# เอกสารประกอบการสอน

# วงจรดิจิตอลและการออกแบบลอจิก

เรียบเรียง โดย นายชัยวัฒน์ จงกุลสถิตชัย

มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์

# คำนำ

เอกสารประกอบการสอน ในวิชาวงจรดิจิตอลและการออกแบบวงจรลอจิก (Digital Circuits and Logic Design) สำหรับสอนนักศึกษาระดับปริญญาตรีสาขาวิศวกรรมศาสตร์

เนื้อหาสาระเกี่ยวกับ ระบบจำนวนและรหัส การแปลงฐานเลข การแทนเลขฐานสิบ ด้วยเลขฐานสองแบบไม่มีเครื่องหมาย แบบมีเครื่องหมาย การบวก ลบ คูณ หาร พีชคณิตบูลลีน ผังคาร์โนห์ การออกแบบวงจรลอจิกคอมไบเนชันนอน วงจรแปลงรหัส วงจรถอดรหัส

ผู้เขียนได้รวบรวมจัดทำโครงการสอน โดยแบ่งรายละเอียด 3 ตอน คือ

ตอนที่1 หลักสูตรรายวิชา และกำหนดการสอน ประกอบด้วย ความมุ่งหมาย หลักสูตรลักษณะรายวิชา การแบ่งหน่วยหน่วยเรียน/บทเรียน/หัวข้อ/จุดประสงค์การสอน/การประเมินผลรายวิชา ตาราง กำหนดน้ำหนักคะแนน กำหนดการสอน/หนังสืออ้างอิง

ตอนที่2 การเตรียมการสอนรายสัปดาห์ ประกอบด้วย ใบเตรียมการสอน 1-16 สัปดาห์ โดยจัดสอบกลางภาค และปลายภาคใน สัปดาห์ที่9, 16

ตอนที่3 บันทึกการสอนใช้สำหรับบันทึกการสอนทุกสัปดาห์ โดยบันทึกเกี่ยวกับปัญหาจากการสอน แนวทางป้องกัน/แก้ไข และข้อเสนอแนะที่เกี่ยวกับการเรียนการสอนแต่ละครั้ง

ทั้งนี้ผู้เขียนคาดหวังว่า เอกสารฉบับนี้ จะเป็นเอกสารคู่มือผู้สอน ประกอบการสอนที่ได้มีการเตรียม และการวางแผนการสอนไว้อย่างรอบคอบ ซึ่งจะส่งผลให้การเรียนการสอนในรายวิชานี้มีประสิทธิภาพ และมีคุณภาพยิ่งขึ้น อันเป็นผลให้นักศึกษาได้รับความรู้อย่างเต็มที่ พร้อมกับได้ข้อคิด แนวคิด คุณธรรมจริยธรรม จากผู้สอนพอสมควรกับการศึกษาในระดับปริญญาตรีที่พึงมี

(นายชัยวัฒน์ จงกุลสถิตชัย)

**สารบัญ**


| **ตอนที่1** | หน้า |
|---|---|
| ความมุ่งหมายของหลักสูตร | 1 |
| ลักษณะรายวิชา | 2 |
| การแบ่งหน่วยเรียน /บทเรียน/หัวข้อ | 3-14 |
| กำหนดการสอน | 15-17 |
| จุดประสงค์การสอน | 18-29 |
| การประเมินผลรายวิชา | 30 |
| ตารางกำหนดน้ำหนักคะแนน | 31 |
| **ตอนที่2** | |
| ใบเตรียมการสอน | 32-46 |
| สัปดาห์ที่ 1 ระบบแอนะลอก/ระบบดิจิตอล ระบบตัวเลขฐานต่างๆ | 47-87 |
| สัปดาห์ที่ 2 การคำนวณทางคณิตศาสตร์ในระบบดิจิตอล | 88-122 |
| สัปดาห์ที่ 3 การลดรูปสมการลอจิก โดยใช้วิธีพีชคณิตบูลลีน และแผนผังคาร์นอร์ | 123-167 |
| สัปดาห์ที่ 4 การออกแบบวงจร Combination Logic Gates | 168-211 |
| สัปดาห์ที่ 5 การแปลงรหัส (Encoder) | 212-239 |
| สัปดาห์ที่ 6 การถอดรหัส ( Decoder) | 240-269 |
| สัปดาห์ที่ 7 การมัลติเพล็กซ์(Multiplex) และดีมัลติเพล็กซ์(Demultiplex) | 270-299 |
| สัปดาห์ที่ 8 วงจรเปรียบเทียบ และวงจร ALU | 300-331 |
| สัปดาห์ที่ 9 **สอบกลางภาค** | |
| สัปดาห์ที่ 10 การออกแบบวงจรบวก ลบ คูณ หาร | 332-369 |
| สัปดาห์ที่ 11 การออกแบบวงจรลอจิกซีเควนเชียล | 370-416 |
| สัปดาห์ที่ 12 การเลื่อนข้อมูล (Shift Register) | 417-452 |
| สัปดาห์ที่ 13 วงจรนับเลข (Counter) | 453-499 |
| สัปดาห์ที่ 14 การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล | 500-527 |
| สัปดาห์ที่ 15 การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม | 528-543 |
| สัปดาห์ที่ 16 **สอบปลายภาค** | |
| **ตอนที่3** | |
| ภาคผนวก วิธีการสอนและกิจกรรม , แบบฟอร์มบันทึกการสอน, | 544-546 |

# ตอนที่ 1

## ความมุ่งหมายของหลักสูตร

## ระดับปริญญาตรี

1. หลักสูตรวิศวกรรมศาสตร์ เป็นหลักสูตรที่มุ่งเน้นการสร้างบัณฑิตที่มีความรู้ ความสามารถทางด้านต่างๆ และสร้างเสริมประสบการณ์ ตลอดจนแนวคิดสู่การประกอบวิชาชีพในด้านวิศวกร

2. เพื่อผลิตบัณฑิตที่มีความรู้และความสามารถในการปฏิบัติงานด้านเทคโนโลยีโทรคมนาคม คอมพิวเตอร์ และทางเมคคาทรอนิกส์ ทั้งภาครัฐบาลและเอกชน

3. เพื่อผลิตบัณฑิตให้มีความรู้ ความสามารถและทักษะในการทำงาน ตลอดจนการประสานงานการติดตามและประเมินผลการปฏิบัติงาน

4. เพื่อฝึกฝนให้บัณฑิตมีความคิดริเริ่มสร้างสรรค์ มีกิจนิสัยในการค้นคว้า วางแผน เตรียมการรวมทั้งปรับปรุงและพัฒนาตนเองให้ก้าวหน้าอยู่เสมอ สามารถแก้ปัญหาด้วยหลักการและเหตุผล ปฏิบัติงานด้วยหลักวิชาการที่มีการวางแผนอย่างเป็นระบบซึ่งก่อให้เกิดผลสัมฤทธิ์ตามเป้าหมายอย่างประหยัด รวดเร็วและมีประสิทธิภาพ

5. เพื่อปลูกฝังให้บัณฑิตเป็นผู้มีคุณธรรม มีระเบียบวินัย ตรงต่อเวลา ซื่อสัตย์สุจริต ขยันหมั่นเพียร สำนึกในจรรยาวิชาชีพ รับผิดชอบต่อหน้าที่และสังคม

**ลักษณะรายวิชา**

| | |
|---|---|
| **1. รหัสและชื่อวิชา** | วงจรดิจิตอลและการออกแบบวงจรลอจิก<br>(Digital Logic Circuits and Design) |
| **2. สภาพรายวิชา** | วิชาชีพบังคับ |
| **3. ระดับรายวิชา** | - |
| **4. พื้นฐาน** | - |
| **5. เวลาศึกษา** | 54 คาบเรียนตลอด 18 สัปดาห์ ทฤษฎี 3 คาบต่อสัปดาห์ และนักศึกษาต้องใช้เวลาศึกษาค้นคว้านอกเวลา 3 ชั่วโมงต่อสัปดาห์ |
| **6. จำนวนหน่วยกิต** | 3 หน่วยกิต |
| **7. จุดมุ่งหมายรายวิชา** | 7.1.เข้าใจระบบตัวเลขฐานต่างๆ<br>7.2 เข้าใจเรื่องตรรกศาสตร์โดยใช้วงจรทางอิเล็กทรอนิกส์ได้ดี<br>7.3 สามารถวิเคราะห์ค่าเวลาของช่วงการทำงานวงจรลอจิกเกตได้<br>7.4 สามารถออกแบบวงจรลอจิกเกตได้<br>7.5 สามารถออกแบบวงจรพื้นฐาน ALU ได้<br>7.6 สามารถนำความรู้ไปใช้งานด้านการออกแบบทางอุตสาหกรรมอิเล็กทรอนิกส์<br>7.7 นำไปปฏิบัติการซ่อมเครื่องมือทางอิเล็กทรอนิกส์ที่มีการนำวงจรลอจิกเกตเป็นส่วนควบคุมระบบไฟฟ้าของเครื่องได้เป็นอย่างดี |
| **8. คำอธิบายรายวิชา** | ศึกษาเกี่ยวกับระบบแอนะลอกระบบดิจิตอล , ระบบตัวเลขฐานต่างๆการคำนวณด้านคณิตศาสตร์ในระบบดิจิตอล, การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีน และแผนผังคาร์นอร์, การออกแบบวงจรลอจิกคอมไบเนชัน, วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น, การออกแบบวงจรลอจิกซีเควนเชียล, การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม, การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการ | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 1 | ระบบแอนะลอกระบบดิจิตอล ระบบตัวเลขฐานต่างๆ<br>11 ระบบแอนะลอก/ระบบดิจิตอล<br>12 ระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน<br>13 การแปลงเลขฐานสองเป็นเลขฐานต่างๆ<br>131 หลักการแปลงเลขฐานสองเป็นฐานสิบ<br>132 หลักการแปลงเลขฐานสองเป็นฐานแปด<br>133 หลักการแปลงเลขฐานสองเป็นฐานสิบหก<br>14 การแปลงเลขฐานสิบเป็นเลขฐานต่างๆ<br>141 หลักการแปลงเลขฐานสิบเป็นฐานสอง<br>142 หลักการแปลงเลขฐานสิบเป็นฐานแปด<br>143 หลักการแปลงเลขฐานสิบเป็นฐานสิบหก<br>15 การแปลงเลขฐานแปดเป็นเลขฐานต่างๆ<br>151 หลักการแปลงเลขฐานแปดเป็นฐานสอง<br>152 หลักการแปลงเลขฐานแปดเป็นฐานสิบ<br>153 หลักการแปลงเลขฐานแปดเป็นฐานสิบหก<br>16 การแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ<br>161 หลักการแปลงเลขฐานสิบหกเป็นฐานสอง<br>162 หลักการแปลงเลขฐานสิบหกเป็นฐานแปด<br>163 หลักการแปลงเลขฐานสิบหกเป็นฐานสิบ<br>17 สรุปการแปลงเลขฐานต่าง ๆ | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 2 | การคำนวณด้านคณิตศาสตร์<br>21 หลักการคอมพลีเมนต์ของเลขฐาน<br>211 การคอมพลีเมนต์ของเลขฐาน2<br>212 การคอมพลีเมนต์ของเลขฐาน 8<br>213 การคอมพลีเมนต์ของเลขฐาน10<br>214 การคอมพลีเมนต์ของเลขฐาน16<br><br>22 หลักการบวกเลข<br>221 การบวกเลขฐาน2<br>222 การบวกเลขฐาน8<br>223 การบวกเลขฐาน16<br><br>23 หลักการลบเลข<br>231 การลบเลขฐาน2<br>232 การลบเลขฐาน8<br>233 การลบเลขฐาน16<br><br>24 หลักการคูณเลข<br>241 การคูณเลขฐาน 2<br>242 การคูณเลขฐาน 8<br>243 การคูณเลขฐาน 16<br><br>25 หลักการหารเลข<br>251 หลักการหารเลขฐาน 2<br>252 หลักการหารเลขฐาน 8<br>253 หลักการหารเลขฐาน 16 | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 3 | การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีนและแผนผังคาร์นอ<br>31 ทฤษฎีพีชคณิตบูลลีน<br>311 คำจำกัดความ Postulate<br>312 Venn Diagrams for Postulates<br>313 Duality<br>314 ทฤษฎีพื้นฐานหลักของพีชคณิตพร้อมถึงการพิสูจน์<br>315 การนำทฤษฎีพีชคณิตบูลลีนไปใช้ประโยชน์ในการลดเทอม<br>32 การพิสูจน์ทฤษฎีพีชคณิตบูลลีน<br>321 พิสูจน์โดยการแทนค่าตัวคงที่<br>322 พิสูจน์โดยใช้ตารางความจริง<br>323 พิสูจน์โดยใช้พีชคณิต<br>33 การลดรูปฟังก์ชัน<br>331 การลดรูปโดยใช้พีชคณิตบูลลีน<br>332 การลดรูปโดยใช้ K-Maps<br>34 พิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps | 3 | |
| 4 | การออกแบบวงจร Combination Logic Gates<br>41 วงจรเกตเบื้องต้นกับสัญลักษณ์มาตรฐาน IEEE<br>411 Truth Table of Logic Gates<br>412 การนำทฤษฎีของดีมอร์แกน มาใช้งานในรูปแบบ NAND กับ NOR<br>413 การออกแบบวงจรลอจิกเกตในเทอมของ minterm , Maxterm<br>414 Don’t care terms<br>42 การเขียนลอจิกเกตจากฟังก์ชันพีชคณิต<br>421 การเขียนTiming Diagram จากวงจรลอจิกเกต<br>422 การเขียนลอจิกฟังก์ชันจากสวิทซ์ คอนแทครีเลย์ | 12<br>3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 (ต่อ) | 43 รหัสข้อมูล<br>431ชนิดและลักษณะของรหัสแบบต่างๆ<br>432 รูปแบบรหัสแทนข้อมูล<br>433 ตารางเลขฐาน, รหัส<br>434 การออกแบบวงจรเข้ารหัส<br>435 การออกแบบวงจรเข้ารหัสBCD-8421<br>436 การออกแบบวงจรสร้างบิตตรวจสอบ<br>437 รหัสที่ใช้ในเครื่องคอมพิวเตอร์ | 3 | |
| | 44 การถอดรหัสทางดิจิตอลของเลขฐานสอง<br>441 การออกแบบวงจรถอดรหัสฐานสองขนาด 2บิต<br>442 การออกแบบวงจรถอดรหัสฐานสองขนาด 3บิต<br>443 การออกแบบวงจรถอดรหัสฐานสองขนาด 4บิต แบบ Tree<br>444 การออกแบบวงจรถอดรหัสฐานสองขนาด 14 บิตแบบ Tree<br>445 การออกแบบวงจรถอดรหัสฐานสองขนาด n บิต | 3 | |
| | 45 การออกแบบวงจรถอดรหัส<br>451 การออกแบบวงจรถอดรหัสBCD -8421 แสดงผลแบบแอลอีดี Seven Segment มีคาโถดต่อร่วมกัน (Common Cathoad)<br>452 การออกแบบวงจรถอดรหัสBCD -8421 แสดงผล แบบแอลอีดี Seven Segment มีแอโนดต่อร่วมกัน (Common Anode) | | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 (ต่อ) | 453 การออกแบบวงจรถอดรหัส แบบแสดงผลเอาต์พุตเดียว<br>454 การออกแบบวงจรถอดรหัสBCD เป็นรหัสเกรย์<br><br>46 การประยุกต์ใช้วงจรถอดรหัสในการออกแบบวงจรลอจิกเกต<br>461 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Minterm<br>462 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Maxterm<br><br>47 หลักการพื้นฐานของมัลติเพล็กซ์ และดีมัลติเพล็กซ์<br>471 หลักการเบื้องต้นของการมัลติเพล็กขนาด 2 ช่องข้อมูลเป็น 1 ช่องข้อมูล<br>472 หลักการมัลติเพล็กซ์ขนาด 4 ช่องข้อมูลเป็น1 ช่องข้อมูล<br>473 การออกแบบวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูลโดยใช้วงจรมัลติเพล็กซ์ 4 อินพุต 2ภาค<br>474 การนำวงจรมัลติเพล็กซ์ไปใช้งาน<br><br>48 หลักการทำงานของดีมัลติเพล็กซ์(Demultiplex)<br>481 หลักการวงจรดีมัลติเพล็กซ์ขนาด 1ช่องข้อมูลทางอินพุตเป็น 2 ช่องข้อมูลทางเอาต์พุต<br>482 หลักการออกแบบวงจรดีมัลติเพล็กซ์ขนาด 1 ช่องทางอินพุตเป็น 4 ช่องข้อมูลทางเอาต์พุต<br>483 การออกแบบวงจรดีมัลติเพล็กซ์ขนาด 4 เอาต์พุต<br>484 การออกแบบวงจรดีมัลติเพล็กซ์ขนาดn Bit | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 (ต่อ) | 49 การนำวงจรมัลติเพล็กซ์/วงจรดีมัลติเพล็กซ์มาใช้งาน<br>491 การนำวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์มาต่อใช้งาน<br>492 การส่งข้อมูลและกระจายข้อมูลขนาด nบิต | | |
| 5 | วงจรเปรียบเทียบ และการออกแบบวงจรALU | 6 | |
| | 51 การเปรียบเทียบทางแอนะลอก<br>511 การนำวงจรเปรียบเทียบทางแอนะลอกไปใช้งานร่วมกับวงจรดิจิตอลในการควบคุมสัญญาณ | 3 | |
| | 52 การเปรียบเทียบทางดิจิตอล<br>521 การออกแบบวงจรเปรียบเทียบ1 บิต ทางดิจิตอล<br>522 การออกแบบวงจรเปรียบเทียบ 2 บิต ทางดิจิตอล<br>523 การออกแบบวงจรเปรียบเทียบ 4 บิต ทางดิจิตอล<br>524 การออกแบบวงจรเปรียบเทียบ n บิต ทางดิจิตอล | | |
| | 53 หลักการของ ALU<br>531 การทำงานของหน่วยลอจิก (LU)<br>532 การทำงานของหน่วยคำนวณทางคณิตศาสตร์ (AU) | | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 5 (ต่อ) | 54 หลักการออกแบบวงจร ALU<br>541 การออกแบบการทำงานของวงจร ALU ขนาด 1 บิต<br>542 การออกแบบการทำงานของวงจร ALU ขนาด n บิต<br>55 หลักการวงจรบวก<br>551 การออกแบบวงจรบวกแบบไม่คิดตัวทดเข้า (Half Adder)<br>552 การออกแบบวงจรบวกแบบคิดตัวทดเข้า (Full Adder)<br>553 วงจรบวกโดยวิธีการมองตัวทดล่วงหน้า (Carry Look Ahead)<br>554 หลักการบวก4 จำนวนแบบ Carry Save<br><br>56 หลักการวงจรลบ<br>561 การออกแบบวงจรลบแบบไม่คิดตัวทดเข้า (Half Subtracter)<br>562 การออกแบบวงจรลบแบบคิดตัวทดเข้า (Full Subtracter)<br>563 การออกแบบวงจรบวก/ลบเลขขนาด 2 บิต แบบใช้ 2'Complement<br><br>57 หลักการวงจรคูณ<br>571 การออกแบบวงจรคูณ<br>572 การออกแบบวงจรคูณขนาด 2 บิต แบบใช้โมดุล | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 5 (ต่อ) | 58 หลักการวงจรหาร<br>581 การออกแบบวงจรหาร<br>582 การออกแบบหารเลขแบบ Algorithm ขนาด 2 บิต | | |
| 6 | การออกแบบวงจรลอจิกซีเควนเชียล | 9 | |
| | 61 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ | 3 | |
| | 611 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรขยายความกว้างของพัลส์ | | |
| | 612 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรลดความกว้างของพัลส์ | | |
| | 613 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรกำจัดพัลส์ที่ไม่ต้องการ | | |
| | 614 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ในรูปแบบไอซีสำเร็จรูป | | |
| | 62 การทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์ | | |
| | 621 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ทรานซิสเตอร์ | | |
| | 622 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีทีทีแอล | | |
| | 623 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้คริสตอลควบคุมความถี่ | | |
| | 624 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีซีมอส | | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 6 (ต่อ) | 63 การนำไอซี 555 ไปใช้งานในรูปแบบต่างๆ<br>631 วงจรโมโนสเตเบิลใช้ไอซี 555<br>632 วงจรอะสเตเบิลใช้ไอซี 555<br><br>64 การทำงานของวงจรไบสเตเบิล<br>641 R-S Latch<br>642 R-S ฟลิปฟลอป<br>643 T ฟลิปฟลอป<br>644 D ฟลิปฟลอป<br>645 J-K ฟลิปฟลอป<br>646 J-K Master-Slave ฟลิบฟลอบ<br>647 การแปลงฟลิบฟลอบแบบต่างๆ<br><br>65 ลักษณะทั่วไปของการเลื่อนข้อมูล<br>651 การเลื่อนข้อมูลแบบ( Serial In , Serial Out )<br>652 การเลื่อนข้อมูลไปทางขวา (Shift- Right Register)<br>653 การเลื่อนข้อมูลไปทางซ้าย (Shift -Left Register)<br>654 การเลื่อนข้อมูลแบบวนรอบกลับ (Shift Around Register)<br>655 การเลื่อนข้อมูลแบบขนาน (Parallel In , Parallel Out )<br>656 การเลื่อนข้อมูลเข้าแบบขนานและเลื่อนข้อมูลออกแบบอนุกรม( Parallel In , Serial Out)<br>657 การเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน (Serial In , Parallel Out)<br>658 วงจรเลื่อนข้อมูลแบบ Universal ขนาด 4 บิต | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 6 (ต่อ) | 66 หลักการทั่วไปของการนับ<br><br>67 การนับแบบริปเปิล (Asynchronous Counter )<br>671 วงจรนับขึ้นแบบริปเปิล<br>672 วงจรนับลงแบบริปเปิล<br>673 วงจรนับขึ้น/ลงแบบริปเปิล<br><br>68 การนับตามโหมด<br>681 วงจรนับขึ้นโหมด 6<br>682 วงจรนับขึ้นโหมด 11<br>683 วงจรนับลงโหมด 5<br>684 วงจรนับลงโหมด 8<br><br>69 การนับแบบซิงโครนัส (Synchronous Counter )<br>691 การออกแบบวงจรนับซิงโครนัส<br>692 การออกแบบวงจรนับซิงโครนัสรหัสเกรย์<br>693 การออกแบบวงจรนับแบบซิงโครนัส โหมด5<br>694 การออกแบบวงจรนับขึ้นสิบ<br>695 การออกแบบวงจรนับลงแบบซิงโครนัส ขนาด 4 บิต<br>696 การออกแบบวงจรนับกระโดดแบบซิงโครนัสชนิดกระโดดขนาด 4 บิต<br>697 การนำวงจรนับไปใช้ในการหารความถี่<br>698 การนำวงจรนับไปประยุกต์ใช้งาน | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 7 | การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล<br>71 การแปลงสัญญาณแอนะลอกเป็นดิจิตอล และดิจิตอลเป็นแอนะลอก<br>711 พื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอล<br>712 การชักตัวอย่าง (Sampling)<br>713 การควอนไตซ์<br>72 หลักการแปลงสัญญาณดิจิตอลเป็นแอนะลอก<br>721 Resolution<br>722 Percentage Resolution<br>723 การหาค่าแรงดันเอาต์พุต<br>724 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกโดยใช้วงจรแบบ R-2R Ladder<br>73 หลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล<br>731 การแปลงแอนะลอกเป็นดิจิตอล<br>732 การทำงานของA/Dแบบเคาว์เตอร์แร็มพ<br>733 Successive- Approximation ADC<br>734 การออกแบบวงจรแบบ Flash ADCs<br>74 การประยุกต์ใช้งานจากวงจรADC กับ DAC<br>741 การนำไอซี เบอร์ DAC 0808 ไปใช้งานแปลงค่า ดิจิตอลเป็นแอนะลอก<br>742 การนำระบบดิจิตอลไปใช้งานในเครื่องมือวัดแบบตัวเลข | 3 | |

**การแบ่งหน่วย / บทเรียน / และหัวข้อ**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 8 | การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม<br>81 โปรแกรมเมเบิลคอนโทรลเลอร์<br>811 ลักษณะเด่นของ PLC<br>812 ส่วนประกอบของระบบใน PLC<br>813 การจำแนกขนาด PLC<br>814 PLC แบบ Compact<br>815 PLC แบบ Modular<br>82 มาตรฐานสำหรับ PLC<br>83 มาตรฐานใหม่ IEC 1131 สำหรับ PLC<br>84 พื้นฐานการออกแบบระบบ PLC<br>841 หน่วยอินพุต<br>842 บล็อกไดอะแกรมของหน่วยอินพุต<br>843 หน่วยเอาต์พุต<br>844 บล็อกไดอะแกรมของหน่วยเอาต์พุต<br>85 การเขียนโปรแกรมของ PLC<br>851 ภาษาโปรแกรม<br>852 แลดเดอร์ไดอะแกรม (Ladder diagram : LD)<br>853 ฟังก์ชันบล็อกไดอะแกรม (Function block diagram : FBD)<br>854 อินสตรัคชันลิสท์ (Instruction list : IL)<br>855 สตรัคเจอร์เทซท์ (Structure text : ST)<br>856 ซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)<br>857 ซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)<br>86 อุปกรณ์ถ่ายกำลังงาน | 3 | |

**กำหนดการสอน**

| สัปดาห์ที่ | วัน/เดือน | คาบที่ | รายการสอน | หมายเหตุ |
|---|---|---|---|---|
| 1 | | 1 – 3 | 11 ระบบแอนะลอก/ระบบดิจิตอล<br>12 ระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน<br>13 การแปลงเลขฐานสองเป็นเลขฐานต่างๆ<br>14 การแปลงเลขฐานสิบเป็นเลขฐานต่างๆ<br>15 การแปลงเลขฐานแปดเป็นเลขฐานต่างๆ<br>15 การแปลงเลขฐานแปดเป็นเลขฐานต่างๆ<br>16 การแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ<br>17 สรุปการแปลงเลขฐานต่าง ๆ | |
| 2 | | 1 – 3 | 21 หลักการคอมพลีเมนต์ของเลขฐาน<br>22 หลักการบวกเลข<br>23 หลักการลบเลข<br>24 หลักการคูณเลข<br>25 หลักการหารเลข | |
| 3 | | 1 – 3 | 31 ทฤษฎีพีชคณิตบูลลีน<br>32 การพิสูจน์ทฤษฎีพีชคณิตบูลลีน<br>33 การลดรูปฟังก์ชัน<br>34 พิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps | |
| 4 | | 1 – 3 | 41 วงจรเกตเบื้องต้นกับสัญลักษณ์มาตรฐาน IEEE<br>42 การเขียนลอจิกเกตจากฟังก์ชันพีชคณิต | |
| 5 | | 1 – 3 | 43 รหัสข้อมูล | |
| 6 | | 1 – 3 | 44 การถอดรหัสทางดิจิตอลของเลขฐานสอง<br>45 การออกแบบวงจรถอดรหัส | |

**กำหนดการสอน (ต่อ)**

| สัปดาห์ที่ | วัน/เดือน | คาบที่ | รายการสอน | หมายเหตุ |
|---|---|---|---|---|
| 6 (ต่อ) | | | 46 การประยุกต์ใช้วงจรดีโค้ดเดอร์ในการออกแบบวงจรลอจิกเกต | |
| 7 | | 1 – 3 | 47 หลักการพื้นฐานของมัลติเพล็กซ์ และดีมัลติเพล็กซ์<br>48 หลักการดีมัลติเพล็กซ์<br>49 การนำวงจรมัลติเพล็กซ์ และดีมัลติเพล็กซ์มาใช้งาน | |
| 8 | | 1 – 3 | 51 การเปรียบเทียบทางแอนะลอก<br>52 วงจรเปรียบเทียบทางดิจิตอล<br>53 หลักการของ ALU<br>54 หลักการออกแบบวงจร ALU ขนาด 1 บิต | |
| 9 | | | **สอบกลางภาคหัวข้อที่ 11 - 54** | |
| 10 | | 1 – 3 | 55 หลักการวงจรบวก<br>56 หลักการวงจรลบ<br>57 หลักการคูณ<br>58 หลักการวงจรหาร | |
| 11 | | 1 – 3 | 61 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์<br>62 การทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์<br>63 การนำไอซี 555 ไปใช้งานในรูปแบบต่างๆ<br>64 การทำงานของวงจรไบสเตเบิล | |
| 12 | | 1 – 3 | 65 ลักษณะทั่วไปของการเลื่อนข้อมูล | |

**กำหนดการสอน (ต่อ)**

| สัปดาห์ที่ | วัน/เดือน | คาบที่ | รายการสอน | หมายเหตุ |
|---|---|---|---|---|
| 13 | | 1 – 3 | 66 หลักการทั่วไปของการนับ<br>67 การนับแบบริปเปิล (Asynchronous Counter )<br>68 การนับตามโหมด<br>69 การนับแบบซิงโครนัส (Synchronous Counter ) | |
| 14 | | 1 – 3 | 71 การแปลงสัญญาณแอนะลอกเป็นดิจิตอล และดิจิตอลเป็นแอนะลอก<br>72 หลักการแปลงสัญญาณดิจิตอลเป็นแอนะลอก<br>73 หลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล<br>74 การประยุกต์ใช้งานจากวงจรADC กับ DAC | |
| 15 | | | 81 โปรแกรมเมเบิลคอนโทรลเลอร์<br>82 มาตรฐานสำหรับ PLC<br>83 มาตรฐานใหม่ IEC 1131 สำหรับ PLC<br>84 พื้นฐานการออกแบบระบบ PLC<br>85 การเขียนโปรกแกรมของ PLC<br>86 อุปกรณ์ถ่ายกำลังงาน | |
| 16 | | | **สอบปลายภาคหัวข้อที่** 55 **-86** | |

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 1 | ระบบแอนะลอกระบบดิจิตอล ระบบตัวเลขฐานต่างๆ<br>11 เข้าใจระบบแอนะลอก/ระบบดิจิตอล<br><br>12 เข้าใจระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน<br><br>13 เข้าใจการแปลงเลขฐานสองเป็นเลขฐานต่างๆ<br>131 อธิบายหลักการแปลงเลขฐานสองเป็นฐานสิบได้<br>132 อธิบายหลักการแปลงเลขฐานสองเป็นฐานแปดได้<br>133 อธิบายหลักการแปลงเลขฐานสองเป็นฐานสิบหกได้<br><br>14 เข้าใจการแปลงเลขฐานสิบเป็นเลขฐานต่างๆ<br>141 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานสองได้<br>142 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานแปดได้<br>143 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานสิบหกได้<br><br>15 เข้าใจการแปลงเลขฐานแปดเป็นเลขฐานต่างๆ<br>151 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสองได้<br>152 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสิบได้<br>153 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสิบหกได้<br><br>16 เข้าใจการแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ<br>161 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานสองได้<br>162 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานแปดได้<br>163 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานสิบได้<br><br>17 สรุปการแปลงเลขฐานต่าง ๆได้ | 3 | |

## จุดประสงค์การสอน

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 2 | การคำนวณด้านคณิตศาสตร์ในระบบดิจิตอล<br><br>21 เข้าใจหลักการคอมพลีเมนต์ของเลขฐาน<br>211 อธิบายการคอมพลีเมนต์ของเลขฐาน2 ได้<br>212 อธิบายการคอมพลีเมนต์ของเลขฐาน 8 ได้<br>213 อธิบายการคอมพลีเมนต์ของเลขฐาน10 ได้<br>214 อธิบายการคอมพลีเมนต์ของเลขฐาน16 ได้<br><br>22 เข้าใจหลักการบวกเลข<br>221 อธิบายการบวกเลขฐาน2 ได้<br>222 อธิบายการบวกเลขฐาน8 ได้<br>223 อธิบายการบวกเลขฐาน16 ได้<br><br>23 เข้าใจหลักการลบเลข<br>231 อธิบายการลบเลขฐาน 2 ได้<br>232 อธิบายการลบเลขฐาน 8 ได้<br>233 อธิบายการลบเลขฐาน 16 ได้<br><br>24 เข้าใจหลักการคูณเลข<br>241 อธิบายการคูณเลขฐาน 2 ได้<br>242 อธิบายการคูณเลขฐาน 8 ได้<br>243 อธิบายการคูณเลขฐาน 16 ได้<br><br>25 เข้าใจหลักการหารเลข<br>251 อธิบายหลักการหารเลขฐาน 2 ได้<br>252 อธิบายหลักการหารเลขฐาน 8 ได้<br>253 อธิบายหลักการหารเลขฐาน 16 ได้ | 3 | |

