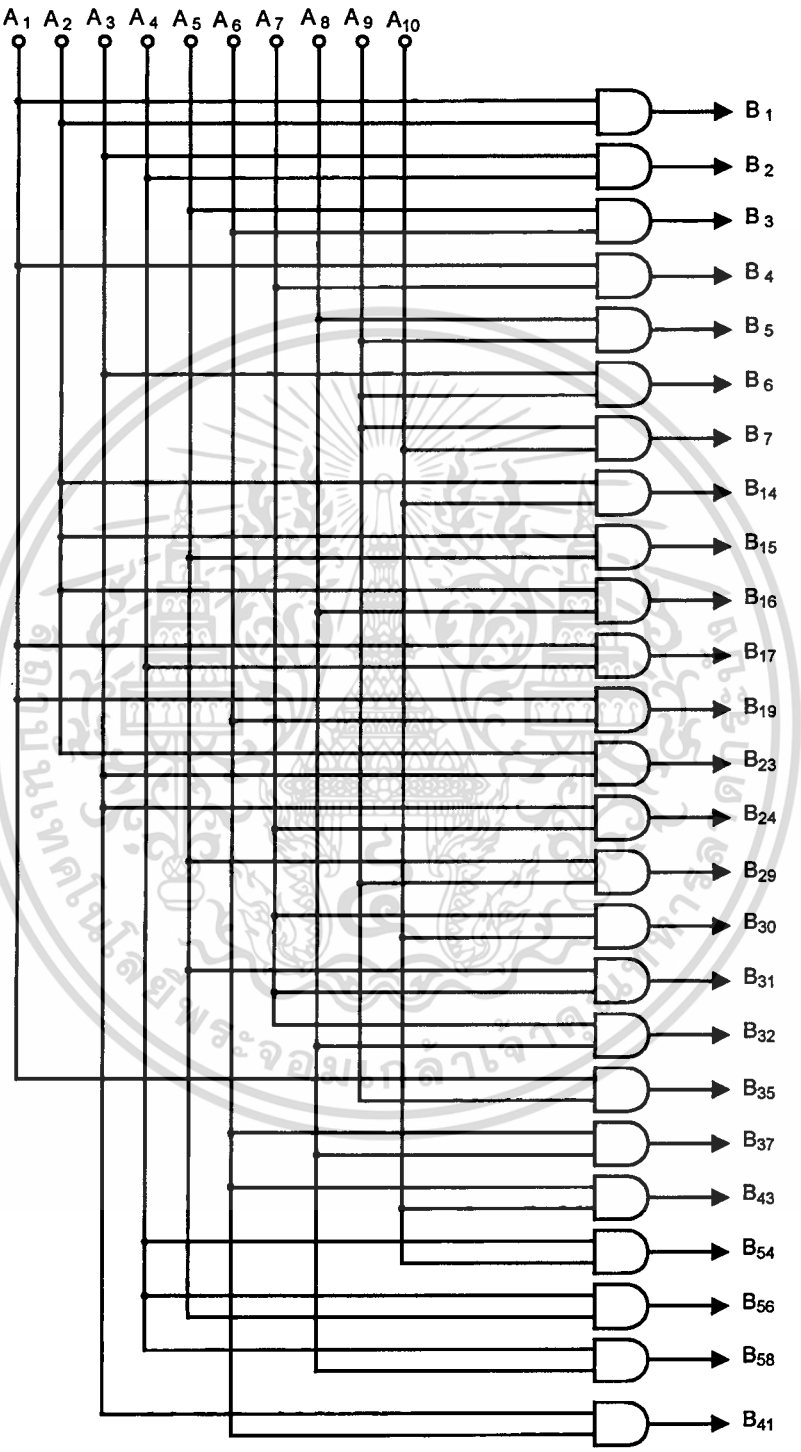
ลักษณะการออกแบบจากวงจรแก้ไขบิตที่ผิดของเวกเตอร์คำรหัสในรูปที่ 3.13 นั้นจะเป็นการนำค่าเวกเตอร์ซีนโดรมที่ได้จากการกำหนดรูปแบบบิตที่ผิดมาออกแบบวงจร โดยจะแบ่งเวกเตอร์ซีนโดรมออกเป็น 2 ชุด ชุดแรก คือ $s_1$, $s_2$, $s_3$ และชุดที่สองคือ $s_4$, $s_5$, $s_6$ ซึ่งแต่ละชุดจะมีรูปแบบที่แตกต่างกันชุดละ 8 รูปแบบ ซึ่งจากรูปจะใช้แอนด์เกต 3 อินพุตจำนวน 16 ตัว เป็นตัวกำหนดรูปแบบต่างๆ ของเวกเตอร์ซีนโดรมและค่าเอาต์พุตที่ได้กำหนดให้เป็น A1,A2, ..., A16 จะนำค่า A1-A16 ไปจัดเรียงตามรูปแบบบิตที่ผิดของเวกเตอร์คำรหัสที่รับเข้ามา เพื่อจะกำหนดตำแหน่งที่ผิดของคำรหัสที่รับเข้ามาซึ่งมีทั้งผิด 1 บิต, ผิด 2 บิต, ผิด 3 บิต, ผิด 4 บิต และผิด 5 บิต จากวงจรค่าเอาต์พุตที่ได้ คือ B1, B2, ...B62 และเมื่อรู้ตำแหน่งบิตที่ผิดของรูปแบบการผิดพลาดจากทั้ง 5 ชุด แล้วก็นำเอาแต่ละชุดที่มีบิตผิดที่ตำแหน่งเดียวกันมารวมกัน ซึ่งถ้าจัดเป็นกลุ่มก็จะได้ 12 กลุ่ม ตามจำนวนตำแหน่งบิตของเวกเตอร์คำรหัสที่รับเข้ามา โดยกำหนดให้เป็นเอาต์พุต E1-E12 จะเป็นตัวแสดงตำแหน่งบิตที่ผิดของแต่ละเวกเตอร์คำรหัสเพื่อจะนำเอารูปแบบ E1-E12 นี้ไปทำการแก้ไขตามกฎการบวกของ GF(2) กับตำแหน่งบิตของเวกเตอร์คำรหัสที่รับเข้ามา จะทำให้ได้เวกเตอร์คำรหัสใหม่คือ C1, C2, ..., C12 ที่ประมาณได้ว่าเป็นเวกเตอร์คำรหัสเดียวกันกับเวกเตอร์คำรหัสที่ส่งจากชุดเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



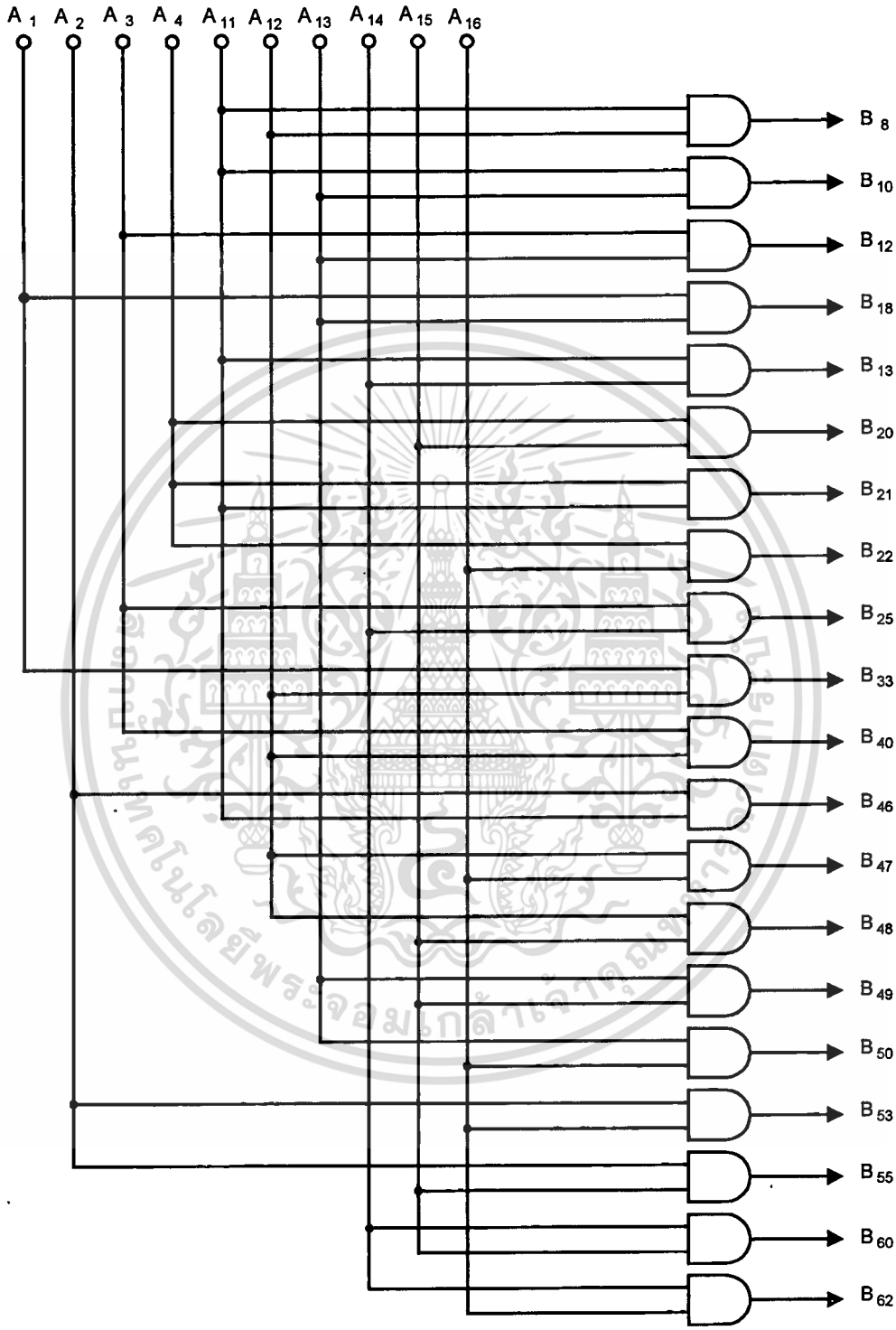

**รูปที่ 3.13 วงจรแก้ไขความผิดพลาดของเวกเตอร์คำรหัสของชุดถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



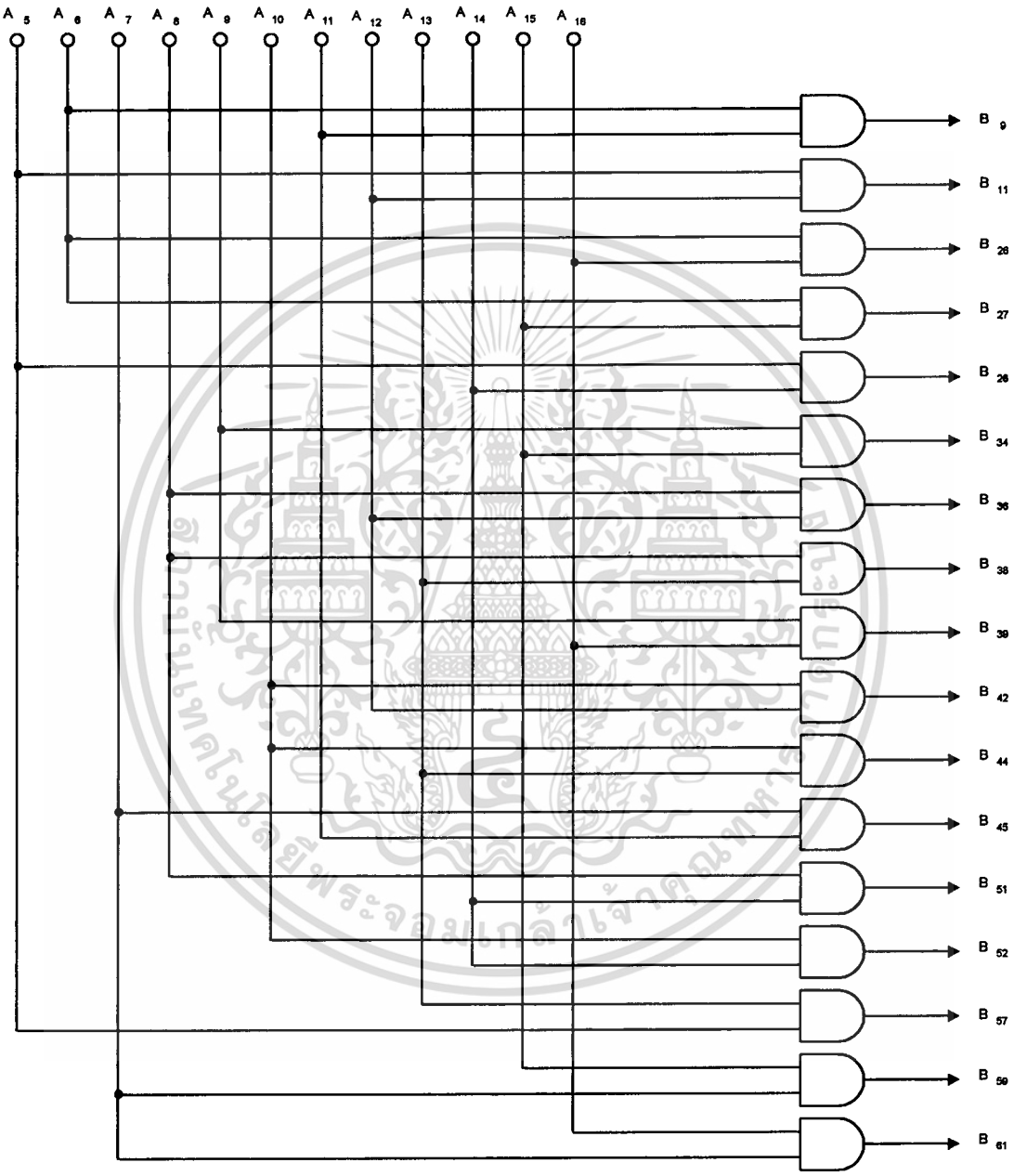

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 3.13 (ต่อ)**



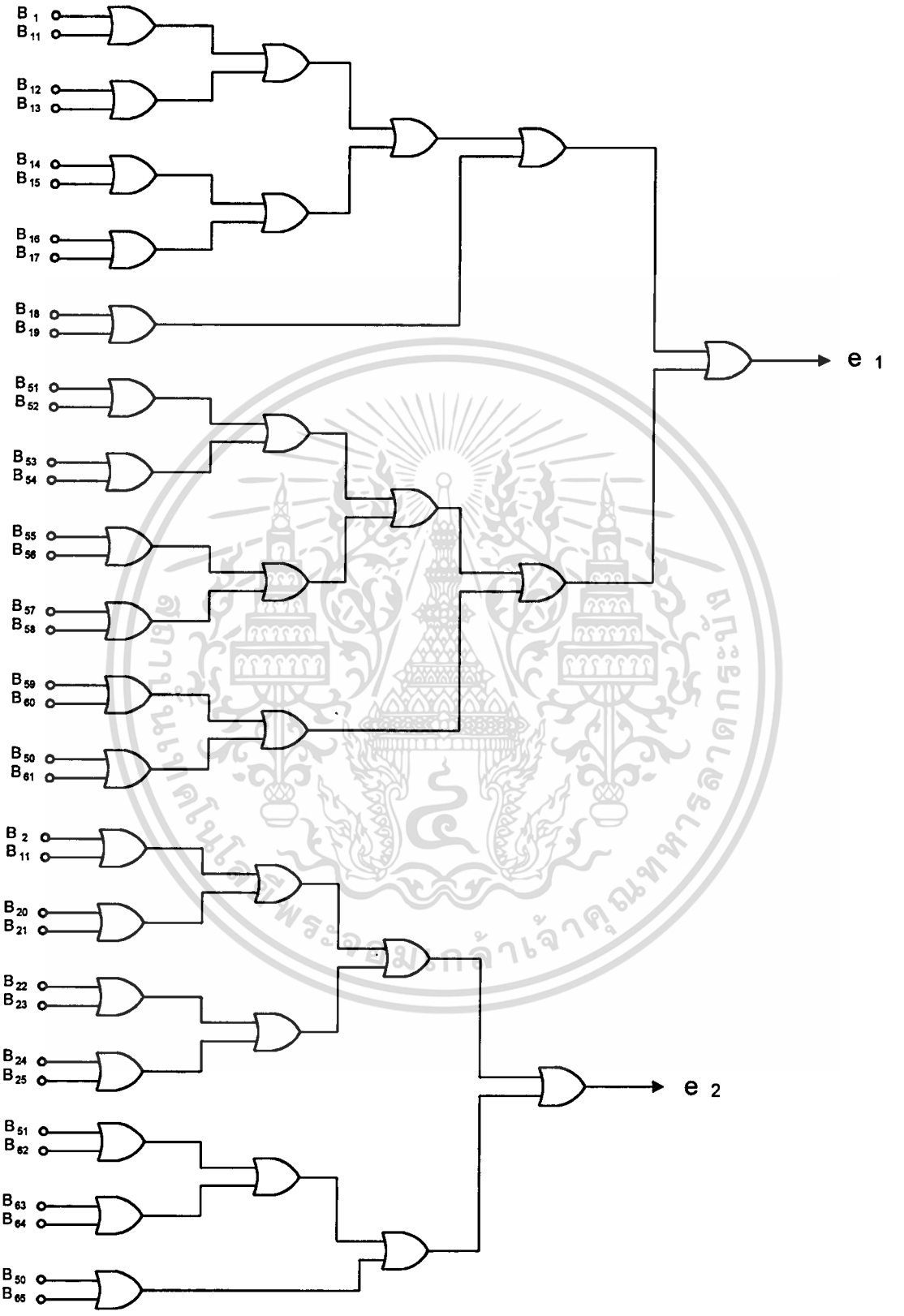

**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



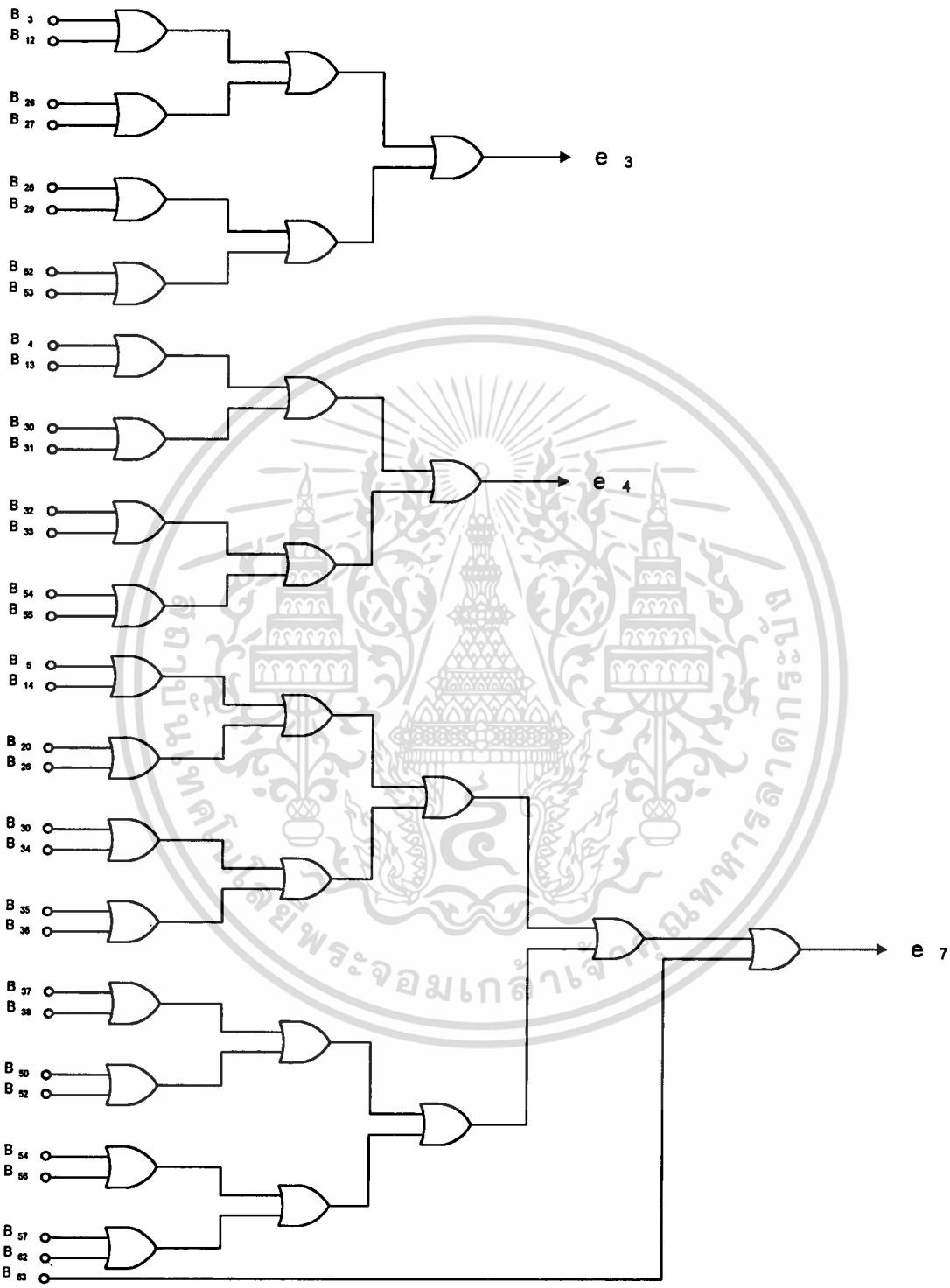

**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



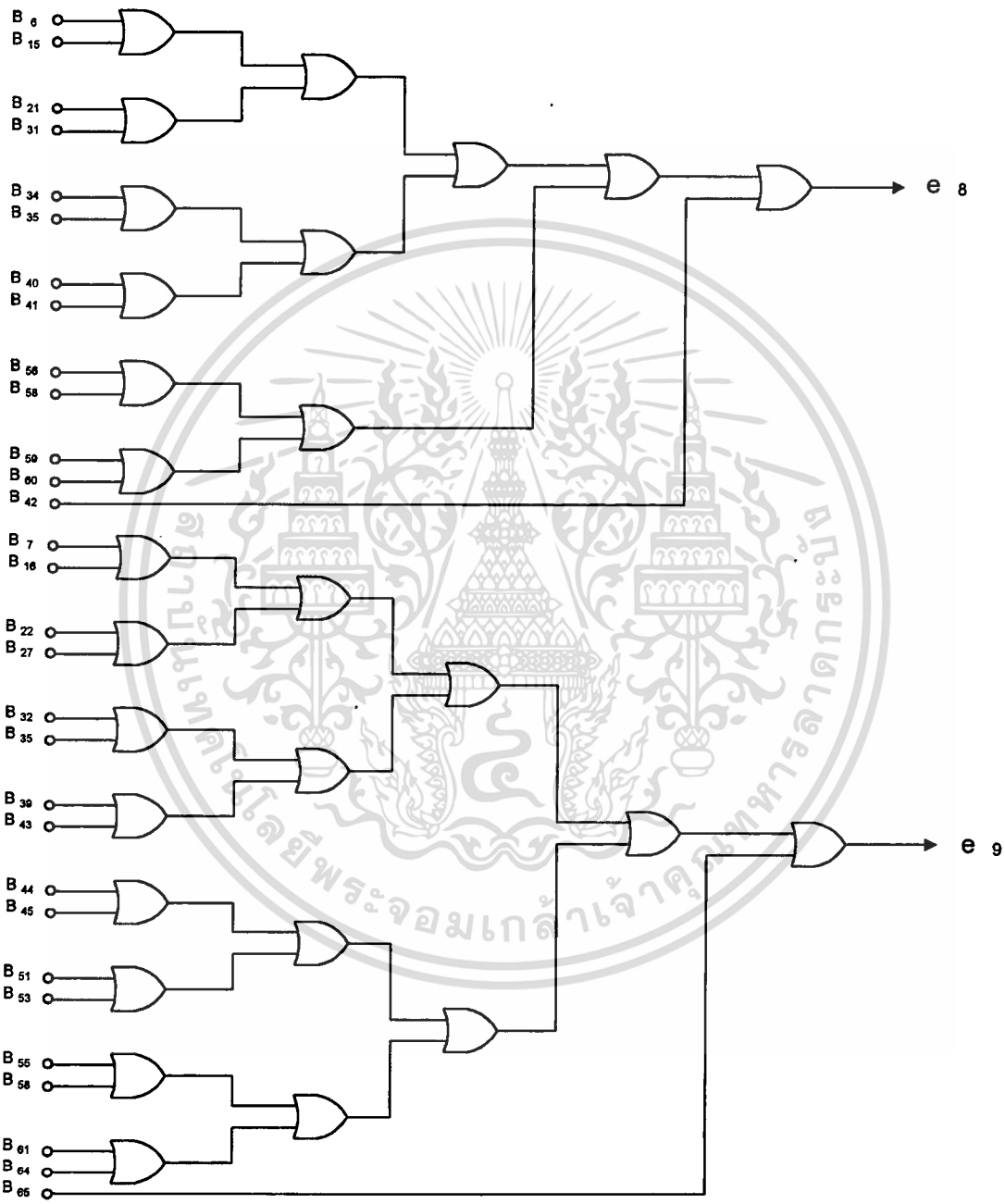

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 3.13 (ต่อ)**



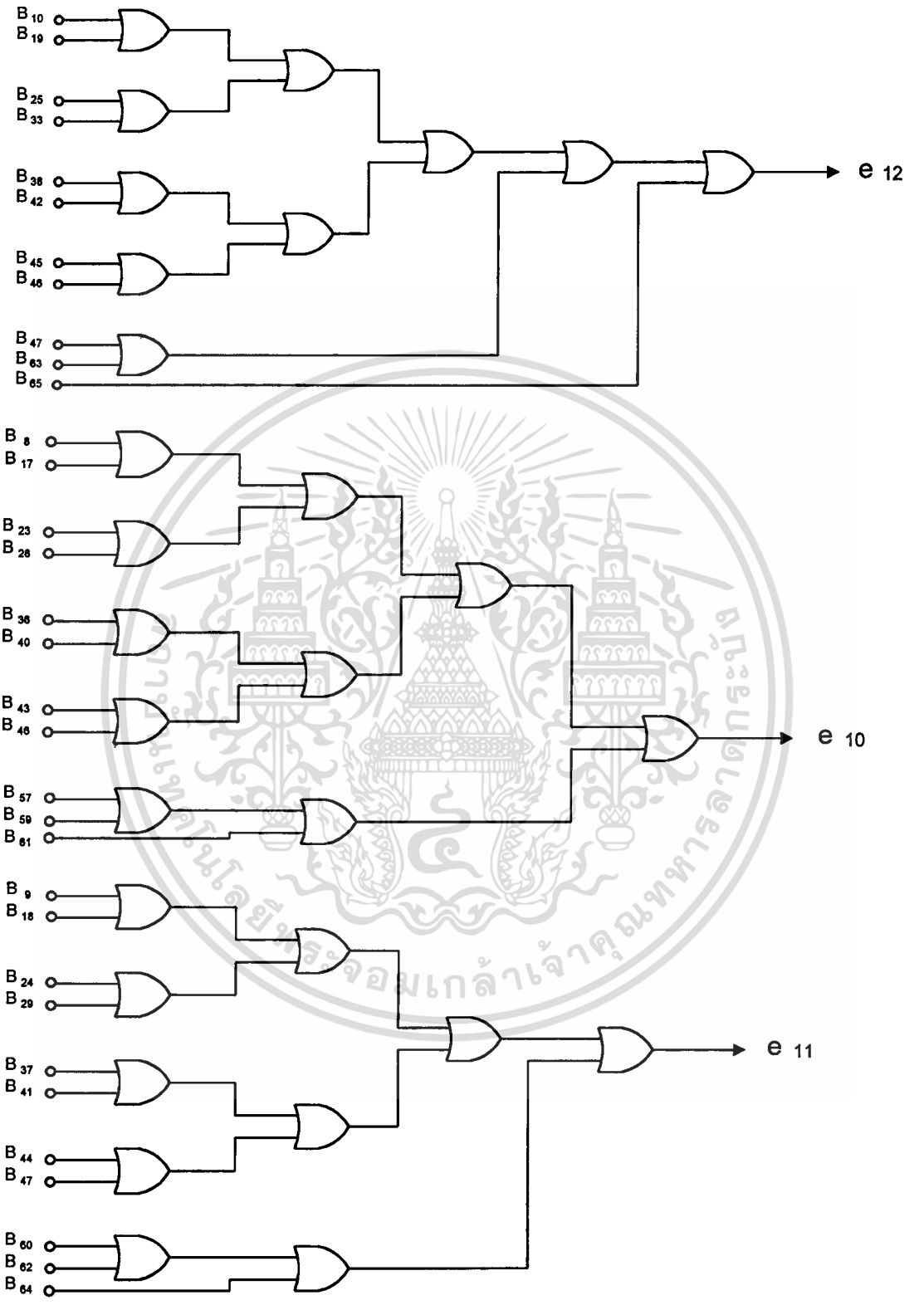

**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



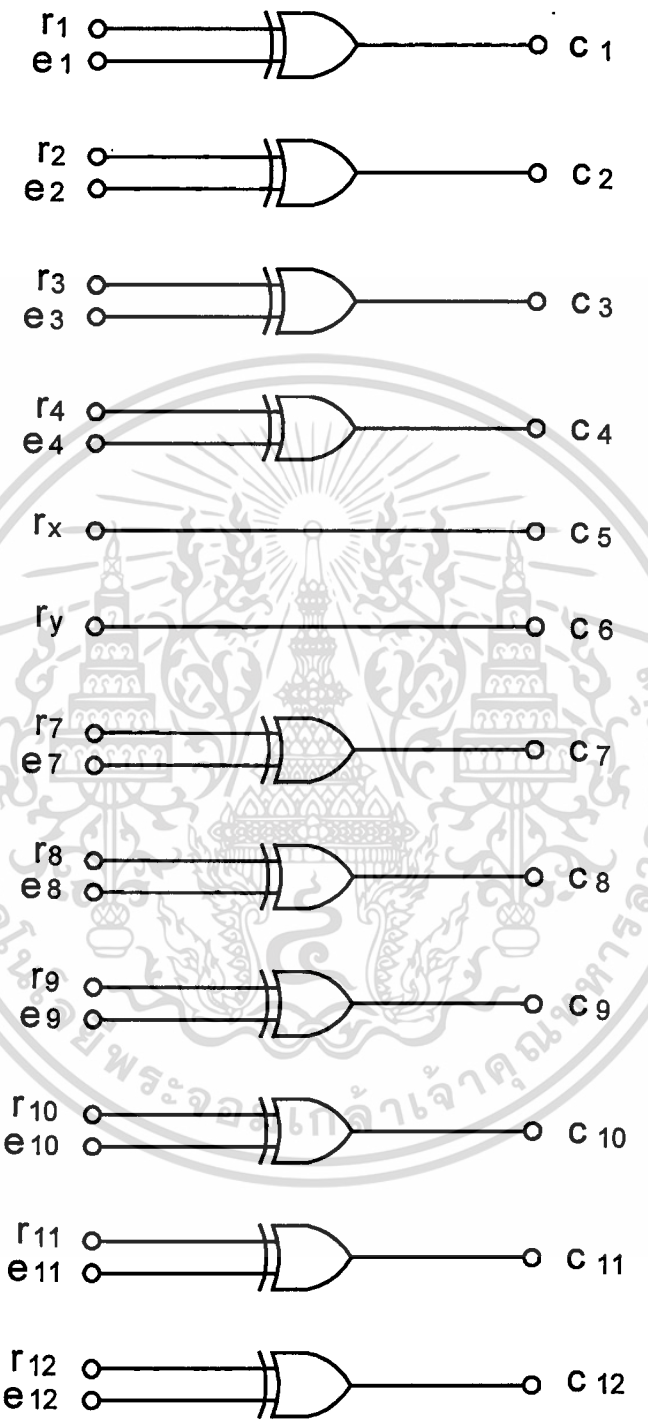

**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

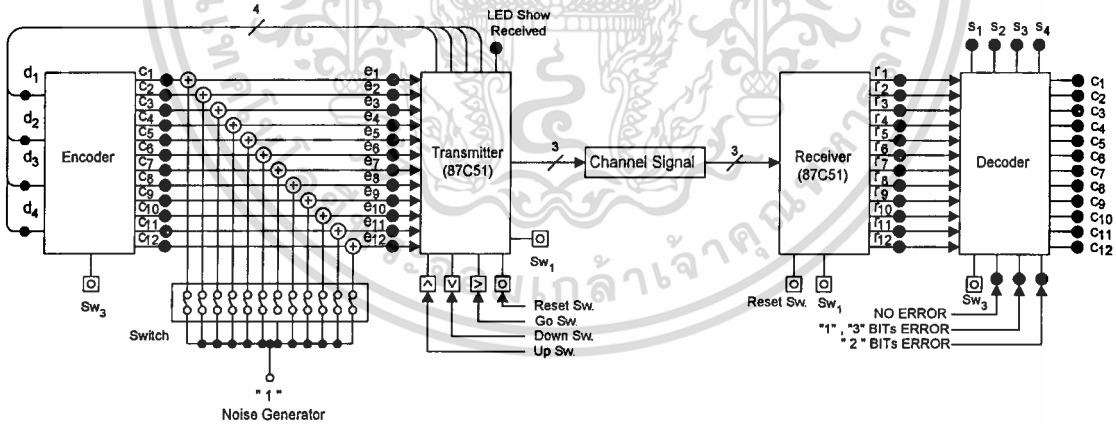



**รูปที่ 3.13 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.5 การออกแบบชุดทดลองปฏิบัติการ

หลังจากการออกแบบชุดเข้ารหัสและชุดถอดรหัสแล้วจำเป็นจะต้องพิจารณาถึงการทำงานร่วมกันระหว่างชุดเข้ารหัสกับชุดถอดรหัส ให้สามารถรับ-ส่งข้อมูลได้ เพื่อนำไปใช้เป็นชุดทดลองปฏิบัติการ นอกจากนี้ยังต้องคำนึงถึงหลักการออกแบบชุดทดลองปฏิบัติการ เพื่อให้เกิดการเรียนรู้กับผู้ที่ทำการทดลองอีกด้วย ดังนั้นปริญญานิพนธ์นี้จึงได้กำหนดรูปแบบชุดทดลองปฏิบัติการดังรูปที่ 3.14 โดยในส่วนของการรับ-ส่งข้อมูลนั้น ได้พิจารณานำเอาอุปกรณ์สนับสนุนการสื่อสาร ไอซีไมโครคอนโทรลเลอร์ เบอร์87C51 ซึ่งมีคุณสมบัติในการรับ-ส่งข้อมูลแบบอนุกรมและสามารถบันทึกรายการคำสั่งการทำงานไว้ภายในได้ด้วย และสำหรับชุดวงจรเข้ารหัสกับวงจรถอดรหัสนั้น ได้ทำการแยกเป็นชุดเข้ารหัสและชุดถอดรหัสภายนอกที่ผู้ทดลองสามารถจะออกแบบวงจรและทดลองต่อวงจรเพื่อใช้รับ-ส่งข้อมูลร่วมกับชุดทดลองปฏิบัติการ,ชุดเข้ารหัสและชุดถอดรหัสภายในที่ได้จัดสร้างขึ้น เพื่อใช้ในการพิสูจน์ข้อเท็จจริงของวิธีการรหัสระบบเชิงเส้น โดยในชุดเข้ารหัสและชุดถอดรหัสภายในนี้ยังสามารถเลือกใช้ได้ 2 แบบ คือ แบบที่มีขนาดของเวกเตอร์คำรหัสจำนวน 8 บิตและ 12 บิต



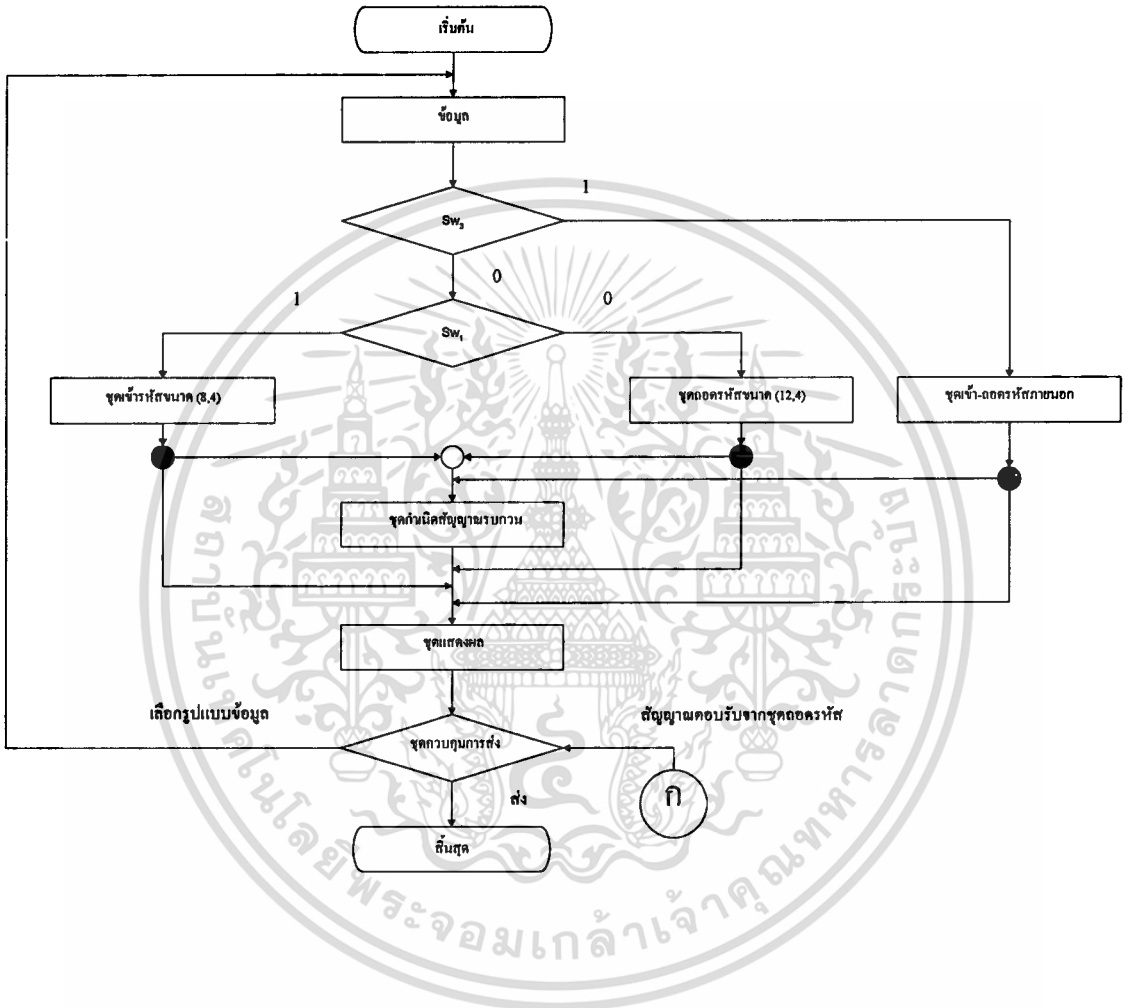

**จากรูป 3.14 ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิต**

จากรูปที่ 3.14 เป็นชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิต ที่สามารถเลือกได้ว่า จะให้มีการเข้ารหัสแบบ (8,4) บิต หรือแบบ (12,4) บิต โดยเลือกที่ $Sw_1$ ถ้ากำหนดให้เป็นลอจิก 0 จะเป็นการเลือกการเข้ารหัสแบบ (12,4) แต่ถ้าเป็นลอจิก 1 จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

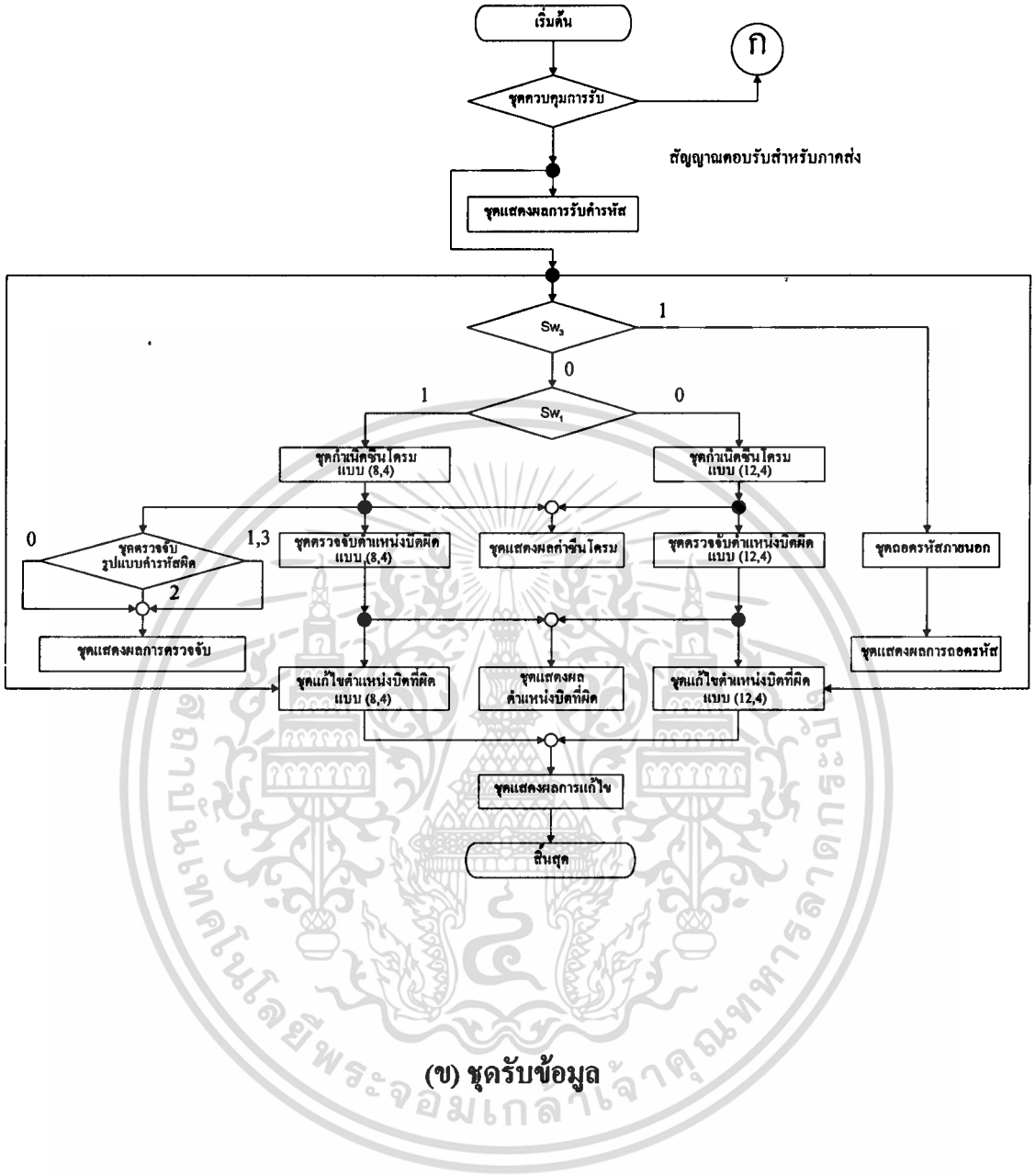
เลือกการเข้ารหัสแบบ (8,4) และในตัวส่งข้อมูล ( Transmitter ) จะทำการผลิตข้อมูลขนาด 4 บิต ออกที่พอร์ต 0 ได้ถึง 16 รูปแบบ ซึ่งสามารถจะเลือกรูปแบบของข้อมูลได้จาก สวิตช์ขึ้น-ลง ( Up-Down Switch ) ส่วนข้อมูลในบิตที่ 5 และบิตที่ 6 ($d_5$,$d_6$) จะกำหนดให้เป็นลอจิก 1 เสมอและจะแสดงรูปแบบของข้อมูลด้วยแอลอีดี $d_1$-$d_6$ เมื่อภาคเข้ารหัสได้รับข้อมูลแล้ว จะทำการเข้ารหัสให้ได้เป็นเวกเตอร์คำรหัสขนาด 8 บิต หรือ 12 บิต และแสดงผลที่แอลอีดี $c_1$-$c_{12}$ เมื่อได้เวกเตอร์คำรหัสแล้วจะสามารถป้อนสัญญาณรบกวนเข้าที่บิตใดก็ได้ โดยผ่านสวิตช์ป้อนสัญญาณรบกวน ( Noise Switch ) แล้วจะได้เวกเตอร์คำรหัสที่มีตำแหน่งบิตที่ผิดตามที่กำหนด สังเกตได้จากแอลอีดี $e_1$-$e_{12}$ เทียบกับแอลอีดี $c_1$-$c_{12}$ และเมื่อตัวส่งข้อมูลได้รับเวกเตอร์คำรหัสเข้ามาที่พอร์ต 1 ก็จะทำการค้างสภาวะเวกเตอร์คำรหัสนั้นๆ ไว้ เพื่อรอคำสั่งส่งจากสวิตช์ส่ง (Go Switch) เมื่อกดสวิตช์ส่งเวกเตอร์คำรหัสนั้น จะถูกแปลงจากข้อมูลแบบขนานให้อยู่ในรูปของข้อมูลแบบอนุกรมผ่านทางพอร์ต 3 แล้วส่งผ่านช่องสัญญาณ ซึ่งเป็นสายนำสัญญาณมายังภาครับข้อมูลที่พอร์ต 3 เช่นกันเมื่อตัวรับข้อมูล (Receiver) ได้รับเวกเตอร์คำรหัสแล้วจะทำการแปลงข้อมูลจากข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนาน โดยสังเกตได้จากแอลอีดี $r_1$-$r_{12}$ แล้วจึงส่งเข้าตัวถอดรหัส พร้อมกันนั้นตัวรับข้อมูลก็จะส่งสัญญาณตอบรับมายังตัวส่งข้อมูล เพื่อบอกให้ทราบว่าตัวรับข้อมูลได้รับข้อมูลแล้ว โดยจะแสดงที่แอลอีดีตอบรับ (Received)ที่ตัวส่งและที่ตัวถอดรหัสก็จะทำการตรวจจับและแก้ไขความผิดพลาดของเวกเตอร์คำรหัสที่มีความสอดคล้องกับการเลือกตัวเข้ารหัสที่ภาคส่ง และผลที่ได้จากการแก้ไขเวกเตอร์คำรหัสที่ผิดพลาด จะแสดงได้ตามแอลอีดี $c_1$-$c_{12}$ แต่ถ้าหากต้องการจะใช้ชุดเข้ารหัสและถอดรหัสจากภายนอกรับ-ส่งข้อมูล โดยผ่านตัวรับ-ส่ง ในชุดทดลองนี้ก็กระทำได้โดยการเลือก $Sw_3$ ให้เป็นลอจิก 1 พร้อมกับต่อชุดข้อมูลที่เป็นเวกเตอร์คำรหัสของทั้งชุดเข้ารหัสและชุดถอดรหัสภายนอกเข้าที่จุดต่อที่กำหนดไว้ในชุดทดลองแล้วก็ปฏิบัติในลักษณะเดียวกันกับการส่งข้อมูลจากชุดเข้ารหัสและชุดถอดรหัสภายในกรณีที่เกิดการทำงานที่ผิดพลาดระหว่าง การรับ-ส่งข้อมูล ก็ทำการแก้ไขโดยการกดสวิตช์รีเซตของทั้งภาครับและภาคส่ง และจากลักษณะการทำงานทั้งหมดที่กล่าวมา จะสามารถเขียนเป็นผังลำดับกระบวนการทำงานของชุดทดลองนี้ได้ดังรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**(ก) ชุดส่งข้อมูล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(ข) ชุดรับข้อมูล

**รูปที่ 3.15 ผังกระบวนการชุดรับ-ส่งข้อมูลการเข้ารหัสช่องสัญญาณ**

จากผังกระบวนการในรูปที่ 3.15 สามารถนำมาสร้างเป็นวงจรภาคส่งและวงจรภาครับได้ โดยการนำเอาชุดวงจรเข้ารหัสทั้งแบบ (8,4) บิตและ (12,4) บิตของภาคส่งและวงจร ถอดรหัสทั้งแบบ (8,4) บิตและ (12,4) บิตที่ออกแบบไว้ในหัวข้อที่ 3.3 และ 3.4 มาจัดทำเป็น วงจรใหม่ที่ประกอบขึ้นจากอุปกรณ์ชนิดโปรแกรมได้ (Programmable Logic Device : PLD) ดังรูปในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.6 การออกแบบใบงานการทดลอง

จากหลักการสอนและจุดมุ่งหมายการสอนเชิงปฏิบัติ ในหัวข้อ2.5 จะได้นำมาใช้ในการกำหนดรูปแบบการทดลอง ลักษณะการทดลอง รวมทั้งใบงานการทดลอง ให้เป็นไปตามลำดับการเรียนรู้ที่เริ่มจากง่ายไปยาก ทั้งนี้ก็เพื่อให้เกิดทักษะการเรียนรู้และความเข้าใจในวิธีการออกแบบรวมถึงลักษณะการทำงาน ซึ่งจะแบ่งใบงานการทดลองออกเป็น 3 ใบงานการทดลองด้วยกัน คือ

1. ชุดทดลองการเข้ารหัสช่องสัญญาณแบบรหัสระบบเชิงเส้นขนาด (8,4) บิต
2. ชุดทดลองการถอดรหัสช่องสัญญาณแบบรหัสระบบเชิงเส้นขนาด (8,4) บิต
3. ชุดทดลองการเข้ารหัสและถอดรหัสช่องสัญญาณแบบรหัสเชิงเส้นขนาด (8,4) บิต และ (12,4) บิต

โดยอาศัยรูปแบบตัวอย่างใบงานดังรูปที่ 2.28 จะได้นำมากำหนดรูปแบบใบงานการทดลอง ซึ่งมีลักษณะดังนี้

ตัวอย่างใบงาน

การทดลองที่ .....

เรื่อง .....

วิชา ..... รหัสวิชา ..... หน่วยกิต .....

สาขาวิชาวิศวกรรมโทรคมนาคม ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

วัตถุประสงค์

.....

.....

ทฤษฎีและหลักการ

.....

.....

เครื่องมือและอุปกรณ์

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ลำดับขั้นการทดลอง**

.....

.....

.....

**บันทึกผลการทดลอง**

.....

.....

.....

**คำถามท้ายการทดลอง**

1. .....
2. .....
3. .....

**สรุปผลการทดลอง**

.....

.....

.....

**แหล่งค้นคว้า**

1. .....
2. .....
3. .....

**รูปที่ 3.16 ตัวอย่างใบงานการทดลอง**

**สำหรับรายละเอียดของแต่ละใบงานจะสามารถดูได้ที่ภาคผนวก ก.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

## การทดลองและผลการทดลอง

สิ่งที่ถือได้ว่าเป็นการพิสูจน์ข้อเท็จจริงของทฤษฎีและหลักการต่างๆนั้น ก็คือ การทดลองและจากการที่ได้ออกแบบวงจรชุดเข้ารหัสและชุดถอดรหัสตามทฤษฎีและหลักการมาแล้วนั้น จะสามารถพิสูจน์ให้เห็นถึงข้อเท็จจริงได้ว่าเป็นไปตามทฤษฎีและหลักการได้โดยการทดลองวงจรที่ได้ออกแบบไว้ ซึ่งในการทดลองเพื่อให้เห็นข้อเท็จจริงต่างๆ นั้นจะแบ่งออกเป็น

## 4.1 การทดลองวงจรส่วนประกอบต่างๆ ของชุดทดลอง

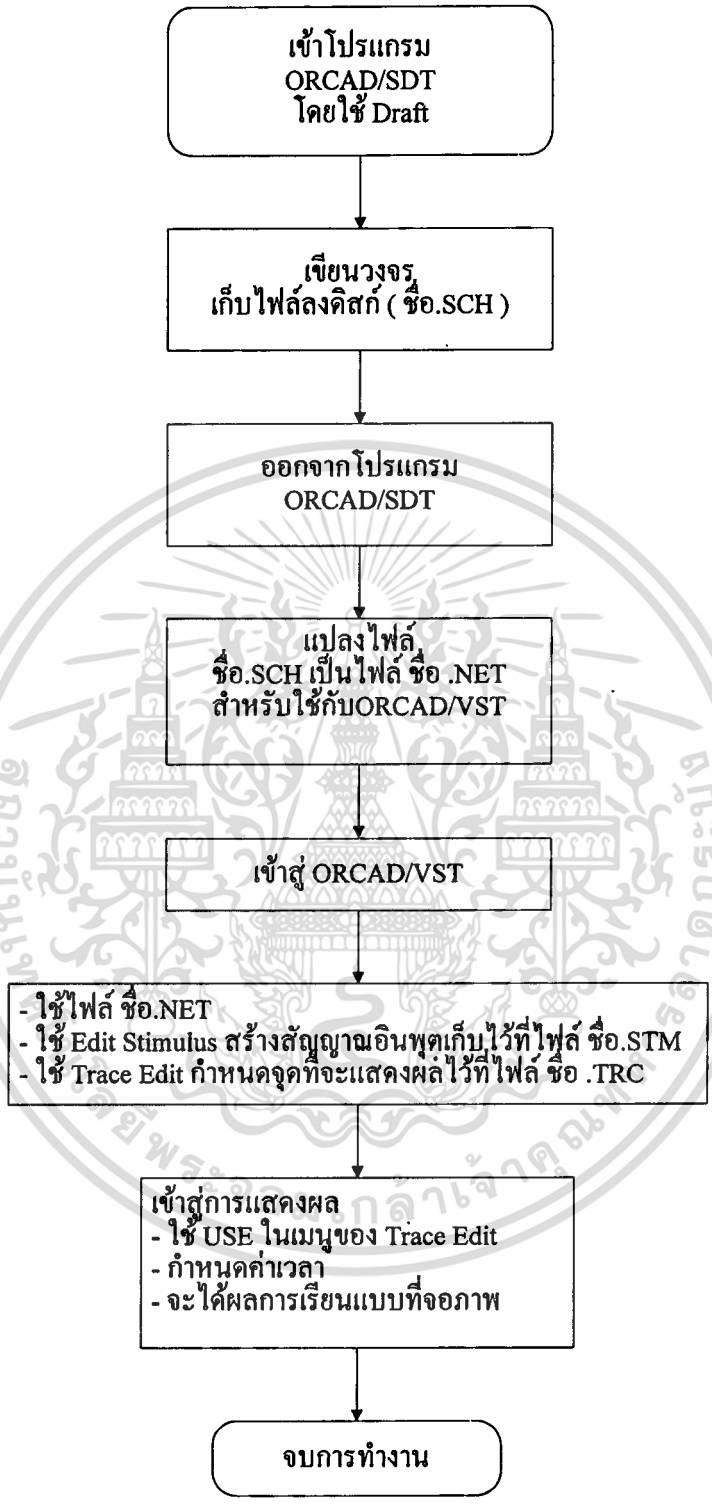
การทดลองวงจรในส่วนนี้เป็นการทดลองในแต่ละส่วนของวงจรก่อนที่จะนำมาประกอบเป็นชุดทดลองปฏิบัติการที่เสร็จสมบูรณ์ โดยจัดเป็น

### 4.1.1 การทดลองชุดวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต

หลังจากที่ได้ออกแบบวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิตแล้ว จะมีการทดลองและทดสอบการทำงานของวงจรทั้งทางซอฟต์แวร์ ( Software ) และทางฮาร์ดแวร์( Hardware )

**การทดลองทางซอฟต์แวร์** ได้แก่ การจำลองการทำงานของวงจรด้วยโปรแกรมสำเร็จรูปออร์แคด ( ORCAD Simulation Program ) ซึ่งมีขั้นตอนในการใช้งานดังนี้

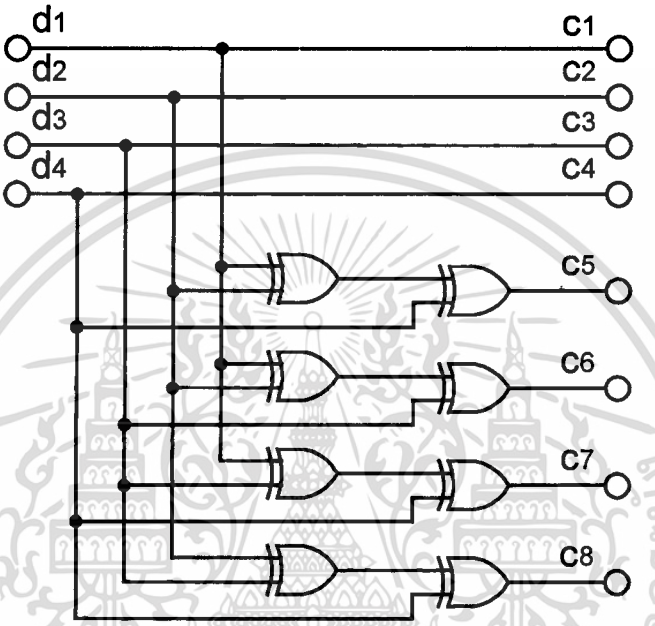
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.1 ผังขั้นตอนการใช้งานของชุดโปรแกรมสำเร็จรูปออร์แคด**

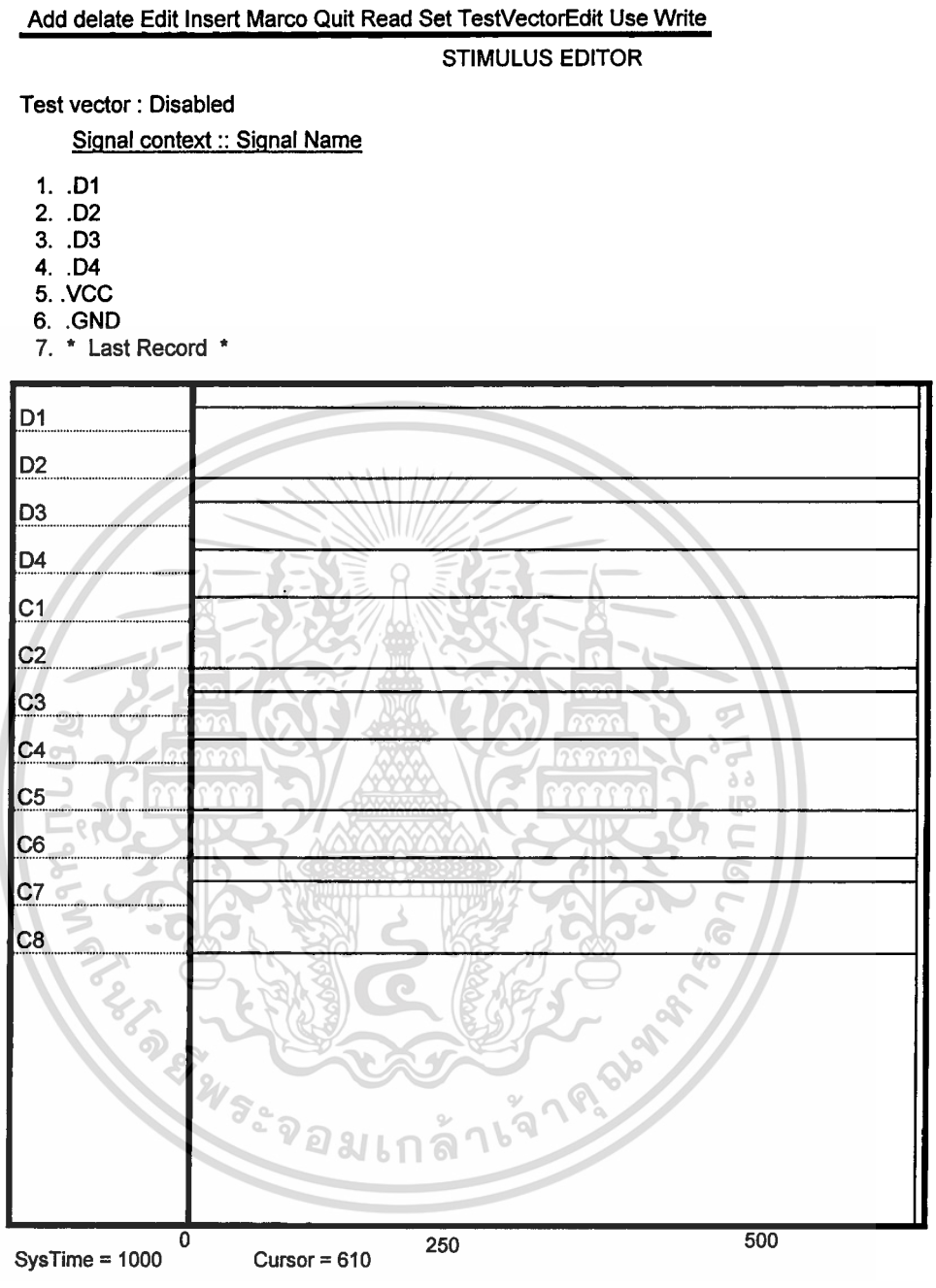
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากขั้นตอนที่ 2 จะเขียนวงจรเข้ารหัสได้ดังรูปที่ 4.2 และด้วยการกำหนดค่าข้อมูลเป็น 1011 ในขั้นตอนที่ 6 แล้วจะสามารถจำลองการทำงานของวงจรให้ได้ผลเป็นคำรหัส 10110010 ดังรูปที่ 4.3




**รูปที่ 4.2 วงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




$D_{1-4}$ : ลำดับข้อมูล

$C_{1-8}$ : เวกเตอร์คำรหัส

**รูปที่ 4.3 ผลจากการจำลองการทำงานด้วยข้อมูล 1011**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

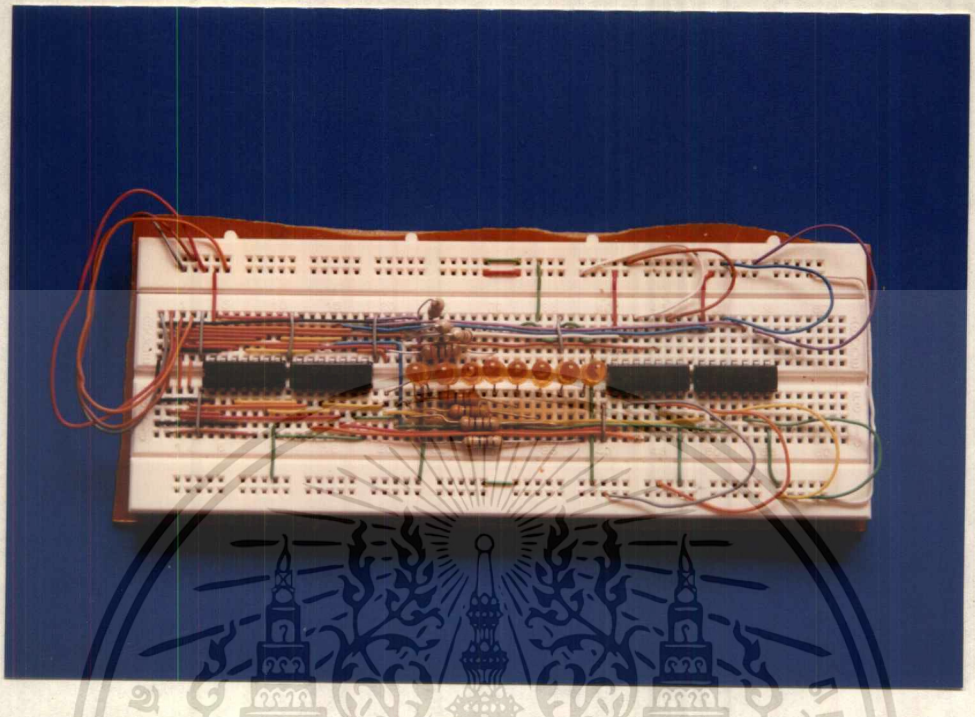
และจากการจำลองการทำงานของวงจรชุดเข้ารหัสด้วยโปรแกรมสำเร็จรูปออร์แคด จะได้ผลของการจำลองดังตารางที่ 4.1

| ลำดับที่ | ลำดับข้อมูล | ลำดับคำรหัส |
|---|---|---|
| 1 | 0000 | 00000000 |
| 2 | 0001 | 00011011 |
| 3 | 0010 | 00100111 |
| 4 | 0011 | 00111100 |
| 5 | 0100 | 01001101 |
| 6 | 0101 | 01010110 |
| 7 | 0110 | 01101010 |
| 8 | 0111 | 01110001 |
| 9 | 1000 | 10001110 |
| 10 | 1001 | 10010101 |
| 11 | 1010 | 10101001 |
| 12 | 1011 | 10110010 |
| 13 | 1100 | 11000011 |
| 14 | 1101 | 11011000 |
| 15 | 1110 | 11100100 |
| 16 | 1111 | 11111111 |

**ตารางที่ 4.1 ผลของเวกเตอร์คำรหัสที่ได้จากการจำลองการทำงานของข้อมูลทั้ง 16 รูปแบบ**

**การทดลองทางฮาร์ดแวร์** ได้แก่ การทดลองด้วยการนำวงจรที่ได้ออกแบบมาสร้างเป็นวงจรลอจิกเกต โดยจะนำเอาอุปกรณ์ไอซีทีทีแอล ( IC.TTL ) ชนิดเอ็กคลูซีฟออร์เกตเบอร์ 74LS86 มาประกอบวงจรบนแผงประกอบวงจร ( Photo Board ) ดังรูปที่ 4.4

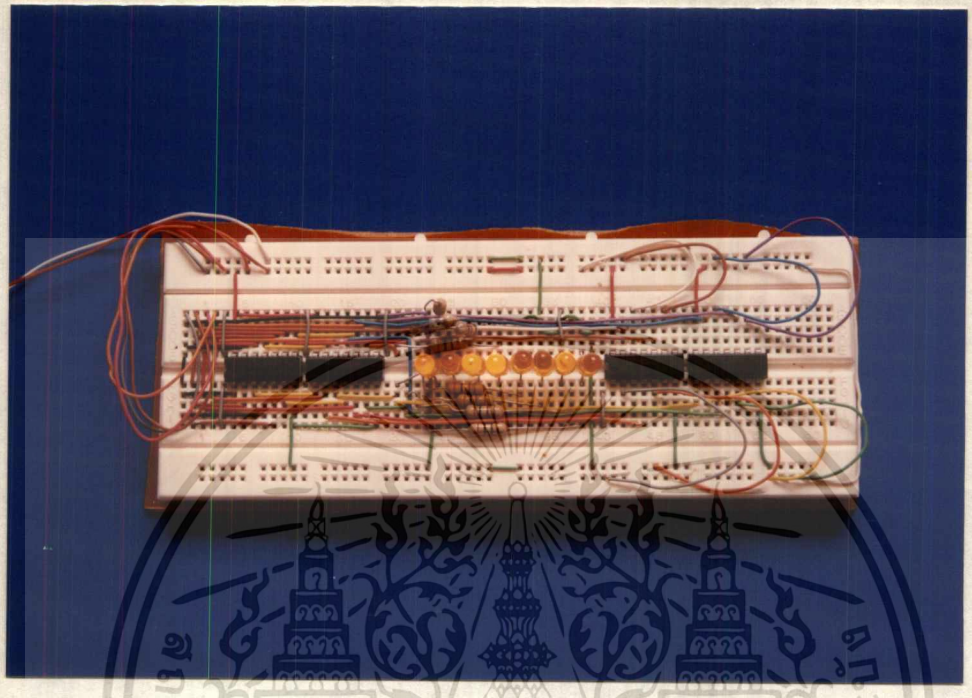
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.4 ภาพวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต**