## จุดประสงค์การสอน

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 3 | การลดรูปสมการลอจิก<br><br>31 เข้าใจทฤษฎีพีชคณิตบูลลีน<br>311 คำจำกัดความ Postulate<br>312 Venn Diagrams for Postulates<br>313 Duality<br>314 ทฤษฎีพื้นฐานหลักของพีชคณิตพร้อมการพิสูจน์<br>315 การนำพีชคณิตบูลลีนไปใช้ประโยชน์ในการลดเทอม<br><br>32 เข้าใจการพิสูจน์ทฤษฎีพีชคณิตบูลลีน<br>321 คำนวณพิสูจน์โดยการแทนค่าตัวคงที่ได้<br>322 คำนวณพิสูจน์โดยใช้ตารางความจริงได้<br>323 คำนวณพิสูจน์โดยใช้พีชคณิตได้<br><br>33 เข้าใจการลดรูปฟังก์ชัน<br>331 อธิบายการลดรูปโดยใช้พีชคณิตบูลลีนๆได้<br>332 อธิบายการลดรูปโดยใช้ K-Maps ได้<br><br>34 เข้าใจพิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps | 3 | |

## จุดประสงค์การสอน

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 | การออกแบบวงจรลอจิกคอมไบเนชัน | 12 | |
| | 41 เข้าใจวงจรเกตเบื้องต้นกับสัญลักษณ์มาตรฐาน IEEE | 3 | |
| | 411 อธิบาย Truth Table of Logic Gates ได้ | | |
| | 412 อธิบายการนำทฤษฎีของดีมอร์แกน มาใช้งานในรูปแบบ NAND กับ NOR ได้ | | |
| | 413 อธิบายการออกแบบวงจร ลอจิกเกตในเทอมของ minterm , Maxterm ได้ | | |
| | 414 อธิบายการใช้ Don’t care terms | | |
| | 42 เข้าใจการเขียนลอจิกเกตจากฟังก์ชันพีชคณิต | | |
| | 421 อธิบายการเขียนTiming Diagram จากวงจรลอจิกเกตได้ | | |
| | 422 อธิบายการเขียนลอจิกฟังก์ชันจากสวิทซ์ คอนแทครีเลย์ ได้ | | |
| | 43 เข้าใจรหัสข้อมูล | 3 | |
| | 431 อธิบายการออกแบบวงจรเข้ารหัสBCD-8421ได้ | | |
| | 432 อธิบายชนิดและลักษณะของรหัสแบบต่างๆ ได้ | | |
| | 433 อธิบายรูปแบบรหัสแทนข้อมูลได้ | | |
| | 434 เขียนตารางเลขฐาน, รหัสได้ | | |
| | 435 อธิบายการออกแบบวงจรเข้ารหัสได้ | | |
| | 436 อธิบายการออกแบบวงจรสร้างบิตตรวจสอบได้ | | |
| | 437 อธิบายรหัสที่ใช้ในเครื่องคอมพิวเตอร์ได้ | | |
| | 44 เข้าใจการถอดรหัสทางดิจิตอลของเลขฐานสอง | 3 | |
| | 441 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 2บิต ได้ | | |
| | 442 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 3บิต ได้ | | |
| | 443 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 4บิต แบบ Tree ได้ | | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 (ต่อ) | 444 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 14 บิตแบบ Tree ได้<br>445 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด n บิตได้<br><br>45 เข้าใจการออกแบบวงจรถอดรหัส<br>451 อธิบายการออกแบบวงจรถอดรหัสBCD -8421 แสดงผล แบบแอลอีดี Seven Segment มีแคโทดต่อร่วมกัน (Common Cathoad) ได้<br>452 อธิบายการออกแบบวงจรถอดรหัสBCD -8421 แสดงผล แบบแอลอีดี Seven Segment มีแอโนดต่อร่วมกัน (Common Anode) ได้<br>453 อธิบายการออกแบบวงจรถอดรหัสฐานสอง แบบแสดงผลเอาต์พุตเดียวได้<br>454 อธิบายการออกแบบวงจรถอดรหัสฐานสองเป็นรหัสเกรย์ได้<br><br>46 เข้าใจการประยุกต์ใช้วงจรถอดรหัสในการออกแบบวงจรลอจิกเกต<br>461 อธิบายการนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Minterm ได้<br>462 อธิบายการนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Maxterm ได้<br>47 เข้าใจหลักการพื้นฐานของมัลติเพล็กซ์ และดีมัลติเพล็กซ์<br>471 อธิบายหลักการเบื้องต้นของการมัลติเพล็กเซอร์ขนาด 2 ช่องข้อมูล เป็น 1 ช่องข้อมูลได้<br>472 อธิบายหลักการมัลติเพล็กเซอร์ขนาด 4 ช่องข้อมูล เป็น1 ช่องข้อมูลได้ | 3 | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 4 (ต่อ) | 473 อธิบายการออกแบบวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูล โดยใช้วงจรมัลติเพล็กซ์ 4 อินพุต 2 ภาคได้<br>474 อธิบายการนำวงจรมัลติเพล็กเซอร์ไปใช้งานได้<br>48 เข้าใจหลักการดีมัลติเพล็กเซอร์(Demultiplex)<br>481 อธิบายหลักการดีมัลติเพล็กเซอร์ขนาด 1 ช่องข้อมูลเป็น 2 ช่องข้อมูลได้<br>482 อธิบายหลักการออกแบบดีมัลติเพล็กเซอร์ขนาด 1 ช่องเป็น 4 ช่องข้อมูลได้<br>483 อธิบายการออกแบบดีมัลติเพล็กเซอร์ขนาด 4 เอาต์พุตได้<br>484 อธิบายการออกแบบวงจรดีมัลติเพล็กเซอร์ n เอาต์พุตได้<br>49 เข้าใจการนำวงจรมัลติเพล็กซ์ และดีมัลติเพล็กซ์มาใช้งาน<br>491 อธิบายการนำวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์มาต่อใช้งานได้<br>492 อธิบายการส่งข้อมูลและกระจายข้อมูลขนาด n บิตได้<br>วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น | | |
| 5 | 51 เข้าใจการเปรียบเทียบทางแอนะลอก<br>511 อธิบายการนำวงจรเปรียบเทียบทางแอนะลอกไปใช้งานร่วมกับวงจรดิจิตอลไปใช้ในการควบคุมได้<br>52 เข้าใจวงจรเปรียบเทียบทางดิจิตอล<br>521 อธิบายการออกแบบวงจรเปรียบเทียบ1 บิต ทางดิจิตอลได้<br>522 อธิบายการออกแบบวงจรเปรียบเทียบ 2 บิต ทางดิจิตอลได้<br>523 อธิบายการออกแบบวงจรเปรียบเทียบ 4 บิต ทางดิจิตอลได้ | 6<br>3 | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 5 (ต่อ) | 524 อธิบายการออกแบบวงจรเปรียบเทียบ n บิต ทางดิจิตอลได้<br>53 หลักการALU<br>531 การทำงานของหน่วยลอจิก(LU)<br>532 การทำงานของหน่วยทางคณิตศาสตร์(AU)<br>54 เข้าใจหลักการออกแบบวงจร ALU ขนาด 1 บิต<br>541 อธิบายการทำงานของวงจร ALU ขนาด 1 บิตได้<br>542 ออกแบบวงจร ALU ขนาด 1 บิตได้<br>55 เข้าใจหลักการวงจรบวก<br>551 อธิบายการออกแบบวงจรบวกแบบไม่คิดตัวทดเข้า (Half Adder) ได้<br>552 อธิบายการออกแบบวงจรบวกแบบคิดตัวทดเข้า (Full Adder) ได้<br>553 อธิบายวงจรบวกโดยวิธีการมองตัวทดล่วงหน้า (Carry Look Ahead) ได้<br>554 อธิบายหลักการบวก4 จำนวนแบบ Carry Save ได้<br>56 เข้าใจหลักการวงจรลบ<br>561 อธิบายการออกแบบวงจรลบแบบไม่คิดตัวทดเข้า (Half Subtracter) ได้<br>562 อธิบายการออกแบบวงจรลบแบบคิดตัวทดเข้า (Full Subtracter) ได้<br>563 อธิบายการออกแบบวงจรบวก/ลบเลขขนาด 2 บิต แบบใช้ 2'Complement ได้<br>57 เข้าใจหลักการคูณ<br>571 อธิบายการออกแบบวงจรคูณได้ | 3<br>3 | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 5 (ต่อ) | 572 อธิบายการออกแบบวงจรคูณขนาด 2 บิต แบบใช้โมดูลได้ | | |
| | 58 เข้าใจหลักการวงจรหาร | | |
| | 581 อธิบายการออกแบบวงจรหารได้ | | |
| | 582 อธิบายการออกแบบหารเลขแบบ Algorithm ขนาด 2 บิตได้ | | |
| 6 | การออกแบบวงจรลอจิกซีเควนเชียล | 9 | |
| | 61 เข้าใจการทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ | 3 | |
| | 611 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรขยายความกว้างของพัลซ์ได้ | | |
| | 612 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรลดความกว้างของพัลซ์ได้ | | |
| | 613 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรกำจัดพัลซ์ที่ ไม่ต้องการได้ | | |
| | 614 อธิบายการทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ในรูปแบบไอซี สำเร็จรูปได้ | | |
| | 62 เข้าใจการทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์ | | |
| | 621 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ทรานซิสเตอร์ได้ | | |
| | 622 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซี TTL ได้ | | |
| | 623 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้คริสตอลควบคุมความถี่ได้ | | |
| | 624 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีซีมอส ได้ | | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 6 (ต่อ) | 63 เข้าใจการนำไอซี 555 ไปใช้งานในรูปแบบต่างๆ<br>631 อธิบายวงจรโมโนสเตเบิลใช้ไอซี 555 ได้<br>632 อธิบายวงจรอะสเตเบิลได้<br>64 เข้าใจการทำงานของวงจรไบสเตเบิล<br>641 อธิบาย R-S Latch ได้<br>642 อธิบาย R-S ฟลิปฟลอป ได้<br>643 อธิบาย T ฟลิปฟลอปได้<br>644 อธิบาย D ฟลิปฟลอปได้<br>645 อธิบาย J-K ฟลิปฟลอปได้<br>646 อธิบาย J-K Master-Slave ฟลิบฟลอบได้<br>647 สามารถแปลงฟลิบฟลอบแบบต่างๆได้<br>65 เข้าใจลักษณะทั่วไปของการเลื่อนข้อมูล<br>651 อธิบายการเลื่อนข้อมูลแบบ Serial In , Serial Out ได้<br>652 อธิบายการเลื่อนข้อมูลไปทางขวา (Shift- Right Register) ได้<br>653 อธิบายการเลื่อนข้อมูลไปทางซ้าย (Shift -Left Register) ได้<br>654 อธิบายการเลื่อนข้อมูลแบบวนรอบกลับ Shift Around Register ได้<br>655 อธิบายการเลื่อนข้อมูลแบบขนาน Parallel In , Parallel Out ได้<br>656 อธิบายการเลื่อนข้อมูลเข้าแบบขนานและเลื่อนข้อมูลออกแบบอนุกรม Parallel In , Serial Out ได้<br>657 อธิบายการเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน Serial In , Parallel Out ได้<br>658 อธิบายวงจรเลื่อนข้อมูลแบบ Universal ขนาด 4 บิต ได้ | 3 | |

## จุดประสงค์การสอน

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| | 66 เข้าใจหลักการทั่วไปของการนับ<br><br>67 เข้าใจการนับแบบ ริปเปิล (Asynchronous Counter )<br>671 อธิบายวงจรนับขึ้นแบบริปเปิลได้<br>672 อธิบายวงจรนับลงแบบริปเปิลได้<br>673 อธิบายวงจรนับขึ้น/ลงแบบริปเปิลได้<br><br>68 เข้าใจการนับตามโหมด<br>681 อธิบายวงจรนับขึ้นโหมด 6 ได้<br>682 อธิบายวงจรนับขึ้นโหมด 11 ได้<br>683 อธิบายการนับลงโหมด 5 ได้<br>684 อธิบายการนับลงโหมด 8 ได้<br><br>69 เข้าใจการนับแบบซิงโครนัส (Synchronous Counter )<br>691 อธิบายการออกแบบวงจรนับแบบซิงโครนัสได้<br>692 อธิบายการออกแบบวงจรนับแบบรหัสเกรย์ได้<br>693 อธิบายการออกแบบวงจรนับแบบซิงโครนัสโหมด5 ได้<br>694 อธิบายการออกแบบวงจรนับขึ้นสิบได้<br>695 อธิบายการออกแบบวงจรนับลงแบบซิงโครนัส ขนาด 4 บิตได้<br>696 อธิบายการออกแบบวงจรนับกระโดดแบบ ซิงโครนัสขนาด 4 บิตได้<br>697 เข้าใจการนำวงจรนับไปใช้ในการหารความถี่<br>698 เข้าใจการนำวงจรนับไปประยุกต์ใช้งาน | 3 | |

## จุดประสงค์การสอน

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 7 | การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล<br><br>71 เข้าใจการแปลงสัญญาณแอนะลอกเป็นดิจิตอล และดิจิตอลเป็นแอนะลอก<br>711 อธิบายพื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอลได้<br>712 อธิบายการ Sampling ได้<br>713 อธิบายการควอนไทซ์ได้<br><br>72 เข้าใจหลักการแปลงสัญญาณดิจิตอลเป็นแอนะลอก<br>721 อธิบายResolution ได้<br>722 อธิบายPercentage Resolutionได้<br>723 อธิบายAnalog Outputได้<br>724 อธิบายวงจรแปลงรหัสดิจิตอลเป็นสัญญาณแอนะลอกได้<br><br>73 เข้าใจหลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล<br>731 อธิบายAnalog-To-Digital Conversion ได้<br>732 อธิบายDigital –Ramp ADCได้<br>733 อธิบายSuccessive- Approximation ADCได้<br>734 อธิบายการออกแบบวงจรแบบ Flash ADCs ได้<br><br>74 เข้าใจการประยุกต์ใช้งานจากวงจรADC กับ DAC<br>741 อธิบายการนำไอซี เบอร์ DAC 0808 ไปใช้งานแปลงค่า ดิจิตอลเป็นแอนะลอกได้<br>742 อธิบายการนำระบบดิจิตอลไปใช้งานในเครื่องมือวัดแบบตัวเลขได้ | 3 | |

**จุดประสงค์การสอน**

| หน่วยที่ | รายการสอน | เวลาเรียน ( คาบ) | |
|---|---|---|---|
| | | ทฤษฎี | ปฏิบัติ |
| 8 | การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม<br><br>81 เข้าใจโปรแกรมเมเบิลคอนโทรลเลอร์<br>811 อธิบายลักษณะเด่นของ PLC<br>812 อธิบายส่วนประกอบของระบบใน PLC<br>813 อธิบายการจำแนกขนาด PLC<br>814 อธิบาย PLC แบบ Compact<br>815 อธิบาย PLC แบบ Modular<br>82 เข้าใจมาตรฐานสำหรับ PLC<br><br>83 เข้าใจมาตรฐานใหม่ IEC 1131 สำหรับ PLC<br><br>84 เข้าใจพื้นฐานการออกแบบระบบ PLC<br>841 อธิบายหน่วยอินพุต<br>842 อธิบายบล็อกไดอะแกรมของหน่วยอินพุต<br>843 อธิบายหน่วยเอาต์พุต<br>844 อธิบายบล็อกไดอะแกรมของหน่วยเอาต์พุต<br>85 เข้าใจการเขียนโปรแกรมของ PLC<br>851 อธิบายภาษาโปรแกรม PLC ได้<br>852 อธิบายแลดเดอร์ไดอะแกรม (Ladder diagram : LD)<br>853 อธิบายฟังก์ชันบล็อกไดอะแกรม (Function block diagram : FBD)<br>854 อธิบายอินสตรัคชันลิสท์ (Instruction list : IL)<br>855 อธิบายสตรัคเจอร์เทซท์ (Structure text : ST)<br>856 อธิบายซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)<br>857 อธิบายซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)<br>86 เข้าใจอุปกรณ์ถ่ายกำลังงาน | 3 | |

# การประเมินผลรายวิชา

รายวิชานี้แบ่งเป็น 8 หน่วย แยกได้ 14 บทเรียน การวัดและประเมินผลรายวิชาจะ ดำเนินการ ดังนี้

1. วิธีการ ดำเนินการรวบรวมข้อมูลเพื่อการประเมินผลแยกออกเป็น 3 ส่วน โดยแบ่งแยกคะแนนแต่ละส่วนจากคะแนนเต็ม 100 คะแนน

1.1 ผลงานที่มอบหมาย ใบมอบงาน 20 คะแนน หรือ 20%

1.2 พิจารณาจากจิตพิสัย ความตั้งใจ และการเข้าร่วมกิจกรรม 10 คะแนน หรือ 10%

1.3 การทดสอบแต่ละหน่วยเรียน 70 คะแนน หรือ 70% โดยจัดแบ่งน้ำหนักคะแนนในแต่ละหน่วยตามตารางกำหนดน้ำหนักคะแนน

2. เกณฑ์ผ่านรายวิชา ผู้ที่จะผ่านรายวิชานี้จะต้อง

2.1 มีเวลาเข้าห้องเรียนไม่ต่ำกว่าร้อยละ 80 ของเวลาเรียนทั้งหมด

2.2 ผลรวมของคะแนนทั้งหมดต้องไม่ต่ำกว่าร้อยละ 50

3. เกณฑ์ค่าระดับคะแนน

3.1 พิจารณาตามเกณฑ์ผ่านรายวิชาตามข้อ 2. ผู้ที่ไม่ผ่านเกณฑ์ข้อ2. จะได้รับค่าระดับคะแนน F.

3.2 ผู้ที่ผ่านเกณฑ์ข้อ 2 จะได้รับค่าระดับคะแนนตามเกณฑ์การวัดผล

3.2.1 ค่าระดับคะแนนแบบอิงเกณฑ์ ดังนี้

| | |
|---|---|
| คะแนนร้อยละ 80 | ขึ้นไปได้ A |
| คะแนนร้อยละ 75 - 79 | ได้ B+ |
| คะแนนร้อยละ 70 - 74 | ได้ B |
| คะแนนร้อยละ 65 - 69 | ได้ C+ |
| คะแนนร้อยละ 60 - 64 | ได้ C |
| คะแนนร้อยละ 55 - 59 | ได้ D+ |
| คะแนนร้อยละ 50 - 54 | ได้ D |
| คะแนนร้อยละ 0 - 49 | ได้ F |

**ตารางกำหนดน้ำหนักคะแนน**

| เลขที่หน่วย | คะแนนรายหน่วยและน้ำหนักคะแนน / ชื่อหน่วย | น้ำหนักคะแนน | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | น้ำหนักคะแนนงานย่อย | พุทธิพิสัย | | | | ทักษะพิสัย | จิตพิสัย |
| | | | ความรู้-ความจำ | ความเข้าใจ | การนำไปใช้ | สูงกว่านำไปใช้ | | |
| 1 | ระบบแอนะลอกระบบดิจิตอล ระบบตัวเลขฐานต่างๆ | 10 | 1 | 1 | 4 | 4 | | |
| 2 | การคำนวณด้านคณิตศาสตร์ในระบบดิจิตอล | 4 | 1 | 1 | 1 | 1 | | |
| 3 | การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีนและแผนผังคาร์นอร์ | 4 | 1 | 1 | 1 | 1 | | |
| 4 | การออกแบบวงจรลอจิกคอมไบเนชัน | 10 | 1 | 1 | 2 | 6 | | |
| 5 | วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น | 12 | 1 | 2 | 4 | 5 | | |
| 6 | การออกแบบวงจรลอจิกซีเควนเชียล | 10 | 1 | 1 | 2 | 6 | | |
| 7 | การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล | 10 | 1 | 2 | 2 | 5 | | |
| 8 | การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม | 10 | 1 | 2 | 2 | 5 | | |
| ก | คะแนนภาควิชาการ | 70 | 8 | 11 | 18 | 33 | - | - |
| ข | คะแนนภาคผลงาน | 20 | หมายเหตุ | | | | | |
| ค | คะแนนภาคจิตพิสัย | 10 | | | | | | |
| รวมทั้งสิ้น | | 100 | | | | | | |

| สัปดาห์ที่ 1 | ตอนที่2<br>**ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 1 ระบบแอนะลอกระบบดิจิตอล ระบบตัวเลขฐานต่างๆ | |

**ชื่อบทเรียน**

ระบบแอนะลอกระบบดิจิตอล ระบบตัวเลขฐานต่างๆ

**จุดประสงค์การสอน**

11 เข้าใจระบบแอนะลอก/ระบบดิจิตอล

12 เข้าใจระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน

13 เข้าใจการแปลงเลขฐานสองเป็นเลขฐานต่างๆ

131 อธิบายหลักการแปลงเลขฐานสองเป็นฐานสิบได้

132 อธิบายหลักการแปลงเลขฐานสองเป็นฐานแปดได้

133 อธิบายหลักการแปลงเลขฐานสองเป็นฐานสิบหกได้

14 เข้าใจการแปลงเลขฐานสิบเป็นเลขฐานต่างๆ

141 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานสองได้

142 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานแปดได้

143 อธิบายหลักการแปลงเลขฐานสิบเป็นฐานสิบหกได้

15 เข้าใจการแปลงเลขฐานแปดเป็นเลขฐานต่างๆ

151 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสองได้

152 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสิบได้

153 อธิบายหลักการแปลงเลขฐานแปดเป็นฐานสิบหกได้

16 เข้าใจการแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ

161 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานสองได้

162 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานแปดได้

163 อธิบายหลักการแปลงเลขฐานสิบหกเป็นฐานสิบได้

17 สรุปการแปลงเลขฐานต่าง ๆ ได้

| สัปดาห์ที่ 2 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 2 การคำนวณด้านคณิตศาสตร์ในระบบดิจิตอล | |

**ชื่อบทเรียน**

การคำนวณด้านคณิตศาสตร์ในระบบดิจิตอล

**จุดประสงค์การสอน**

21 เข้าใจหลักการคอมพลีเมนต์ของเลขฐาน

211 อธิบายการคอมพลีเมนต์ของเลขฐาน2 ได้

212 อธิบายการคอมพลีเมนต์ของเลขฐาน 8 ได้

213 อธิบายการคอมพลีเมนต์ของเลขฐาน10 ได้

214 อธิบายการคอมพลีเมนต์ของเลขฐาน16 ได้

22 เข้าใจหลักการบวกเลข

221 อธิบายการบวกเลขฐาน2 ได้

222 อธิบายการบวกเลขฐาน8 ได้

223 อธิบายการบวกเลขฐาน16 ได้

23 เข้าใจหลักการลบเลข

231 อธิบายการลบเลขฐาน 2 ได้

232 อธิบายการลบเลขฐาน 8 ได้

233 อธิบายการลบเลขฐาน 16 ได้

24 เข้าใจหลักการคูณเลข

241 อธิบายการคูณเลขฐาน 2 ได้

242 อธิบายการคูณเลขฐาน 8 ได้

243 อธิบายการคูณเลขฐาน 16 ได้

25 เข้าใจหลักการหารเลข

251 อธิบายหลักการหารเลขฐาน 2 ได้

252 อธิบายหลักการหารเลขฐาน 8 ได้

253 อธิบายหลักการหารเลขฐาน 16 ได้

| สัปดาห์ที่ 3 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 3 การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีน และแผนผัง คาร์นอร์ | |

**ชื่อบทเรียน**

การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีน และแผนผัง คาร์นอร์

**จุดประสงค์การสอน**

31 เข้าใจทฤษฎีพีชคณิตบูลลีน

311 คำจำกัดความ Postulate

312 Venn Diagrams for Postulates

313 Duality

314 ทฤษฎีพื้นฐานหลักของพีชคณิตพร้อมการพิสูจน์

315 การนำพีชคณิตบูลลีนไปใช้ประโยชน์ในการลดเทอม

32 เข้าใจการพิสูจน์ทฤษฎีพีชคณิตบูลลีน

321 คำนวณพิสูจน์โดยการแทนค่าตัวคงที่ได้

322 คำนวณพิสูจน์โดยใช้ตารางความจริงได้

323 คำนวณพิสูจน์โดยใช้พีชคณิตได้

33 เข้าใจการลดรูปฟังก์ชัน

331 อธิบายการลดรูปโดยใช้พีชคณิตบูลลีนๆได้

332 อธิบายการลดรูปโดยใช้ K-Maps ได้

34 เข้าใจพิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps

| | | |
|---|---|---|
| สัปดาห์ที่ 4 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
| เวลา 3 คาบ | หน่วยที่ 4 การออกแบบวงจรลอจิกคอมไบเนชัน | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกคอมไบเนชัน

**จุดประสงค์การสอน**

41 เข้าใจวงจรเกตเบื้องต้นกับสัญลักษณ์มาตรฐาน IEEE

411 อธิบาย Truth Table of Logic Gates ได้

412 อธิบายการนำทฤษฎีของดีมอร์แกน มาใช้งานในรูปแบบ NAND กับ NOR ได้

413 อธิบายการออกแบบวงจร ลอจิกเกตในเทอมของ minterm , Maxterm ได้

414 อธิบายการใช้งาน Don’t care terms

42 เข้าใจการเขียนลอจิกเกตจากฟังก์ชันพีชคณิต

421 อธิบายการเขียนTiming Diagram จากวงจรโลจิกเกตได้

422 อธิบายการเขียนลอจิกฟังก์ชันจากสวิทซ์ คอนแทครีเลย์ได้

| สัปดาห์ที่ 5 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 4 การออกแบบวงจรลอจิกคอมไบเนชัน (ต่อ) | |

**ชื่อบทเรียน**

รหัสข้อมูล

**จุดประสงค์การสอน**

43 เข้าใจรหัสข้อมูล

431 อธิบายการออกแบบวงจรเข้ารหัสBCD-8421ได้

432 อธิบายชนิดและลักษณะของรหัสแบบต่างๆ ได้

433 อธิบายรูปแบบรหัสแทนข้อมูลได้

434 เขียนตารางเลขฐาน, รหัสได้

435 อธิบายการออกแบบวงจรเข้ารหัสได้

436 อธิบายการออกแบบวงจรสร้างบิตตรวจสอบได้

437 อธิบายรหัสที่ใช้ในเครื่องคอมพิวเตอร์ได้

| สัปดาห์ที่ 6 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 4 การออกแบบวงจรลอจิกคอมไบเนชัน (ต่อ) | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกคอมไบเนชัน

**จุดประสงค์การสอน**

44 เข้าใจการถอดรหัสทางดิจิตอลของเลขฐานสอง

441 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 2บิตได้

442 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 3บิตได้

443 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 4บิตแบบ Tree ได้

444 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด 14บิตแบบ Tree ได้

445 อธิบายการออกแบบวงจรถอดรหัสฐานสองขนาด n บิตได้

45 เข้าใจการออกแบบวงจรถอดรหัส

451 อธิบายการออกแบบวงจรถอดรหัสBCD -8421 แสดงผลแบบแอลอีดี Seven Segment มีแคโทดต่อร่วมกัน(Common Cathoad) ได้

452 อธิบายการออกแบบวงจรถอดรหัสBCD -8421 แสดงผล แบบแอลอีดี Seven Segment มีอาโนดต่อร่วมกัน(Common Anode) ได้

453 อธิบายการออกแบบวงจรถอดรหัสฐานสอง แบบแสดงผลเอาต์พุตเดียวได้

454 อธิบายการออกแบบวงจรถอดรหัสฐานสองเป็นรหัสเกรย์ได้

46 เข้าใจการประยุกต์ใช้วงจรดีโค้ดเดอร์ในการออกแบบวงจรลอจิกเกต

461 อธิบายการนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Minterm ได้

462 อธิบายการนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Maxterm ได้

| สัปดาห์ที่ 7 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 4 การออกแบบวงจรลอจิกคอมไบเนชัน (ต่อ) | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกคอมไบเนชัน

**จุดประสงค์การสอน**

47 เข้าใจหลักการพื้นฐานของมัลติเพล็กซ์ และดีมัลติเพล็กซ์

471 อธิบายหลักการเบื้องต้นของการมัลติเพล็กซ์ขนาด 2 ช่องข้อมูลเป็น 1 ช่องข้อมูลได้

472 อธิบายหลักการมัลติเพล็กซ์ขนาด 4 ช่องข้อมูลเป็น1 ช่องข้อมูลได้

473 อธิบายการออกแบบวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูล โดยใช้วงจรมัลติเพล็กซ์ 4 อินพุต 2 ภาคได้

474 อธิบายการนำวงจรมัลติเพล็กซ์ไปใช้งานได้

48 เข้าใจหลักการดีมัลติเพล็กซ์

481 อธิบายหลักการดีมัลติเพล็กซ์ขนาด 1 ช่องข้อมูลเป็น 2 ช่องข้อมูลได้

482 อธิบายหลักการออกแบบดีมัลติเพล็กซ์ขนาด 1 ช่องเป็น 4 ช่องข้อมูลได้

483 อธิบายการออกแบบดีมัลติเพล็กซ์ขนาด 4 เอาต์พุตได้

484 อธิบายการออกแบบวงจรดีมัลติเพล็กซ์ n เอาต์พุตได้

49 เข้าใจการนำวงจรมัลติเพล็กซ์ และดีมัลติเพล็กซ์มาใช้งาน

491 อธิบายการนำวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์มาต่อใช้งานได้

492 อธิบายการส่งข้อมูลและกระจายข้อมูลขนาด n บิตได้

| สัปดาห์ที่ 8 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 5 วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น | |

**ชื่อบทเรียน**

วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น

**จุดประสงค์การสอน**

51 เข้าใจการเปรียบเทียบทางแอนะลอก

511 อธิบายการนำวงจรเปรียบเทียบทางแอนะลอกไปใช้งานร่วมกับวงจรดิจิตอลไปใช้ในการควบคุมได้

52 เข้าใจวงจรเปรียบเทียบทางดิจิตอล

521 อธิบายการออกแบบวงจรเปรียบเทียบ1 บิต ทางดิจิตอลได้

522 อธิบายการออกแบบวงจรเปรียบเทียบ 2 บิต ทางดิจิตอลได้

523 อธิบายการออกแบบวงจรเปรียบเทียบ 4 บิต ทางดิจิตอลได้

524 อธิบายการออกแบบวงจรเปรียบเทียบ n บิต ทางดิจิตอลได้

53 เข้าใจหลักการของ ALU

531 อธิบายการทำงานของหน่วยลอจิก (LU) ได้

532 อธิบายการทำงานของหน่วยคำนวณทางคณิตศาสตร์ (AU) ได้

54 เข้าใจหลักการออกแบบวงจร ALU ขนาด 1 บิต

541 อธิบายการทำงานของวงจร ALU ขนาด 1 บิตได้

542 อธิบายการทำงานของวงจร ALU ขนาด n บิตได้

| สัปดาห์ที่ 10 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 5 วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น (ต่อ) | |

**ชื่อบทเรียน**

วงจรเปรียบเทียบ และการออกแบบวงจรเบื้องต้น

**จุดประสงค์การสอน**

55 เข้าใจหลักการวงจรบวก

551 อธิบายการออกแบบวงจรบวกแบบไม่คิดตัวทดเข้า (Half Adder) ได้

552 อธิบายการออกแบบวงจรบวกแบบคิดตัวทดเข้า (Full Adder) ได้

553 อธิบายวงจรบวกโดยวิธีการมองตัวทดล่วงหน้า(Carry Look Ahead)ได้

554 อธิบายหลักการบวก4 จำนวนแบบ Carry Save ได้

56 เข้าใจหลักการวงจรลบ

561 อธิบายการออกแบบวงจรลบแบบไม่คิดตัวทดเข้า(Half Subtracter)ได้

562 อธิบายการออกแบบวงจรลบแบบคิดตัวทดเข้า (Full Subtracter) ได้

563 อธิบายการออกแบบวงจรบวก/ลบเลขขนาด 2 บิตแบบใช้ 2'Complement ได้

57 เข้าใจหลักการคูณ

571 อธิบายการออกแบบวงจรคูณได้

572 อธิบายการออกแบบวงจรคูณขนาด 2 บิต แบบใช้โมดูลได้

58 เข้าใจหลักการวงจรหาร

581 อธิบายการออกแบบวงจรหารได้

582 อธิบายการออกแบบหารเลขแบบ Algorithm ขนาด 2 บิตได้

| สัปดาห์ที่ 11 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 6 การออกแบบวงจรลอจิกซีเควนเชียล | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกซีเควนเชียล

**จุดประสงค์การสอน**

61 เข้าใจการทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์

611 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรขยายความกว้างของพัลซ์ได้

612 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรลดความกว้างของพัลซ์ได้

613 อธิบายการนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรกำจัดพัลซ์ที่ ไม่ต้องการได้

614 อธิบายการทำงานของวงจรโมโนสเตเบิลมัลตไวเบรเตอร์ในรูปแบบไอซีสำเร็จรูปได้

62 เข้าใจการทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์

621 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ทรานซิสเตอร์ได้

622 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซี TTL ได้

623 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้คริสตอลควบคุมความถี่ได้

624 อธิบายวงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซี ซีมอส ได้

| สัปดาห์ที่ 11 (ต่อ) | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 6 การออกแบบวงจรลอจิกซีเควนเชียล (ต่อ) | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกซีเควนเชียล

**จุดประสงค์การสอน**

63 เข้าใจการนำไอซี 555 ไปใช้งานในรูปแบบต่างๆ

631 อธิบายวงจรโมโนสเตเบิลใช้ไอซี 555 ได้

632 อธิบายวงจรอะสเตเบิลได้

64 เข้าใจการทำงานของวงจรไบสเตเบิล

641 อธิบาย R-S Latch ได้

642 อธิบาย R-S ฟลิบฟลอบได้

643 อธิบาย T ฟลิบฟลอบได้

644 อธิบาย D ฟลิบฟลอบได้

645 อธิบาย J-K ฟลิบฟลอบได้

646 อธิบาย J-K Master-Slave ฟลิบฟลอบได้

647 คำนวณการแปลงฟลิบฟลอบแบบต่างๆได้

| สัปดาห์ที่ 12 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 6 การออกแบบวงจรลอจิกซีเควนเชียล | |

**ชื่อบทเรียน**

ลักษณะทั่วไปของการเลื่อนข้อมูล

**จุดประสงค์การสอน**

65 เข้าใจลักษณะทั่วไปของการเลื่อนข้อมูล

651 อธิบายการเลื่อนข้อมูลแบบ (Serial In , Serial Out) ได้

652 อธิบายการเลื่อนข้อมูลไปทางขวา (Shift- Right Register) ได้

653 อธิบายการเลื่อนข้อมูลไปทางซ้าย (Shift -Left Register) ได้

654 อธิบายการเลื่อนข้อมูลแบบวนรอบกลับ (Shift Around Register) ได้

655 อธิบายการเลื่อนข้อมูลแบบขนาน (Parallel In , Parallel Out) ได้

656 อธิบายการเลื่อนข้อมูลเข้าแบบขนานและเลื่อนข้อมูลออกแบบอนุกรม (Parallel In , Serial Out) ได้

657 อธิบายการเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน (Serial In , Parallel Out) ได้

658 อธิบายวงจรเลื่อนข้อมูลแบบ Universal ขนาด 4 บิต ได้

| สัปดาห์ที่ 13 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 6 การออกแบบวงจรลอจิกซีเควนเชียล (ต่อ) | |

**ชื่อบทเรียน**

การออกแบบวงจรลอจิกซีเควนเชียล

**จุดประสงค์การสอน**

66 เข้าใจหลักการทั่วไปของการนับ

67 เข้าใจการนับแบบริปเปิล (Asynchronous Counter )

671 อธิบายวงจรนับขึ้นแบบริปเปิลได้

672 อธิบายวงจรนับลงแบบริปเปิลได้

673 อธิบายวงจรนับขึ้น/ลงแบบริปเปิลได้

68 เข้าใจการนับตามโหมด

681 อธิบายวงจรนับขึ้นโหมด6 ได้

682 อธิบายวงจรนับขึ้นโหมด11 ได้

683 อธิบายการนับลงโหมด5 ได้

684 อธิบายการนับลงโหมด8 ได้

69 เข้าใจการนับแบบซิงโครนัส (Synchronous Counter )

691 อธิบายการออกแบบวงจรนับแบบซิงโครนัสได้

692 อธิบายการออกแบบวงจรนับแบบรหัสเกรย์ได้

693 อธิบายการออกแบบวงจรนับแบบซิงโครนัสโหมด5 ได้

694 อธิบายการออกแบบวงจรนับขึ้นสิบได้

695 อธิบายการออกแบบวงจรนับลงแบบซิงโครนัสขนาด 4 บิตได้

696 อธิบายการออกแบบวงจรนับกระโดดแบบซิงโครนัสขนาด 4 บิตได้

697 เข้าใจการนำวงจรนับไปใช้ในการหารความถี่

698 เข้าใจการนำวงจรนับไปประยุกต์ใช้งาน

| สัปดาห์ที่ 14 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 7 การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล | |

**ชื่อบทเรียน**

การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล

**จุดประสงค์การสอน**

71 เข้าใจการแปลงสัญญาณแอนะลอกเป็นดิจิตอล และดิจิตอลเป็นแอนะลอก

711 อธิบายพื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอลได้

712 อธิบายการ Sampling ได้

713 อธิบายการควอนไตซ์ได้

72 เข้าใจหลักการแปลงสัญญาณดิจิตอลเป็นแอนะลอก

721 อธิบายResolution ได้

722 อธิบายPercentage Resolutionได้

723 อธิบายAnalog Outputได้

724 อธิบายวงจรแปลงรหัสดิจิตอลเป็นสัญญาณแอนะลอกได้

73 เข้าใจหลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

731 อธิบายAnalog-To-Digital Conversion ได้

732 อธิบายDigital –Ramp ADCได้

733 อธิบายSuccessive- Approximation ADCได้

734 อธิบายการออกแบบวงจรแบบ Flash ADCs ได้

74 เข้าใจการประยุกต์ใช้งานจากวงจรADC กับ DAC

741 อธิบายการนำไอซี เบอร์ DAC 0808 ไปใช้งานแปลงค่าดิจิตอลเป็นแอนะลอกได้

742 อธิบายการนำระบบดิจิตอลไปใช้งานในเครื่องมือวัดแบบตัวเลขได้

| สัปดาห์ที่ 15 | **ใบเตรียมการสอน** | วิชา ท.วงจรดิจิตอลและการออกแบบวงจรลอจิก |
|---|---|---|
| เวลา 3 คาบ | หน่วยที่ 8 การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม | |

**ชื่อบทเรียน**

การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม

**จุดประสงค์การสอน**

81 เข้าใจโปรแกรมเมเบิลคอนโทรลเลอร์

811 อธิบายลักษณะเด่นของ PLC

812 อธิบายส่วนประกอบของระบบใน PLC

813 อธิบายการจำแนกขนาด PLC

814 อธิบาย PLC แบบ Compact

815 อธิบาย PLC แบบ Modular

82 เข้าใจมาตรฐานสำหรับ PLC

83 เข้าใจมาตรฐานใหม่ IEC 1131 สำหรับ PLC

84 เข้าใจพื้นฐานการออกแบบระบบ PLC

841 อธิบายหน่วยอินพุต

842 อธิบายบล็อกไดอะแกรมของหน่วยอินพุต

843 อธิบายหน่วยเอาต์พุต

844 อธิบายบล็อกไดอะแกรมของหน่วยเอาต์พุต

85 เข้าใจการเขียนโปรแกรมของ PLC

851 อธิบายภาษาโปรแกรม

852 อธิบายแลดเดอร์ไดอะแกรม (Ladder diagram : LD)

853 อธิบายฟังก์ชันบล็อกไดอะแกรม (Function block diagram : FBD)

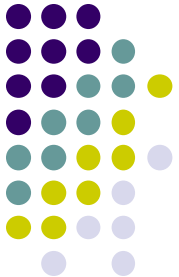
854 อธิบายอินสตรัคชันลิสท์ (Instruction list : IL)

855 อธิบายสตรัคเจอร์เทซท์ (Structure text : ST)

856 อธิบายซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)

857 อธิบายซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)

86 เข้าใจอุปกรณ์ถ่ายกำลังงาน



# DIGITAL CIRCUITS AND LOGIC DESIGN

# เนื้อหาคำอธิบาย

- วงจรดิจิตอลและการออกแบบวงจรลอจิก

  - หลักสูตรปริญญาตรีวิศวกรรมศาสตร์

  - สื่อการสอนประกอบด้วยเครื่องคอมพิวเตอร์

  - เครื่องฉายโปรเจคเตอร์

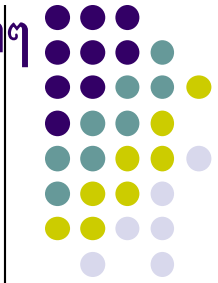
  - จำนวน 5 คาบ ทฤษฎี 2 คาบ ปฏิบัติ 3 คาบ


Chaiwatt.j@gmail.com

Wat8765

Line ID อ.เต๋า: tao_thaelasutt

# เนื้อหาคำบรรยายในสัปดาห์ที่ 1 ระบบแอนะล็อก/ระบบดิจิตอล ระบบตัวเลขฐานต่างๆ



11 ระบบแอนะลอก/ระบบดิจิตอล

12 ระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน

13 การแปลงเลขฐานสองเป็นเลขฐานต่างๆ

131 หลักการแปลงเลขฐานสองเป็นฐานสิบ

132 หลักการแปลงเลขฐานสองเป็นฐานแปด

133 หลักการแปลงเลขฐานสองเป็นฐานสิบหก

14 การแปลงเลขฐานสิบเป็นเลขฐานต่างๆ

141 หลักการแปลงเลขฐานสิบเป็นฐานสอง

142 หลักการแปลงเลขฐานสิบเป็นฐานแปด

143 หลักการแปลงเลขฐานสิบเป็นฐานสิบหก

15 การแปลงเลขฐานแปดเป็นเลขฐานต่างๆ

151 หลักการแปลงเลขฐานแปดเป็นฐานสอง

152 หลักการแปลงเลขฐานแปดเป็นฐานสิบ

153 หลักการแปลงเลขฐานแปดเป็นฐานสิบหก

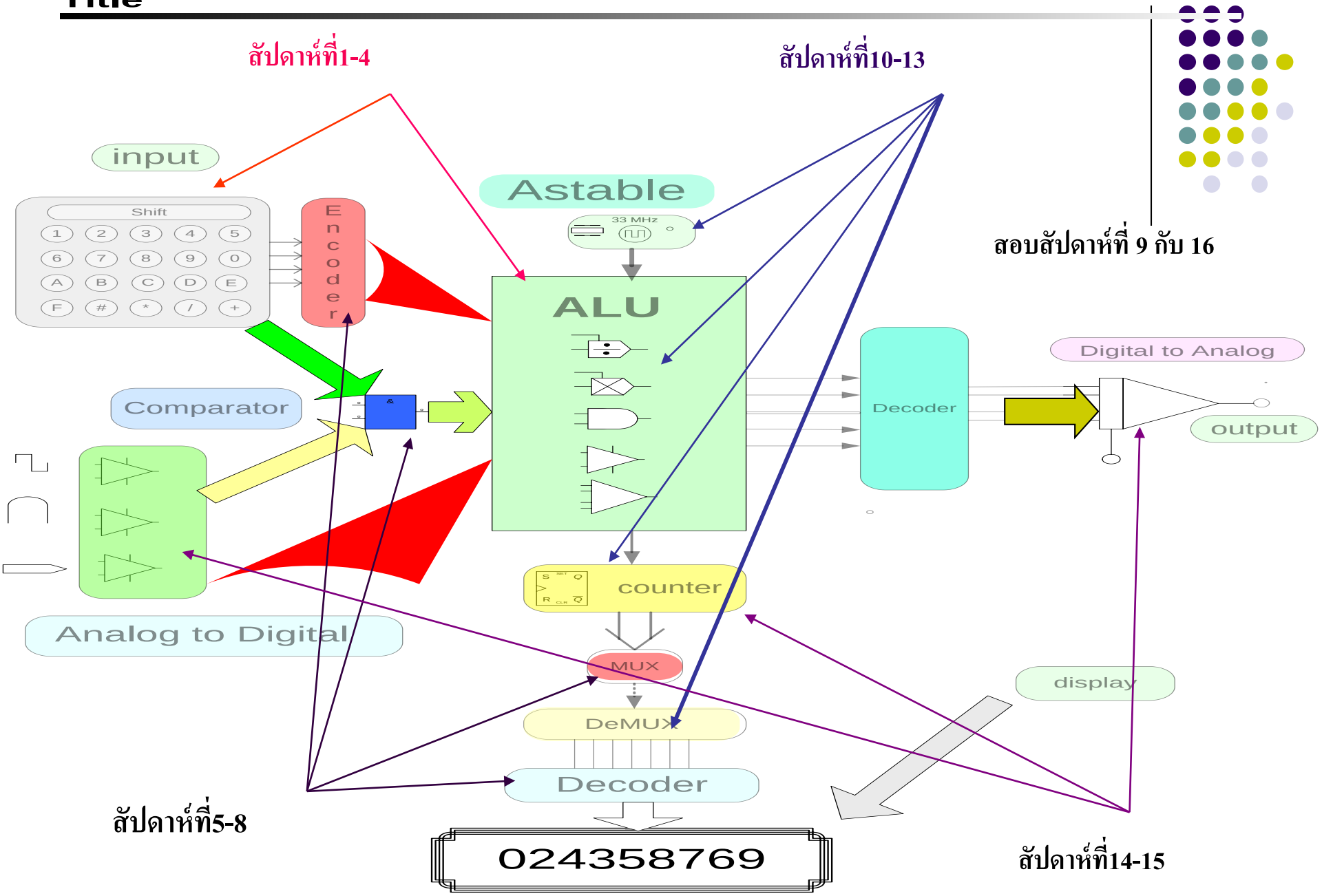
16 การแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ

161 หลักการแปลงเลขฐานสิบหกเป็นฐานสอง

162 หลักการแปลงเลขฐานสิบหกเป็นฐานแปด

163 หลักการแปลงเลขฐานสิบหกเป็นฐานสิบ

17 `สรุปการแปลงเลขฐานต่างๆ


สัปดาห์ที่1-4
สัปดาห์ที่10-13
input
Shift
1 2 3 4 5
6 7 8 9 0
A B C D E
F # * / +
Encoder
Astable
33 MHz
ALU
สอบสัปดาห์ที่ 9 กับ 16
Comparator
Decoder
Digital to Analog
output
I/O
counter
Analog to Digital
MUX
display
DeMUX
Decoder
024358769
สัปดาห์ที่5-8
สัปดาห์ที่14-15

# อธิบาย

- **สัปดาห์ที่ 1-4** อธิบายถึงความสัมพันธ์ระหว่าง ระบบแอนะลอกกับระบบดิจิตอล เลขฐานต่างๆ จนถึงระบบแปลงเลขฐาน 2 ,3, 8 , 10 , 16 ซึ่งจะนำไปใช้ในวงจรเข้ารหัส เอ็นโค้ดเดอร์(Encoder) ในสัปดาห์ที่ 3 อธิบายการทำงานของวงจรดิจิตอล ซึ่งมีเนื้อหาเกี่ยวกับกฎพีชคณิตบูลลีน , ดีมอร์แกน สมการบูลลีน และการลดรูปสมการบูลลีน ด้วยตาราง K-Map เพื่อนำไปออกแบบวงจรคอมไบเน่อร์ต่อไป
- **สัปดาห์ที่ 5-8** อธิบายถึงหลักการคำนวณทางคณิตศาสตร์โดยใช้วงจรเกต การทำงานของวงจรเข้ารหัสแบบต่างๆ และวงจรถอดรหัสชนิดต่างๆ จนถึงการออกแบบวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์ รวมไปจนถึงการออกแบบตัวแสดงผลทางเอาต์พุตในรูปแบบของเลขฐาน10 โดยใช้ Seven Segment

# อธิบาย(ต่อ)

- **สัปดาห์ที่ 10-13** จะกล่าวถึงการออกแบบวงจรซีเควนเชียลชนิดของฟลิบฟลอบ (Flip-flop) ต่างๆ ตลอดจนการนำไปใช้งานในรูปแบบรีจิสเตอร์ (Register) และวงจรนับเลข (Counter) ที่ใช้ในระบบดิจิตอล
- **สัปดาห์ที่14-15** การประยุกต์ใช้วงจรดิจิตอลในทางอุตสาหกรรมในรูปแบบ PLC หลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล และการแปลง สัญญาณดิจิตอลเป็นสัญญาณแอนะลอก
- จากเนื้อหาที่กล่าวมาทั้งหมดในวิชาวงจรดิจิตอลและการออกแบบลอจิก จะครอบคลุมเนื้อหาพื้นฐานของระบบดิจิตอลทั้งหมดซึ่งความรู้พื้นฐานที่จะนำไปศึกษาระบบคอมพิวเตอร์ทางฮาร์ดแวร์ได้เป็นอย่างดี อีกทั้งนำความรู้ที่ได้ไปออกแบบวงจรลอจิกเกต รวมถึงการนำไปวิเคราะห์วงจรที่ถูกใช้งานทางอุตสาหกรรมเมื่อนักศึกษาจบการศึกษา
- สัปดาห์ที่ 9,16 สอบกลางภาคและปลายภาค

**-เอกสารอ้างอิง** ที่นำมาใช้เป็นตัวอย่างขึ้นมา 12 รายการ ให้นักศึกษาไปค้นคว้าเพิ่มเติมในห้องสมุดที่ยังมีอีกจำนวนมาก เนื้อหาก็คงแตกต่างกันบ้าง แต่ก็ยังคงสาระหลักๆ ที่จะศึกษาในภาคการศึกษานี้

# เอกสารอ้างอิง


1. Douglas V.Hall Digital circuit and System, MC Graw-Hill Publishing Company, 1989

2. Milos D.Ercegovac, Tomas Lang and Jame H.Morino, Introduction to Digital System, John wiley & sons, Inc 1999.

3. Victor P.Nelson, H.Troy Nagle, Bill D.carrall J.David Trwin, Digital logic circuit analysis and design, Prentice Hall, International edition,1995.

4. M.Morris Mano, Digital design, Third edition, Prentice Mall, International edition , 2002.

5. Ronald J.Tocci , Digital System Principles and Applications, Prentice Hall , Fifth Edition, 1991

6. วัฒนา แก้วดุก การออกแบบดิจิตอลวิศวกรรม, ทฤษฎีและการประยุกต์ใช้งาน, สำนักพิมพ์ สสท.(ไทย-ญี่ปุ่น), 2546.

7. ธนัท ชัยยุทธ, กณพ แก้วพิชัย ดิจิตอลพื้นฐาน, บริษัทซีเอ็ดยูเคชัน จำกัด 2521

8. ธีรวัฒน์ ประกอบผล ดิจิตอลเทนิค, บริษัทซีเอ็ดยูเคชัน 2546

9. ถวัลย์วงศ์ ไกรโรจนานันท์, อิเล็กทรอนิกส์ระบบดิจิตอล ฉบับปรับปรุง, สวทช. 2542

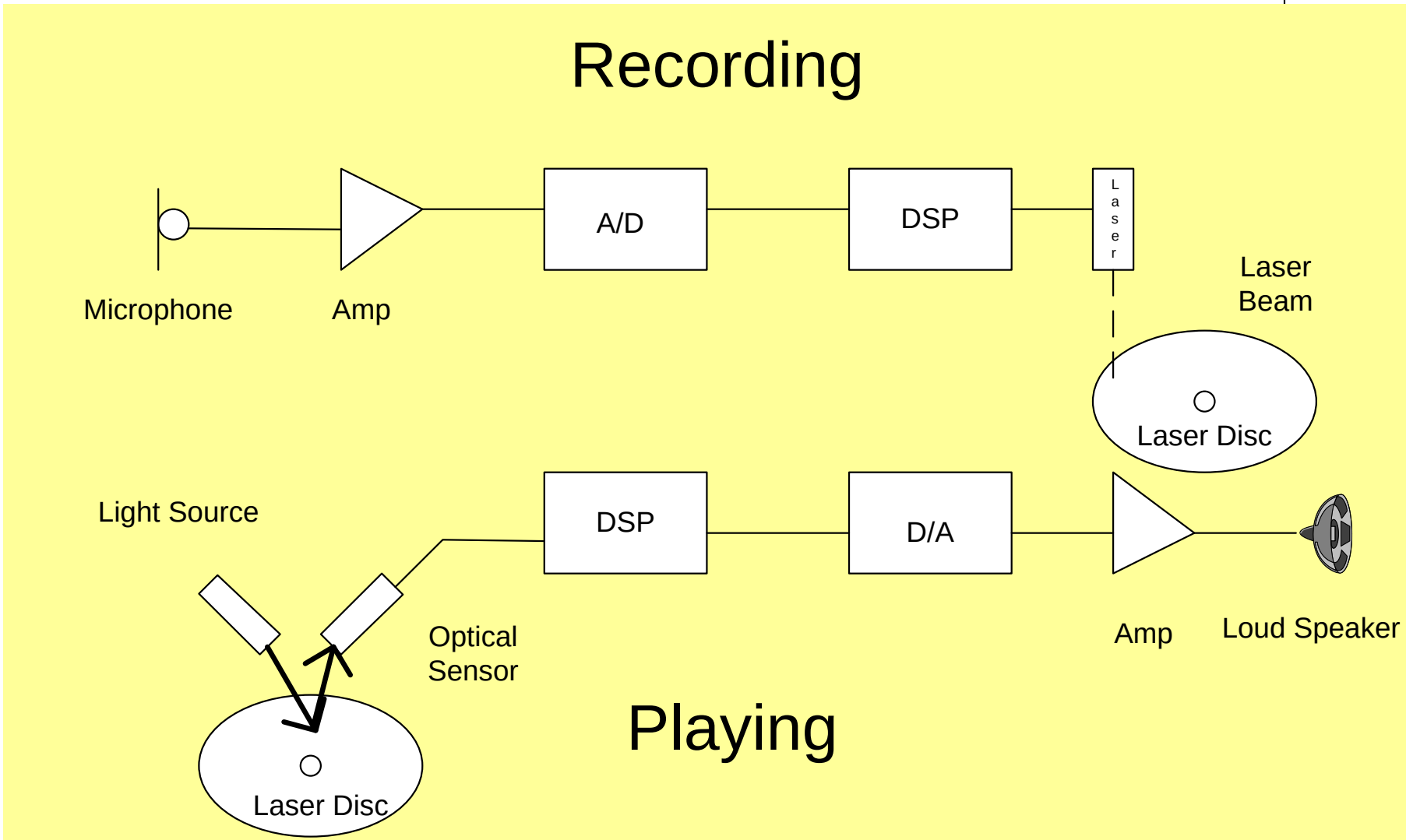
10.ธวัชชัย เลื่อนฉวี, อนุรักษ์ เถื่อนศิริ, ดิจิตอลเทคนิค เล่ม 1-2 , 2532, 2543

11.มงคล อาวัชนาการ , โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ,มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น

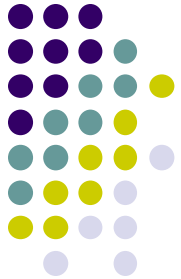
12.ศัพท์เทคนิควิศวกรรมไฟฟ้าสื่อสาร สมาคมวิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์, กันยายน, 2541

# 11 ระบบแอนะลอก/ระบบดิจิตอล

## ระบบการทำงานแอนะลอกกับดิจิตอล



# อธิบาย



- ความสัมพันธ์การทำงานของสัญญาณแอนะลอกกับสัญญาณดิจิตอล (Recording)

  ระบบการบันทึกสัญญาณเสียงแบบดิจิตอล

- ไมค์โครโฟน (Microphone)ทำหน้าที่รับสัญญาณเสียงเปลี่ยนเป็นสัญญาณทางไฟฟ้า ในรูปแบบสัญญาณแอนะลอก

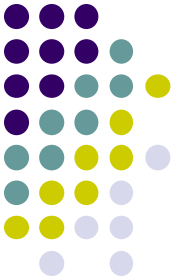
- วงจรขยายสัญญาณเสียง(Amplifier) ทำหน้าที่ขยายสัญญาณเสียงทางแอนะลอกให้มีขนาดเพียงพอต่อการทำงานในภาคถัดไป ในวงจรนี้มักจะรวมวงจรการปรับแต่งคุณภาพสัญญาณเสียงไว้ด้วย

- วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล A/Dทำหน้าที่เปลี่ยนสัญญาณเสียงแบบสัญญาณทางแอนะลอกเป็นสัญญาณทางเสียงในรูปแบบสัญญาณทางดิจิตอล

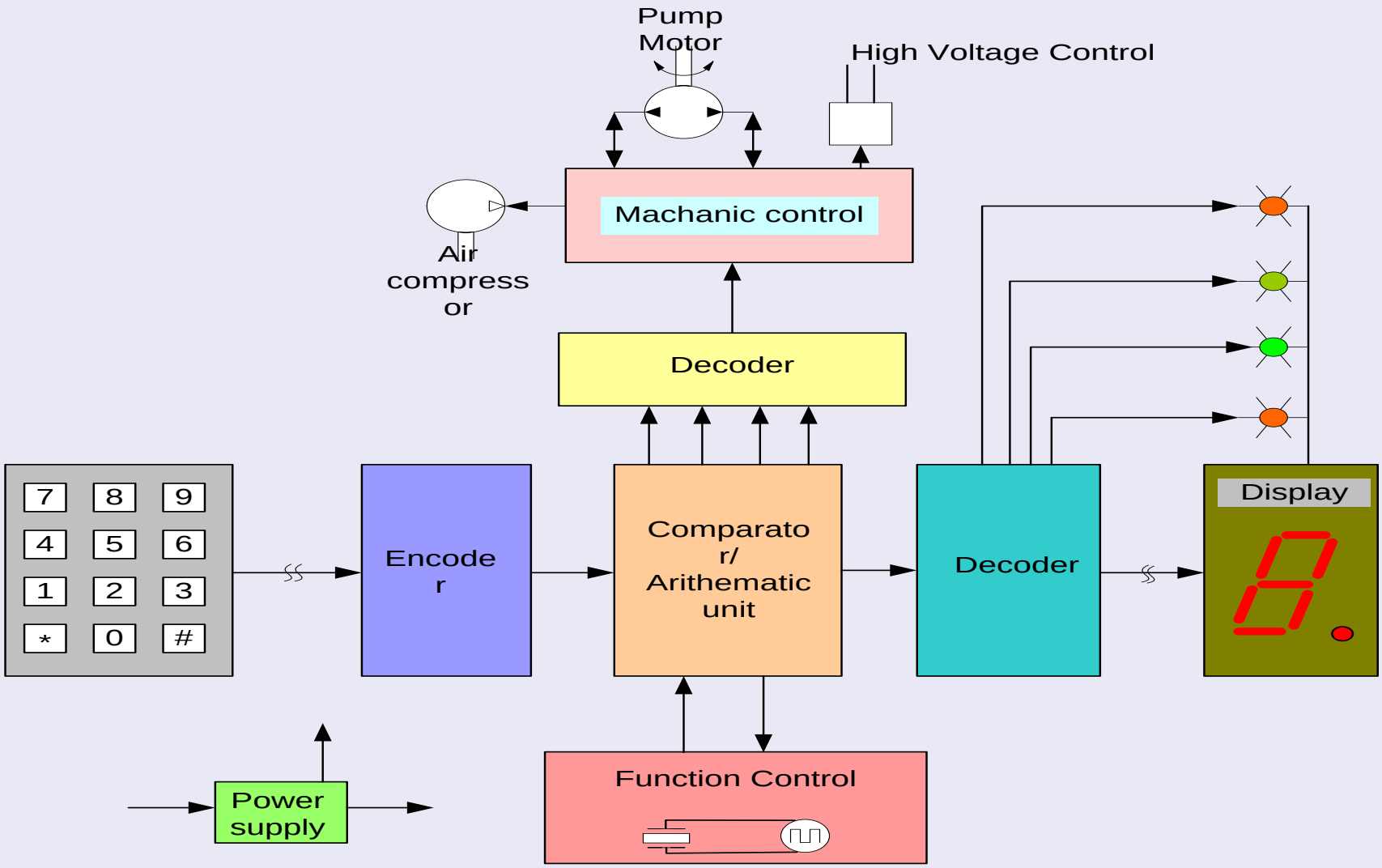
- วงจร DSP (Digital Signal Processing )ทำหน้าที่จัดรูปแบบสัญญาณทางดิจิตอลในการบันทึกลงบนแผ่น CD

- วงจร Laser ทำหน้าที่เปลี่ยนสัญญาณทางไฟฟ้าในรูปแบบดิจิตอลให้เป็นแสง เพื่อบันทึกลงในแผ่นเลเซอร์ดิสก์

# อธิบาย(ต่อ)



- ความสัมพันธ์การทำงานของสัญญาณดิจิตอล กับสัญญาณ แอนะลอกขณะ Playing
- รูปแบบสัญญาณเสียงที่ถูกบันทึกในรูปแผ่นเลเซอร์ดิสก์ ถูกนำกลับมาเป็นสัญญาณทางเสียงมีหลักการดังนี้
- Light Source เป็นแหล่งจ่ายแสงเลเซอร์
- Optical Sensor รับสัญญาณแสงสะท้อนจากแผ่นเลเซอร์ดิสก์ที่บันทึกเสียงไว้ แปลงเป็นสัญญาณทางไฟฟ้าในรูปแบบสัญญาณทางดิจิตอล
- วงจรDSP ทำหน้าที่จัดรูปแบบสัญญาณทางดิจิตอลให้เป็นไปตามการจัดเรียงสัญญาณข้อมูลทางเสียงให้เหมือนกับตอนบันทึกครั้งแรก
- วงจร D/A ทำหน้าที่แปลงสัญญาณเสียงในรูปแบบดิจิตอลให้เป็นเสียงในรูปแบบแอนะลอก
- วงจร Amplifier ทำหน้าที่ขยายสัญญาณเสียงรวมถึงการปรับแต่งคุณภาพทางเสียง ที่อยู่ในรูปแบบสัญญาณทางไฟฟ้า ให้มีค่าความแรงเพียงพอที่จะเปลี่ยนเป็นสัญญาณทางเสียง
- Loud –Speaker ทำหน้าที่เปลี่ยนสัญญาณทางไฟฟ้าในรูปแบบแอนะลอกเป็นสัญญาณทางเสียง




# การทำงานของวงจรดิจิตอลในการควบคุมหน้าที่ต่างๆ

# อธิบาย

- **หลักการทำงานของวงจรดิจิตอลในระบบควบคุม**

- แป้นรับสัญญาณทางอินพุตทำหน้าที่รับอินพุตเพื่อควบคุมการทำงานในรูปแบบดิจิตอล ในเลขฐาน10 ฐาน8 ฐาน16 หรืออักขระในรูปแบบทางภาษาควบคุม

- วงจรเข้ารหัส (Encoder ) ทำหน้าที่แปลงรหัสที่ส่งมาจากแป้นตัวเลขเป็นรหัสฐาน 2ในรูปแบบที่เรียกว่ารหัสทางดิจิตอล ซึ่งมีคุณสมบัติในการนำไปคำนวณและเปรียบเทียบทางคณิตศาสตร์ได้ ผลจากการคำนวณสามารถนำไปเป็นสัญญาณควบคุมทางเอาต์พุต