และหลังจากการป้อนข้อมูล 1011 จะได้ผลที่เป็นคำรหัส 1011001 ปรากฏบนตัวอุปกรณ์แอลอีดี ( LED ) โดยถ้าแอลอีดีสว่างแทนลอจิก 1 และแอลอีดีไม่สว่างแทนลอจิก 0 ดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



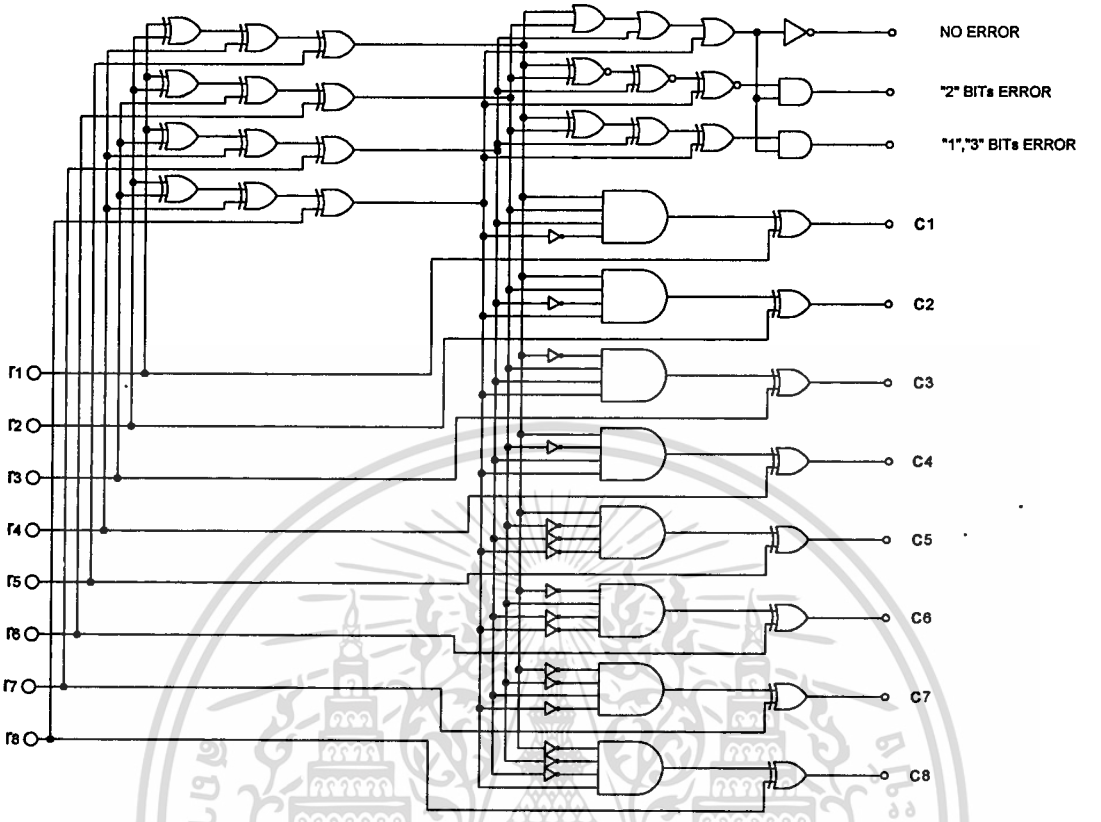
**รูปที่ 4.5 ภาพการทดสอบวงจรเข้ารหัสระบบเชิงเส้นขนาด (8,4) บิต ด้วยข้อมูล 1011**

### 4.1.2 การทดลองชุดวงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต

ในทำนองเดียวกันกับชุดวงจรเข้ารหัสระบบเชิงเส้นที่มีการทดลองทั้งทางด้านซอฟต์แวร์ และฮาร์ดแวร์

**การทดลองทางด้านซอฟต์แวร์** จะใช้โปรแกรมสำเร็จรูปออร์แคดในการจำลองการทำงานเช่นกัน ซึ่งจากขั้นตอนที่ 2 จะเขียนวงจรถอดรหัสได้ดังรูปที่ 4.6 แล้วจากนั้นทำการกำหนดค่าเวกเตอร์คำรหัสซึ่งแบ่งได้เป็น 4 กรณี ตามขั้นตอนการใช้งานของรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.6 วงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต**

**กรณีแรก** คำรหัสที่ไม่มีบิตที่ผิด กำหนดเป็น 10110010 และเมื่อจำลองการทำงานให้กับวงจรแล้วค่าของเวกเตอร์ซีนโดรมที่ได้คือ 0000 ดังรูปที่ 4.7 ค่าของบิตที่ผิดพลาดคือ 00000000 ดังรูปที่ 4.8 และค่าของข้อมูลที่ได้จากคำรหัส 10110010 คือ 10110000 ดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

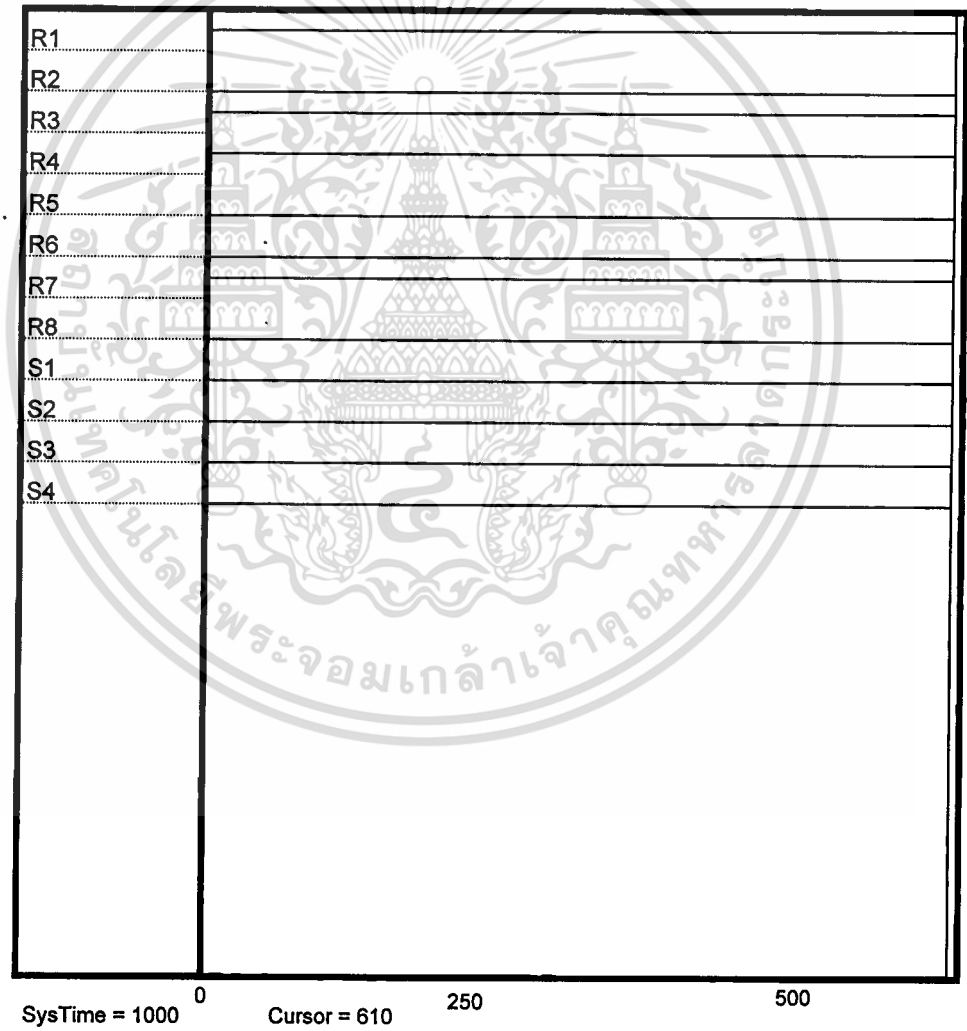
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .VCC
10. .GND
11. * Last Record *

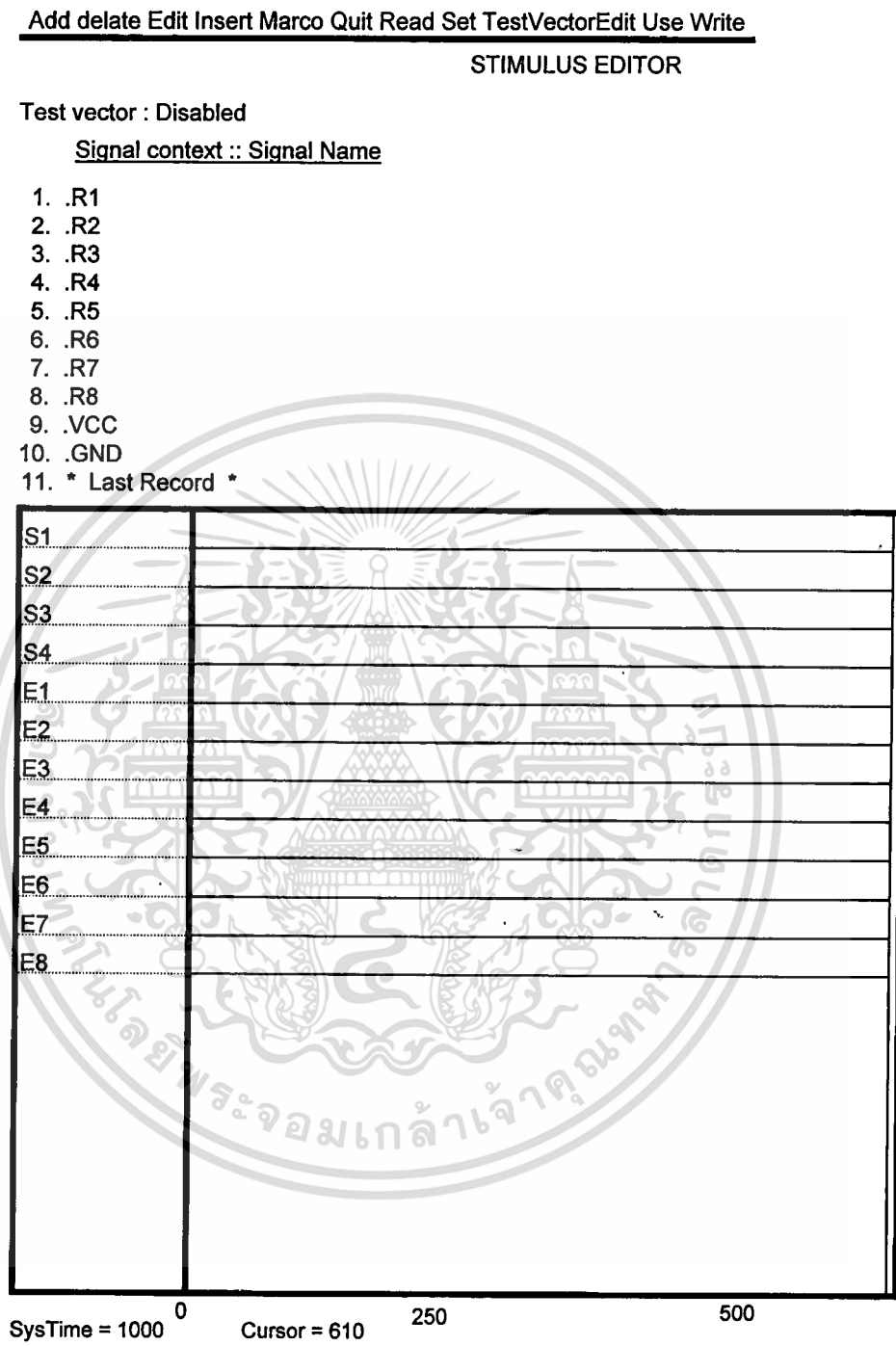



$R_{1-8}$ : เวกเตอร์คำรหัส

$S_{1-4}$ : เวกเตอร์ซีนโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.7 ผลของเวกเตอร์ซีนโดรมที่ได้จากคำรหัส 10110010**

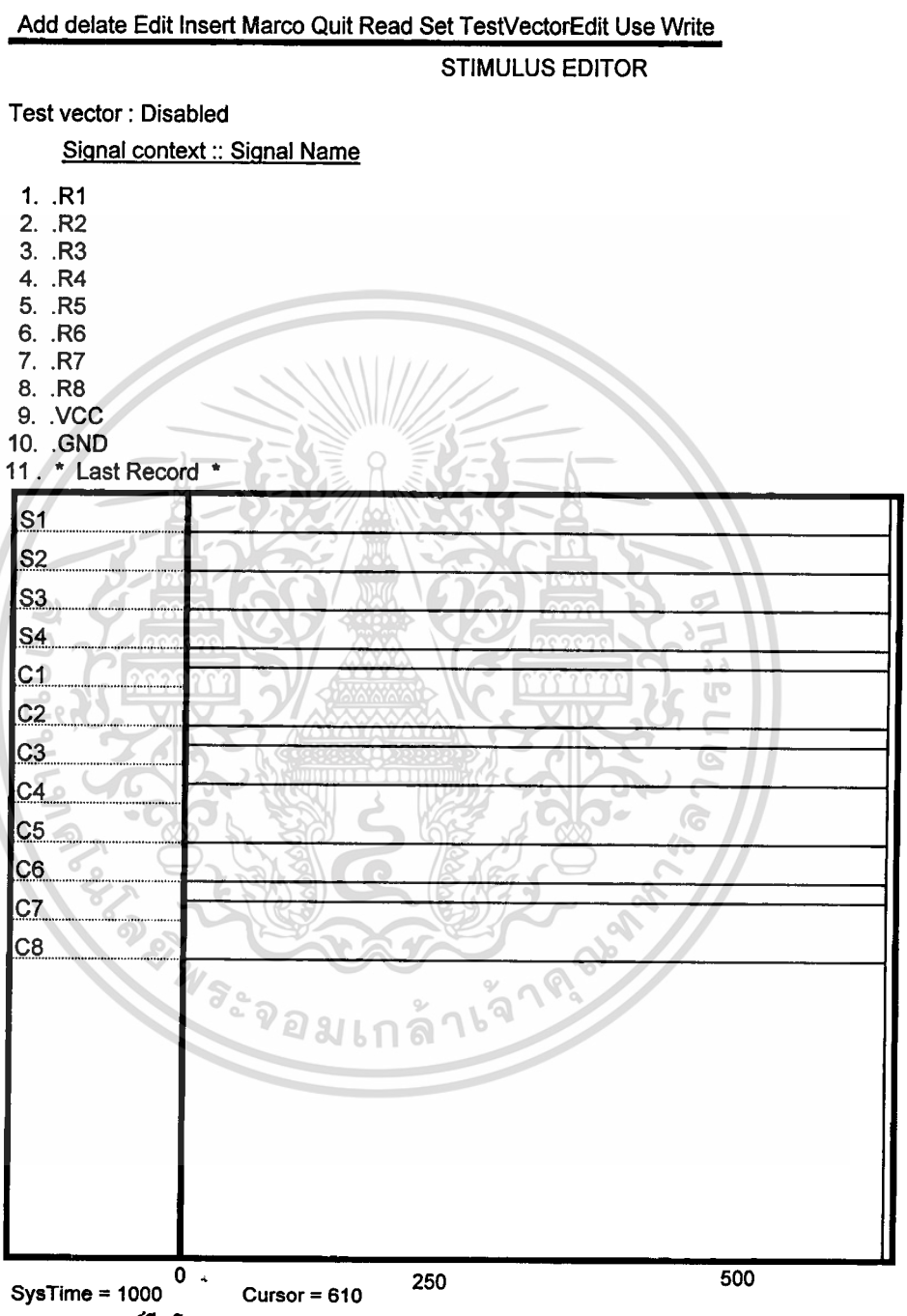



$S_{1-4}$ : เวกเตอร์ซีนโครม

$E_{1-8}$ : เวกเตอร์รูปแบบตำแหน่งบิตผิด

**รูปที่ 4.8 ผลของรูปแบบบิตที่ผิดจากเวกเตอร์คำรหัส 10110010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.9 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 10110010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

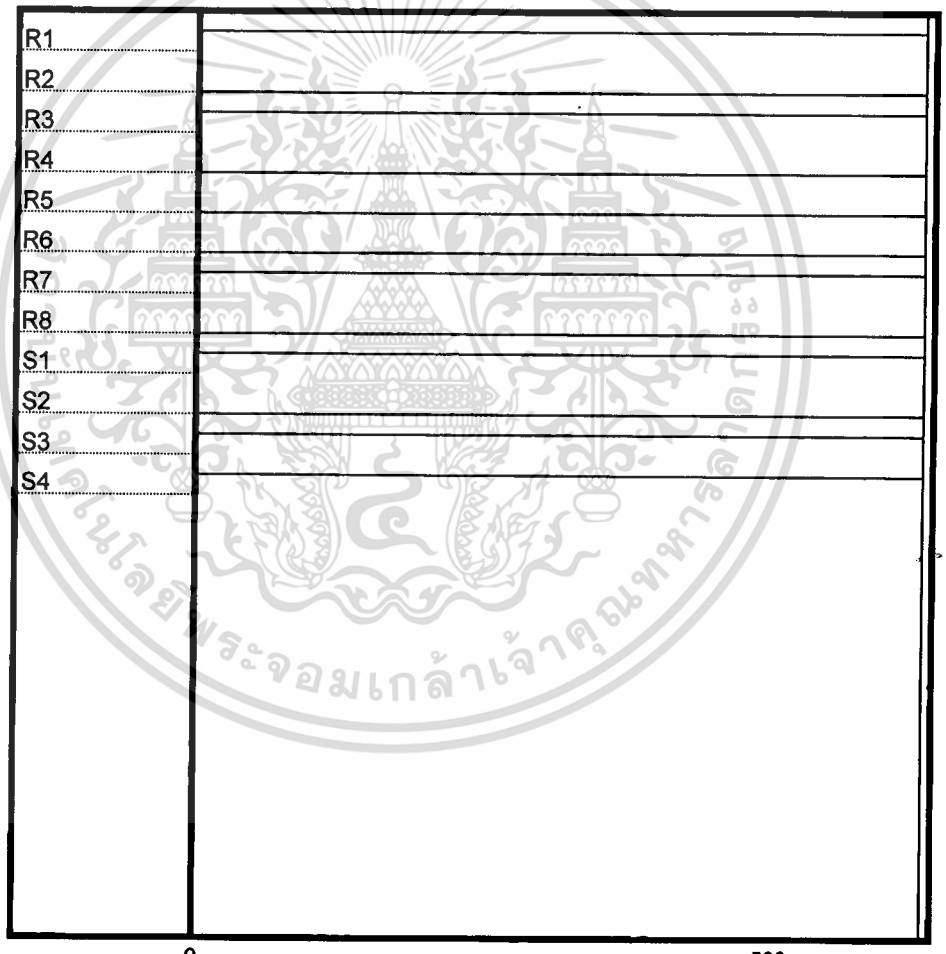
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .VCC
10. .GND
11. * Last Record *

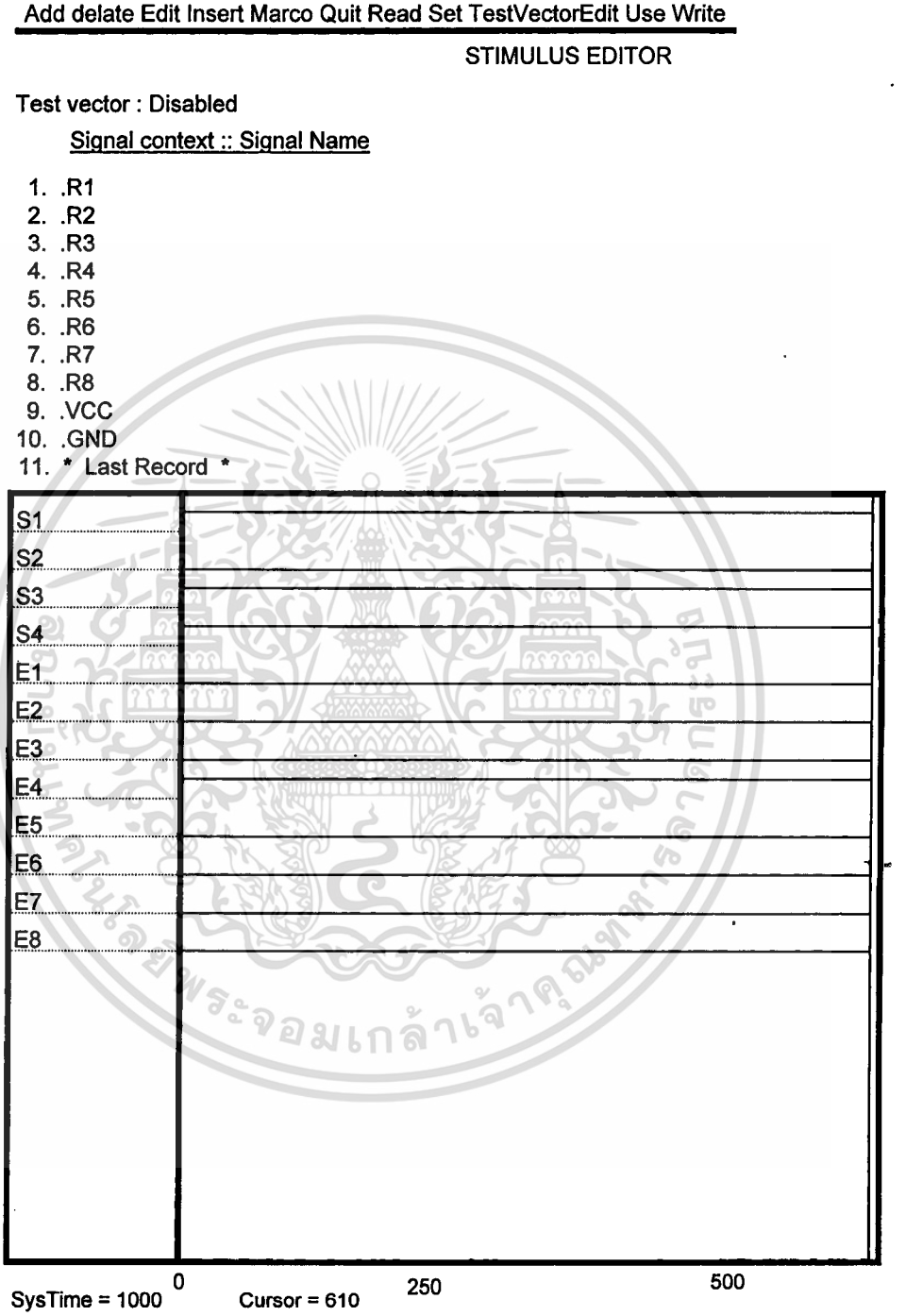



$R_{1-8}$ : เวกเตอร์คำรหัส

$S_{1-4}$ : เวกเตอร์ซีนโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.10 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 10100010**

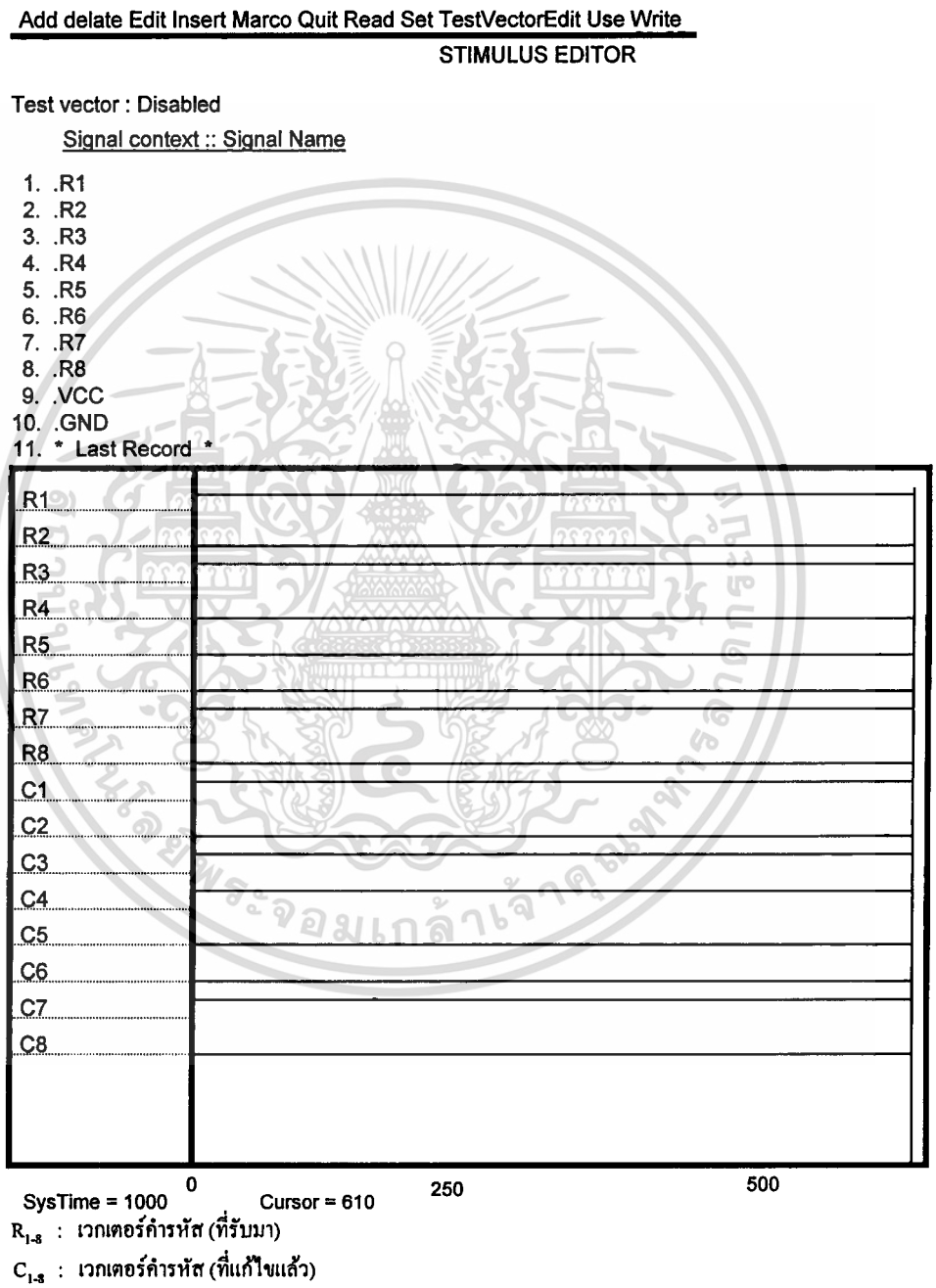



$S_{1-4}$ : เวกเตอร์ซีนโดรม

$E_{1-8}$ : เวกเตอร์รูปแบบตำแหน่งบิตผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.11 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 10100010**




$R_{1-8}$ : เวกเตอร์คำรหัส (ที่รับมา)

$C_{1-8}$ : เวกเตอร์คำรหัส (ที่แก้ไขแล้ว)

**รูปที่ 4.12 ผลจากการแก้ไขบิตที่ผิดของเวกเตอร์คำรหัส 10100010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 1 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | 0000000x | 0001 |
| 2 | 000000x0 | 0010 |
| 3 | 00000x00 | 0100 |
| 4 | 0000x000 | 1000 |
| 5 | 000x0000 | 1011 |
| 6 | 00x00000 | 0111 |
| 7 | 0x000000 | 1101 |
| 8 | x0000000 | 1110 |

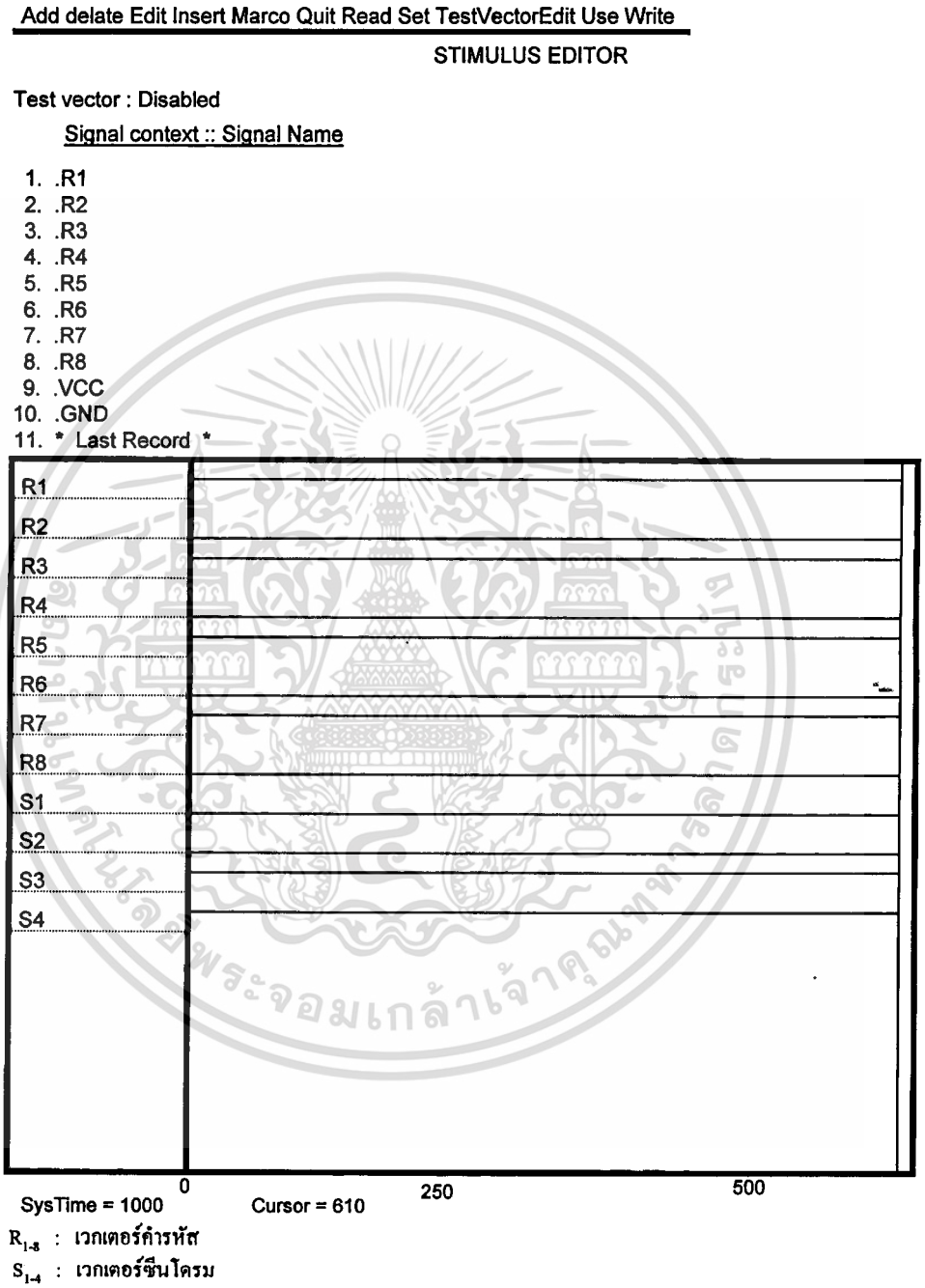
x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

**ตารางที่ 4.2 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 1 บิต**

**กรณีที่สอง** คำรหัสที่มีบิตผิด 1 บิต กำหนดให้ 10100010 เป็นเวกเตอร์คำรหัสที่ผิดจากเดิม คือ 10110010 จะทำให้ได้ค่าเวกเตอร์ซีนโดรมเป็น 1011 ดังรูปที่ 4.10 และรูปแบบบิตผิดที่เกิดขึ้น คือ 00010000 ดังรูปที่ 4.11 เมื่อวงจรถอดรหัสทำการแก้ไขแล้วจะได้ข้อมูลจากเวกเตอร์คำรหัสที่ถูกคือ 10110010 ดังรูปที่ 4.12 และรูปแบบของบิตที่ผิด 1 บิตรวมกันแล้ว จะมีอยู่ทั้งหมด 8 รูปแบบดังตารางที่ 4.2

**กรณีที่สาม** คำรหัสที่มีบิตผิดพลาด 2 บิต กำหนดให้ 10101010 เป็นเวกเตอร์คำรหัสที่ผิดจากเดิมคือ 10110010 ทำให้ได้ค่าเวกเตอร์ซีนโดรมเป็น 0011 ดังรูปที่4.13 ซึ่งค่าเวกเตอร์ซีนโดรมนี้จะทำให้เกิดรูปแบบที่ผิดพลาดได้ 3 รูปแบบ คือ 00100100, 00011000, 00000011" ทำให้ไม่สามารถพิจารณาแก้ไขให้คำรหัสที่กำหนดให้ดังกล่าวเป็นข้อมูลที่ถูกต้องได้ แต่จะสามารถที่จะแสดงได้ว่าในเวกเตอร์คำรหัสนี้มีการผิดพลาดจำนวน 2 บิต โดยจะสังเกตจากเวกเตอร์ซีนโดรมซึ่งจะมีบิตที่มีลอจิก 1 จำนวน 2 บิต ดังนั้นจะสามารถแสดงค่าเวกเตอร์ซีนโดรมของคำรหัสที่มีบิตผิดพลาด 2 บิตจำนวน 28 ชุดได้ดังตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




$R_{1-8}$ : เวกเตอร์คำรหัส

$S_{1-4}$ : เวกเตอร์ซีนโดรม

**รูปที่ 4.13 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 10101010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 1 | xx000000 | 0011 |
| 2 | x0x00000 | 1001 |
| 3 | x00x0000 | 0101 |
| 4 | x000x000 | 0110 |
| 5 | x0000x00 | 1010 |
| 6 | x00000x0 | 1100 |
| 7 | x000000x | 1111 |
| 8 | 0xx00000 | 1010 |
| 9 | 0x0x0000 | 0110 |
| 10 | 0x00x000 | 0101 |
| 11 | 0x000x00 | 1001 |
| 12 | 0x0000x0 | 1111 |
| 13 | 0x00000x | 1100 |
| 14 | 00xx0000 | 1100 |
| 15 | 00x0x000 | 1111 |
| 16 | 00x00x00 | 0011 |
| 17 | 00x000x0 | 0101 |
| 18 | 00x0000x | 0110 |
| 19 | 000xx000 | 0011 |
| 20 | 000x0x00 | 1111 |
| 21 | 000x00x0 | 1001 |
| 22 | 000x000x | 1010 |
| 23 | 0000xx00 | 1100 |

x : ตำแหน่งบิตที่ผิดมีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลง เผยแพร่ หรือจำหน่ายต่อ ในรูปแบบใดๆ ทั้งสิ้น ไม่ว่ากรณีใดๆ ที่มีการนำไปใช้

**ตารางที่ 4.3 ผลของเวกเตอร์ซินโดรมจากการกำหนดรูปแบบบิตผิด 2 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 24 | 0000x0x0 | 1010 |
| 25 | 0000x00x | 1001 |
| 26 | 00000xx0 | 0110 |
| 27 | 00000x0x | 0101 |
| 28 | 000000xx | 0011 |

**ตารางที่ 4.3 (ต่อ)**

**กรณีที่สี่** คำรหัสที่มีบิตผิด 3 บิต กำหนดให้เป็น 10011110 จากเวกเตอร์คำรหัสเดิม 10110010 ซึ่งจะได้เวกเตอร์ซีนโดรมเป็น 1101 และจากค่าเวกเตอร์ซีนโดรมนี้จะทำให้รูปแบบของบิตผิดเกิดการซ้ำกับในรูปแบบอื่นๆ และซ้ำกันกับค่าเวกเตอร์ซีนโดรมในกรณีสอง คือ เวกเตอร์คำรหัสมีบิตผิด 1 บิต ซึ่งจะทำให้ไม่สามารถตรวจสอบ และแก้ไขเวกเตอร์คำรหัสที่มีบิตผิดพลาด 3 บิตนี้ได้เลย โดยรูปแบบของเวกเตอร์คำรหัสที่มีบิตผิด 3 บิต จะมีอยู่ทั้งหมด 56 รูปแบบ ดังตารางที่ 4.4

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xxx00000 | 0100 |
| 2 | xx0x0000 | 1000 |
| 3 | xx00x000 | 1011 |
| 4 | xx000x00 | 0111 |
| 5 | xx0000x0 | 0001 |
| 6 | xx00000x | 0010 |

x : ตำแหน่งบิตที่ผิดมีค่าเป็น 0 หรือ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.4 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตผิด 3 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 7 | x0xx0000 | 0010 |
| 8 | x0x0x000 | 0001 |
| 9 | x0x00x00 | 1101 |
| 10 | x0x000x0 | 1011 |
| 11 | x0x0000x | 1000 |
| 12 | x00xx000 | 1101 |
| 13 | x00x0x00 | 0001 |
| 14 | x00x00x0 | 0111 |
| 15 | x00x000x | 0100 |
| 16 | x000xx00 | 0010 |
| 17 | x000x0x0 | 0100 |
| 18 | x000x00x | 0111 |
| 19 | x0000xx0 | 1000 |
| 20 | x0000x0x | 1011 |
| 21 | x00000xx | 1101 |
| 22 | 0xxx0000 | 0001 |
| 23 | 0xx0x000 | 0010 |
| 24 | 0xx00x00 | 1110 |
| 25 | 0xx000x0 | 1000 |
| 26 | 0xx0000x | 1011 |
| 27 | 0x0xx000 | 1110 |
| 28 | 0x0x0x00 | 0010 |
| 29 | 0x0x00x0 | 0100 |
| 30 | 0x0x000x | 0111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

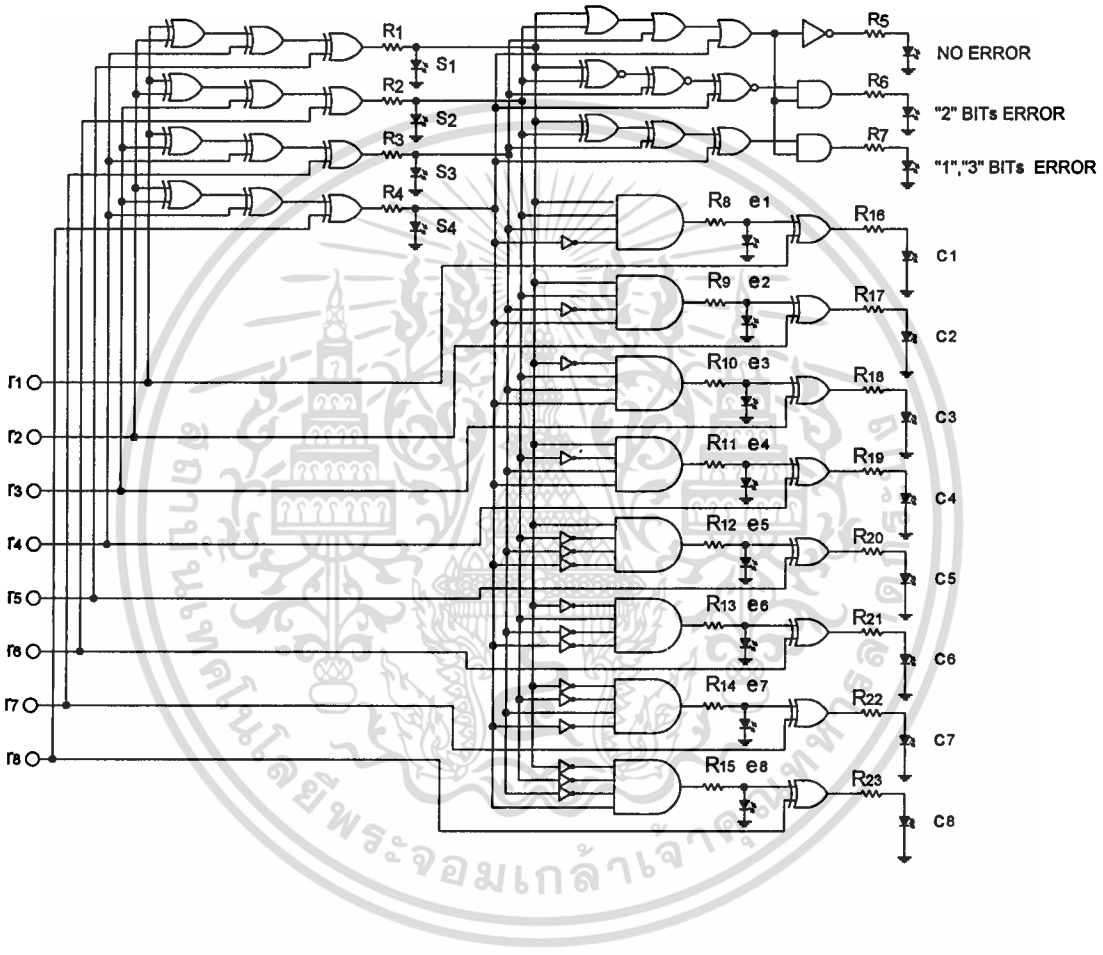
**ตารางที่ 4.4 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 31 | 0x00xx00 | 0001 |
| 32 | 0x00x0x0 | 0111 |
| 33 | 0x00x00x | 0100 |
| 34 | 0x000xx0 | 1011 |
| 35 | 0x000x0x | 1000 |
| 36 | 0x0000xx | 1110 |
| 37 | 00xxx000 | 0100 |
| 38 | 00xx0x00 | 1000 |
| 39 | 00xx00x0 | 1110 |
| 40 | 00xx000x | 1101 |
| 41 | 00x0xx00 | 1011 |
| 42 | 00x0x0x0 | 1101 |
| 43 | 00x0x00x | 1110 |
| 44 | 00x00xx0 | 0001 |
| 45 | 00x00x0x | 0010 |
| 46 | 00x000xx | 0100 |
| 47 | 000xxx00 | 0111 |
| 48 | 000xx0x0 | 0001 |
| 49 | 000xx00x | 0010 |
| 50 | 000x0xx0 | 1101 |
| 51 | 000x0x0x | 1110 |
| 52 | 000x00xx | 1000 |
| 53 | 0000xxx0 | 1110 |
| 54 | 0000xx0x | 1101 |
| 55 | 0000x0xx | 1011 |
| 56 | 00000xxx | 0111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

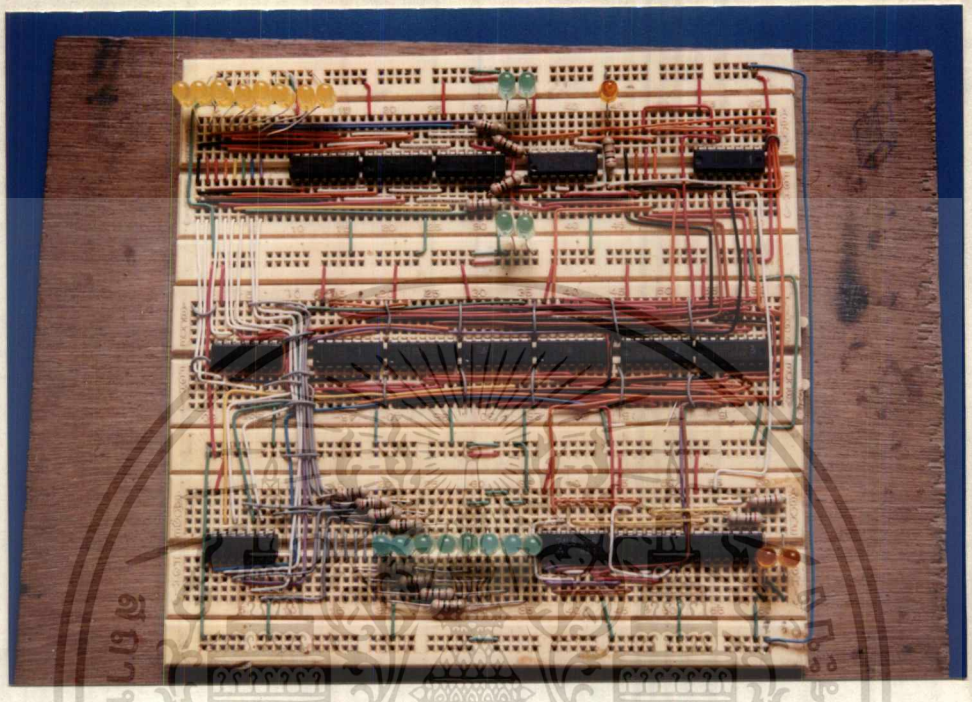
**ตารางที่ 4.4 (ต่อ)**

**การทดลองทางด้านฮาร์ดแวร์** เป็นการทดลองด้วยการนำเอาวงจรชุดถอดรหัส ดังรูปที่ 4.14 มาสร้างเป็นชุดวงจรลอจิกเกตบนแผงประกอบวงจร ดังรูปที่ 4.15 จากนั้นทำการทดลองในแต่ละกรณีดังนี้




**รูปที่ 4.14 วงจรถอดรหัสระบบเชิงเส้นขนาด (8,4) บิต**

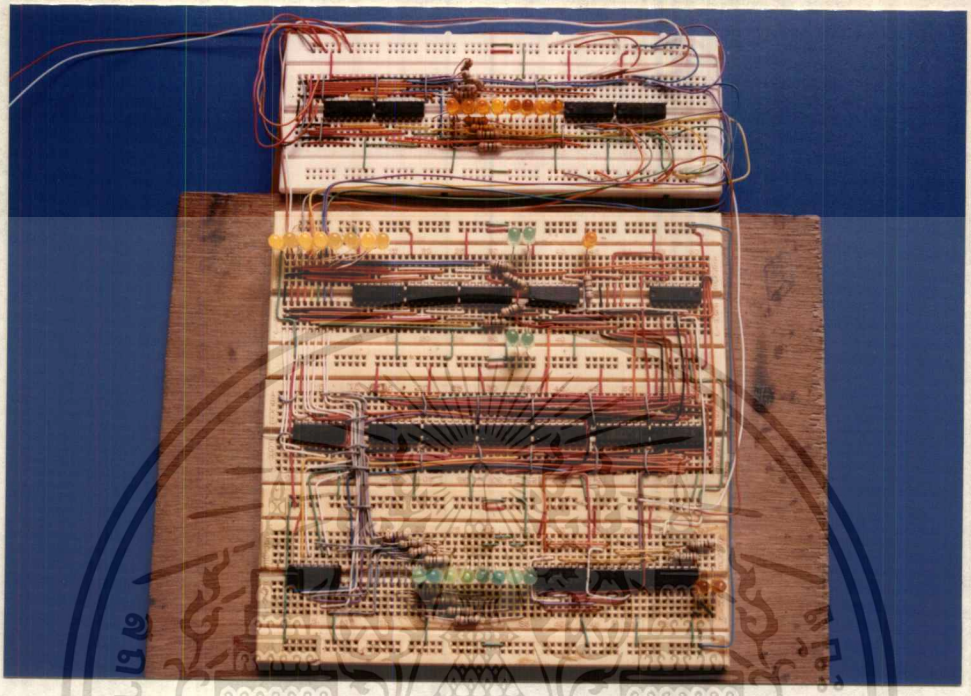
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.15 ภาพวงจรถอดรหัสที่ประกอบบนแผงประกอบวงจร**

**กรณีแรก** คำรหัสที่ไม่มีบิตผิด กำหนดเป็น 10110010 จะเห็นได้ว่าตัวแอลอีดีสีแดง (LED) ของชุดเวกเตอร์ซีนโดรมจะแสดงค่าเป็น 0000 ( ทุกหลอดไม่สว่าง ) และรูปแบบที่ผิดที่มีตัวแอลอีดีแสดงผลนั้นก็จะไม่สว่างเช่นกัน และที่ชุดแก้ไขบิตผิดพลาดนั้นจะแสดงค่าของข้อมูลเป็น 10110000 ดังรูปที่ 4.16

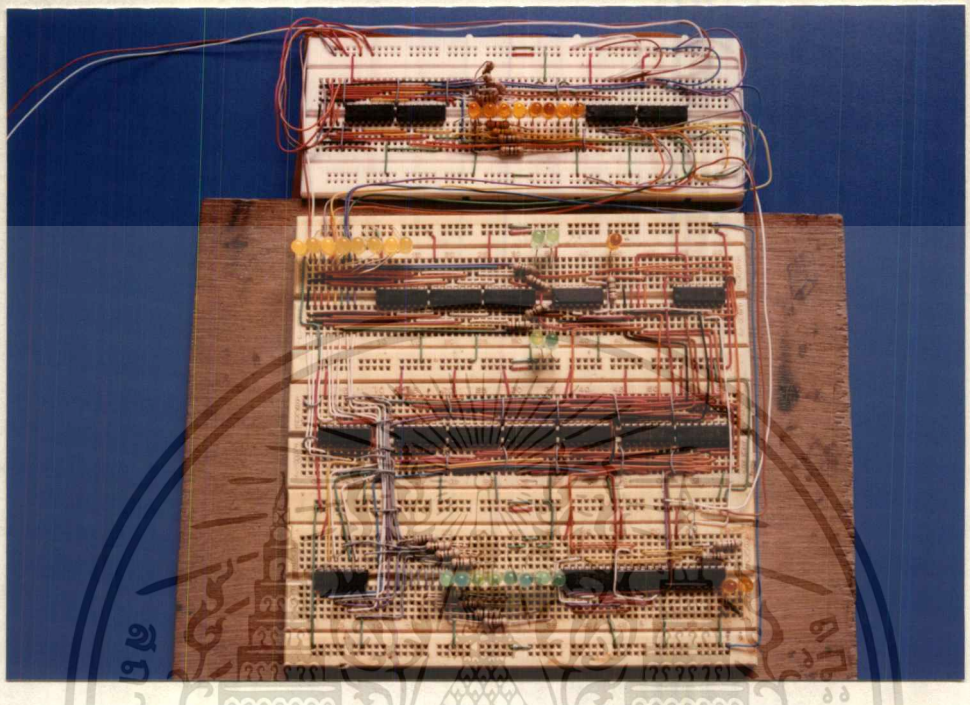
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.16 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10110010**

**กรณีที่สอง** คำรหัสที่มีบิตผิด 1 บิต โดยกำหนดให้ 10100010 เป็นเวกเตอร์คำรหัสเมื่อป้อนเข้าที่วงจรชุดถอดรหัส จะทำให้ได้ค่าซินโดรมเวกเตอร์ 1011 ( สังเกตจากแอลอีดี ) ซึ่งจะได้รหัสเวกเตอร์ของตำแหน่งผิดเป็น 00010000 และผลจากการแก้ไขบิตที่ผิด จะทำให้ได้เวกเตอร์ข้อมูลเป็น 10110010 ดังรูปที่ 4.17

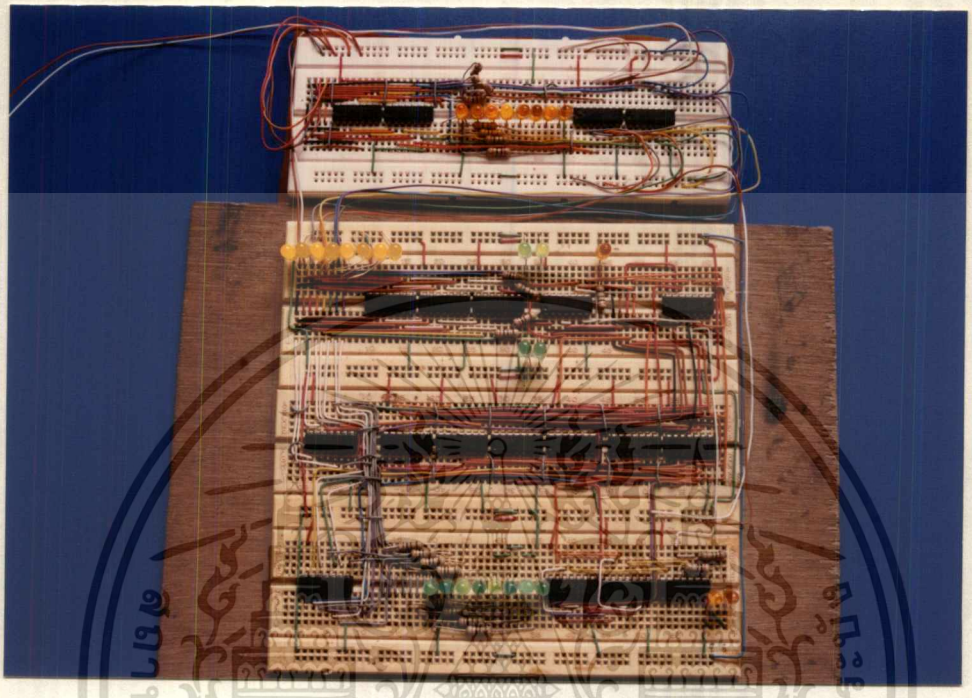
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.17 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10100010**

**กรณีที่สาม** คำรหัสที่มีบิตผิด 2 บิต กำหนดให้ 10101010 เป็นเวกเตอร์คำรหัสที่ตัวถอดรหัสรับเข้ามาผ่านวงจรถอดรหัสแล้วจะได้ค่าเวกเตอร์ซีนโดรมเป็น 0011 ซึ่งค่าเวกเตอร์ซีนโดรมจะมีการซ้ำกันอยู่ภายในรูปแบบของบิตที่ผิด 2 บิต อันจะทำให้ค่าผลจากการแก้ไขเป็น 10101010 แต่ที่ชุดวงจรถอดรหัสสามารถแจ้งได้ว่า เวกเตอร์คำรหัสนี้มีบิตผิดพลาดจำนวน 2 บิต ดังแอลอีดีสีเขียวที่สว่างอยู่ ดังรูปที่ 4.18

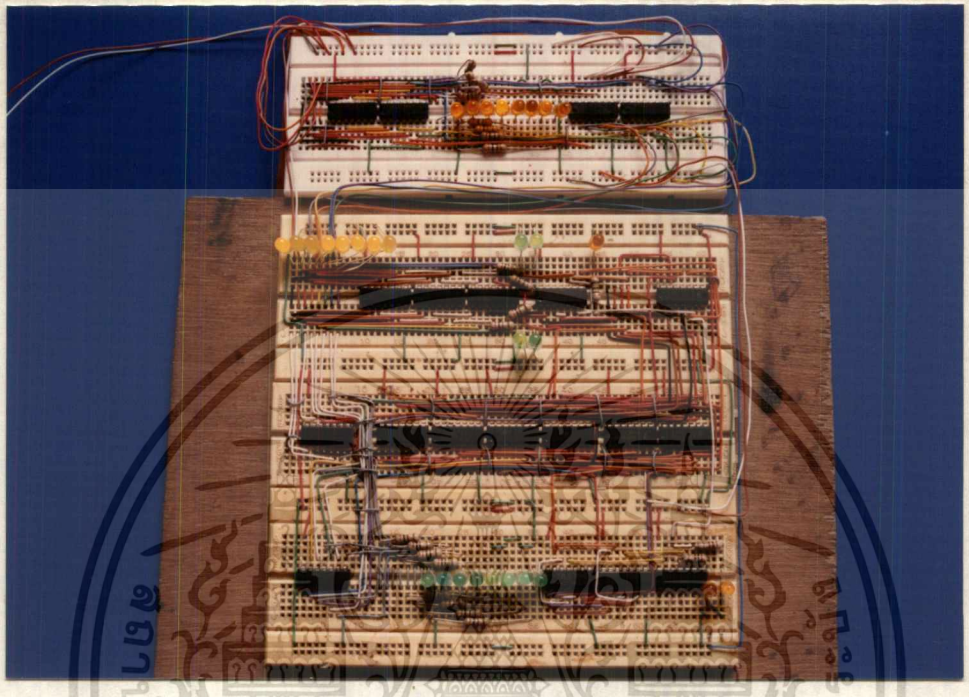
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 4.18 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10101010**

**กรณีที่สี่** คำรหัสที่มีบิตผิด 3 บิต กำหนดเวกเตอร์คำรหัสเป็น 10011110 เมื่อทำการป้อนเข้าวงจรชุดถอดรหัสแล้วจะได้ค่าเวกเตอร์ซินโดรมเป็น 1011 ซึ่งค่าเวกเตอร์ซินโดรมนี้จะซ้ำกันเองในรูปแบบการผิด 3 บิต และจะซ้ำกับเวกเตอร์ซินโดรมที่มีรูปแบบผิด 1 บิต จึงทำให้ไม่สามารถแก้ไขบิตที่ผิดพลาดได้เลย และผลของเวกเตอร์ข้อมูลที่ได้จะมีค่าเป็น 10001110 ดังรูปที่ 4.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



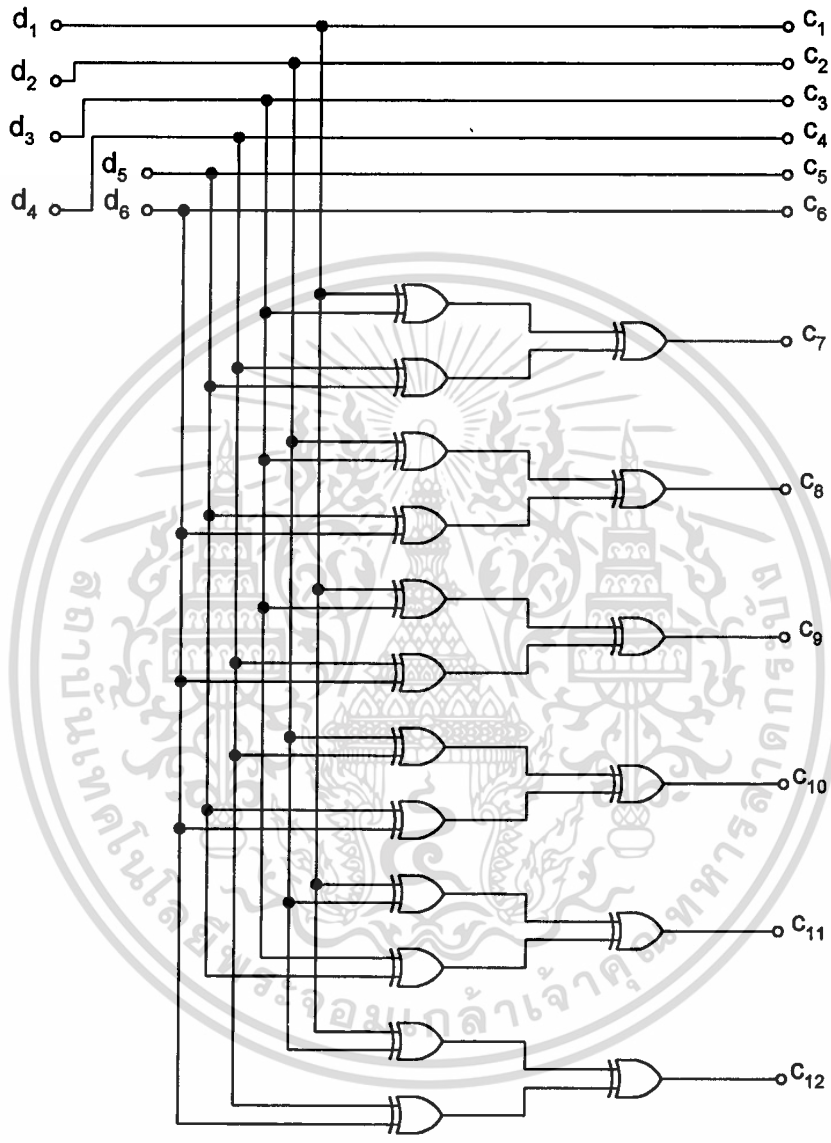
**รูปที่ 4.19 ภาพวงจรถอดรหัสที่ได้รับเวกเตอร์คำรหัส 10011110**

### 4.1.3 การทดลองวงจรชุดเข้ารหัสและถอดรหัสระบบเชิงเส้นขนาด (12,4) บิต

การทดลองวงจรในส่วนนี้จะเป็นการทดลองทางด้านซอฟต์แวร์เท่านั้น ซึ่งวงจรในส่วนนี้ได้รับการพัฒนาขึ้นเพื่อให้สามารถทำการแก้ไขข้อมูลได้มากขึ้น ในการทดลองจะแบ่งออกเป็น

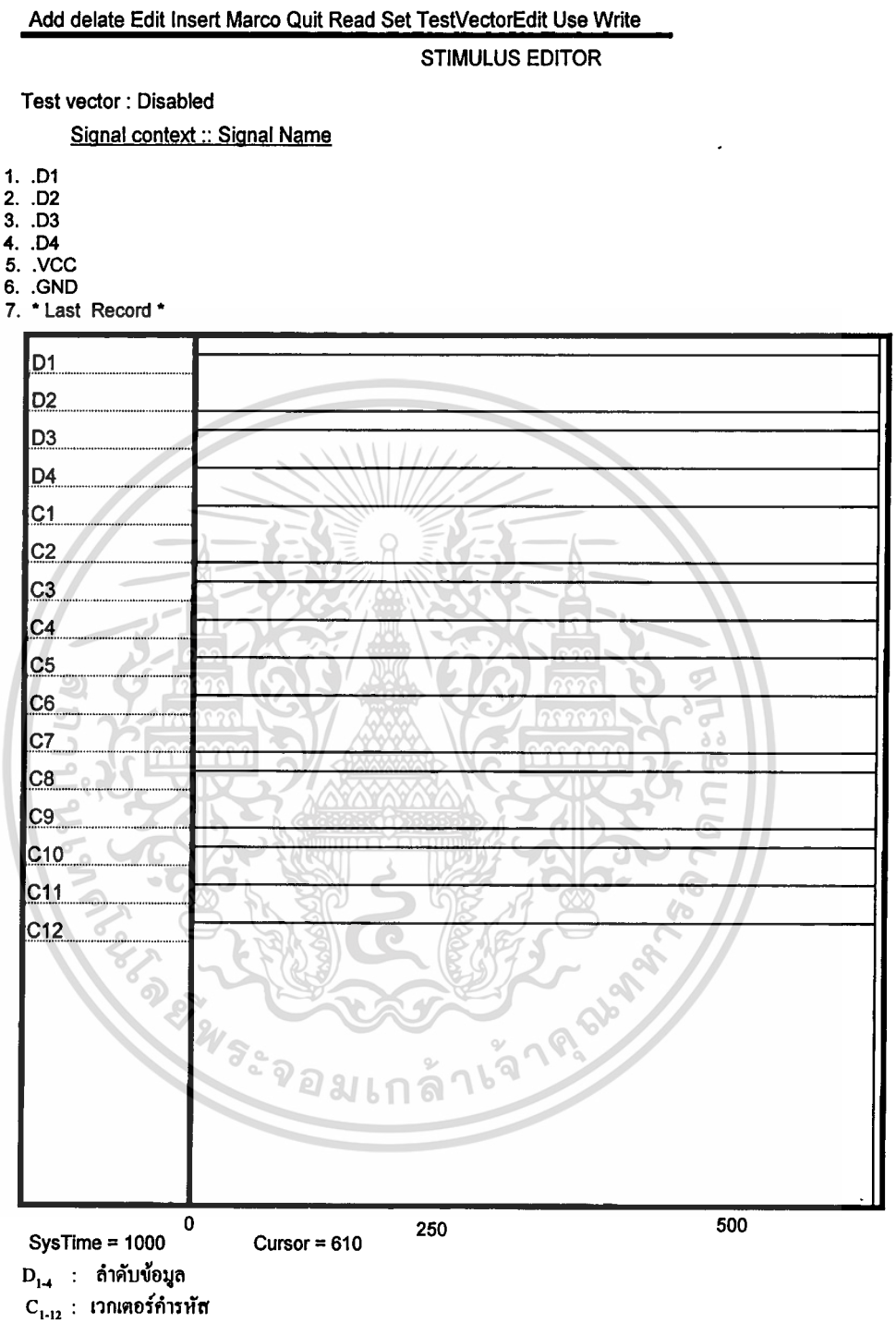
**การทดลองชุดวงจรเข้ารหัส** ชุดวงจรเข้ารหัสนั้นจะทำการทดลองด้วย การจำลองการทำงานของโปรแกรมสำเร็จรูปออร์แคด และจากรูปที่ 4.20 เป็นวงจรเข้ารหัส ซึ่งเมื่อทำการป้อนข้อมูล 1011 เข้าไปโดยข้อมูลใน 2 บิตหลังจะเป็นบิตข้อมูลที่ฝากไปเพื่อช่วยในการสร้างรหัสแก้ไข และผลของการเข้ารหัสจะได้คำรหัสเป็น 101111010111 ดังรูปที่ 4.21 และผลของการเข้ารหัสข้อมูลขนาด 4 บิต จะมีคำรหัสทั้งหมด 16 รูปแบบ ดังตารางที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.20 วงจรเข้ารหัสขนาด (12,4) บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.21 ผลการจำลองการทำงานด้วยข้อมูล 1011**

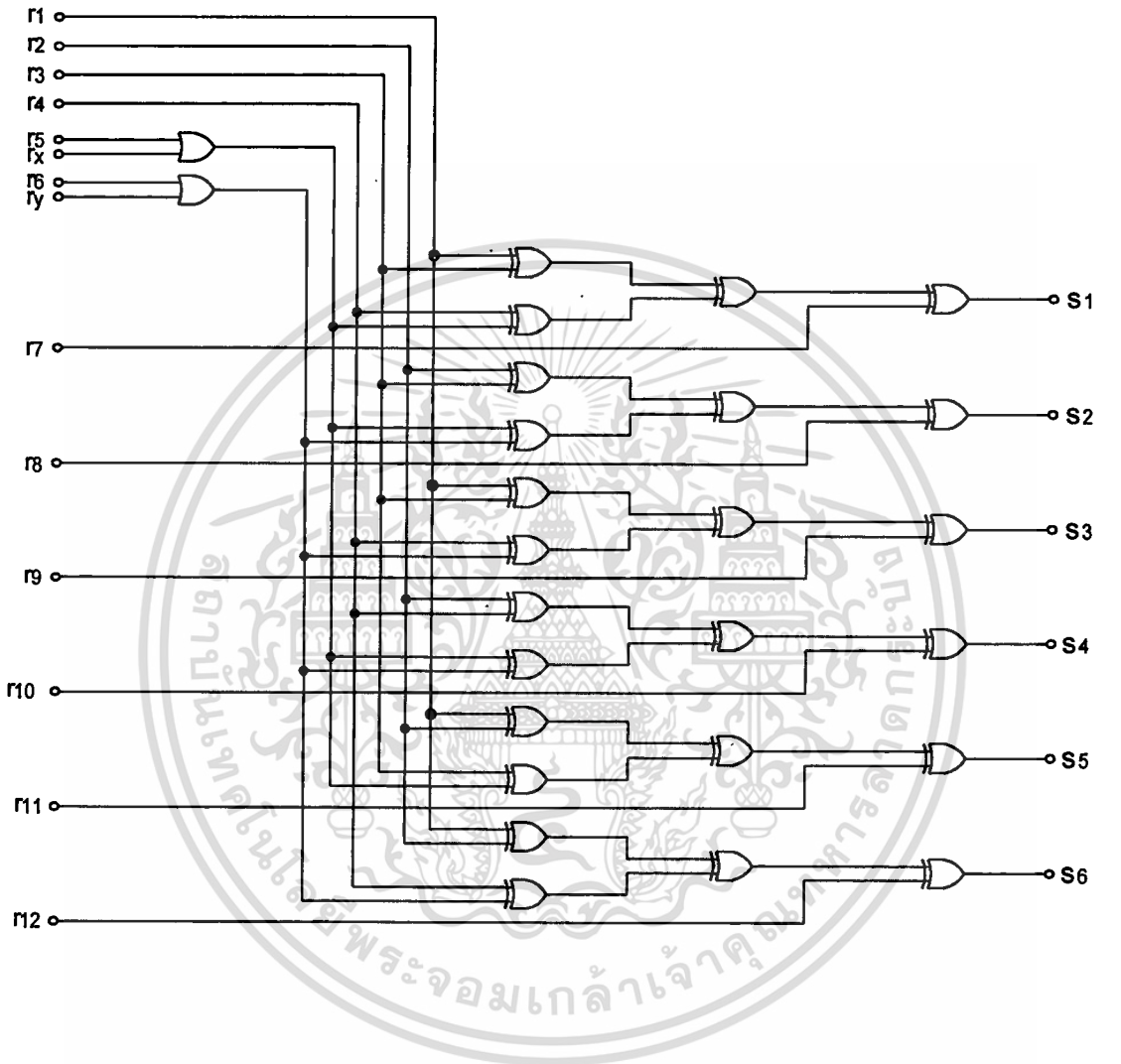
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | ลำดับข้อมูล | ลำดับคำรหัส |
|---|---|---|
| 1 | 000011 | 000011101011 |
| 2 | 000111 | 000111000110 |
| 3 | 001011 | 001011010001 |
| 4 | 001111 | 001111111100 |
| 5 | 010011 | 010011111100 |
| 6 | 010111 | 010111010001 |
| 7 | 011011 | 011011000110 |
| 8 | 011111 | 011111101011 |
| 9 | 100011 | 100011000000 |
| 10 | 100111 | 100111101101 |
| 11 | 101011 | 101011111010 |
| 12 | 101111 | 101111010111 |
| 13 | 110011 | 110011010111 |
| 14 | 110111 | 110111111010 |
| 15 | 111011 | 111011101101 |
| 16 | 111111 | 111111000000 |

**ตารางที่ 4.5 ผลของเวกเตอร์คำรหัสที่ได้จากการจำลองการทำงานของข้อมูลทั้ง 16 รูปแบบ**

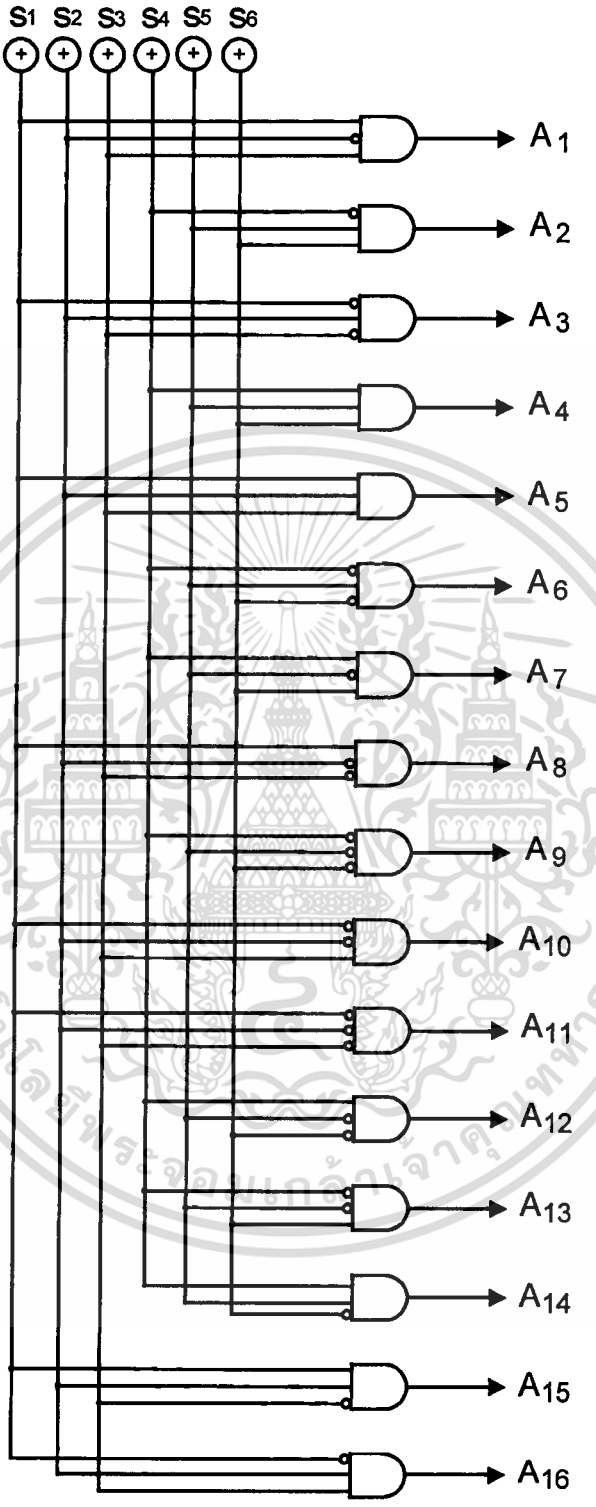
**การทดลองชุดวงจรถอดรหัส** วงจรถอดรหัสจะมีความซับซ้อนมากขึ้น ดังรูปที่ 4.22 โดยในการจำลองการทำงานของวงจรจะแบ่งออกเป็น 5 กรณี คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



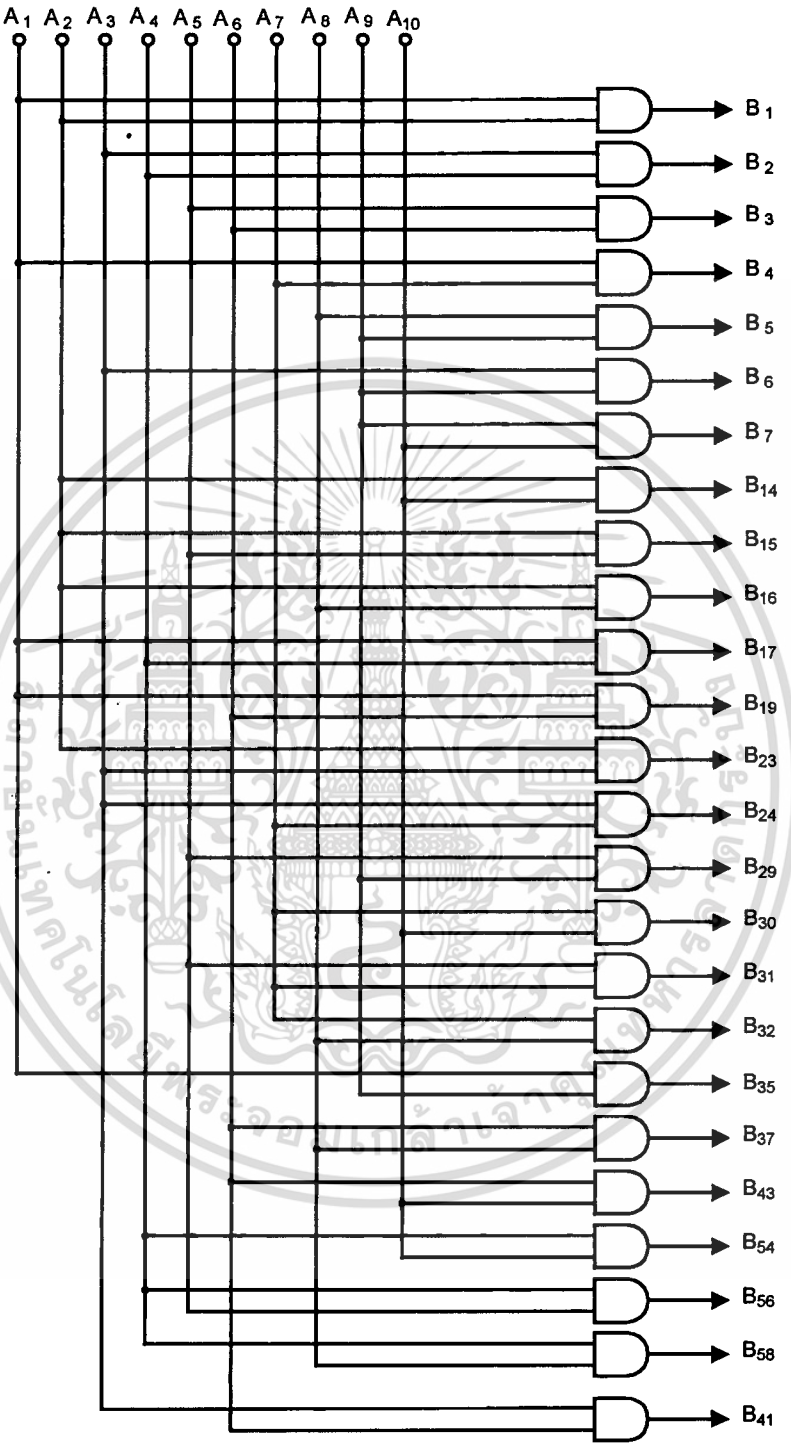

**รูปที่ 4.22 วงจรถอดรหัสขนาด (12,4) บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



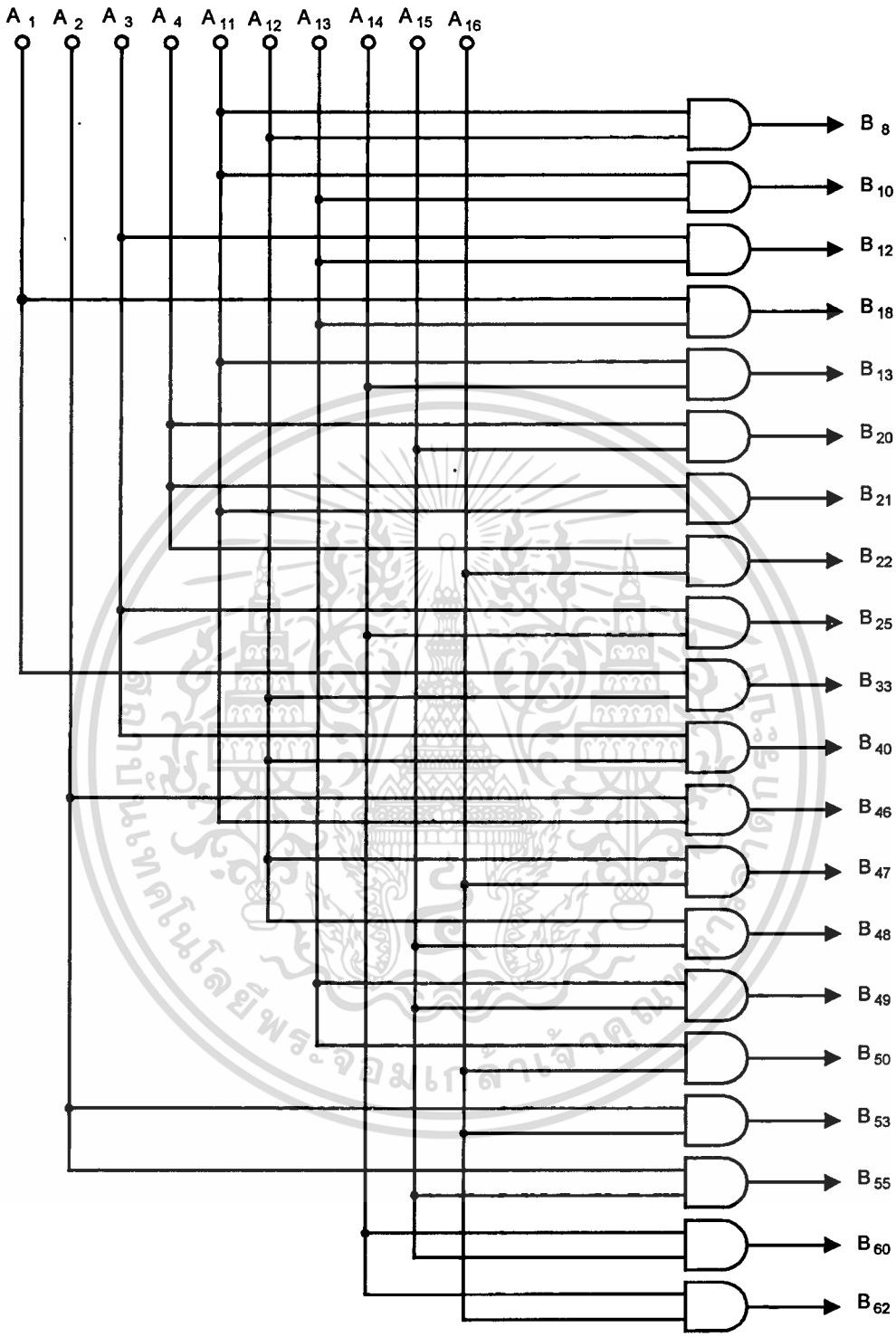

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



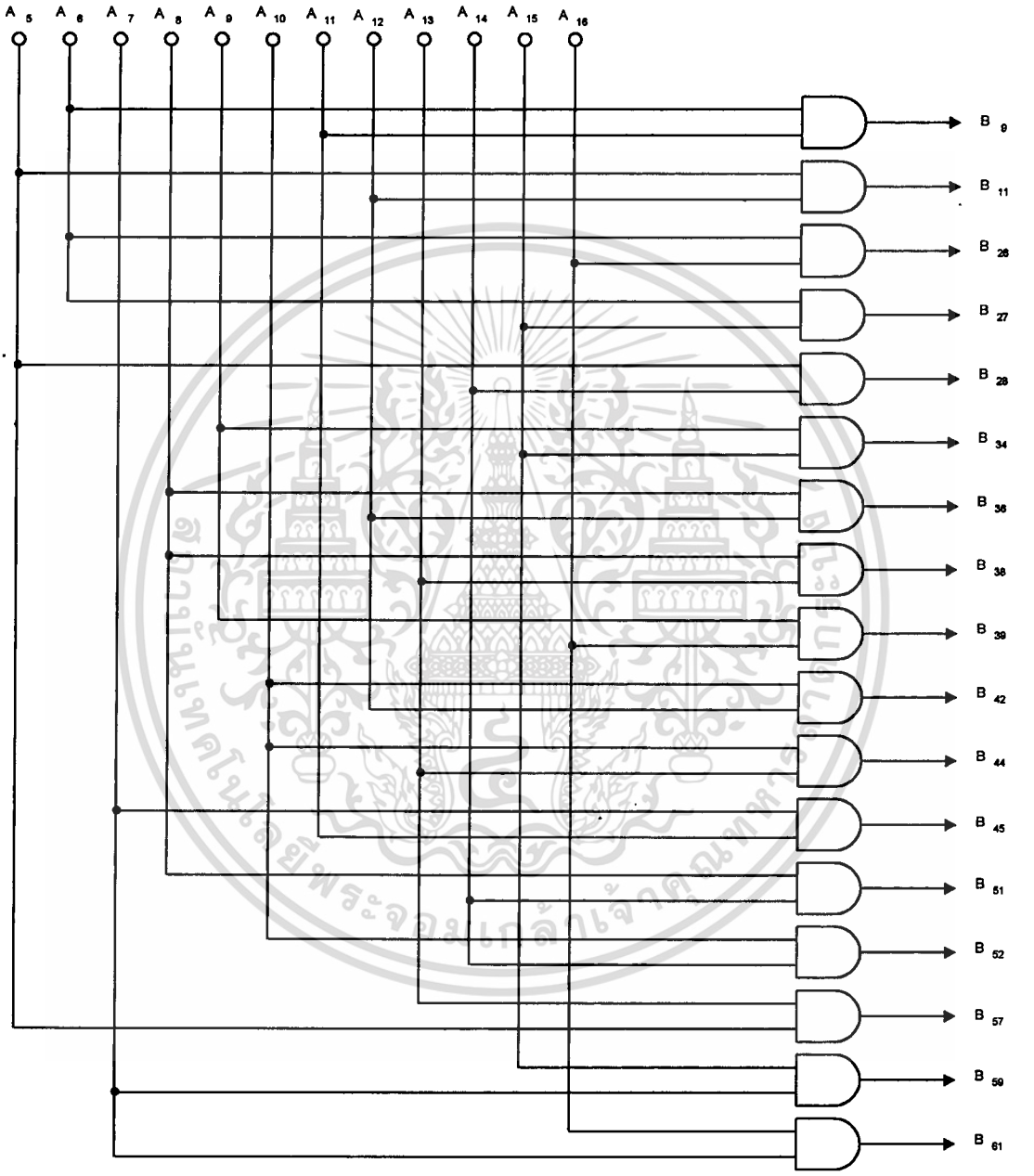

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



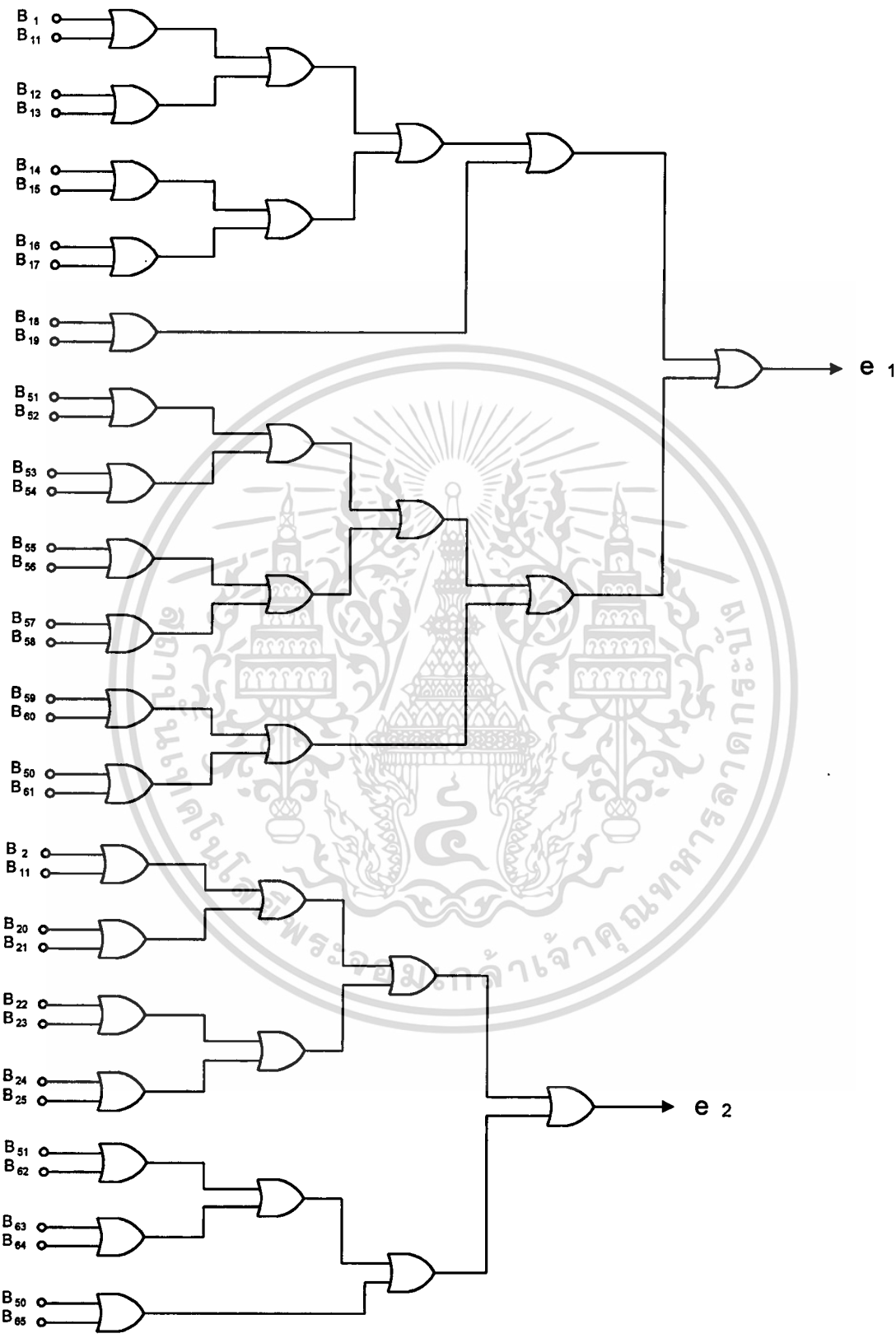

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



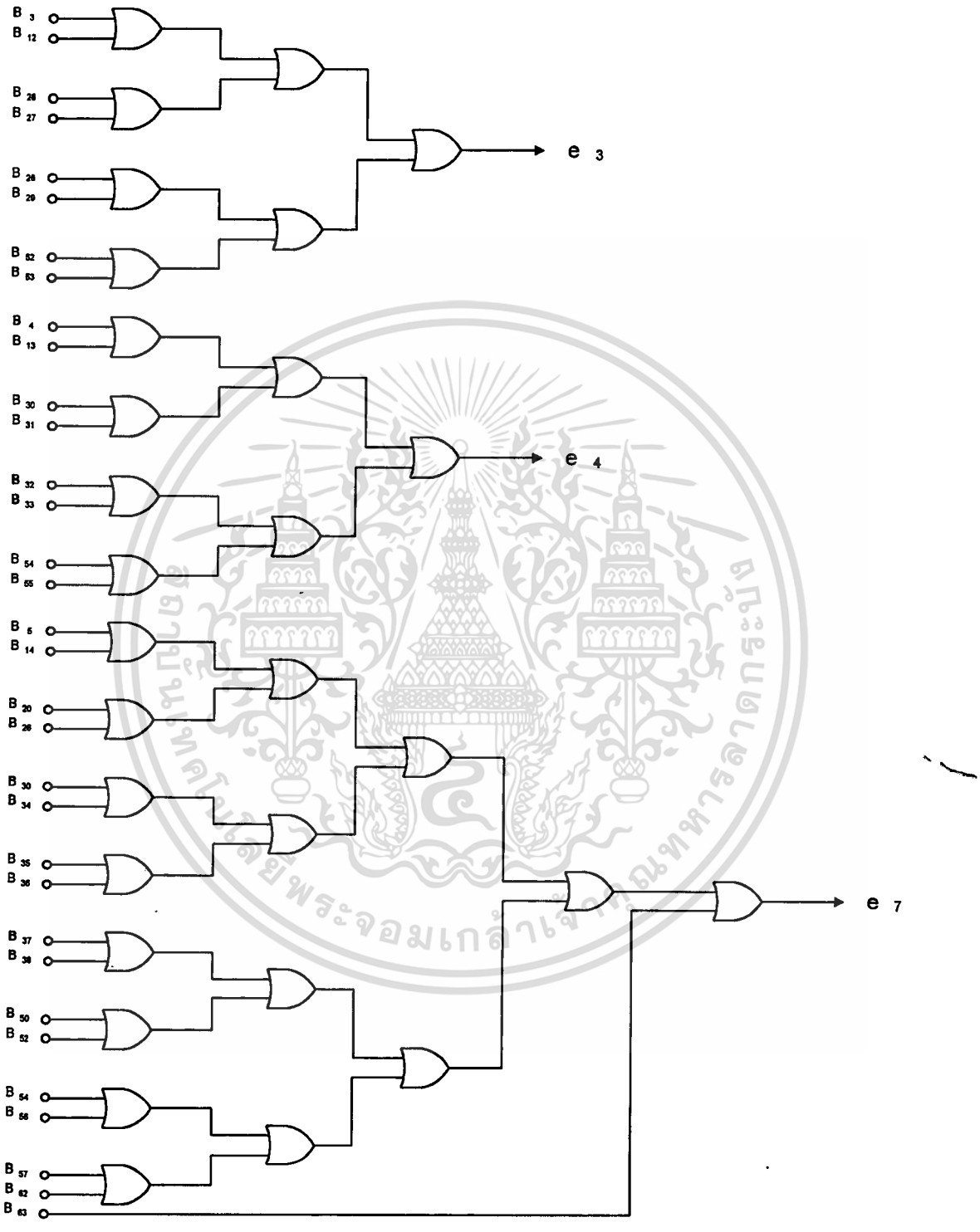

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



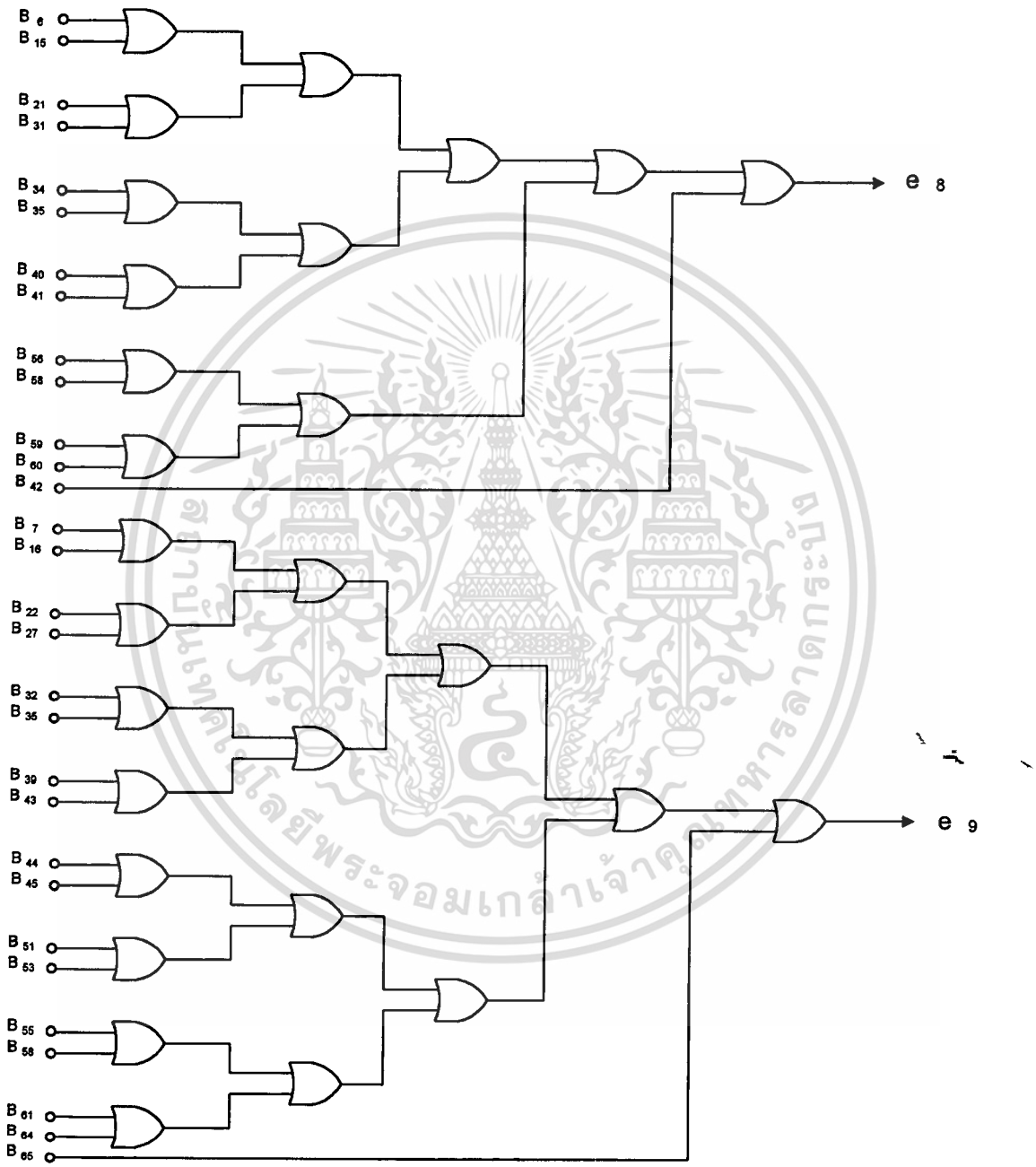

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.22 (ต่อ)**



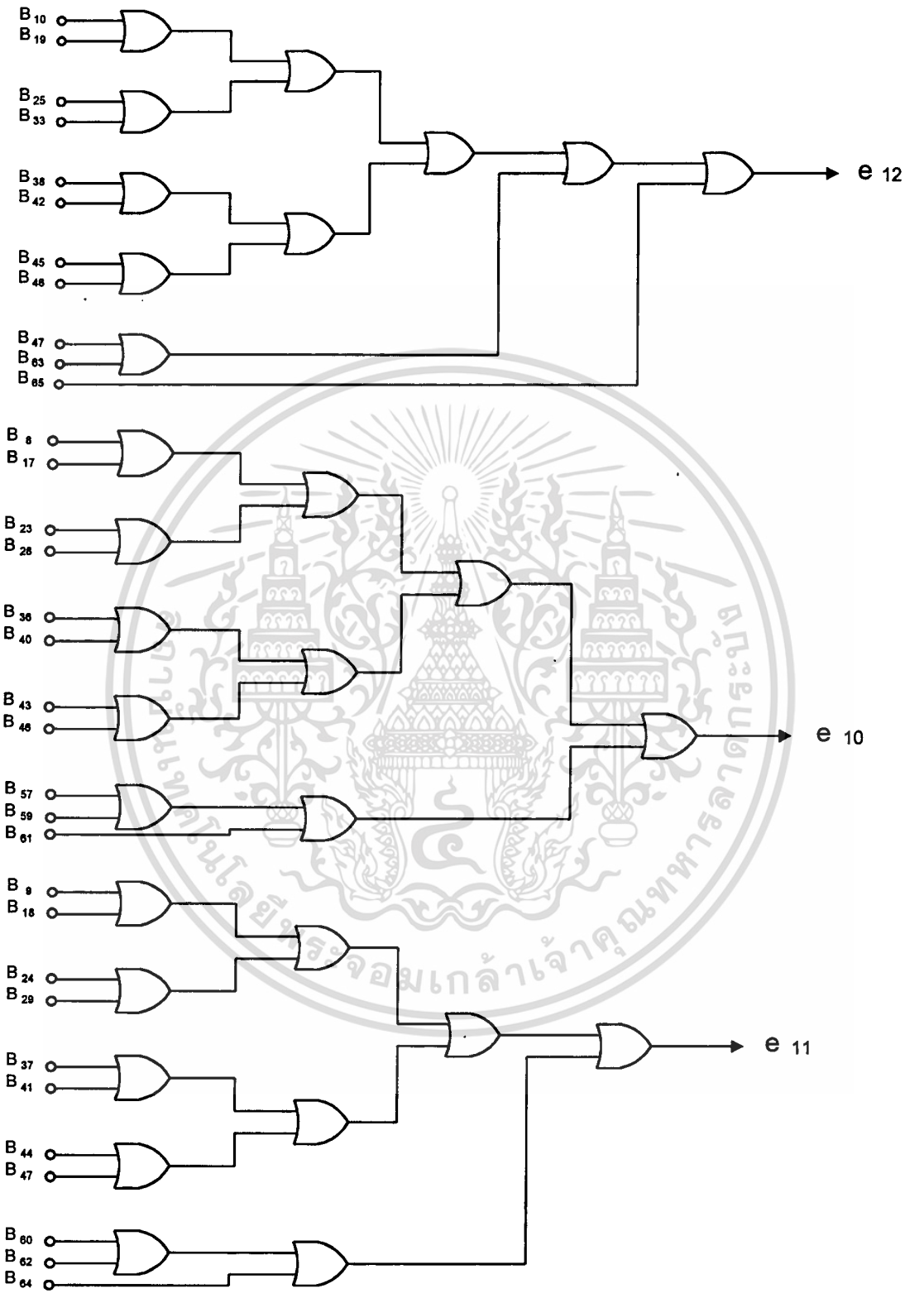

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



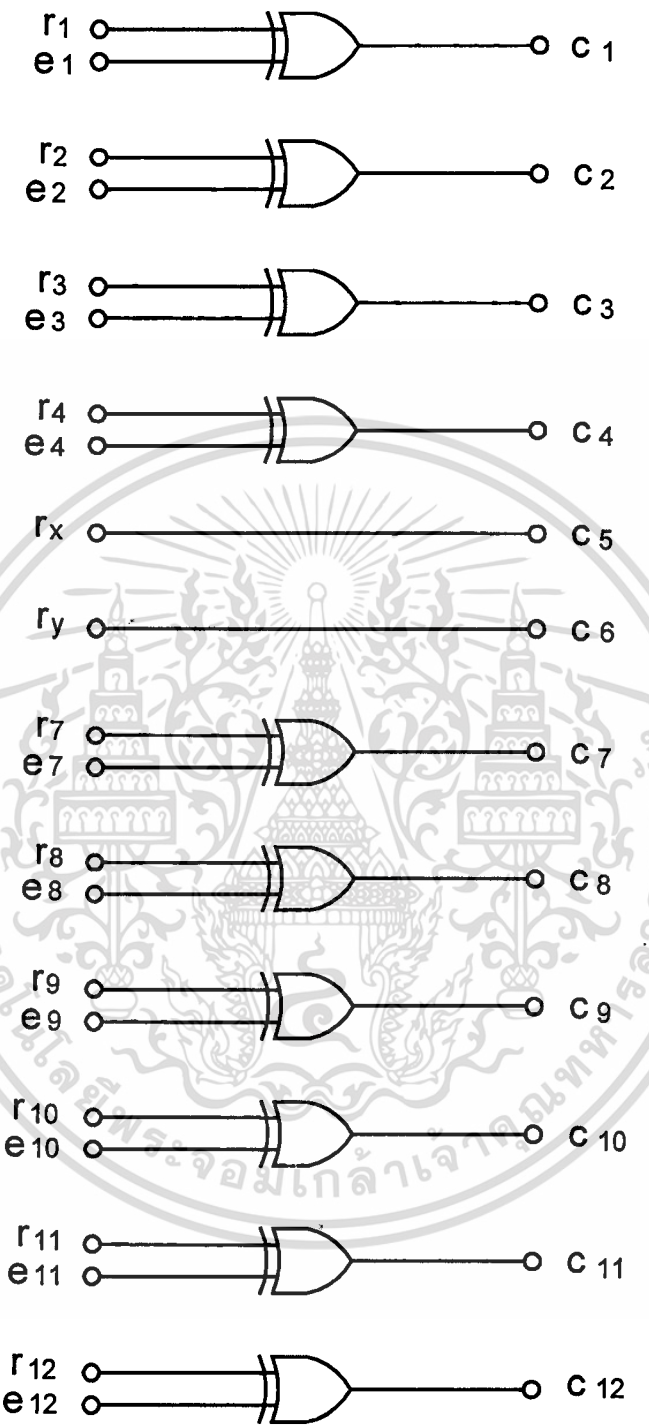

**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.22 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

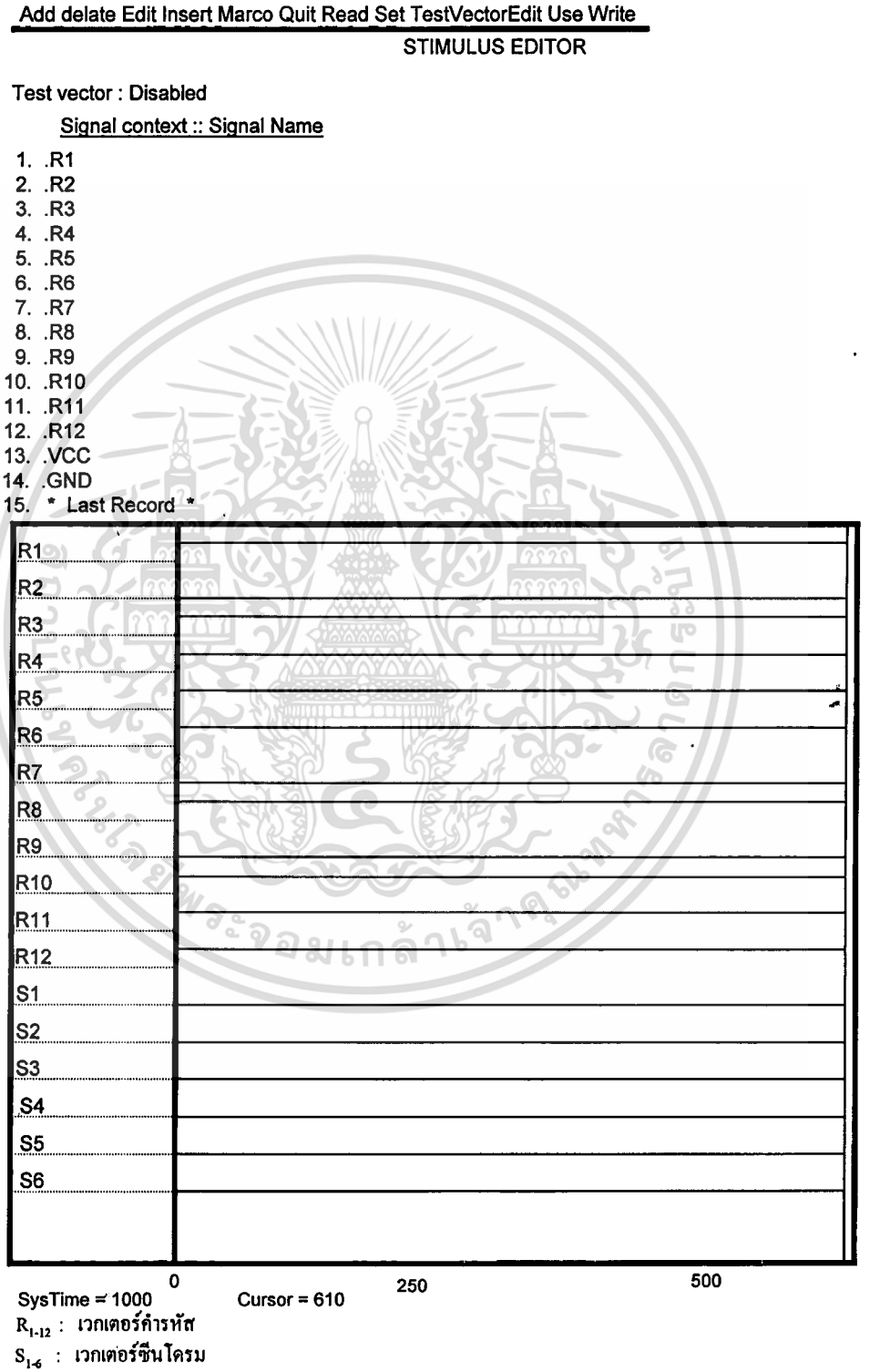



**รูปที่ 4.22 (ต่อ)**

**กรณีแรก** คำรหัสที่ ไม่มีบิตผิด เมื่อทำการป้อนเวกเตอร์คำรหัสที่ไม่มีบิตผิด 101111010111 จะได้ค่าซีนโดรมเป็น 000000 ดังรูปที่ 4.23 และรูปแบบของบิตที่ผิดจะเป็น 000000000000

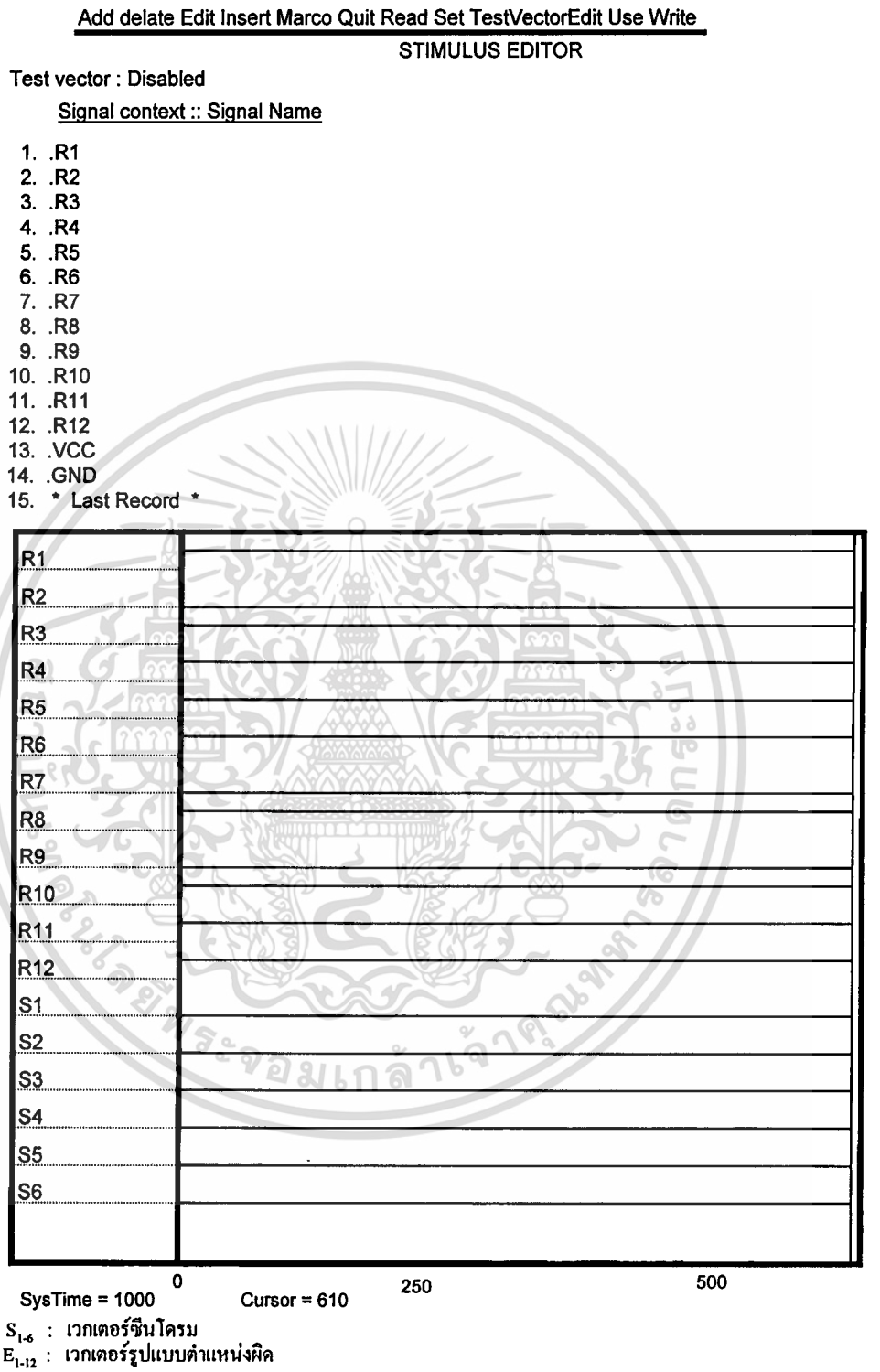
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังรูปที่ 4.24 ซึ่งจะทำให้ผลของการแก้ไขเวกเตอร์คำรหัสออกมาเป็นข้อมูล 101111010111 ดังรูปที่ 4.25




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

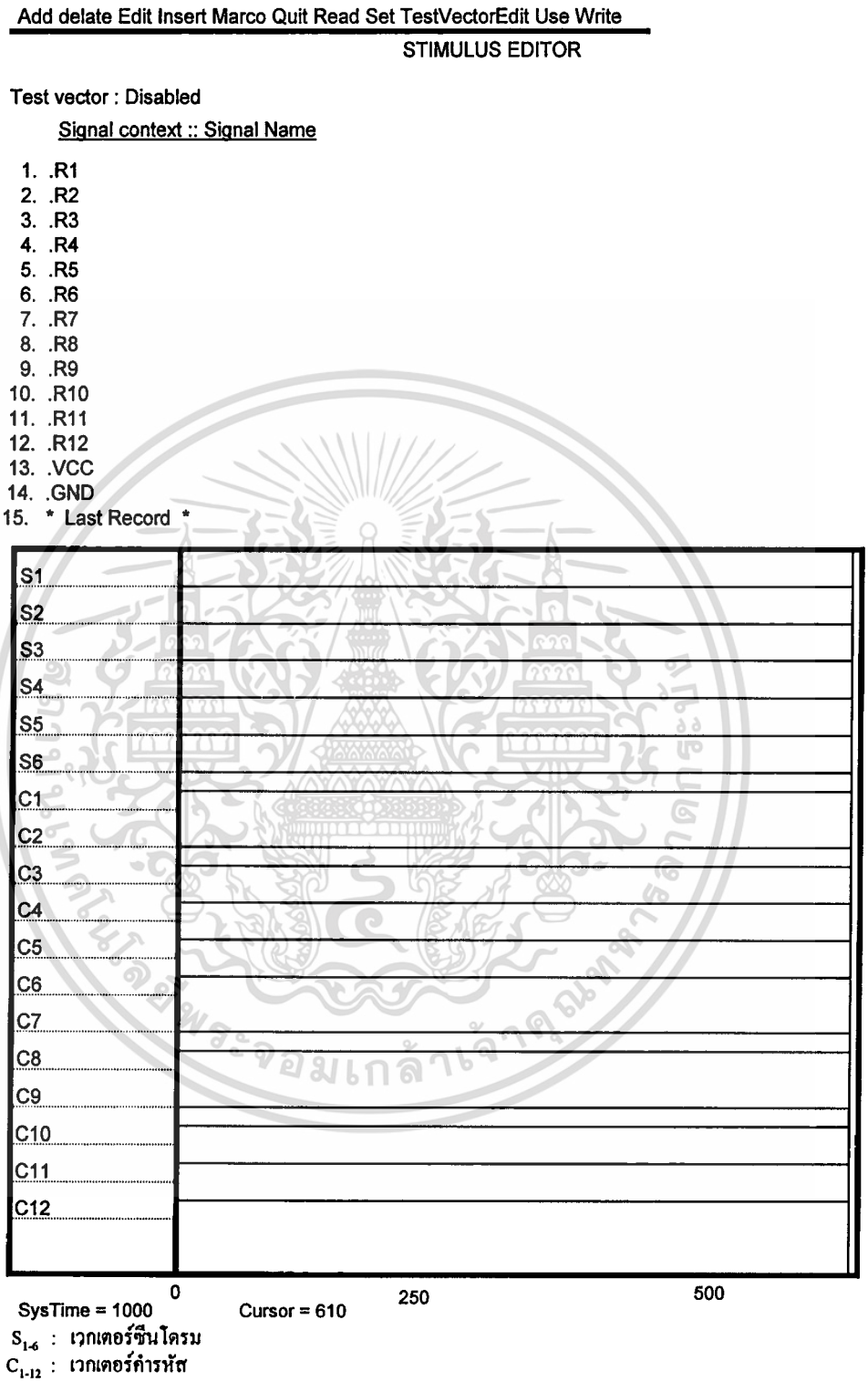
**รูปที่ 4.23 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 101111010111**




$S_{1\text{-}6}$ : เวกเตอร์ซีนโครม
$E_{1\text{-}12}$ : เวกเตอร์รูปแบบตำแหน่งผิด

**รูปที่ 4.24 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 101111010111**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

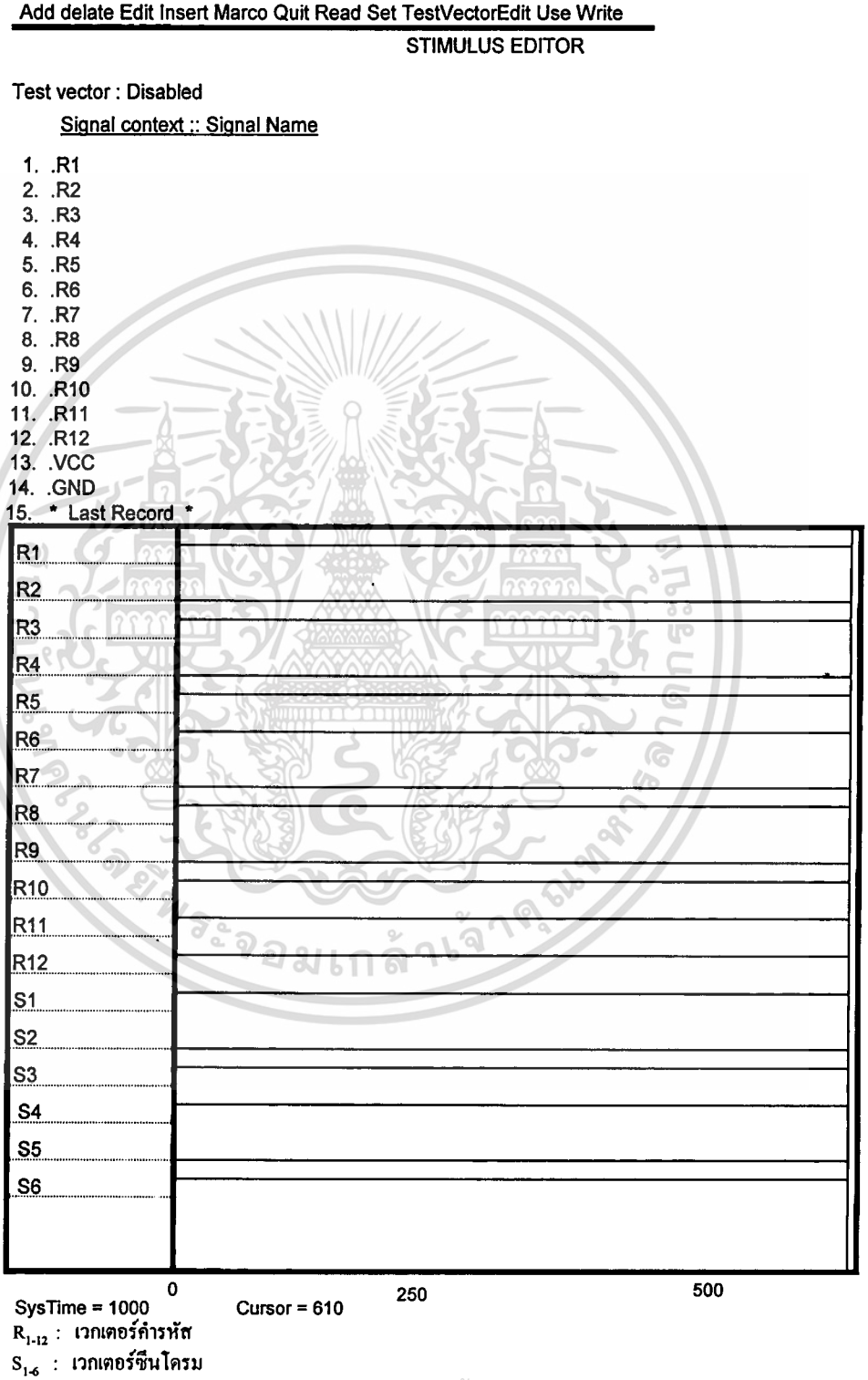



**รูปที่ 4.25 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 101111010111**

**กรณีที่สอง** คำรหัสที่มีบิตผิด 1 บิต กำหนดให้เวกเตอร์ข้อมูลเป็น 101011010111 โดยกำหนดให้ผิดบิตที่ 4 เมื่อส่งเข้าชุดถอดรหัสจะได้ค่าเวกเตอร์ซีนโดรมเป็น 101101 ดังรูปที่ 4.26 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีรูปแบบที่ผิดเป็น 000100000000 ดังรูปที่ 4.27 ผลที่ได้จากการแก้ไขได้เป็นข้อมูล 101111010111 ดังรูปที่ 4.28




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.26 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 101011010111**

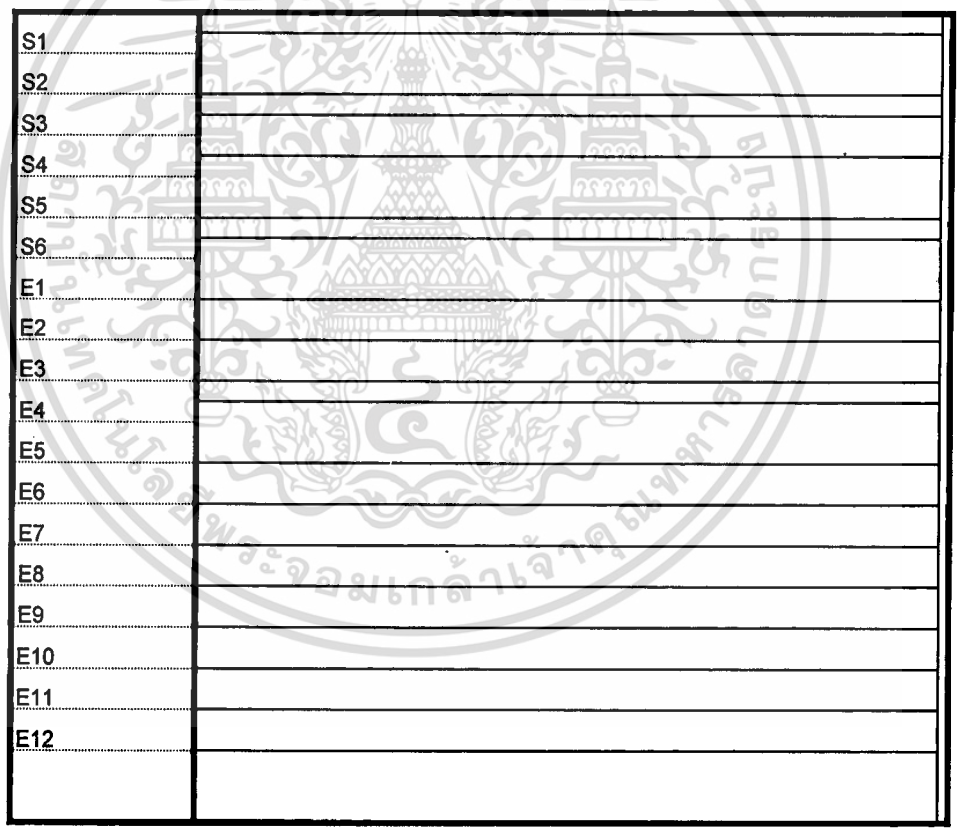
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




$S_{1-6}$ : เวกเตอร์ซีนโครม

$E_{1-12}$ : เวกเตอร์รูปแบบตำแหน่งผิด

**รูปที่ 4.27 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 101011010111**

เอกสารนี้เป็นเอกสาร ... ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

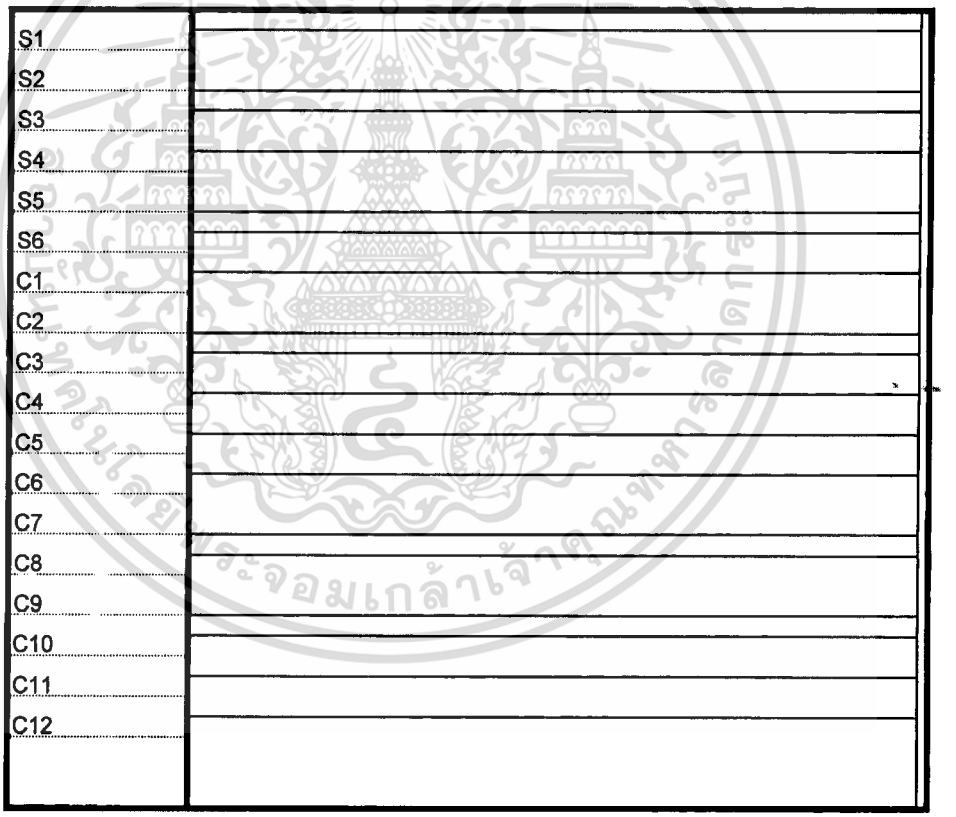
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




SysTime = 1000 Cursor = 610

$S_{1-6}$ : เวกเตอร์ซีนโครม

$C_{1-12}$ : เวกเตอร์คำรหัส

**รูปที่ 4.28 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 101011010111**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฉะนั้นรูปแบบบิตที่ผิดของคำรหัส 1 บิตจะมีอยู่ 8 รูปแบบ จะได้ค่าเวกเตอร์ซีนโดรมที่แตกต่างกัน ดังตารางที่ 4.6

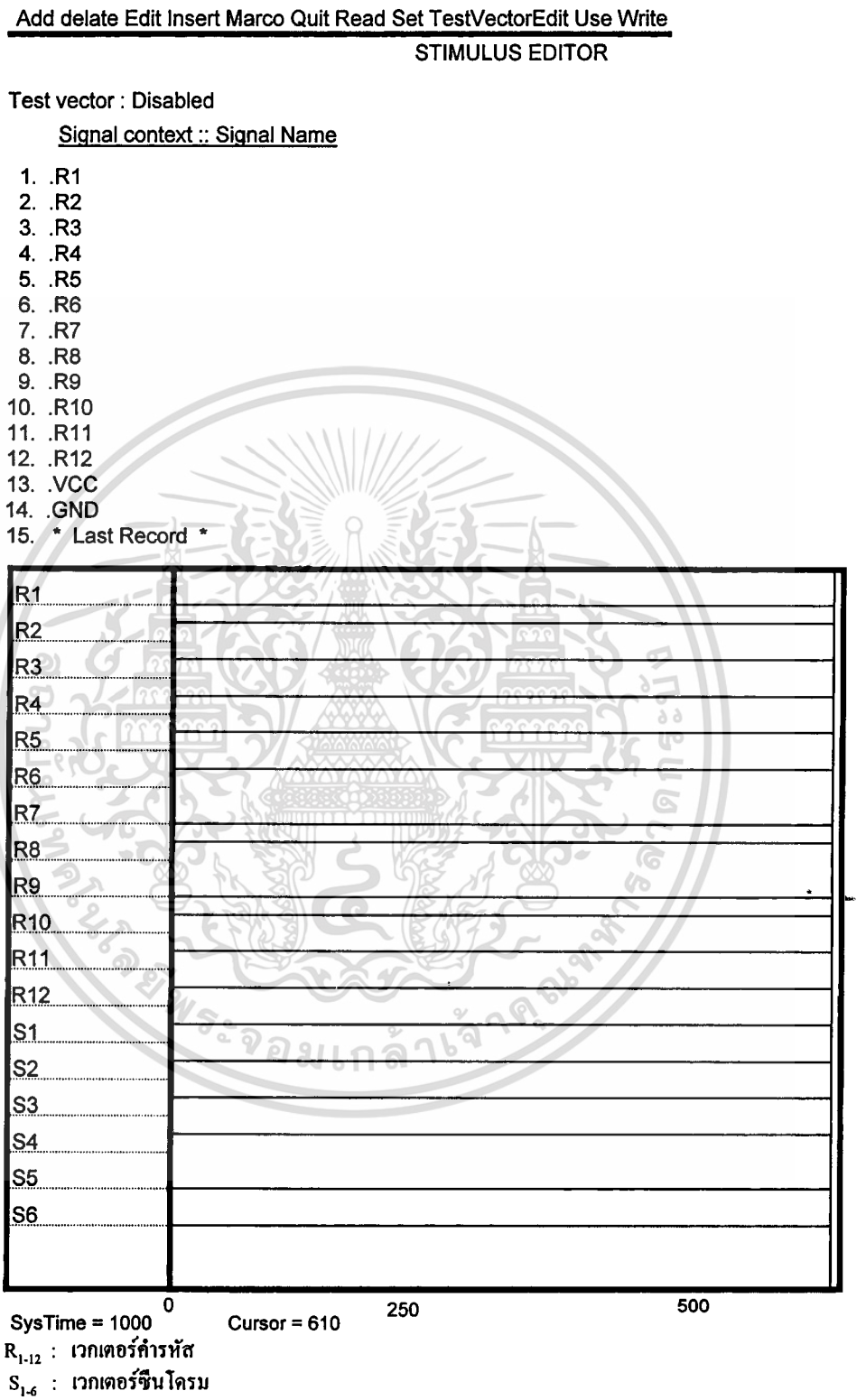
| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 1 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | x00000000000 | 101011 |
| 2 | 0x0000000000 | 010111 |
| 3 | 00x000000000 | 111010 |
| 4 | 000x00000000 | 101101 |
| 5 | 0000x0000000 | 000000 |
| 6 | 00000x000000 | 000000 |
| 7 | 000000x00000 | 100000 |
| 8 | 0000000x0000 | 010000 |
| 9 | 00000000x000 | 001000 |
| 10 | 000000000x00 | 000100 |
| 11 | 0000000000x0 | 000010 |
| 12 | 00000000000x | 000001 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

**ตารางที่ 4.6 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตที่ผิด 1 บิต**

**กรณีที่สาม** คำรหัสที่มีบิตผิด 2 บิต โดยกำหนดให้ 011111010111 เป็นเวกเตอร์คำรหัสที่มีบิตผิด 2 บิต จากเวกเตอร์คำรหัสเดิม 101111010111 ซึ่งจะทำให้ได้ค่าเวกเตอร์ซีนโดรมเป็น 111100 ดังรูปที่ 4.29 จะทำให้ได้รูปแบบผิดเป็น 110000000000 ดังรูปที่ 4.30 และผลจากการแก้ไขจะได้ข้อมูลเป็น 101111010111 ดังรูปที่ 4.31 ซึ่งในรูปแบบการผิด 2 บิต จะมีทั้งหมด 66 รูปแบบ แต่สามารถแก้ไขได้ 57 รูปแบบ ตามตารางที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.29 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 011111010111**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

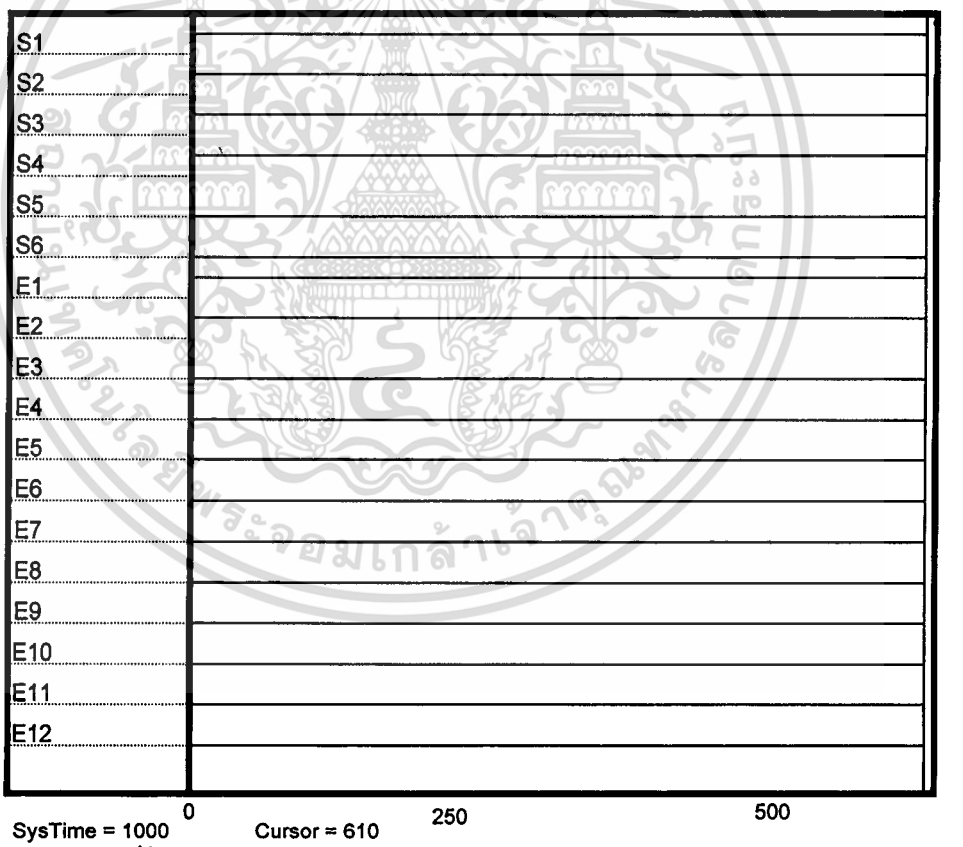
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *

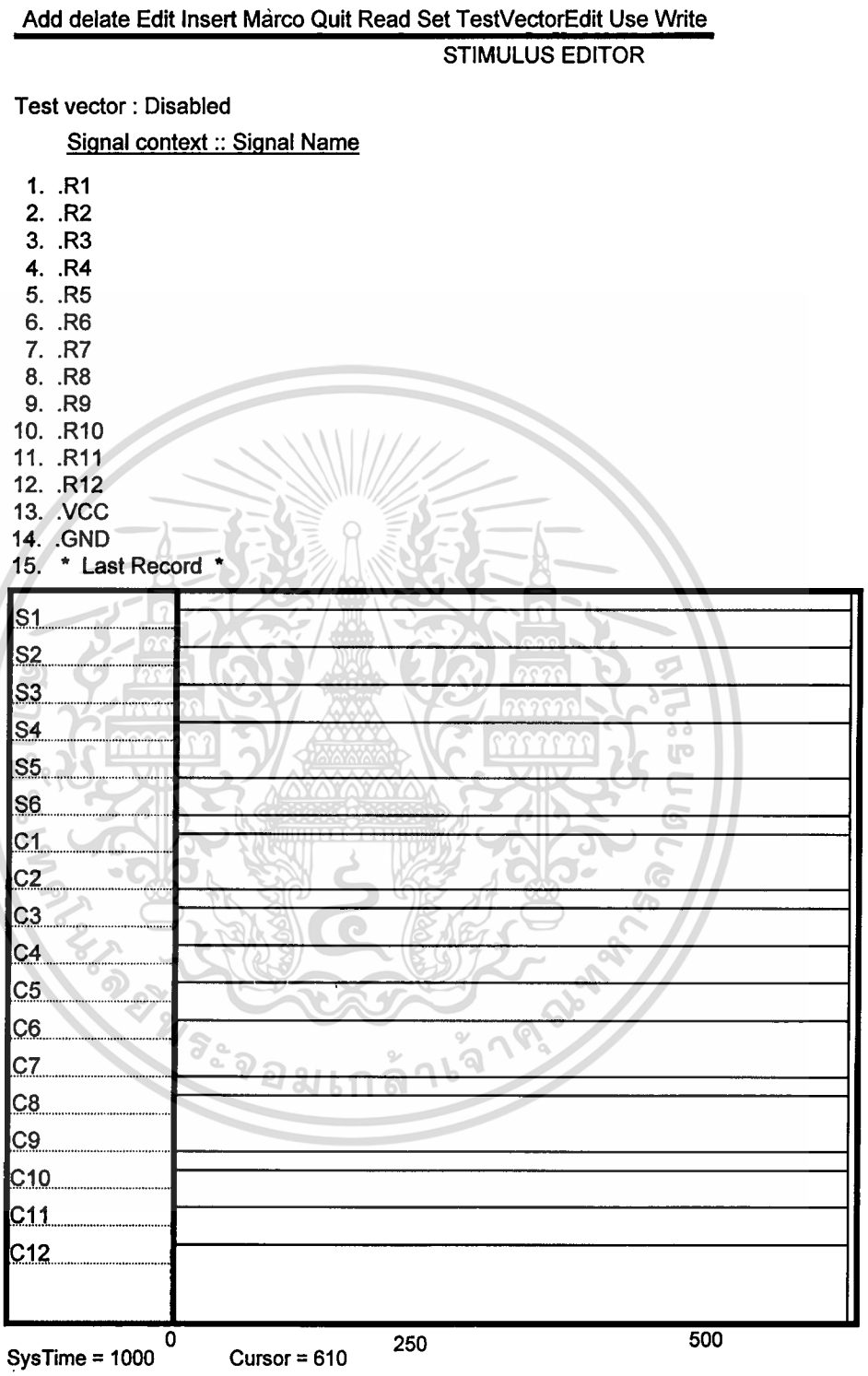



$S_{1-6}$ : เวกเตอร์ซีนโดรม

$E_{1-12}$ : เวกเตอร์รูปแบบบิตผิด

**รูปที่ 4.30 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 011111010111**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