- วงจรเปรียบเทียบและคำนวณทางคณิตศาสตร์ ( Comparator /Arithmetic Logic Unit) เป็นวงจรคำนวณทางคณิตศาสตร์โดยใช้ฟังก์ชันลอจิกเกตทำหน้าที่เปรียบเทียบค่าทางดิจิตอล แล้วส่งผลไปควบคุมทางอุปกรณ์ทางกล

# อธิบาย(ต่อ)

**วงจรควบคุมมีหน้าที่การทำงาน (Function Control) เป็นส่วนที่ถูกกำหนดมา**

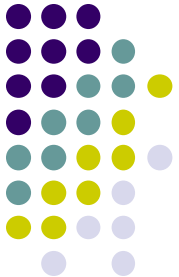
- เพื่อควบคุมการทำงานของเครื่องตามหน้าที่ที่มีการกำหนดไว้ ส่วนประกอบหลักของวงจรจะทำงานตามลำดับเวลา (Sequential Times) และตั้งค่าเวลาการทำงาน (Timer) สัญญาณควบคุมจะถูกส่งไปเปรียบเทียบค่าที่ป้อนทางอินพุตโดยวงจร ALU
- วงจรถอดรหัส (Decorder) ทำหน้าที่แปลงรหัสฐานต่าง ๆ เช่น ฐาน2 ให้อยู่ในรูปแบบของเลขฐาน 10 มี LED ต่อร่วมกันในรูปแบบ(Seven Segment) แสดงค่าได้ตั้งแต่ 0 ,1 , 2 .....9 หากวงจรจะแสดงค่าเป็นรหัส BCD ก็สามารถแสดงผลด้วย LED 4 ตัว ด้านบน ส่วนบนเป็นวงจรถอดรหัสจาก BCD เป็นรหัสฐาน2 ส่งไปยังส่วนควบคุมระบบทางกล กับทางไฟฟ้า ระดับแรงดันสูง (220 VAC)
- วงจรควบคุมทางกล (Mechanic Control ) ทำหน้าที่รับสัญญาณควบคุมในรูปแบบดิจิตอล แปลงเป็นสัญญาณทางแอนะลอกเพื่อควบคุมแรงดันนำไปปรับความเร็วของเครื่องกลในรูปแบบมอเตอร์ไฟฟ้า ในส่วนนี้มอเตอร์ไฟฟ้าจะควบคุมการไหลของปั๊มอากาศในรูปแบบนิวส์เมติกส์ และปั๊มน้ำมันในรูปแบบไฮดรอลิกส์

# 12 ระบบตัวเลขฐานต่างๆ เปรียบเทียบกัน

| ฐาน10 | ฐาน2 | ฐาน3 | ฐาน8 | ฐาน16 |
|---|---|---|---|---|
| 0 | 0000 | 00 | 00 | 0 |
| 1 | 0001 | 01 | 01 | 1 |
| 2 | 0010 | 02 | 02 | 2 |
| 3 | 0011 | 10 | 03 | 3 |
| 4 | 0100 | 11 | 04 | 4 |
| 5 | 0101 | 12 | 05 | 5 |
| 6 | 0110 | 20 | 06 | 6 |
| 7 | 0111 | 21 | 07 | 7 |
| 8 | 1000 | 22 | 10 | 8 |
| 9 | 1001 | 100 | 11 | 9 |
| 10 | 1010 | 101 | 12 | A |
| 11 | 1011 | 102 | 13 | B |
| 12 | 1100 | 110 | 14 | C |
| 13 | 1101 | 111 | 15 | D |
| 14 | 1110 | 112 | 16 | E |
| 15 | 1111 | 120 | 17 | F |

Number System

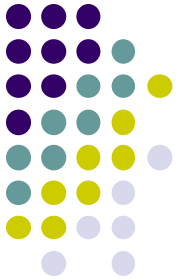
# อธิบาย



- ระบบเลขฐานต่างๆ

- ระบบเลขฐานสิบ เป็นเลขที่ที่มีการนำไปใช้แทนจำนวนคน สัตว์ สิ่งของ เครื่องใช้ ฯลฯ ในโลกโดยที่มีค่าแทนสิ่งเหล่านี้ ด้วยจำนวนเลขที่ใช้แทนค่าที่ไม่เหมือนกัน 10 ค่า คือ ค่า 0,1,2...9 หากจะแทนค่าเพิ่มขึ้นอีกนอกเหนือจากนี้ ก็ใช้วิธีการทดเป็นหลัก 10 มาทางซ้ายมือ จากนั้นก็แทนค่าในหลักสิบจนครบ 00-99 ก็เพิ่มมาเป็นหลักร้อยทางด้านซ้ายมือแทนค่า 100-999 การทดเป็นอย่างนี้ไปจนถึงค่าที่ต้องการ
- ระบบเลขฐานสอง เป็นเลขที่นำมาใช้แทนค่าในระบบดิจิตอลที่จะนำมาทดแทนการทำงานของระบบแอนะลอกในปัจจุบัน สามารถแทนค่าได้ตามจำนวนฐาน 2 ค่า คือ ค่า 0,1 หากต้องการทดแทนมากกว่า 2 ค่าก็ใช้หลักการทดไปทางด้านซ้ายมือเป็น 10, 11,100,101, 110 ,111,1000……1111และก็ทดไปยังหลักต่อไป
- ระบบเลขฐานสามมีค่าเลขหลักคือ 0,1,2 3ค่าเป็นเลขฐาน เมื่อมีค่ามากกว่า 3 ก็จะทดมาหลักทางซ้ายมือเป็น 10,11,12 จากนั้นเป็น 20,21,22 , 100,101,102, 110, 111,112, 120, 121,122, 200,201,202……..

# อธิบาย(ต่อ)



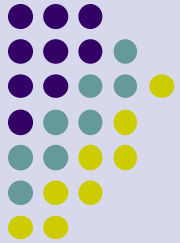
- ในทำนองเดียวกัน

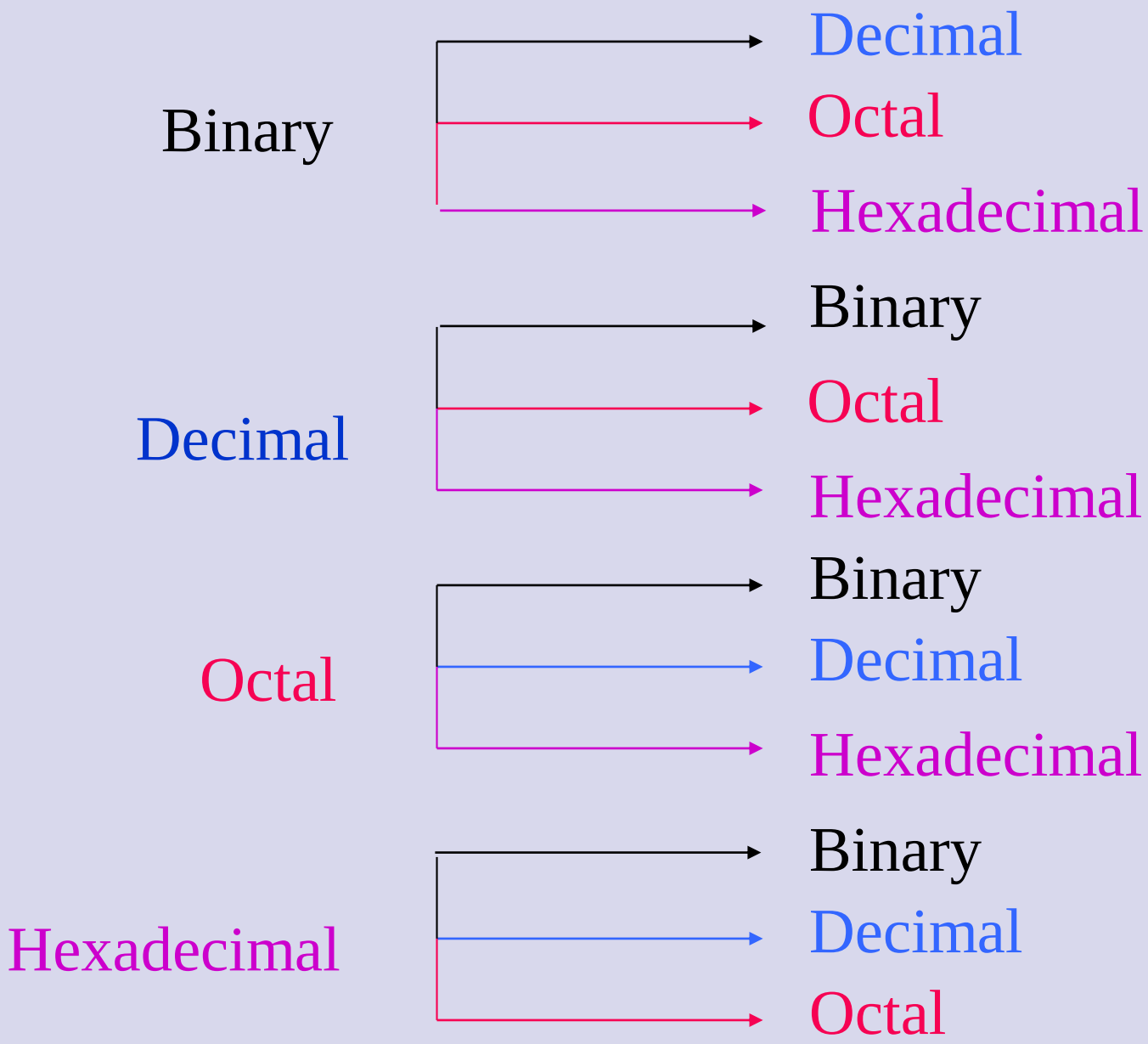
-ระบบเลขฐานแปด มีค่าประจำหลัก8 ค่า คือ ค่า 0,1,2,3,4,5,6,7 หากต้องการแทนค่ามากกว่านี้จะทำการทดไปทางซ้ายมือ คือ 10,11,12,13,14,15,16, 17,20,21,22,23,24….27….77,100…777,1000 อย่างนี้ไปจนถึงค่าที่ต้องการ

- ระบบเลขฐานสิบหกมีค่าประจำหลักที่ไม่ซ้ำกันอยู่จำนวน16 ค่า ระบบฐานสิบหกนี้มีค่าเกินจากเลขฐาน10 อยู่จำนวน 6 ค่าๆที่เกิดขึ้นเกินจากค่าของเลขฐาน 10 จะกำหนดให้เป็นตัวอักษรที่ไม่ซ้ำกับเลขฐาน 10 คือ A , B, C, D, E, F

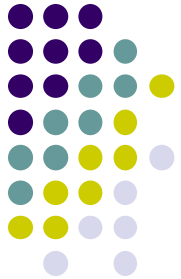
- หากต้องการแสดงค่ามากกว่านี้ก็ใช้หลักการทดไปทางซ้ายเช่น 10,11,12……1F, 20…….2F,30….3F ดังตารางแสดงค่าความสัมพันธ์ เลข10 ของฐาน10 มีค่าเท่ากับ 1010 ฐาน 2 , 12 ของฐาน 8, A ของฐาน16

# ค่าความสัมพันธ์ของเลขฐานต่างๆที่นำมาใช้ในวงจรดิจิตอล





# อธิบาย



**หลักการแปลงเลขฐานต่างๆ**

- จากรูปแสดงให้เห็นถึงความสัมพันธ์ของฐานเลขต่างๆ ที่สามารถหาค่าจากฐานสอง ไปยังฐานสิบ , ฐานแปด , ฐานสิบหกโดยตรงได้
- หากมีเลขฐานสิบ ก็สามารถแปลงเป็นฐานสอง ฐานแปด ฐานสิบหกโดยตรง
- ฐานแปดก็สามารถแปลงเป็นฐานสอง ฐานสิบ ฐานสิบหก
- ในทำนองเดียวกันหากมีเลขฐานสิบหก ก็แปลงเป็นฐานสอง ฐานสิบ ฐานแปดได้ เช่นกัน ในบทนี้จะทำการแปลงความสัมพันธ์ของเลขฐานต่างๆ ไว้อย่างละเอียด

# 13 การแปลงเลขฐานสองเป็นเลขฐานต่างๆ

131 หลักการแปลงเลขฐานสองเป็นฐานสิบ

132 หลักการแปลงเลขฐานสองเป็นฐานแปด

133 หลักการแปลงเลขฐานสองเป็นฐานสิบหก

# การแปลงเลขฐานใดๆ เป็นเลขฐานสิบ

## ใช้หลักเกณฑ์ดังนี้คือ

$$N = d_n R^n + \ldots + d_3 R^3 + d_2 R^2 + d_1 R^1 + d_0 R^0$$

ในที่นี้ R = ค่าของฐานใดๆ เช่น 2,3,8,16

$\therefore$

d = จำนวนค่าของเลขฐานนั้น

N = จำนวนเลขฐาน10

# อธิบาย

- การแปลงเลขฐานใดๆ ให้เป็นฐานสิบมีหลักเกณฑ์ดังนี้

ให้นำค่าที่จะหาคูณด้วยค่าประจำหลักของฐานที่จะหาค่านำผลคูณทั้งหมดมาบวกกันก็จะเป็นค่าของเลขฐานสิบสามารถเขียนเป็นสมการในรูป

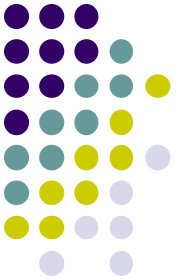
- ใช้สูตรคำณวนค่าดังรูปสไลด์

ค่า N = เลขฐานสิบ

R = ค่าน้ำหนักของเลขฐานนั้น มีค่าเท่ากับเลขสูงสุดของฐานยกกำลังตามหลัก

d = ค่าเลขที่ใดๆ ที่ต้องการหาค่า

# 131 หลักการแปลงเลขฐานสองเป็นฐานสิบ



R = ค่าของฐานใดๆ เช่นในที่นี้มีค่าเท่ากับ 2

N= ค่าที่บอกถึงตำแหน่งน้ำหนักประจำซึ่งจะเป็นเลขที่ฐานยกกำลัง$(R)^n$

d = จำนวนค่าที่อยู่ตำแหน่งของเลขฐาน2, มีค่า 1 กับ 0 N = จำนวนเลขฐาน10

$$N = d_n R^n + \ldots + d_3 R^3 + d_2 R^2 + d_1 R^1 + d_0 R^0$$

**สมการที่ 1**

ในที่นี้ $R^3 = 8$ , $R^2 = 4$ , $R^1 = 2$ , $R^0 = 1$ ; $R = 2$

$\therefore$ จะได้ $N = \ldots + 8d_3 + 4d_2 + 2d_1 + d_0$

ตัวอย่างการนำไปใช้กับฐานใดๆ เช่น หลักการแปลงเลขฐานสามเป็นฐานสิบ

ใช้หลักเกณฑ์คือ

R= มีค่าเท่ากับ 3

d= จำนวนค่าที่อยู่ในตำแหน่งของเลขฐาน 3

# อธิบาย

- การแปลงเลขฐานสามเป็นฐานสิบ

-ใช้สมการที่1 แทนค่าดังนี้ ค่า R มีค่าเท่ากับ 3 ค่าน้ำหนักประจำตำแหน่งหาโดย

-ในตำแหน่งต่างๆ สามารถแทนค่า R ดังนี้

$2^N = X$

$3^4 = 81$ ตำแหน่งที่ 4 ของเลขฐาน 3 มีค่าน้ำหนักเท่ากับ 81

$3^3 = 27$ ตำแหน่งที่ 3 ของเลขฐาน 3 มีค่าน้ำหนักเท่ากับ 27

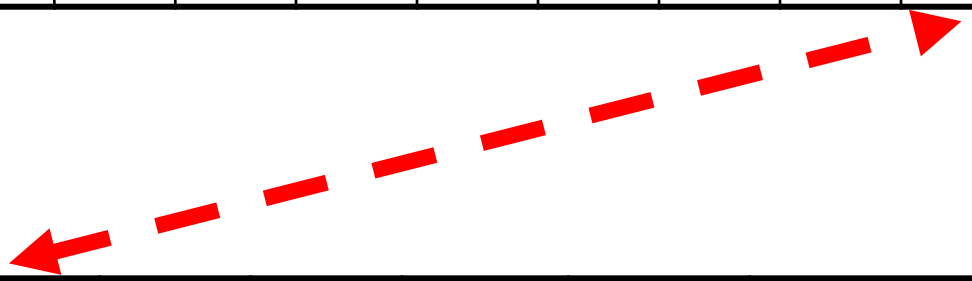
$3^2 = 9$ ตำแหน่งที่ 2 ของเลขฐาน 3 มีค่าน้ำหนักเท่ากับ 9

$3^1 = 3$ ตำแหน่งที่ 1 ของเลขฐาน 3 มีค่าน้ำหนักเท่ากับ 3

$3^0 = 1$ ตำแหน่งที่เริ่มต้นของเลขฐาน 3 มีค่าน้ำหนักเท่ากับ 1

# ตารางแสดงค่าของจำนวนบิตในแต่ละฐาน

| บิตที่ | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| น้ำหนัก | $2^8$ | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| ค่าน้ำหนัก | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |



| บิตที่ | 0 | -1 | -2 | -3 | -4 | -5 | -6 |
|---|---|---|---|---|---|---|---|
| น้ำหนัก | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ |
| ค่าน้ำหนัก | 1 | 0.5 | 0.25 | 0.125 | 0.0625 | 0.03125 | 0.015625 |

# อธิบาย

- จากการแทนค่า R ที่ผ่านมา ในตำแหน่งต่างๆ สามารถสรุปออกเป็นค่าน้ำหนักของเลขดังตารางซึ่งแสดงจากค่า -n ถึงค่า +n (-6 ถึง +8) ก็จะได้ค่าน้ำหนักทั้งหมด หากพิจารณาจากตารางสังเกตุตัวเลขค่าน้ำหนักที่ ตำแหน่ง 0 ค่าที่เกิดขึ้นสามารถหาได้อย่างง่าย โดยดูจากตำแหน่งที่ n มีค่าเป็นบวกค่าน้ำหนักที่เพิ่มขึ้นแต่ละตำแหน่งเป็น 2 เท่า (คูณ2) ส่วนในตำแหน่งที่ n มีค่าเป็นลบค่าน้ำหนักแต่ละตำแหน่งจะลดลง 2 เท่า (หาร2)ในส่วนนี้ให้นักศึกษาจำไว้ เพื่อนำ ไปแปลงค่าตัวเลขในเฟรมถัดไป

# การแปลงเลขฐานสองเป็นฐานสิบ

## ตัวอย่างที่ 1.1

วิธีทำ $(1110101.110101)_2$ เป็น $(\ldots\ldots)_{10}$

$= (1\times64) + (1\times32) + (1\times16) + (0\times8) + (1\times4) + (0\times2) + (1\times1) + (1\times0.5) +$

$(1\times0.25) + (0\times0.125) + (1\times0.0625) + (0\times0.03125) + (1\times0.015625)$

$= 64 + 32 + 16 + 0 + 4 + 0 + 1 + 0.5 + 0.25 + 0 + 0.0625 + 0 + 0.015625$

จะได้ $= 117.828125$

# อธิบาย

- ตัวอย่างที่ 1.1 วิธีแปลงเลขฐานสอง เป็นฐานสิบ

  - ใช้สูตรในการคำนวณ

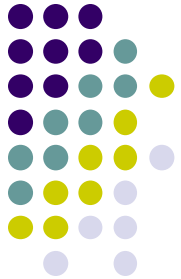
  - ค่าn มีค่าเท่ากับ -6 ถึง +6

  - ค่าR มีค่าเท่ากับ 2

  - ค่า d แทนค่าในตำแหน่ง-1=1, ตำแหน่งที่ -2=1, ตำแหน่งที่ -3=0, ตำแหน่งที่ -4=1, ตำแหน่งที่ -5=0, ตำแหน่งที่ -6=1 ,ตำแหน่งที่ 0=1,ตำแหน่งที่1=0,ตำแหน่งที่2=1 ตำแหน่งที่ 3=0, ตำแหน่งที่ 4=1, ตำแหน่งที่ 5=1, ตำแหน่งที่ 6=1

  แทนค่าทั้งหมดจะได้ $(117.828125)_{10}$

# ตัวอย่างการใช้สูตร



- การแปลงเลขฐานสองเป็นเลขฐานสิบ

  **โจทย์** $(1011)_2 = (\ldots\ldots)_{10}$

  **วิธีทำ** $N = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

  $= 8 + 0 + 2 + 1$

  $(1011)_2 = (11)_{10}$

# อธิบาย

- ตัวอย่างการใช้สูตร ในการแปลงเลขฐานสองเป็นฐานสิบที่มีเลขเป็นจำนวนเต็ม

- ค่า n เป็นบวกมีค่าตั้งแต่ 0 ถึง 3

- ค่า R เป็นเลขฐานสองมีค่าเท่ากับ 2

- ค่า d ตำแหน่งที่ 0=1, ตำแหน่งที่ 1=1, ตำแหน่งที่ 2=0, ตำแหน่งที่ 3=1

  ค่าที่ได้เกิดจากตำแหน่งที่ 3 มีค่าเท่ากับ 8 บวกกับค่าตำแหน่งที่ 2 , มีค่าเท่ากับ 0

  บวกกับค่าตำแหน่งที่1 มีค่าเท่ากับ 2 บวกกับค่าตำแหน่งที่ 0 มีค่าเท่ากับ 1

- รวมค่าทั้งหมดเป็นเลข 11 ฐานสิบ

**Binary** ⟶ **Decimal**

# ตัวอย่างที่ 1.2

$$(111.11)_2 = (\ldots\ldots)_{10}$$

$$1 * 2^2 + 1 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 1 * 2^{-2}$$

$$4 + 2 + 1 + 0.5 + 0.25$$

$$= 7.75$$

$$(111.11)_2 = (7.75)_{10}$$

# อธิบาย

- จากตัวอย่างที่ 1.2 แปลงค่าเลขฐานสองเป็นฐานสิบแบบมีหลักทศนิยม

  - n มีค่าเท่ากับ -2 ถึง +2

  - R มีค่าเท่ากับ 2

  - ค่า d แทนค่าในตำแหน่ง -1 = 1,ตำแหน่งที่ -2 = 1, ตำแหน่งที่ 0 = 1, ตำแหน่งที่ 1 = 1, ตำแหน่งที่ 2 = 1

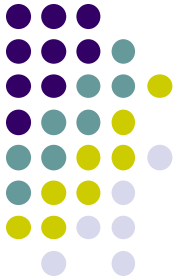
  - ค่าที่ได้เกิดจากตำแหน่ง $d_2$ เท่ากับ 4บวกกับ ตำแหน่ง $d_1$ เท่ากับ 2 บวกกับตำแหน่ง $d_0$ เท่ากับ 1บวกกับตำแหน่ง $d_{-1}$เท่ากับ0.5บวกกับตำแหน่ง $d_{-2}$เท่ากับ 0.25 ค่า N มีค่าเท่ากับ 7.75 ฐานสิบ

# การแปลงเลขฐาน3 เป็นฐาน 10
# เพื่อเป็นตัวอย่างในการวิเคราะห์เลขฐานอื่นได้

ตัวอย่างที่1.3

- กำหนดค่า $(1202)_3$, $(112)_3$ $=(\ldots N\ldots.)_{10}$
- $N=(1x3^3+2x3^2+0x3^1+2x3^0)$
- $N=(47)_{10}$
- $(112)_3=(\ldots N\ldots)_{10}$
- $N=(1x3^2+1x3^1+2x3^0)$
- $N=(14)_{10}$
- พิจารณาจากตารางจะเห็นว่ามีค่าเท่ากัน$(112)_3=14$

# อธิบาย



- จากตัวอย่างที่ 1.3
- นำสมการที่1 สามารถนำไปประยุกต์ได้หลายฐาน ให้นักศึกษาลองทำความเข้าใจดูดังตัวอย่างที่1.3
- ในที่นี้ขอยกตัวอย่างของเลขฐาน3
- - ค่าR เป็นเลขฐาน 3 มีค่าเป็น 3
- - ค่าd บอกถึงตำแหน่งที่ 0 มีค่าเป็น2 ,
- ตำแหน่งที่ 1 มีค่าเป็น0 ,
- ตำแหน่งที่ 2 มีค่าเป็น2 ,
- ตำแหน่งที่ 3 มีค่าเป็น1
- ค่า n บอกถึงตำแหน่งน้ำหนักประจำซึ่งจะเป็นเลขที่ฐานยกกำลัง $(R)^n$

## 132 หลักการแปลงเลขฐานสองเป็นฐานแปด

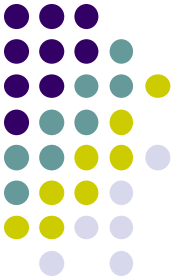
$(1110101.110101)_2$ เป็น $(\ldots\ldots)_8$

วิธีทำ แบ่งบิตออกให้เป็นฐาน 8 คือบิตจะมี 3 หลัก จะได้

$$(001,110,101.110,101)_2$$

$$= (165.65)_8$$

# อธิบาย

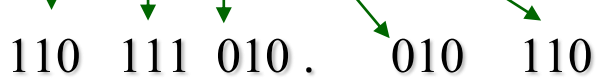


- หลักการแปลงเลขฐานสองเป็นเลขฐานแปด
  - พิจารณาที่จุดทศนิยมเป็นหลัก
  - แบ่งตัวเลขด้านหน้าจุดทศนิยมไปทางซ้ายมือทีละ 3 บิต
  - แบ่งตัวเลขหลังจุดทศนิยมทีละ 3 บิต นำไปหาค่าก็จะได้เป็นค่าเลขฐานแปดหลังจุดทศนิยมจำนวน 2บิต
  - ค่าที่เกิดขึ้นเป็นเลขฐานแปดคือ $(165.65)_8$

**Binary** ⟶ **Octal**

ตัวอย่างที่ 1.4

- $(\underline{110}\underline{111}010.010110)_2 = (\ldots\ldots)_8$



110 111 010 . 010 110

110 = 6

111 = 7

010 = 2

010 = 2

110 = 6

$(110111010.010110)_2 = (672.26)_8$

# อธิบาย

- **ตัวอย่างที่ 1.4** **เป็นการแปลงเลขฐานสองเป็นฐานแปด**

- จำนวนบิตที่หน้าจุดทศนิยมเป็นจำนวน 9 บิต และหลังจุดทศนิยมจำนวน 6 บิต

- หาค่าเป็นเลขฐาน 8 หน้าจุดทศนิยมเป็นจำนวน 3 บิต

- ส่วนค่าที่เกิดหลังจุดทศนิยม6 บิต สามารถแปลงเป็นเลขฐานแปดได้จำนวน 2 บิต

- ค่าที่เกิดขึ้นคือ $(672.26)_8$ แสดงในสไลด์

# 133 การแปลงเลขฐานสองเป็นฐานสิบหก

$(0011111100001101.1000\ 1100)_2 = (\ldots\ldots)_{16}$

0011 1111 0000 1101 . 1000 1100

0011 = 3

1111 = F

0000 = 0

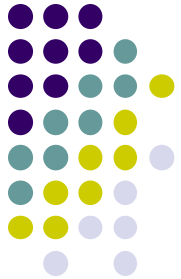
1101 = D

1000 = 8

1100 = C

$(0011111100001101.10001100)_2 = (3F0D.8C)_{16}$

# อธิบาย



- วิธีการแปลงเลขฐานสองเป็นเลขฐานสิบหก จำนวน 24 บิต

- พิจารณาจำนวนเต็มมี16 หลัก จำนวนที่อยู่หลังจุดทศนิยมมีจำนวน 8หลัก

- จัดกลุ่มตัวเลขหน้าจุดทศนิยมไปทางซ้ายทีละ 4บิต

- จัดกลุ่มตัวเลขหลังจุดทศนิยมไปทางขวามือทีละ 4บิต

- เลขฐานสิบหกที่เกิดขึ้นเป็นจำนวนเต็ม 4 บิต และจำนวนทศนิยมจำนวน 2บิต

- หาค่าเลขที่ถูกแบ่งขนาด 4บิต เป็นเลขฐานสิบหก

- คำตอบที่ได้เป็นเลขฐานสิบหกจำนวน 6บิตคือ $(3F0D.8C)_{16}$

# การแปลงเลขฐานสองเป็นเลขฐานสิบและฐานสิบหก

**ตัวอย่างที่ 1.5** แปลงเลขฐานสองให้อยู่ในฐานที่กำหนด

$$(1101)_2 \longrightarrow (\ldots ?\ldots)_{16}$$

**วิธีทำ** $(1101)_2 \longrightarrow (\ldots\ldots)_{10} \longrightarrow (\ldots\ldots)_{16}$

$$(1101)_2 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

$$= 8 + 4 + 0 + 1$$

$$= (13)_{10}$$

$\therefore (13)_{10} = (\ldots\ldots)_{16}$

$16\overline{)13}$ เศษ $D$

$0$

$\therefore (13)_{10} = (D)_{16}$

# อธิบาย

- ตัวอย่างที่ 1.5 แสดงวิธีการแปลงเลขฐานสองเป็นฐานสิบและสิบหก

  - กำหนดให้ $(1101)_2$ แปลงเป็นเลขฐานสิบ โดยการใช้สมการที่ 1 ค่าที่เกิดขึ้นเท่ากับ $(13)_{10}$

  - นำค่าเลข (13) ของเลขฐานสิบมาแปลงเป็นเลขฐานสิบหก โดยเอาค่า16 ไปหาร(13) ได้ 0 เศษ 13 ตรงกับอักษร D ในเลขฐานสิบหก

  - คำตอบของเลข $(1101)_2$ คือ$(13)_{10}$ , $(D)_{16}$

# ตารางแสดงค่าน้ำหนักของตำแหน่งบิตต่างๆในแต่ละฐาน

| บิตที่ | n | 2 | 1 | 0 | -1 | -2 | -3 |
|---|---|---|---|---|---|---|---|
| ฐาน 8 | $8^n$ | $8^2$ | $8^1$ | $8^0$ | $8^{-1}$ | $8^{-2}$ | $8^{-3}$ |
| ค่าฐาน 8 | $8^n$ | 64 | 8 | 1 | 0.125 | 0.015625 | 0.00195 |
| ฐาน 10 | $10^n$ | $10^2$ | $10^1$ | $10^0$ | $10^{-1}$ | $10^{-2}$ | $10^{-3}$ |
| ค่าฐาน 10 | $10^n$ | 100 | 10 | 1 | 0.1 | 0.01 | 0.001 |
| ฐาน 16 | $16^n$ | $16^2$ | $16^1$ | $16^0$ | $16^{-1}$ | $16^{-2}$ | $16^{-3}$ |
| ค่าฐาน 16 | $16^n$ | 256 | 16 | 1 | 0.0625 | 0.003906 | 0.0002441 |

# อธิบาย

- ตารางแสดงค่าความสัมพันธ์ระหว่างเลขฐานแปด, เลขฐานสิบ, เลขฐานสิบหก

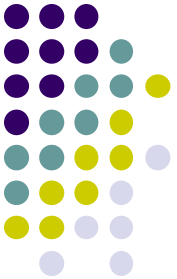
  - แสดงค่าตำแหน่งของเลขฐานแปด ฐานสิบ ฐานสิบหกต่างๆ