$S_{1-6}$ : เวกเตอร์ซีนโดรม

$C_{1-12}$ : เวกเตอร์คำรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.31 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 011111010111**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx0000000000 * | 111100 |
| 2 | x0x000000000 * | 010001 |
| 3 | x00x00000000 * | 000110 |
| 4 | x000x0000000 * | 101011 |
| 5 | x0000x000000 * | 101011 |
| 6 | x00000x00000 * | 001011 |
| 7 | x000000x0000 * | 111011 |
| 8 | x0000000x000 * | 100011 |
| 9 | x00000000x00 * | 101111 |
| 10 | x000000000x0 * | 101001 |
| 11 | x0000000000x * | 101010 |
| 12 | 0xx000000000 | 101101 |
| 13 | 0x0x00000000 | 111010 |
| 14 | 0x00x0000000 * | 010111 |
| 15 | 0x000x000000 * | 010111 |
| 16 | 0x0000x00000 * | 110111 |
| 17 | 0x00000x0000 * | 000111 |
| 18 | 0x000000x000 * | 011111 |
| 19 | 0x0000000x00 * | 010011 |
| 20 | 0x00000000x0 * | 010101 |
| 21 | 0x000000000x * | 010110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

* : รูปแบบตำแหน่งบิตผิดที่สามารถแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.7 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตที่ผิด 2 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 22 | 00xx00000000 | 010111 |
| 23 | 00x0x0000000 * | 111010 |
| 24 | 00x00x000000 * | 111010 |
| 25 | 00x000x00000 * | 011010 |
| 26 | 00x0000x0000 | 101010 |
| 27 | 00x00000x000 * | 110010 |
| 28 | 00x000000x00 * | 111110 |
| 29 | 00x0000000x0 * | 111000 |
| 30 | 00x00000000x | 111011 |
| 31 | 000xx0000000 * | 101101 |
| 32 | 000x0x000000 * | 101101 |
| 33 | 000x00x00000 * | 001101 |
| 34 | 000x000x0000 * | 111101 |
| 35 | 000x0000x000 * | 100101 |
| 36 | 000x00000x00 | 101001 |
| 37 | 000x000000x0 | 101111 |
| 38 | 000x0000000x * | 101100 |
| 39 | 0000xx000000 * | 000000 |
| 40 | 0000x0x00000 * | 100000 |
| 41 | 0000x00x0000 * | 010000 |
| 42 | 0000x000x000 * | 001000 |
| 43 | 0000x0000x00 * | 000100 |
| 44 | 0000x00000x0 * | 000010 |
| 45 | 0000x000000x * | 000001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.7 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 2บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 46 | 00000xx00000 * | 100000 |
| 47 | 00000x0x0000 * | 010000 |
| 48 | 00000x00x000 * | 001000 |
| 49 | 00000x000x00 * | 000100 |
| 50 | 00000x0000x0 * | 000010 |
| 51 | 00000x00000x * | 000001 |
| 52 | 000000xx0000 * | 110000 |
| 53 | 000000x0x000 * | 101000 |
| 54 | 000000x00x00 * | 100100 |
| 55 | 000000x000x0 * | 100010 |
| 56 | 000000x0000x * | 100001 |
| 57 | 0000000xx000 * | 011000 |
| 58 | 0000000x0x00 * | 010100 |
| 59 | 0000000x000x | 010001 |
| 60 | 0000000x00x0 * | 010010 |
| 61 | 00000000xx00 * | 001100 |
| 62 | 00000000x0x0 * | 001010 |
| 63 | 00000000x00x * | 001001 |
| 64 | 000000000xx0 | 000110 |
| 65 | 000000000x0x * | 000101 |
| 66 | 0000000000xx * | 000011 |

**ตารางที่ 4.7 (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**กรณีที่สี่** คำรหัสที่มีบิตผิด 3 บิต กำหนดให้ 011101010111 มีบิตที่ 1,2 และ 5 ผิดจากเวกเตอร์คำรหัสเดิม 101111010111 จะทำให้ได้ค่าซินโดรมเป็น 111100 ดังรูปที่ 4.32 และได้รูปแบบบิตผิดเป็น 110010000000 ดังรูปที่ 4.33 เมื่อทำการแก้ไขจะได้ข้อมูลเป็น 101111010111 ดังรูปที่ 4.34 ซึ่งรูปแบบผิด 3 บิต จะมีอยู่ 220 รูปแบบ สามารถแก้ไขได้ 97 รูปแบบ ดังแสดงในตารางที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

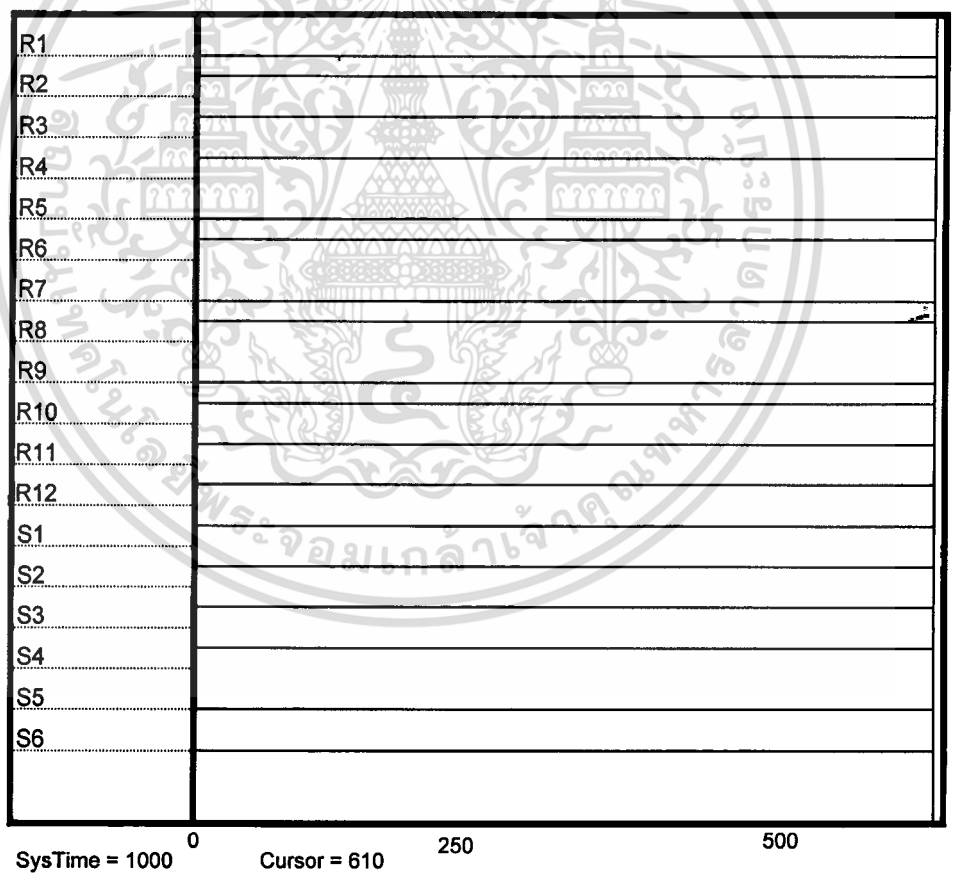
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *

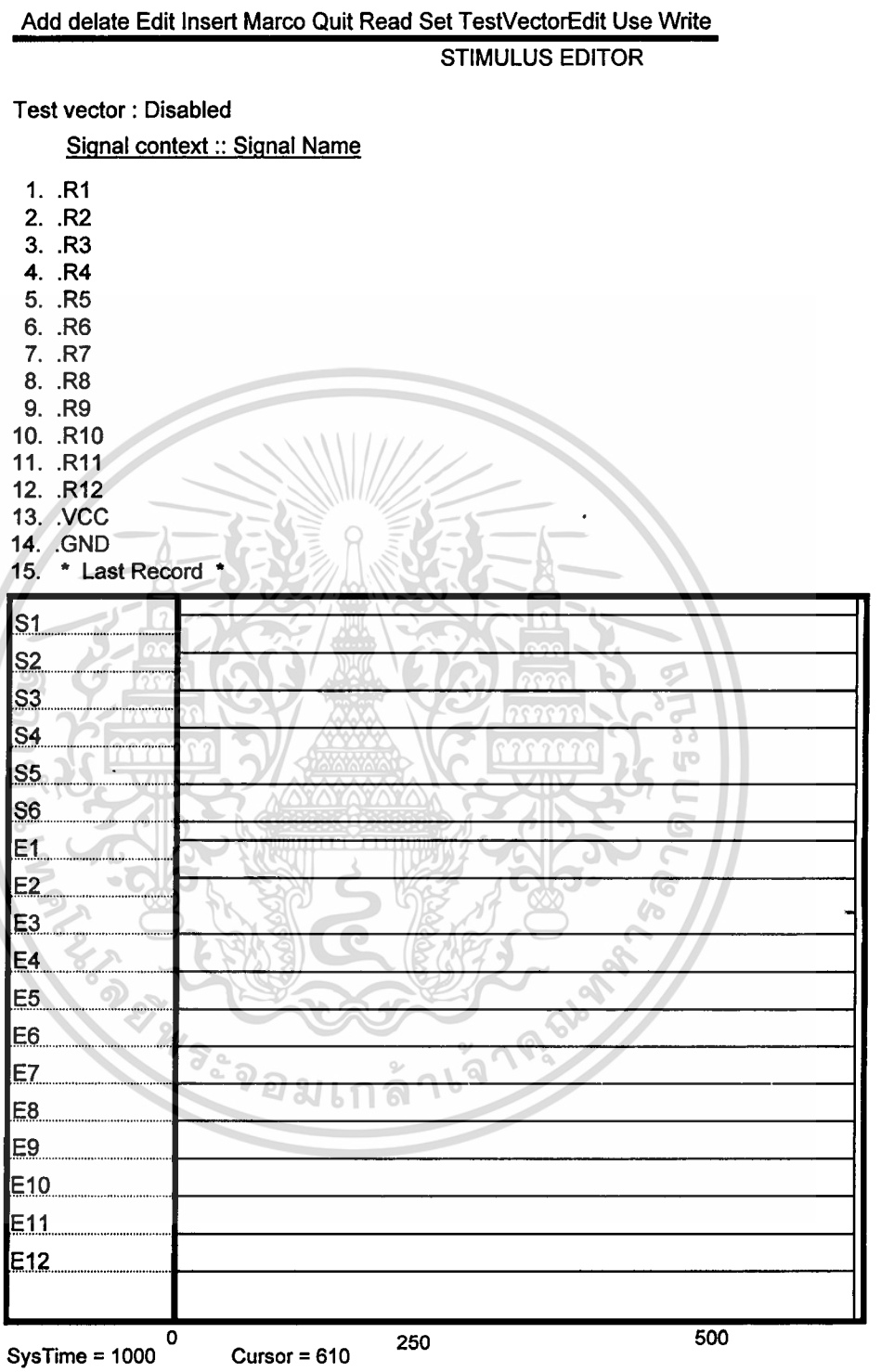



$R_{1-12}$ : เวกเตอร์ซีนโดรม

$S_{1-6}$ : เวกเตอร์ซีนโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.32 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 011101010111**

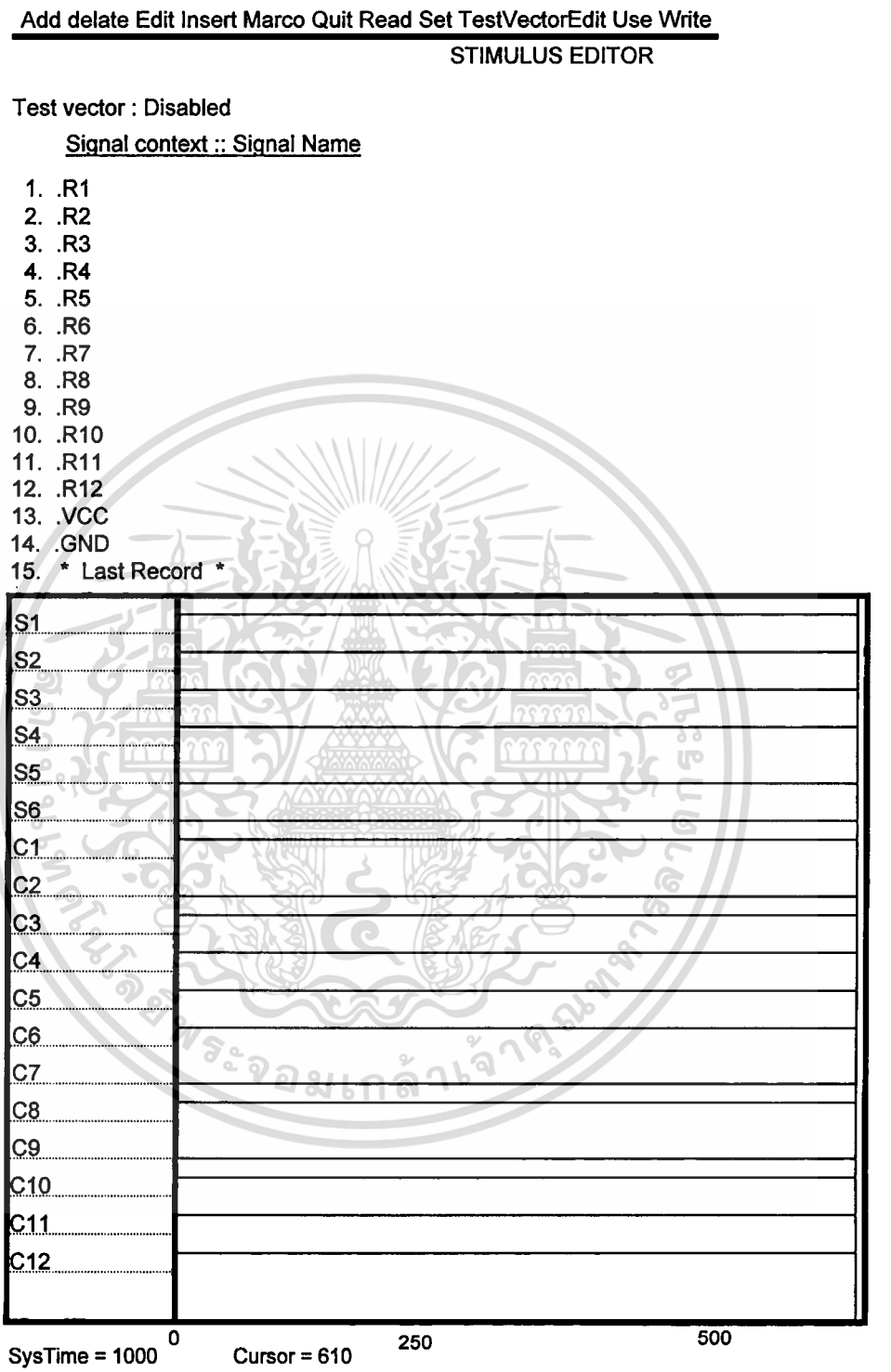



$S_{1-6}$ : เวกเตอร์ซีนโดรม

$E_{1-12}$ : เวกเตอร์รูปแบบตำแหน่งบิตผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งนี้ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ในเชิงการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

**รูปที่ 4.33 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 011101010111**




$S_{1-6}$ : เวกเตอร์ซีนโครม

$C_{1-12}$ : เวกเตอร์คำรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.34 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 011101010111**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xxx000000000 | 000110 |
| 2 | xx0x00000000 | 010001 |
| 3 | xx00x0000000 * | 111100 |
| 4 | xx000x000000 * | 111100 |
| 5 | xx0000x00000 * | 011100 |
| 6 | xx00000x0000 | 101100 |
| 7 | xx000000x000 * | 110100 |
| 8 | xx0000000x00 | 111000 |
| 9 | xx00000000x0 | 111110 |
| 10 | xx000000000x | 111101 |
| 11 | x0xx00000000 | 111100 |
| 12 | x0x0x0000000 * | 010001 |
| 13 | x0x00x000000 * | 010001 |
| 14 | x0x000x00000 * | 110001 |
| 15 | x0x0000x0000 | 000001 |
| 16 | x0x00000x000 * | 011001 |
| 17 | x0x000000x00 | 010101 |
| 18 | x0x0000000x0 | 010011 |
| 19 | x0x00000000x | 010000 |
| 20 | x00xx0000000 * | 000110 |
| 21 | x00x0x000000 * | 000110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

* : รูปแบบตำแหน่งบิตผิดที่สามารถแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตที่ผิด 3 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 22 | x00x00x00000 * | 100110 |
| 23 | x00x000x0000 | 010110 |
| 24 | x00x0000x000 * | 001110 |
| 25 | x00x00000x00 | 000010 |
| 26 | x00x000000x0 | 000100 |
| 27 | x00x0000000x | 000111 |
| 28 | x000xx000000 * | 101011 |
| 29 | x000x0x00000 * | 001011 |
| 30 | x000x00x0000 * | 111011 |
| 31 | x000x000x000 * | 100011 |
| 32 | x000x0000x00 * | 101111 |
| 33 | x000x00000x0 * | 101001 |
| 34 | x000x000000x * | 101010 |
| 35 | x0000xx00000 * | 001011 |
| 36 | x0000x0x0000 * | 111011 |
| 37 | x0000x00x000 * | 100011 |
| 38 | x0000x000x00 * | 101111 |
| 39 | x0000x0000x0 * | 101001 |
| 40 | x0000x00000x * | 101010 |
| 41 | x00000xx0000 * | 011011 |
| 42 | x00000x0x000 | 000011 |
| 43 | x000000x0x00 * | 001111 |
| 44 | x00000x000x0 | 001001 |
| 45 | x00000x0000x | 001010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 46 | x000000xx000 * | 110011 |
| 47 | x000000x0x00 * | 111111 |
| 48 | x000000x00x0 * | 111001 |
| 49 | x000000x000x | 111010 |
| 50 | x0000000xx00 * | 100111 |
| 51 | x0000000x0x0 | 100001 |
| 52 | x0000000x00x | 100010 |
| 53 | x00000000xx0 | 101101 |
| 54 | x00000000x0x | 101110 |
| 55 | x000000000xx | 101000 |
| 56 | 0xxx00000000 | 000000 |
| 57 | 0xx0x0000000 | 101101 |
| 58 | 0xx00x000000 | 101101 |
| 59 | 0xx000x00000 | 001101 |
| 60 | 0xx0000x0000 | 111101 |
| 61 | 0xx00000x000 | 100101 |
| 62 | 0xx000000x00 | 101001 |
| 63 | 0xx0000000x0 | 101111 |
| 64 | 0xx00000000x | 101100 |
| 65 | 0x0xx0000000 | 111010 |
| 66 | 0x0x0x000000 | 111010 |
| 67 | 0x0x00x00000 | 011010 |
| 68 | 0x0x000x0000 | 101010 |
| 69 | 0x0x0000x000 | 110010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 70 | 0x0x00000x00 | 111110 |
| 71 | 0x0x000000x0 | 111000 |
| 72 | 0x0x0000000x | 111011 |
| 73 | 0x00xx000000 * | 010111 |
| 74 | 0x00x0x00000 * | 110111 |
| 75 | 0x00x00x0000 * | 000111 |
| 76 | 0x00x000x000 * | 011111 |
| 77 | 0x00x0000x00 * | 010011 |
| 78 | 0x00x00000x0 * | 010101 |
| 79 | 0x00x000000x * | 010110 |
| 80 | 0x000xx00000 * | 110111 |
| 81 | 0x000x0x0000 * | 000111 |
| 82 | 0x000x00x000 * | 011111 |
| 83 | 0x000x000x00 * | 010011 |
| 84 | 0x000x0000x0 * | 010101 |
| 85 | 0x000x00000x * | 010110 |
| 86 | 0x0000xx0000 | 100111 |
| 87 | 0x0000x0x000 | 111111 |
| 88 | 0x0000x00x00 | 110011 |
| 89 | 0x0000x000x0 * | 110101 |
| 90 | 0x0000x0000x * | 110110 |
| 91 | 0x00000xx000 | 001111 |
| 92 | 0x00000x0x00 | 000011 |
| 93 | 0x00000x00x0 | 000101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 94 | 0x00000x000x | 000110 |
| 95 | 0x000000xx00 | 011011 |
| 96 | 0x000000x0x0 * | 011101 |
| 97 | 0x000000x00x * | 011110 |
| 98 | 0x0000000xx0 | 010001 |
| 99 | 0x0000000x0x | 010010 |
| 100 | 0x00000000xx | 010100 |
| 101 | 00xxx0000000 | 010111 |
| 102 | 00xx0x000000 | 010111 |
| 103 | 00xx00x00000 | 110111 |
| 104 | 00xx000x0000 | 000111 |
| 105 | 00xx0000x000 | 011111 |
| 106 | 00xx00000x00 | 010011 |
| 107 | 00xx000000x0 | 010101 |
| 108 | 00xx0000000x | 010110 |
| 109 | 00x0xx000000 * | 111010 |
| 110 | 00x0x0x00000 * | 011010 |
| 111 | 00x0x00x0000 | 101010 |
| 112 | 00x0x000x000 * | 110010 |
| 113 | 00x0x0000x00 * | 111110 |
| 114 | 00x0x00000x0 * | 111000 |
| 115 | 00x0x000000x | 111011 |
| 116 | 00x00xx00000 * | 011010 |
| 117 | 00x00x0x0000 | 101010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 118 | 00x00x00x000 * | 110010 |
| 119 | 00x00x000x00 * | 111110 |
| 120 | 00x00x0000x0 * | 111000 |
| 121 | 00x00x00000x | 111011 |
| 122 | 00x000xx0000 | 001010 |
| 123 | 00x000x0x000 | 010010 |
| 124 | 00x000x00x00 | 011110 |
| 125 | 00x000x000x0 | 011000 |
| 126 | 00x000x0000x | 011011 |
| 127 | 00x0000xx000 | 100010 |
| 128 | 00x0000x0x00 | 101110 |
| 129 | 00x0000x00x0 | 101000 |
| 130 | 00x0000x000x | 101011 |
| 131 | 00x00000xx00 | 110110 |
| 132 | 00x00000x0x0 | 110000 |
| 133 | 00x00000x00x | 110011 |
| 134 | 00x000000xx0 | 111100 |
| 135 | 00x000000x0x | 111111 |
| 136 | 00x0000000xx | 111001 |
| 137 | 000xxx000000 * | 101101 |
| 138 | 000xx0x00000 * | 001101 |
| 139 | 000xx00x0000 * | 111101 |
| 140 | 000xx000x000 * | 100101 |
| 141 | 000xx0000x00 | 101001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 142 | 000xx00000x0 | 101111 |
| 143 | 000xx000000x * | 101100 |
| 144 | 000x0xx00000 * | 001101 |
| 145 | 000x0x0x0000 * | 111101 |
| 146 | 000x0x00x000 * | 100101 |
| 147 | 000x0x000x00 | 101001 |
| 148 | 000x0x0000x0 | 101111 |
| 149 | 000x0x00000x * | 101100 |
| 150 | 000x00xx0000 | 011101 |
| 151 | 000x00x0x000 | 000101 |
| 152 | 000x00x00x00 | 001001 |
| 153 | 000x00x000x0 | 001111 |
| 154 | 000x00x0000x | 001100 |
| 155 | 000x000xx000 | 110001 |
| 156 | 000x000x0x00 | 111001 |
| 157 | 000x000x00x0 | 111111 |
| 158 | 000x000x000x | 111100 |
| 159 | 000x0000xx00 | 100001 |
| 160 | 000x0000x0x0 | 100111 |
| 161 | 000x0000x00x | 100100 |
| 162 | 000x00000xx0 | 101011 |
| 163 | 000x00000x0x | 101000 |
| 164 | 000x000000xx | 101110 |
| 165 | 0000xxx00000 * | 100000 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 166 | 0000xx0x0000 * | 010000 |
| 167 | 0000xx00x000 * | 001000 |
| 168 | 0000xx000x00 * | 000100 |
| 169 | 0000xx0000x0 * | 000010 |
| 170 | 0000xx00000x * | 000001 |
| 171 | 0000x0xx0000 * | 110000 |
| 172 | 0000x0x0x000 * | 101000 |
| 173 | 0000x0x00x00 * | 100100 |
| 174 | 0000x0x000x0 * | 100010 |
| 175 | 0000x0x0000x * | 100001 |
| 176 | 0000x00xx000 * | 011000 |
| 177 | 0000x00x0x00 * | 010100 |
| 178 | 0000x00x00x0 * | 010010 |
| 179 | 0000x00x000x | 010001 |
| 180 | 0000x000xx00 * | 001100 |
| 181 | 0000x000x0x0 * | 001010 |
| 182 | 0000x000x00x * | 001001 |
| 183 | 0000x0000xx0 | 000110 |
| 184 | 0000x0000x0x * | 000101 |
| 185 | 0000x00000xx * | 000011 |
| 186 | 00000xxx0000 * | 110000 |
| 187 | 00000xx0x000 * | 101000 |
| 188 | 00000xx00x00 * | 100100 |
| 189 | 00000xx000x0 * | 100010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 190 | 00000xx0000x * | 100001 |
| 191 | 00000x0xx000 * | 011000 |
| 192 | 00000x0x0x00 * | 010100 |
| 193 | 00000x0x00x0 * | 010010 |
| 194 | 00000x0x000x | 010001 |
| 195 | 00000x00xx00 * | 001100 |
| 196 | 00000x00x0x0 * | 001010 |
| 197 | 00000x00x00x * | 001001 |
| 198 | 00000x000xx0 | 000110 |
| 199 | 00000x000x0x * | 000101 |
| 200 | 00000x0000xx * | 000011 |
| 201 | 000000xxx000 | 111000 |
| 202 | 000000xx0x00 | 110100 |
| 203 | 000000xx00x0 | 110010 |
| 204 | 000000xx000x | 110001 |
| 205 | 000000x0xx00 | 101100 |
| 206 | 000000x0x0x0 | 101010 |
| 207 | 000000x0x00x | 101001 |
| 208 | 000000x00xx0 | 100110 |
| 209 | 000000x00x0x | 100101 |
| 210 | 000000x000xx | 100011 |
| 211 | 0000000xxx00 | 011100 |
| 212 | 0000000xx0x0 | 011010 |
| 213 | 0000000xx00x | 011001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.8 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 3 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 214 | 0000000x0xx0 | 010110 |
| 215 | 0000000x0x0x | 010101 |
| 216 | 0000000x00xx | 010011 |
| 217 | 00000000xxx0 | 001110 |
| 218 | 00000000xx0x | 001101 |
| 219 | 00000000x0xx | 001011 |
| 220 | 000000000xxx | 000111 |

**ตารางที่ 4.8 (ต่อ)**

**กรณีที่ห้า** เวกเตอร์คำรหัสที่มีบิตผิด 4 บิต กำหนดให้110010100011 เป็นเวกเตอร์คำรหัสที่มีบิตผิดไป 4 บิต จากเวกเตอร์คำรหัสเดิม 000011101011 ทางด้านอินพุตจะได้ผลของเวกเตอร์ซีนโดรมเป็น 111100 ดังรูปที่ 4.35 และปรากฏรูปแบบบิตผิดเป็น 110001001000 ดังรูปที่ 4.36 และผลจากการแก้ไขเวกเตอร์คำรหัสที่เข้ามาเป็น 000011101011 ดังรูปที่4.37 และจากรูปแบบบิตที่ผิด 4 บิตนี้จะมีอยู่ทั้งหมด 495 รูปแบบ แต่สามารถแก้ไขได้ 98 รูปแบบ ดังแสดงในตารางที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

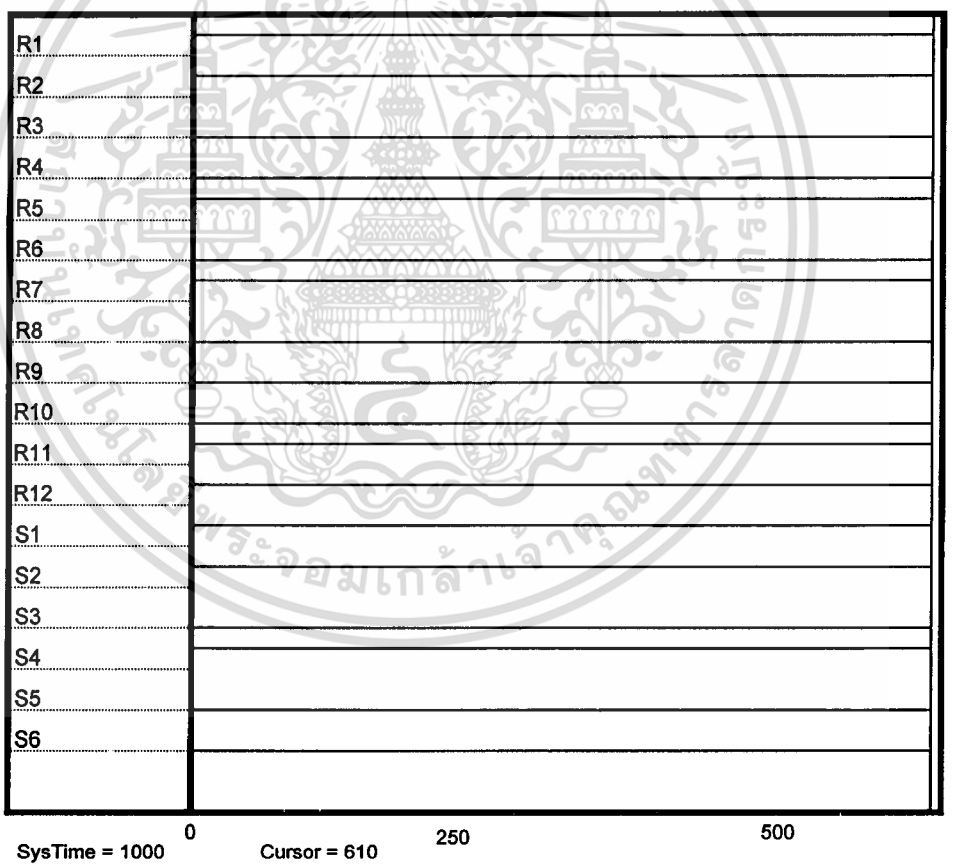
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




$R_{1-12}$ : เวกเตอร์คำรหัส

$S_{1-6}$ : เวกเตอร์ซีนโครม

**รูปที่ 4.35 ผลของเวกเตอร์ซีนโครมที่ได้จากเวกเตอร์คำรหัส 110010100011**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

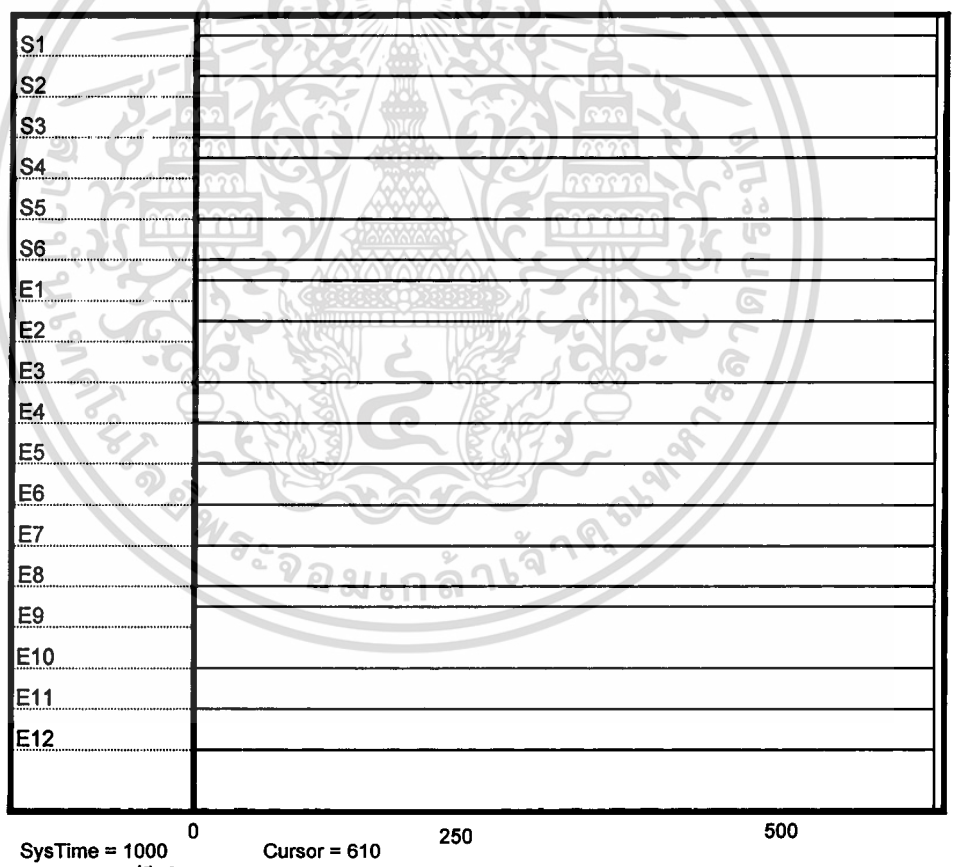
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




$S_{1-6}$ : เวกเตอร์ชีนโครม

$E_{1-12}$ : เวกเตอร์รูปแบบตำแหน่งบิตผิด

**รูปที่ 4.36 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 110010100011**

เอกสารนี้เป็นเอกสาร[illegible]ระโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

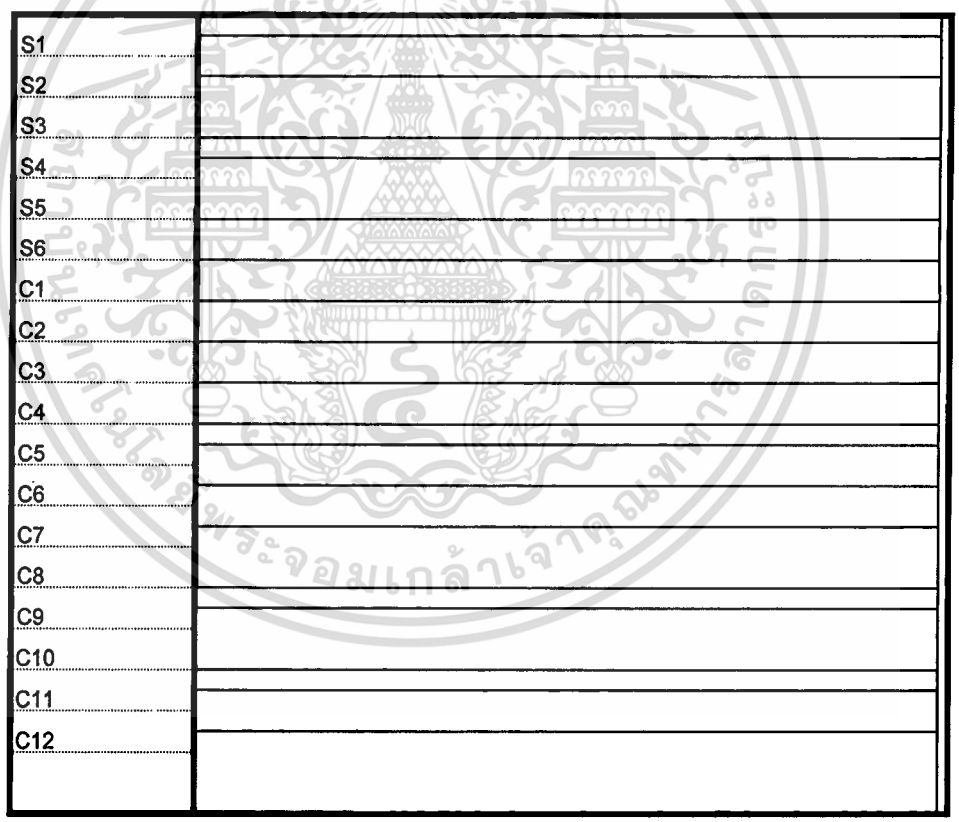
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




SysTime = 1000 Cursor = 610

$S_{1-6}$ : เวกเตอร์ซีนโครม

$C_{1-12}$ : เวกเตอร์คำรหัส

**รูปที่ 4.37 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 110010100011**

เอกสารนี้เป็นเอกสาร ได้จำกัดการใช้ ที่ ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xxxx00000000 | 101011 |
| 2 | xxx0x0000000 | 000110 |
| 3 | xxx00x000000 | 000110 |
| 4 | xxx000x00000 | 100110 |
| 5 | xxx0000x0000 | 010110 |
| 6 | xxx00000x000 | 001110 |
| 7 | xxx000000x00 | 000010 |
| 8 | xxx0000000x0 | 000100 |
| 9 | xxx00000000x | 000111 |
| 10 | xx0xx0000000 | 010001 |
| 11 | xx0x0x000000 | 010001 |
| 12 | xx0x00x00000 | 110001 |
| 13 | xx0x000x0000 | 000001 |
| 14 | xx0x0000x000 | 011001 |
| 15 | xx0x00000x00 | 010101 |
| 16 | xx0x000000x0 | 010011 |
| 17 | xx0x0000000x | 010000 |
| 18 | xx00xx000000 * | 111100 |
| 19 | xx00x0x00000 * | 011100 |
| 20 | xx00x00x0000 | 101100 |
| 21 | xx00x000x000 * | 110100 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

* : รูปแบบตำแหน่งบิตผิดที่สามารถแก้ไขได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตที่ผิด 4 บิต**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 22 | xx00x0000x00 | 111000 |
| 23 | xx00x00000x0 | 111110 |
| 24 | xx00x000000x | 111101 |
| 25 | xx000xx00000 * | 011100 |
| 26 | xx000x0x0000 | 101100 |
| 27 | xx000x00x000 * | 110100 |
| 28 | xx000x000x00 | 111000 |
| 29 | xx000x0000x0 | 111110 |
| 30 | xx000x00000x | 111101 |
| 31 | xx0000xx0000 | 001100 |
| 32 | xx0000x0x000 | 010100 |
| 33 | xx0000x00x00 | 011000 |
| 34 | xx0000x000x0 | 011110 |
| 35 | xx0000x0000x | 011101 |
| 36 | xx00000xx000 | 100100 |
| 37 | xx00000x0x00 | 101000 |
| 38 | xx00000x00x0 | 101110 |
| 39 | xx00000x000x | 101101 |
| 40 | xx000000xx00 | 110000 |
| 41 | xx000000x0x0 | 110110 |
| 42 | xx000000x00x | 110101 |
| 43 | xx0000000xx0 | 111010 |
| 44 | xx0000000x0x | 111001 |
| 45 | xx00000000xx | 111111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง เผยแพร่ หรืออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 46 | x0xxx0000000 | 111100 |
| 47 | x0xx0x000000 | 111100 |
| 48 | x0xx00x00000 | 011100 |
| 49 | x0xx000x0000 | 101100 |
| 50 | x0xx0000x000 | 110100 |
| 51 | x0xx00000x00 | 111000 |
| 52 | x0xx000000x0 | 111110 |
| 53 | x0xx0000000x | 111101 |
| 54 | x0x0xx000000 * | 010001 |
| 55 | x0x0x0x00000 * | 110001 |
| 56 | x0x0x00x0000 | 000001 |
| 57 | x0x0x000x000 * | 011001 |
| 58 | x0x0x0000x00 | 010101 |
| 59 | x0x0x00000x0 | 010011 |
| 60 | x0x0x000000x | 010000 |
| 61 | x0x00xx00000 * | 110001 |
| 62 | x0x00x0x0000 | 000001 |
| 63 | x0x00x00x000 * | 011001 |
| 64 | x0x00x000x00 | 010101 |
| 65 | x0x00x0000x0 | 010011 |
| 66 | x0x00x00000x | 010000 |
| 67 | x0x000xx0000 | 100001 |
| 68 | x0x000x0x000 | 111001 |
| 69 | x0x000x00x00 | 110101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 70 | x0x000x000x0 | 110011 |
| 71 | x0x000x0000x | 110000 |
| 72 | x0x0000xx000 | 001001 |
| 73 | x0x0000x0x00 | 000101 |
| 74 | x0x0000x00x0 | 000011 |
| 75 | x0x0000x000x | 000000 |
| 76 | x0x00000xx00 | 011101 |
| 77 | x0x00000x0x0 | 011011 |
| 78 | x0x00000x00x | 011000 |
| 79 | x0x000000xx0 | 010111 |
| 80 | x0x000000x0x | 010100 |
| 81 | x0x0000000xx | 010010 |
| 82 | x00xxx000000 * | 000110 |
| 83 | x00xx0x00000 * | 100110 |
| 84 | x00xx00x0000 | 010110 |
| 85 | x00xx000x000 * | 001110 |
| 86 | x00xx0000x00 | 000010 |
| 87 | x00xx00000x0 | 000100 |
| 88 | x00xx000000x | 000111 |
| 89 | x00x0xx00000 * | 100110 |
| 90 | x00x0x0x0000 | 010110 |
| 91 | x00x0x00x000 * | 001110 |
| 92 | x00x0x000x00 | 000010 |
| 93 | x00x0x0000x0 | 000100 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 94 | x00x0x00000x | 000111 |
| 95 | x00x00xx0000 | 110110 |
| 96 | x00x00x0x000 | 101110 |
| 97 | x00x00x00x00 | 100010 |
| 98 | x00x00x000x0 | 100100 |
| 99 | x00x00x0000x | 100111 |
| 100 | x00x000xx000 | 011110 |
| 101 | x00x000x0x00 | 010010 |
| 102 | x00x000x00x0 | 010100 |
| 103 | x00x000x000x | 010111 |
| 104 | x00x0000xx00 | 001010 |
| 105 | x00x0000x0x0 | 001100 |
| 106 | x00x0000x00x | 001111 |
| 107 | x00x00000xx0 | 000000 |
| 108 | x00x00000x0x | 000011 |
| 109 | x00x000000xx | 000101 |
| 110 | x000xxx00000 * | 001011 |
| 111 | x000xx0x0000 * | 111011 |
| 112 | x000xx00x000 * | 100011 |
| 113 | x000xx000x00 * | 101111 |
| 114 | x000xx0000x0 * | 101001 |
| 115 | x000xx00000x * | 101010 |
| 116 | x000x0xx0000 * | 011011 |
| 117 | x000x0x0x000 | 000011 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 118 | x000x0x00x00 * | 001111 |
| 119 | x000x0x000x0 | 001001 |
| 120 | x000x0x0000x | 001010 |
| 121 | x000x00xx000 * | 110011 |
| 122 | x000x00x0x00 * | 111111 |
| 123 | x000x00x00x0 * | 111001 |
| 124 | x000x00x000x | 111010 |
| 125 | x000x000xx00 * | 100111 |
| 126 | x000x000x0x0 | 100001 |
| 127 | x000x000x00x | 100010 |
| 128 | x000x0000xx0 | 101101 |
| 129 | x000x0000x0x | 101110 |
| 130 | x000x00000xx | 101000 |
| 131 | x0000xxx0000 * | 011011 |
| 132 | x0000xx0x000 | 000011 |
| 133 | x0000xx00x00 * | 001111 |
| 134 | x0000xx000x0 | 001001 |
| 135 | x0000xx0000x | 001010 |
| 136 | x0000x0xx000 * | 110011 |
| 137 | x0000x0x0x00 * | 111111 |
| 138 | x0000x0x00x0 * | 111001 |
| 139 | x0000x0x000x | 111010 |
| 140 | x0000x00xx00 * | 100111 |
| 141 | x0000x00x0x0 | 100001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 142 | x0000x00x00x | 100010 |
| 143 | x0000x000xx0 | 101101 |
| 144 | x0000x000x0x | 101110 |
| 145 | x0000x0000xx | 101000 |
| 146 | x00000xxx000 | 010011 |
| 147 | x00000xx0x00 | 011111 |
| 148 | x00000xx00x0 | 011001 |
| 149 | x00000xx000x | 011010 |
| 150 | x00000x0xx00 | 000111 |
| 151 | x00000x0x0x0 | 000001 |
| 152 | x00000x0x00x | 000010 |
| 153 | x00000x00xx0 | 001101 |
| 154 | x00000x00x0x | 001110 |
| 155 | x00000x000xx | 001000 |
| 156 | x000000xxx00 | 110111 |
| 157 | x000000xx0x0 | 110001 |
| 158 | x000000xx00x | 110010 |
| 159 | x000000x0xx0 | 111101 |
| 160 | x000000x0x0x | 111110 |
| 161 | x000000x00xx | 111000 |
| 162 | x0000000xxx0 | 100101 |
| 163 | x0000000xx0x | 100110 |
| 164 | x0000000x0xx | 100000 |
| 165 | x00000000xxx | 101100 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 166 | 0xxxx0000000 | 000000 |
| 167 | 0xxx0x000000 | 000000 |
| 168 | 0xxx00x00000 | 100000 |
| 169 | 0xxx000x0000 | 010000 |
| 170 | 0xxx0000x000 | 001000 |
| 171 | 0xxx00000x00 | 000100 |
| 172 | 0xxx000000x0 | 000010 |
| 173 | 0xxx0000000x | 000001 |
| 174 | 0xx0xx000000 | 101101 |
| 175 | 0xx0x0x00000 | 001101 |
| 176 | 0xx0x00x0000 | 111101 |
| 177 | 0xx0x000x000 | 100101 |
| 178 | 0xx0x0000x00 | 101001 |
| 179 | 0xx0x00000x0 | 101111 |
| 180 | 0xx0x000000x | 101100 |
| 181 | 0xx00xx00000 | 001101 |
| 182 | 0xx00x0x0000 | 111101 |
| 183 | 0xx00x00x000 | 100101 |
| 184 | 0xx00x000x00 | 101001 |
| 185 | 0xx00x0000x0 | 101111 |
| 186 | 0xx00x00000x | 101100 |
| 187 | 0xx000xx0000 | 011101 |
| 188 | 0xx000x0x000 | 000101 |
| 189 | 0xx000x00x00 | 001001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 190 | 0xx000x000x0 | 001111 |
| 191 | 0xx000x0000x | 001100 |
| 192 | 0xx0000xx000 | 110101 |
| 193 | 0xx0000x0x00 | 111001 |
| 194 | 0xx0000x00x0 | 111111 |
| 195 | 0xx0000x000x | 111100 |
| 196 | 0xx00000xx00 | 100001 |
| 197 | 0xx00000x0x0 | 100111 |
| 198 | 0xx00000x00x | 100100 |
| 199 | 0xx000000xx0 | 101011 |
| 200 | 0xx000000x0x | 101000 |
| 201 | 0xx0000000xx | 101110 |
| 202 | 0x0xxx000000 | 111010 |
| 203 | 0x0xx0x00000 | 011010 |
| 204 | 0x0xx00x0000 | 101010 |
| 205 | 0x0xx000x000 | 110010 |
| 206 | 0x0xx0000x00 | 111110 |
| 207 | 0x0xx00000x0 | 111000 |
| 208 | 0x0xx000000x | 111011 |
| 209 | 0x0x0xx00000 | 011010 |
| 210 | 0x0x0x0x0000 | 101010 |
| 211 | 0x0x0x00x000 | 110010 |
| 212 | 0x0x0x000x00 | 111110 |
| 213 | 0x0x0x0000x0 | 111000 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 214 | 0x0x0x00000x | 111011 |
| 215 | 0x0x00xx0000 | 001010 |
| 216 | 0x0x00x0x000 | 010010 |
| 217 | 0x0x00x00x00 | 011110 |
| 218 | 0x0x00x000x0 | 011000 |
| 219 | 0x0x00x0000x | 011011 |
| 220 | 0x0x000xx000 | 100010 |
| 221 | 0x0x000x0x00 | 101110 |
| 222 | 0x0x000x00x0 | 101000 |
| 223 | 0x0x000x000x | 101011 |
| 224 | 0x0x0000xx00 | 110110 |
| 225 | 0x0x0000x0x0 | 110000 |
| 226 | 0x0x0000x00x | 110011 |
| 227 | 0x0x00000xx0 | 111100 |
| 228 | 0x0x00000x0x | 111111 |
| 229 | 0x0x000000xx | 111001 |
| 230 | 0x00xxx00000 * | 110111 |
| 231 | 0x00xx0x0000 * | 000111 |
| 232 | 0x00xx00x000 * | 011111 |
| 233 | 0x00xx000x00 * | 010011 |
| 234 | 0x00xx0000x0 * | 010101 |
| 235 | 0x00xx00000x * | 010110 |
| 236 | 0x00x0xx0000 | 100111 |
| 237 | 0x00x0x0x000 | 111111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 238 | 0x00x0x00x00 | 110011 |
| 239 | 0x00x0x000x0 * | 110101 |
| 240 | 0x00x0x0000x * | 110110 |
| 241 | 0x00x00xx000 | 001111 |
| 242 | 0x00x00x0x00 | 000011 |
| 243 | 0x00x00x00x0 | 000101 |
| 244 | 0x00x00x000x | 000110 |
| 245 | 0x00x000xx00 | 011011 |
| 246 | 0x00x000x0x0 * | 011101 |
| 247 | 0x00x000x00x * | 011110 |
| 248 | 0x00x0000xx0 | 010001 |
| 249 | 0x00x0000x0x | 010010 |
| 250 | 0x00x00000xx | 010100 |
| 251 | 0x000xxx0000 | 100111 |
| 252 | 0x000xx0x000 | 111111 |
| 253 | 0x000xx00x00 | 110011 |
| 254 | 0x000xx000x0 * | 110101 |
| 255 | 0x000xx0000x * | 110110 |
| 256 | 0x000x0xx000 | 001111 |
| 257 | 0x000x0x0x00 | 000011 |
| 258 | 0x000x0x00x0 | 000101 |
| 259 | 0x000x0x000x | 000110 |
| 260 | 0x000x00xx00 | 011011 |
| 261 | 0x000x00x0x0 * | 011101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 262 | 0x000x00x00x * | 011110 |
| 263 | 0x000x000xx0 | 010001 |
| 264 | 0x000x000x0x | 010010 |
| 265 | 0x000x0000xx | 010100 |
| 266 | 0x0000xxx000 | 101111 |
| 267 | 0x0000xx0x00 | 100011 |
| 268 | 0x0000xx00x0 | 100101 |
| 269 | 0x0000xx000x | 100110 |
| 270 | 0x0000x0xx00 | 111011 |
| 271 | 0x0000x0x0x0 | 111101 |
| 272 | 0x0000x0x00x | 111110 |
| 273 | 0x0000x00xx0 | 110001 |
| 274 | 0x0000x00x0x | 110010 |
| 275 | 0x0000x000xx | 110100 |
| 276 | 0x00000xxx00 | 001011 |
| 277 | 0x00000xx0x0 | 001101 |
| 278 | 0x00000xx00x | 001110 |
| 279 | 0x00000x0xx0 | 000001 |
| 280 | 0x00000x0x0x | 000010 |
| 281 | 0x00000x00xx | 000100 |
| 282 | 0x000000xxx0 | 011001 |
| 283 | 0x000000xx0x | 011010 |
| 284 | 0x000000x0xx | 011100 |
| 285 | 0x0000000xxx | 010000 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 286 | 00xxxx000000 | 010111 |
| 287 | 00xxx0x00000 | 110111 |
| 288 | 00xxx00x0000 | 000111 |
| 289 | 00xxx000x000 | 011111 |
| 290 | 00xxx0000x00 | 010011 |
| 291 | 00xxx00000x0 | 010101 |
| 292 | 00xxx000000x | 010110 |
| 293 | 00xx0xx00000 | 110111 |
| 294 | 00xx0x0x0000 | 000111 |
| 295 | 00xx0x00x000 | 011111 |
| 296 | 00xx0x000x00 | 010011 |
| 297 | 00xx0x0000x0 | 010101 |
| 298 | 00xx0x00000x | 010110 |
| 299 | 00xx00xx0000 | 100111 |
| 300 | 00xx00x0x000 | 111111 |
| 301 | 00xx00x00x00 | 110011 |
| 302 | 00xx00x000x0 | 110101 |
| 303 | 00xx00x0000x | 110110 |
| 304 | 00xx000xx000 | 001111 |
| 305 | 00xx000x0x00 | 000011 |
| 306 | 00xx00010010 | 000101 |
| 307 | 0011000x000x | 000110 |
| 308 | 00xx0000xx00 | 011011 |
| 309 | 00xx0000x0x0 | 011101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 310 | 00xx0000x00x | 011110 |
| 311 | 00xx00000xx0 | 010001 |
| 312 | 00xx00000x0x | 010010 |
| 313 | 00xx000000xx | 010100 |
| 314 | 00x0xxx00000 * | 011010 |
| 315 | 00x0xx0x0000 | 101010 |
| 316 | 00x0xx00x000 * | 110010 |
| 317 | 00x0xx000x00 * | 111110 |
| 318 | 00x0xx0000x0 * | 111000 |
| 319 | 00x0xx00000x | 111011 |
| 320 | 00x0x0xx0000 | 001010 |
| 321 | 00x0x0x0x000 | 010010 |
| 322 | 00x0x0x00x00 | 011110 |
| 323 | 00x0x0x000x0 | 011000 |
| 324 | 00x0x0x0000x | 011011 |
| 325 | 00x0x00xx000 | 100010 |
| 326 | 00x0x00x0x00 | 101110 |
| 327 | 00x0x00x00x0 | 101000 |
| 328 | 00x0x00x000x | 101011 |
| 329 | 00x0x000xx00 | 110110 |
| 330 | 00x0x000x0x0 | 110000 |
| 331 | 00x0x000x00x | 110011 |
| 332 | 00x0x0000xx0 | 111100 |
| 333 | 00x0x0000x0x | 111111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 334 | 00x0x00000xx | 111001 |
| 335 | 00x00xxx0000 | 001010 |
| 336 | 00x00xx0x000 | 010010 |
| 337 | 00x00xx00x00 | 011110 |
| 338 | 00x00xx000x0 | 011000 |
| 339 | 00x00xx0000x | 011011 |
| 340 | 00x00x0xx000 | 100010 |
| 341 | 00x00x0x0x00 | 101110 |
| 342 | 00x00x0x00x0 | 101000 |
| 343 | 00x00x0x000x | 101011 |
| 344 | 00x00x00xx00 | 110110 |
| 345 | 00x00x00x0x0 | 110000 |
| 346 | 00x00x00x00x | 110011 |
| 347 | 00x00x000xx0 | 111100 |
| 348 | 00x00x000x0x | 111111 |
| 349 | 00x00x0000xx | 111001 |
| 350 | 00x000xxx000 | 000010 |
| 351 | 00x000xx0x00 | 001110 |
| 352 | 00x000xx00x0 | 001000 |
| 353 | 00x000xx000x | 001011 |
| 354 | 00x000x0xx00 | 010110 |
| 355 | 00x000x0x0x0 | 010000 |
| 356 | 00x000x0x00x | 010011 |
| 357 | 00x000x00xx0 | 011100 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 358 | 00x000x00x0x | 011111 |
| 359 | 00x000x000xx | 011001 |
| 360 | 00x0000xxx00 | 100110 |
| 361 | 00x0000xx0x0 | 100000 |
| 362 | 00x0000xx00x | 100011 |
| 363 | 00x0000x0xx0 | 101100 |
| 364 | 00x0000x0x0x | 101111 |
| 365 | 00x0000x00xx | 101001 |
| 366 | 00x00000xxx0 | 110100 |
| 367 | 00x00000xx0x | 110111 |
| 368 | 00x00000x0xx | 110001 |
| 369 | 00x000000xxx | 111101 |
| 370 | 000xxxx00000 * | 001101 |
| 371 | 000xxx0x0000 * | 111101 |
| 372 | 000xxx00x000 * | 100101 |
| 373 | 000xxx000x00 | 101001 |
| 374 | 000xxx0000x0 | 101111 |
| 375 | 000xxx00000x * | 101100 |
| 376 | 000xx0xx0000 | 011101 |
| 377 | 000xx0x0x000 | 000101 |
| 378 | 000xx0x00x00 | 001001 |
| 379 | 000xx0x000x0 | 001111 |
| 380 | 000xx0x0000x | 001100 |
| 381 | 000xx00xx000 | 110101 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 382 | 000xx00x0x00 | 111001 |
| 383 | 000xx00x00x0 | 111111 |
| 384 | 000xx00x000x | 111100 |
| 385 | 000xx000xx00 | 100001 |
| 386 | 000xx000x0x0 | 100111 |
| 387 | 000xx000x00x | 100100 |
| 388 | 000xx0000xx0 | 101011 |
| 389 | 000xx0000x0x | 101000 |
| 390 | 000xx00000xx | 101110 |
| 391 | 000x0xxx0000 | 011101 |
| 392 | 000x0xx0x000 | 000101 |
| 393 | 000x0xx00x00 | 001001 |
| 394 | 000x0xx000x0 | 001111 |
| 395 | 000x0xx0000x | 001100 |
| 396 | 000x0x0xx000 | 110101 |
| 397 | 000x0x0x0x00 | 111001 |
| 398 | 000x0x0x00x0 | 111111 |
| 399 | 000x0x0x000x | 111100 |
| 400 | 000x0x00xx00 | 100001 |
| 401 | 000x0x00x0x0 | 100111 |
| 402 | 000x0x00x00x | 100100 |
| 403 | 000x0x000xx0 | 101011 |
| 404 | 000x0x000x0x | 101000 |
| 405 | 000x0x0000xx | 101110 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 406 | 000x00xxx000 | 010101 |
| 407 | 000x00xx0x00 | 011001 |
| 408 | 000x00xx00x0 | 011111 |
| 409 | 000x00xx000x | 011100 |
| 410 | 000x00x0xx00 | 000001 |
| 411 | 000x00x0x0x0 | 000111 |
| 412 | 000x00x0x00x | 000100 |
| 413 | 000x00x00xx0 | 001011 |
| 414 | 000x00x00x0x | 001000 |
| 415 | 000x00x000xx | 001110 |
| 416 | 000x000xxx00 | 110001 |
| 417 | 000x000xx0x0 | 110111 |
| 418 | 000x000xx00x | 110100 |
| 419 | 000x000x0xx0 | 111011 |
| 420 | 000x000x0x0x | 111000 |
| 421 | 000x000x00xx | 111110 |
| 422 | 000x0000xxx0 | 100011 |
| 423 | 000x0000xx0x | 100000 |
| 424 | 000x0000x0xx | 100100 |
| 425 | 000x00000xxx | 101010 |
| 426 | 0000xxxx0000 * | 110000 |
| 427 | 0000xxx0x000 * | 101000 |
| 428 | 0000xxx00x00 * | 100100 |
| 429 | 0000xxx000x0 * | 100010 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซินโดรม |
|---|---|---|
| 430 | 0000xxx0000x * | 100001 |
| 431 | 0000xx0xx000 * | 011000 |
| 432 | 0000xx0x0x00 * | 010100 |
| 433 | 0000xx0x00x0 * | 010010 |
| 434 | 0000xx0x000x | 010001 |
| 435 | 0000xx00xx00 * | 001100 |
| 436 | 0000xx00x0x0 * | 001010 |
| 437 | 0000xx00x00x * | 001001 |
| 438 | 0000xx000xx0 | 000110 |
| 439 | 0000xx000x0x * | 000101 |
| 440 | 0000xx0000xx * | 000011 |
| 441 | 0000x0xxx000 | 111000 |
| 442 | 0000x0xx0x00 | 110100 |
| 443 | 0000x0xx00x0 | 110010 |
| 444 | 0000x0xx000x | 110001 |
| 445 | 0000x0x0xx00 | 101100 |
| 446 | 0000x0x0x0x0 | 101010 |
| 447 | 0000x0x0x00x | 101001 |
| 448 | 0000x0x00xx0 | 100110 |
| 449 | 0000x0x00x0x | 100101 |
| 450 | 0000x0x000xx | 100011 |
| 451 | 0000x00xxx00 | 011100 |
| 452 | 0000x00xx0x0 | 011010 |
| 453 | 0000x00xx00x | 011001 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 454 | 0000x00x0xx0 | 010110 |
| 455 | 0000x00x0x0x | 010101 |
| 456 | 0000x00x00xx | 010011 |
| 457 | 0000x000xxx0 | 001110 |
| 458 | 0000x000xx0x | 001101 |
| 459 | 0000x000x0xx | 001011 |
| 460 | 0000x0000xxx | 000111 |
| 461 | 00000xxxx000 | 111000 |
| 462 | 00000xxx0x00 | 110100 |
| 463 | 00000xxx00x0 | 110010 |
| 464 | 00000xxx000x | 110001 |
| 465 | 00000xx0xx00 | 101100 |
| 466 | 00000xx0x0x0 | 101010 |
| 467 | 00000xx0x00x | 101001 |
| 468 | 00000xx00xx0 | 100110 |
| 469 | 00000xx00x0x | 100101 |
| 470 | 00000xx000xx | 100011 |
| 471 | 00000x0xxx00 | 011100 |
| 472 | 00000x0xx0x0 | 011010 |
| 473 | 00000x0xx00x | 011001 |
| 474 | 00000x0x0xx0 | 010110 |
| 475 | 00000x0x0x0x | 010101 |
| 476 | 00000x0x00xx | 010011 |
| 477 | 00000x00xxx0 | 001110 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

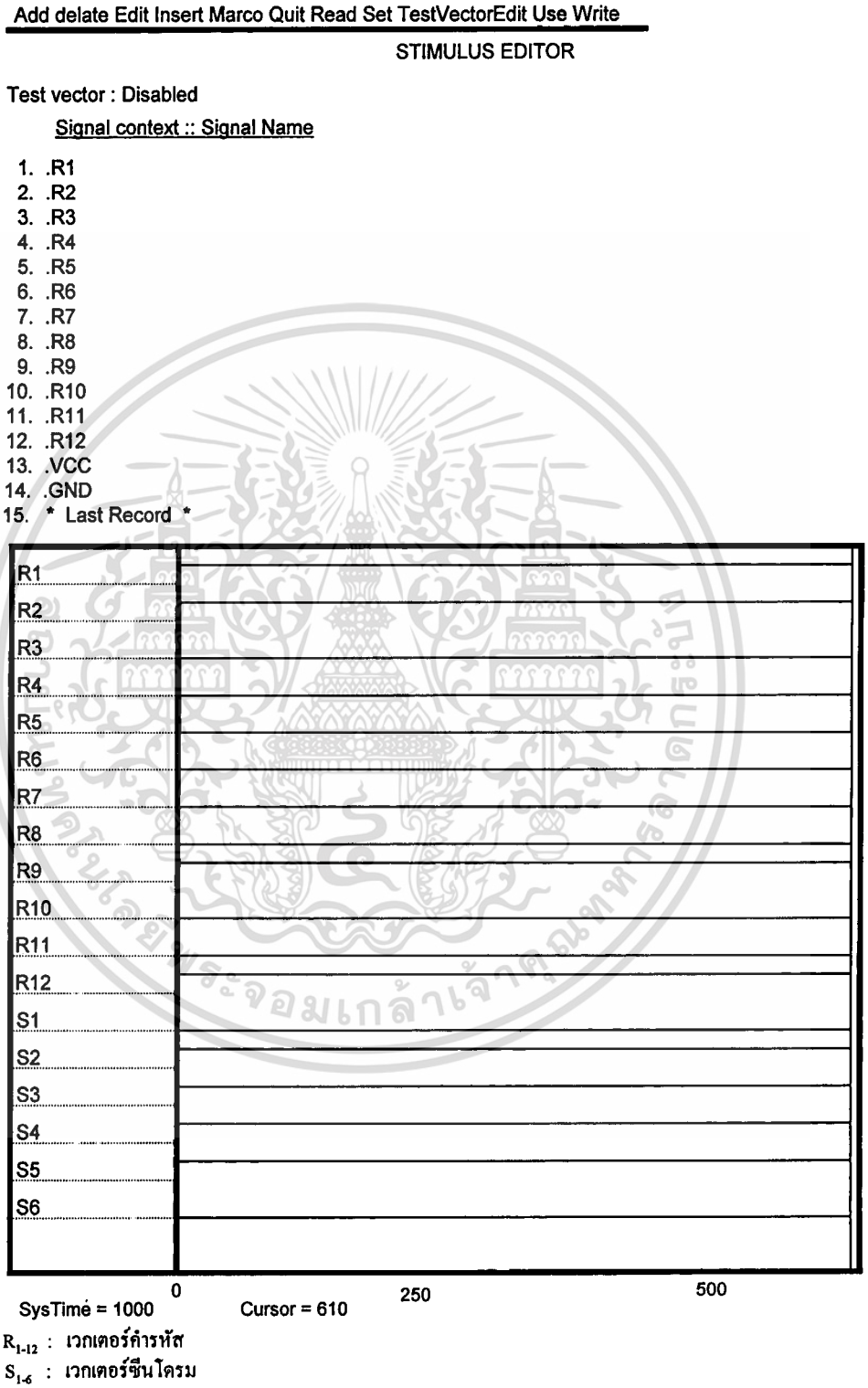
**ตารางที่ 4.9 (ต่อ)**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 4 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 478 | 00000x00xx0x | 001101 |
| 479 | 00000x00x0xx | 001011 |
| 480 | 00000x000xxx | 000111 |
| 481 | 000000xxxx00 | 111100 |
| 482 | 000000xxx0x0 | 111010 |
| 483 | 000000xxx00x | 111001 |
| 484 | 000000xx0xx0 | 110110 |
| 485 | 000000xx0x0x | 110101 |
| 486 | 000000xx00xx | 110011 |
| 487 | 000000x0xxx0 | 101110 |
| 488 | 000000x0xx0x | 101101 |
| 489 | 000000x0x0xx | 101011 |
| 490 | 000000x00xxx | 100111 |
| 491 | 0000000xxxx0 | 011110 |
| 492 | 0000000xxx0x | 011101 |
| 493 | 0000000xx0xx | 011011 |
| 494 | 0000000x0xxx | 010111 |
| 495 | 00000000xxxx | 001111 |

**ตารางที่ 4.9 (ต่อ)**

**กรณีที่หก** เวกเตอร์คำรหัสที่มีบิตผิด 5 บิต กำหนดให้ 110000001001 เป็นเวกเตอร์คำรหัสที่มีบิตผิดไป 5 บิต จากเวกเตอร์คำรหัสเดิม 100011000000 ทางด้านอินพุตจะได้ผลของเวกเตอร์ซีนโดรมเป็น 011110 ดังรูปที่ 4.38 และปรากฏรูปแบบบิตที่ผิดเป็น 010011001001 ดังรูปที่ 4.39 ซึ่งผลจากการแก้ไขเวกเตอร์คำรหัสที่รับเข้ามาเป็น 100011000000 ดังรูปที่ 3.40 แล้วรูปแบบบิตที่ผิด 5 บิต จะสามารถแก้ไขได้เพียง 16 รูปแบบเท่านั้น ดังตารางที่ 4.10

เอกสารนี้ [illegible] ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




$R_{1\text{-}12}$ : เวกเตอร์คำรหัส

$S_{1\text{-}6}$ : เวกเตอร์ซีนโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.38 ผลของเวกเตอร์ซีนโดรมที่ได้จากเวกเตอร์คำรหัส 100011000000**

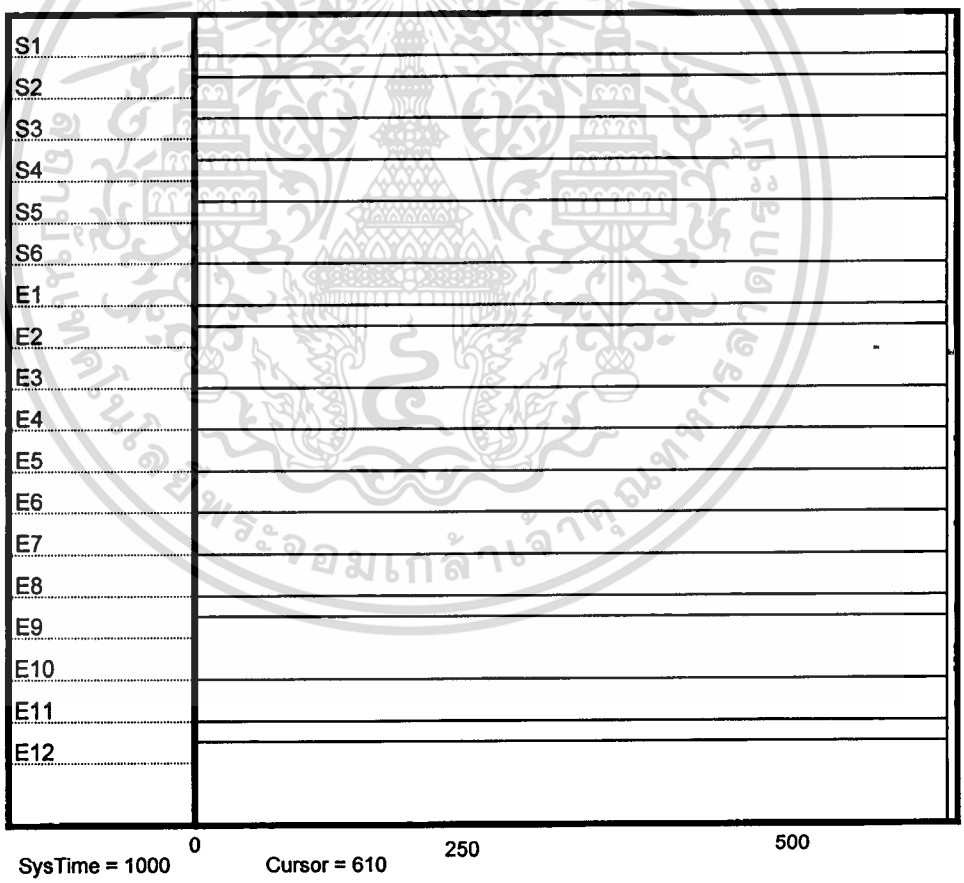
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




$S_{1\text{-}6}$ : เวกเตอร์ซีนโครม

$E_{1\text{-}12}$ : เวกเตอร์รูปแบบตำแหน่งบิตผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

**รูปที่ 4.39 ผลของรูปแบบบิตผิดจากเวกเตอร์คำรหัส 100011000000**

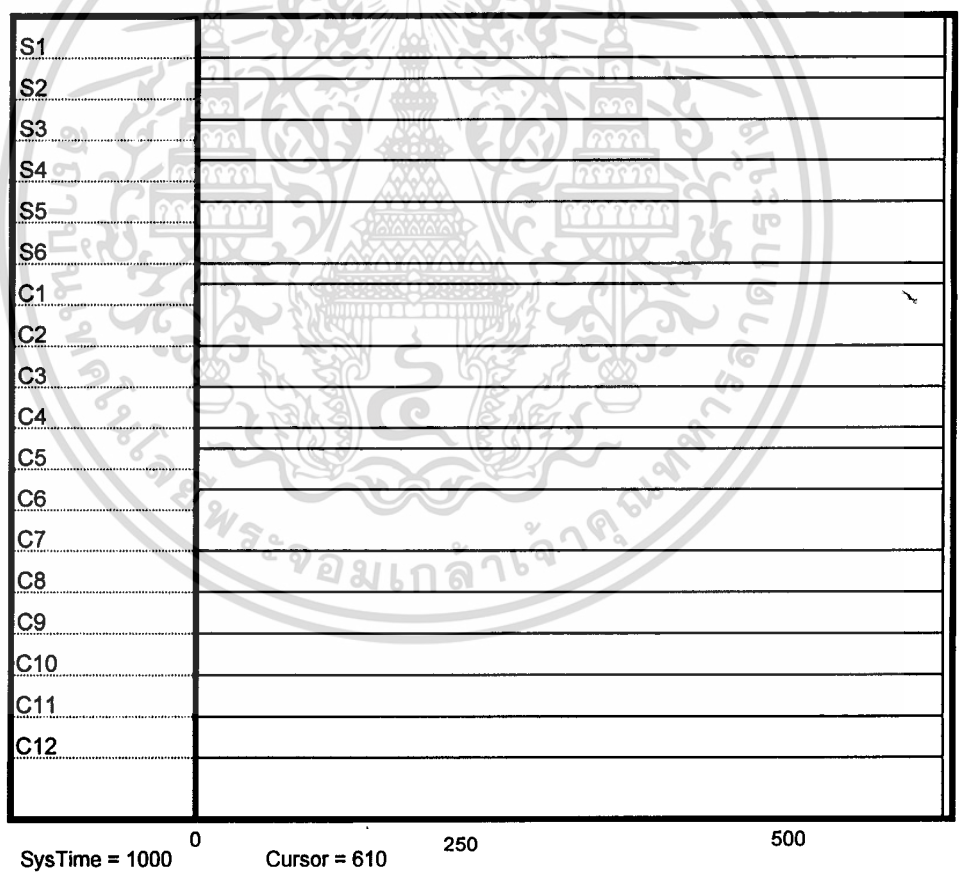
Add delate Edit Insert Marco Quit Read Set TestVectorEdit Use Write

STIMULUS EDITOR

Test vector : Disabled

Signal context :: Signal Name

1. .R1
2. .R2
3. .R3
4. .R4
5. .R5
6. .R6
7. .R7
8. .R8
9. .R9
10. .R10
11. .R11
12. .R12
13. .VCC
14. .GND
15. * Last Record *




$S_{1-6}$ : เวกเตอร์ซีนโดรม

$C_{1-12}$ : เวกเตอร์คำรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 4.40 ผลจากการแก้ไขบิตผิดของเวกเตอร์คำรหัส 100011000000**

| ลำดับที่ | รูปแบบตำแหน่งบิตที่ผิด 5 บิต | เวกเตอร์ซีนโดรม |
|---|---|---|
| 1 | xx00xxx00000 | 011100 |
| 2 | xx00xx00x000 | 110100 |
| 3 | x0x0xxx00000 | 110001 |
| 4 | x0x0xx00x000 | 011001 |
| 5 | x00xxxx00000 | 100110 |
| 6 | x00xxx00x000 | 001110 |
| 7 | x000xxxx0000 | 011011 |
| 8 | x000xxx00x00 | 001111 |
| 9 | x000xx0xx000 | 110011 |
| 10 | x000xx0x0x00 | 111111 |
| 11 | x000xx0x00x0 | 111001 |
| 12 | x000xx00xx00 | 100110 |
| 13 | 0x00xxx000x0 | 110101 |
| 14 | 0x00xxx0000x | 110110 |
| 15 | 0x00xx00x0x0 | 011101 |
| 16 | 0x00xx00x00x | 011110 |

x : ตำแหน่งบิตที่ผิด มีค่าเป็น 0 หรือ 1

**ตารางที่ 4.10 ผลของเวกเตอร์ซีนโดรมจากการกำหนดรูปแบบบิตที่ผิด 5 บิต**

## 4.2 การทดลองส่งข้อมูลผ่านชุดรับ-ส่งข้อมูลแบบอนุกรม

ในหัวข้อนี้เป็นการทดลองส่งคำรหัสจากชุดเข้ารหัสที่เป็นข้อมูลแบบขนานให้เป็นข้อมูลแบบอนุกรมไปยังชุดรับ ซึ่งจะแปลงข้อมูลแบบอนุกรมให้เป็นแบบขนาน แล้วส่งข้อมูลต่อไปยังชุดถอดรหัสโดยจะแบ่งการทดลองออกเป็น 2 ส่วน คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2.1 การทดลองระบบรับ-ส่งข้อมูล กับชุดวงจรเข้ารหัสและถอดรหัสภายนอก**

เป็นการทดลองโดยการนำชุดวงจรเข้ารหัสและวงจรถอดรหัส ที่ประกอบจากภายนอกชุดทดลองมาต่อเข้ากับชุดรับ-ส่งข้อมูลแบบอนุกรม ซึ่งมีขั้นตอนการทดลองดังต่อไปนี้

1. ทำการประกอบวงจรเข้ารหัสจากภายนอกแบบ (8,4) บิต หรือ (12,4) บิต

2. โยกสวิตช์ $Sw_3$ ไปที่ตำแหน่ง 1 เพื่อเลือกวงจรเข้ารหัสที่อยู่ภายนอก แล้วกดสวิตช์ $Sw_1$ เพื่อเลือกชุดเข้ารหัสขนาด (8,4) บิต หรือ (12,4) บิต โดยสังเกตจากแอลอีดีสีเหลือง หรือสีส้ม ตามลำดับทั้งภาครับและภาคส่ง

3. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ไปที่ตำแหน่ง 0

4. ทำการเชื่อมโยงผลของคำรหัสจากวงจรเข้ารหัสภายนอกมายังชุดทดลองที่จุด PIN 1 และให้เชื่อมโยงข้อมูลจากชุดทดลองไปยังงวงจรเข้ารหัสที่ต่อไว้ภายนอกที่จุด PIN 2

5. เลือกกดสวิตช์ Up หรือ Down เพื่อเลือกแบบของข้อมูลที่จะนำไปเข้ารหัสที่วงจรเข้ารหัสที่ต่อไว้ โดยดูที่แอลอีดีสีแดงแล้วสังเกตผลการทดลองของคำรหัสที่แอลอีดีสีเขียว ซึ่งพร้อมที่จะส่งไปยังภาครับ

6. ทำการประกอบวงจรถอดรหัสจากภายนอกแบบ (8,4) บิต หรือ (12,4) บิต

7. ทำการเชื่อมโยงผลของคำรหัสที่ได้รับมาจากภาคส่งของชุดทดลองไปยังวงจรถอดรหัสภายนอก โดยใช้ PIN 4

8. ทำการเชื่อมโยงเส้นทางทางการรับ-ส่งเวกเตอร์คำรหัสจากภาคส่งไปยังภาครับของชุดทดลองที่จุด PIN 3

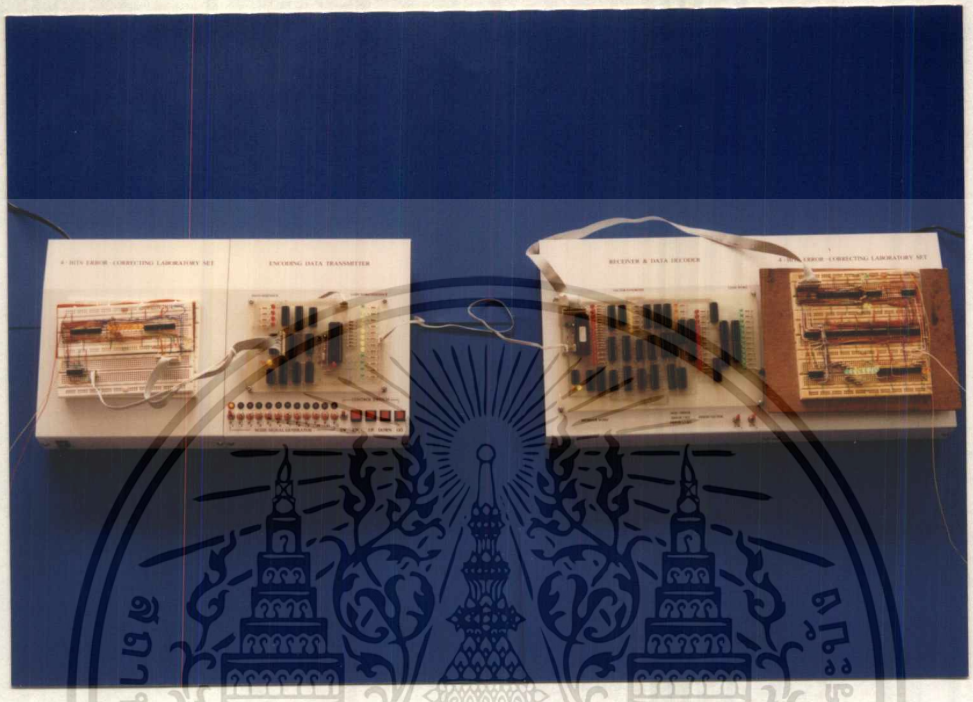
9. เลือกกดสวิตช์ Up หรือ Down เพื่อเลือกข้อมูลเข้ารหัสภายนอก เช่น กำหนดให้ $\underline{d}$ = 1111 แล้วโยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ไปที่ตำแหน่ง 0 ทุกบิตจะได้เวกเตอร์คำรหัสคำรหัสที่ภาคส่งของชุดทดลองเป็น $\underline{c}$ = 11111111

10. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ทางชุดเข้ารหัสของภาคส่งในชุดทดลองไปยังตำแหน่ง 1 หรือ 0 เพื่อเลือกตำแหน่งบิตที่ผิดของเวกเตอร์คำรหัส

11. กดสวิตช์ส่ง (Go Sw.) เพื่อส่งข้อมูลคำรหัสจากภาคส่งมายังภาครับข้อมูลคำรหัสแล้วสังเกตแอลอีดีแสดงผลการรับข้อมูลพร้อมกันที่ภาคส่งจะได้รับสัญญาณตอบรับจากภาครับ โดยแสดงผลที่แอลอีดีสีแดง ( Received LED ) ซึ่งจะกระพริบ 3 ครั้ง และผลของการถอดรหัสจะสังเกตได้จากแอลอีดีเวกเตอร์ซินโดรม (Vector Syndrome) แอลอีดีแสดงเวกเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งบิตที่ผิด ( Vector Error ) และเวกเตอร์คำรหัสที่ถูกแก้ไข ( Corrected Code Word ) ของวงจรถอดรหัสจากภายนอก



**รูปที่ 4.41 ภาพการทดลองระบบรับ-ส่งข้อมูล ที่ต่อร่วมกับวงจรชุดเข้ารหัสและถอดรหัสภายนอก**

## 4.2.2 การทดลองระบบรับ-ส่งข้อมูล กับชุดวงจรเข้ารหัสและถอดรหัสภายใน

เป็นการทดลองโดยการนำชุดวงจรเข้ารหัสและวงจรถอดรหัสทั้งขนาด (8,4) บิต และ (12,4) บิต มาเชื่อมต่อกับชุดระบบรับ-ส่งที่ควบคุมด้วยไมโครคอนโทรลเลอร์เบอร์ 87C51 ซึ่งมีขั้นตอนการทดลองดังนี้

1. โยกสวิตช์ $Sw_3$ ไปที่ 0 เพื่อเลือกใช้ชุดวงจรเข้ารหัสและวงจรถอดรหัสภายในชุดทดลองและกดสวิตช์ $Sw_1$ เพื่อเลือกชุดเข้ารหัสขนาด (8,4) บิต หรือ (12,4) บิต โดยสังเกตจากแอลอีดีสีเหลืองหรือสีส้มตามลำดับ ทั้งภาครับและภาคส่ง

2. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ไปที่ตำแหน่ง 0 ทุกบิต

3. เลือกกดสวิตช์ Up หรือ Down เพื่อเลือกข้อมูลที่จะนำไปเข้ารหัสยังชุดวงจรเข้ารหัสภายในชุดทดลอง สังเกตได้จากแอลอีดีสีแดง ( Data )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

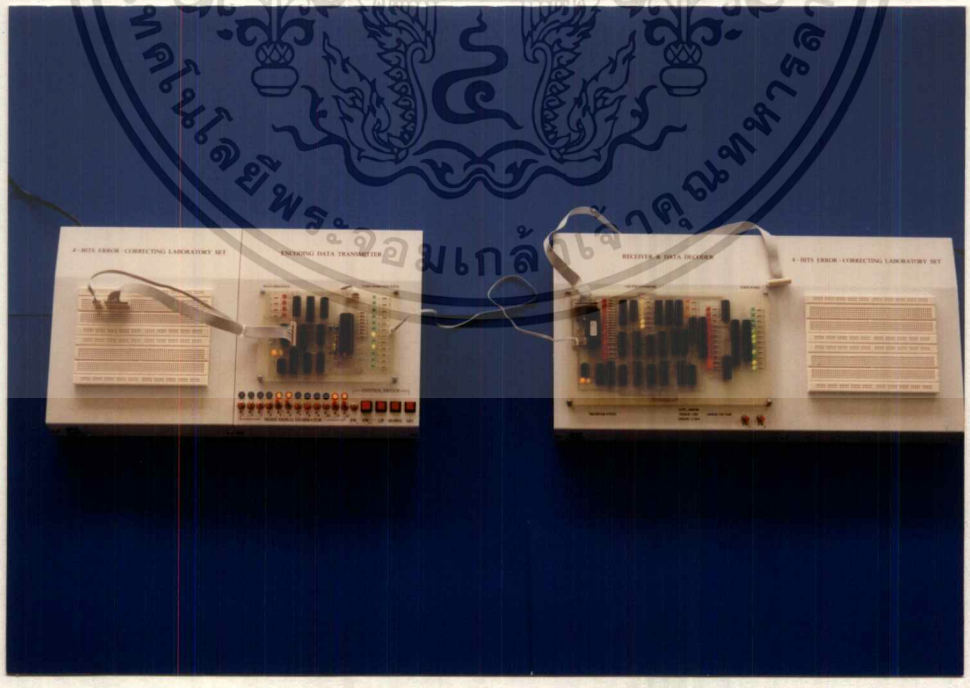
4. โยกสวิตช์ $Sw_3$ ของภาคถอดรหัสไปที่ตำแหน่ง 0 เพื่อเลือกใช้ชุดทดลองวงจรถอดรหัสและโยกสวิตช์ Sw1 ของภาคถอดรหัสไปที่ตำแหน่ง 1 เพื่อเลือกใช้ชุด (8,4) บิต หรือไปที่ตำแหน่ง 0 เพื่อใช้ชุด (12,4) บิต

5. ทำการเชื่อมโยงเส้นทางการรับ-ส่งเวกเตอร์คำรหัสจากภาคส่งไปยังภาครับของชุดทดลองที่จุด PIN 3

6. เลือกกดสวิตช์ Up หรือ Down เพื่อกำหนดข้อมูลที่จะนำมาเข้ารหัสที่ภาคเข้ารหัส เช่น กำหนดให้ $\underline{d}$ = 0000 จะได้คำรหัสที่ภาคเข้ารหัส $\underline{c}$ = 00000000 เป็นต้น

7. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ทางภาคเข้ารหัสสลับไปยังตำแหน่ง 1 หรือ 0 เพื่อกำหนดตำแหน่งบิตที่ผิดของคำรหัส

8. กดสวิตช์ส่ง ( Go Sw. ) ของภาคส่งข้อมูลคำรหัสเพื่อส่งเวกเตอร์คำรหัสมายังภาครับข้อมูลคำรหัส แล้วสังเกตผลของเวกเตอร์คำรหัสที่ได้รับทางแอลอีดีสีแดงพร้อมกันกับที่ภาคส่งจะได้รับสัญญาณตอบรับจากภาครับ โดยจะแสดงผลเป็นแอลอีดีสีแดง ( Received LED ) ซึ่งจะกระพริบ 3 ครั้ง และผลของการถอดรหัสจะสังเกตได้จากแอลอีดีแสดงเวกเตอร์ซินโดรมสีเหลือง, แอลอีดีสีแดงแสดงเวกเตอร์ตำแหน่งบิตที่ผิด และแอลอีดีแสดงเวกเตอร์คำรหัสที่ถูกแก้ไขสีเขียวของวงจรถอดรหัสจากภายใน



**รูปที่ 4.42 ภาพการทดลองระบบรับ-ส่งข้อมูล ที่ต่อร่วมกับวงจรชุดเข้ารหัสและถอดรหัสภายใน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

## บทสรุป วิจารณ์ และแนวทางในการพัฒนา

### 5.1 บทสรุป

จากชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต ในปริญญานิพนธ์ฉบับนี้ เป็นการนำทฤษฎีของรหัสระบบเชิงเส้นมาทำการออกแบบเป็นวงจรชุดเข้ารหัสและชุดถอดรหัสขนาด (8,4) บิต ซึ่งมีขีดความสามารถในการแก้ไขคำรหัสที่มีบิตผิด 1 บิต ได้ทุกรูปแบบ และตรวจจับคำรหัสที่มีบิตผิด 2 บิตได้ แต่ไม่สามารถตรวจจับคำรหัสที่มีบิตผิด 3 บิตได้ เนื่องจากข้อจำกัดทางทฤษฎีรหัสระบบเชิงเส้นที่มีรูปแบบของเวกเตอร์ซินโดรมที่จำกัด แต่อย่างไรก็ตาม คณะผู้จัดทำได้พยายามพัฒนาวิธีการถอดรหัสให้มีความสามารถในการแก้ไขความผิดพลาดของคำรหัสได้มากขึ้น โดยการเพิ่มจำนวนบิตตรวจสอบของเวกเตอร์คำรหัสที่ชุดเข้ารหัสจาก 8 บิตเป็น 12 บิต ซึ่งการเพิ่มจำนวนบิตตรวจสอบให้กับคำรหัสทำให้รูปแบบการแก้ไขคำรหัสที่มีบิตผิดได้มากกว่า 1 บิต จึงทำให้วงจรชุดใหม่ที่พัฒนาขึ้นมีขีดความสามารถในการแก้ไขคำรหัสได้มากขึ้น โดยจะแก้ไขคำรหัสที่มีบิตผิด 1 บิต ได้ทุกรูปแบบ แก้ไขคำรหัสที่มีบิตผิด 2 บิตได้ 57 รูปแบบจากทั้งหมด 66 รูปแบบ คิดเป็น 86.36 เปอร์เซ็นต์ แก้ไขคำรหัสที่มีบิตผิด 3 บิต ได้ 98 รูปแบบจากทั้งหมด 220 รูปแบบ คิดเป็น 44.54 เปอร์เซ็นต์ แก้ไขคำรหัสที่มีบิตผิด 4 บิตได้ 68 รูปแบบจาก 495 รูปแบบ คิดเป็น 13.73 เปอร์เซ็นต์และแก้ไขคำรหัสที่มีบิตผิด 5 บิตได้ถึง 16 รูปแบบ นอกจากนี้ยังได้นำชุดเข้ารหัสและชุดถอดรหัสนี้ไปใช้งานร่วมกับไมโครคอนโทรลเลอร์เบอร์ 87C51 เพื่อใช้รับ-ส่งข้อมูลแบบ Simplex และจัดทำเป็นชุดทดลองปฏิบัติการ สำหรับนักศึกษาภาควิชาครุศาสตร์วิศวกรรมด้วย ซึ่งผลจากการทำปริญญานิพนธ์นี้ ถือได้ว่าเป็นที่น่าพอใจตามจุดประสงค์ที่ได้กำหนดไว้

ถึงแม้ว่าปริญญานิพนธ์นี้ ในภาพรวมแล้วจะไม่ชัดเจนนัก คือ เป็นปริญญานิพนธ์ที่มุ่งเน้นในจุดเล็กๆโดยเริ่มต้นจากการศึกษาทฤษฎีและนำทฤษฎีที่ได้มาประยุกต์ออกแบบและจัดสร้างเป็นชุดทดลองเพื่อทดสอบผลทางทฤษฎี แต่ปริญญานิพนธ์นี้ก็มีลักษณะของการจัดสร้างอย่างมีระบบและมีหลักการที่พยายามจะพิสูจน์ทฤษฎี และค้นหาสิ่งใหม่ด้วยการนำทฤษฎีมาประยุกต์ออกแบบ ซึ่งคณะผู้จัดทำหวังไว้ว่าอย่างน้อย ปริญญานิพนธ์นี้คงจะเป็นอีกรูปแบบหนึ่งที่ใช้เป็นแนวทางในการนำทฤษฎีต่างๆ มาออกแบบและจัดสร้างรวมถึงการพัฒนาเพื่อให้ได้นวัตกรรมใหม่ๆ ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ปัญหาและการแก้ไข

### 5.2.1 ปัญหา

1. ความสามารถของทฤษฎีรหัสที่นำมาใช้ไม่ครอบคลุมขีดความสามารถและขอบเขตของปริญญานิพนธ์ที่ได้กำหนดไว้ตั้งแต่ต้น

2. การจำลองการทำงานของวงจรถอดรหัสขนาด (12,4) บิต ไม่สามารถทำการจำลองทั้งวงจรได้ในครั้งเดียว เนื่องจากรายละเอียดและอุปกรณ์ที่ใช้มีมากเกินไป

3. การศึกษาการใช้งานของไอซีไมโครคอนโทรลเลอร์เบอร์ 87C51 ต้องใช้เวลาในการศึกษาที่เพิ่มเติม และการสร้างชุดคำสั่งรวมถึงการบันทึกชุดคำสั่งลงในไมโครคอนโทรลเลอร์มีความยุ่งยาก เนื่องจากขาดประสบการณ์

4. ขาดคู่มือการใช้งานอุปกรณ์ PLD ซึ่งจำเป็นต้องนำมาใช้ในการออกแบบวงจรและสร้างชุดคำสั่ง

### 5.2.2 การแก้ไข

1. ทำการศึกษาทฤษฎีรหัสอื่นเพิ่มเติม ได้แก่ ทฤษฎีรหัสการประสาน ปรากฏว่าขีดความสามารถของทฤษฎีก็มีจำกัดเช่นกัน คือ ถ้าต้องการให้มีความสามารถของชุดถอดรหัสในการแก้ไขบิตที่ผิดเพิ่มมากขึ้น ก็จะต้องเพิ่มจำนวนบิตในคำรหัส ซึ่งคล้ายกับทฤษฎีรหัสกลุ่มที่ใช้ในปริญญานิพนธ์นี้เพียงแต่วิธีการต่างกัน ฉะนั้นการแก้ไขจึงได้ทำการเพิ่มจำนวนบิตตรวจสอบในคำรหัส ซึ่งจะทำให้ชุดถอดรหัสสามารถแก้ไขบิตผิดของคำรหัสได้เพิ่มขึ้น

2. ทำการแยกส่วนของวงจรถอดรหัสในการจำลองการทำงาน ได้แก่ ส่วนวงจรสร้างซินโดรม, ส่วนวงจรแสดงตำแหน่งบิตผิด และส่วนวงจรแก้ไขบิตที่ผิด

3. ได้ขอความร่วมมือจากนักศึกษาคณะวิศวกรรม ภาควิชาคอมพิวเตอร์ เพื่อช่วยให้คำแนะนำการใช้งานรวมทั้งตรวจสอบแก้ไขชุดคำสั่งที่ใช้กับตัวไอซีไมโครคอนโทรลเลอร์เบอร์ 87C51 ของปริญญานิพนธ์นี้

4. ได้รับความช่วยเหลือจากนักศึกษาคณะครุศาสตร์อุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรมที่เคยทำงานเกี่ยวกับอุปกรณ์ PLD ทางด้านคู่มือใช้งานรวมถึงตัวอุปกรณ์ PLD ที่ใช้ในปริญญานิพนธ์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.3 แนวทางในการพัฒนา

เนื่องจากทฤษฎีรหัสระบบเชิงเส้นมีความสามารถที่มีขอบเขตจำกัด ดังที่ได้กล่าวมาแล้ว ดังนั้นในการพัฒนาเพื่อให้คำรหัสมีความสามารถเพิ่มขึ้นนั้นเป็นไปได้ยาก แต่ก็จะสามารถนำทฤษฎีรหัสระบบเชิงเส้นนี้ไปประยุกต์ใช้งานซึ่งพอที่จะนำไปเป็นแนวทางได้ดังนี้

1.การลดความยุ่งยากและขนาดของวงจร ทำได้โดย สามารถเขียนชุดคำสั่งและบันทึกชุดคำสั่งนั้นลงในหน่วยความจำภายในของไอซีไมโครคอนโทรลเลอร์ได้

2.พัฒนาลักษณะชุดทดลอง โดยการนำคำสั่งที่มีอยู่บันทึกลงในหน่วยความจำภายนอกเพื่อความสะดวกในการเปลี่ยนแปลงแก้ไข

3.นำไปประยุกต์ใช้กับสัญญาณ โดยลักษณะของ Digital Discamble

4.นำความรู้พื้นฐานของทฤษฎีรหัสระบบเชิงเส้นนี้ไปศึกษาและพัฒนาทฤษฎีรหัสกลุ่ม ในแบบต่างๆต่อไป เช่น Cyclic Codes Theory , BCH Codes และ Read Soloman Codes เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


เกียรติศักดิ์ ศรีพิมานวัฒน์."การแก้ไขความผิดพลาดของการสื่อสารร่วมสัญญาณเสียงและข้อมูล",วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิศวกรรมไฟฟ้า บัณฑิตวิยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2536

บัณฑิต โรจน์อารยานนท์. หลักการไฟฟ้าสื่อสาร. กรุงเทพมหานคร : จุฬาลงกรณ์มหาวิยาลัย 2537

บริษัท ซีเอ็ดยูเคชั่น. คู่มือ/เทียบเบอร์ไอซีTTL. กรุงเทพมหานคร : เอช-เอ็นการพิมพ์,2532

บริษัท อินเทล.MCS-51 Microcontrollers User's Manual. กรุงเทพมหานคร : บริษัทอีทีที

ประสิทธิ์ ประพิฒมงคลการ. หลักการระบบสื่อสาร. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น,2533

ฟูศักดิ์ ชีวสุวิทย์. การแก้ไขรหัสที่ผิด. คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2533

มัลลิกา ศรีกมล. ทฤษฎีรหัส. ภาควิชาคณิตศาสตร์ มหาวิทยาลัยเชียงใหม่,2537

สุเจตน์ จันทวงษ์. ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น,2535

Rhee Man Young. Error Correcting Coding Theory. New York : McGrew-Hill Publishing Company,1989

Shulin and Danial J. Costello,JR. Error Control Coding. New Jersey : Prentice-hall Inc.,1983



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การทดลองที่ 1

## เรื่อง การเข้ารหัสช่องสัญญาณแบบรหัสระบบเชิงเส้นขนาด (8,4) บิต

**วิชา การทดลองการปฏิบัติการวิศวกรรมโทรคมนาคม รหัส 0331106 หน่วยกิต 3**

**สาขาวิชา วิศวกรรมโทรคมนาคม ภาควิชา ครุศาสตร์วิศวกรรม**

**คณะ ครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง**

**วัตถุประสงค์**

1. เพื่อให้นักศึกษาสามารถคำนวณหาค่าคำรหัสของชุดวงจรเข้ารหัสช่องสัญญาณได้
2. เพื่อให้นักศึกษาสามารถต่อวงจรภาคเข้ารหัสช่องสัญญาณได้
3. เพื่อให้นักศึกษาสามารถบันทึกผลของคำรหัสจากการคำนวณและการทดลองได้

**ทฤษฎีและหลักการ**

ในการเข้ารหัสช่องสัญญาณแบบระบบเชิงเส้นนั้น จะสามารถทำการเข้ารหัส โดยความสัมพันธ์ของ

$$\underline{c} = \underline{d}\,G \tag{1}$$

โดยที่ $\underline{c}$ คือ ลำดับของของคำรหัสขนาด n-ทูเปิ้ลส์ ที่ได้จากการเข้ารหัส

$\underline{d}$ คือ ลำดับของข้อมูลขนาด k-ทูเปิ้ลส์ ที่ถูกนำมาเข้ารหัส

G คือ เมตริกซ์ตัวกำเนิด [ Ik : Pk * (n-k) ] ที่กำหนดจากรูปแบบของข้อมูล

ดังนั้นจากสมการที่ (1) จะเขียนใหม่ได้เป็น

$$\underline{c} = d_1, d_2 \ldots, d_k \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} \tag{2}$$

$$( c_1, c_2, \ldots, c_k, c_{n-k}, \ldots, c_n ) = ( d_1, d_2, \ldots, d_k ) [I_k : P_{k*(n-k)}] \tag{3}$$

$$( c_1, c_2, \ldots, c_k, c_{n-k}, \ldots, c_n ) = ( d_1, d_2, \ldots, d_k ) \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ 0 & 1 & 0 & 0 & \cdots & p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ 0 & 0 & 1 & 0 & \cdots & p_{3,1} & p_{3,2} & \cdots & p_{3,n-k} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{bmatrix} \tag{4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากำหนดให้

- ลำดับข้อมูล ; $\underline{d}$ มีขนาด 4 บิต คือ $d_1, d_2, d_3, d_4$
- เมตริกซ์ตัวกำเนิด G มีค่าเป็น

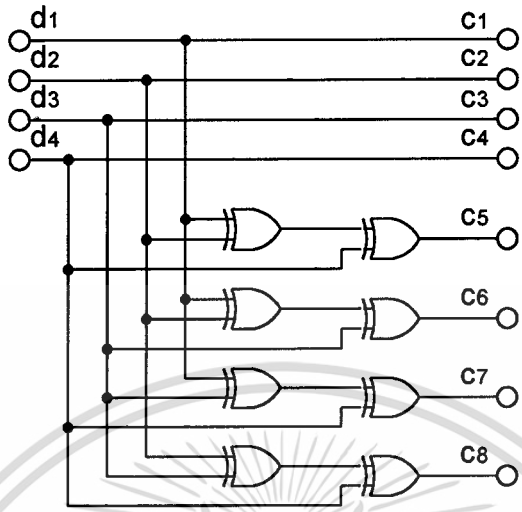
$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (5)$$

-ลำดับคำรหัสที่ได้จากการเข้ารหัส $\underline{c}$ มีขนาด 8 บิต $c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8$ จะสามารถแสดงการคำนวณหาคำรหัสได้ดังนี้

$$\begin{aligned} c_n &= (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8) \\ c_1 &= d_1 \\ c_2 &= d_2 \\ c_3 &= d_3 \\ c_4 &= d_4 \\ c_5 &= d_1 \oplus d_2 \oplus d_4 \\ c_6 &= d_1 \oplus d_2 \oplus d_3 \\ c_7 &= d_1 \oplus d_3 \oplus d_4 \\ c_8 &= d_2 \oplus d_3 \oplus d_4 \end{aligned} \quad (6)$$

**หมายเหตุ** $\oplus$ เป็นการกระทำทางคณิตศาสตร์แบบเอ็กคลูซีฟออร์เกต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป วงจรเข้ารหัส (8,4) บิต**

**เครื่องมือและอุปกรณ์**

| | |
|---|---|
| 1. แหล่งจ่ายไฟตรง 5 V. | 1 เครื่อง |
| 2. มัลติมิเตอร์ | 1 ตัว |
| 3. ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดข้อมูลขนาด 4 บิต | 1 ชุด |
| 4. ไอซีเบอร์ 74LS86 | 2 ตัว |
| 5. ไอซีเบอร์ 74LS32 | 1 ตัว |
| 6. สายคีบและสายต่อวงจร | 1 ชุด |

**ลำดับการทดลอง**

1. จงคำนวณหาคำรหัส ( Code Word ; $\underline{c}$ ) ขนาด (8,4) บิตจากเมตริกซ์ข้างล่างนี้ โดยคำนวณและบันทึกผลในตารางที่ 1

$$c = (d_1, d_2, d_3, d_4)\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ข่าวสารข้อมูล ( Information Data ) | คำรหัส ( Code Word ) |
|---|---|
| $d_1$ $d_2$ $d_3$ $d_4$ | $c_1$ $c_2$ $c_3$ $c_4$ $c_5$ $c_6$ $c_7$ $c_8$ |
| 0 0 0 0 | |
| 0 0 0 1 | |
| 0 0 1 0 | |
| 0 0 1 1 | |
| 0 1 0 0 | |
| 0 1 0 1 | |
| 0 1 1 0 | |
| 0 1 1 1 | |
| 1 0 0 0 | |
| 1 0 0 1 | |
| 1 0 1 0 | |
| 1 0 1 1 | |
| 1 1 0 0 | |
| 1 1 0 1 | |
| 1 1 1 0 | |
| 1 1 1 1 | |

**ตารางที่ 1 ผลที่ได้จากการคำนวณสมการเข้ารหัส**

2. ต่อวงจรเข้ารหัสจากภายนอกแบบ (8,4) บิต ตามวงจรรูปที่ 1

2.1 โยกสวิตช์ 3 ; $Sw_3$ ไปที่ตำแหน่ง 1 เพื่อเลือกวงจรเข้ารหัสที่อยู่ภายนอกและกดสวิตช์1 ; $Sw_1$ ของวงจรภาคเข้ารหัสไปที่ชุด (8,4) บิต ( หลอด LED สีเขียวจะติด )

2.2 โยกสวิตช์ป้อนสัญญาณรบกวน ; $Sw.e_1$-$e_{12}$ ไปที่ตำแหน่ง 0 ให้หมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทำการเชื่อมโยงข้อมูลจากชุดทดลองไปยังวงจรเข้ารหัส (8,4) บิต ที่ต่อไว้ภายนอกโดยมี PIN 1 เป็นตัวเชื่อมโยงและทำการเชื่อมโยงผลของคำรหัสจากวงจรเข้ารหัสภายนอกมายังชุดทดลอง โดยมี PIN 2 เป็นตัวเชื่อมโยง

2.4 เลือกกดสวิตช์ Up หรือDownเพื่อเลือกข้อมูลที่จะนำไปเข้ารหัสที่วงจรเข้ารหัส (8,4) บิต ที่ต่อไว้โดยดูจาก LED สีแดงแล้วบันทึกผลของคำรหัสที่ได้จาก LED สีเขียว ลงในตารางที่ 2

| ข่าวสารข้อมูล ( Information Data ) | คำรหัส ( Code Word ) |
|---|---|
| $d_1$ $d_2$ $d_3$ $d_4$ | $c_1$ $c_2$ $c_3$ $c_4$ $c_5$ $c_6$ $c_7$ $c_8$ |
| 0 0 0 0 | |
| 0 0 0 1 | |
| 0 0 1 0 | |
| 0 0 1 1 | |
| 0 1 0 0 | |
| 0 1 0 1 | |
| 0 1 1 0 | |
| 0 1 1 1 | |
| 1 0 0 0 | |
| 1 0 0 1 | |
| 1 0 1 0 | |
| 1 0 1 1 | |
| 1 1 0 0 | |
| 1 1 0 1 | |
| 1 1 1 0 | |
| 1 1 1 1 | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2 ผลของคำรหัสที่ได้จากการต่อวงจรเข้ารหัส (8,4) บิต**

# การทดลองที่ 2

## เรื่อง การถอดรหัสช่องสัญญาณของรหัสระบบเชิงเส้น ขนาด (8,4) บิต

**วิชา การทดลองการปฏิบัติการวิศวกรรมโทรคมนาคม รหัส 0331106 หน่วยกิต 3**

**สาขาวิชา วิศวกรรมโทรคมนาคม ภาควิชา ครุศาสตร์วิศวกรรม**

**คณะ ครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง**

**วัตถุประสงค์**

1. เพื่อให้นักศึกษาสามารถ คำนวนหาค่าเวกเตอร์ซินโดรมของคำรหัสที่ผิดพลาดได้
2. เพื่อให้นักศึกษาสามารถ ต่อวงจร เวกเตอร์ซีนโดรม , ตรวจจับ และวงจรแก้ไขบิตที่ผิดพลาดได้
3. เพื่อให้นักศึกษาสามารถบันทึกผลของค่าต่าง ๆ ที่ได้จากการคำนวนและการทดลองได้

**ทฤษฎีและหลักการ**

จากความสัมพันธ์ของการเข้ารหัสช่องสัญญาณ

$$\underline{c} = \underline{d}\,G$$

ซึ่ง G เป็นแมตริกซ์ตัวกำเนิด

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & p_{1,1} & p_{1,2} & \cdots & p_{1,n-k} \\ 0 & 1 & 0 & 0 & \cdots & p_{2,1} & p_{2,2} & \cdots & p_{2,n-k} \\ 0 & 0 & 1 & 0 & \cdots & p_{3,1} & p_{3,2} & \cdots & p_{3,n-k} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p_{k,1} & p_{k,2} & \cdots & p_{k,n-k} \end{bmatrix}$$

จะได้แมตริกซ์ตรวจสอบ ( Parity Check Matrix ; H )

$$H = \begin{bmatrix} p_{1,1} & p_{2,1} & \cdots & p_{k,1} & 1 & 0 & \cdots & 0 \\ p_{1,2} & p_{2,2} & \cdots & p_{k,2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 0 & 0 & \cdots & 1 \end{bmatrix} \qquad (1)$$

$$H = [\,P^{T}(n\text{-}k) : I\,(n\text{-}k)\,] \qquad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนเวกเตอร์ซีนโดรม ( Vector Syndrome ; $\underline{s}$ ) หาได้จาก

$$\underline{s} = (s_1, s_2, s_3, \ldots, s_{n-k}) = rH^T \quad (3)$$

โดยที่ r คือ คำรหัสที่รับเข้าเป็นคำรหัสเดียวกันกับชุดเข้ารหัส แต่อาจจะมีรหัสบิตผิด $\underline{e}$ นั่นคือ $r \neq c$ จะได้

$$\underline{s} = (\underline{c} + \underline{e})H^T \quad (4)$$

ถ้ากำหนดให้

- คำรหัสที่รับมาถอดรหัส ; $\underline{r}$ มีขนาด 8 บิต คือ $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$
- ทรานซ์โพสเมตริกซ์ตรวจสอบ ; $H^T$ มีค่าดังนี้

$$H^T = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$\underline{s} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

จะได้

$$s_1 = r_1 \oplus r_2 \oplus r_4 \oplus r_5$$

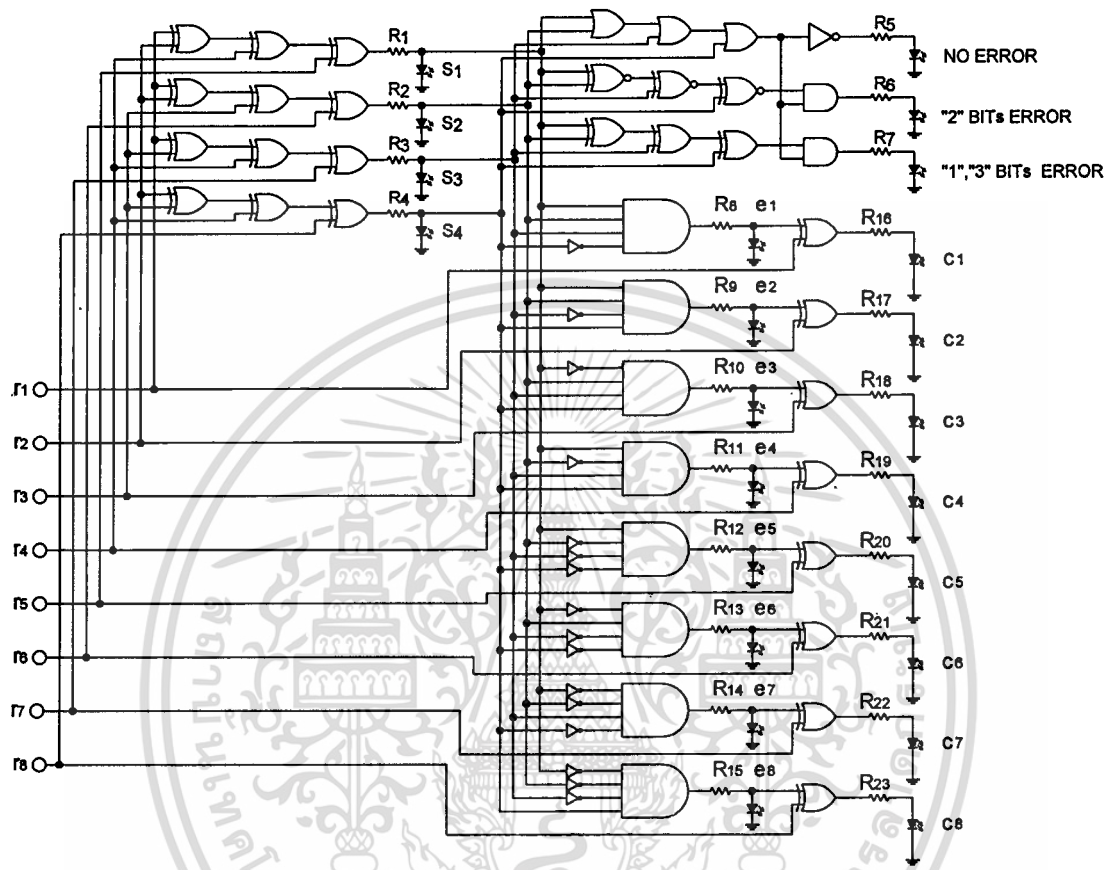
$$s_2 = r_1 \oplus r_2 \oplus r_3 \oplus r_6$$

$$s_3 = r_1 \oplus r_3 \oplus r_4 \oplus r_7$$

$$s_4 = r_2 \oplus r_3 \oplus r_4 \oplus r_8 \quad (6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจากสมการ $s_1$ - $s_4$ จึงเขียนเป็นวงจรถอดรหัสได้ คือ




**รูปที่ 1 วงจรถอดรหัส (8,4)**

**เครื่องมือและอุปกรณ์**

| | | |
|---|---|---|
| 1. แหล่งจ่ายไฟตรง 5 โวลท์ | 1 | เครื่อง |
| 2. มัลติมิเตอร์ | 1 | เครื่อง |
| 3. ชุดทดลองการตรวจจับและแก้ไขความผิดพลาดของข้อมูลขนาด 4 บิต | 1 | ชุด |
| 4. ไอซีเบอร์ 74LS86, 74LS08 , 74LS04 | | |
| 5. สายคีบและสายต่อวงจร | 1 | ชุด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ลำดับขั้นการทดลอง**

1. จงคำนวณหาค่าเวกเตอร์ซีนโดรม ; $\underline{s}$ หากคำรหัสที่รับเข้ามา ; $\underline{r}$ แบบถูกต้องเป็น 11111111 โดยสูตรคำนวณคือ

$$\underline{s} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8) \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

| คำรหัส ( Received Word ) | | | | | | | | เวกเตอร์ซีนโดรม | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ | $r_7$ | $r_8$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | | | | |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | | | |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | | | | |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | | | | |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | | | | |

**ตารางที่ 1 ผลของเวกเตอร์ซีนโดรมที่ได้จากการคำนวณ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

| คำรหัส ( Received Word ) | | | | | | | | เวกเตอร์ซีนโดรม | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ | $r_7$ | $r_8$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | | | | |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | | | | |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | | | | |

**ตารางที่ 1 (ต่อ)**

2. ต่อวงจรถอดรหัสจากภายนอกแบบ (8,4) บิต ตามรูปที่ 1

2.1 โยกสวิตช์ $Sw_3$ ไปที่ตำแหน่ง 1 เพื่อเลือกวงจรถอดรหัสที่อยู่ภายนอกและโยกสวิตช์ $Sw_1$ ของวงจรภาคถอดรหัสไปที่ตำแหน่ง 1 เพื่อเลือกชุด (8,4) บิต ( หลอด LED สีเขียวจะติด )

2.2 ทำการเชื่อมโยงผลของการเข้ารหัสจากภาคเข้ารหัสกับภาคถอดรหัสเข้าด้วยกัน โดยใช้ชุดซิงโครไนซ์ (Syncronize) ของชุดทดลองไมโครคอนโทรลเลอร์ MCS 87C51 โดยมี PIN 3 เป็นตัวเชื่อมโยง

2.3 ทำการเชื่อมโยงผลของคำรหัสที่รับมาจาก MCS 87C51 ของชุดทดลองถอดรหัสไปยังวงจรถอดรหัสภายนอกโดยมี PIN 4 เป็นตัวเชื่อมโยง

2.4 เลือกกดปุ่ม Up หรือ Down เพื่อเลือกข้อมูลเข้ารหัสที่วงจรเข้ารหัส (8,4)บิตภายนอกเช่นที่ $\underline{d}$ = 1111แล้วโยกสวิทช์ผิดพลาด $Sw.e_1$- $e_{12}$ ไปที่ตำแหน่ง 0 คำรหัสที่ภาคเข้ารหัส $\underline{c}$ = 11111111

3. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$- $e_{12}$ ทางภาคเข้ารหัสสลับไปยังตำแหน่ง 1 หรือ 0 ดังตารางบันทึกผลการทดลองที่ 2

4. กดสวิทช์ส่ง ( Go Sw. ) เพื่อส่งข้อมูลจากภาคเข้ารหัสมายังภาคถอดรหัสแล้วบันทึกผลของ เวกเตอร์ซีนโดรม ; $\underline{s}$ และรูปแบบตำแหน่งบิตที่ผิด E จาก LED สีแดงลงในตารางบันทึกผลการทดลองที่ 2 และ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สวิตช์ผิดพลาด | | | | | | | | เวกเตอร์ซีนโดรม | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | | | | |

**ตารางที่ 2 ผลของเวกเตอร์ซีนโดรมที่เกิดจากการผิดพลาดที่ภาคเข้ารหัส**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สวิตชที่ผิดพลาด | | | | | | | | รูปแบบตำแหน่งบิตผิด ( 0 หรือ 1 ) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | | | | | | | | |

**ตารางที่ 3 รูปแบบตำแหน่งที่ผิดพลาด E**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การทดลองที่ 3

## เรื่อง ชุดทดลองการเข้ารหัสและถอดรหัสช่องสัญญาณ แบบรหัสเชิงเส้นขนาด (8,4) และ (12,4) บิต

**วิชา การทดลองการปฏิบัติการวิศวกรรมโทรคมนาคม รหัส 0331106 หน่วยกิต 3**

**สาขาวิชา วิศวกรรมโทรคมนาคม ภาควิชา ครุศาสตร์วิศวกรรม**

**คณะ ครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง**

**วัตถุประสงค์**

1. เพื่อให้นักศึกษาสามารถใช้ชุดทดลองวงจรเข้ารหัสได้
2. เพื่อให้นักศึกษาสามารถอธิบายหลักการเข้ารหัส และถอดรหัสระบบเชิงเส้น
3. เพื่อให้นักศึกษาสามารถอธิบายหลักการซิงโคไนซ์ได้

**ทฤษฎีและหลักการ**

จากการคำนวณหาค่าคำรหัส และเวกเตอร์ซินโดรมของวงจรเข้ารหัสและถอดรหัสช่องสัญญาณแบบรหัสเชิงเส้นขนาด (8,4) บิต ดังในใบงานการทดลองที่ 1 และ 2 นั้น รูปแบบของชุดทดลองการเข้ารหัสขนาด (8,4) บิต หรือการทดลองที่ 3 นี้ ก็จะมีเมตริกซ์ตัวกำเนิดที่เหมือนกับใบงานการทดลองที่ 1 และ 2 ด้วยคือ

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

ส่วนในชุดพัฒนาการเข้ารหัสและถอดรหัสช่องสัญญาณแบบรหัสเชิงเส้นนั้น จะมีการพัฒนาให้เข้ารหัสได้ (12,4) บิต โดยให้ $d_1$ - $d_4$ เป็นข่าวสารข้อมูล ( Information Data ) ที่จะถูกเข้ารหัสและ $d_5$,$d_6$ จะถูกเซตให้เป็น 1 ตลอด เพื่อนำมาช่วยในการเข้ารหัสแบบรหัสเชิงเส้นจึงได้เมตริกซ์ตัวกำเนิด ( Generator Matrix ) ดังสมการคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

ดังนั้นผลของคำรหัสและผลของเวกเตอร์ซีนโดรมของชุดทดลองการเข้ารหัสช่องสัญญาณแบบรหัสระบบเชิงเส้นขนาด (8,4) บิต และ (12,4) บิตนี้ จะมีสมการการคำนวณคล้ายกับใบงานการทดลองที่ 1 และ 2

**เครื่องมือและอุปกรณ์**

1. ชุดทดลองการเข้ารหัสและถอดรหัสช่องสัญญาณแบบรหัสเชิงเส้นขนาด (8,4) และ (12,4) บิต

**ลำดับขั้นการทดลอง ภาคเข้ารหัสและภาคถอดรหัสขนาด (8,4) บิต**

1. โยกสวิตซ์ $Sw_3$ ไปที่ 0 เพื่อเลือกใช้ชุดทดลองวงจรเข้ารหัส และกดสวิตซ์ $Sw_1$ ไปเลือกชุด (8,4) บิต แล้วบันทึกผลการทดลองคำรหัส ; $\underline{c}$ ลงในตารางที่ 1

2. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ไปที่ตำแหน่ง 0 ให้หมด

3. เลือกกดสวิตช์ Up หรือ Down เลือกข้อมูลที่จะนำไปเข้ารหัสชุดวงจรขนาด (8,4) บิต แล้วบันทึกผลการทดลองคำรหัส ; $\underline{c}$ ลงในตารางที่ 1

| ข่าวสารข้อมูล ( Information Data ) | คำรหัส ( Code Word ) |
|---|---|
| $d_1$ $d_2$ $d_3$ $d_4$ | $c_1$ $c_2$ $c_3$ $c_4$ $c_5$ $c_6$ $c_7$ $c_8$ |
| 0 0 0 0 | |
| 0 0 0 1 | |
| 0 0 1 0 | |
| 0 0 1 1 | |
| 0 1 0 0 | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

**ตารางที่ 1 ผลของคำรหัสที่ได้จากชุดวงจรเข้ารหัส (8,4) บิต**

| ข่าวสารข้อมูล ( Information Data ) | | | | คำรหัส ( Code Word ) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $d_1$ | $d_2$ | $d_3$ | $d_4$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ | $c_8$ |
| 0 | 1 | 0 | 1 | | | | | | | | |
| 0 | 1 | 1 | 0 | | | | | | | | |
| 0 | 1 | 1 | 1 | | | | | | | | |
| 1 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 0 | 1 | | | | | | | | |
| 1 | 0 | 1 | 0 | | | | | | | | |
| 1 | 0 | 1 | 1 | | | | | | | | |
| 1 | 1 | 0 | 0 | | | | | | | | |
| 1 | 1 | 0 | 1 | | | | | | | | |
| 1 | 1 | 1 | 0 | | | | | | | | |
| 1 | 1 | 1 | 1 | | | | | | | | |

**ตารางที่ 1 (ต่อ)**

4.โยกสวิตช์ $Sw_3$ ของภาคถอดรหัสไปที่ตำแหน่ง 0 เพื่อเลือกใช้ชุดทดลองถอดรหัส และโยกสวิตช์ $Sw_1$ ของวงจรภาคถอดรหัสไปที่ 1 เพื่อเลือกใช้ชุดทดลองขนาด (8,4) บิต

5. ทำการเชื่อมโยงผลของคำรหัสจากภาคเข้ารหัสกับภาคถอดรหัสเข้าด้วยกัน โดยใช้ชุดเข้าจังหวะ ( Synchronize ) ของชุดทดลองไมโครคอนโทรลเลอร์ MCS 87C51 โดยมี PIN 3 เป็นตัวเชื่อมโยง

6. เลือกกดปุ่ม Up หรือ Down เพื่อเลือกข้อมูลเข้ารหัสที่ภาคเข้ารหัส (8,4) บิต เช่นที่ $\underline{d}$ = 0000 จะได้คำรหัสที่ภาคเข้ารหัส $\underline{c}$ = 0000 0000

7. โยกสวิตช์ป้อนสัญญาณรบกวน $Sw.e_1$-$e_{12}$ ทางภาคเข้ารหัสสลับไปยังตำแหน่ง 1 หรือ 0 ดังตารางบันทึกผลการทดลองที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. กดสวิตช์ Go ของภาคเข้ารหัส เพื่อส่งข้อมูลจากภาคเข้ารหัสไปยังภาคถอดรหัสแล้ว บันทึกผลของเวกเตอร์ซีนโดรม ; $\underline{s}$ และรูปแบบตำแหน่งบิตผิด ; $\underline{e}$ บันทึกผลการทดลองลงในตารางที่ 2 และ 3

| สวิตช์ผิดพลาด | | | | | | | | เวกเตอร์ซีนโดรม | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | | | | |

**ตารางที่ 2 ผลของเวกเตอร์ซีนโดรมที่เกิดจากการผิดพลาดของคำรหัสที่ภาคเข้ารหัส**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สวิตช์ผิดพลาด | | | | | | | | รูปแบบตำแหน่งบิตที่ผิดทางภาครับ | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | | | | |

**ตารางที่ 3 รูปแบบตำแหน่งของบิตผิด**

**ลำดับขั้นการทดลอง ภาคเข้ารหัสและภาคถอดรหัสขนาด (12,4) บิต [ ชุดพัฒนา ]**

1. โยกสวิตช์ $Sw_3$ ไปที่ตำแหน่ง 0 เพื่อเลือกใช้ชุดทดลองวงจรเข้ารหัส และกดสวิตช์ $Sw_1$ ไปเลือกชุดวงจร (12,4) บิต

2. โยกสวิตช์ $e_1$-$e_{12}$ ไปที่ตำแหน่ง 0 ให้หมด

3. เลือกกดปุ่ม Up หรือ Down เลือกข้อมูลที่จะนำไปเข้ารหัสชุดวงจร (12,4) บิต แล้วบันทึกผลการทดลองของคำรหัสลงในตารางที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ข่าวสารข้อมูล ( Information Data ) | คำรหัส ( Code Word ) |
|---|---|
| $d_1$ $d_2$ $d_3$ $d_4$ | $c_1$ $c_2$ $c_3$ $c_4$ $c_5$ $c_6$ $c_7$ $c_8$ $c_9$ $c_{10}$ $c_{11}$ $c_{12}$ |
| 0 0 0 0 | |
| 0 0 0 1 | |
| 0 0 1 0 | |
| 0 0 1 1 | |
| 0 1 0 0 | |
| 0 1 0 1 | |
| 0 1 1 0 | |
| 0 1 1 1 | |
| 1 0 0 0 | |
| 1 0 0 1 | |
| 1 0 1 0 | |
| 1 0 1 1 | |
| 1 1 0 0 | |
| 1 1 0 1 | |
| 1 1 1 0 | |
| 1 1 1 1 | |

**ตารางที่ 4 ผลของคำรหัสที่ได้จากชุดวงจรเข้ารหัสขนาด (12,4) บิต**

4. โยกสวิตช์ $Sw_3$ ของภาคถอดรหัสไปที่ตำแหน่ง 0 เพื่อเลือกใช้ชุดทดลองวงจรถอดรหัส และโยกสวิตช์ $Sw_1$ ของวงจรถอดรหัสไปที่ตำแหน่ง 0 เพื่อเลือกใช้ชุดวงจร (8,4) บิต

5. ทำการเชื่อมโยงผลของคำรหัสจากภาคเข้ารหัสกับภาคถอดรหัสเข้าด้วยกัน โดยใช้ชุดเข้าจังหวะของชุดทดลองไมโครคอนโทรลเลอร์ MCS 87C51 โดยมี PIN3 เป็นตัวเชื่อมโยง

6. เลือกกดสวิตช์ Up หรือ Down เพื่อเลือกข้อมูลเข้ารหัสที่ภาคเข้ารหัส (12,4) บิต เช่น ที่ $\underline{d}$ = 0000 จะได้คำรหัสที่ภาคเข้ารหัส $\underline{c}$ = 000011101011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. โยกสวิตช์ Sw.$e_1$-$e_{12}$ ทางภาคเข้ารหัสสลับไปยังตำแหน่ง 1 หรือ 0 ดังตารางบันทึกผลการทดลองที่ 5

8. กดสวิตช์ Go ของภาคเข้ารหัสเพื่อส่งข้อมูลจากภาคเข้ารหัสมายังภาคถอดรหัส แล้วบันทึกผลของเวกเตอร์ซีนโดรมและรูปแบบตำแหน่งบิตผิด ลงในตารางบันทึกผลการทดลองที่ 5 และ 6

| สวิตช์ป้อนค่าความผิดพลาด | | | | | | | | | | | | เวกเตอร์ซีนโดรม | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $e_9$ | $e_{10}$ | $e_{11}$ | $e_{12}$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |

**ตารางที่ 5 ผลของเวกเตอร์ซีนโดรมที่เกิดจากการผิดพลาดของคำรหัสที่ภาคเข้ารหัส**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สวิตช์ป้อนค่าความผิดพลาด | | | | | | | | | | | | เวกเตอร์ซีนโดรม | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $e_9$ | $e_{10}$ | $e_{11}$ | $e_{12}$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | | | | | | |

**ตารางที่ 5 (ต่อ)**

| สวิตช์ป้อนค่าความผิดพลาด | | | | | | | | | | | | เวกเตอร์ซีนโดรม | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $e_9$ | $e_{10}$ | $e_{11}$ | $e_{12}$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |

**ตารางที่ 6 รูปแบบตำแหน่งบิตที่ผิด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สวิตช์ผิดพลาด | | | | | | | | | | | | เวกเตอร์ซีนโดรม | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $e_9$ | $e_{10}$ | $e_{11}$ | $e_{12}$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | | | | | | |

**ตารางที่ 6 (ต่อ)**

**คำถามท้ายการทดลอง**

1. จงอธิบายหลักการ การเข้ารหัสช่องสัญญาณแบบรหัสเชิงเส้นขนาด (8,4) บิต มาเข้าใจ
2. จงอธิบายหลักการ การเข้ารหัสช่องสัญญาณแบบรหัสเชิงเส้นขนาด (8,4) บิต มาเข้าใจ
3. จงให้ความหมายของคำต่อไปนี้

   - ข่าวสารข้อมูล ( Data Information )
   - คำรหัสส่ง ( Code Word )
   - คำรหัสรับ ( Received Word )
   - เวกเตอร์ซีนโดรม ( Vector Syndrome )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการทดลอง

## แหล่งค้นคว้า

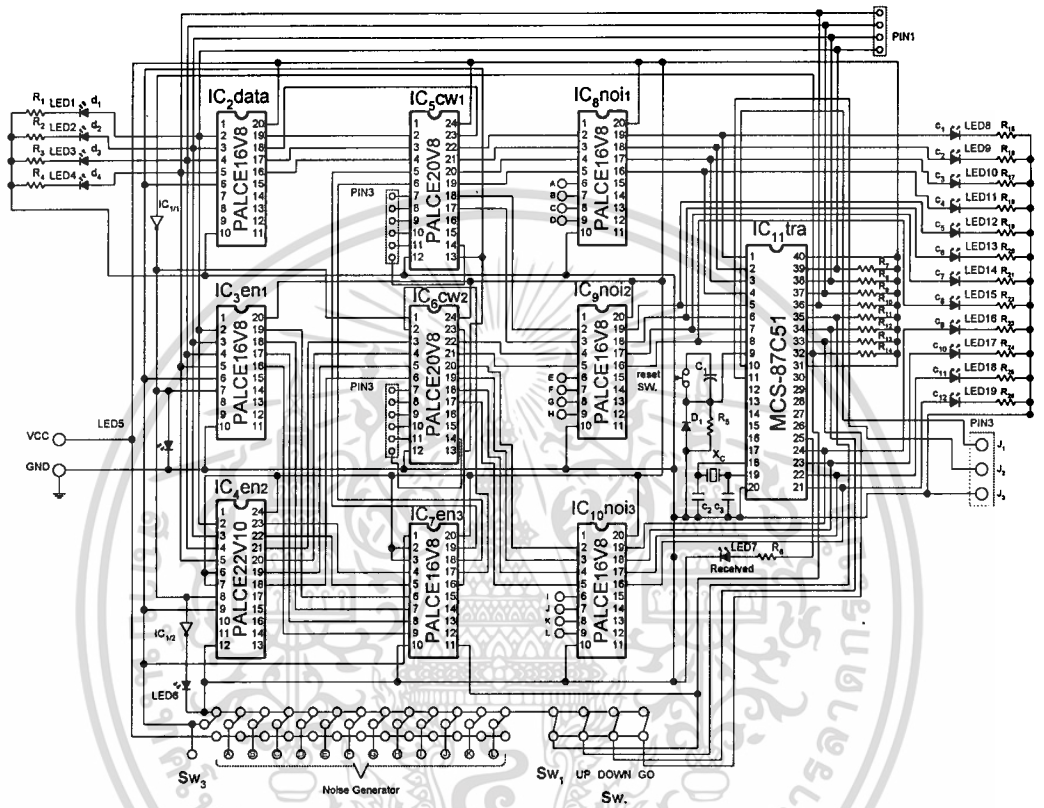

1. หลักการระบบสื่อสาร ประสิทธิ์ ประพิณมงคลการ.
2. การแก้ไขรหัสที่ผิด ฟูศักดิ์ ชีวสุวิทย์.
3. Error Correcting Coding Theory Rhee Man Young.