  - แสดงถึงค่าน้ำหนักของตัวเลขในบิตต่างของเลขบานแปด ฐานสิบ ฐานสิบหก

  ทั้งเลขจำนวนเต็มและ เลขทศนิยม

  - จากตารางสามารถนำค่าน้ำหนักของฐานไปใช้แทนค่าในสมการที่1 ในการแปลงฐานได้

# 14 การแปลงเลขฐานสิบเป็นเลขฐานต่างๆ



141 หลักการแปลงเลขฐานสิบเป็นฐานสอง

142 หลักการแปลงเลขฐานสิบเป็นฐานแปด

143 หลักการแปลงเลขฐานสิบเป็นฐานสิบหก

# 141 หลักการแปลงเลขฐานสิบเป็นฐานสอง

ตัวอย่างที่ 1.6 **การแปลงเลขฐานสิบเป็นเลขฐานสอง**

$$(26)_{10} = (\ldots ?\ldots)_2$$

**วิธีทำ**

$2\overline{)26}$ เศษ 0

$2\overline{)13}$ เศษ 1

$2\overline{)6}$ เศษ 0

$2\overline{)3}$ เศษ 1

$2\overline{)1}$ เศษ 1

0 1 1 0 1 0

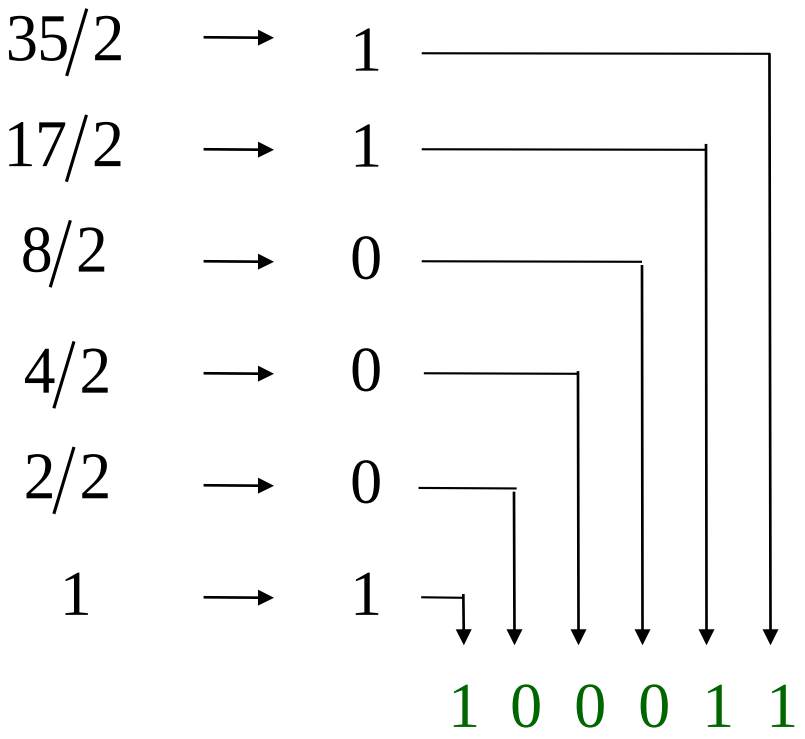
$\therefore \quad (26)_{10} = (11010)_2$

# อธิบาย

- หลักการแปลงเลขฐานสิบเป็นเลขฐานสองในกรณีเลขจำนวนเต็ม
  - นำค่าเลขฐานสิบมาหารด้วยเลข 2 ค่าที่ได้คือผลลัพธ์กับเศษที่เหลือ
  - ผลลัพธ์ที่ได้จะนำไปหารต่อไปส่วนเศษที่เหลือ นำไปเป็นคำตอบไว้ในส่วนขวามือดังรูปตัวอย่างที่ 9
  - ผลลัพธ์จะถูกนำไปหารในระดับต่อไปจนไม่สามารถหารต่อไปได้อี
  - เศษที่เหลือในระดับต่างๆ จะมีค่าเท่ากับ 0 กับ 1
  - คำตอบของเลขฐานจะนำเอาจากส่วนล่างไปหาส่วนบน
  - ค่าคำตอบของเลข $(26)_{10}$ เท่ากับ $(11010)_2$

**Decimal ⟶ Binary**

ตัวอย่างที่ 1.7 $(35)_{10} = (\ldots ?\ldots)_2$




$(35)_{10} = (100011)_2$

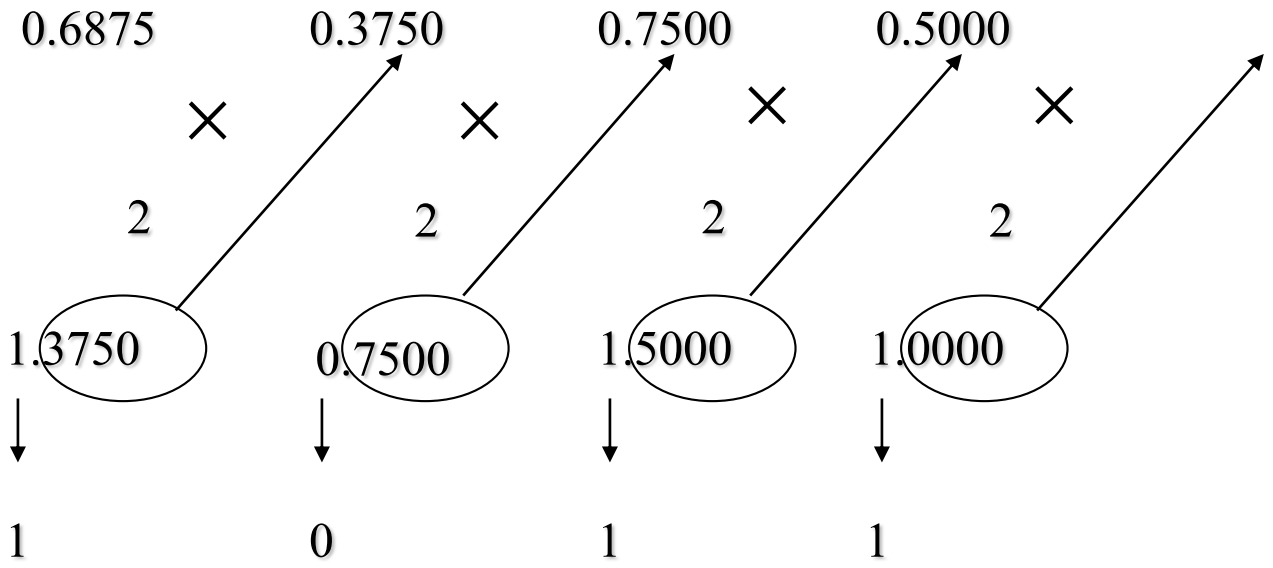
# อธิบาย

- **จากตัวอย่างที่ 1.7**

  - กำหนดเลข 35 ของฐานสิบแปลงเป็นเลขฐานสอง
  - นำเลข 35 หาร 2 ได้ผลลัพธ์ 17 เหลือเศษ 1
  - นำเลข 17 หาร 2 ได้ผลลัพธ์ 8 เหลือเศษ 1
  - นำเลข 8 หาร 2 ได้ผลลัพธ์ 4 เหลือเศษ 0
  - นำเลข 4 หาร 2 ได้ผลลัพธ์ 2 เหลือเศษ 0
  - นำเลข 2 หาร 2 ได้ผลลัพธ์ 1 เหลือเศษ 0
  - นำเลข 1 หาร 2 ได้ผลลัพธ์ 0 เหลือเศษ 1
  - ค่าผลลัพธ์สุดท้ายคือเลข 0 ไม่สามารถนำไปหารได้อีก
  - ค่าคำตอบจะไล่เศษจากระดับล่างไปบนคือ $(100011)_2$

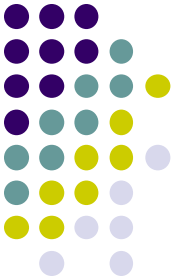
# การแปลงเลขทศนิยมจากฐานสิบเป็นฐานสอง

**ตัวอย่างที่ 1.8** $(0.6875)_{10} = (\ldots?\ldots)_2$




$(0.6875)_{10} = (0.1011)_2$

# อธิบาย



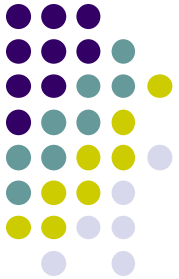
**การแปลงเลขฐานสิบเป็นฐานสองในส่วนทศนิยม**

- นำค่าหลังจุดทศนิยมมาคูณด้วยเลข 2
- ค่าที่เกิดขึ้นจะมี2 ส่วน คือ ส่วนของเลขจำนวนเต็มนำไปเป็นคำตอบหลังจุดทศนิยมหลักที่ 1
- ส่วนค่าหลังจุดทศนิยมก็นำไปคูณด้วย 2 จะเป็นคำตอบในหลักทศนิยมตำแหน่งที่ 2 ค่าที่เกิดขึ้นก็จะมีอีกสองส่วน คือ จำนวนเต็มเป็นคำตอบหลักที่ 2
- ส่วนค่าหลังจุดทศนิยมก็นำไปคูณด้วย 2 จะเป็นคำตอบในหลักทศนิยมตำแหน่งที่ 3
- ทำอย่างนี้ไปจนถึงจำนวนค่าตำแหน่งของทศนิยมที่ต้องการ

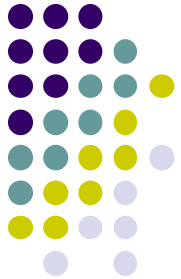
# อธิบาย(ต่อ)

- **ส่วนของเลขจุดทศนิยมฐานสิบแปลงเป็นฐานสองพิจารณาดังนี้**
  - นำค่า 0.6875 คูณด้วย 2 ได้ 1.375
  - ผลลัพธ์ที่ได้ค่าหน้าจุดทศนิยมคือคำตอบหลังจุดทศนิยมหลักที่1
  - นำค่า 0. 375 คูณด้วย 2 ได้ 0.75
  - ผลลัพธ์ที่ได้ค่าหน้าจุดทศนิยมคือคำตอบหลังจุดทศนิยมหลักที่2
  - นำค่า 0. 75 คูณด้วย 2 ได้ 1.5
  - ผลลัพธ์ที่ได้ค่าหน้าจุดทศนิยมคือคำตอบหลังจุดทศนิยมหลักที่3
  - นำค่า 0.5 คูณด้วย 2 ได้ 1.00
  - ผลลัพธ์ที่ได้ค่าหน้าจุดทศนิยมคือคำตอบหลังจุดทศนิยมหลักที่4
  - ผลลัพธ์ได้ดังนี้ $(1011)_2$

# 142 หลักการแปลงเลขฐานสิบเป็นฐานแปด



- หลักการแปลงเลขฐานสิบเป็นเลขฐานแปด
  - มีสองส่วนที่เป็นจำนวนเต็มกับทศนิยม
  - หลักการเหมือนกับการแปลงฐานสิบเป็นฐานสองโดยเปลี่ยนค่าเลขที่หารมาเป็นเลข 8
  - ค่าที่ได้เป็นผลลัพธ์กับเศษที่เหลือ
  - ผลลัพธ์นำไปหารต่อ ส่วนเศษที่เหลือจะเป็นคำตอบในหลักถัดไป ในส่วนที่เป็นทศนิยมให้นำไปคูณกับเลข 8
  - จำนวนเต็มหน้าทศนิยมเป็นคำตอบ
  - หลังจุดทศนิยมนำไปคูณเพื่อหาผลลัพธ์หลักที่สองหลังจุดทศนิยมของฐานแปด



**ตัวอย่างที่ 1.9** $(13)_{10} = (\ldots ?\ldots)_8$

$8\overline{)13}$ เศษ 5

$8\overline{)1}$ เศษ 1

0

1 5

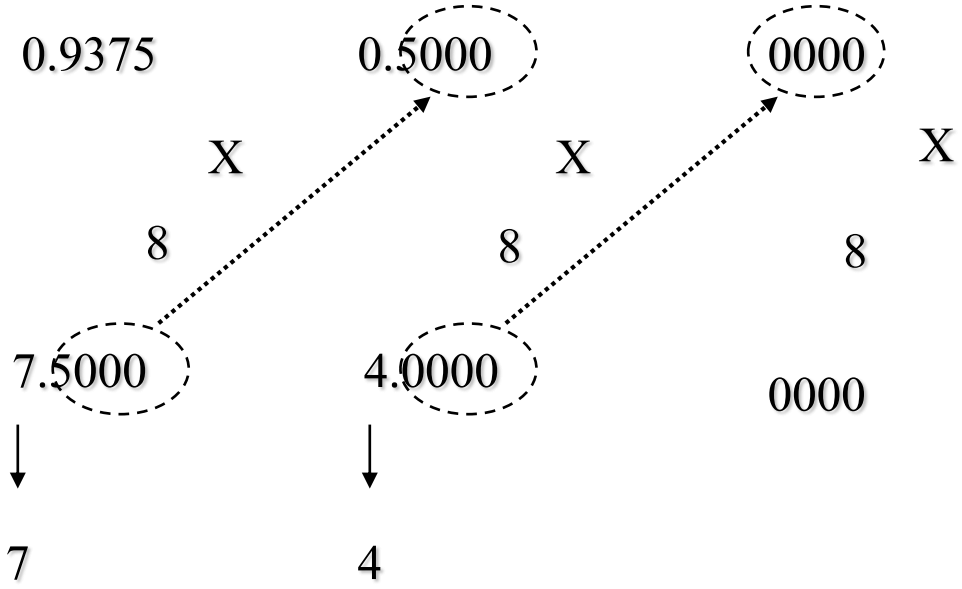
$(13)_{10} = (15)_8$

# อธิบาย

- **ตัวอย่างที่ 1.9 กำหนดเลข 13 ของเลขฐานสิบ แปลงเป็นเลขฐานแปด**
  - นำเลข 13 หารด้วย 8 ผลลัพธ์เป็น 1 เหลือเศษ 5
  - นำผลลัพธ์ 1 หารด้วย 8 ผลลัพธ์เป็น 0 เหลือเศษ 1
  - คำตอบให้นำค่าเศษจากผลที่ได้สุดท้ายเรียงไปยังเศษต่อไปในที่นี้ก็จะได้ $(15)_8$ จะมีค่าเท่ากับ 13 ของเลขฐาน 10

# การแปลงเลขฐานสิบเป็นฐานแปดในส่วนของทศนิยม

**ตัวอย่างที่ 1.10** $(0.9375)_{10} = (\ldots?\ldots)_8$




$(0.9375)_{10} = (0.74)_8$

# อธิบาย

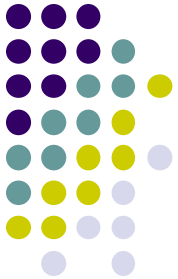
- การแปลงเลขฐานสิบเป็นเลขฐานแปดในหลักทศนิยม

  - นำค่าเลข 0.9375 คูณด้วย 8 ค่าที่ได้จำนวนเต็มเป็นค่าตำแหน่งแรกของเลขฐานสิบ หลักนี้คือเลขฐาน 7

  - จากนั้นนำเลขที่เป็นจุดทศนิยมค่า 0.500 คูณด้วย 8 ได้ค่าจำนวนเต็มคือ 4 ซึ่งเราจะนำมาเป็นหลักที่สองของเลขฐานสิบ

  - คำตอบของค่าที่ได้จะมีค่าเท่ากับ $(0.74)_8$

# 143 หลักการแปลงเลขฐานสิบเป็นฐานสิบหก



**ตัวอย่างที่ 1.11** $(87)_{10} = (\ldots ?\ldots)_{16}$

16 )87 → 7

16 )5 → 5

0

5 7

$(87)_{10} = (57)_{16}$

# อธิบาย

- **ตัวอย่างที่1.11 แสดงการแปลงเลขฐานสิบเป็นฐานสิบหก**
  - วิธีคิดเหมือนเลขฐานสิบ
  - นำค่า 87 มาหารด้วย 16 ค่าที่เกิดขึ้นเท่ากับ 5 เหลือเศษ 7
  - นำค่า 7 เป็นคำตอบของหลัก
  - เศษ 5 นำไปหารด้วย 16 ค่าที่เกิดขึ้นเท่ากับ 0 เหลือเศษ 5

## Decimal ⟶ Hexadecimal

$(259.6875)_{10}$ เป็น $(\ldots?\ldots)_{16}$

**วิธีทำ** นำเลขจำนวนเต็มหารด้วย 16แล้วเลขจุดทศนิยมนำ 16 มาคูณจะได้

$259 \div 16 = 16$ เศษ 3 $(L_{SB})$

$16 \div 16 = 1$ เศษ 0

$1 \div 16 = 0$ เศษ 1 $(M_{SB})$

$0.6875 \times 16 = 11$ หรือ B

จะได้ $(259.6875)_{10} = (103.B)_{16}$

# อธิบาย

- **การแปลงเลขฐานสิบเป็นเลขฐานสิบหก ในส่วนจำนวนเต็ม และทศนิยม**
  - นำเลขจำนวนเต็มหารด้วยเลขฐานที่จะหาในที่นี้คือเลขฐานสิบหก
  - ขั้นที่ 1 นำค่า 259 มาหารด้วย 16 ได้ผลลัพธ์ 16 เหลือเศษ 3
  - ขั้นที่ 2 นำผลลัพธ์ 16 มาหารด้วย 16 ได้ค่า 1 เหลือเศษ 0
  - ขั้นที่ 3 นำผลลัพธ์ 1 มาหารด้วย 16 ได้ค่า 0 เหลือเศษ 1
  - คำตอบที่ได้ในการหาค่าของเลขจำนวนเต็ม จะนำเศษที่เหลือจากการหาร ไล่จากการหารครั้งสุดท้ายไปหาเศษที่ได้ครั้งแรก
  - ในส่วนของทศนิยมของเลขฐานสิบให้นำค่าทศนิยมไปคูณด้วยเลขฐาน16 ค่าที่เกิดคือ เลข 11 หรือเท่ากับ B
  - คำตอบครั้งสุดท้ายทั้งหมดทั้งจำนวนเต็มและทศนิยมมารวมกันคือ $(103.B)_{16}$

# 15 การแปลงเลขฐานแปดเป็นเลขฐานต่างๆ

151 หลักการแปลงเลขฐานแปดเป็น ฐานสอง

152 หลักการแปลงเลขฐานแปดเป็นฐานสิบ

153 หลักการแปลงเลขฐานแปดเป็นฐานสิบหก

# 151 หลักการการแปลงเลขฐาน 8 เป็นฐาน 2

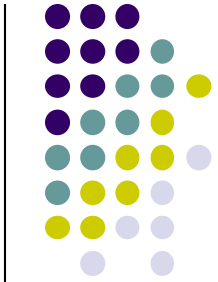
$(346.375)_8$ เป็น $(\ldots?\ldots)_2$

**วิธีทำ** ทำการแปลงฐาน8 เป็นฐาน 2 โดยแบ่งเป็น 3 บิตจะได้คค

$= 011/100/110/.011/111/101$

จะได้ $(346.375)_8 = (011100110.011111101)_2$

# อธิบาย



- การแปลงเลขฐานแปดเป็นเลขฐานสอง ในหลักจำนวนเต็มและทศนิยม
  - พิจารณาส่วนของจำนวนเต็มให้แบ่งแต่ละบิตของเลขฐานแปด สามารถแบ่งได้เป็น3 บิต คือเลข 3, 4, 6
  - ในที่นี้เลข 3ของฐานแปดมีค่าเท่ากับ 011 ค่าของเลข4 มีค่าเท่ากับ 100 และค่าสุดท้ายของจำนวนเต็มคือเลข6 ซึ่งมีค่าเท่ากับ 110
  - พิจารณาในส่วนที่เป็นเลขทศนิยม คือเลข จุด 3, 7, 5 แปลงเป็นเลขฐานสอง คือ 011, 111, 101
  - คำตอบที่ได้คือค่าเลขฐานสองนำมาจัดเรียงรวมกัน $(011100110.011111101)_2$

# 152 หลักการการแปลงเลขฐาน 8 เป็นฐาน 10

**Octal ⟶ Decimal**

**ตัวอย่างที่ 1.12** $(374.24)_8 = (\ldots ?\ldots)_{10}$

$$3 \times 8^2 + 7 \times 8^1 + 4 \times 8^0 + 2 \times 8^{-1} + 4 \times 8^{-2}$$

$$192 + 56 + 4 + 0.25 + 0.0625$$

$$= 252.3125$$

$$(374.24)_8 = (252.3125)_{10}$$

# อธิบาย

- **ตัวอย่างที่ 1.12 การแปลงเลขฐานแปดให้เป็นเลขฐานสิบ**
  - ใช้สูตรดังสมการที่1คำนวณหาค่าโดยมีค่า R=8 , n มีค่าตั้ง -2 ถึง 2
  - ค่าสัมประสิทธิของตัวอย่างนี้คือ 3, 7, 4, จุด 2, 4 เป็นตัวคูณในตำแหน่งที่แสดงในสไลด์
  - การหาค่านี้กำหนดง่ายๆ โดยนำค่าสัมประสิทธิคูณกับค่าน้ำหนักของตำแหน่งในตารางที่กล่าวมา ผลลัพธ์ที่เกิดนำค่าที่คูณได้มาบวกกันก็จะเป็นคำตอบ
  - ค่าผลลัพธ์ที่เกิดขึ้นจะมีค่าเท่ากับ $(252.3125)_{10}$ แสดงในสไลด์

# การแปลงเลขฐาน 8 เป็นฐาน 10

**ตัวอย่างที่ 1.13** $(346.375)_8$ เป็น $(\ldots?\ldots)_{10}$

**วิธีทำ** นำเลขน้ำหนักของบิตมาคูณกับโจทย์จะได้

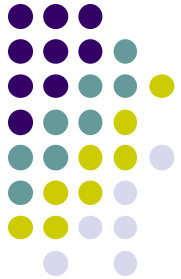
$$= (3 \times 64) + (4 \times 8) + (6 \times 1) + (3 \times 0.125) + (7 \times 0.0156205) + (3 \times 0.001953)$$

$$= 192 + 32 + 6 + 0.375 + 0.109375 + 0.005859$$

$$= 230.490234$$

จะได้ $(346.375)_8 = (230.490234)_{10}$

# อธิบาย



- การแปลงเลขฐานแปดให้เป็นเลขฐานสิบ
  - มีเพียงหลักการเดียวที่เปลี่ยนจากเลขฐานใดๆ เป็นเลขฐานสิบโดยวิธีการใช้สมการที่เปลี่ยนเลขฐาน
  - วิธีการจากหลักการนี้โดยพิจารณาจากตารางแสดงค่าน้ำหนักของบิตในเลขฐานแปด
  - ในตารางน้ำหนักของบิตแต่ละฐานพิจารณาที่ฐานแปดคือค่าตำแหน่งที่ n -2 ถึง 2
  - ค่าที่เกิดขึ้นเป็นดังสไลด์

## 153 หลักการแปลงเลขฐาน 8 เป็นฐาน 16

$(346.375)_8$ เป็น $(\ldots?\ldots)_{16}$

วิธีทำ แปลงฐาน 8 เป็นฐาน 16 โดยการแบ่งเป็นฐาน 2 ก่อน

แล้วทำการแบ่งบิตให้ได้ 4 บิต จะได้

= (0/1110/0110.0111/1110/1000)2

= E6.7E8

จะได้ $(346.375)_8 = (E6.7E8)_{16}$

# อธิบาย

- **การแปลงเลขฐานแปดเป็นฐานสิบหก**

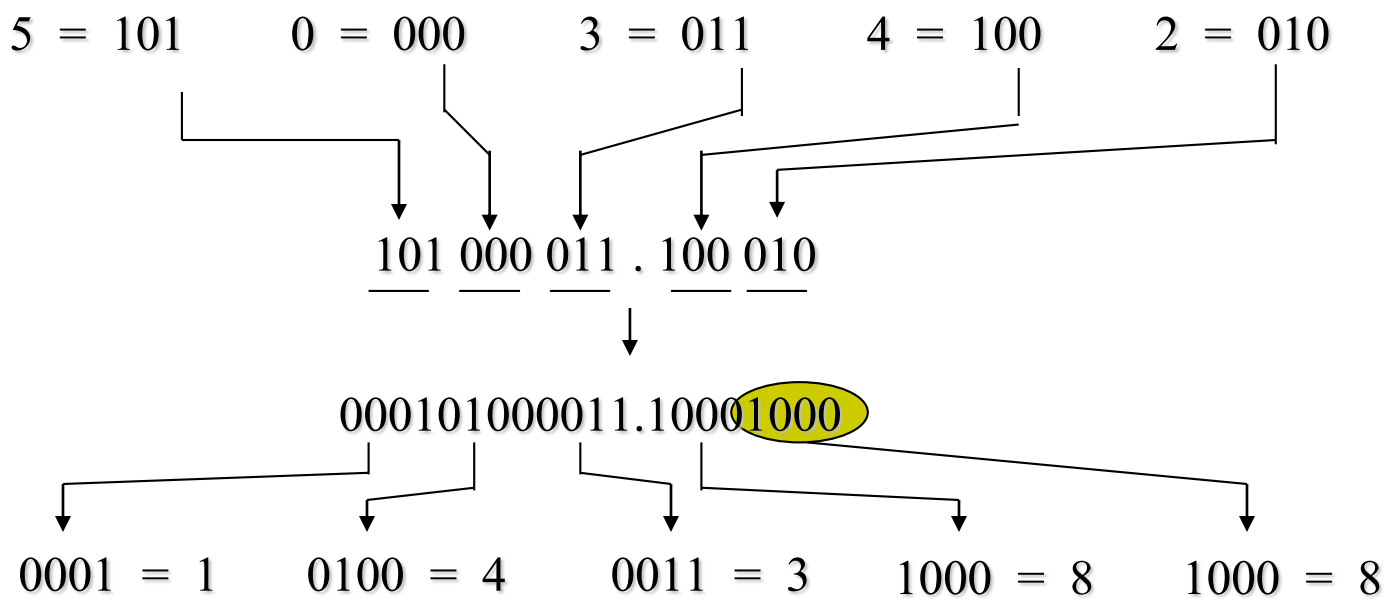
- วิธีการแปลงทำโดยเปลี่ยนค่าเลขฐานแปดเป็นฐานสองจำนวนเต็ม 9บิตและทศนิยมจำนวน 9 บิตเช่นกันดังสไลด์

- ทำการแปลงเป็นเลขฐานสิบหกโดยจัดแบ่งเป็นสี่บิตทางซ้ายหน้าจุดทศนิยมกับการจัดแบ่งหลังจุดทศนิยมไปทางขวามือสี่บิตดังรูปสไลด์

- ค่าที่เกิดขึ้นได้ค่า $(E6.7E8)_{16}$

# Octal ⟶ Hexadecimal

**ตัวอย่างที่ 1. 14** $(503.42)_8 = (\ldots?\ldots)_{16}$




$(504.32)_8 = (143.88)_{16}$

# อธิบาย

- **การแปลงเลขฐานแปดเป็นฐานสิบหก**

  - แปลงเลข 5 ให้เป็นเลขฐานสองคือ 101
  - แปลงเลข 0 ให้เป็นเลขฐานสองคือ 000
  - แปลงเลข 3ให้เป็นเลขฐานสองคือ 011
  - แปลงเลข 4 ให้เป็นเลขฐานสองคือ 100
  - แปลงเลข 2 ให้เป็นเลขฐานสองคือ 010
  - นำเลขฐานสองมาจัดเรียงกันตามเลขฐานแปดจัดกลุ่ม 4 บิตเพื่อนำไปแปลงเป็นเลขฐานสิบหกด้านหน้าจุดทศนิยมเพิ่มจำนวนศูนย์อีก 3 บิตซ้ายสุดค่าก็ยังคงเดิม

    ส่วนทางหลังจุดทศนิยมเพิ่มจำนวนศูนย์อีก 2บิตทางขวาสุด ค่าหลังจุดก็ยังคงเดิมการเพิ่มศูนย์เพื่อนำไปเปรียบเทียบกับค่าฐานสิบหกได้สะดวกขึ้น

# 16 การแปลงเลขฐานสิบหกเป็นเลขฐานต่างๆ

161 หลักการแปลงเลขฐานสิบหกเป็นฐานสอง

162 หลักการแปลงเลขฐานสิบหกเป็นฐานแปด

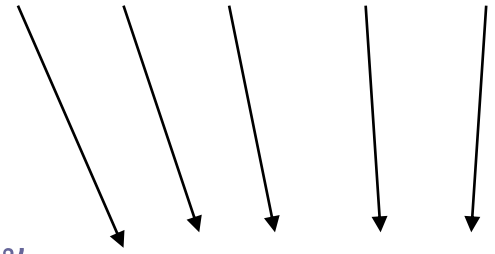
163 หลักการแปลงเลขฐานสิบหกเป็นฐานสิบ

# 161 หลักการแปลงเลขฐาน 16 เป็นฐาน 2

**ตัวอย่างที่ 1.15** $(D3C.2F)_{16}$ เป็น $(.....)_2$

**วิธีทำ** ทำการแปลงเลขฐาน 16 เป็นฐาน 2 โดยการแบ่งบิตเป็น 4 บิต

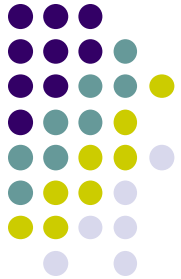
= 1101/0011/1100.0010/1111



จะได้ ( D 3 C . 2 F )$_{16}$ =

**คำตอบ** $(110100111100.00101111)_2$

# อธิบาย



- ตัวอย่างที่ 1.15 การแปลงเลขฐานสิบหกเป็นเลขฐานสอง

  - ในวิธีเดียวกันที่ผ่านมาในการแปลงเลขฐานสองเป็นฐานสิบหก

  - วิธีนี้เลขฐานสิบหก1 บิต ก็สามารถแปลงเป็นเลขฐานสองได้ 4 บิต เช่นกัน

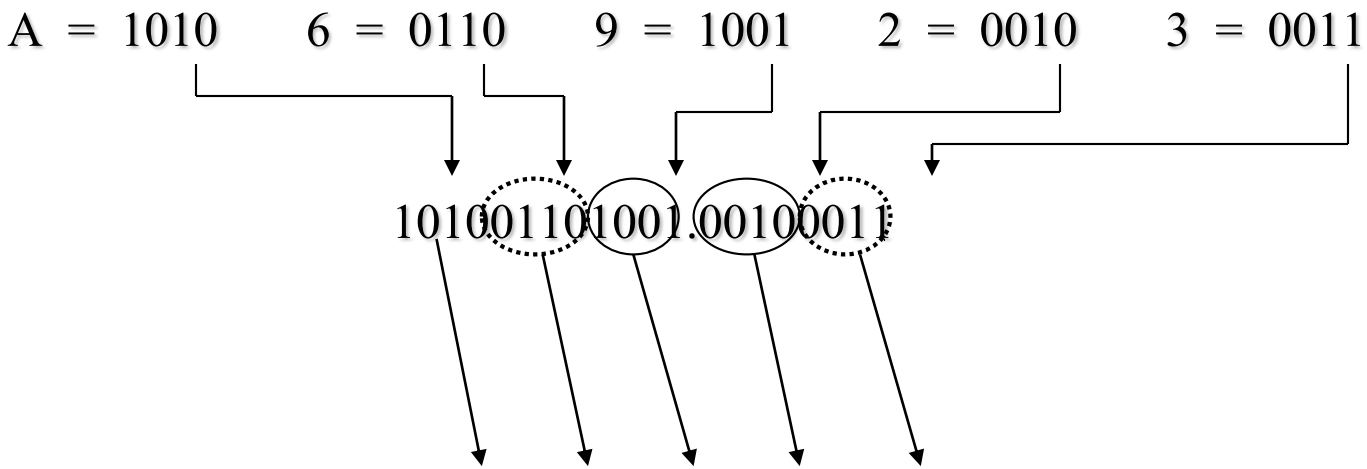
  - เลขฐานฐานสิบหก D =1101,3=0011, C=1100, 2=0010,F=1111