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

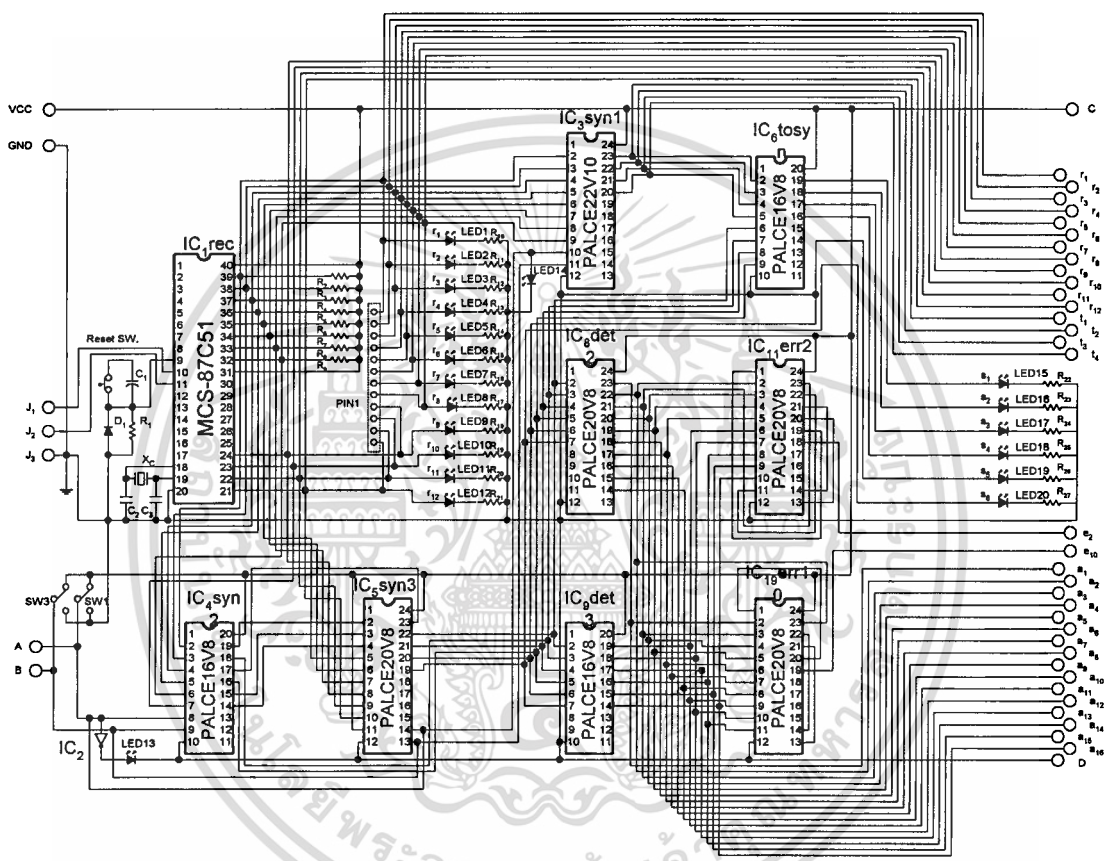
# ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



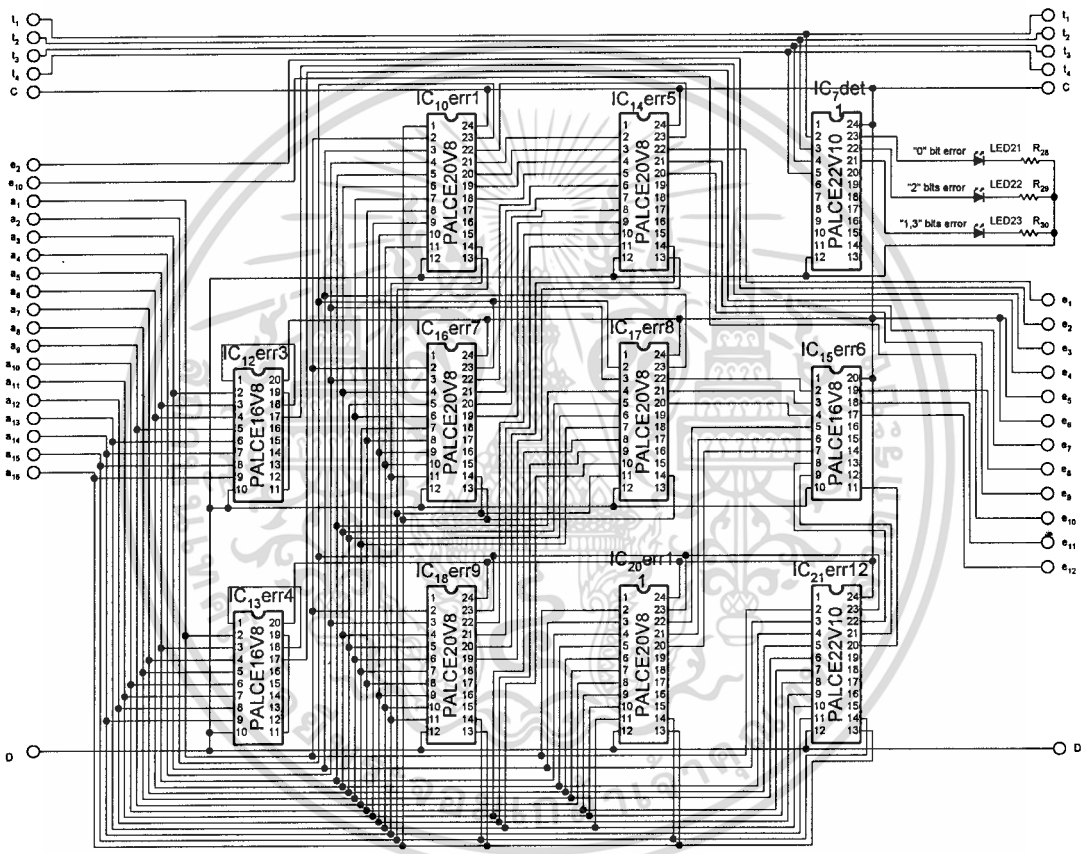

**รูป วงจรชุดส่งข้อมูลการเข้ารหัสช่องสัญญาณ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



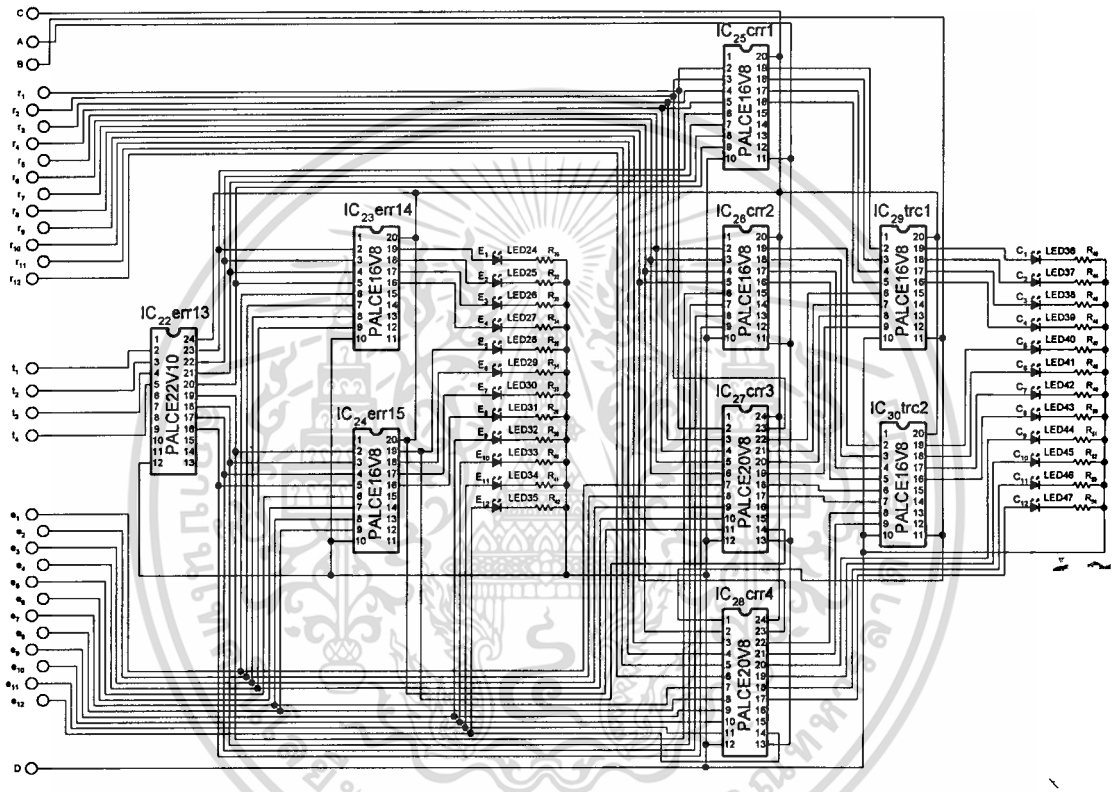

**รูป วงจรชุดรับข้อมูลการเข้ารหัสช่องสัญญาณ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป วงจรชุดรับข้อมูลการเข้ารหัสช่องสัญญาณ (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูป วงจรชุดรับข้อมูลการเข้ารหัสช่องสัญญาณ (ต่อ)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รายการอุปกรณ์ชุดส่งข้อมูล

**ตัวต้านทานขนาด 1/4 W ± 5%**

| | | |
|---|---|---|
| R1-R4 , R6-R17 , R26 | : 240 Ω | 17 ตัว |
| R5 | : 4.7 KΩ | 1 ตัว |
| R16-R25 | : 10 KΩ | 8 ตัว |
| R26 | : 50 Ω | 1 ตัว |

**ตัวเก็บประจุ**

| | | |
|---|---|---|
| C1 | : 10 μF 50 V อิเล็กทรอไลต์ | 1 ตัว |
| C2,C3 | : 30 pF เซรามิค | 2 ตัว |

**อุปกรณ์สารกึ่งตัวนำ**

| | | |
|---|---|---|
| D1 | : 1N4148 | 1 ตัว |
| IC1, IC2, IC6-IC9 | : PAL CE 16V8 | 6 ตัว |
| IC3 | : PAL CE 22V10 | 1 ตัว |
| IC4-IC5 | : PAL CE 20V8 | 2 ตัว |
| IC10 | : MCS 87C51 | 1 ตัว |
| IC11 | : 74LS04 | 1 ตัว |
| LED1-LED5 | : สีแดง | 4 ตัว |
| LED6-LED17 | : สีเขียว | 12 ตัว |
| LED18,LED$e_1$-$e_{12}$ | : สีส้ม | 1 ตัว |
| LED19 | : สีเหลือง | 1 ตัว |

**อื่นๆ**

| | | |
|---|---|---|
| XTAL 1 | : คริสตอล | 1 ตัว |
| $Sw_1$ กดติดกดดับ | | 1 ตัว |
| $Sw_2$ DIP Switch | | 1 ตัว |
| $Sw_3$ สวิตช์ 2 ทาง | | 1 ตัว |
| Swe$_1$-e$_{12}$ สวิตช์ 2 ทาง | | 12 ตัว |
| GO Sw กดติดกดดับ | | 1 ตัว |
| UP Sw กดติดกดดับ | | 1 ตัว |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| DOWN Sw กดติดกดดับ | 1 ตัว |
| POWER SWITCH สวิตช์เปิด-ปิด | 1 ตัว |
| PIN 1 PIN HEADER 4 ขา | 1 ตัว |
| PIN 2 PIN HEADER 12 ขา | 1 ตัว |
| PIN 3 PIN HEADER 3 ขา | 1 ตัว |
| ซ็อกเก็ตไอซี | 11 ตัว |
| แผงประกอบวงจร ( Proto Board ) | 2 ตัว |

## รายการอุปกรณ์ชุดรับข้อมูลการเข้ารหัส

**ตัวต้านทาน ขนาด 1/4 W $\pm$ 5%**

| | |
|---|---|
| R1 : 4.7 KΩ | 1 ตัว |
| R2-R13, R22-R27, R28-R30, R31-R42, R43-R54 : 240 Ω | 45 ตัว |
| R14-R21 : 10 KΩ | 8 ตัว |

**ตัวเก็บประจุ**

| | |
|---|---|
| C1 : 10 μF 50 V อิเลกทรอไลต์ | 1 ตัว |
| C2,C3 : 30 pF เซรามิค | 2 ตัว |

**อุปกรณ์สารกึ่งตัวนำ**

| | |
|---|---|
| D1 : 1N4148 | 1 ตัว |
| IC1 : MCS 87C51 | 1 ตัว |
| IC2 : 74LS04 | 1 ตัว |
| IC3, IC7, IC8 : PAL CE 22V10 | 3 ตัว |
| IC5, IC10, IC11, IC14, IC16-IC22, IC27, IC28 : PAL CE 20V8 | 13 ตัว |
| IC4, IC6, IC9, IC12, IC13, IC15, IC23-IC26, IC29, IC30 : PAL CE 16V8 | 12 ตัว |
| LED1-LED12, LED22-LED33 : สีแดง | 24 ตัว |
| LED13-LED18 : สีเหลือง | 6 ตัว |
| LED19-LED21 : สีส้ม | 3 ตัว |
| LED34-LED45 : สีเขียว | 12 ตัว |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**อื่นๆ**

| | |
|---|---|
| XTAL 1 : คริสตอล | 1 ตัว |
| $SW_1$ กดติดปล่อยดับ | 1 ตัว |
| $SW_2$ DIP Switch | 1 ตัว |
| POWER SWITCH สวิตช์เปิด-ปิด | 1 ตัว |
| PIN 1 PIN HEADER 3 ขา | 1 ตัว |
| PIN 3 PIN HEADER 12 ขา | 1 ตัว |
| ซ็อกเก็ตไอซี | 30 ตัว |
| แผงประกอบวงจร | 2 ตัว |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

PRELIMINARY

# 87C51/87C51-1/87C51-2
# CHMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 4K BYTES OF EPROM PROGRAM MEMORY

**87C51—3.5 to 12 MHz, $V_{CC}$ = 5V ± 10%**
**87C51-1—3.5 to 16 MHz, $V_{CC}$ = 5V ± 10%**
**87C51-2—0.5 to 12 MHz, $V_{CC}$ = 5V ± 10%**

- High Performance CHMOS EPROM
- Quick-Pulse Programming™ Algorithm
- 2-Level Program Memory Lock
- Boolean Processor
- 128-Byte Data RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 5 Interrupt Sources
- Programmable Serial Channel
- TTL- and CMOS-Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- IDLE and POWER DOWN Modes
- ONCE™ Mode Facilitates System Testing
- LCC, PLCC, and DIP Packaging Available

The 87C51 is the EPROM version of the 80C51BH. It is fabricated on Intel's CHMOS II-E process. It contains 4K bytes of on-chip Program memory that can be electrically programmed, and can be erased by exposure to ultraviolet light.

The 87C51 EPROM array uses a modified Quick-Pulse programming algorithm, by which the entire 4K-byte array can be programmed in about 12 seconds.

The extremely low operating power, along with the two reduced power modes, Idle and Power Down, make this part very suitable for low power applications. The Idle mode freezes the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down mode saves the RAM contents but freezes the oscillator, causing all other chip functions to be inoperative.

October 1988
Order Number: 270147-005



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ....0°C to +70°C
Storage Temperature .....−65°C to +150°C
Voltage on $\overline{EA}$/$V_{PP}$ Pin to $V_{SS}$ .....0V to +13.0V
Voltage on Any Other Pin to $V_{SS}$ ..−0.5V to +6.5V
Maximum $I_{OL}$ per I/O Pin ..... 15 mA
Power Dissipation.....1.5W
(Based on package heat transfer limitations, not device power consumption).

*Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

*NOTICE: Specifications contained within the following tables are subject to change.*

## D.C. CHARACTERISTICS: ($T_A$ = 0°C to +70°C; $V_{CC}$ = 5V ±10%; $V_{SS}$ = 0V)

| Symbol | Parameter | Min | Typ(1) | Max | Unit | Test Conditions |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage (Except $\overline{EA}$) | −0.5 | .. | .2$V_{CC}$−.1 | V | |
| $V_{IL1}$ | Input Low Voltage to $\overline{EA}$ | 0 | | .2$V_{CC}$−.3 | V | |
| $V_{IH}$ | Input High Voltage (Except XTAL1, RST) | .2$V_{CC}$+.9 | | $V_{CC}$+.5 | V | |
| $V_{IH1}$ | Input High Voltage (XTAL1, RST) | 0.7$V_{CC}$ | | $V_{CC}$+.5 | V | |
| $V_{OL}$ | Output Low Voltage (Ports 1, 2, 3) (7) | | | 0.45 | V | $I_{OL}$ = 1.6 mA (2) |
| $V_{OL1}$ | Output Low Voltage (Port 0, ALE, $\overline{PSEN}$) (7) | | | 0.45 | V | $I_{OL}$ = 3.2 mA (2) |
| $V_{OH}$ | Output High Voltage (Ports 1, 2, 3, ALE, $\overline{PSEN}$) | 2.4 | | | V | $I_{OH}$ = −60 µA |
| | | .75$V_{CC}$ | | | V | $I_{OH}$ = −25 µA |
| | | .9$V_{CC}$ | | | V | $I_{OH}$ = −10 µA |
| $V_{OH1}$ | Output High Voltage (Port 0 in External Bus Mode) | 2.4 | | | V | $I_{OH}$ = −800 µA |
| | | .75 $V_{CC}$ | | | V | $I_{OH}$ = −300 µA |
| | | .9$V_{CC}$ | | | V | $I_{OH}$ = −80 µA (3) |
| $I_{IL}$ | Logical 0 Input Current (Ports 1, 2, 3) | | | −50 | µA | $V_{IN}$ = 0.45 V |
| $I_{TL}$ | Logical 1-to-0 transition current (Ports 1, 2, 3) | | | −650 | µA | (4) |
| $I_{LI}$ | Input Leakage Current (Port 0) | | | ±10 | µA | $V_{IN}$ = $V_{IL}$ or $V_{IH}$ |
| $I_{CC}$ | Power Supply Current:<br>Active Mode @ 12 MHz (5)<br>Idle Mode @ 12 MHz (5)<br>Power Down Mode | | <br>11.5<br>1.3<br>3 | <br>25<br>4<br>50 | <br>mA<br>mA<br>µA | <br><br>(6) |
| RRST | Internal Reset Pulldown Resistor | 50 | | 300 | kΩ | |
| $C_{IO}$ | Pin Capacitance | | | 10 | pF | |

**NOTES:**
1. "Typicals" are based on a limited number of samples taken from early manufacturing lots and are not guaranteed. The values listed are at room temp, 5V.
2. Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the $V_{OL}$s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading > 100pF), the noise pulse on the ALE pin may exceed 0.8V. In such cases it may be desirable to qualify ALE with a Schmitt Trigger, or use an address latch with a Schmitt Trigger STROBE input.
3. Capacitive loading on Ports 0 and 2 may cause the $V_{OH}$ on ALE and $\overline{PSEN}$ to momentarily fall below the 0.9$V_{CC}$ specification when the address bits are stabilizing.
4. Pins of Ports 1, 2, and 3 source a transition current when they are being externally driven from 1 to 0. The transition current reaches its maximum value when $V_{IN}$ is approximately 2V.
5. $I_{CC}$MAX at other frequencies is given by:

Active Mode: $I_{CC}$MAX = 0.94 × FREQ + 13.71
Idle Mode: $I_{CC}$MAX = 0.14 × FREQ + 2.31

where FREQ is the external oscillator frequency in MHz. $I_{CC}$MAX is given in mA. See Figure 5.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. See Figures 6 through 9 for $I_{CC}$ test conditions. Minimum $V_{CC}$ for Power Down is 2V.
7. Under steady state (non-transient) conditions, $I_{OL}$ must be externally limited as follows:

| | |
|---|---|
| Maximum $I_{OL}$ per port pin: | 10 mA |
| Maximum $I_{OL}$ per 8-bit port— | |
| Port 0: | 26 mA |
| Ports 1, 2, and 3: | 15 mA |
| Maximum total $I_{OL}$ for all output pins: | 71 mA |

If $I_{OL}$ exceeds the test condition, $V_{OL}$ may exceed the related specification. Pins are not guaranteed to sink greater than the listed test conditions.

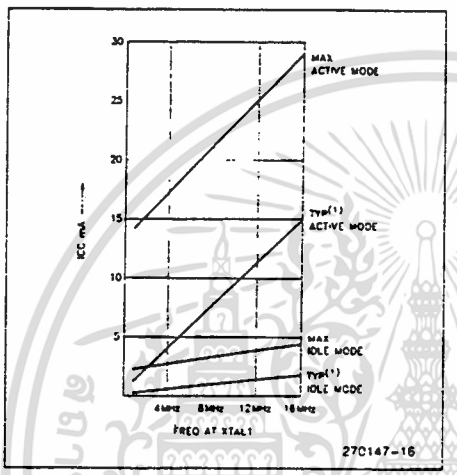



Figure 5. $I_{CC}$ vs. FREQ. Valid only within frequency specifications of the device under test.

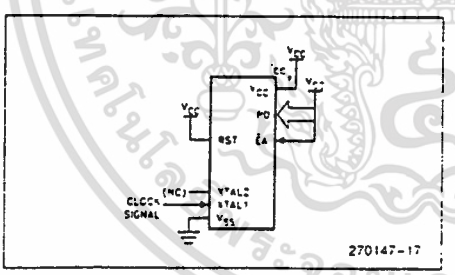



Figure 6. $I_{CC}$ Test Condition, Active Mode. All other pins are disconnected.

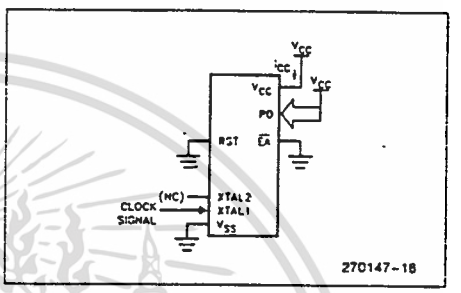



Figure 7. $I_{CC}$ Test Condition, Idle Mode. All other pins are disconnected.

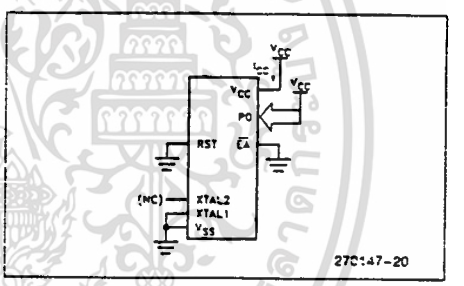



Figure 9. $I_{CC}$ Test Condition, Power Down Mode. All other pins are disconnected. $V_{CC}$ = 2V to 5.5V.

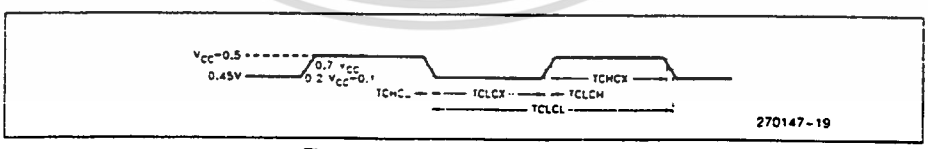



Figure 8. Clock Signal Waveform for $I_{CC}$ tests in Active and Idle Modes. TCLCH = TCHCL = 5 ns.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list of all the characters and what they stand for.

A:Address.
C:Clock.
D:Input data.
H:Logic level HIGH.
I:Instruction (program memory contents).
L:Logic level LOW, or ALE.
P:$\overline{PSEN}$.
Q:Output data.
R:$\overline{RD}$ signal.
T:Time.
V:Valid.
W:$\overline{WR}$ signal.
X:No longer a valid logic level.
Z:Float.

For example,

TAVLL = Time from Address Valid to ALE Low.
TLLPL = Time from ALE Low to PSEN Low.

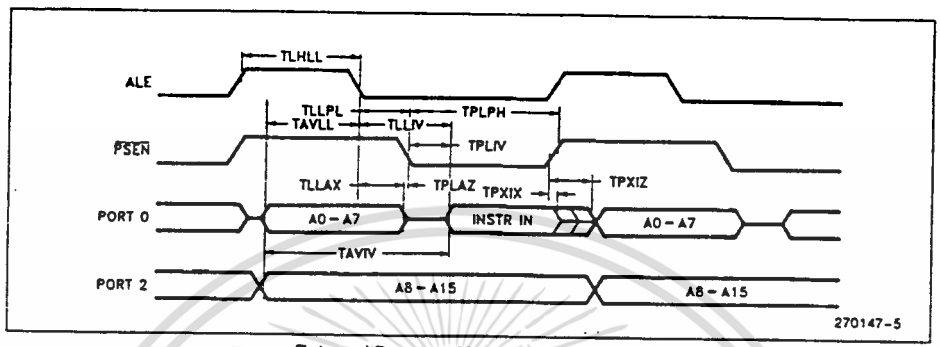
**A.C. CHARACTERISTICS:** ($T_A$ = 0°C to +70°C; $V_{CC}$ = 5V ± 10%; $V_{SS}$ = 0V; Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100 pF; Load Capacitance for All Other Outputs = 80 pF)

## EXTERNAL PROGRAM AND DATA MEMORY CHARACTERISTICS

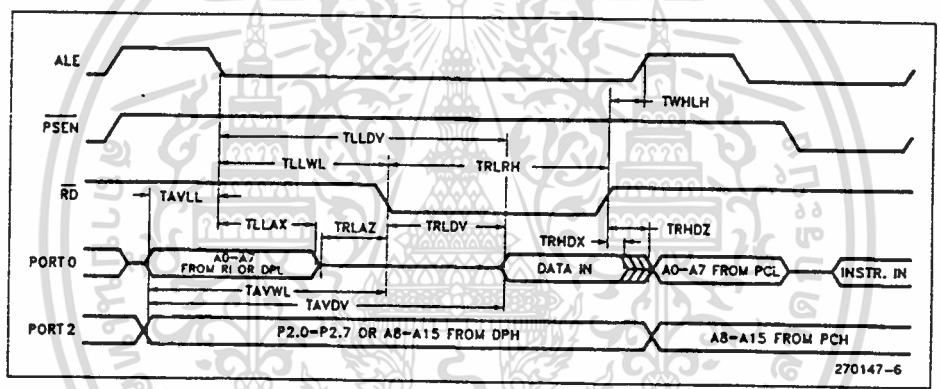
| Symbol | Parameter | 12 MHz Oscillator | | Variable Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| 1/TCLCL | Oscillator Frequency<br>87C51<br>87C51-1<br>87C51-2 | | | <br>3.5<br>3.5<br>0.5 | <br>12<br>16<br>12 | MHz |
| TLHLL | ALE Pulse Width | 127 | | 2TCLCL − 40 | | ns |
| TAVLL | Address Valid to ALE Low | 28 | | TCLCL − 55 | | ns |
| TLLAX | Address Hold After ALE Low | 48 | | TCLCL − 35 | | ns |
| TLLIV | ALE Low to Valid Instr In | | 234 | | 4TCLCL − 100 | ns |
| TLLPL | ALE Low to $\overline{PSEN}$ Low | 43 | | TCLCL − 40 | | ns |
| TPLPH | $\overline{PSEN}$ Pulse Width | 205 | | 3TCLCL − 45 | | ns |
| TPLIV | $\overline{PSEN}$ Low to Valid Instr In | | 145 | | 3TCLCL − 105 | ns |
| TPXIX | Input Instr Hold After $\overline{PSEN}$ | 0 | | 0 | | ns |
| TPXIZ | Input Instr Float After $\overline{PSEN}$ | | 59 | | TCLCL − 25 | ns |
| TAVIV | Address to Valid Instr In | | 312 | | 5TCLCL − 105 | ns |
| TPLAZ | $\overline{PSEN}$ Low to Address Float | | 10 | | 10 | ns |
| TRLRH | $\overline{RD}$ Pulse Width | 400 | | 6TCLCL − 100 | | ns |
| TWLWH | $\overline{WR}$ Pulse Width | 400 | | 6TCLCL − 100 | | ns |
| TRLDV | $\overline{RD}$ Low to Valid Data In | | 252 | | 5TCLCL − 165 | ns |
| TRHDX | Data Hold After $\overline{RD}$ | 0 | | 0 | | ns |
| TRHDZ | Data Float After $\overline{RD}$ | | 97 | | 2TCLCL − 70 | ns |
| TLLDV | ALE Low to Valid Data In | | 517 | | 8TCLCL − 150 | ns |
| TAVDV | Address to Valid Data In | | 585 | | 9TCLCL − 165 | ns |
| TLLWL | ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | 200 | 300 | 3TCLCL − 50 | 3TCLCL + 50 | ns |
| TAVWL | Address to $\overline{RD}$ or $\overline{WR}$ Low | 203 | | 4TCLCL − 130 | | ns |
| TQVWX | Data Valid to $\overline{WR}$ Transition | 23 | | TCLCL − 60 | | ns |
| TWHQX | Data Hold After $\overline{WR}$ | 33 | | TCLCL − 50 | | ns |
| TRLAZ | $\overline{RD}$ Low to Address Float | | 0 | | 0 | ns |
| TWHLH | $\overline{RD}$ or $\overline{WR}$ High to ALE High | 43 | 123 | TCLCL − 40 | TCLCL + 40 | ns |



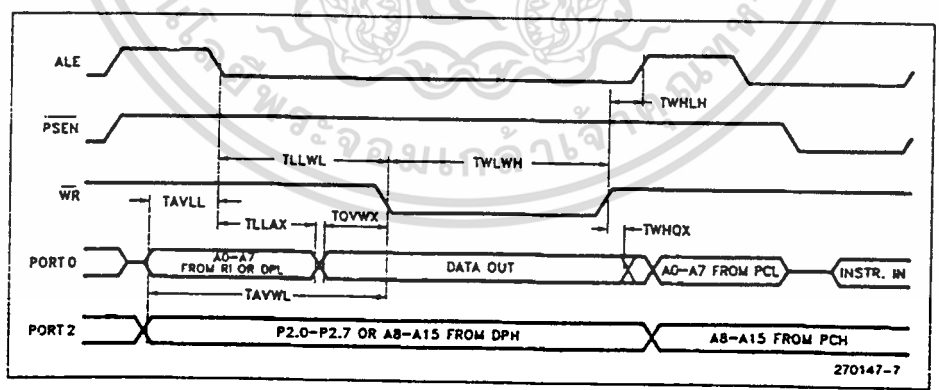
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




External Program Memory Read Cycle




External Data Memory Read Cycle




External Data Memory Write Cycle

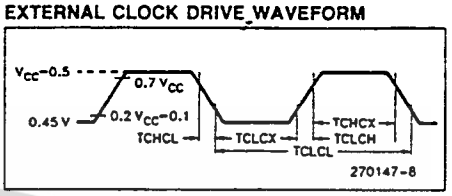


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**EXTERNAL CLOCK DRIVE**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| 1/TCLCL | Oscillator Frequency<br>87C51<br>87C51-1<br>87C51-2 | <br>3.5<br>3.5<br>0.5 | <br>12<br>16<br>12 | MHz |
| TCHCX | High Time | 20 | | ns |
| TCLCX | Low Time | 20 | | ns |
| TCLCH | Rise Time | | 20 | ns |
| TCHCL | Fall Time | | 20 | ns |

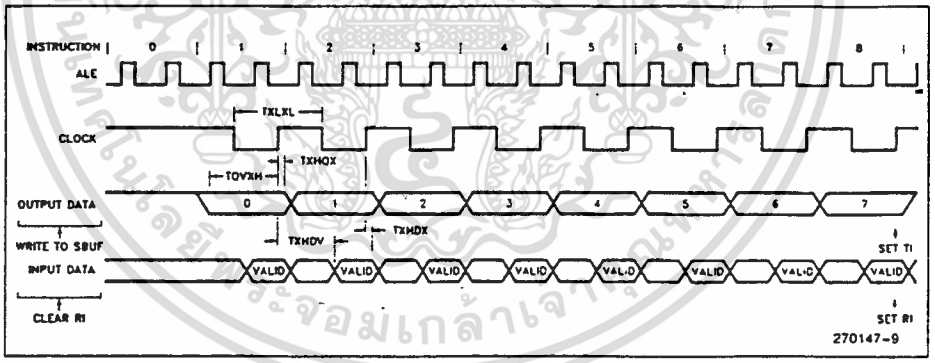
**EXTERNAL CLOCK DRIVE WAVEFORM**




**SERIAL PORT TIMING—SHIFT REGISTER MODE**

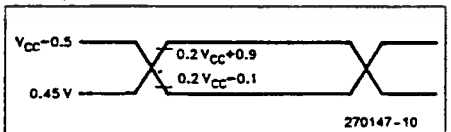
| Symbol | Parameter | 12 MHz Oscillator | | Variable Oscillator | | Units |
|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | |
| TXLXL | Serial Port Clock Cycle Time | 1.0 | | 12TCLCL | | μs |
| TQVXH | Output Data Setup to Clock Rising Edge | 700 | | 10TCLCL − 133 | | ns |
| TXHQX | Output Data Hold After Clock Rising Edge | 50 | | 2TCLCL − 117 | | ns |
| TXHDX | Input Data Hold After Clock Rising Edge | 0 | | 0 | | ns |
| TXHDV | Clock Rising Edge to Input Data Valid | | 700 | | 10TCLCL − 133 | ns |

**SHIFT REGISTER MODE TIMING WAVEFORMS**



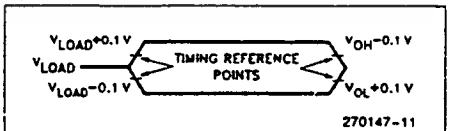

**A.C. TESTING:**

**INPUT, OUTPUT WAVEFORMS**




AC inputs during testing are driven at $V_{CC}$ − 0.5 for a Logic "1" and 0.45V for a Logic "0." Timing measurements are made at $V_{IH}$ min for a Logic "1" and $V_{IL}$ max for a Logic "0"

**FLOAT WAVEFORM**




For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, and begins to float when a 100 mV change from the loaded $V_{OH}/V_{OL}$ level occurs $I_{OL}/I_{OH} \geq \pm 20$ mA.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## EPROM CHARACTERISTICS

The 87C51 is programmed by a modified Quick-Pulse Programming™ algorithm. It differs from older methods in the value used for $V_{PP}$ (Programming Supply Voltage) and in the width and number of the ALE/$\overline{PROG}$ pulses.

The 87C51 contains two signature bytes that can be read and used by an EPROM programming system to identify the device. The signature bytes identify the device as an 87C51 manufactured by Intel.

Table 2 shows the logic levels for reading the signature byte, and for programming the Program Memory, the Encryption Table, and the Lock Bits. The circuit configuration and waveforms for Quick-Pulse Programming™ are shown in Figures 10 and 11. Figure 12 shows the circuit configuration for normal Program Memory verification.

**Table 2. EPROM Programming Modes**

| MODE | RST | $\overline{PSEN}$ | ALE/$\overline{PROG}$ | $\overline{EA}$/$V_{PP}$ | P2.7 | P2.6 | P3.7 | P3.6 |
|---|---|---|---|---|---|---|---|---|
| Read Signature | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Program Code Data | 1 | 0 | 0* | $V_{PP}$ | 1 | 0 | 1 | 1 |
| Verify Code Data | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| Pgm Encryption Table | 1 | 0 | 0* | $V_{PP}$ | 1 | 0 | 1 | 0 |
| Pgm Lock Bit 1 | 1 | 0 | 0* | $V_{PP}$ | 1 | 1 | 1 | 1 |
| Pgm Lock Bit 2 | 1 | 0 | 0* | $V_{PP}$ | 1 | 1 | 0 | 0 |

NOTES:
"1" = Valid high for that pin
"0" = Valid low for that pin
$V_{PP}$ = 12.75V ± 0.25V
$V_{CC}$ = 5V ± 10% during programming and verification
*ALE/$\overline{PROG}$ receives 25 programming pulses while $V_{PP}$ is held at 12.75V. Each programming pulse is low for 100 µS(±10 µS) and high for a minimum of 10 µS.

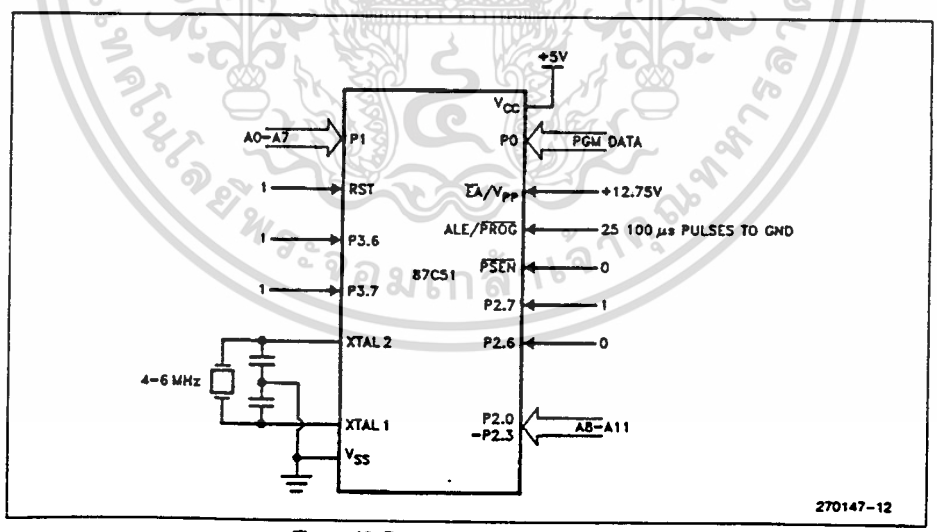



**Figure 10. Programming Configuration**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

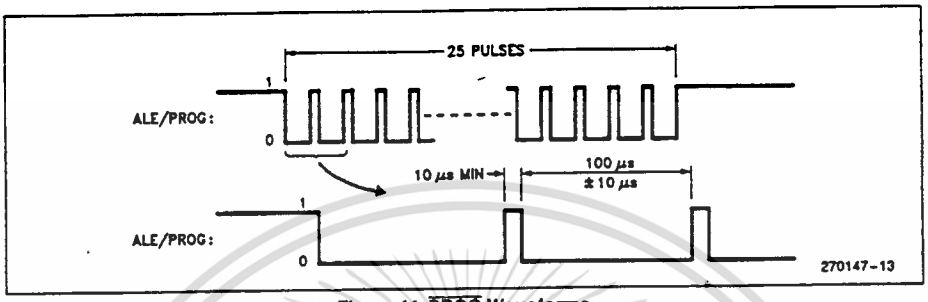



Figure 11. $\overline{PROG}$ Waveforms

## Quick-Pulse Programming™

The setup for Microcontroller Quick-Pulse Programming™ is shown in Figure 10. Note that the 87C51 is running with a 4 to 6 MHz oscillator. The reason the oscillator needs to be running is that the device is executing internal address and program data transfers.

The address of the EPROM location to be programmed is applied to Ports 1 and 2, as shown in Figure 10. The code byte to be programmed into that location is applied to Port 0. RST, $\overline{PSEN}$, and pins of Ports 2 and 3 specified in Table 2 are held at the "Program Code Data" levels indicated in Table 2. Then ALE/$\overline{PROG}$ is pulsed low 25 times as shown in Figure 11.

To program the Encryption Table, repeat the 25-pulse programming sequence for addresses 0 through 1FH, using the "Pgm Encryption Table" levels. Don't forget that after the Encryption Table is programmed, verify cycles will produce only encrypted data.

To program the Lock Bits, repeat the 25-pulse programming sequence using the "Pgm Lock Bit" levels. After one Lock Bit is programmed, further programming of the Code Memory and Encryption Table is disabled. However, the other Lock Bit can still be programmed.

Note that the $\overline{EA}$/$V_{PP}$ pin must not be allowed to go above the maximum specified $V_{PP}$ level for any amount of time. Even a narrow glitch above that voltage level can cause permanent damage to the device. The $V_{PP}$ source should be well regulated and free of glitches and overshoot.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

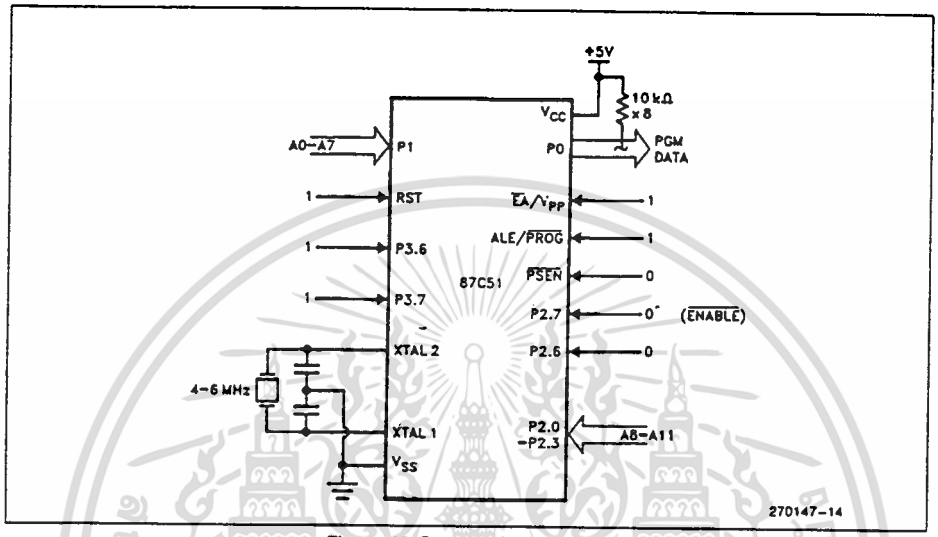



Figure 12. Program Verification

## Program Verification

If Lock Bit 2 has not been programmed, the on-chip Program Memory can be read out for program verification. The address of the Program Memory location to be read is applied to Ports 1 and 2 as shown in Figure 12. The other pins are held at the "Verify Code Data" levels indicated in Table 2. The contents of the addressed location will be emitted on Port 0. External pullups are required on Port 0 for this operation. Detailed timing specifications are shown in later sections of this data sheet.

If the Encryption Table has been programmed, the data presented at Port 0 will be the Exclusive NOR of the program byte with one of the encryption bytes. The user will have to know the Encryption Table contents in order to correctly decode the verification data. The Encryption Table itself can not be read out.

## Reading the Signature Bytes

The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 89H indicates manufactured by Intel

(031H) = 57H indicates 87C51

## Program/Verify Algorithms

Any algorithm in agreement with the conditions listed in Table 2, and which satisfies the timing specifications, is suitable.

## Erasure Characteristics

Erasure of the EPROM begins to occur when the chip is exposed to light with wavelengths shorter than approximately 4,000 Angstroms. Since sunlight and fluorescent lighting have wavelengths in this range, exposure to these light sources over an extended time (about 1 week in sunlight, or 3 years in room level fluorescent lighting) could cause inadvertent erasure. If an application subjects the device to this type of exposure, it is suggested that an opaque label be placed over the window.

The recommended erasure procedure is exposure to ultraviolet light (at 2537 Angstroms) to an integrated dose of at least 15 W-sec/cm$^2$. Exposing the EPROM to an ultraviolet lamp of 12,000 $\mu$W/cm$^2$ rating for 30 minutes, at a distance of about 1 inch, should be sufficient.

Erasure leaves the array in an all 1s state.



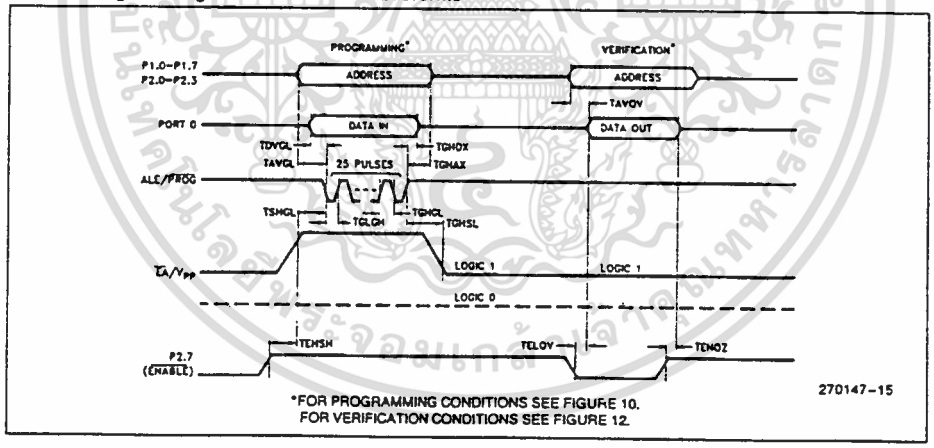
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## EPROM PROGRAMMING AND VERIFICATION CHARACTERISTICS:

($T_A$ = 21°C to 27°C, $V_{CC}$ = 5V ±10%, $V_{SS}$ = 0V)

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{PP}$ | Programming Supply Voltage | 12.5 | 13.0 | V |
| $I_{PP}$ | Programming Supply Current | | 50 | mA |
| 1/TCLCL | Oscillator Frequency | 4 | 6 | MHz |
| TAVGL | Address Setup to $\overline{PROG}$ Low | 48TCLCL | | |
| TGHAX | Address Hold After $\overline{PROG}$ | 48TCLCL | | |
| TDVGL | Data Setup to $\overline{PROG}$ Low | 48TCLCL | | |
| TGHDX | Data Hold After $\overline{PROG}$ | 48TCLCL | | |
| TEHSH | P2.7 ($\overline{ENABLE}$) High to $V_{PP}$ | 48TCLCL | | |
| TSHGL | $V_{PP}$ Setup to $\overline{PROG}$ Low | 10 | | µs |
| TGHSL | $V_{PP}$ Hold After $\overline{PROG}$ | 10 | | µs |
| TGLGH | $\overline{PROG}$ Width | 90 | 110 | µs |
| TAVQV | Address to Data Valid | | 48TCLCL | |
| TELQV | $\overline{ENABLE}$ Low to Data Valid | | 48TCLCL | |
| TEHQZ | Data Float After $\overline{ENABLE}$ | 0 | 48TCLCL | |
| TGHGL | $\overline{PROG}$ High to $\overline{PROG}$ Low | 10 | | µs |

**EPROM Programming and Verification Waveforms**




*FOR PROGRAMMING CONDITIONS SEE FIGURE 10.
FOR VERIFICATION CONDITIONS SEE FIGURE 12.

## DATA SHEET REVISION SUMMARY

The following are the key differences between this and the -004 version of the 87C51BH data sheet:

1. Package table was added.
2. Note 7 on maximum current specifications added to DC Characteristics.
3. Data Sheet Revision Summary was added.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM'L: H-7/10/15/25, Q-15/25 MIL: H-10/15/20/25

# PALCE16V8 Family

## EE CMOS 20-Pin Universal Programmable Array Logic

Advanced Micro Devices

### DISTINCTIVE CHARACTERISTICS

- Pin, function and fuse-map compatible with all 20-pin GAL® devices
- Electrically erasable CMOS technology provides reconfigurable logic and full testability
- High-speed CMOS technology
  - 7.5 ns propagation delay for "-7" version
  - 10 ns propagation delay for "-10" version
  - 15 ns propagation delay for "-15" version
- Direct plug-in replacement for the PAL16R8 series and most of the PAL10H8 series
- Outputs programmable as registered or combinatorial in any combination
- Programmable output polarity
- Programmable enable/disable control
- Preloadable output registers for testability
- Automatic register reset on power up
- Cost-effective 20-pin plastic DIP, PLCC, and SOIC packages
- Extensive third-party software and programmer support through FusionPLD partners
- Fully tested for 100% programming and functional yields and high reliability

### GENERAL DESCRIPTION

The PALCE16V8 is an advanced PAL device built with low-power, high-speed, electrically-erasable CMOS technology. It is functionally compatible with all 20-pin GAL devices. The macrocells provide a universal device architecture. The PALCE16V8 will directly replace the PAL16R8 and PAL10H8 series devices, with the exception of the PAL16C1.

The PALCE16V8 utilizes the familiar sum-of-products (AND/OR) architecture that allows users to implement complex logic functions easily and efficiently. Multiple levels of combinatorial logic can always be reduced to sum-of-products form, taking advantage of the very wide input gates available in PAL devices. The equations are programmed into the device through floating-gate cells in the AND logic array that can be erased electrically.

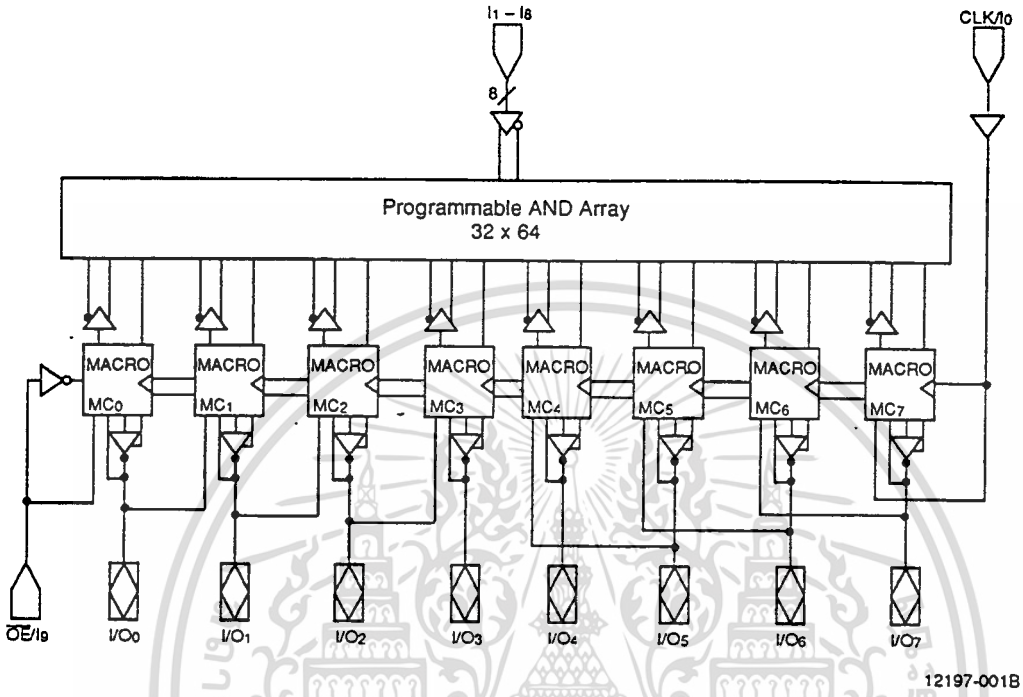
The fixed OR array allows up to eight data product terms per output for logic functions. The sum of these products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial with an active-high or active-low output. The output configuration is determined by two global bits and one local bit controlling four multiplexers in each macrocell.

AMD's FusionPLD program allows PALCE16V8 designs to be implemented using a wide variety of popular industry-standard design tools. By working closely with the FusionPLD partners, AMD certifies that the tools provide accurate, quality support. By ensuring that third-party tools are available, costs are lowered because a designer does not have to buy a complete set of new tools for each device. The FusionPLD program also greatly reduces design time since a designer can use a tool that is already installed and familiar. Please refer to the PLD Software Reference Guide for certified development systems and the Programmer Reference Guide for approved programmers.

Publication# 16493 Rev. A Amendment/0
Issue Date: January 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

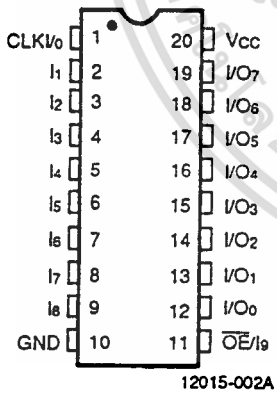
## BLOCK DIAGRAM




12197-001B

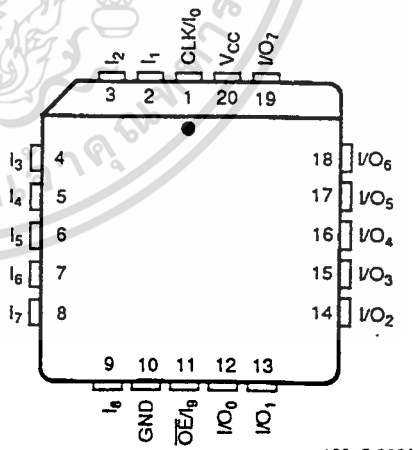
## CONNECTION DIAGRAMS
**Top View**

### DIP/SOIC




12015-002A

### PLCC/LCC




12015-003A

**Note:** Pin 1 is marked for orientation

## PIN DESIGNATIONS

CLK = Clock
GND = Ground
I = Input
I/O = Input/Output
$\overline{OE}$ = Output Enable
Vcc = Supply Voltage



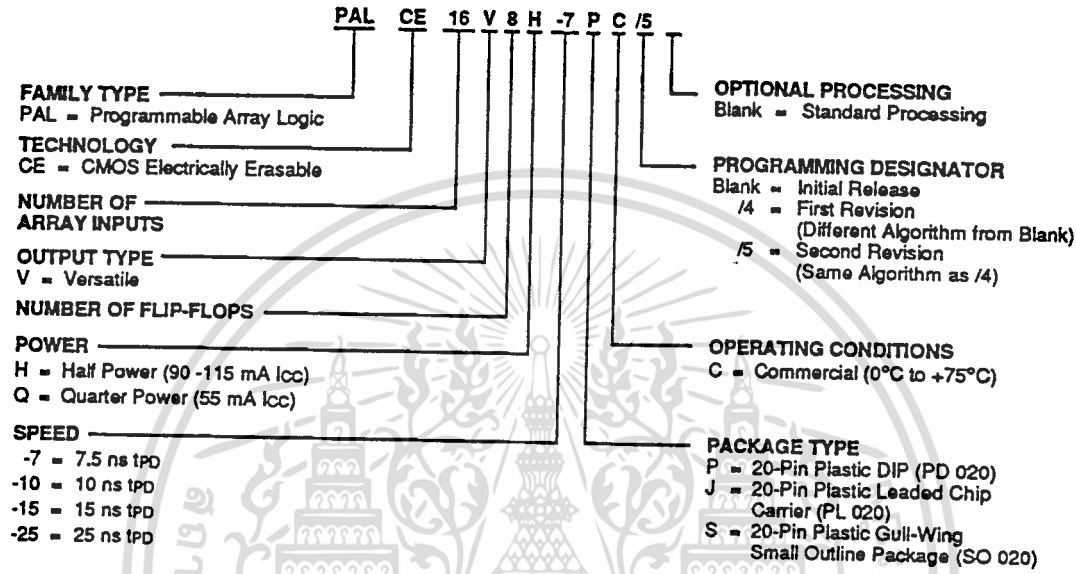
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ORDERING INFORMATION

### Commercial Products

AMD programmable logic products for commercial applications are available with several ordering options. The order number (Valid Combination) is formed by a combination of:




| Valid Combinations | | |
|---|---|---|
| PALCE16V8H-7 | PC, JC | /5 |
| PALCE16V8H-10 | PC, JC, SC | /4, /5 |
| PALCE16V8H-15 | PC, JC, SC | Blank, /4 |
| PALCE16V8H-25 | | |
| PALCE16V8Q-15 | PC, JC | |
| PALCE16V8Q-25 | | |

**Valid Combinations**

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

**Note:** Marked with AMD logo.

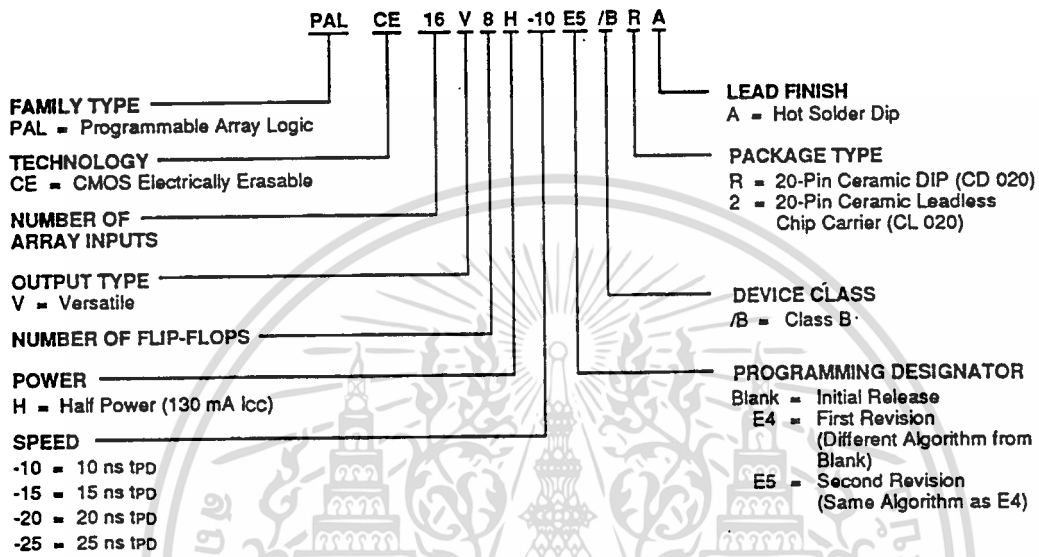


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ORDERING INFORMATION

### APL Products (Military)

AMD programmable logic products for Aerospace and Defense applications are available with several ordering options. APL (Approved Products List) products are fully compliant with MIL-STD-883 requirements. The order number (Valid Combination) is formed by a combination of:




| Valid Combinations | | |
|---|---|---|
| PALCE16V8H-10 | E5 | /BRA /B2A |
| PALCE16V8H-15 | E4, E5 | |
| PALCE16V8H-20 | Blank, E4 | |
| PALCE16V8H-25 | | |

**Valid Combinations**

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations and to obtain additional data on AMD's standard military grade products.

Note: Marked with AMD logo.

**Group A Tests**

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

**Military Burn-in**

Military burn-in is in accordance with the current revision of MIL-STD-883, Test Method 1015, Conditions A through E. Test conditions are selected at AMD's option.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

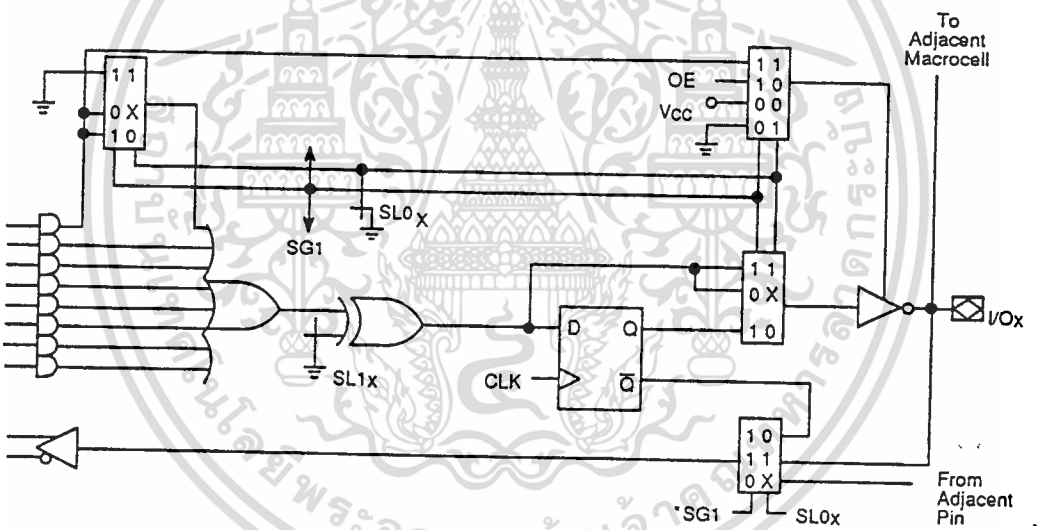
## FUNCTIONAL DESCRIPTION

The PALCE16V8 is a universal PAL device. It has eight independently configurable macrocells ($MC_0$–$MC_7$). Each macrocell can be configured as registered output, combinatorial output, combinatorial I/O or dedicated input. The programming matrix implements a programmable AND logic array, which drives a fixed OR logic array. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Pins 1 and 11 serve either as array inputs or as clock (CLK) and output enable ($\overline{OE}$), respectively, for all flip-flops.

Unused input pins should be tied directly to Vcc or GND. Product terms with all bits unprogrammed (disconnected) assume the logical HIGH state and product terms with both true and complement of any input signal connected assume a logical LOW state.

The programmable functions on the PALCE16V8 are automatically configured from the user's design specification, which can be in a number of formats. The design specification is processed by development software to verify the design and create a programming file. This file, once downloaded to a programmer, configures the device according to the user's desired function.

The user is given two design options with the PALCE16V8. First, it can be programmed as a standard PAL device from the PAL16R8 and PAL10H8 series. The PAL programmer manufacturer will supply device codes for the standard PAL device architectures to be used with the PALCE16V8. The programmer will program the PALCE16V8 in the corresponding architecture. This allows the user to use existing standard PAL device JEDEC files without making any changes to them. Alternatively, the device can be programmed as a PALCE16V8. Here the user must use the PALCE16V8 device code. This option allows full utilization of the macrocell.




*In macrocells $MC_0$ and $MC_7$, SG1 is replaced by $\overline{SG0}$ on the feedback multiplexer.

14408C-001A

PALCE16V8 Macrocell



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Configuration Options

Each macrocell can be configured as one of the following: registered output, combinatorial output, combinatorial I/O, or dedicated input. In the registered output configuration, the output buffer is enabled by the $\overline{OE}$ pin. In the combinatorial configuration, the buffer is either controlled by a product term or always enabled. In the dedicated input configuration, it is always disabled. With the exception of $MC_0$ and $MC_7$, a macrocell configured as a dedicated input derives the input signal from an adjacent I/O. $MC_0$ derives its input from pin 11 ($\overline{OE}$) and $MC_7$ from pin 1 (CLK).

The macrocell configurations are controlled by the configuration control word. It contains 2 global bits (SG0 and SG1) and 16 local bits ($SL0_0$ through $SL0_7$ and $SL1_0$ through $SL1_7$). SG0 determines whether registers will be allowed. SG1 determines whether the PALCE16V8 will emulate a PAL16R8 family or a PAL10H8 family device. Within each macrocell, $SL0_x$, in conjunction with SG1, selects the configuration of the macrocell, and $SL1_x$ sets the output as either active low or active high for the individual macrocell.

The configuration bits work by acting as control inputs for the multiplexers in the macrocell. There are four multiplexers: a product term input, an enable select, an output select, and a feedback select multiplexer. SG1 and $SL0_x$ are the control signals for all four multiplexers. In $MC_0$ and $MC_7$, $\overline{SG0}$ replaces SG1 on the feedback multiplexer. This accommodates CLK being the adjacent pin for $MC_7$ and $\overline{OE}$ the adjacent pin for $MC_0$.

## Registered Output Configuration

The control bit settings are SG0 = 0, SG1 = 1 and $SL0_x$ = 0. There is only one registered configuration. All eight product terms are available as inputs to the OR gate. Data polarity is determined by $SL1_x$. The flip-flop is loaded on the LOW-to-HIGH transition of CLK. The feedback path is from $\overline{Q}$ on the register. The output buffer is enabled by $\overline{OE}$.

## Combinatorial Configurations

The PALCE16V8 has three combinatorial output configurations: dedicated output in a non-registered device, I/O in a non-registered device and I/O in a registered device.

## Dedicated Output In a Non-Registered Device

The control bit settings are SG0 = 1, SG1 = 0 and $SL0_x$ = 0. All eight product terms are available to the OR gate. Although the macrocell is a dedicated output, the feedback is used, with the exception of pins 15 and 16. Pins 15 and 16 do not use feedback in this mode. Because CLK and $\overline{OE}$ are not used in a non-registered device, pins 1 and 11 are available as input signals. Pin 1 will use the feedback path of $MC_7$ and pin 11 will use the feedback path of $MC_0$.

## Combinatorial I/O In a Non-Registered Device

The control bit settings are SG0 = 1, SG1 = 1, and $SL0_x$ = 1. Only seven product terms are available to the OR gate. The eighth product term is used to enable the output buffer. The signal at the I/O pin is fed back to the AND array via the feedback multiplexer. This allows the pin to be used as an input.

Because CLK and $\overline{OE}$ are not used in a non-registered device, pins 1 and 11 are available as inputs. Pin 1 will use the feedback path of $MC_7$ and pin 11 will use the feedback path of $MC_0$.

## Combinatorial I/O in a Registered Device

The control bit settings are SG0 = 0, SG1 = 1 and $SL0_x$ = 1. Only seven product terms are available to the OR gate. The eighth product term is used as the output enable. The feedback signal is the corresponding I/O signal.

## Dedicated Input Configuration

The control bit settings are SG0 = 1, SG1 = 0 and $SL0_x$ = 1. The output buffer is disabled. Except for $MC_0$ and $MC_7$ the feedback signal is an adjacent I/O. For $MC_0$ and $MC_7$ the feedback signals are pins 1 and 11. These configurations are summarized in Table 1 and illustrated in Figure 2.

**Table 1. Macrocell Configuration**

| SG0 | SG1 | SL0x | Cell Configuration | Devices Emulated |
|---|---|---|---|---|
| **Device Uses Registers** | | | | |
| 0 | 1 | 0 | Registered Output | PAL16R8, 16R6, 16R4 |
| 0 | 1 | 1 | Combinatorial I/O | PAL16R6, 16R4 |
| **Device Uses No Registers** | | | | |
| 1 | 0 | 0 | Combinatorial Output | PAL10H8, 12H6, 14H4, 16H2, 10L8, 12L6, 14L4, 16L2 |
| 1 | 0 | 1 | Input | PAL12H6, 14H4, 16H2, 12L6, 14L4, 16L2 |
| 1 | 1 | 1 | Combinatorial I/O | PAL16L8 |

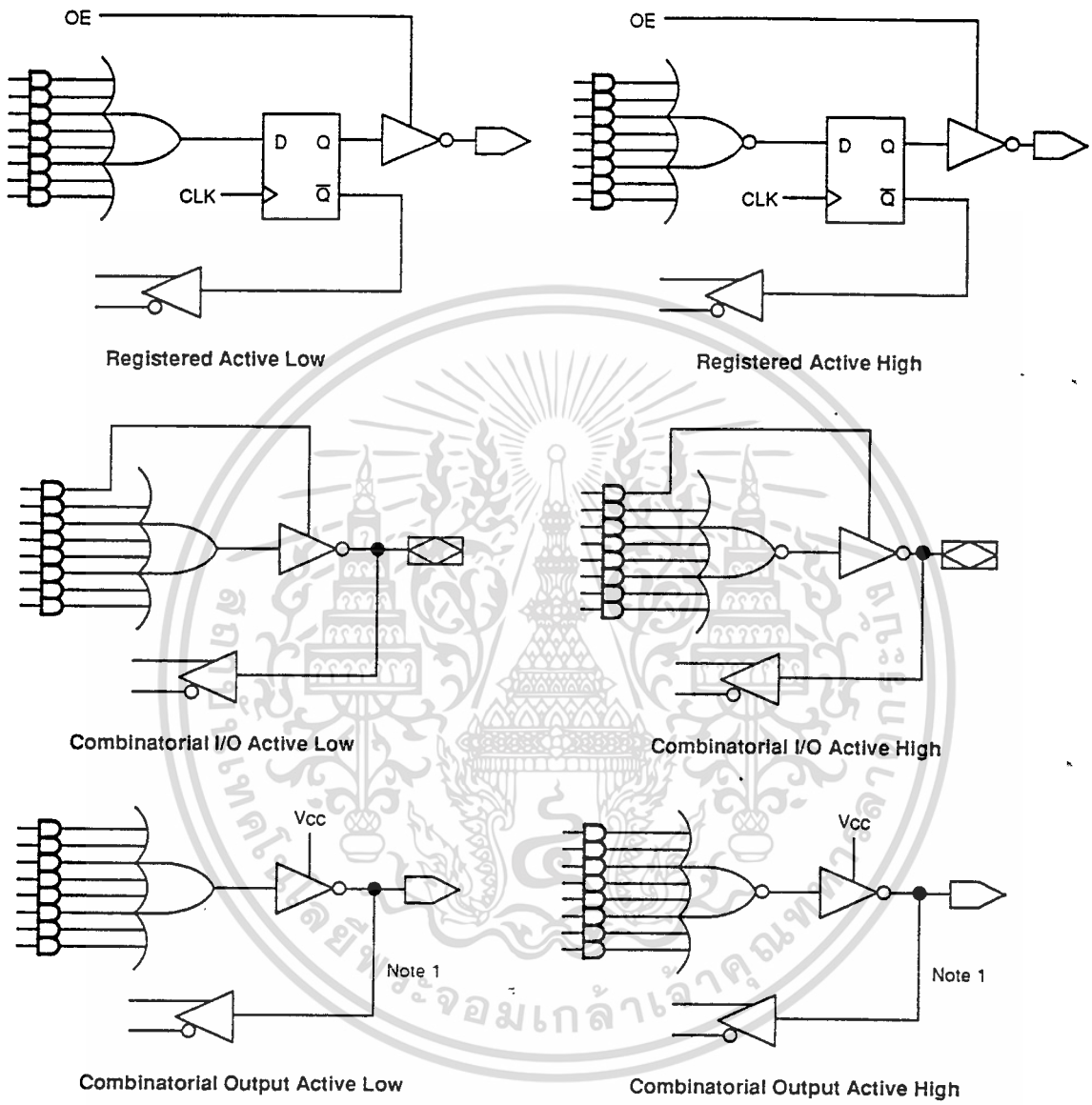
## Programmable Output Polarity

The polarity of each macrocell can be active-high or active-low, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is through a programmable bit $SL1_x$ which controls an exclusive-OR gate at the output of the AND/OR logic. The output is active high if $SL1_x$ is 1 and active low if $SL1_x$ is 0.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**Notes:**

1. Feedback is not available on pins 15 and 16 in the combinatorial output mode.
2. This configuration is not available on pins 15 and 16.

Figure 2. Macrocell Configurations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Power-Up Reset

All flip-flops power up to a logic LOW for predictable system initialization. Outputs of the PALCE16V8 will depend on whether they are selected as registered or combinatorial. If registered is selected, the output will be HIGH. If combinatorial is selected, the output will be a function of the logic.

## Register Preload

The register on the PALCE16V8 can be preloaded from the output pins to facilitate functional testing of complex state machine designs. This feature allows direct loading of arbitrary states, making it unnecessary to cycle through long test vector sequences to reach a desired state. In addition, transitions from illegal states can be verified by loading illegal states and observing proper recovery.

## Security Bit

A security bit is provided on the PALCE16V8 as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, this bit defeats readback and verification of the programmed pattern by a device programmer, securing proprietary designs from competitors. The bit can only be erased in conjunction with the array during an erase cycle.

## Electronic Signature Word

An electronic signature word is provided in the PALCE16V8 device. It consists of 64 bits of programmable memory that can contain user-defined data. The signature data is always available to the user independent of the security bit.

## Programming and Erasing

The PALCE16V8 can be programmed on standard logic programmers. It also may be erased to reset a previously configured device back to its virgin state. Erasure is automatically performed by the programming hardware. No special erase operation is required.

## Quality and Testability

The PALCE16V8 offers a very high level of built-in quality. The erasability of the device provides a direct means of verifying performance of all AC and DC parameters. In addition, this verifies complete programmability and functionality of the device to provide the highest programming yields and post-programming functional yields in the industry.

## Technology

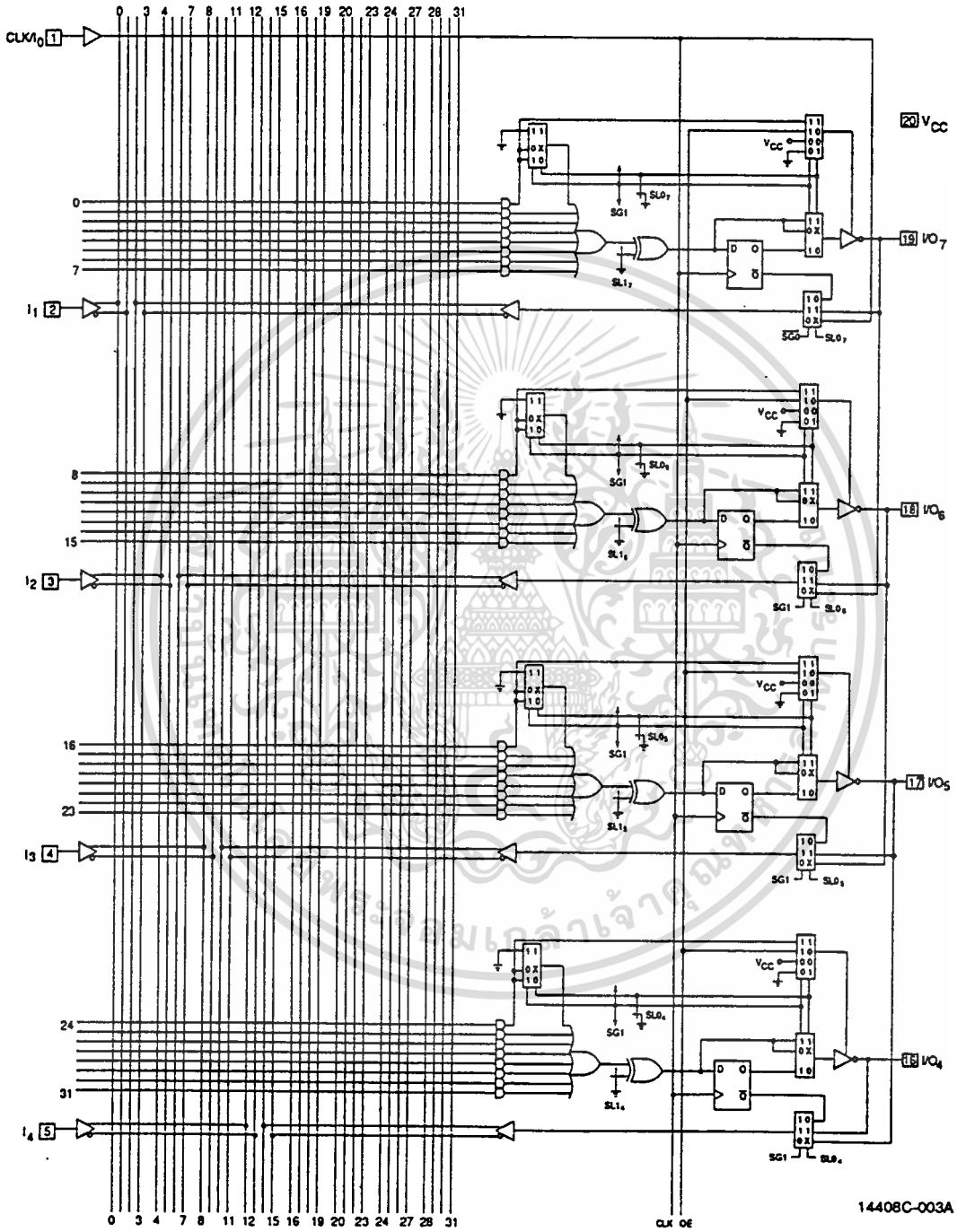
The high-speed PALCE16V8 is fabricated with AMD's advanced electrically erasable (EE) CMOS process. The array connections are formed with proven EE cells. Inputs and outputs are designed to be compatible with TTL devices. This technology provides strong input clamp diodes, output slew-rate control, and a grounded substrate for clean switching.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## LOGIC DIAGRAM






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Storage Temperature | −65°C to +150°C |
| Ambient Temperature with Power Applied | −55°C to +125°C |
| Supply Voltage with Respect to Ground | −0.5 V to +7.0 V |
| DC Input Voltage | −0.5 V to Vcc + 1.0 V |
| DC Output or I/O Pin Voltage | −0.5 V to Vcc + 1.0 V |
| Static Discharge Voltage | 2001 V |
| Latchup Current (Tc = −55°C to +125°C) | 100 mA |

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ. Absolute Maximum Ratings are for system design reference; parameters given are not tested.*

## OPERATING RANGES

Military (M) Devices (Note 1)

| | |
|---|---|
| Operating Case Temperature (Tc) | −55°C to +125°C |
| Supply Voltage (Vcc) with Respect to Ground | +4.5 V to +5.5 V |

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

Note:

1. Military products are tested at Tc = +25°C, +125°C and −55°C, per MIL-STD-883.

## DC CHARACTERISTICS over MILITARY operating ranges unless otherwise specified (Note 2)

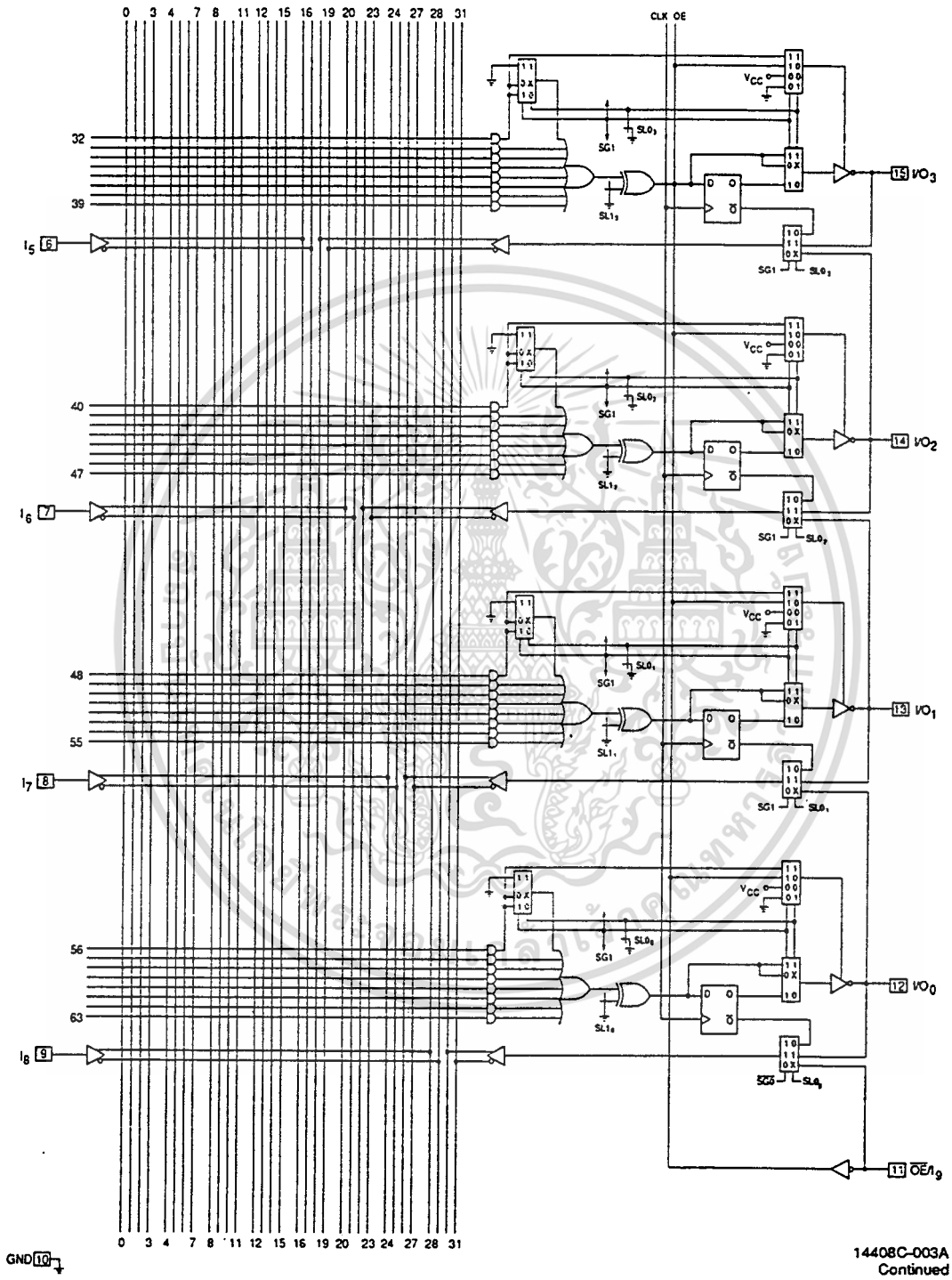
| Parameter Symbol | Parameter Description | Test Conditions | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{OH}$ | Output HIGH Voltage | $I_{OH}$ = −2.0 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | 2.4 | | V |
| $V_{OL}$ | Output LOW Voltage | $I_{OL}$ = 12 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | | 0.5 | V |
| $V_{IH}$ | Input HIGH Voltage | Guaranteed Input Logical HIGH Voltage for all Inputs (Note 3) | 2.0 | | V |
| $V_{IL}$ | Input LOW Voltage | Guaranteed Input Logical LOW Voltage for all Inputs (Note 3) | | 0.8 | V |
| $I_{IH}$ | Input HIGH Leakage Current | $V_{IN}$ = 5.5 V, Vcc = Max. (Note 4) | | 10 | µA |
| $I_{IL}$ | Input LOW Leakage Current | $V_{IN}$ = 0 V, Vcc = Max. (Note 4) | | −10 | µA |
| $I_{OZH}$ | Off-State Output Leakage Current HIGH | $V_{OUT}$ = 5.5 V, Vcc = Max. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ (Note 4) | | 10 | µA |
| $I_{OZL}$ | Off-State Output Leakage Current LOW | $V_{OUT}$ = 0 V, Vcc = Max. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ (Note 4) | | −100 | µA |
| $I_{SC}$ | Output Short-Circuit Current | Vcc = 5.0 V, $V_{OUT}$ = 0.5 V (Note 5), T = 25°C | −30 | −150 | mA |
| $I_{CC}$ | Supply Current (Dynamic) | Outputs Open ($I_{OUT}$ = 0 mA) Vcc = Max., f = 25 MHz | | 130 | mA |

**Notes:**

2. For APL products, Group A, Subgroups 1, 2 and 3 are tested per MIL-STD-833, Method 5005, unless otherwise noted.
3. $V_{IL}$ and $V_{IH}$ are input conditions of output tests and are not themselves directly tested. $V_{IL}$ and $V_{IH}$ are absolute voltages with respect to device ground and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.
4. I/O pin leakage is the worst case of $I_{IL}$ and $I_{OZL}$ (or $I_{IH}$ and $I_{OZH}$).
5. Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second. $V_{OUT}$ = 0.5 V has been chosen to avoid test problems caused by tester ground degradation. This parameter is not 100% tested, but is evaluated at initial characterization and at any time the design is modified where $I_{SC}$ may be affected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## LOGIC DIAGRAM (Continued)




14408C-003A
Continued



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CAPACITANCE (Note 1)

| Parameter Symbol | Parameter Descriptions | Test Conditions | | Typ. | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 2.0 V | $V_{CC}$ = 5.0 V, $T_A$ = 25°C, f = 1 MHz | 8 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 2.0 V | | 8 | pF |

**Note:**

1. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

## SWITCHING CHARACTERISTICS over MILITARY operating ranges (Note 2)

| Parameter Symbol | Parameter Description | | | -20 Min. | -20 Max. | -25 Min. | -25 Max. | Unit |
|---|---|---|---|---|---|---|---|---|
| $t_{PD}$ | Input or Feedback to Combinatorial Output | | | | 20 | | 25 | ns |
| $t_S$ | Setup Time from Input or Feedback to Clock | | | 15 | | 15 | | |
| $t_H$ | Hold Time | | | 0 | | 0 | | ns |
| $t_{CO}$ | Clock to Output | | | | 15 | | 20 | ns |
| $t_{WL}$ | Clock Width | LOW | | 12 | | 15 | | ns |
| $t_{WH}$ | | HIGH | | 12 | | 15 | | ns |
| $f_{MAX}$ | Maximum Frequency (Note 3) | External Feedback | 1/(ts+tco) | 33.3 | | 28.6 | | MHz |
| | | Internal Feedback ($f_{CNT}$) | | 35.7 | | 30.3 | | MHz |
| | | No Feedback | 1/(twH+twL) | 41.7 | | 33.3 | | MHz |
| $t_{PZX}$ | $\overline{OE}$ to Output Enable (Note 3) | | | | 20 | | 20 | ns |
| $t_{PXZ}$ | $\overline{OE}$ to Output Disable (Note 3) | | | | 20 | | 20 | ns |
| $t_{EA}$ | Input to Output Enable Using Product Term Control (Note 3) | | | | 20 | | 25 | ns |
| $t_{ER}$ | Input to Output Disable Using Product Term Control (Note 3) | | | | 20 | | 55 | ns |

**Notes:**

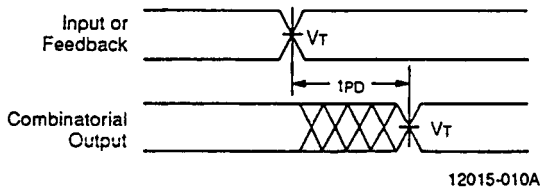
2. See Switching Test Circuit for test conditions. For APL Products, Group A, Subgroups 9, 10, and 11 are tested per MIL-STD-883, Method 5005, unless otherwise noted.
3. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where these parameters may be affected.



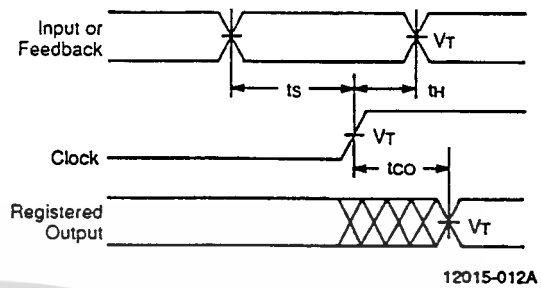
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

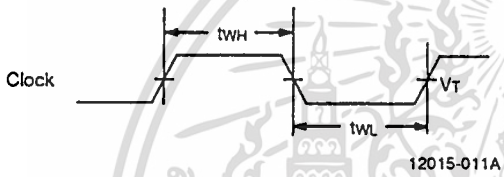
## SWITCHING WAVEFORMS



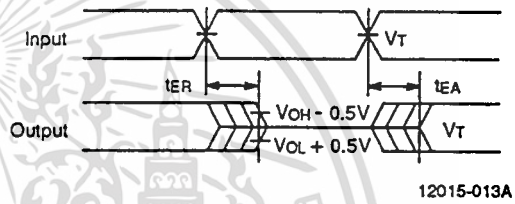

Combinatorial Output



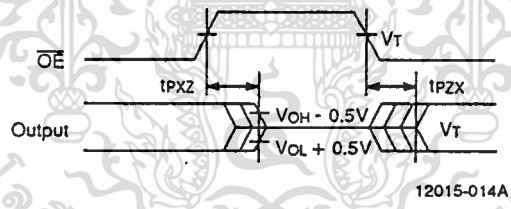

Registered Output




Clock Width




Input to Output Disable/Enable




OE to Output Disable/Enable

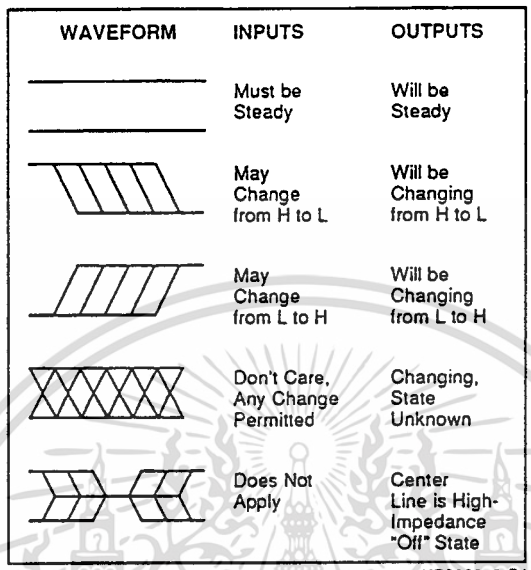
**Notes:**

1. $V_T$ = 1.5 V
2. Input pulse amplitude 0 V to 3.0 V.
3. Input rise and fall times 2–5 ns typical.

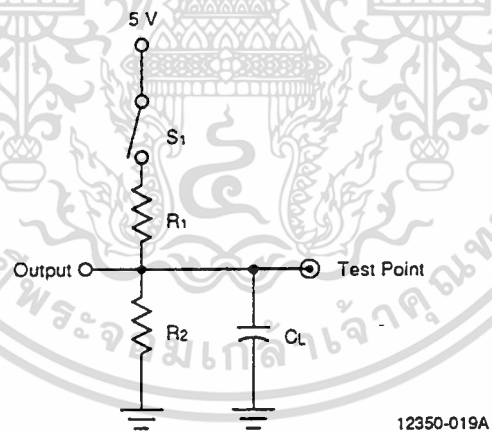


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## KEY TO SWITCHING WAVEFORMS




## SWITCHING TEST CIRCUIT




| Specification | $S_1$ | $C_L$ | Commercial | | Military | | Measured Output Value |
|---|---|---|---|---|---|---|---|
| | | | $R_1$ | $R_2$ | $R_1$ | $R_2$ | |
| $t_{PD}$, $t_{CO}$ | Closed | 50 pF | 200 Ω | 390 Ω | 390 Ω | 750 Ω | 1.5 V |
| $t_{PZX}$, $t_{EA}$ | Z → H: Open<br>Z → L: Closed | | | | | | 1.5 V |
| $t_{PXZ}$, $t_{ER}$ | H → Z: Open<br>L → Z: Closed | 5 pF | | | | | H → Z: $V_{OH}$ – 0.5 V<br>L → Z: $V_{OL}$ + 0.5 V |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

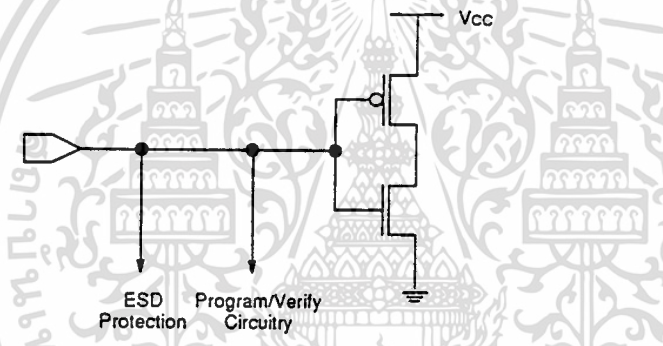
## ENDURANCE CHARACTERISTICS

The PALCE16V8 is manufactured using AMD's advanced Electrically Erasable process. This technology uses an EE cell to replace the fuse link used in bipolar parts. As a result, the device can be erased and reprogrammed – a feature which allows 100% testing at the factory.

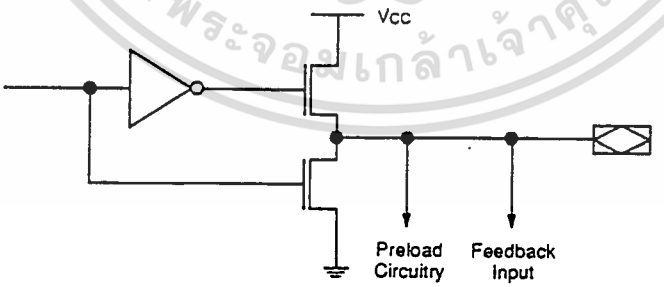
### Endurance Characteristics

| Symbol | Parameter | Min. | Units | Test Conditions |
|---|---|---|---|---|
| $t_{DR}$ | Min. Pattern Data Retention Time | 10 | Years | Max. Storage Temperature |
| | | 20 | Years | Max. Operating Temperature (Military) |
| N | Min. Reprogramming Cycles | 100 | Cycles | Normal Programming Conditions |

## INPUT/OUTPUT EQUIVALENT SCHEMATICS




Typical Input




Typical Output

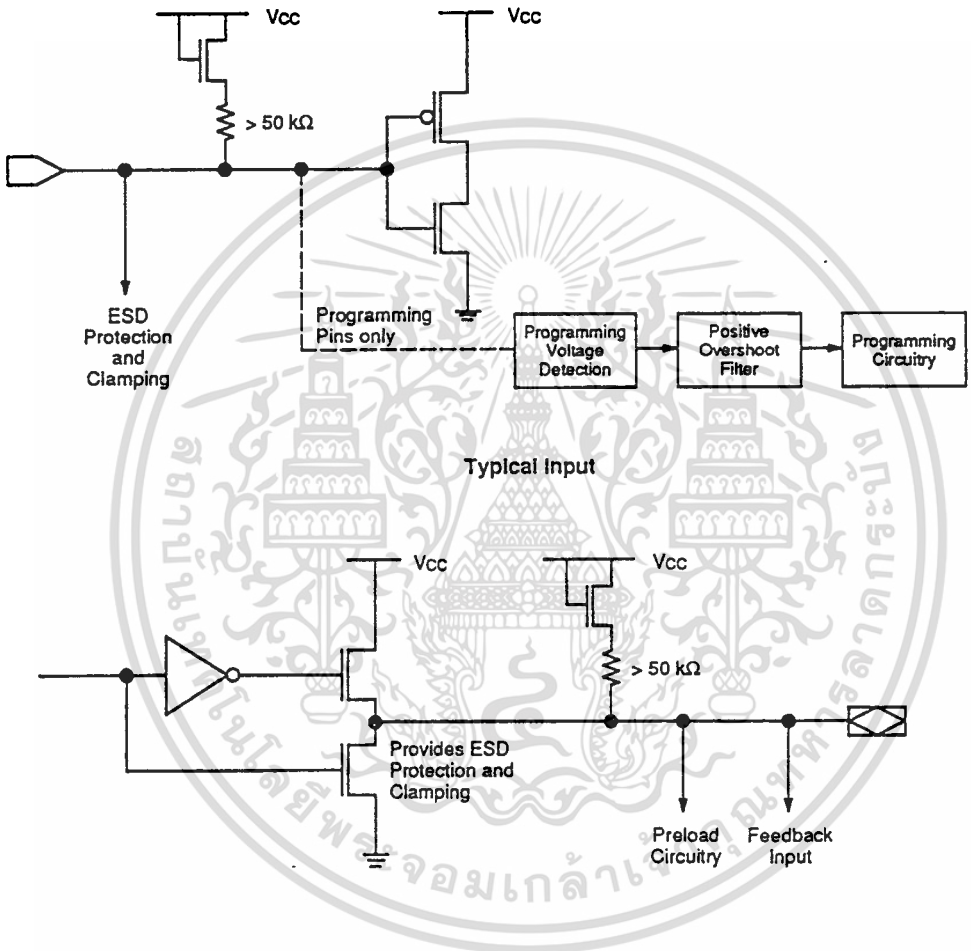
12197-013A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ROBUSTNESS FEATURES FOR /5 VERSIONS

The PALCE16V8H-7/5 has some unique features that make it extremely robust, especially when operating in high-speed design environments. Pull-up resistors on inputs and I/O pins cause unconnected pins to default to a known state. Input clamping circuitry limits negative overshoot, eliminating the possibility of false clocking caused by subsequent ringing. A special noise filter makes the programming circuitry completely insensitive to any positive overshoot that has a pulse width of less than about 100 ns.

## INPUT/OUTPUT EQUIVALENT SCHEMATICS FOR /5 VERSION




16407A-001B



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

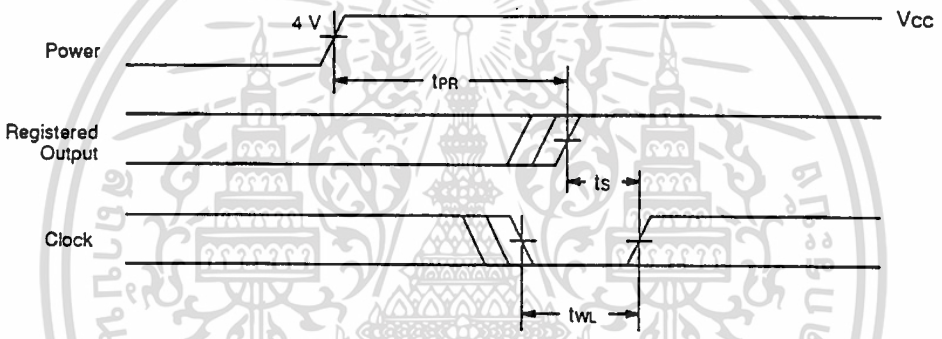
AMD

## POWER-UP RESET

The PALCE16V8 has been designed with the capability to reset during system power-up. Following power-up, all flip-flops will be reset to LOW. The output state will be HIGH independent of the logic polarity. This feature provides extra flexibility to the designer and is especially valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways Vcc can rise to its steady state, two conditions are required to insure a valid power-up reset. These conditions are:

1. The Vcc rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

| Parameter Symbol | Parameter Descriptions | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{PR}$ | Power-Up Reset Time | | 1000 | ns |
| $t_S$ | Input or Feedback Setup Time | See Switching Characteristics | | |
| $t_{WL}$ | Clock Width LOW | | | |




Power-Up Reset Waveform



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM'L: H-10/15/25, Q-15/25 MIL: H-15/20/25

# PALCE20V8 Family

## EE CMOS 24-Pin Universal Programmable Array Logic

Advanced Micro Devices

## DISTINCTIVE CHARACTERISTICS

- Pin, function and fuse-map compatible with all GAL 20V8/As
- Electrically erasable CMOS technology provides reconfigurable logic and full testability
- High-speed CMOS technology
  - -10 ns propagation delay for "-10" version
  - -15 ns propagation delay for "-15" version
- Direct plug-in replacement for a wide range of 24-pin PAL devices
- Outputs individually programmable as registered or combinatorial
- Programmable output polarity
- Programmable enable/disable control
- Preloadable output registers for testability
  Automatic register reset on power-up
- Cost-effective 24-pin plastic SKINNYDIP and 28-pin PLCC packages
- Extensive third-party software and programmer support through FusionPLD partners
- Fully tested for 100% programming and functional yields and high reliability

## GENERAL DESCRIPTION

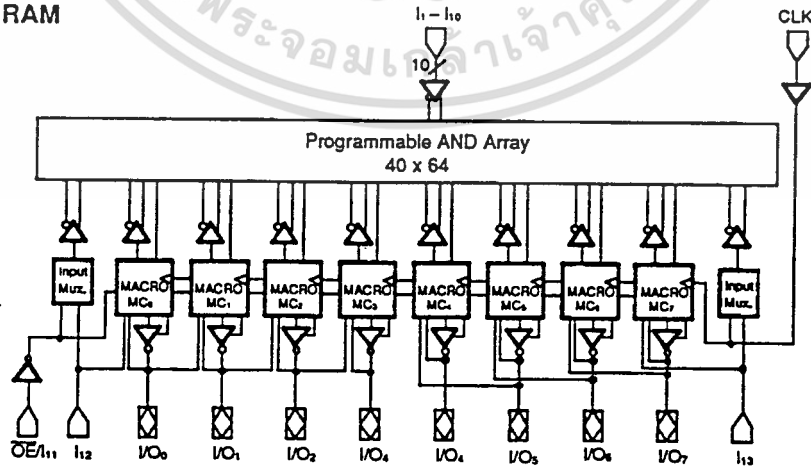
The PALCE20V8 is an advanced PAL device built with low-power, high-speed, electrically-erasable CMOS technology. Its macrocells provide a universal device architecture. The PALCE20V8 is fully compatible with the GAL20V8 and can directly replace PAL20R8 series devices and most 24-pin combinatorial PAL devices.

Device logic is automatically configured according to the user's design specification. A design is implemented using any of a number of popular design software packages, allowing automatic creation of a programming file based on Boolean or state equations. Design software also verifies the design and can provide test vectors for the finished device. Programming can be accomplished on standard PAL device programmers.

The PALCE20V8 utilizes the familiar sum-of-products (AND/OR) architecture that allows users to implement complex logic functions easily and efficiently. Multiple levels of combinatorial logic can always be reduced to sum-of-products form, taking advantage of the very wide input gates available in PAL devices. The equations are programmed into the device through floating-gate cells in the AND logic array that can be erased electrically.

The fixed OR array allows up to eight data product terms per output for logic functions. The sum of these products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial with an active-high or active-low output. The output configuration is determined by two global bits and one local bit controlling four multiplexers in each macrocell.

## BLOCK DIAGRAM

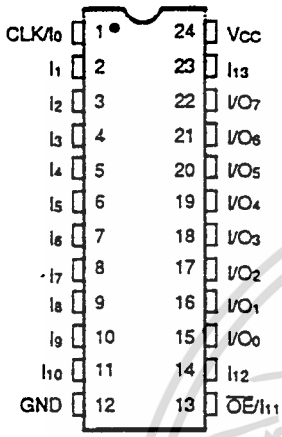



Publication# 16491 Rev. A Amendment/0
Issue Date: January 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

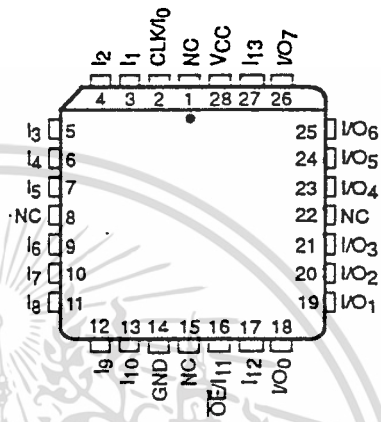
## CONNECTION DIAGRAMS
(Top View)

### SKINNYDIP




12197-002B

### PLCC/LCC




12197-003A

**Note:** Pin 1 is marked for orientation.

## PIN DESIGNATIONS

| | | |
|---|---|---|
| CLK | = | Clock |
| GND | = | Ground |
| I | = | Input |
| I/O | = | Input/Output |
| NC | = | No Connect |
| $\overline{OE}$ | = | Output Enable |
| Vcc | = | Supply Voltage |

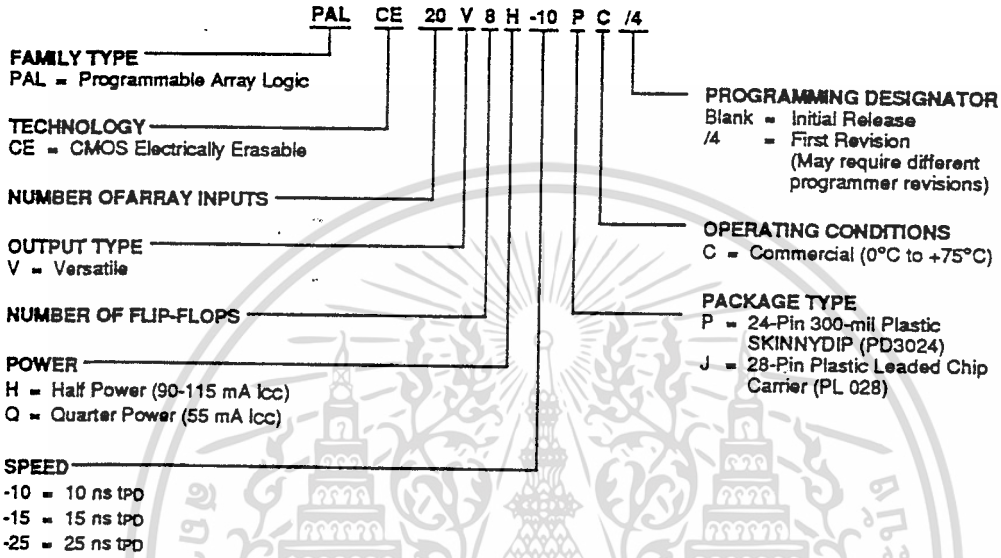


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ORDERING INFORMATION
### Commercial Products

AMD programmable logic products for commercial applications are available with several ordering options. The order number (Valid Combination) is formed by a combination of:




| Valid Combinations | | |
|---|---|---|
| PALCE20V8H-10 | PC, JC | /4 |
| PALCE20V8H-15 | | |
| PALCE20V8H-25 | | Blank, /4 |
| PALCE20V8Q-15 | | |
| PALCE20V8Q-25 | | |

**Valid Combinations**

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

**Note:** Marked with AMD logo.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

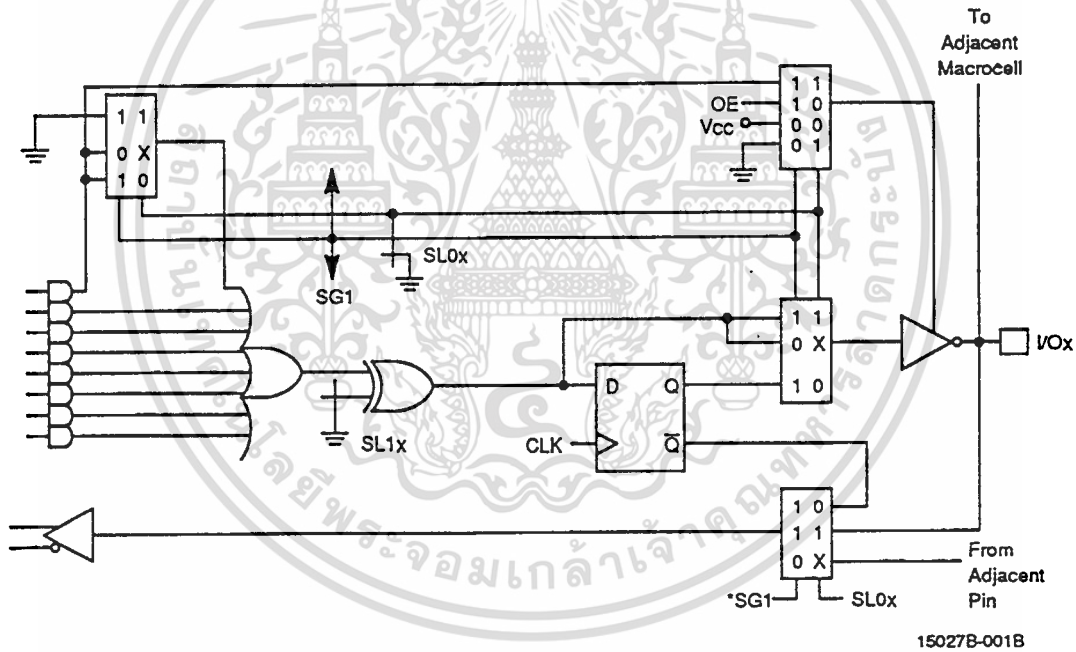
## FUNCTIONAL DESCRIPTION

The PALCE20V8 is a universal PAL device. It has eight independently configurable macrocells ($MC_0..MC_7$). Each macrocell can be configured as a registered output, combinatorial output, combinatorial I/O, or dedicated input. The programming matrix implements a programmable AND logic array, which drives a fixed OR logic array. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Pins 1 and 13 serve either as array inputs or as clock (CLK) and output enable ($\overline{OE}$) for all flip-flops.

Unused input pins should be tied directly to Vcc or GND. Product terms with all bits unprogrammed (disconnected) assume the logical HIGH state and product terms with both true and complement of any input signal connected assume a logical LOW state.

The programmable functions on the PALCE20V8 are automatically configured from the user's design specification, which can be in a number of formats. The design specification is processed by development software to verify the design and create a programming file. This file, once downloaded to a programmer, configures the device according to the user's desired function.

The user is given two design options with the PALCE20V8. First, it can be programmed as an emulated PAL device. This includes the PAL20R8 series and most 24-pin combinatorial PAL devices. The PAL device programmer manufacturer will supply device codes for the standard PAL architectures to be used with the PALCE20V8. The programmer will program the PALCE20V8 to the corresponding PAL device architecture. This allows the user to use existing standard PAL device JEDEC files without making any changes to them. Alternatively, the device can be programmed directly as a PALCE20V8. Here the user must use the PALCE20V8 device code. This option provides full utilization of the macrocells, allowing non-standard architectures to be built.




* In Macrocells $MC_0$ and $MC_7$, SG1 is replaced by $\overline{SG0}$ on the feedback multiplexer.

**Figure 1. PALCE20V8 Macrocell**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Configuration Options

Each macrocell can be configured as one of the following: registered output, combinatorial output, combinatorial I/O or dedicated input. In the registered output configuration, the output buffer is enabled by the $\overline{OE}$ pin. In the combinatorial configuration, the buffer is either controlled by a product term or always enabled. In the dedicated input configuration, the buffer is always disabled. A macrocell configured as a dedicated input derives the input signal from an adjacent I/O.

The macrocell configurations are controlled by the configuration control word. It contains 2 global bits (SG0 and SG1) and 16 local bits ($SL0_0$ through $SL0_7$ and $SL1_0$ through $SL1_7$). SG0 determines whether registers will be allowed. SG1 determines whether the PALCE20V8 will emulate a PAL20R8 family or a combinatorial device. Within each macrocell, $SL0_x$, in conjunction with SG1, selects the configuration of the macrocell and $SL1_x$ sets the output as either active low or active high.

The configuration bits work by acting as control inputs for the multiplexers in the macrocell. There are four multiplexers: a product term input, an enable select, an output select, and a feedback select multiplexer. SG1 and $SL0_x$ are the control signals for all four multiplexers. In $MC_0$ and $MC_7$, $\overline{SG0}$ replaces SG1 on the feedback multiplexer.

These configurations are summarized in table 1 and illustrated in figure 2.

If the PALCE20V8 is configured as a combinatorial device, the CLK and $\overline{OE}$ pins may be available as inputs to the array. If the device is configured with registers, the CLK and $\overline{OE}$ pins cannot be used as data inputs.

## Registered Output Configuration

The control bit settings are SG0 = 0, SG1 = 1 and $SL0_x$ = 0. There is only one registered configuration. All eight product terms are available as inputs to the OR gate. Data polarity is determined by $SL1_x$. $SL1_x$ is an input to the exclusive-OR gate which is the D input to the flip-flop. $SL1_x$ is programmed as 1 for inverted output or 0 for non-inverted output. The flip-flop is loaded on the LOW-to-HIGH transition of CLK. The feedback path is from $\overline{Q}$ on the register. The output buffer is enabled by $\overline{OE}$.

## Combinatorial Configurations

The PALCE20V8 has three combinatorial output configurations: dedicated output in a non-registered device, I/O in a non-registered device and I/O in a registered device.

## Dedicated Output in a Non-Registered Device

The control settings are SG0 = 1, SG1 = 0, and $SL0_x$ = 0. All eight product terms are available to the OR gate. Although the macrocell is a dedicated output, the feedback is used, with the exception of pins 18(21) and 19(23). Pins 18(21) and 19(23) do not use feedback in this mode.

## Dedicated Input in a Non-Registered Device

The control bit settings are SG0 = 1, SG1 = 0 and $SL0_x$ = 1. The output buffer is disabled. The feedback signal is an adjacent I/O pin.

## Combinatorial I/O in a Non-Registered Device

The control settings are SG0 = 1, SG1 = 1, and $SL0_x$ = 1. Only seven product terms are available to the OR gate. The eighth product term is used to enable the output buffer. The signal at the I/O pin is fed back to the AND array via the feedback multiplexer. This allows the pin to be used as an input.

## Combinatorial I/O in a Registered Device

The control bit settings are SG0=0, $\overline{SG1}$=1 and $\overline{SL0_x}$ =1. Only seven product terms are available to the OR gate. The eighth product term is used as the output enable. The feedback signal is the corresponding I/O signal.

**Table 1. Macrocell Configurations**

| SG0 | SG1 | $SL0_x$ | Cell Configuration | Devices Emulated |
|---|---|---|---|---|
| | | | **Device has registers** | |
| 0 | 1 | 0 | Registered Output | PAL20R8, 20R6, 20R4 |
| 0 | 1 | 1 | Combinatorial I/O | PAL20R6, 20R4 |
| | | | **Device has no registers** | |
| 1 | 0 | 0 | Combinatorial Output | PAL20L2, 18L4,16L6,14L8 |
| 1 | 0 | 1 | Dedicated Input | PAL20L2,18L4, 16L6 |
| 1 | 1 | 1 | Combinatorial I/O | PAL20L8 |

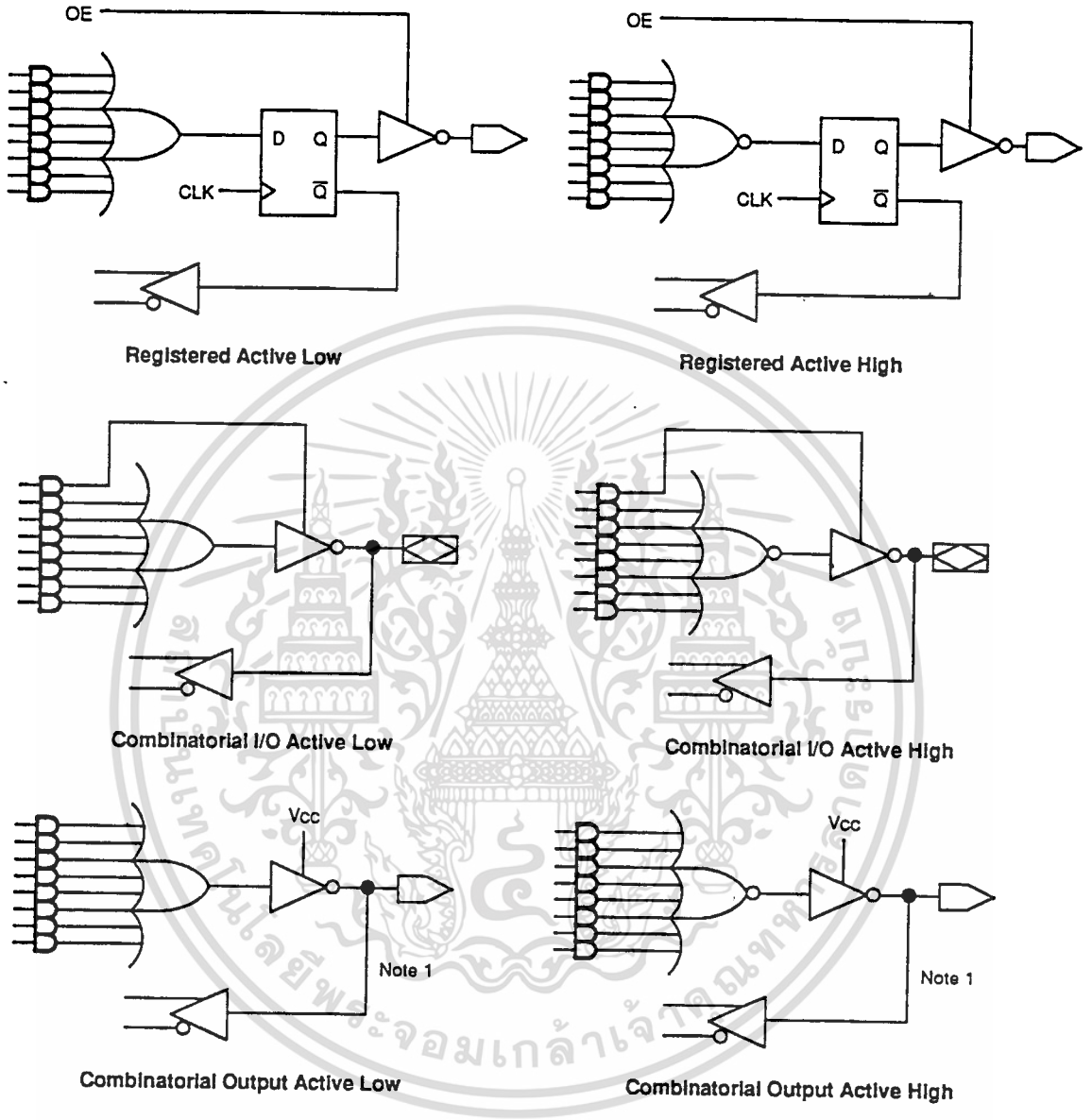
## Programmable Output Polarity

The polarity of each macrocell output can be active high or active low, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is made through a programmable bit $SL1_x$ which controls an exclusive-OR gate at the output of the AND/OR logic. The output is active high if $SL1_x$ is a 0 and active low if $SL1_x$ is a 1.

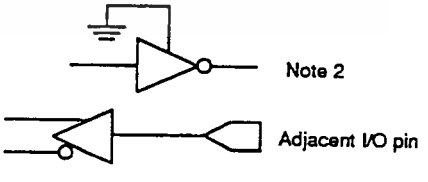


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**Notes:**

1. Feedback is not available on pins 18 (21) and 19 (23) in the combinatorial output mode.
2. This macrocell configuration is not available on pins 18 (21) and 19 (23).




12197-012C

**Figure 2. Macrocell Configurations**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Power-Up Reset

All flip-flops power up to a logic LOW for predictable system initialization. Outputs of the PALCE20V8 depend on whether they are selected as registered or combinatorial. If registered is selected, the output will be HIGH. If combinatorial is selected, the output will be a function of the logic.

## Register Preload

The register on the PALCE20V8 can be preloaded from the output pins to facilitate functional testing of complex state machine designs. This feature allows direct loading of arbitrary states, making it unnecessary to cycle through long test vector sequences to reach a desired state. In addition, transitions from illegal states can be verified by loading illegal states and observing proper recovery.

## Security Bit

A security bit is provided on the PALCE20V8 as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, this bit defeats readback and verification of the programmed pattern by a device programmer, securing proprietary designs from competitors. The bit can only be erased in conjunction with the array during an erase cycle.

## Electronic Signature Word

An electronic signature word is provided in the PALCE20V8. It consists of 64 bits of programmable memory that can contain any user-defined data. The signature data is always available to the user independent of the security bit.

## Programming and Erasing

The PALCE20V8 can be programmed on standard logic programmers. It also may be erased to reset a previously configured device back to its virgin state. Erasure is automatically performed by the programming hardware. No special erase operation is required.

## Quality and Testability

The PALCE20V8 offers a very high level of built-in quality. The erasability of the device provides a direct means of verifying performance of all AC and DC parameters. In addition, this verifies complete programmability and functionality of the device to provide the highest programming and post-programming functional yields in the industry.

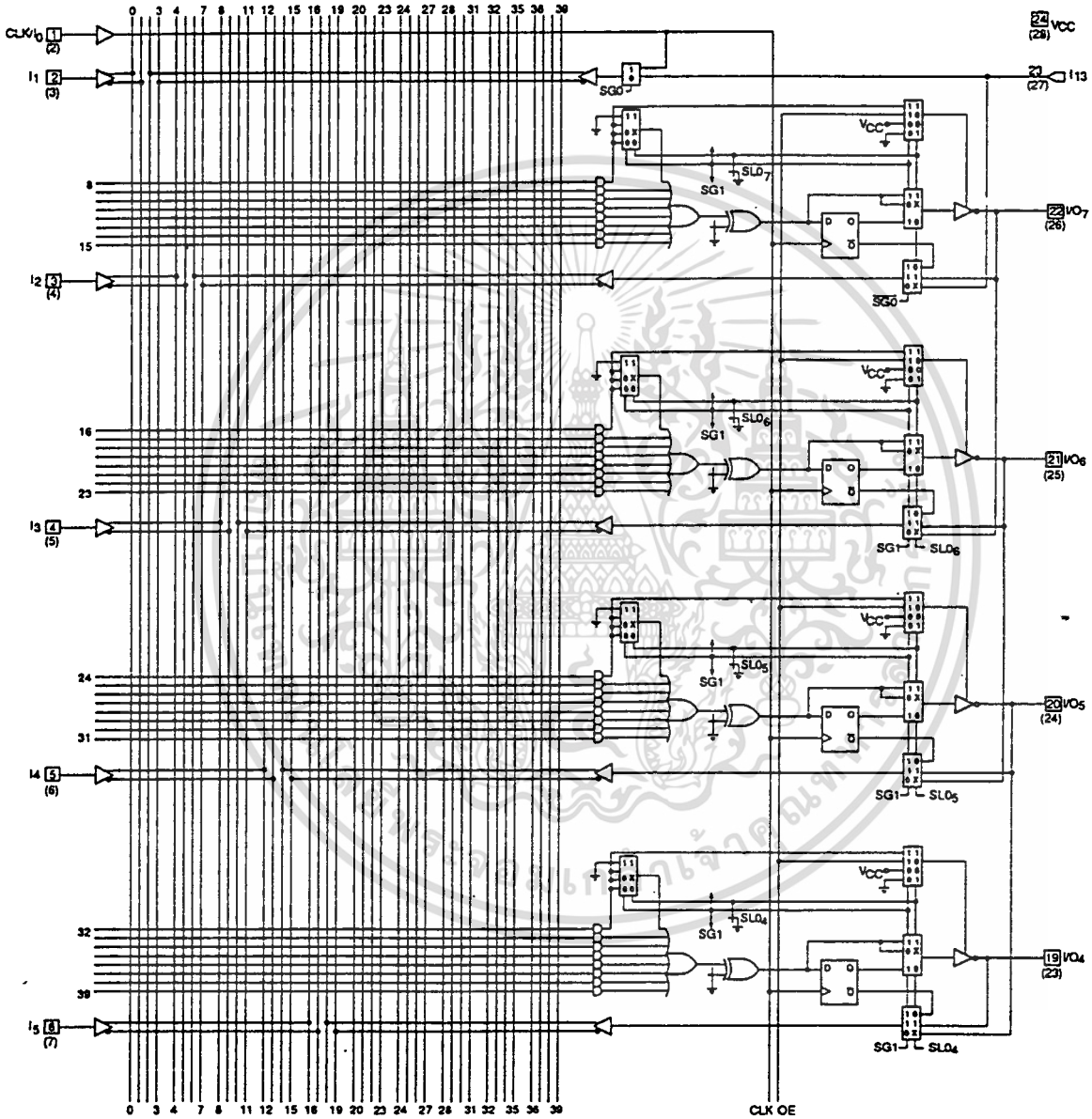
## Technology

The high-speed PALCE20V8H is fabricated with AMD's advanced electrically erasable (EE) CMOS process. The array connections are formed with proven EE cells. Inputs and outputs are designed to be compatible with TTL devices. This technology provides strong input clamp diodes, output slew-rate control, and a grounded substrate for clean switching.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## LOGIC DIAGRAM
## SKINNYDIP (PLCC and LCC) Pinouts




15027B-002A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Storage Temperature | –65°C to +150°C |
| Ambient Temperature with Power Applied | –55°C to +125°C |
| Supply Voltage with Respect to Ground | –0.5 V to +7.0 V |
| DC Input Voltage | –0.5 V to Vcc + 0.5 V |
| DC Output or I/O Pin Voltage | –0.5 V to Vcc + 0.5 V |
| Static Discharge Voltage | 2001 V |
| Latchup Current (Tc = –55°C to +125°C) | 100 mA |

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ. Absolute Maximum Ratings are for system design reference; parameters given are not tested.*

## OPERATING RANGES

**Military (M) Devices (Note 1)**

| | |
|---|---|
| Operating Case Temperature (Tc) | –55°C to +125°C |
| Supply Voltage (Vcc) with Respect to Ground | +4.5 V to +5.5 V |

*Operating ranges define those limits between which the functionality of the device is guaranteed.*

**Note:**

1. Military products are tested at Tc = +25°C, +125°C and –55°C, per MIL-STD-883.

## DC CHARACTERISTICS over MILITARY operating ranges unless otherwise specified (Note 2)

| Parameter Symbol | Parameter Description | Test Conditions | Min. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{OH}$ | Output HIGH Voltage | $I_{OH}$ = –2.0 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | 2.4 | | V |
| $V_{OL}$ | Output LOW Voltage | $I_{OL}$ = 12 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | | 0.5 | V |
| $V_{IH}$ | Input HIGH Voltage | Guaranteed Input Logical HIGH Voltage for all Inputs (Note 3) | 2.0 | | V |
| $V_{IL}$ | Input LOW Voltage | Guaranteed Input Logical LOW Voltage for all Inputs (Note 3) | | 0.8 | V |
| $I_{IH}$ | Input HIGH Leakage Current | $V_{IN}$ = 5.5 V, Vcc = Max. (Note 4) | | 10 | µA |
| $I_{IL}$ | Input LOW Leakage Current | $V_{IN}$ = 0 V, Vcc = Max. (Note 4) | | –10 | µA |
| $I_{OZH}$ | Off-State Output Leakage Current HIGH | $V_{OUT}$ = 5.5 V, Vcc = Max. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ (Note 4) | | 10 | µA |
| $I_{OZL}$ | Off-State Output Leakage Current LOW | $V_{OUT}$ = 0 V, Vcc = Max. $V_{IN}$ = $V_{IH}$ or $V_{IL}$ (Note 4) | | –10 | µA |
| $I_{SC}$ | Output Short-Circuit Current | $V_{OUT}$ = 0.5 V, Vcc = 5.0 V, $T_A$ = 25°C (Note 5) | –30 | –150 | mA |
| $I_{CC}$ | Supply Current (Dynamic) | Outputs Open ($I_{OUT}$ = 0 mA) Vcc = Max., f = 25 MHz | | 130 | mA |

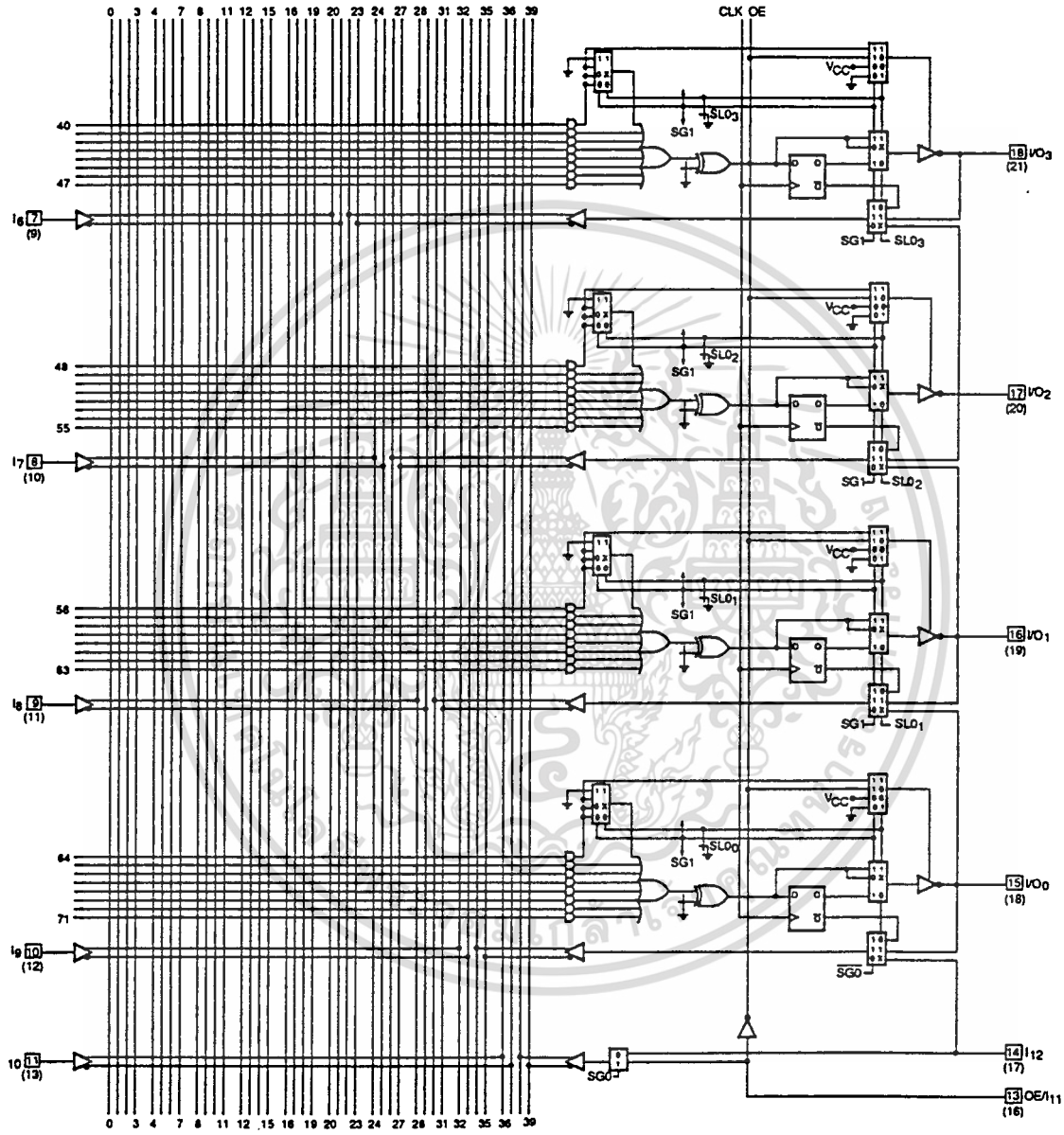
**Notes:**

2. For APL products, Group A, Subgroups 1, 2 and 3 are tested per MIL-STD-833, Method 5005, unless otherwise noted.
3. $V_{IL}$ and $V_{IH}$ are input conditions of output tests and are not themselves directly tested. $V_{IL}$ and $V_{IH}$ are absolute voltages with respect to device ground and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.
4. I/O pin leakage is the worst case of $I_{IL}$ and $I_{OZL}$ (or $I_{IH}$ and $I_{OZH}$).
5. Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second. $V_{OUT}$ = 0.5 V has been chosen to avoid test problems caused by tester ground degradation. This parameter is not 100% tested, but is evaluated at initial characterization and at any time the design is modified where $I_{SC}$ may be affected.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## LOGIC DIAGRAM (Continued)
## SKINNYDIP (PLCC and LCC) Pinouts




15027B-002A
concluded



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CAPACITANCE (Note 1)

| Parameter Symbol | Parameter Description | Test Conditions | | Typ. | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 2.0 V | $V_{CC}$ = 5.0 V, $T_A$ = 25°C, | 8 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 2.0 V | f = 1 MHz | 8 | pF |

**Note:**

1. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

## SWITCHING CHARACTERISTICS over MILITARY operating ranges (Note 2)

| Parameter Symbol | Parameter Description | | | -20 | | -25 | | Unit |
|---|---|---|---|---|---|---|---|---|
| | | | | Min. | Max. | Min. | Max. | |
| $t_{PD}$ | Input or Feedback to Combinatorial Output | | | | 20 | | 25 | ns |
| $t_S$ | Setup Time from Input or Feedback to Clock | | | 15 | | 20 | | ns |
| $t_H$ | Hold Time (Note 5) | | | 0 | | 0 | | ns |
| $t_{CO}$ | Clock to Output | | | | 15 | | 20 | ns |
| $t_{WL}$ | Clock Width | LOW | | 12 | | 15 | | ns |
| $t_{WH}$ | | HIGH | | 12 | | 15 | | ns |
| $f_{MAX}$ | Maximum Frequency (Note 3) | External Feedback | 1/(ts+tco) | 33.3 | | 25 | | MHz |
| | | Internal Feedback ($f_{CNT}$) | | 35.7 | | 26.3 | | MHz |
| | | No Feedback | 1/($t_{WH}$+$t_{WL}$) | 41.7 | | 33.3 | | MHz |
| $t_{PZX}$ | $\overline{OE}$ to Output Enable (Note 3) | | | | 18 | | 20 | ns |
| $t_{PXZ}$ | $\overline{OE}$ to Output Disable (Note 3) | | | | 18 | | 20 | ns |
| $t_{EA}$ | Input to Output Enable Using Product Term Control (Note 3) | | | | 20 | | 25 | ns |
| $t_{ER}$ | Input to Output Disable Using Product Term Control (Note 3) | | | | 20 | | 25 | ns |

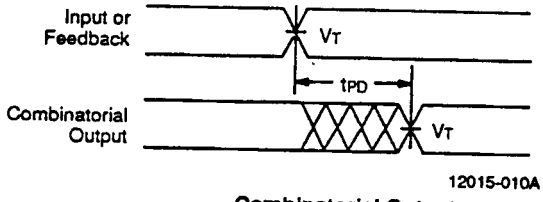
**Notes:**

2. See Switching Test Circuit for test conditions. For APL Products, Group A, Subgroups 9, 10, and 11 are tested per MIL-STD-883, Method 5005, unless otherwise noted.
3. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where these parameters may be affected.

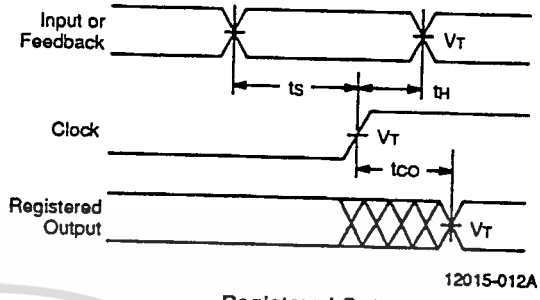


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

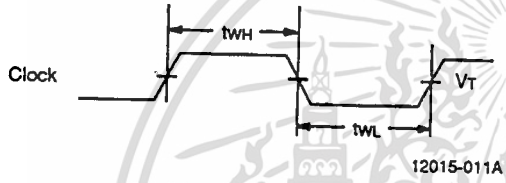
## SWITCHING WAVEFORMS



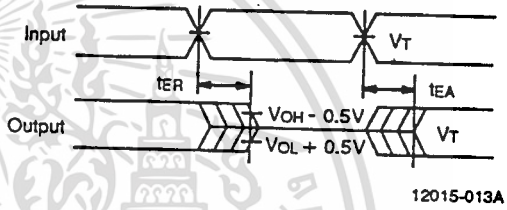

**Combinatorial Output**



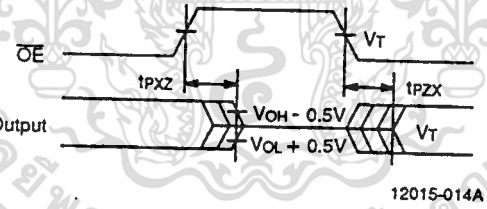

**Registered Output**




**Clock Width**




**Input to Output Disable/Enable**




**$\overline{OE}$ to Output Disable/Enable**

**Notes:**

1. $V_T$ = 1.5 V
2. Input pulse amplitude 0 V to 3.0 V.
3. Input rise and fall times 2–5 ns typical.



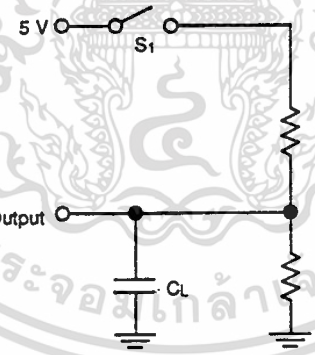
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## KEY TO SWITCHING WAVEFORMS

| WAVEFORM | INPUTS | OUTPUTS |
|---|---|---|
| | Must be Steady | Will be Steady |
| | May Change from H to L | Will be Changing from H to L |
| | May Change from L to H | Will be Changing from L to H |
| | Don't Care, Any Change Permitted | Changing, State Unknown |
| | Does Not Apply | Center Line is High-Impedance "Off" State |

KS000010-PAL

## SWITCHING TEST CIRCUIT




12197-007A

**Switching Test Circuit**

| Specification | $S_1$ | $C_L$ | Commercial | | Military | | Measured Output Value |
|---|---|---|---|---|---|---|---|
| | | | $R_1$ | $R_2$ | $R_1$ | $R_2$ | |
| $t_{PD}$, $t_{CO}$ | Closed | 50 pF | 200 Ω | 390 Ω | 390 Ω | 750 Ω | 1.5 V |
| $t_{PZX}$, $t_{EA}$ | Z → H: Open<br>Z → L: Closed | 50 pF | 200 Ω | 390 Ω | 390 Ω | 750 Ω | 1.5 V |
| $t_{PXZ}$, $t_{ER}$ | H → Z: Open<br>L → Z: Closed | 5 pF | 200 Ω | 390 Ω | 390 Ω | 750 Ω | H → Z: $V_{OH}$ – 0.5 V<br>L → Z: $V_{OL}$ + 0.5 V |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

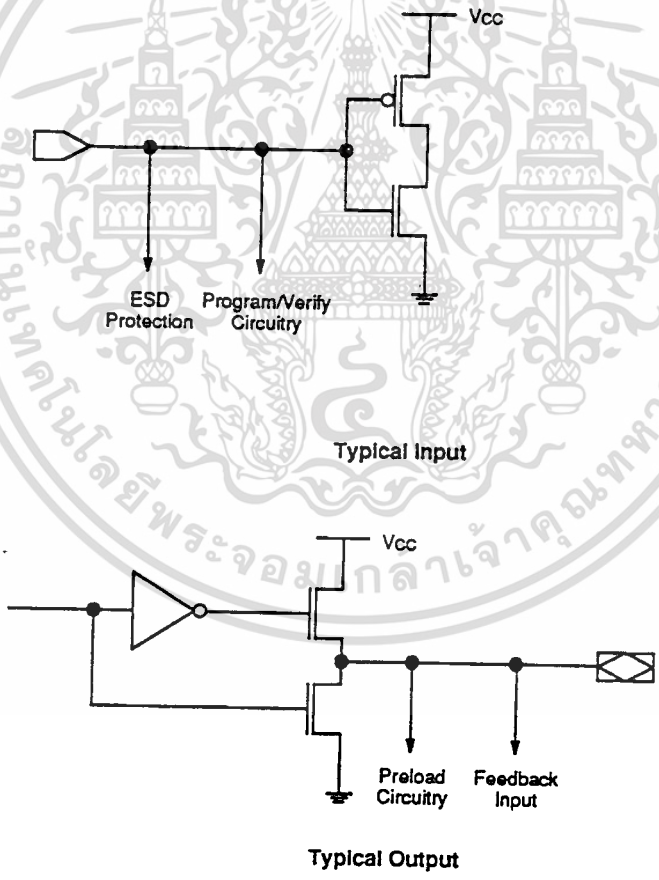
## ENDURANCE CHARACTERISTICS

The PALCE20V8 is manufactured using AMD's advanced electrically erasable process. This technology uses an EE cell to replace the fuse link used in bipolar parts. As a result, the device can be erased and reprogrammed – a feature which allows 100% testing at the factory.