  - จัดเรียงค่าเลขฐานสองตามตำแหน่งเลขฐานสิบหกได้ดังนี้

  $(1101001111100.00101111)_2$

**Hexadecimal ⟶ Binary**

ตัวอย่างที่ 1.16 $(A69.23)_{16} = (\ldots?\ldots)_2$




$(A69.23)_{16} = (1010011010001.00100011)_2$

# อธิบาย

- ตัวอย่างที่1.16 แปลงเลขฐานสิบหกเป็นเลขฐานสอง
  - พิจารณาจากตารางระบบตัวเลขที่ผ่านมาในช่องเลขฐานสิบหกเทียบกับฐานสอง
  - ค่าตำแหน่งอักษร A ฐานสิบหกมีค่าเท่ากับ 1010 ของฐานสอง
    ค่าตำแหน่งอักษร 6 ฐานสิบหกมีค่าเท่ากับ 0110 ของฐานสอง
    ค่าตำแหน่งอักษร 9 ฐานสิบหกมีค่าเท่ากับ 1001 ของฐานสอง
    ค่าตำแหน่งอักษร 2 ฐานสิบหกมีค่าเท่ากับ 0010 ของฐานสอง
    ค่าตำแหน่งอักษร 3 ฐานสิบหกมีค่าเท่ากับ 0011 ของฐานสอง
  - จัดเรียงเลขฐานสองตามตำแหน่งของเลขฐานสิบหกได้ค่าดังนี้

$$(10100110100100100011)_2$$

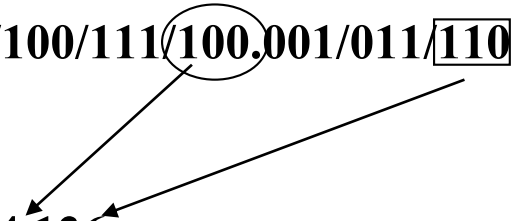
# 162 การแปลงเลขฐาน 16 เป็นฐาน 8

$(D3C.2F)_{16}$ เป็น $(\ldots?\ldots)_8$

**วิธีทำ** ทำการแปลงเป็นฐานสองก่อนแล้วแบ่งบิตเป็น 3 บิตจะได้

= 1101 / 0011/ 1100 .0010 /1111 /0

= 110/100/111/100.001/011/110



= 6474.136

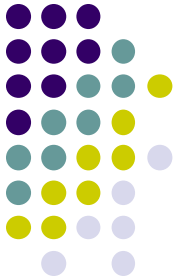
จะได้ $(D3C.2F)_{16} = (6474.136)_8$ #

# อธิบาย

- **การแปลงเลขฐานสิบหกเป็นเลขฐานแปด**
  - ขั้นตอนแรก เพื่อให้หาค่าง่ายโดยการแปลงเป็นเลขฐานสองก่อน 1 บิตของเลขฐานสิบหกจะมีค่าเท่ากับ 4 บิตของเลขฐานสอง
  - ขั้นตอนที่สอง ทำการรวมบิตของเลขฐานสองทั้งหมดเข้าด้วยกัน
  - ขั้นตอนที่สาม ทำการจัดกลุ่มโดยพิจารณาจากจุดทศนิยมเป็นหลักคือแบ่งไปทางซ้ายมือ หน้าจุดทศนิยมทีละ 3 บิต ส่วนหลังจุดทศนิยมให้แบ่งไปทางขวาทีละ 3 บิตจนครบ
  - ขั้นตอนที่สี่ พิจารณาเลขฐานสองที่จัดแบ่งไว้เทียบกับเลขฐานแปดในตารางระบบตัวเลข
  - ค่าคำตอบที่ได้คือ $(6474.136)_8$

# 163 หลักการแปลงเลขฐานสิบหกเป็นฐานสิบ

**Hexadecimal ⟶ Decimal**



**ตัวอย่างที่1.17** การแปลงเลขฐาน 16 เป็นฐาน 10

$(D3C.2F)_{16}$ เป็น $(.....)_{10}$

**วิธีทำ** นำค่าน้ำหนักของแต่ละบิตมาคูณกับโจทย์จะได้

$= (D \times 256) + (3 \times 16) + (C \times 1) + (2 \times 0.0625) + (F \times 0.00390)$

$= 3328 + 48 + 12 + 0.125 + 0.0585$

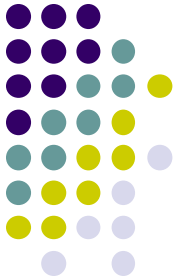
$= 3388.1835$

จะได้ $(D3C.2F)_{16} = (3388.1835)_{10}$

# อธิบาย

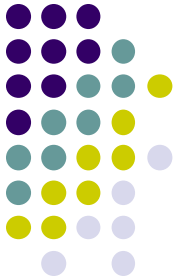
- **ตัวอย่างที่1.17 เป็นการแปลงค่าเลขฐานสิบหกให้เป็นเลขฐานสิบ**
  - ใช้สมการแปลงค่ามีค่าฐานเท่ากับ 16
  - ค่าn มีค่าเท่ากับ -2 ถึง 2
  - ค่าสัมประสิทธิ์คือ D, 3, C, 2, F
  - แทนค่าสมการด้วยค่าต่างๆในสมการ แล้วนำค่าที่ได้นำมารวมกัน
  - คำตอบที่ได้รับ คือ $(3388.18359)_{10}$

# สรุปการแปลงเลขฐานต่างมีหลักเกณฑ์



- เลขฐานใดๆแปลงเป็นฐาน 10 ให้ใช้สูตรแทนค่า
- เลขฐาน10 แปลงเป็นเลขฐานใดๆให้เอาฐานนั้นหารเลขฐาน 10 หากเป็นทศนิยมให้นำไปคูณ
- เลขฐาน 2 แปลงเป็นเลขฐาน8 ให้จัดกลุ่ม 3 บิต
- เลขฐาน 8 แปลงเป็นเลขฐาน2 ให้แยกออกเป็น 3 บิต
- เลขฐาน 2 แปลงเป็นเลขฐาน16 ให้จัดกลุ่ม 4 บิต
- เลขฐาน 16 แปลงเป็นเลขฐาน2 ให้แยกออกเป็น 4 บิต

# ทดสอบความรู้สัปดาห์ที่ 1



- 1.จงแปลงเลขฐาน 10 เป็นฐาน 2 , ฐาน 8, ฐาน16
- (a) 27 (b) 915 (C) 0.375 (d) 0.65 (e)174.25
- (f) 250.8
- 2. จงแปลงเลขฐาน 2 เป็นฐาน 8, ฐาน 10, ฐาน16
- (a)1100 (b) 101110 (C) 0.101 (d)0.01101
- (e)10101.11 (f)10110110.001
- 3.จงแปลงเลขฐาน 8 เป็นฐาน 2 , ฐาน 10, ฐาน16
- (a)65 (b) 371 (C) 240.51 (d)2000
- (e)111111 (f)177777
- 4.จงแปลงเลขฐาน3 เป็นเลขฐาน 2,8,10,16
- (a)012 (b)222 (c)101 (d)2222 (e)11111

# เนื้อหาคำบรรยายในสัปดาห์ที่2 การคำนวณทางคณิตศาสตร์ในระบบดิจิตอล

21หลักการคอมพลีเมนต์ของเลขฐาน

211 การคอมพลีเมนต์ของเลขฐาน2

212 การคอมพลีเมนต์ของเลขฐาน 8

213 การคอมพลีเมนต์ของเลขฐาน10

214 การคอมพลีเมนต์ของเลขฐาน16

22หลักการบวกเลข

221 การบวกเลขฐาน2

222 การบวกเลขฐาน8

223 การบวกเลขฐาน16

23 หลักการลบเลข

231 การลบเลขฐาน2

232 การลบเลขฐาน8

233 การลบเลขฐาน16

24 หลักการคูณเลข

241 การคูณเลขฐาน 2

242 การคูณเลขฐาน 8

243 การคูณเลขฐาน 16

25 หลักการหารเลข

251 หลักการหารเลขฐาน 2

252 หลักการหารเลขฐาน 8

253 หลักการหารเลขฐาน 16

# 21 หลักการคอมพลีเมนต์ ของเลขฐาน

**211หลักการคอมพลีเมนต์ ของเลขฐาน 2**

**ตัวอย่าง 2.1** คอมพลีเมนต์ 1 ของ *1010101*

**วิธีทำ** (1111111-1010101) คอมพลีเมนต์ *1 = 0101010*

**ตัวอย่าง 2.2** คอมพลีเมนต์ 1 ของ *101101110* 1'S

*(111111111-101101110) = 010010001*

**วิธีทำ** คอมพลีเมนต์ 1 ของเลข *101101110* คือ *010010001*

# อธิบาย

**ตัวอย่างที่ 2.1** การหาค่า คอมพลีเมนต์1 ของเลข 1010101
นำค่าสูงสุดที่จะหามาลบด้วยค่าที่ต้องการหา ได้ค่า 0101010
เป็นค่าของคอมพลีเมนต์ 1

**ตัวอย่างที่2.2** นำค่าสูงสุดจะมีจำนวน 9 บิต(111111111) เอาไปลบกับค่าที่ต้องการ
หาคือค่า101101110 ก็จะได้ เป็นคอมพลีเมนต์ 1 คือ 010010001

ข้อสังเกตุ ค่าคอมพลีเมนต์จะมีค่ากลับจากค่าจริงของฐานสองคือ จาก0 เป็น 1 และ 1
เป็น0 ค่าจริง 1010101 ค่าคอมพลีเมนต์ 1 คือค่า 0101010

# 2'S หาจากค่า 1'S

**ตัวอย่าง 2.3** คอมพลีเมนต์ 2 ของ *101101110*

*(111111111-101101110) = 010010001*

**วิธีทำ** คอมพลีเมนต์ 2 = คอมพลีเมนต์ *1 ( +1 )*

*101101110* คอมพลีเมนต์ *1 ( +1 ) = 010010001*

\+ *1*

*010010010*

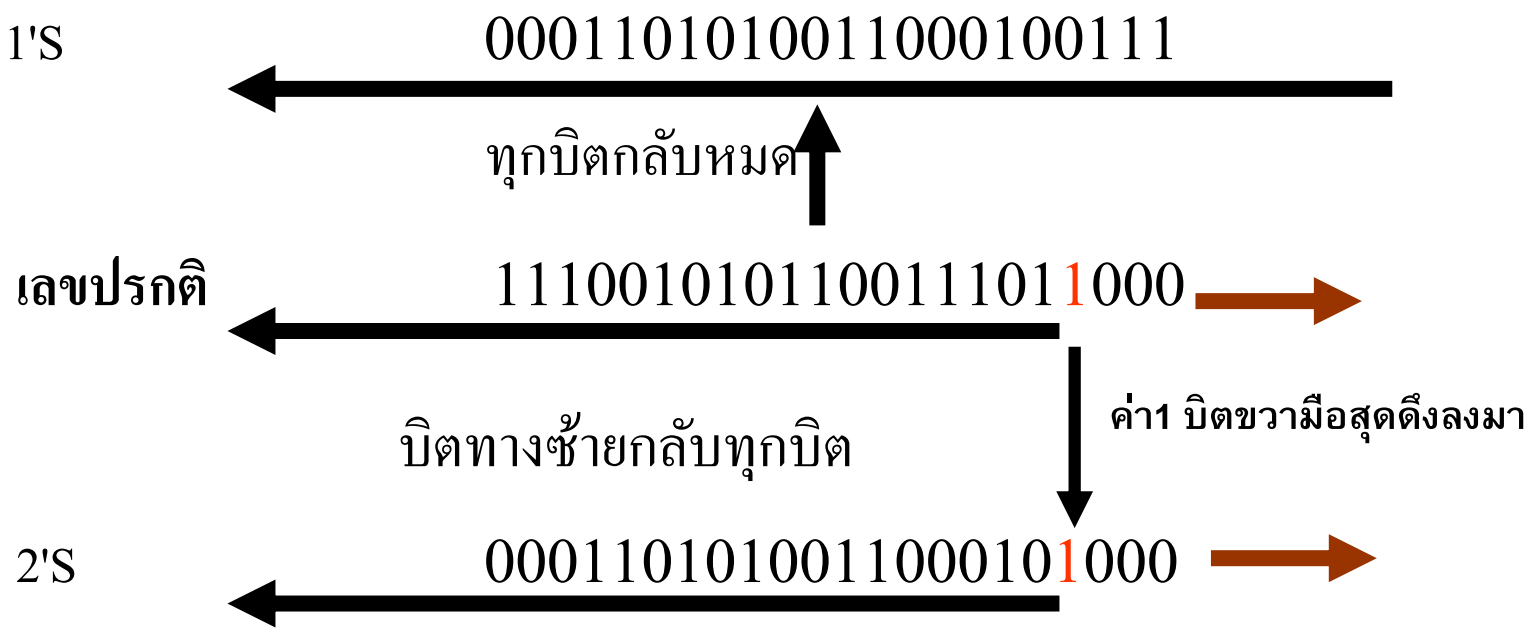
∴ คอมพลีเมนต์ 2 ของ *101101110 = 010010010*

# อธิบาย

- **ตัวอย่างที่ 2.3** การหาค่าคอมพลีเมนต์ 2 นี้ทำได้โดยนำค่าคอมพลีเมนต์ 1 ไปบวกด้วยค่า1 ค่าคอมพลีเมนต์ของเลข 101101110

- ขั้นตอนแรกนำค่าสูงสุดจำนวน 9 บิตคือ 111111111 ลบออกจากเลขที่ต้องการหาค่าคือ 101101110 ได้ค่าเป็นคอมพลีเมนต์ 1 คือ 010010001
- ขั้นตอนต่อไปนำค่าคอมพลีเมนต์ 1 ไปบวกด้วยค่า1 ได้เป็นค่าคอมพลีเมนต์ 2ตามต้องการ คือ 010010010 ดังแสดงในสไลด์

# วิธีทำ 1'S ,2'S อย่างง่ายจากค่าเลขปรกติ



# อธิบาย

- ตัวอย่าง การแปลงเลขฐานสองเป็น คอมพลีเมนต์1 และ คอมพลีเมนต์ 2 อย่างง่ายๆ

- จากค่าเลขปรกติจากค่า 111001010110011101100 0
- วิธีทำเป็นเลขคอมพลีเมนต์ 1 โดยนำค่าเลขปรกติมาเปลี่ยนเป็นค่าตรงข้ามกันทุกบิตก็จะได้ 000110101001100010011 1 แสดงดังค่าบน
- ส่วนวิธีทำเป็นเลขคอมพลีเมนต์ 2 โดยพิจารณาเลขปรกติจากขวาไปซ้ายที่ละบิต เลข 1 ที่บิตใดเกิดขึ้นก่อนให้คงค่าตั้งแต่บิตนี้ไปทางขวามือทั้งหมด ส่วนบิตที่เหลือทางซ้ายมือทั้งหมดจะถูกเปลี่ยนเป็นค่าเลขตรงข้าม ดังแสดงในสไลด์

# 212 การคอมพลีเมนต์เลขฐานแปด

**ตัวอย่าง 2.4** คอมพลีเมนต์ 7 ของ $(7653)_8$

**วิธีทำ** (*7777-7653* )

คอมพลีเมนต์ *7* = *0124*

**ตัวอย่าง2.5** คอมพลีเมนต์ 8 ของ $(7653)_8$

**วิธีทำ** $(7\,777\text{-}7653)_8$+1 คอมพลีเมนต์ *7 ( + 1 )* = *0124*

*1* +

∴ คอมพลีเมนต์ 8 ของ *0125*

หรือหาจาก (10000-*7653)* = *0125*

# อธิบาย

- เลขฐานแปดจะมีส่วนประกอบของเลขคอมพลีเมนต์7 กับคอมพลีเมนต์ 8
- ตัวอย่างที่2.4 หาค่าคอมพลีเมนต์7 ของเลข 7653ฐาน 8
- ขั้นตอนแรกนำค่าสูงสุดของเลขฐานแปดจำนวนเท่ากับบิตที่ต้องการหาค่าคือ 7777 ลบออกจากค่า 7653
- ได้คำตอบเป็นค่าคอมพลีเมนต์ของ7 คือ 0124
- ตัวอย่างที่2.5 หาค่าคอมพลีเมนต์8 ของเลข 7653 ฐาน 8
- ขั้นตอนแรกนำค่าสูงสุดของเลขฐานแปดจำนวนเท่ากับบิตที่ต้องการหาค่าคือ 7777 ลบออกจากค่า 7653
- ได้คำตอบเป็นค่าคอมพลีเมนต์ของ7 คือ 0124
- -ขั้นตอนที่สองนำค่าคอมพลีเมนต์7 ไปบวกด้วยค่า1 จะได้ค่า 0125 เป็นคอมพลีเมนต์8 ของเลข 7653 ฐาน8

# 213การคอมพลีเมนต์เลขฐานสิบ

## ใช้คอมพลีเมนต์ 9 กับคอมพลีเมนต์ 10

**ตัวอย่าง 2.5** คอมพลีเมนต์ 9 ของ $(789524)_{10}$ คือ (999999- 789524 )

คอมพลีเมนต์ 9 ของเลข $(789524)_{10}$ มีค่าเท่ากับ 2 1 0 4 7 5

**วิธีทำ**

$9 - 7 = 2$

$9 - 8 = 1$

$9 - 9 = 0$

$9 - 5 = 4$

$9 - 2 = 7$

$9 - 4 = 5$

# อธิบาย

- เลขฐาน 10 จะมีการใช้คอมพลีเมนต์ และ คอมพลีเมนต์ 10
- ตัวอย่างที่ 2.5 แสดงวิธีการหาค่าคอมพลีเมนต์ 9 ของเลข 789524 ฐาน 10
- นำ ค่าสูงสุดคือ 999999 ลบออกจาก 789524 ค่าคอมพลีเมนต์9 ที่เกิดขึ้นคือ 210475 ของเลขฐาน 10

# ตัวอย่างที่ 2.6

- คอมพลีเมนต์10 ของเลข( 789524 )$_{10}$
- คอมพลีเมนต์10 มีค่าเท่ากับคอมพลีเมนต์ 9 ( +1 )
- **วิธีทำ** ( 999999-789524 )$_{10}$

หาค่าจากคอมพลีเมนต์ 9+(1) = 210475 + 1 = 210476

หรือคอมพลีเมนต์ 10 = (1000000-789524) = 210476

# อธิบาย

- ตัวอย่างที่ 2.6 แสดงวิธีการหาค่าคอมพลีเมนต์ 10 ของเลข 789524 ฐาน 10
- นำค่าสูงสุดคือ 999999 ลบออกจาก 789524
- ค่าคอมพลีเมนต์9 ที่เกิดขึ้นคือ 210475 ของเลขฐาน 10
- นำค่าที่เกิดขึ้นของคอมพลีเมนต์ 9 ไปบวก 1
- ก็จะได้เป็นค่าคอมพลีเมนต์ 10 คือค่า 210475 บวกด้วย 1
- เป็นค่า 210476 ของเลขฐาน 10

# 214 การคอมพลีเมนต์เลขฐานสิบหก

ส่วนประกอบของเลขฐาน16 เป็นคอมพลีเมนต์ 15 กับคอมพลีเมนต์ 16

**ตัวอย่างที่ 2.7** คอมพลีเมนต์*15* ของ $(9AE)_{16}$

**วิธีทำ** $(FFF-9AE)_{16}$ *9AE* คอมพลีเมนต์ *15* = *651*

**ตัวอย่างที่ 2.8** คอมพลีเมนต์ *16* ของ *9AE*

**วิธีทำ** *9AE* คอมพลีเมนต์ *15 ( +1 )* = *651*

\+ *1*

คอมพลีเมนต์ *16* ของ *9AE* = *652* *652*

*หรือ (1000-9AE)=652*

# อธิบาย

**ตัวอย่างที่ 2.7** การหาค่าคอมพลีเมนต์15 ของเลข9AE ฐาน16

- นำค่าสูงสุดของบิตที่จะหาคือ FFF ลบออกจากค่า 9AE ค่าที่ได้จากการลบคือ 651 เป็นค่าคอมพลีเมนต์15 ของเลข9AE ฐาน16

ตัวอย่างที่2.8 เป็นตัวอย่างของการหาค่าคอมพลีเมนต์16 ของเลข 9AEฐาน16 วิธีการเป็นดังตัวอย่างที่ 2.7 ได้ค่าคอมพลีเมนต์ 15 ของเลข 9AE ฐาน 16 แล้วทำการบวกด้วย 1 จะได้ค่า 652

# หลักการนำคอมพลีเมนต์ไปใช้งาน

**ตัวอย่างที่ 2.9** 123-456 เลขฐาน10

123-456 = -333

| | | |
|---|---|---|
| | 123 | |
| | | + |
| 9'S (999-456) | 543 | |
| | 666 | |

**คำตอบคือ (999-666) = ( - ) 333**

**อยู่ในรูปคอมพลีเมนต์เพราะว่าค่าตัวตั้งน้อยกว่าตัวลบ**

# อธิบาย

- การนำคอมพลีเมนต์มาใช้งานเพื่อให้ง่ายต่อการอธิบายจะยกตัวอย่างในเลขฐานสิบก่อน
  - จากตัวอย่างที่2.9 ในตัวอย่างกำหนดให้ตัวตั้งน้อยกว่าตัวลบ
  - หลักการต้องนำค่าที่เป็นเลขลบไปทำคอมพลีเมนต์ 9
  - ในส่วนนี้ค่าที่เกิดขึ้นของคอมพลีเมนต์ 9 คือเลข 543
  - นำค่าที่เป็นตัวตั้งไปบวกกับค่าคอมพลีเมนต์ได้ 666
  - ค่าที่ผลลัพธ์จะอยู่ในรูปคอมพลีเมนต์ 9 เพราะตัวตั้งน้อยกว่าตัวลบ
  - คำตอบเปลี่ยนผลลัพธ์กลับเป็นเลขปรกติโดยใส่เครื่องหมายลบบอกค่า
  - คำตอบของค่า 123-456 = -333

# หลักการนำคอมพลีเมนต์มาใช้งาน

**ตัวอย่างที่ 2.10** 123 -100 = (+23)

การนำเอาคอมพลีเมนต์ของเลขฐาน 9

| | | |
|---|---|---|
| | 123 | |
| | | + |
| 9'S (999-100=899) | 899 | |
| over flow >> 1 | 022 | |
| | 1 | |
| | +23 | |

# อธิบาย

- หลักการนำคอมพลีเมนต์ 9 มาใช้ในรูปแบบตัวตั้งมากกว่าตัวลบ

- ตัวอย่างที่ 2.10 ค่าตัวเลขที่จะนำไปหักออกมีค่าติดลบนำไปทำเป็นคอมพลีเมนต์ 9 ได้ค่าเป็น 899
- นำค่าตัวตั้ง123 ไปบวกกับค่าคอมพลีเมนต์ 9 ของเลข 100 คือค่า 899 ได้ค่าผลลัพธ์เป็น 22 พร้อมกับมีโอเวอร์โฟล์เกิดขึ้น
- ค่าคำตอบของการทำแบบคอมพลีเมนต์ 9 นำโอเวอร์โฟล์ไปบวกกับค่าผลลัพธ์ค่าคำตอบก็คือ +23

# หลักการนำคอมพลีเมนต์ 10 มาใช้งาน

ตัวอย่างที่2.11 123 -100 = (+23)

การใช้คอมพลีเมนต์ของเลขฐาน 10

| | | |
|---|---|---|
| | 123 | + |
| 10'S (1000-100=900) | 900 | |
| 1 | 023 | |

# อธิบาย

- ตัวอย่างที่ 2.11 เป็นการนำค่าตัวตั้งมากกว่าตัวลบมาหาค่าโดยใช้หลักการของคอมพลีเมนต์ 10
- วิธีการเหมือนกับตัวอย่างที่ 2.10 แตกต่างกันอยู่ที่ค่าที่นำไปลบอยู่ในรูปของคอมพลีเมนต์ 10
- นำค่าตัวตั้งไปบวกกับค่าคอมพลีเมนต์10 ของตัวเลข 100 คือค่า 900 จะเกิดผลลัพธ์กับโอเวอร์โฟล์
- หลักการของคอมพลีเมนต์10 โอเวอร์โฟล์ที่ขึ้นให้ตัดทิ้ง
- คำตอบที่ได้คือ +23

# การทำงานของ 1’S, 2’S Complement

กำหนด x =12 ,y = 18 ใช้ 1’SComplement ในการหาค่า

กำหนดx =12 , y = 18 ใช้ 2’SComplement ในการหาค่า

# อธิบาย

- หลักการของการทำคอมพลีเมนต์1 กับการทำคอมพลีเมนต์ 2

- กำหนดค่าเลขปกติฐาน 10
- โดยกำหนดค่าx=12 กับ Y=18
- คอมพลีเมนต์1กับ 2 มีค่าต่างกันอยู่ที่ 1
- ให้หาค่าคอมพลีเมนต์1 กับคอมพลีเมนต์ 2
- เปรียบเทียบค่าที่เกิดขึ้นลงในตาราง

# กำหนดให้ x = 12 , y = 18

| ค่าฐาน10 | ค่าฐาน2 | ค่า 1'S complement | ค่า 2'S complement |
|---|---|---|---|
| X=12 | 01100 | 10011 | 10100 |
| Y=18 | 10010 | 01101 | 01110 |
| บิตเครื่องหมาย | -<br>+ | -= 1<br>+= 0 | -=1<br>+=0 |

ตารางที่2.1

# อธิบาย

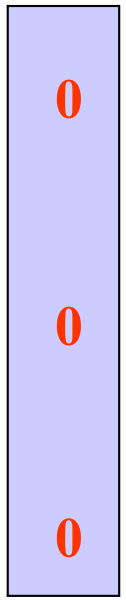
- จากตารางที่ทำการหาค่าคอมพลีเมนต์ 1 กับ 2

- กำหนดค่า x มีค่าเท่ากับ12 ของฐาน10 เปลี่ยนเป็นเลขฐานสองเท่ากับ 01100
- แปลงเป็นคอมพลีเมนต์1 เป็นค่า10011
- แปลงเป็นคอมพลีเมนต์2 เป็นค่า10100
- x มีค่าเท่ากับ18 ของฐาน10 เปลี่ยนเป็นเลขฐานสองเท่ากับ 10010
- แปลงเป็นคอมพลีเมนต์1 เป็นค่า 01101
- แปลงเป็นคอมพลีเมนต์2 เป็นค่า 01110
- บิตเครื่องหมายใช้ ค่า1 = เครื่องหมายลบ 0 = เครื่องหมายบวก

# ตัวอย่างที่ 2.12 x + y

ใช้ 1'S Complement ในการหาค่าของ x = 12 , y = 18

เนื่องจากเป็นการบวกกันระหว่าง x และy จึงสามารถบวกกันได้เลย




$0\ 1\ 1\ 0\ 0 = (12)_{10}$

$+ \qquad +$

$1\ 0\ 0\ 1\ 0 = (18)_{10}$

ตอบ $1\ 1\ 1\ 1\ 0 = +(30)_{10}$

บิตเครื่องหมาย

# อธิบาย

- ตัวอย่างที่2.12 หาค่า x+y

- โดยใช้คอมพลีเมนต์ 1
- จากตัวอย่างบิตเครื่องหมายเป็นบวกทั้งตัวตั้งและตัวที่จะนำไปบวก ค่าในบิตเครื่องหมายคือ 0
- ค่าทั้งสองเป็นบวกจึงแทนเป็นค่าเลขฐานสอง และบวกได้โดยตรง
- ค่าที่เกิดขึ้นในบิตเครื่องหมายคือ 0 นั่นแสดงว่าคำตอบที่ได้เป็นค่าบวก

# ตัวอย่างที่ 2.13 X - Y

ทำ 1' S Complement Y จาก 10010 เป็น 01101

บิตเครื่องหมาย

| | | |
|---|---|---|
| 0 | 0 1 1 0 0 = $(12)_{10}$ | |
| | + | + |
| 1 | 0 1 1 0 1 = $(-18)_{10}$ | |
| | ↓ ↓ ↓ ↓ ↓ | |
| 1 | 1 1 0 0 1 | บิตเครื่องหมาย=1 คำตอบติดลบ |

ทำการเปลี่ยนสถานะ 0 0 1 1 0 = $(-6)_{10}$

ตอบ $(-00110)_2 = (-6)_{10}$

# อธิบาย

- ตัวอย่างที่ 2.13 X-Y โดยใช้คอมพลีเมนต์ 1

- จากตัวอย่างบิตเครื่องหมายเป็นบวกตัวตั้ง และส่วนตัวลบเป็นเครื่องหมายลบแสดงในบิตซ้ายสุดในกรอบสี่เหลี่ยม
- แทนค่าเลขฐานสองทั้งตัวตั้ง และตัวลบหลังเส้นทึบด้านขวามือ
- ทำการบวกค่าทั้งสองเข้าด้วยกันพร้อมบิตเครื่องหมาย
- ค่าทั้งสองที่บวกกันได้เป็นเลขฐานสองมีค่าเท่ากับ 11001 และมีบิตเครื่องหมายเป็น 1ซึ่งแสดงให้เห็นว่าคำตอบอยู่ในรูปของคอมพลีเมนต์
- ทำการเปลี่ยนสถานะจากคอมพลีเมนต์ 1 ให้เป็นเลขปรกติโดยการกลับเลขทุกหลักเป็นตรงกันข้าม และบิตเครื่องหมายให้แสดงค่าเป็นค่าลบ

# ตัวอย่างที่2.14 -X + Y

ทำ 1’ S Complement X จาก 01100 เป็น 10011

บิตเครื่องหมาย

| | | |
|---|---|---|
| 1 | 1 0 0 1 1 | $= (-12)_{10}$ |
| | + + | |
| 0 | 1 0 0 1 0 | $= (18)_{10}$ |
| | ↓ ↓ ↓ ↓ ↓ | |
| 0 | 0 0 1 0 1 | |

เกิน 1

\+

1

0 0 1 1 0 $= (6)_{10}$

ตอบ $(00110)_2 = (6)_{10}$

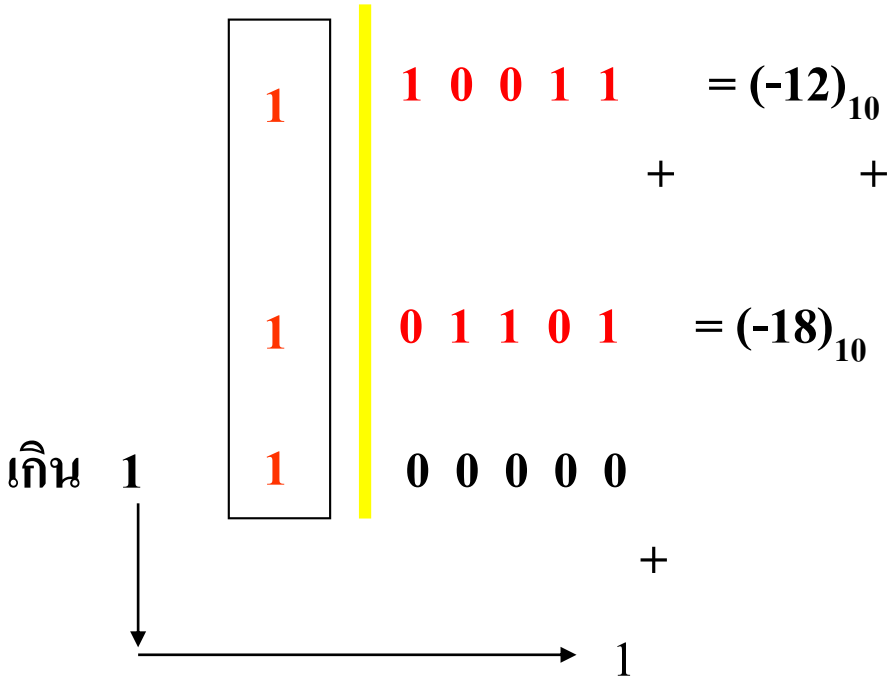
# อธิบาย

- ตัวอย่างที่ 2.14 – X+Y ตัวอย่างนี้แสดงตัวตั้งมีเครื่องหมายเป็นลบมีค่าน้อยกว่าบวกด้วยค่าที่มากกว่ามีเครื่องหมายเป็นบวก

- ขั้นตอนแรกทำการแปลงค่าตัวตั้งที่เป็นลบให้อยู่ในรูปคอมพลีเมนต์1ได้ค่าเป็นเลข 10011 โดยมีบิตเครื่องหมายเป็นลบคือค่า 1
- นำตัวบวกแทนค่าของเลข 18 ฐาน 10 คือ10010
- ทำการบวกกันทุกบิตรวมบิตเครื่องหมายด้วย ค่าที่เกิดขึ้นจะมี 3 ส่วนคือ ผลลัพธ์ บิตเครื่องหมาย และโอเวอร์โฟล์
- นำค่าโอเวอร์โฟล์ไปบวกกับผลลัพธ์จะได้คำตอบโดยแทนค่าบิตเครื่องหมายเป็นบวกเท่ากับ (-12+18=+6)

# ตัวอย่างที่2.15 -X-Y

ทำ 1' S X จาก 01100 เป็น 10011 และ Y จาก 10010 เป็น 01101




คำตอบติดลบ 1 0 0 0 0 1

ทำการเปลี่ยนสถานะ - 1 1 1 1 0 = $(-30)_{10}$

**ตอบ** $(-11110)_2 = (-30)_{10}$

# อธิบาย

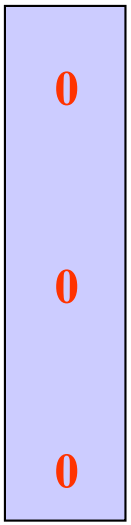
- ตัวอย่างที่ 2.15 เมื่อทั้งตัวตั้งและตัวบวกมีค่าเป็นลบ –X-Y ค่าทั้งสองอยู่ในรูปคอมพลีเมนต์

- ขั้นตอนแรกแปลงค่าทั้งสองเป็นคอมพลีเมนต์ 1 โดยแสดงบิตเครื่องหมายเป็น 1 ทั้งตัวตั้งและตัวบวก
- ทำการบวกเลขสองจำนวนเข้าด้วยกันรวมทั้งบิตเครื่องหมายด้วย
- ค่าที่เกิดขึ้นจะมีสามส่วนด้วยกันคือ ผลลัพธ์ บิตเครื่องหมาย และโอเวอร์โฟล์
- นำโอเวอร์โฟล์ไปบวกกับผลลัพธ์จะได้เป็นค่าคำตอบแต่บิตเครื่องหมายแสดงค่าเป็น1 ซึ่งหมายถึงค่าติดลบ ก็คือค่าเลขคอมพลีเมนต์
- คำตอบสุดท้ายต้องแปลงเลขผลลัพธ์ให้เป็นเลขปกติโดยการกลับค่าผลลัพธ์ให้ตรงกันข้ามยกเว้นบิตเครื่องหมายให้แสดงค่าลบ (-12-18=-30)

# ตัวอย่างที่2.16 X + Y

ใช้ 2’S Complement ในการหาค่าของ x = 12 , y = 18

เนื่องจากเป็นการบวกกันระหว่าง x และy จึงสามารถบวกกันได้เลย




$0\ 1\ 1\ 0\ 0 = (12)_{10}$

\+ +

$1\ 0\ 0\ 1\ 0 = (18)_{10}$

$1\ 1\ 1\ 1\ 0 = (30)_{10}$

# อธิบาย

- ตัวอย่างที่ 2.16 แสดงการใช้คอมพลีเมนต์ 2 ในการหาค่า X+Y
- ค่าที่กำหนดให้เป็นค่าที่มีบิตเครื่องหมายเป็นบวกทั้งตัวตั้งและตัวบวก
- สามารถแทนค่าเลขฐานสองทั้งสองจำนวนโดยมีบิตเครื่องหมายเป็น 0
- นำค่าทุกบิตไปบวกกันพร้อมบิตเครื่องหมาย
- ค่าที่เกิดขึ้นมีสองส่วนคือ ผลลัพธ์กับบิตเครื่องหมาย
- ค่าบิตเครื่องหมายเป็นบวกผลลัพธ์เป็นค่าที่สามารถแปลงเป็นฐานสิบโดยตรง เท่ากับ +30

# ตัวอย่างที่ 2.17 X - Y

| | | |
|---|---|---|
| 0 | 0 1 1 0 0 | = $(12)_{10}$ |
| | + | + |
| 1 | 0 1 1 1 0 | = $(-18)_{10}$ |
| 1 | 1 1 0 1 0 | |

2'S

ทำการลบด้วย "1"

ทำการเปลี่ยนสถานะ 2'S เป็นค่าปรกติ

ทำ Yเป็น 2'S Complementโดย
เปลี่ยนสถานะแล้วบวกด้วย 1

1'S → 1 1 0 0 1

10010 = 01101

\+

1

01110

ค่าปรกติ → - 0 0 1 1 0 = $(-6)_{10}$

**ตอบ** $(-00110)_2 = (-6)_{10}$

# อธิบาย

- ตัวอย่างที่ 2.17 การนำคอมพลีเมนต์ 2 มาใช้หาค่า X-Y ตัวตั้งเป็นค่าบวก ตัวบวกเป็นติดค่าลบ

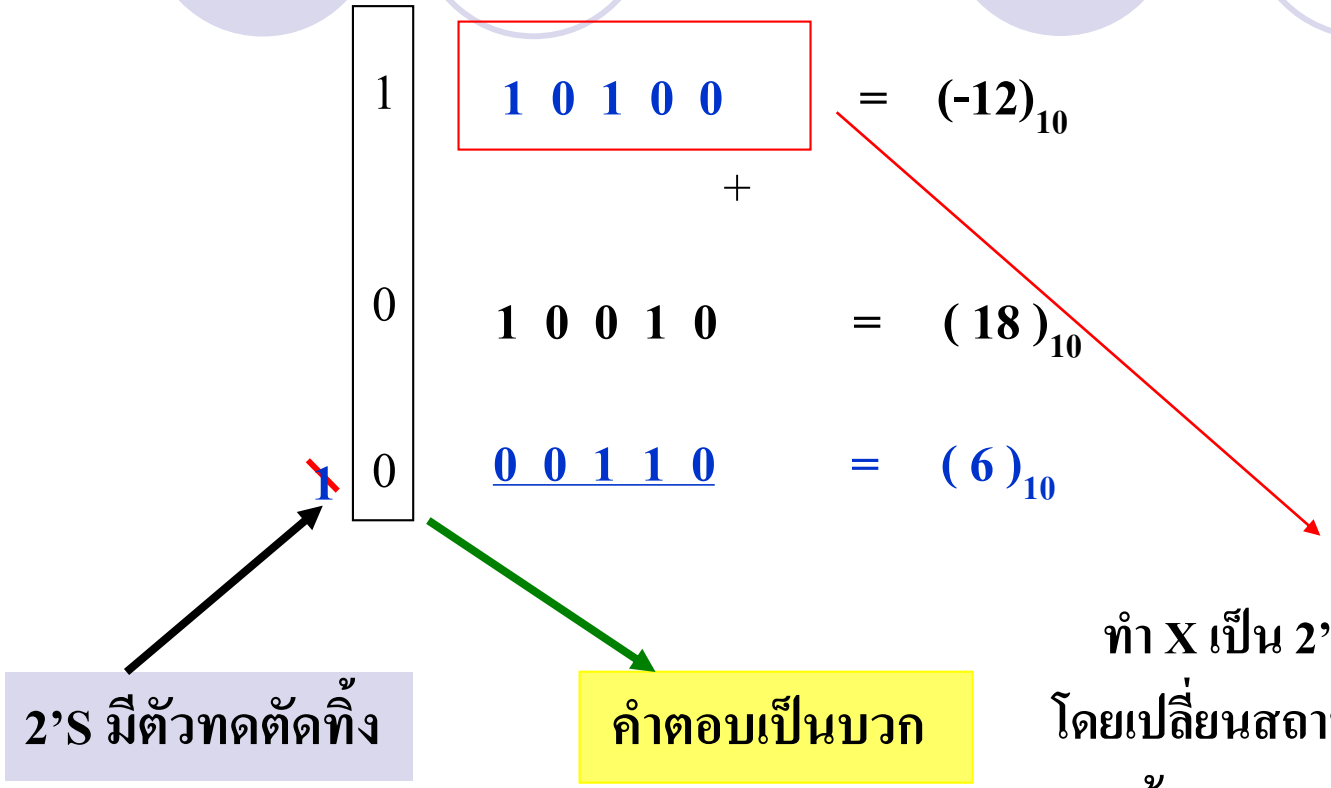
- ขั้นตอนแรกแทนค่าเลข 12 ฐานสิบมีค่าเท่า01100 มีบิตเครื่องหมายเป็น 0
- แทนค่าเลข -18 เป็นแบบคอมพลีเมนต์ 2 ซึ่งมีค่าเท่ากับ 01110จากตาราง ที่ผ่านมาเครื่องหมายเป็น 1
- แทนค่า 0และ1 ในบิตเครื่องหมายกรอบสี่เหลี่ยม
- ทำการบวกสองจำนวนเข้าด้วยกันรวมทั้งบิตเครื่องหมายด้วย
- ค่าที่เกิดขึ้นจะมีสองส่วนคือส่วนของผลลัพธ์ กับ ค่าบิตเครื่องหมาย
- จากตัวอย่างนี้ค่าเครื่องหมายเป็นลบบอกถึงผลลัพธ์ว่าอยู่ในรูปคอมพลีเมนต์

# อธิบาย(ต่อ)

- การแปลงค่าคอมพลีเมนต์2ให้กลับเป็นค่าปกติใช้หลักย้อนทางเดิม

- โดยการลบค่าคอมพลีเมนต์ 2 ออก 1 เหลือค่าเป็นคอมพลีเมนต์ 1 แล้วทำการกลับเลขทุกบิตของคอมพลีเมนต์1ก็จะได้เป็นค่าปกติ
- จากตัวอย่างค่าที่ได้ 11010 เป็นคอมพลีเมนต์2 ลบด้วย 1 จะได้เป็น11001 เป็นคอมพลีเมนต์ 1
- ทำการแปลงค่าจากคอมพลีเมนต์ 1 เป็นค่าปกติคือ 00110
- คำตอบที่ได้จากตัวอย่างนี้คือ -6

# ตัวอย่างที่ 2.18 -X+Y




ทำ X เป็น 2’S Complement โดยเปลี่ยนสถานะจาก 1’s แล้วบวกด้วย 1

01100 = 10011 +1=10100

ตอบ $(00110)_2 = (6)_{10}$

# อธิบาย

- ตัวอย่างที่ 2.18 ค่าที่กำหนดXเป็นตัวตั้งมีค่าเป็นลบบวกด้วยค่าY มีค่าเป็นบวก
- - ค่า X ทำการแทนด้วยคอมพลีเมนต์ 2 คือค่าที่ได้จากตารางที่2.1 เป็น 10100 และ บิตเครื่องหมายเป็น 1
- - ค่า Y ทำการแทนด้วยเลขฐาน2คือค่า10010 และมีบิตเครื่องหมายเป็น 0 ทำการบวกกันทุกบิต
- - ได้ค่าเป็นสามส่วนคือผลลัพธ์ บิตเครื่องหมาย และบิตโอเวอร์โฟล์ใน การทำคอมพลีเมนต์ 2 ค่าโอเวอร์โฟล์ตัดทิ้ง
- - คำตอบบิตเครื่องหมายเป็นบวก คำตอบไม่มีการเปลี่ยนค่า 00110

ผลลัพธ์มีค่าดังนี้ (-12+18=+6)

# ตัวอย่างที่ 2.19 -X -Y

| | | |
|---|---|---|
| 1 | 1 0 1 0 0 | $= (-12)_{10}$ |
| | + | |
| 1 | 0 1 1 1 0 | $= (-18)_{10}$ |

ทำ Xและyเป็น2's Complement

มีตัวทดตัดทิ้ง ~~1~~ 1 0 0 0 1 0 อยู่ในรูป 2's

เปลี่ยนสถานะ 1 1 1 0 1

บวกด้วย "1" + 1

ตอบ ติดลบ - 1 1 1 1 0 $= (-30)_{10}$

$(11110)_2 = (-30)_{10}$

# อธิบาย

- ตัวอย่างที่ 2.19 กำหนดให้ค่าตัวตั้งX เป็นลบค่าที่นำมาบวกYก็เป็นค่าลบ ใช้วิธีการของคอมพลีเมนต์2 จากคารางที่2.1
  - การนำX แทนค่าคอมพลีเมนต์2 ได้ค่าเป็น 10100 บิตเครื่องหมายเป็นค่า1
  - นำค่า Y แทนค่าคอมพลีเมนต์2 ได้ค่าเป็น 01110 บิตเครื่องหมายเป็นค่า1
  - ทำการบวกทั้งค่า X กับ Y เข้าด้วยกันพร้อมบิตเครื่องหมาย
  - ผลที่เกิดขึ้นมีสามส่วน บิตผลลัพธ์ บิตเครื่องหมาย บิตโอเวอร์โฟล์
  - การทำแบบคอมพลีเมนต์2 เมื่อเกิดโอเวอร์โฟล์ให้ตัดทิ้ง
  - พิจารณาบิตเครื่องหมายเป็นลบค่าผลลัพธ์ต้องทำการแปลงเป็นค่าปกติ

# 22หลักการบวกเลข

## 221การบวกเลขฐานสอง

**ตัวอย่างที่ 2.20** การบวกเลขฐานสอง $(110)_2 + (011)_2 = (\ldots\ldots)_2$

**วิธีทำ**

$$\begin{array}{rcl} 110 & & 6 \\ + & & + \\ \underline{011} & & \underline{3} \\ 1001 & & 9 \end{array} \qquad \therefore$$

$\therefore \quad (110)_2 + (011)_2 = (1001)_2$

# อธิบาย

- หลักการบวกเลขฐานสอง

- กฎเกณฑ์การบวกเลขฐานสองง่ายกว่าฐานจะใช้กฎดังนี้

  0+0 = 0

  0+1 = 1

  1+0 = 1

  1+1 = 0 ทดไปหลักทางซ้าย1

- ตัวอย่างที่ 2.20 การบวกเลขฐานสอง 3บิต สองจำนวนเข้าด้วยกันจำนวน เลข110 +011 = 1001 ดังแสดงในรูปสไลด์

## 222 การบวกเลขฐานแปด

**ตัวอย่างที่ 2.21** การบวกเลขฐานแปด

$$(76534)_8 + (35726)_8 = (..)_8$$

วิธีทำ

$$\begin{array}{r} 76534 \\ + \\ \underline{35726} \\ \underline{134462} \end{array}$$

# อธิบาย

- หลักการบวกเลขฐานแปดโดยใช้หลักการเหมือนกับฐานสิบทุกประการ มีข้อแตกต่างอยู่ที่ผลบวกของเลขฐานแปดจะไม่เกินค่าเลข 0-7 ถ้ามีค่าตั้งแต่8 ขึ้นไปก็ให้เอา 8 ไปลบออกเหลือเท่าไรก็ให้ใส่ค่าลงไปแล้วทดไปหลักหน้าหรือหลักทางซ้ายมือ 1

ตัวอย่างที่ 2.21 การบวกเลขฐานแปดจากขวาไปซ้ายค่าหลักซ้าย

4+6 = 10 เกินค่า 8 นำไปลบด้วย 8 มีค่าเท่า 2 ทดไปหลักต่อไป1

3+2 = 5 บวกกับตัวทดอีก 1 ได้ 6

5+7 = 12 ลบออก8 ได้ค่า 4 ทดไปหลักถัดไป 1

6+5 +1 = 12 -8 = 4 ทดไปหลักถัดไป 1

7+3+1 = 11 -8 ได้ผลลัพธ์ 3 ทดไปหลักหน้าสุด 1

ได้คำตอบ คือ $(134462)_8$

# 223 การบวกเลขฐานสิบหก

**ตัวอย่างที่ 2.22** $(99A)_{16} + (825)_{16} = (\ldots\ldots)_{16}$

**วิธีทำ**

$$\begin{array}{r} 9\ \ 9\ \ A \\ + \\ \underline{8\ \ 2\ \ 5} \\ 1\ \ 1\ \ B\ \ F \end{array}$$

$(99A)_{16} + (825)_{16} = (11BF)_{16}$

# อธิบาย

**ตัวอย่างที่ 2.22** กำหนดให้ $(99A)_{16}$ บวกด้วย $(825)_{16}$

ดูจากตารางการบวกเลข

$$A+5 = F$$

$$9+2 = B$$

$$9+8 = 1 \quad \text{ทด1}$$

คำตอบมีค่าเท่ากับ $(11BF)_{16}$

# ตารางการบวกเลขฐาน 16

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 |
| 3 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 |
| 4 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 |
| 5 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 |
| 6 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 |
| 7 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 8 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 9 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| A | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| B | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A |
| C | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B |
| D | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C |
| E | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| F | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E |

# อธิบาย

- ตารางการบวกเลขในคอลัมน์บนแสดงเลขฐาน 16 มีค่าเท่ากับ 0 ถึง Fและในส่วนของแถวก็จะแสดงเลขฐาน16 มีค่าเท่ากับ0 ถึง F

- ค่าตัวตั้งถูกกำหนดเป็นแถว
- ค่าตัวบวกกำหนดให้เป็นคอลัมน์
- นำค่าทางแถวบวกกับทางคอลัมน์ จุดตัดที่เกิดขึ้นในตารางคือค่าผลลัพธ์
- ตัวอย่างเช่น F+C จุดตัดแถว F กับคอลัมน์ C มีค่าเท่ากับ 1B

# 23หลักการลบเลข

## 231 การลบเลขฐานสอง

**ตัวอย่างที่** 2.23 การลบเลขฐานสอง $(10101)_2 - (01101)_2 = (\ldots)_2$

**วิธีทำ**

$$\begin{array}{rr} 1\,0\,1\,0\,1 & 21 \\ - & - \\ \underline{0\,1\,1\,0\,1} & \underline{13} \\ 0\,1\,0\,0\,0 & 8 \end{array}$$

$(10101)_2 - (01101)_2 = (01000)_2$

# อธิบาย

- หลักการลบเลขฐานสองแต่จะใช้หลักการตรงกันข้ามกับการบวก กรณีที่มีค่าในหลักไม่เพียงพอก็มีการยืมหลักที่สูงกว่ามาในที่นี้ก็คือหลักทางซ้ายมือมีกฎดังนี้

0-0=0

1-0=1

1-1=0

0-1=1 ต้องยืมจากหลักที่สูงมา 1

จากตัวอย่างที่ 2.23 แสดงวิธีการลบเลขฐานสองจำนวน 5 บิต ใช้หลักการลบจากหลักขวาสุดไปทางซ้าย พิจารณาที่หลักที่ 4

กรณีมีการยืมจากหลักซ้ายจะมีค่าเท่ากับ10ลบด้วย1จะเหลือค่า 1

# 232การลบเลขฐานแปด

ตัวอย่างที่ 2.24 การลบเลขฐานแปด

$$(76534)_8 - (35726)_8 = (\ldots\ldots)_8$$

วิธีทำ

$$\begin{array}{r} 76534 \\ -\ 35726 \\ \hline 40606 \end{array}$$

# อธิบาย

- การลบฐานแปดหลักการเหมือนกับการลบเลขฐานสิบ
- ตัวอย่างที่2.24 การลบเลขฐานแปด
- จากหลักการพื้นฐานทางขวามือสุดเมื่อเกิน 8 จะทดเป็นค่า1 ทางซ้ายหลักถัดไป
- ในการลบให้พิจารณาหลักขวาสุดตัวตั้ง 4 -6 ไม่ได้ยืมหลักที่มากกว่ามา 1 ซึ่งมีค่าเท่ากับ (8 +4)-6 = 6

3 ตัวตั้งถูกยืม 1 เหลือ 2 -2 = 0

หลักถัดมา 5+8-7 = 6

(6-1)-5 = 0

7-3 = 4

คำตอบที่ได้จะมีค่าเท่ากับ $(40606)_8$

# 233 การลบเลขฐานสิบหก

**ตัวอย่างที่ 2.25** $(92A)_{16} - (352)_{16} = (...?...)_{16}$

**วิธีทำ**

$$92A$$
$$-$$
$$\underline{352}$$
$$\therefore \quad \underline{\underline{5D8}}$$

$$(92A)_{16} - (352)_{16} = (5D8)_{16}$$

# อธิบาย

ตัวอย่างที่ 2.25 ใช้หลักการเดียวกับการลบฐาน10 หากตัวตั้งมีค่าน้อยกว่าตัวลบให้ทำการยืมตัวหลักที่มีค่ามากกว่าเพียงแต่การยืมของหลักที่มีค่ามากกว่าจะมีค่าเท่ากับ16 นำไปบวกกับตัวตั้งและนำไปลบออก

A=10 , A-2 = 8

(2+16)-5 = D

(9-1)-3 = 5

คำตอบ $(5D8)_{16}$

## 24หลักการคูณเลข

### 241การคูณเลขฐานสอง

**ตัวอย่างที่ 2.26** การคูณเลขฐานสอง $(11011)_2 \times (101)_2 = (\ldots?\ldots)_2$

**วิธีทำ**

$$
\begin{array}{r}
11011 \\
\times \quad 101 \\
\hline
11011 \\
00000 \\
+ \quad 11011 \\
\hline
10000111
\end{array}
$$

$\therefore$ $10000111$

$(11011)_2 \times (101)_2 = (10000111)_2$

# อธิบาย

- การคูณในเลขฐานสองทำโดยง่ายกว่าฐานอื่นเพราะค่าที่เกิดขึ้นมี 2 ค่า คือ 0 กับค่า 1

  0 คูณ 1 มีค่าเท่ากับ 1 คูณ 0 ผลลัพธ์มีค่าเท่ากับ 0

  1 คูณ 1 มีค่าเท่ากับ 0 ผลลัพธ์มีค่าเท่ากับ 1

หลักการในตัวอย่างที่ 2.26 ตัวตั้งฐานสองจำนวน 5 บิต ตัวคูณฐานสองจำนวน 3 บิต คูณจากขวามือไปซ้ายมือ ได้ค่าทั้งหมดนำมารวมกัน

$(11011 * 101 = 10000111)_2$ ตรวจสอบคือ $(27*5=135)_{10}$

# 242 การคูณเลขฐานแปด

**ตัวอย่างที่ 2.27** $(3)_8 \times (2)_8 = (\ldots\ldots)_8$

**วิธีทำ**

$$\begin{array}{r} 3 \\ \times\ 2 \\ \hline 6 \end{array}$$

**ตัวอย่างที่ 2.28** $(7)_8 \times (2)_8 = (\ldots\ldots)_8$

**วิธีทำ**

$$\begin{array}{r} 7 \\ \times\ 2 \\ \hline 14 \end{array}$$

**เกินเลขฐานแปด**

14-8 = 6 ทดไว้ 1

$\therefore (16)_8$

# ตารางการคูณฐาน8

| คูณ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 0 | 2 | 4 | 6 | 10 | 12 | 14 | 16 |
| 3 | 0 | 3 | 6 | 11 | 14 | 17 | 22 | 25 |
| 4 | 0 | 4 | 10 | 14 | 20 | 24 | 30 | 34 |
| 5 | 0 | 5 | 12 | 17 | 24 | 31 | 36 | 43 |
| 6 | 0 | 6 | 14 | 22 | 30 | 36 | 44 | 52 |
| 7 | 0 | 7 | 16 | 25 | 34 | 43 | 52 | 61 |

# อธิบาย

- จากตารางการคูณของเลขฐาน 8 สามารถหาค่าเลขฐานแปดโดยนำค่าทางแถวมาคูณกับทางคอลัมน์ ค่าตัดกันเป็นค่าของผลลัพธ์
- ตัวอย่างที่ 2.27 กำหนดให้ 3 เป็นเลขฐานแปดคูณกับค่า 2 ของเลขฐานแปดมีค่าเท่ากับ $(6)_8$
- ตัวอย่างที่ 2.28 กำหนดค่า 7 คูณกับ 2 ในเลขฐาน8 ค่าที่เกิดเป็น$(16)_8$หรือหาจากค่าจากการคูณ7 กับ2 เท่ากับ 14 -8 =6 ทดค่า 1 ดังแสดงในตัวอย่างนี้

# ตารางการคูณเลขฐาน 16

| คูณ | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *A* | *B* | *C* | *D* | *E* | *F* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *0* | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| *1* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| *2* | 0 | 2 | 4 | 6 | 8 | A | C | E | 10 | 12 | 14 | 16 | 18 | 1A | 1C | 1E |
| *3* | 0 | 3 | 6 | 9 | C | F | 12 | 15 | 18 | 1B | 1E | 21 | 24 | 27 | 2A | 2D |
| *4* | 0 | 4 | 8 | C | 10 | 14 | 18 | 1C | 20 | 24 | 28 | 2C | 30 | 34 | 38 | 3C |
| *5* | 0 | 5 | A | F | 14 | 19 | 1E | 23 | 28 | 2D | 32 | 37 | 3C | 41 | 46 | 4B |
| *6* | 0 | 6 | C | 12 | 18 | 1E | 24 | 2A | 30 | 36 | 3C | 42 | 48 | 4E | 54 | 5A |
| *7* | 0 | 7 | E | 15 | 1C | 23 | 2A | 31 | 38 | 3F | 46 | 4D | 54 | 5B | 62 | 69 |
| *8* | 0 | 8 | 10 | 18 | 20 | 28 | 30 | 38 | 40 | 48 | 50 | 58 | 60 | 68 | 70 | 78 |
| *9* | 0 | 9 | 12 | 1B | 24 | 2D | 36 | 3F | 48 | 51 | 5A | 63 | 6C | 75 | 7E | 87 |
| *A* | 0 | A | 14 | 1E | 28 | 32 | 3C | 46 | 50 | 5A | 64 | 6E | 78 | 82 | 8C | 96 |
| *B* | 0 | B | 16 | 21 | 2C | 37 | 42 | 4D | 58 | 63 | 6E | 79 | 84 | 8F | 9A | A5 |
| *C* | 0 | C | 18 | 24 | 30 | 3C | 48 | 54 | 60 | 6C | 78 | 84 | 90 | 9C | A8 | B4 |
| *D* | 0 | D | 1A | 27 | 34 | 41 | 4E | 5B | 68 | 75 | 82 | 8F | 9C | A9 | B6 | C3 |
| *E* | 0 | E | 1C | 2A | 38 | 46 | 54 | 62 | 70 | 7E | 8C | 9A | A8 | B6 | C4 | D2 |
| *F* | 0 | F | 1E | 2D | 3C | 4B | 5A | 69 | 78 | 87 | 96 | A5 | B4 | C3 | D2 | E1 |

# 243การคูณเลขฐานสิบหก

**ตัวอย่างที่ 2.29** $(95A)_{16} \times (3B)_{16} = (\ldots?\ldots)_{16}$

**วิธีทำ**

$$\begin{array}{r} 95A \\ \times \quad 3B \\ \hline 66DE \\ 1C0E \\ \hline 227BE \end{array}$$

$\therefore \quad (95A)_{16} \times (3B)_{16} = (227BE)_{16}$

# อธิบาย

- ตารางการคูณเลขฐาน16 เพื่อให้ง่ายต่อการคูณค่าจำเป็นต้องใช้ตารางการหาค่ามาช่วยในการคำนวณ โดยใช้ตัวตั้งที่จะไปคูณเป็นค่าแทนเลขแถว ส่วนตัวคูณเป็นค่าเลขที่แนวนอนทางคอลัมน์

- นำค่าเลขทางแถวมาคูณคอลัมน์ก็จะได้เป็นผลลัพธ์

จากตัวอย่างที่2.29 การนำค่า $(95A)_{16}$ มาคูณกับ $(3B)_{16}$

หลักแรกคือค่าผลคูณของ $(95A)_{16}$ กับ B

- B×A = E ทดค่าไว้ 6
- (B×5)+6 = 7 ทดไว้ 3 นำ 7 ไปบวก 6 เท่ากับ D
- (B×9)+3 = 3 ทดไว้ 6 นำ 3 ไปบวก 3 เท่ากับ 6
- 6 ที่ถูกทดนำลงไปเป็นหลักแรก

ค่าที่ได้เมื่อนำเอาB มาคูณได้ $(66D3)_{16}$

# อธิบาย(ต่อ)

- นำค่าของ 3 ไปคูณกับ 95A

$3 \times A = E$ ทดค่าไว้ 1 วางให้ตรงกับตำแหน่งที่คูณ

$(3 \times 5)+1 = 0$ ทดค่าไว้ 1

$(3 \times 9)+1 = C$ ทดค่าไว้ 1

1 ที่ถูกทดนำเป็นหลักแรกค่าที่ได้คือ $(1C0E)_{16}$

นำค่าทั้งสองมารวมกันได้เป็นผลคูณ คือ $(227BE)_{16}$

# 25หลักการหารเลข

## 251การหารเลขฐานสอง

**ตัวอย่างที่2.30** การหารเลขฐานสอง $(11010)_2 \div (10)_2 = (.....)_2$

**วิธีทำ**

$$\begin{array}{r} 1101 \\ 10\overline{)11010} \\ \underline{10} \\ 10 \\ \underline{10} \\ 0010 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 13 \\ 2\overline{)26} \\ \underline{2} \\ 06 \\ \underline{06} \\ \underline{00} \end{array}$$

$\therefore (11010)_2 \div (10)_2 = (1101)_2$

# 252 การหารเลขฐานแปด

**ตัวอย่างที่ 2.31** $(33314)_8 \div (765)_8 = (\ldots\ldots)_8$

**วิธีทำ**

$$
\begin{array}{r}
34 \\
765\overline{)33314} \\
\underline{2737} \\
3724 \\
\underline{3724} \\
0000
\end{array}
$$

$(33314)_8 \div (765)_8 = (34)_8$

# 253การหารเลขฐานสิบหก

**ตัวอย่างที่ 2.32** $(227BE)_{16}(95A)_{16} = (\ldots\ldots)_{16}$

*วิธีทำ*

$$95\ A\overline{)227BE}\ \ 3B$$

$$\underline{1C0E}\ -$$

$$66DE$$

$$\therefore\quad \underline{66DE}\ -$$

$$(227BE)_{16}\ (95A)_{16} = (3B)_{16}$$

# อธิบาย

- หลักการหารของฐานสองใช้วิธีการแบบฐานสิบผลของการหารแต่ละค่าจะเป็นไปได้คือค่า 1 กับค่า 0 เพียงเท่านั้นดังตัวอย่างที่ 2.30
- ตัวอย่างที่2.31 และ 2.32 เป็นการหารของเลขฐาน8 กับเลขฐาน 16 มีหลักการหารเหมือนฐานสิบทุกประการ และสิ่งที่ต้องคำนึงจำเป็นต้องใช้การคูณเข้ามาช่วย และเพื่อความสะดวกจึงต้องมีตารางการคูณเลขฐาน 8 กับตารางการคูณของเลขฐาน 16 เข้าช่วย
- นักศึกษาควรทำความเข้าใจในหลักพื้นฐานของฐาน10 ให้ดีก็สามารถที่จะนำไปใช้ได้ทุกฐาน

# ทดสอบความรู้ในสัปดาห์ที่2

- 1.คำนวณหาค่าA+B,A-B,A*B และ A/Bของเลขฐานสองตามลำดับที่ให้มา
- (a)10101,1011 (b)1011010,101111 (c)101,1011 (d)10110110, 01011011 (e) 1101011,1010 (f)1010101,101010 (g)10000,1001 (h)1011.0101,11011
- 2. คำนวณหาค่าA+B , A –B , A*B และ A/B ของเลขฐานแปดตามลำดับที่ให้มา
- (a) 372,156 (b)704,230 (c) 1000,777 (d) 423, 651
- 3.คำนวณหาค่าA+B,A-B,A*B และ A/Bของเลขฐานสิบหกตามลำดับที่ให้มา
- (a)2CF3,2B (b) FFFF, 1000 (c) 9A5, D17 (d) 372, 156
- 4.จงหาค่าของ Z ที่เกิดจากค่า X= $(24)_{10}$,Y=$(16)_{10}$ โดยวิธีของ 1’S ,2’S
- (a) Z=X+Y (b) Z=X-Y (c) Z=-X+Y (d) Z=-X-Y