### Endurance Characteristics

| Symbol | Parameter | Min. | Units | Test Conditions |
|---|---|---|---|---|
| $t_{DR}$ | Min. Pattern Data Retention Time | 10 | Years | Max. Storage Temperature |
| | | 20 | Years | Max. Operating Temperature (Military) |
| N | Min. Reprogramming Cycles | 100 | Cycles | Normal Programming Conditions |

## INPUT/OUTPUT EQUIVALENT SCHEMATICS




12197-013A



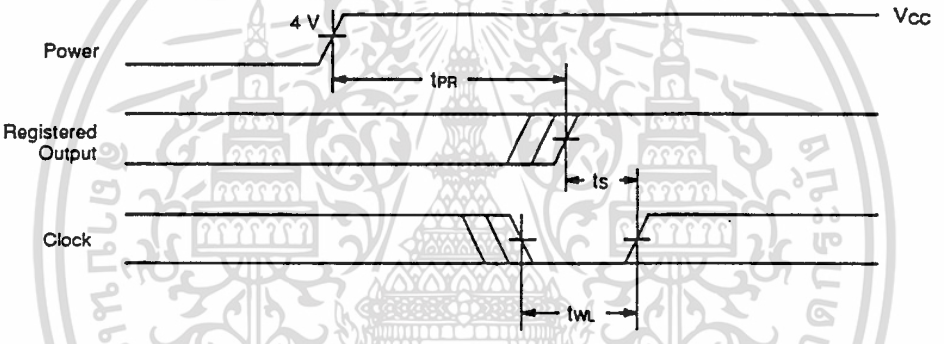
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## POWER-UP RESET

The PALCE20V8 has been designed with the capability to reset during system power-up. Following power-up, all flip-flops will be reset to LOW. The output state will be HIGH independent of the logic polarity. This feature provides extra flexibility to the designer and is especially valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways Vcc can rise to its steady state, two conditions are required to insure a valid power-up reset. These conditions are:

1. The Vcc rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

| Parameter Symbol | Parameter Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{PR}$ | Power-Up Reset Time | | 1000 | ns |
| $t_S$ | Input or Feedback Setup Time | See Switching Characteristics | | |
| $t_{WL}$ | Clock Width LOW | | | |




12197-009A

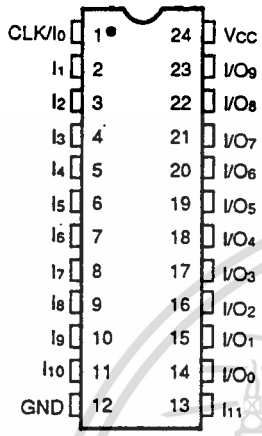
**Power-Up Reset Waveforms**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

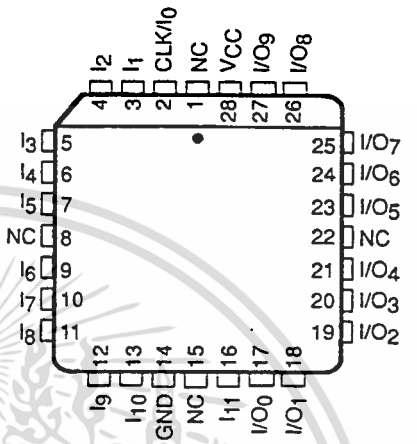
## CONNECTION DIAGRAMS
### Top View

**SKINNYDIP/SOIC/FLATPACK**




09849-002A

**PLCC/LCC**




09849-003A

**Note:**
Pin 1 is marked for orientation.

## PIN DESIGNATIONS

| | |
|---|---|
| CLK | Clock |
| GND | Ground |
| I | Input |
| I/O | Input/Output |
| NC | No Connect |
| Vcc | Supply Voltage |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM'L: H-10/15/25, Q-25 MIL: H-10/15/20/25/30

# PALCE22V10 Family

## 24-Pin EE CMOS Versatile PAL Device

Advanced Micro Devices

## DISTINCTIVE CHARACTERISTICS

- As fast as 10 ns propagation delay and 83.3 MHz $f_{MAX}$
- Low-power EE CMOS
- 10 macrocells programmable as registered or combinatorial, and active high or active low to match application needs
- Varied product term distribution allows up to 16 product terms per output for complex functions
- Global asynchronous reset and synchronous preset for initialization
- Power-up reset for initialization and register preload for testability
- Extensive third-party software and programmer support through FusionPLD partners
- 24-pin SKINNYDIP, 24-pin SOIC, 24-pin Flatpack and 28-pin PLCC and LCC packages save space

## GENERAL DESCRIPTION

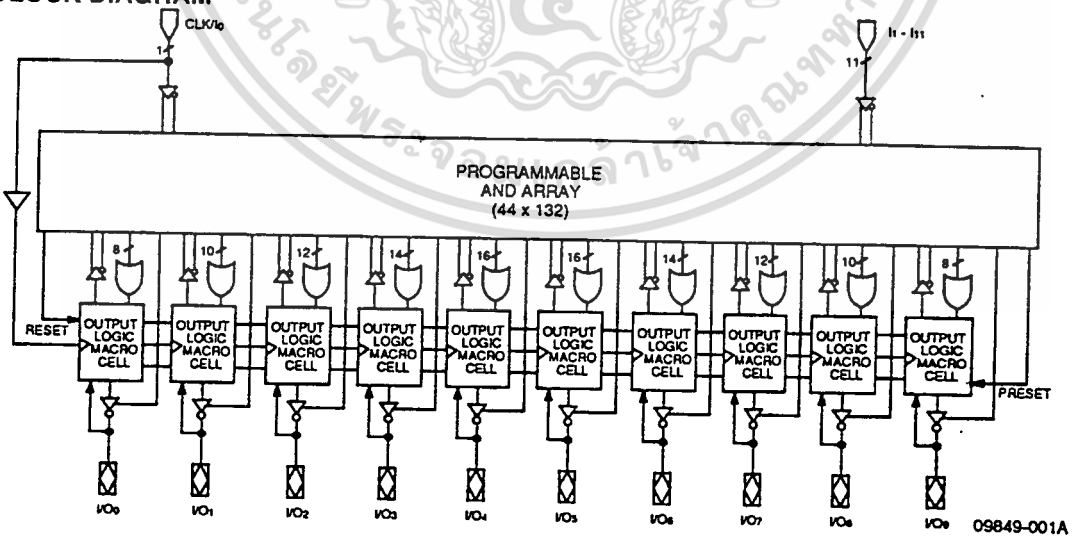
The PALCE22V10 provides user-programmable logic for replacing conventional SSI/MSI gates and flip-flops at a reduced chip count.

The PAL device implements the familiar Boolean logic transfer function, the sum of products. The PAL device is a programmable AND array driving a fixed OR array. The AND array is programmed to create custom product terms, while the OR array sums selected terms at the outputs.

The product terms are connected to the fixed OR array with a varied distribution from 8 to16 across the outputs (see Block Diagram). The OR sum of the products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial, and active high or active low. The output configuration is determined by two bits controlling two multiplexers in each macrocell.

AMD's FusionPLD program allows PALCE22V10 designs to be implemented using a wide variety of popular industry-standard design tools. By working closely with the FusionPLD partners, AMD certifies that the tools provide accurate, quality support. By ensuring that third-party tools are available, costs are lowered because a designer does not have to buy a complete set of new tools for each device. The FusionPLD program also greatly reduces design time since a designer can use a tool that is already installed and familiar. Please refer to the PLD Software Reference Guide for certified development systems and the Programmer Reference Guide for approved programmers.

## BLOCK DIAGRAM




Publication# 16564 Rev. A Amendment/0
Issue Date: January 1992

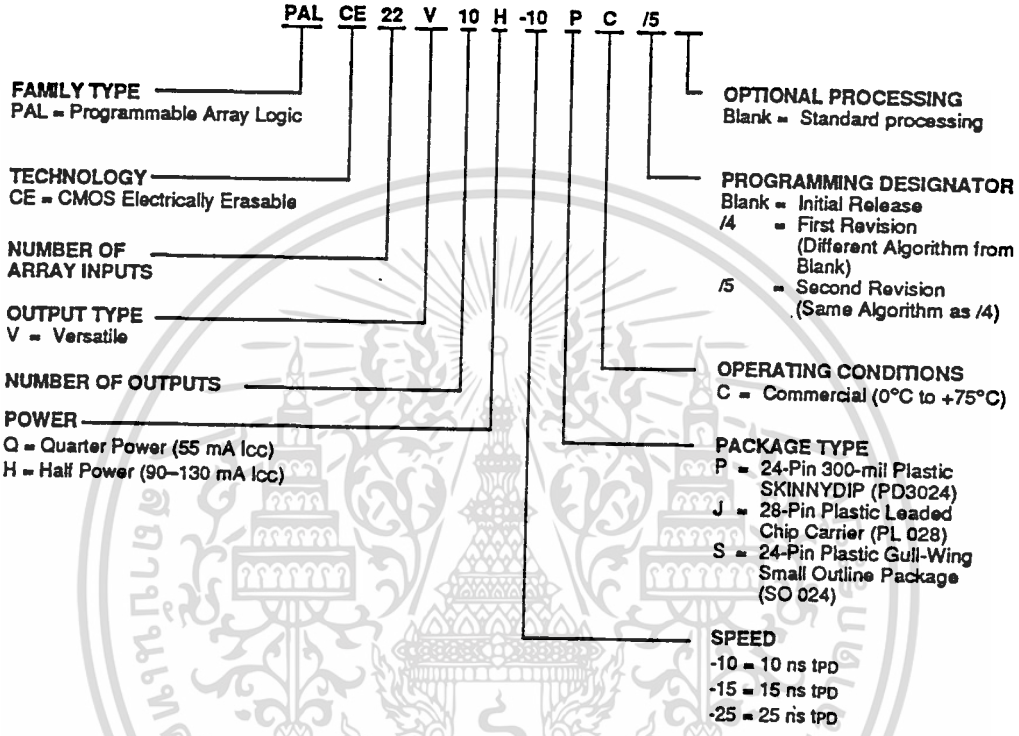
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ORDERING INFORMATION

### Commercial Products

AMD programmable logic products for commercial applications are available with several ordering options. The order number (Valid Combination) is formed by a combination of:




| Valid Combinations | | |
|---|---|---|
| PALCE22V10H-10 | PC, JC | /5 |
| PALCE22V10H-15 | | /4, /5 |
| PALCE22V10H-25 | PC, JC, SC | Blank, /4 |
| PALCE22V10Q-25 | | |
| PALCE22V10H-15 | | |
| PALCE22V10H-25 | | |

**Valid Combinations**

The Valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## FUNCTIONAL DESCRIPTION

The PALCE22V10 allows the systems engineer to implement the design on-chip, by programming EE cells to configure AND and OR gates within the device, according to the desired logic function. Complex interconnections between gates, which previously required time-consuming layout, are lifted from the PC board and placed on silicon, where they can be easily modified during prototyping or production.

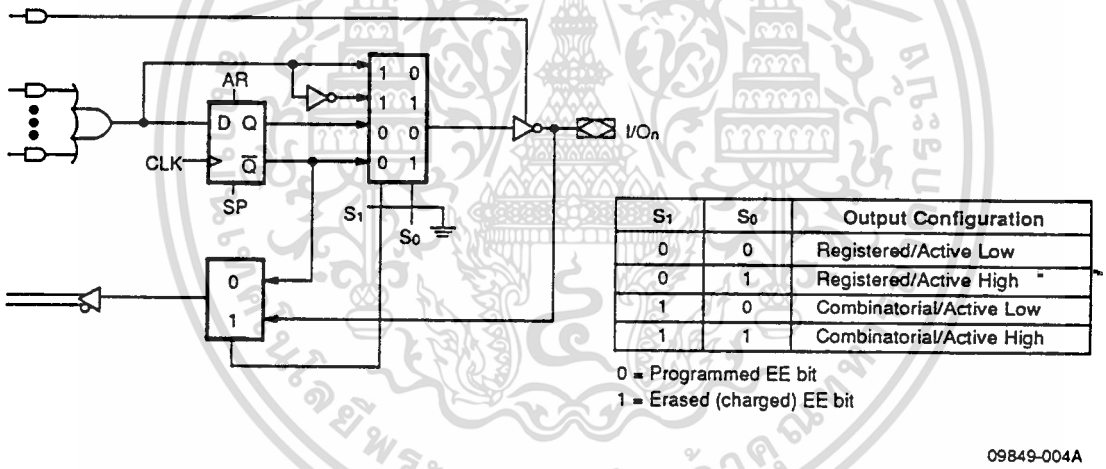
Product terms with all connections opened assume the logical HIGH state; product terms connected to both true and complement of any single input assume the logical LOW state.

The PALCE22V10 has 12 inputs and 10 I/O macrocells. The macrocell Figure 1allows one of four potential output configurations; registered output or combinatorial I/O, active high or active low (see Figure 1). The configuration choice is made according to the user's design specification and corresponding programming of the configuration bits $S_0$ - $S_1$. Multiplexer controls are connected to ground (0) through a programmable bit, selecting the "0" path through the multiplexer. Erasing the bit disconnects the control line from GND and it is driven to a high level, selecting the "1" path.

The device is produced with a EE cell link at each input to the AND gate array, and connections may be selectively removed by applying appropriate voltages to the circuit. Utilizing an easily-implemented programming algorithm, these products can be rapidly programmed to any customized pattern.

### Variable Input/Output Pin Ratio

The PALCE22V10 has twelve dedicated input lines, and each macrocell output can be an I/O pin. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Unused input pins should be tied to Vcc or GND.




| $S_1$ | $S_0$ | Output Configuration |
|---|---|---|
| 0 | 0 | Registered/Active Low |
| 0 | 1 | Registered/Active High |
| 1 | 0 | Combinatorial/Active Low |
| 1 | 1 | Combinatorial/Active High |

0 = Programmed EE bit
1 = Erased (charged) EE bit

09849-004A

**Figure 1. Output Logic Macrocell Diagram**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Registered Output Configuration

Each macrocell of the PALCE22V10 includes a D-type flip-flop for data storage and synchronization. The flip-flop is loaded on the LOW-to-HIGH transition of the clock input. In the registered configuration ($S_1 = 0$), the array feedback is from $\overline{Q}$ of the flip-flop.

## Combinatorial I/O Configuration

Any macrocell can be configured as combinatorial by selecting the multiplexer path that bypasses the flip-flop ($S_1 = 1$). In the combinatorial configuration the feedback is from the pin.

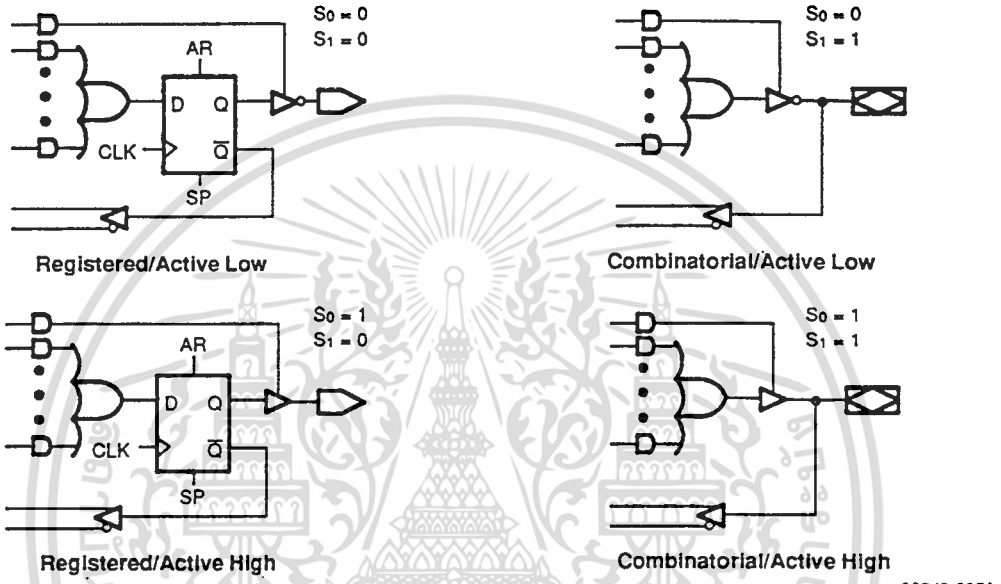



Figure 2. Macrocell Configuration Options



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Programmable Three-State Outputs

Each output has a three-state output buffer with three-state control. A product term controls the buffer, allowing enable and disable to be a function of any product of device inputs or output feedback. The combinatorial output provides a bidirectional I/O pin, and may be configured as a dedicated input if the buffer is always disabled.

## Programmable Output Polarity

The polarity of each macrocell output can be active high or active low, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is controlled by programmable bit $S_0$ in the output macrocell, and affects both registered and combinatorial outputs. Selection is automatic, based on the design specification and pin definitions. If the pin definition and output equation have the same polarity, the output is programmed to be active high ($S_0 = 1$).

## Preset/Reset

For initialization, the PALCE22V10 has Preset and Reset product terms. These terms are connected to all registered outputs. When the Synchronous Preset (SP) product term is asserted high, the output registers will be loaded with a HIGH on the next LOW-to-HIGH clock transition. When the Asynchronous Reset (AR) product term is asserted high, the output registers will be immediately loaded with a LOW independent of the clock.

Note that preset and reset control the flip-flop, not the output pin. The output level is determined by the output polarity selected.

## Power-Up Reset

All flip-flops power-up to a logic LOW for predictable system initialization. Outputs of the PALCE22V10 will depend on the programmed output polarity. The Vcc rise must be monotonic and the reset delay time is 1000 ns maximum.

## Register Preload

The register on the PALCE22V10 can be preloaded from the output pins to facilitate functional testing of complex state machine designs. This feature allows direct loading of arbitrary states, making it unnecessary to cycle through long test vector sequences to reach a desired state. In addition, transitions from illegal states can be verified by loading illegal states and observing proper recovery.

## Security Bit

After programming and verification, a PALCE22V10 design can be secured by programming the security EE bit. Once programmed, this bit defeats readback of the internal programmed pattern by a device programmer, securing proprietary designs from competitors. When the security bit is programmed, the array will read as if every bit is erased, and preload will be disabled.

The bit can only be erased in conjunction with erasure of the entire pattern.

## Programming and Erasing

The PALCE22V10 can be programmed on standard logic programmers. It also may be erased to reset a previously configured device back to its virgin state. Erasure is automatically performed by the programming hardware. No special erase operation is required.

## Quality and Testability

The PALCE22V10 offers a very high level of built-in quality. The erasability of the device provides a direct means of verifying performance of all AC and DC parameters. In addition, this verifies complete programmability and functionality of the device to provide the highest programming yields and post-programming functional yields in the industry.

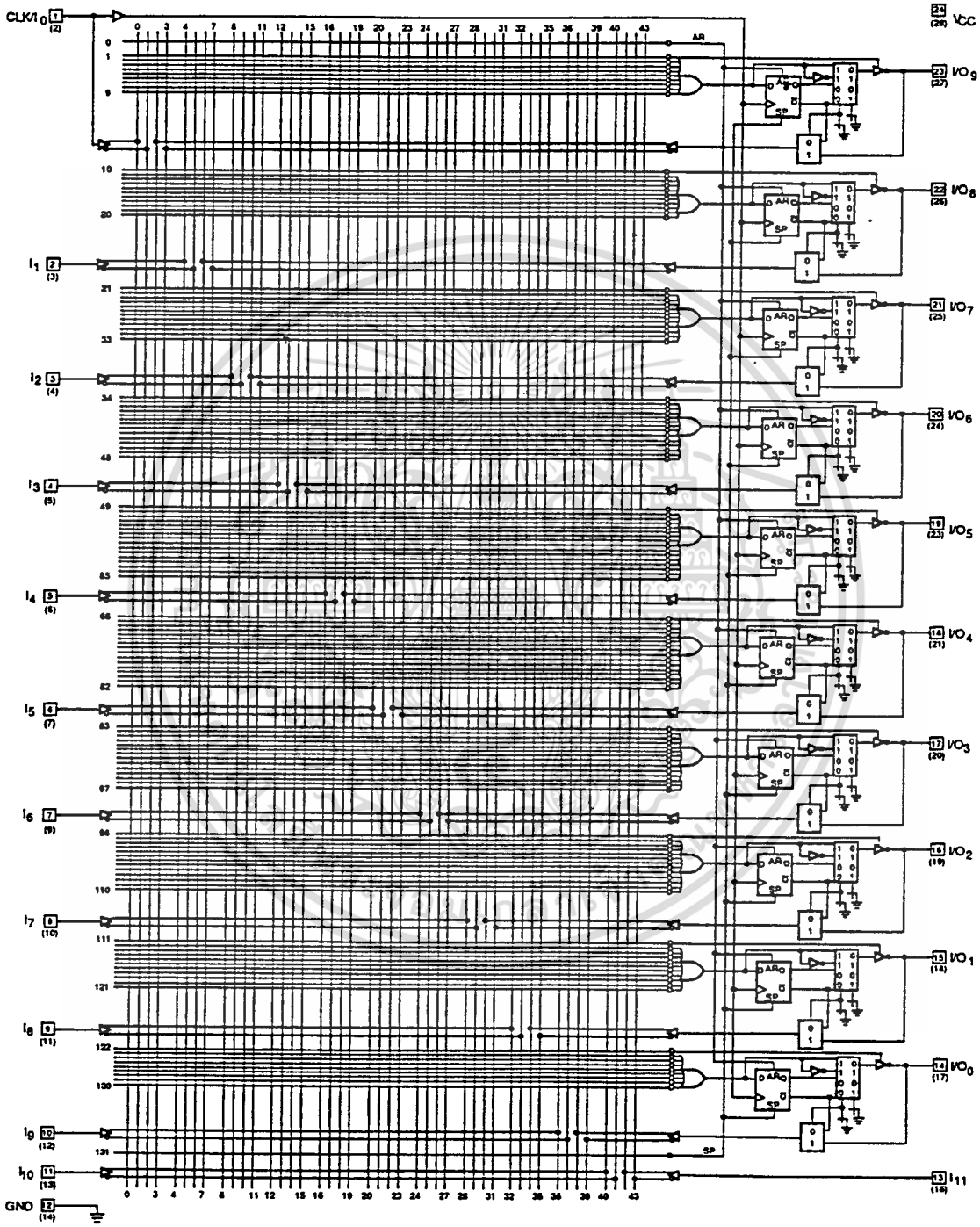
## Technology

The high-speed PALCE22V10 is fabricated with AMD's advanced electrically erasable (EE) CMOS process. The array connections are formed with proven EE cells. Inputs and outputs are designed to be compatible with TTL devices. This technology provides strong input clamp diodes, output slew-rate control, and a grounded substrate for clear switching.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## LOGIC DIAGRAM
## SKINNYDIP/SOIC/FLATPACK (PLCC/LCC) Pinouts




09849-006A



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMD

## ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Storage Temperature | –65°C to +150°C |
| Ambient Temperature with Power Applied | –55°C to +125°C |
| Supply Voltage with Respect to Ground | –0.5 V to +7.0 V |
| DC Input Voltage | –0.5 V to Vcc + 1.0 V |
| DC Output or I/O Pin Voltage | –0.5 V to Vcc + 0.5 V |
| Static Discharge Voltage | 2001 V |
| Latchup Current ($T_A$ = –55°C to +125°C) | 100 mA |

*Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability. Programming conditions may differ. Absolute Maximum Ratings are for system design reference; parameters given are not tested.*

## OPERATING RANGES

**Military (M) Devices (Note 1)**

| | |
|---|---|
| Operating Case Temperature (Tc) | –55°C to +125°C |
| Supply Voltage (Vcc) with Respect to Ground | +4.5 V to +5.5 V |

*Operating Ranges define those limits between which the functionality of the device is guaranteed.*

**Note:**

1. Military products are tested at Tc = +25°C, +125°C and –55°C, per MIL-STD-883.

## DC CHARACTERISTICS over MILITARY operating ranges unless otherwise specified (Note 2)

| Parameter Symbol | Parameter Description | Test Conditions | | Min. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{OH}$ | Output HIGH Voltage | $I_{OH}$ = –2.0 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | | 2.4 | | V |
| $V_{OL}$ | Output LOW Voltage | $I_{OL}$ = 12 mA $V_{IN}$ = $V_{IH}$ or $V_{IL}$ Vcc = Min. | | | 0.4 | V |
| $V_{IH}$ | Input HIGH Voltage | Guaranteed Input Logical HIGH Voltage for all Inputs (Note 3) | | 2.0 | | V |
| $V_{IL}$ | Input LOW Voltage | Guaranteed Input Logical LOW Voltage for all Inputs (Note 3) | | | 0.8 | V |
| $I_{IH}$ | Input HIGH Leakage Current | $V_{IN}$ = 5.5 V, Vcc = Max. (Note 4) | | | 10 | µA |
| $I_{IL}$ | Input LOW Leakage Current | $V_{IN}$ = 0 V, Vcc = Max. (Note 4) | | | –10 | µA |
| $I_{OZH}$ | Off-State Output Leakage Current HIGH | $V_{OUT}$ = 5.5 V, Vcc = Max., $V_{IN}$ = $V_{IL}$ or $V_{IH}$ (Note 4) | | | 10 | µA |
| $I_{OZL}$ | Off-State Output Leakage Current LOW | $V_{OUT}$ = 0 V, Vcc = Max., $V_{IN}$ = $V_{IH}$ or $V_{IL}$ (Note 4) | | | –10 | µA |
| Isc | Output Short-Circuit Current | $V_{OUT}$ = 0.5 V, Vcc = 5 V $T_A$ = 25°C (Note 5) | | –50 | –135 | mA |
| Icc | Supply Current | $V_{IN}$ = 0 V, Outputs Open ($I_{OUT}$ = 0 mA), Vcc = Max. | -15/-20 | | 120 | mA |
| | | | -25/-30 | | 100 | |

**Notes:**

2. For APL products, Group A, Subgroups 1, 2 and 3 are tested per MIL-STD-883, Method 5005, unless otherwise noted.
3. $V_{IL}$ and $V_{IH}$ are input conditions of output tests and are not themselves directly tested. $V_{IL}$ and $V_{IH}$ are absolute voltages with respect to device ground and include all overshoots due to system and/or tester noise. Do not attempt to test these values without suitable equipment.
4. I/O pin leakage is the worst case of $I_{IL}$ and $I_{OZL}$ (or $I_{IH}$ and $I_{OZH}$).
5. Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second. $V_{OUT}$ = 0.5 V has been chosen to avoid test problems caused by tester ground degradation. This parameter is not 100% tested, but is evaluated at initial characterization and at any time the design is modified where Isc may be affected.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CAPACITANCE (Note 1)

| Parameter Symbol | Parameter Description | Test Conditions | | Typ. | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance | $V_{IN}$ = 2.0 V | $V_{CC}$ = 5.0 V, $T_A$ = 25°C, f = 1 MHz | 8 | pF |
| $C_{OUT}$ | Output Capacitance | $V_{OUT}$ = 2.0 V | | 9 | |

**Note:**

1. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

## SWITCHING CHARACTERISTICS over MILITARY operating ranges (Note 2)

| Parameter Symbol | Parameter Description | | -15 | | -20 | | -25 | | -30 | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Max. | Min. | Max. | Min. | Max. | Min. | Max. | |
| $t_{PD}$ | Input or Feedback to Combinatorial Output | | | 15 | | 20 | | 25 | | 30 | ns |
| $t_S$ | Setup Time from Input, Feedback or SP to Clock | | 12 | | 15 | | 18 | | 20 | | ns |
| $t_H$ | Hold Time (Note 3) | | 0 | | 0 | | 0 | | 0 | | ns |
| $t_{CO}$ | Clock to Output | | | 8 | | 15 | | 20 | | 20 | ns |
| $t_{AR}$ | Asynchronous Reset to Registered Output | | | 20 | | 25 | | 25 | | 30 | ns |
| $t_{ARW}$ | Asynchronous Reset Width (Note 3) | | 15 | | 20 | | 25 | | 30 | | ns |
| $t_{ARR}$ | Asynchronous Reset Recovery Time (Note 3) | | 15 | | 20 | | 25 | | 30 | | ns |
| $t_{SPR}$ | Synchronous Preset Recovery Time | | 15 | | 20 | | 25 | | 30 | | ns |
| $t_{WL}$ | Clock Width | LOW | 8 | | 10 | | 15 | | 15 | | ns |
| $t_{WH}$ | | HIGH | 8 | | 10 | | 15 | | 15 | | ns |
| $f_{MAX}$ | Maximum Frequency (Note 3) | External Feedback 1/(ts + tco) | 50 | | 33.3 | | 26.3 | | 25 | | MHz |
| | | Internal Feedback ($f_{CNT}$) | 53 | | 40 | | 32.2 | | 25 | | MHz |
| $t_{EA}$ | Input to Output Enable Using Product Term Control (Note 3) | | | 15 | | 20 | | 25 | | 25 | ns |
| $t_{ER}$ | Input to Output Disable Using Product Term Control (Note 3) | | | 15 | | 20 | | 25 | | 25 | ns |

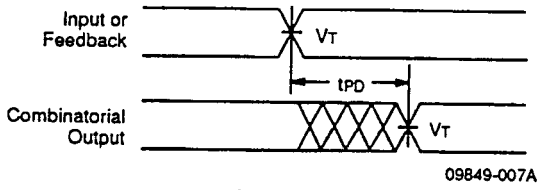
**Notes:**

2. See Switching Test Circuit for test conditions. For APL products Group A, Subgroups 7, 8, 9, 10, and 11 are tested per MIL-STD-883, Method 5005, unless otherwise noted.
3. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where these parameters may be affected.

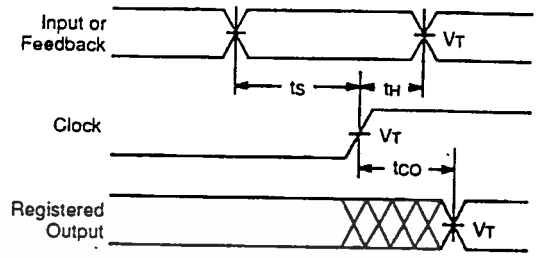


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

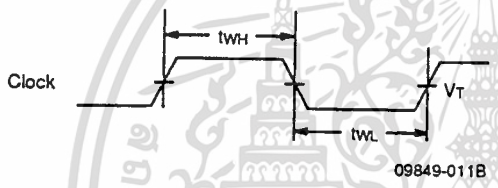
## SWITCHING WAVEFORMS



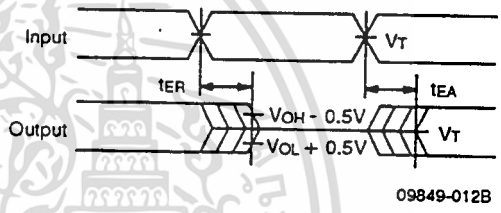

**Combinatorial Output**



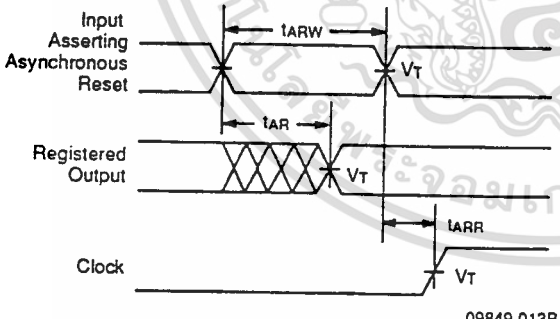

**Registered Output**



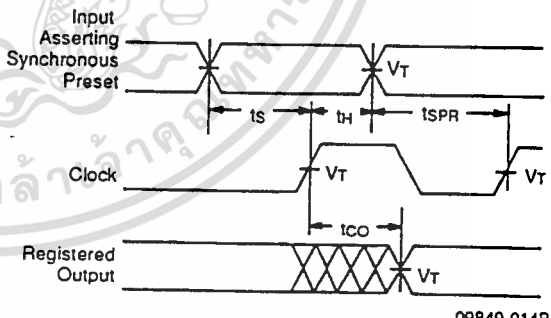

**Clock Width**




**Input to Output Disable/Enable**




**Asynchronous Reset**




**Synchronous Preset**

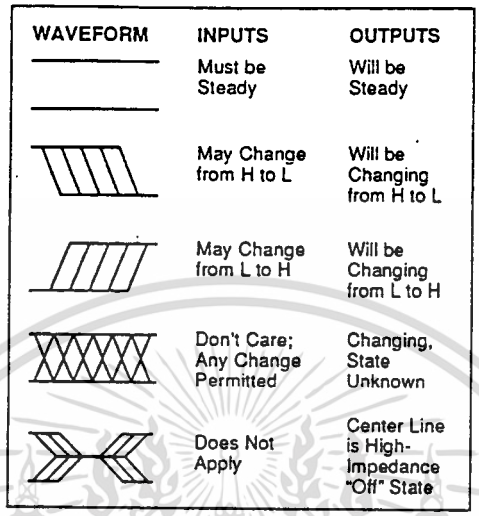
**Notes:**

1. $V_T$ = 1.5 V.
2. Input pulse amplitude 0 V to 3.0 V.
3. Input rise and fall times 2–5 ns typical.



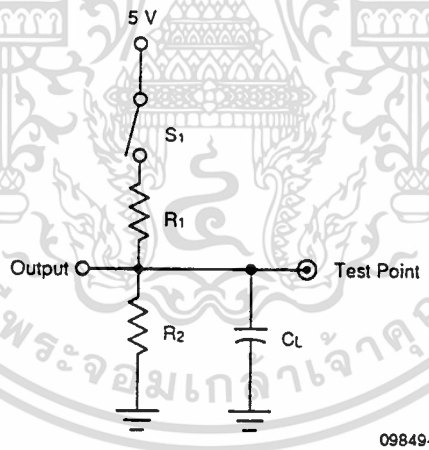
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## KEY TO SWITCHING WAVEFORMS




KS000010-PAL

## SWITCHING TEST CIRCUIT




09849-015A

| Specification | $S_1$ | $C_L$ | Commercial | | Military | | Measured Output Value |
|---|---|---|---|---|---|---|---|
| | | | $R_1$ | $R_2$ | $R_1$ | $R_2$ | |
| $t_{PD}$, $t_{CO}$ | Closed | 50 pF | 300 Ω | 390 Ω | 390 Ω | 750 Ω | 1.5 V |
| $t_{EA}$ | Z → H: Open<br>Z → L: Closed | | | | | | 1.5 V |
| $t_{ER}$ | H → Z: Open<br>L → Z: Closed | 5 pF | | | | | H → Z: $V_{OH}$ – 0.5 V<br>L → Z: $V_{OL}$ + 0.5 V |



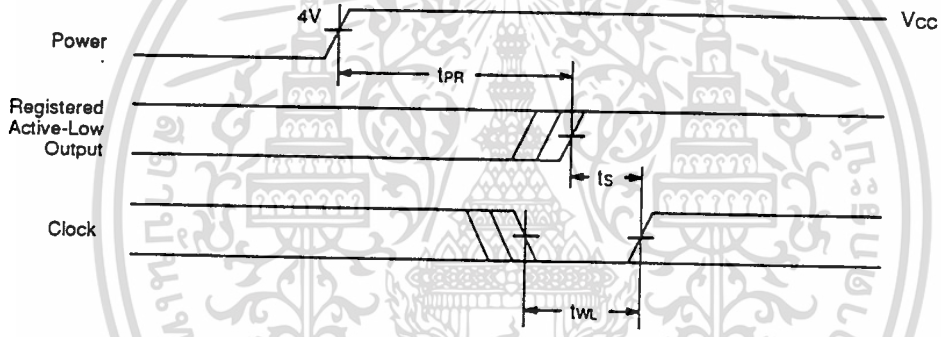
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways Vcc can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The Vcc rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

| Parameter Symbol | Parameter Description | Max. | Unit |
|---|---|---|---|
| $t_{PR}$ | Power-up Reset Time | 1000 | ns |
| $t_S$ | Input or Feedback Setup Time | See Switching Characteristics | |
| $t_{WL}$ | Clock Width LOW | | |




Power-Up Reset Waveform



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

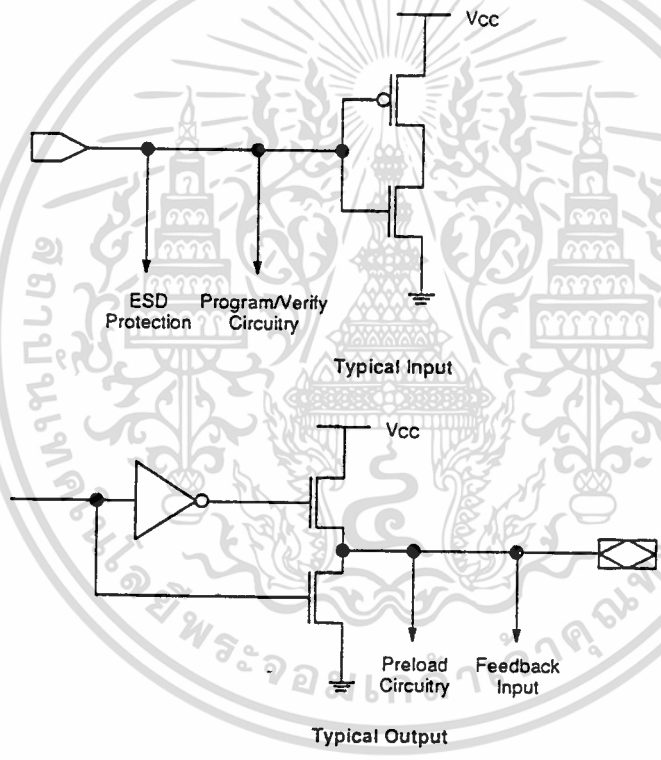
## ENDURANCE CHARACTERISTICS

The PALCE22V10 is manufactured using AMD's advanced Electrically Erasable process. This technology uses an EE cell to replace the fuse link used in bipolar parts. As a result, the device can be erased and reprogrammed – a feature which allows 100% testing at the factory.

### Endurance Characteristics

| Symbol | Parameter | Min. | Units | Test Conditions |
|---|---|---|---|---|
| $t_{DR}$ | Min. Pattern Data Retention Time | 10 | Years | Max. Storage Temperature |
| | | 20 | Years | Max. Operating Temperature (Military) |
| N | Min. Reprogramming Cycles | 100 | Cycles | Normal Programming Conditions |

## INPUT/OUTPUT EQUIVALENT SCHEMATICS




09849-017A

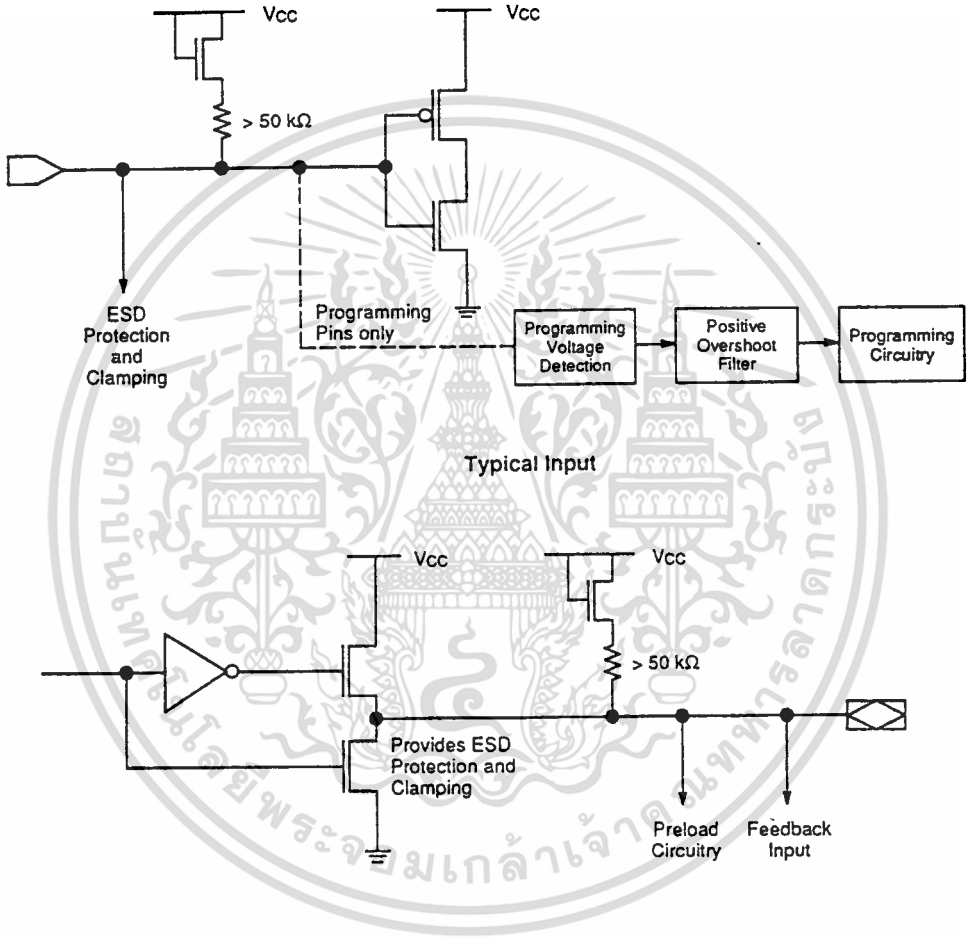


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ROBUSTNESS FEATURES

The PALCE22V10H-10/5 has some unique features that make it extremely robust, especially when operating in high-speed design environments. Pull-up resistors on inputs and I/O pins cause unconnected pins to default to a known state. Input clamping circuitry limits negative overshoot, eliminating the possibility of false clocking caused by subsequent ringing. A special noise filter makes the programming circuitry completely insensitive to any positive overshoot that has a pulse width of less than about 100 ns.

## INPUT/OUTPUT EQUIVALENT SCHEMATICS FOR /5 VERSION




16407A-001B



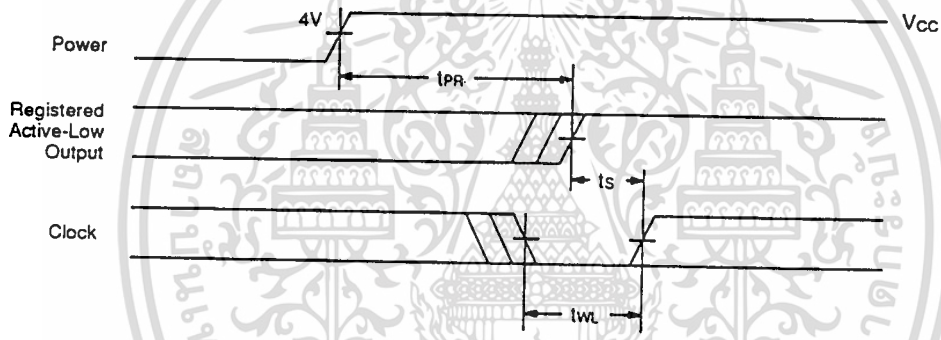
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## POWER-UP RESET

The power-up reset feature ensures that all flip-flops will be reset to LOW after the device has been powered up. The output state will depend on the programmed pattern. This feature is valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below. Due to the synchronous operation of the power-up reset and the wide range of ways Vcc can rise to its steady state, two conditions are required to ensure a valid power-up reset. These conditions are:

1. The Vcc rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

| Parameter Symbol | Parameter Description | Max. | Unit |
|---|---|---|---|
| $t_{PR}$ | Power-up Reset Time | 1000 | ns |
| $t_S$ | Input or Feedback Setup Time | See Switching Characteristics | |
| $t_{WL}$ | Clock Width LOW | | |




Power-Up Reset Waveform



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายการคำสั่งชุดส่งข้อมูล

## ( DATA TRANSMITTER PROGRAM )

```
        ORG    0000H

P07     EQU    87H    ; SW. 8/12 BIT
P24     EQU    0A4H   ; LED FOR CHECK DATA
LEDCHK  EQU    00H    ; (BIT) SAVE STATUS LED CHECK

;################  MAIN  PROGRAM  ##################

MAIN:   MOV    R0,#00H       ; WAIT FOR ANOTHER DEVICE
DEL2:   DJNZ   R0,DEL2

; %%%%%%%%%%%%%%  Control Serial Port  %%%%%%%%%%%%%%%

        MOV    TMOD,#00100000B
        MOV    SCON,#01000000B
        MOV    TCON,#01000000B
        MOV    PCON,#10000000B
        MOV    TH1,#243       ; BAUDARATE DIVISOR

        MOV    A,#0F0H        ; INITIAL LED OFF ALL
        MOV    R7,A           ; R7 SAVE LED DATA
        MOV    P0,A
        CLR    P24            ; CLEAR LED PORT2.4
        CLR    LEDCHK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP:  LCALL  CHKUPDN       ; CHECK UP-DOWN SW.
       AJMP   LOOP

; #######  CHECK UP-DOWN AND ENTER SW. ######

CHKUPDN:
       MOV    A,R7
       ANL    A,#0FH
       ORL    A,#0F0H
       MOV    P0,A          ; CLEAR PORT FOR INPUT
       MOV    A,P0          ; INPUT STATUS SW.
       ANL    A,#70H        ; MASK BIT
       CJNE   A,#70H,CHKENT  ; IF SW IS PUSH JUMP
       MOV    R2,#01H
       LCALL  DELAY
       AJMP   NOTPUSH

CHKENT: CJNE   A,#30H,CHKUP   ; IF ENTER NOT PUSH JUMP
       SETB   P07
       JB     P07,READ12     ; CHECK SW. 8/12 BIT
       MOV    A,#0FFH
       MOV    P1,A
       MOV    A,P1           ; INPUT DATA
       MOV    R5,A           ; SAVE DIP SW. DATA

       MOV    A,#00H         ; DATA SEND 8 BIT
       LCALL  SEND_DATA
       MOV    A,R5
       LCALL  SEND_DATA      ; SEND DATA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   WAIT_DATA      ; WAIT FOR RETURN DATA
        AJMP    CHECK_DATA

READ12: MOV     A,#0FFH
        MOV     P1,A
        MOV     A,P1           ; INPUT DATA BIT 1-8
        MOV     R5,A           ; STORE DATA

        MOV     A,#0FFH
        MOV     P2,A
        MOV     A,P2           ; INPUT DATA BIT 9-12
        ANL     A,#0FH
        MOV     R6,A           ; STORE DATA

        MOV     A,#0FFH        ; SEND DATA FOR SHOW 12 BIT
        LCALL   SEND_DATA
        MOV     A,R5
        LCALL   SEND_DATA      ; SEND BIT 1-8
        MOV     A,R6
        LCALL   SEND_DATA      ; SEND BIT 9-12
        LCALL   WAIT_DATA      ; WAIT FOR RETURN DATA

CHECK_DATA:
        CJNE    A,#00,FALSE    ; CHECK DATA THAT CORECT
        SETB    P24            ; }
        MOV     R1,#1          ; }
        LCALL   SEC            ; }
        CLR     P24            ; }
        MOV     R1,#1          ; }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  SEC        ;  }
        SETB   P24        ;  }
        MOV    R1,#1      ;  }
        LCALL  SEC        ;  }  DISPLAY LED FLASH
        CLR    P24        ;  }  FOR CORECT DATA
        MOV    R1,#1      ;  }
        LCALL  SEC        ;  }
        SETB   P24        ;  }
        MOV    R1,#1      ;  }
        LCALL  SEC        ;  }
        CLR    P24        ;  }
        MOV    R1,#1      ;  }
        LCALL  SEC        ;  }
        CLR    LEDCHK
        AJMP   NOTPUSH

FALSE:  SETB   P24        ; DISPLAY INCORECT DATA
        SETB   LEDCHK
        AJMP   NOTPUSH


CHKUP:  CJNE   A,#60H,CHKDN   ; IF UP NOT PUSH JUMP
        MOV    A,R7
        INC    A          ; INC LED DATA
        CJNE   A,#10H,UP2DN   ; IF OVERFLOW SET TO 00
        MOV    A,#00H
UP2DN:  ORL    A,#0F0H
        MOV    P0,A       ; OUT DATA TO LED
        MOV    R7,A       ; SAVE LED DATA
        MOV    R2,#3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL  DELAY
        AJMP   NOTPUSH

CHKDN:  CJNE   A,#50H,NOTPUSH  ; IF DOWN NOT PUSH JUMP
        MOV    A,R7
        DEC    A               ; DEC LED DATA
        CJNE   A,#0FFH,DN2UP   ; IF OVERFLOW SET TO 0FH
        MOV    A,#0FH
DN2UP:  ORL    A,#0F0H
        MOV    P0,A            ; OUT DATA TO LED
        MOV    R7,A            ; SAVE LED DATA
        MOV    R2,#3
        LCALL  DELAY

NOTPUSH:
        RET

SEND_DATA:
        CLR    REN
        MOV    SBUF,A
RESM2:  JNB    TI,RESM2
        CLR    TI
        RET

WAIT_DATA:
        SETB   REN
WAIT:   JNB    RI,WAIT
        CLR    RI
        MOV    A,SBUF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET

;---------- DELAY SUB ----------
; DELAY 1/10 SECOND
; IN  = R2
; REG = R2,R3,R4

DELAY:  MOV    R3,#179
DELAY1: MOV    R4,#0
        DJNZ   R4,$
        NOP
        NOP
        DJNZ   R3,DELAY1
        DJNZ   R2,DELAY
        RET

;---------- DELAY SUB ----------
; DELAY R1 SECOND
; IN  = R1
; REG = R1,R2,R3,R4

SEC:    MOV    R2,#10
        LCALL  DELAY
        DJNZ   R1,SEC
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## รายการคำสั่งชุดรับข้อมูล

## ( DATA RECEIVER PROGRAM )

```
        ORG    0000H

MAIN:   MOV    R0,#00H        ; WAIT FOR ANOTHER DEVICE
DEL2:   DJNZ   R0,DEL2

; %%%%%%%%  Control Serial Port  %%%%%%%%

        MOV    TMOD,#00100000B
        MOV    SCON,#01010000B
        MOV    TCON,#01000000B
        MOV    PCON,#10000000B
        MOV    TH1,#243       ; BAUDARATE DIVISOR
        SETB   TR1
        MOV    A,SBUF


        MOV    A,#00H         ; INITIAL LED
        MOV    P0,A
        MOV    P2,A

LOOP:   LCALL  READ_DATA      ; CHECK 8/12 BIT
        CJNE   A,#00H,SYS12   ; IF <> 0 => 12 BIT
        LCALL  READ_DATA      ; READ DATA BIT 1-8
        MOV    R5,A
        MOV    P0,A           ; OUT DATA BIT 1-8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    A,#00H
        MOV    P2,A          ; OUT DATA BIT 9-12
        LCALL  SEND_DATA     ; SEND DATA FOR END
        AJMP   LOOP

SYS12:  LCALL  READ_DATA     ; READ DATA BIT 1-8
        MOV    R5,A
        MOV    A,#00H
        LCALL  READ_DATA     ; READ DATA BIT 9-12
        MOV    R6,A

        MOV    P0,R5         ; OUT DATA BIT 1-8
        MOV    P2,R6         ; OUT DATA BIT 9-12
        MOV    A,#00H
        LCALL  SEND_DATA     ; SEND DATA FOR END
        AJMP   LOOP

READ_DATA:
        SETB   REN
WAIT:   JNB    RI,WAIT
        CLR    RI
        MOV    A,SBUF
        RET

SEND_DATA:
        CLR    REN
        MOV    SBUF,A
RESM2:  JNB    TI,RESM2
        CLR    TI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB   REN
        RET

;---------- DELAY SUB ----------
; DELAY 1/10 SECOND
; IN  = R2
; REG = R2,R3,R4

DELAY:  MOV    R3,#179
DELAY1: MOV    R4,#0
        DJNZ   R4,$
        NOP
        NOP
        DJNZ   R3,DELAY1
        DJNZ   R2,DELAY
        RET

;---------- DELAY SUB ----------
; DELAY R1 SECOND
; IN  = R1
; REG = R1,R2,R3,R4

SEC:    MOV    R2,#10
        LCALL  DELAY
        DJNZ   R1,SEC
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายการคำสั่งชุดส่งข้อมูล

## ( DATA TRANSMITTER PROGRAM )

**SAMPLES OF BASIC GATES**

## 1. DATA1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~5]),io:Y[1~4]
|
|   Y1 = A1 & A5'
|   Y2 = A2 & A5'
|   Y3 = A3 & A5'
|   Y4 = A4 & A5'
```

**SAMPLES OF BASIC GATES**

## 2. EN1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~6]),io:Y[1~4]
|
|   Y1 =(A1 ## A2 ## A4) & A5 & A6'
|   Y2 =(A1 ## A2 ## A3) & A5 & A6'
|   Y3 =(A1 ## A3 ## A4) & A5 & A6'
|   Y4 =(A2 ## A3 ## A4) & A5 & A6'
```

**SAMPLES OF BASIC GATES**

## 3. EN2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce22V10  in:(A[1~8]),io:Y[1~6]
|
|   Y1 =( A1 ## A3 ## A4 ## A5) & A7'& A8'
|   Y2 =( A2 ## A3)&A7'& A8'
|   Y3 =( A1 ## A3 ## A4 ## A6) & A7'& A8'
|   Y4 =( A2 ## A4) & A7'& A8'
|   Y5 =( A1 ## A2 ## A3 ## A5) & A7'& A8'
|   Y6 =( A1 ## A2 ## A4 ## A6) & A7'& A8'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SAMPLES OF BASIC GATES**

## 4. EN3

Here are basic logic gates written as Boolean equations. The notation A[1~6]is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~10]),io:Y[1~4]
|
|   Y1 = ( A1 & A9' & A10' ) # A5
|   Y2 = ( A2 & A9' & A10' ) # A6
|   Y3 = A3 # A7
|   Y4 = A4 # A8
```

**SAMPLES OF BASIC GATES**

## 5. CW1

Here are basic logic gates written as Boolean equations. The notation A[1~6]is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~6]
|
|   Y1 = A1 # ( A7 & A13 )
|   Y2 = A2 # ( A8 & A13 )
|   Y3 = A3 # ( A9 & A13 )
|   Y4 = A4 # ( A10 & A13)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Y5 = A5 # ( A11 & A13)

| Y6 = A6 # ( A12 & A13)

**SAMPLES OF BASIC GATES**

## 6. CW2

Here are basic logic gates written as Boolean equations. The notation A[1~6]is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

|PALce20V8 in:(A[1~14]),io:Y[1~6]

|

| Y1 = A1 # ( A7 & A13 )

| Y2 = A2 # ( A8 & A13 )

| Y3 = A3 # ( A9 & A13 & A14')

| Y4 = A4 # ( A10 & A13& A14')

| Y5 = A5 # ( A11 & A13& A14')

| Y6 = A6 # ( A12 & A13& A14')

**SAMPLES OF BASIC GATES**

## 7. NOI1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 ## A5
|   Y2 = A2 ## A6
|   Y3 = A3 ## A7
|   Y4 = A4 ## A8
```

**SAMPLES OF BASIC GATES**

## 8. NOI2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 ## A5
|   Y2 = A2 ## A6
|   Y3 = A3 ## A7
|   Y4 = A4 ## A8
```

**SAMPLES OF BASIC GATES**

## 9. NOI3

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 ## A5
|   Y2 = A2 ## A6
|   Y3 = A3 ## A7
|   Y4 = A4 ## A8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# รายการคำสั่งชุดรับข้อมูล

## ( DATA RECEIVER PROGRAM )

**SAMPLES OF BASIC GATES**

### 1. SYN1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce22V10  in:(A[1~10]),io:Y[1~4]
|
|   Y1 =(A1 ## A2 ## A4 ## A5) & A9 & A10'
|   Y2 =(A1 ## A2 ## A3 ## A6) & A9 & A10'
|   Y3 =(A1 ## A3 ## A4 ## A7) & A9 & A10'
|   Y4 =(A2 ## A3 ## A4 ## A8) & A9 & A10'
```

**SAMPLES OF BASIC GATES**

### 2. SYN2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~8]),io:Y[1~6]
|
|   Y1 = (A1 ## A3 ## A4) & A8'
|   Y2 = (A2 ## A3 ## A5) & A7'& A8'
|   Y3 = (A1 ## A3 ## A4) & A8'
|   Y4 = (A2 ## A4 ## A6) & A7'& A8'
|   Y5 = (A1 ## A2 ## A3) & A8'
|   Y6 = (A1 ## A2 ## A4) & A8'
```

**SAMPLES OF BASIC GATES**

## 3. SYN3

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~4]
|
|   Y1 = (A1 ## (A5 # A11) ## A7)& A12'& A13'
|   Y2 = (A2 ## (A6 # A11) ## A8)& A12'& A13'
|   Y3 = (A3 ## (A5 # A11) ## A9)& A12'& A13'
|   Y4 = (A4 ## (A6 # A11) ## A10)&A12'& A13'
```

**SAMPLES OF BASIC GATES**

## 4. TOSYN

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

six", and represents the signals A1, A2, A3, A4, A5,
and A6.

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 # A5
|   Y2 = A2 # A6
|   Y3 = A3 # A7
|   Y4 = A4 # A8
```

**SAMPLES OF BASIC GATES**

## 5. DET1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce22V10  in:(A[1~4]),io:Y[1~3]
|
|   Y1 = ( A1 # A2 # A3 # A4 )'
|   Y2 = ( A1 ## A2 ## A3 ## A4 )'& ( A1 # A2 # A3 # A4 )
|   Y3 = ( A1 ## A2 ## A3 ## A4 ) & ( A1 # A2 # A3 # A4 )
```

**SAMPLES OF BASIC GATES**

## 6. DET2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

six", and represents the signals A1, A2, A3, A4, A5,
and A6.

```
|PALce22V10  in:(A[1~6]),io:Y[1~10]
|
|   Y1 = A1 & A2' & A3
|   Y2 = A4' & A5 & A6
|   Y3 = A1' & A2 & A3'
|   Y4 = A4 & A5 & A6
|   Y5 = A1 & A2 & A3
|   Y6 = A4' & A5 & A6'
|   Y7 = A4 & A5' & A6
|   Y8 = A1 & A2' & A3'
|   Y9 = A4' & A5' & A6'
|   Y10= A1' & A2' & A3
```

**SAMPLES OF BASIC GATES**

## 7. DET3

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~6]),io:Y[1~6]
|
|   Y1 = A1' & A2' & A3'
|   Y2 = A4 & A5' & A6'
|   Y3 = A4' & A5' & A6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Y4 = A4 & A5 & A6'

| Y5 = A1 & A2 & A3'

| Y6 = A1' & A2 & A3

**SAMPLES OF BASIC GATES**

## 8. ERR1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

|PALce20V8 in:(A[1~14]),io:Y[1~5]

|

| Y1 = (A1 & A2) # (A5 & A10)# (A3 & A11)# (A9 & A12)

| Y2 = (A2 & A8) # (A2 & A5) # (A2 & A7) # (A1 & A4)

| Y3 = (A1 & A11)# (A1 & A6) # (A10& A14)# (A10& A13)

| Y4 = (A11& A13)# (A11& A14)# (A7 & A12)# (A8 & A12)# (A5 & A14)

| Y5 = (A2 & A14)# (A4 & A8) # (A2 & A13)# (A4 & A5) # (A4 & A7)

**SAMPLES OF BASIC GATES**

## 9. ERR2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce20V8  in:(A[1~14]),io:Y[1~5]
|
|   Y1 = (A2 & A3) # (A4 & A7) # (A3 & A9) # (A3 & A6)
|   Y2 = (A3 & A10)# (A1 & A2) # (A2 & A5) # (A2 & A8)
|   Y3 = (A7 & A9) # (A5 & A9) # (A8 & A9) #(A5 & A10)
|   Y4 = (A8 & A10)# (A7 & A10)
|   Y5 = A11 # A12 # A13 # A14
```

**SAMPLES OF BASIC GATES**

## 10. ERR3

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~10]),io:Y[1~3]
|
|   Y1 = (A2 & A3) # (A1 & A5) # (A3 & A8) # (A3 & A7)
|   Y2 = (A2 & A6) # (A4 & A2) # (A7 & A5) # (A5 & A8)
|   Y3 = A9 # A10
```

**SAMPLES OF BASIC GATES**

## 11. ERR4

`Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~10]),io:Y[1~3]
|
|   Y1 = (A1 & A3) # (A6 & A8) # (A3 & A5) # (A2 & A3)
|   Y2 = (A3 & A4) # (A1 & A7) # (A4 & A8) # (A5 & A8)
|   Y3 = A9 # A10
```

**SAMPLES OF BASIC GATES**

## 12. ERR5

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~14]),io:Y[1~5]
|
|   Y1 = (A1 & A2) # (A5 & A10)# (A3 & A13)# (A9 & A12)
|   Y2 = (A2 & A8) # (A2 & A5) # (A2 & A7) # (A1 & A4)
|   Y3 = (A1 & A11)# (A1 & A6) # (A10& A14)# (A10& A13)
|   Y4 = (A11& A13)# (A11& A14)# (A7 & A12)# (A8 & A12)# (A5 & A11)
|   Y5 = (A2 & A14)# (A4 & A8) # (A2 & A13)# (A4 & A5) # (A4 & A7)
```

**SAMPLES OF BASIC GATES**

## 13. ERR6

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce20V8  in:(A[1~13]),io:Y[1~4]
|
|   Y1 = (A6 & A7) # (A2 & A8) # (A3 & A12)# (A4 & A13)
|   Y2 = (A5 & A8) # (A7 & A12)# (A1 & A7) # (A6 & A9)
|   Y3 = (A4 & A6) # (A6 & A10)# (A9 & A13)# (A10 & A12)
|   Y4 = (A6 & A11)# (A2 & A13)# (A3 & A8 )# (A5 & A12)
        # (A11 & A12)
```

**SAMPLES OF BASIC GATES**

## 14. ERR7

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~4]
|
|   Y1 = (A6 & A7) # (A2 & A8) # (A3 & A12)# (A4 & A13)
|   Y2 = (A5 & A8) # (A7 & A12)# (A1 & A7) # (A6 & A9)
|   Y3 = (A4 & A6) # (A6 & A10)# (A9 & A13)# (A10 & A12)
|   Y4 = (A6 & A11)# (A2 & A13)# (A3 & A8 )# (A5 & A12) # (A11 & A12)
```

**SAMPLES OF BASIC GATES**

## 15. ERR8

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

six", and represents the signals A1, A2, A3, A4, A5,
and A6.

```
|PALce20V8  in:(A[1~12]),io:Y[1~3]
|
|   Y1 = (A2 & A7) # (A1 & A4) # (A3 & A8) # (A4 & A6)
|   Y2 = (A7 & A11)# (A7 & A12)# (A2 & A9)# (A2 & A5)
|   Y3 = (A1 & A12)# (A1 & A11)# (A3 & A4) # (A4 & A10)
```

**SAMPLES OF BASIC GATES**

## 16. ERR9

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~4]
|
|   Y1 = (A7 & A8) # (A2 & A6) # (A3 & A13)# (A4 & A12)
|   Y2 = (A5 & A6) # (A7 & A1) # (A13 & A7)# (A8 & A9)
|   Y3 = (A4 & A8) # (A8 & A10)# (A9 & A12)# (A10 & A13)
|   Y4 = (A8 & A11)# (A2 & A12)# (A3 & A6 )# (A5 & A13) # (A11 & A13)
```

**SAMPLES OF BASIC GATES**

## 17. ERR 10

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

six", and represents the signals A1, A2, A3, A4, A5,
and A6.

```
|PALce20V8  in:(A[1~14]),io:Y[1~4]
|
|   Y1 = (A9 & A10)# (A1 & A4) # (A3 & A2) # (A5 & A11)
|   Y2 = (A7 & A10)# (A3 & A10)# (A8 & A10)# (A6 & A9)
|   Y3 = (A4 & A8) # (A4 & A5 )# (A4 & A7)
|   Y4 = A12 # A13 # A14
```

**SAMPLES OF BASIC GATES**

## 18. ERR 11

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~3]
|
|   Y1 = (A5 & A10)# (A1 & A11)# (A3 & A6) # (A4 & A8)
|   Y2 = (A7 & A5) # (A3 & A5) # (A5 & A9) # (A2 & A10)
|   Y3 = (A4 & A11)# (A6 & A12)# (A6 & A13)
```

**SAMPLES OF BASIC GATES**

## 19. ERR 12

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

six", and represents the signals A1, A2, A3, A4, A5,
and A6.

```
|PALce20V8  in:(A[1~13]),io:Y[1~3]
|
|   Y1 = (A8 & A10) # (A1 & A4)  # (A3 & A11)# (A1 & A9)
|   Y2 = (A6 & A10) # (A7 & A10) # (A5 & A8) # (A2 & A8)
|   Y3 = (A11 & A12)# (A11 & A13)
```

**SAMPLES OF BASIC GATES**

## 20. ERR 13

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce22V10  in:(A[1~4]),io:Y[1~8]
|
|   Y1 = A1 & A2 & A3 & A4'
|   Y2 = A1 & A2 & A3'& A4
|   Y3 = A1'& A2 & A3 & A4
|   Y4 = A1 & A2'& A3 & A4
|   Y5 = A1 & A2'& A3'& A4'
|   Y6 = A1'& A2 & A3'& A4'
|   Y7 = A1'& A2'& A3 & A4'
|   Y8 = A1'& A2'& A3'& A4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SAMPLES OF BASIC GATES**

## 21. ERR 14

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 # A5
|   Y2 = A2 # A6
|   Y3 = A3 # A7
|   Y4 = A4 # A8
```

**SAMPLES OF BASIC GATES**

## 22. ERR 15

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~8]),io:Y[1~4]
|
|   Y1 = A1 # A5'
|   Y2 = A2 # A6'
|   Y3 = A3 # A7
|   Y4 = A4 # A8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SAMPLES OF BASIC GATES**

## 23. CRR1

Here are basic logic gates written as Boolean equations.The notation A[1~6] is read "A one through six", and represents the signals A1,A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~9]),io:Y[1~4]
|
|   Y1 = (A1 ## A5) & A9
|   Y2 = (A2 ## A6) & A9
|   Y3 = (A3 ## A7) & A9
|   Y4 = (A4 ## A8) & A9
```

**SAMPLES OF BASIC GATES**

## 24. CRR2

Here are basic logic gates written as Boolean equations.The notation A[1~6]is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

```
|PALce16V8  in:(A[1~9]),io:Y[1~4]
|
|   Y1 = (A1 ## A5) & A9
|   Y2 = (A2 ## A6) & A9
|   Y3 = (A3 ## A7) & A9
|   Y4 = (A4 ## A8) & A9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SAMPLES OF BASIC GATES**

## 25. CRR3

Here are basic logic gates written as Boolean equations.The notation A[1~6] is read "A one through six", and represents the signals A1,A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~14]),io:Y[1~6]
|
|   Y1 = (A1 ## A7) & A13'
|   Y2 = (A2 ## A8) & A13'
|   Y3 = (A3 ## A9) & A13'
|   Y4 = (A4 ## A10) & A13'
|   Y5 = (A5 # A11 # A14) & A13'
|   Y6 = (A6 # A12 # A14) & A13'
```

**SAMPLES OF BASIC GATES**

## 26. CRR4

Here are basic logic gates written as Boolean equations.The notation A[1~6] is read "A one through six", and represents the signals A1,A2, A3, A4, A5, and A6.

```
|PALce20V8  in:(A[1~14]),io:Y[1~6]
|
|   Y1 = (A1 ## A7) & A13'
|   Y2 = (A2 ## A8) & A13'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| Y3 = (A3 ## A9) & A13' & A14'

| Y4 = (A4 ## A10) & A13'& A14'

| Y5 = (A5 ## A11) & A13'& A14'

| Y6 = (A6 ## A12) & A13'& A14'

**SAMPLES OF BASIC GATES**

## 27. TRC1

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

|PALce16V8 in:(A[1~9]),io:Y[1~4]

|

| Y1 =( A1 # A5 )& A9'

| Y2 =( A2 # A6 )& A9'

| Y3 =( A3 # A7 )& A9'

| Y4 =( A4 # A8 )& A9'

**SAMPLES OF BASIC GATES**

## 28. TRC2

Here are basic logic gates written as Boolean equations. The notation A[1~6] is read "A one through six", and represents the signals A1, A2, A3, A4, A5, and A6.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
|PALce16V8  in:(A[1~9]),io:Y[1~4]
|
|   Y1 =( A1 # A5 )& A9'
|   Y2 =( A2 # A6 )& A9'
|   Y3 =( A3 # A7 )& A9'
|   Y4 =( A4 # A8 )& A9'
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้