# เอกสารอ้างอิง


1. Douglas V.Hall Digital circuit and System, MC Graw-Hill Publishing Company, 1989
2. Milos D.Ercegovac, Tomas Lang and Jame H.Morino, Introduction to Digital System, John wiley & sons, Inc 1999.
3. Victor P.Nelson, H.Troy Nagle, Bill D.carrall J.David Trwin, Digital logic circuit analysis and design, Prentice Mall, International edition,1995.
4. M.Morris Mano, Digital design, Third edition, Prentice Mall, International edition , 2002.
5. Ronald J.Tocci , Digital System Principles and Applications, Prentice Hall , Fifth Edition, 1991
6. วัฒนา แก้วดุก การออกแบบดิจิตอลวิศวกรรม, ทฤษฎีและการประยุกต์ใช้งาน, สำนักพิมพ์ สสท.(ไทย-ญี่ปุ่น), 2546.
7. ธนัท ชัยยุทธ, กณพ แก้วพิชัย ดิจิตอลพื้นฐาน, บริษัทซีเอ็ดยูเคชัน จำกัด 2521
8. ธีรวัฒน์ ประกอบผล ดิจิตอลเทนิค, บริษัทซีเอ็ดยูเคชัน 2546
9. ถวัลย์วงศ์ ไกรโรจนานันท์, อิเล็กทรอนิกส์ระบบดิจิตอล ฉบับปรับปรุง, สวทช. 2542
10. ธวัชชัย เลื่อนฉวี, อนุรักษ์ เถื่อนศิริ, ดิจิตอลเทคนิค เล่ม 1-2 , 2532, 2543
11. มงคล อาวัชนาการ , โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ,มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น
12. ศัพท์เทคนิควิศวกรรมไฟฟ้าสื่อสาร สมาคมวิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์, กันยายน, 2541

# เนื้อหาคำบรรยายใน สัปดาห์ที่ 3 การลดรูปสมการลอจิก โดยวิธีพีชคณิตบูลลีน และแผนผังคาร์นอร์

31 ทฤษฎีพีชคณิตบูลลีน

311 คำจำกัดความ Poslulate

312 Venn Diagram For Postulate

313 Duality

314 ทฤษฎีพื้นฐานหลักของพีชคณิตพร้อมการพิสูจน์

315 การนำพีชคณิตบูลลีนไปใช้ประโยชน์ในการลดเทอม

32 การพิสูจน์ทฤษฎีพีชคณิตบูลลีน

321 พิสูจน์โดยการแทนค่าตัวคงที่

322 พิสูจน์โดยใช้ตารางความจริง

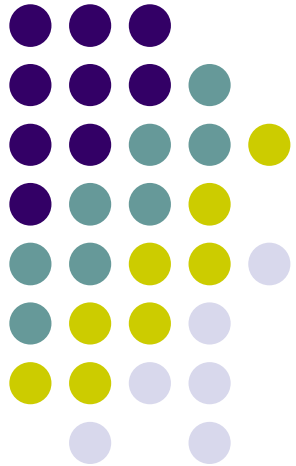
323 พิสูจน์โดยใช้พีชคณิต

33 การลดรูปฟังก์ชัน

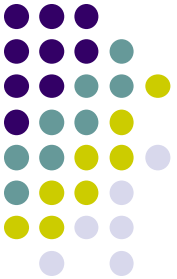
331 การลดรูปโดยใช้พีชคณิตบูลลีน

332 การลดรูปโดยใช้ K-Maps

34 พิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps



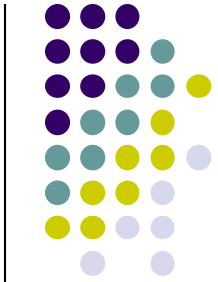
# 31 ทฤษฎีพีชคณิตบูลลีน



## หลักการของพีชคณิตบูลลีน

- ในหัวข้อนี้จะแสดงการวิเคราะห์และสังเคราะห์ พีชคณิตบูลลีน
- ทำการทดสอบตรวจสอบถึงการทำงานของพีชคณิตบูลลีน
- ในปีค.ศ.1849 George Boole ได้แสดงวิธีการทำงานทางลอจิกด้วยหลักการทางพีชคณิตในรูปแบบสูตรทางคณิตศาสตร์หรือที่ทราบกันดีในรูปแบบของพีชคณิตบูลลีน
- เบื้องต้นของรูปแบบทางพีชคณิตบูลลีนเป็นการบอกคำจำกัดความและการกระจาย(Distributive)
- สามารถแสดงให้เห็นพีชคณิตจำนวน6 ข้อ ซึ่งเกิดจากการรวมคำจำกัดความทาง Set และการตั้งสมมุติฐานความจริง สรุปรวมเป็นพีชคณิตบูลลีน ที่จะนำไปช่วยในการลดเทอมในวงจรลอจิก

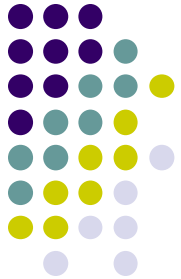
# 311 คำจำกัดความ Poslulate



## Postulate 1 Definition

- a-z เป็นตัวแปร(Variable)ในสมการพีชคณิตบูลลีน
- การกระทำในสมการของพีชคณิตมี AND และ OR ซึ่งใช้คำจำกัดความว่า Operators
- ค่าคงที่เป็นค่า 0 และ 1
- สมการทางพีชคณิตคือ Expression ประกอบด้วยนิพจน์
- การดำเนินการให้พิจารณา Operator AND ก่อน OR ถ้ามีวงเล็บให้กระทำวงเล็บในมาวงเล็บนอกเรื่อยๆ

# Postulate2 Existence of 1 and 0 Element

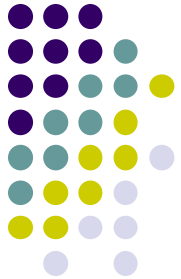


- ความเป็นค่าเดิมของตัวแปรที่กระทำต่อค่าคงที่0 ซึ่งเป็นค่าของเอกลักษณ์ OR และ ค่า1ซึ่งเป็นเอกลักษณ์ของAND จะมีค่าเท่ากับตัวแปรเดิม
- โดยตัวแปรใดๆ ที่กระทำโดยORกับค่าคงที่0 จะมีค่าเท่ากับตัวแปรเดิมเช่น
- P2 (a) $a+0 = a$
- ตัวแปรใดที่กระทำโดยAND ด้วยค่าคงที่1 จะมีค่าเท่ากับตัวแปรเดิมเช่น
- P2 (b) $a.1 = a$

# Postulate3 Commutativity of the + and . Operations

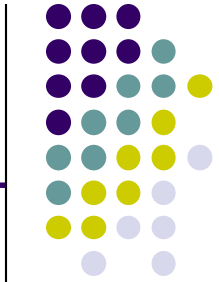
- การสลับที่ของการกระทำในรูปแบบการกระทำ OR และ AND ที่มีค่าเท่ากัน
- P3 (a) $a+b = b+a$
- P3 (b) $a.b = b.a$

# Postulate4 Associativity of the + and .Operations



- การเชื่อมโยงที่มีการทำงานร่วมกันของ OR และ AND สามารถนำมาจับเป็นกลุ่มใหม่และยังคงมีค่าเดิมของกลุ่มOR ใน P4 a และกลุ่ม AND ใน P4b
- P4 (a) $a+(b+c) = (a+b)+c$
- P4 (b) $a.(b.c) = (a.b).c$

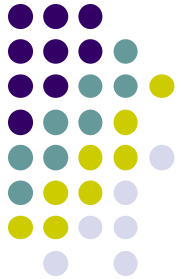
# Postulate5 Distributivity 0f +over . and .over+



- การกระจายในรูปแบบOR เป็น ANDทั้งหมดในรูปแบบ P5a
- กระจายANDเป็น ORทั้งหมดในรูปแบบ P5b

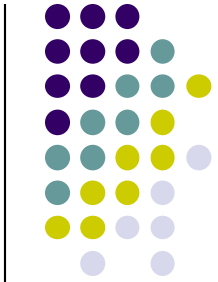
| | | | | |
|---|---|---|---|---|
| P5 | (a) | a+(b.c) | = | (a+b).(a+c) |
| P5 | (b) | a.(b+c) | = | (a.b)+(a.c) |

# Postulate6 Existence of the complement



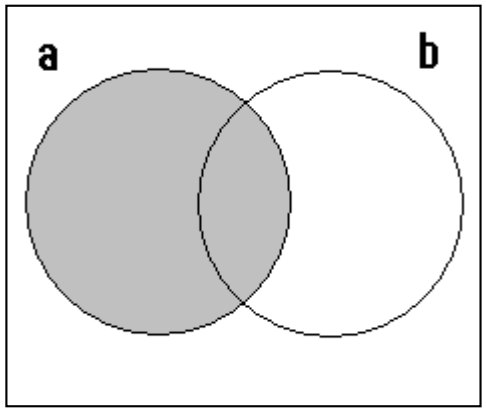
- ตัวแปรใดๆ ที่ OR กับตัวแปรเดียวกันที่มีค่าเป็นคอมพลีเมนต์กัน จะมีค่าเท่ากับ 1 ดังแสดงใน P 6a
- P6 (a) $a+\overline{a} = 1$
- ตัวแปรใดๆ ที่ AND กับตัวแปรเดียวกันที่มีค่าเป็นคอมพลีเมนต์กัน จะมีค่าเป็น0 ดังแสดงใน P 6b
- P6 (b) $a.\overline{a} = 0$

# 312 Venn Diagrams For Postulates

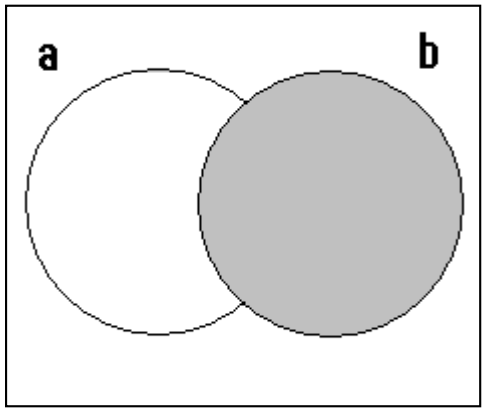


- การยืนยันทฤษฎีพีชคณิตบูลลีน สามารถแสดงทางกราฟในรูปแบบของVenn Diagram
- ลักษณะของพีชคณิตทาง Set ตรงกันทั้งElement และ Operandที่สอดคล้องร่วมกันกับพีชคณิตบูลลีน
- Venn Diagram แสดงในเส้นวงกลม เส้นวงรี,สี่เหลี่ยม
- ในลักษณะของSet ตัวแปรa กับ b โดยมีตัวกระทำ a.b , a+b
- สัญลักษณ์ตัวกระทำ
- การกระทำทาง OR สามารถเขียนในรูปแบบคือ a+b , aUb , aVb
- การกระทำทาง AND สามารถเขียนในรูปแบบคือa.b, anb, aAb

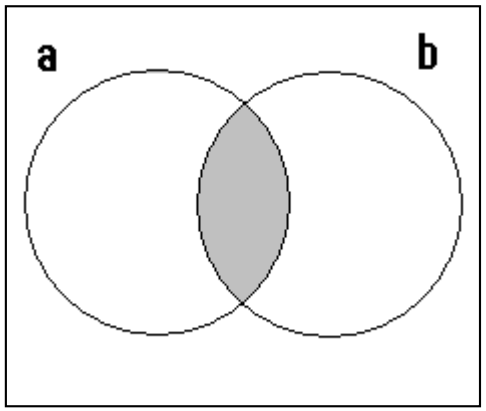
# Venn Diagram



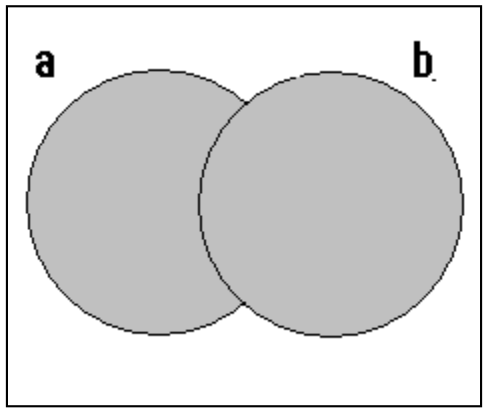

Set *a* is shaded.




Set *b* is shaded.




Set *a* . *b* is shaded.



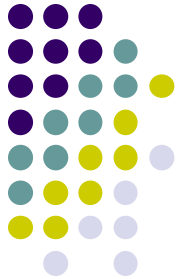

Set *a* + *b* is shaded.

**รูปที่ 3.1 แสดงค่า a, b, a.b, a+b**

# อธิบาย

- รูปที่3.1 เป็นการแสดงตัวอย่างในรูปกราฟของ Venn Diagram
- ตัวแปรa แสดงในวงกลมทางซ้ายมือมีค่าเท่ากับ1 นอกวงกลม Aมีค่าเท่ากับ 0 คือ Complement A
- ในทำนองเดียวกันตัวแปรทางขวามือ b ในวงกลมมีค่าเท่ากับ1 นอกวงกลมมีค่าเท่ากับ 0
- พื้นที่ของ a.b คือพื้นที่ของ aและb ตัดกัน
- พื้นที่ของ a+b คือพื้นที่ของ aและ b รวมกัน

# 313 Duality

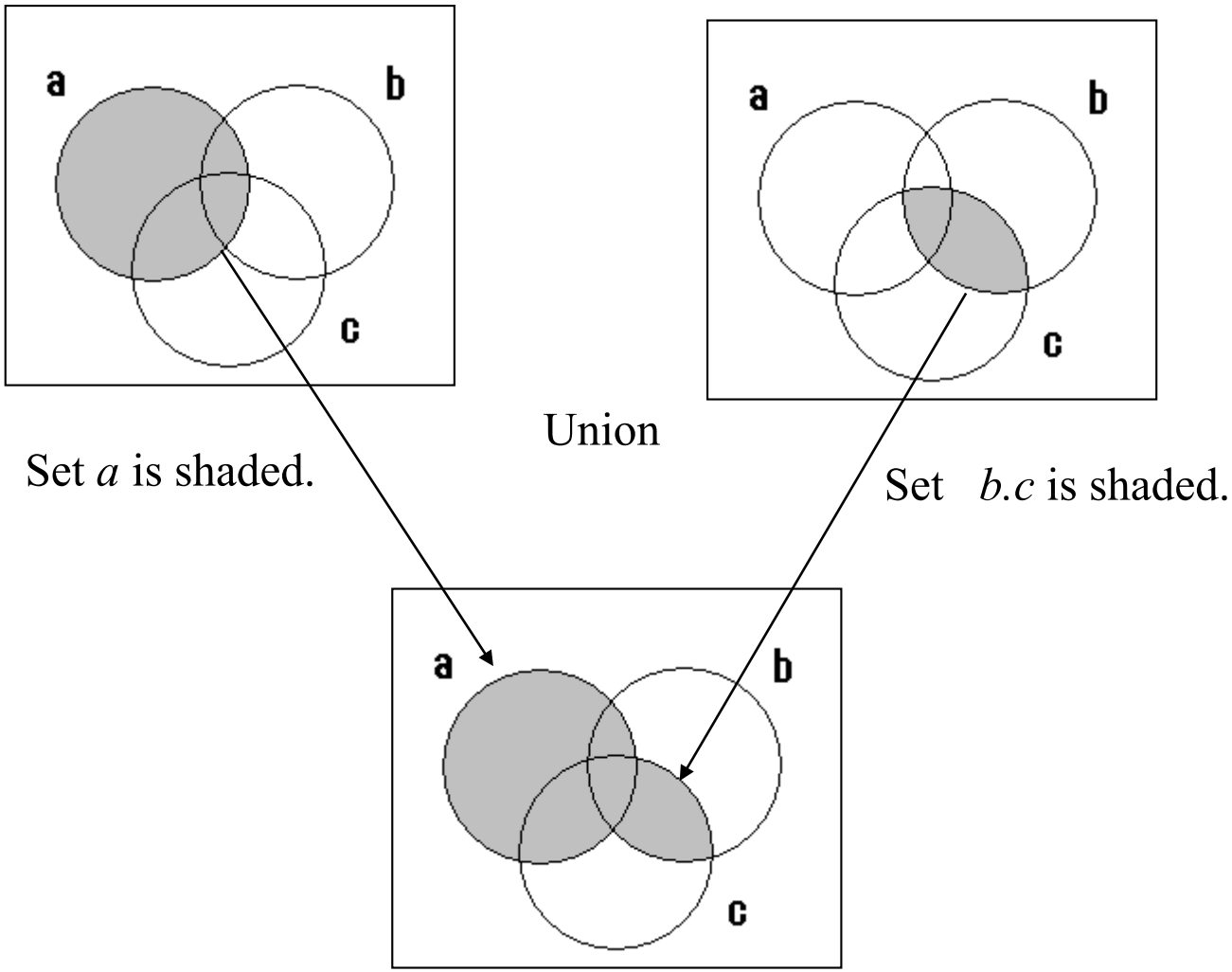


- หลักการของ Duality เป็นส่วนสำคัญในหลักการของพีชคณิตบูลลีน
- สามารถสรุปรูปแบบพีชคณิตของความเป็นคู่ของเครื่องหมายการทำงานของOR กับการทำงานของ AND กับค่าคงที่ 1 และ 0

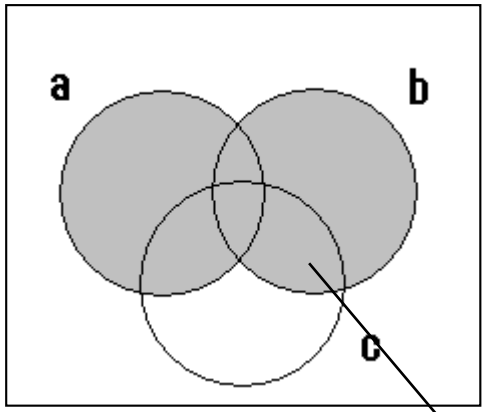
  เช่นการแสดงความเป็นคู่ของเครื่องหมายดังตัวอย่าง นิพจน์

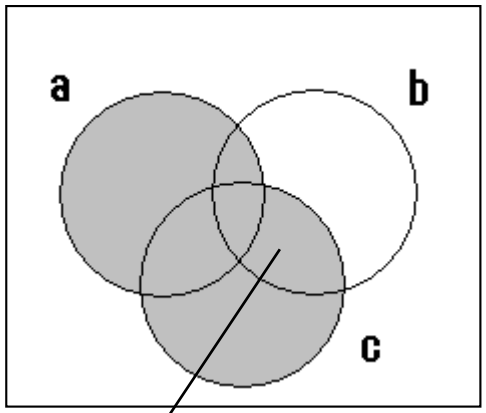
P5 $a+(bc) = (a+b)(a+c)$

Venn Diagram สามารถใช้แสดงยกตัวอย่างในส่วนของPostulate5




รูปที่3.2 แสดงVenn Diagram ของ P5 ด้านซ้าย

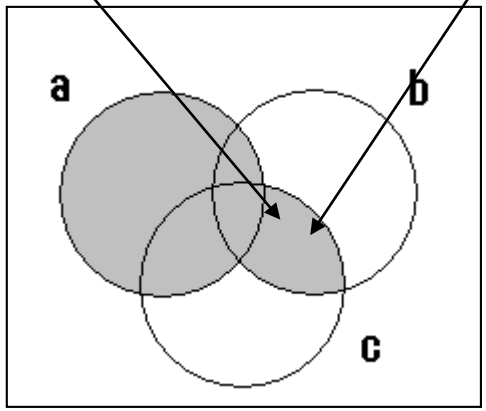



Set $a + b$ is shaded.

Intersection

Set $a + c$ is shaded.

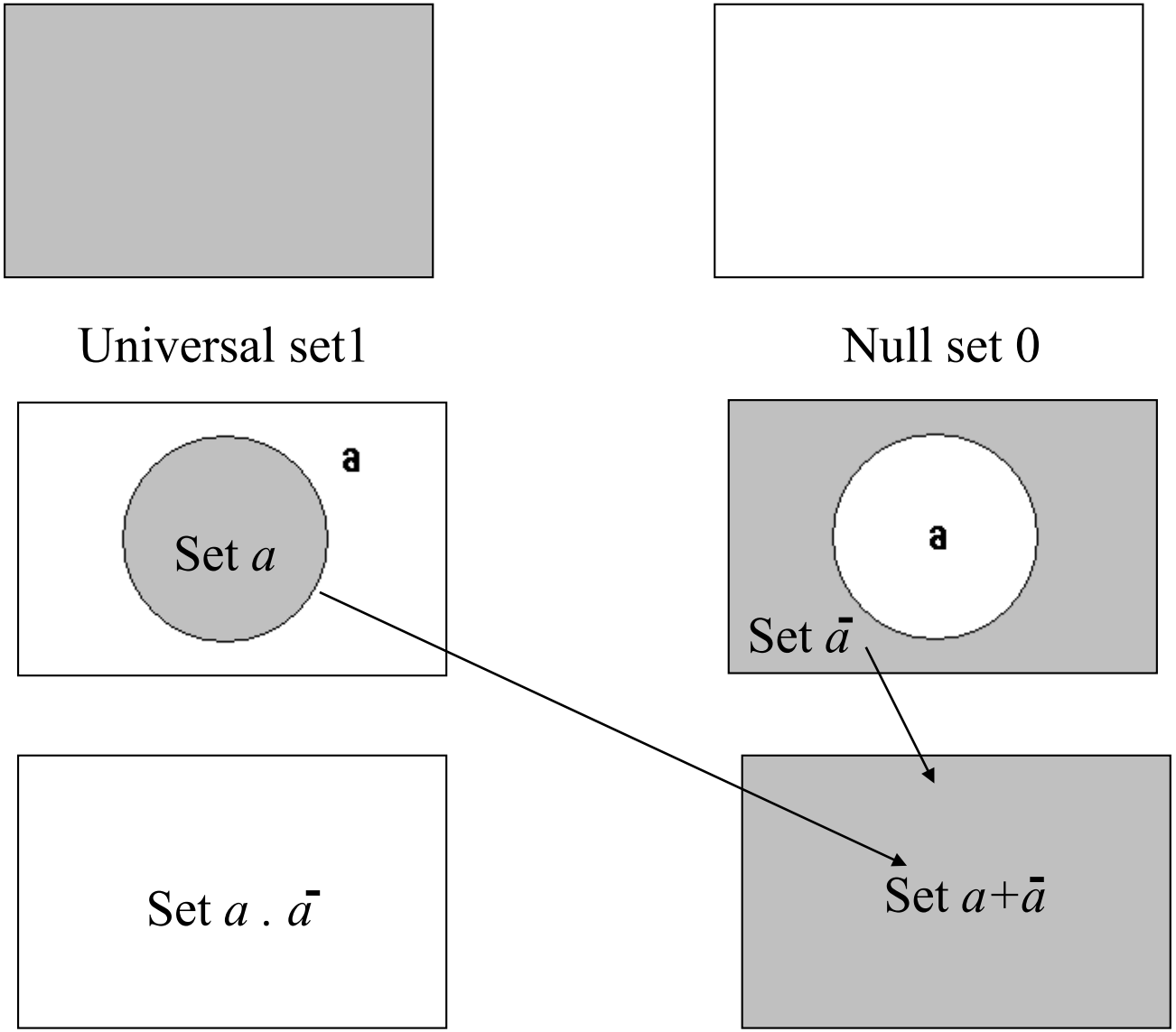



Set $(a + b)(a + c)$ is shaded.

รูปที่3.3 แสดงVenn Diagram ของ P5 ด้านขวา

# อธิบาย

- สามารถนำหัวข้อP5a มาวิเคราะห์ a+b.c จะมีค่าเท่ากับ(a+b)(a+c) ถูกแทนในรูปจะได้พื้นที่aรวมกับพื้นที่ของb,c ตัดกันดังแสดงในรูปที่3.2 ซึ่งมีค่าเท่ากับทางด้านซ้ายสมการP5
- การรวมกันระหว่าง a+b กับ a+c ก็จะมีค่าเท่ากันของa+b.c ดังแสดงในรูปที่ 3.3 ซึ่งมีค่าเท่ากับทางด้านขวาของสมการP5
- จากรูปที่3.2, 3.3 สามารถพิสูจน์ถึง P5a จะมีค่าเท่ากัน
- ในทำนองเดียวกันก็สามารถหาค่า P5b จะมีค่าเท่ากัน

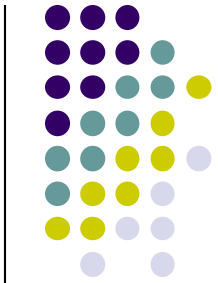



รูปที่3.4 แสดงVenn diagram P6

# อธิบาย

- การตรวจสอบค่าในP6 สามารถใช้ Venn Diagram รูปที่ 3.4
- พื้นที่ทั้งหมดเป็น1 ในกรอบสี่เหลี่ยมไม่ใช่พื้นที่ทั้งหมดเป็น 0
- พื้นที่ของSet a ดังวงกลมใน นอกวงกลมเป็นคอมพลีเมนต์ a
- พื้นที่ของa รวมกับคอมพลีเมนต์ a ($a+\bar{a}$) จะมีค่าเท่ากับ1
- พื้นที่ของa Dotกับคอมพลีเมนต์ a ($a.\bar{a}$) จะมีค่าเท่ากับ0 จะไม่มีส่วนใดที่ตัดกันเลยพื้นที่จึงเป็นศูนย์

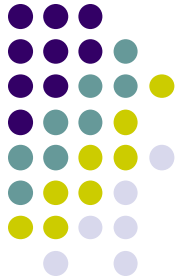
# 314 ทฤษฎีพื้นฐานหลักของพีชคณิตพร้อมถึงการพิสูจน์



- Theorem1 Idempotency
- *(a)* $a+a = a$
- *(b)* $a.a = a$
- พิสูจน์

$$
\begin{aligned}
a+a &= (a+a)1 && P2(b)\\
&= (a+a)(\bar{a}+a) && P6(a)\\
&= (a+\bar{a}a) && P5(a)\\
&= a+0 && P6(b)\\
&= a && P2(a)
\end{aligned}
$$

# Theorem2 Null elements for +and . operators



- *(a) a+1 = 1*
- *(b) a.0 = 0*
- พิสูจน์

$$
\begin{aligned}
a+1 &= (a+1)1 && P2(b)\\
&= 1.(a+1) && P3(b)\\
&= (a+\bar{a})(a+1) && P6(a)\\
&= a+\bar{a}.1 && P5(a)\\
&= a+\bar{a} && P2(b)\\
&= 1 && P6(a)
\end{aligned}
$$

# Theorem3 Involution

$$\bar{\bar{a}} = a$$

จากPostulate 5 $a.\bar{a} = 0$ , $a+\bar{a} = 1$ $\bar{a}$ เป็นคอมพลีเมนต์ของ $a$ ดังนั้นคอมพลีเมนต์ของ $a$ จะมีค่าเท่ากับ $\bar{a}$ ดังตารางแสดงคอมพลีเมนต์ของ1 มีค่าเท่ากับ0 คอมพลีเมนต์0 จะมีค่าเท่ากับ1

| OR | AND | Complement |
|---|---|---|
| $a+0 = a$ | $a.0 = 0$ | $\bar{0} = 1$ |
| $a+1 = 1$ | $a.1 = a$ | $\bar{1} = 0$ |

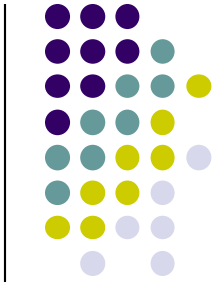
# Theorem4 Absorption

- *(a)* $a+ab = a$
- *(b)* $a(a+b) = a$

- พิสูจน์

$$
\begin{aligned}
a+ab &= a.1+ab && P2(b)\\
&= a(1+b) && P5(b)\\
&= a(b+1) && P3(b)\\
&= a.1 && T2(a)\\
&= a && P2(b)
\end{aligned}
$$

# Theorem 5



$$(a) \quad a + \bar{a}b = a + b.$$

$$(b) \quad a(\bar{a} + b) = ab.$$

พิสูจน์

$$a + \bar{a}b = (a + \bar{a})(a + b) \qquad P5(a)$$

$$= 1.(a + b) \qquad P6(a)$$

$$= (a + b).1 \qquad P3(b)$$

$$= (a + b) \qquad P2(b)$$

# Theorem6

$$(a) \quad ab + a\bar{b} = a.$$

$$(b) \quad (a+b)(a+\bar{b}) = a.$$

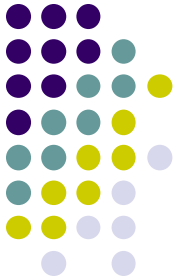
พิสูจน์

$$ab + a\bar{b} = a(b+\bar{b}) \qquad \text{P5(b)}$$

$$= a.1 \qquad \text{P6(a)}$$

$$= a \qquad \text{P2(b)}$$

# Theorem7



(a) $ab + a\bar{b}c = ab + ac.$

(b) $(a + b)(a + \bar{b} + c) = (a + b)(a + c).$

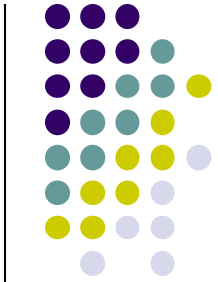
พิสูจน์

$$ab + a\bar{b}c = a(b + \bar{b}c) \qquad \text{P5(b)}$$

$$= a(b + c) \qquad \text{T5(a)}$$

$$= ab + ac \qquad \text{P5(b)}$$

# Theorem8 DeMorgan's theorem



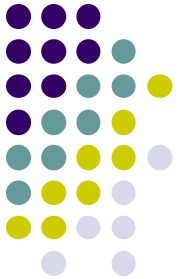
$$(a) \quad \overline{a+b} = \bar{a}.\bar{b}.$$

$$(b) \quad \overline{a.b} = \bar{a}+\bar{b}.$$

พิสูจน์

| | | |
|---|---|---|
| $X.Y$ | $= (a+b)(\bar{a}\bar{b})$ | |
| | $= (\bar{a}\bar{b})(a+b)$ | P3(b) |
| | $= (\bar{a}\bar{b})a + (\bar{a}\bar{b})b$ | P5(b) |
| | $= a(\bar{a}\bar{b}) + (\bar{a}\bar{b})b$ | P3(b) |
| | $= (a\bar{a})\bar{b} + \bar{a}(\bar{b}b)$ | P4(b) |
| | $= 0.\bar{b} + \bar{a}(b.\bar{b})$ | P6(b),P3(b) |
| | $= \bar{b}.0 + \bar{a}.0$ | P3(b),P6(b) |
| | $= 0+0$ | T2(b) |
| | $= 0$ | P2(a) |

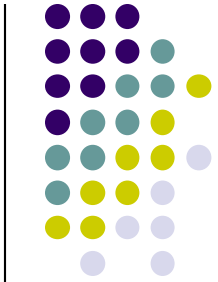
# Theorem8 DeMorgan's theorem(ต่อ)



พิสูจน์(ต่อ)

$$
\begin{aligned}
X + Y &= (a+b) + \bar{a}\bar{b} \\
&= (b+a) + \bar{a}\bar{b} && \text{P3(a)} \\
&= b + (a + \bar{a}\bar{b}) && \text{P4(a)} \\
&= b + (a + \bar{b}) && \text{T5(a)} \\
&= (a + \bar{b}) + b && \text{P3(a)} \\
&= a + (\bar{b} + b) && \text{P4(a)} \\
&= a + (b + \bar{b}) && \text{P3(a)} \\
&= a + 1 && \text{P6(a)} \\
&= 1 && \text{T2(a)}
\end{aligned}
$$

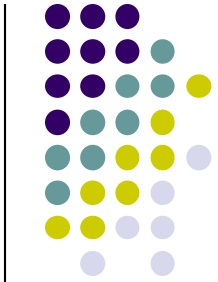
# อธิบาย



- ให้ $x = a+b$ , $\bar{x} = \overline{(a+b)}$
- ใช้ Postulate 6 $x \,.\bar{x} = 0$ , $x+\bar{x} = 1$
- ถ้า $x \,.\, y=0$ , $x + y=1$ ดังนั้น $y = \bar{x}$ เพราะว่า

  คอมพลีเมนต์ของ $x$ เท่ากับ $1$
- ดังนั้น ได้ค่า $y = \bar{a}\,\bar{b}$
- ทดสอบค่า $x \,.\, y$
- ทดสอบค่า $x + y$

# Theorem9 Consensus



(a) $ab + \bar{a}c + bc = ab + \bar{a}c.$

(b) $(a+b)(\bar{a}+c)(b+c) = (a+b)(\bar{a}+c).$

พิสูจน์

$ab + \bar{a}c + bc = ab + \bar{a}c + 1.bc$ P2(b)

$= ab + \bar{a}c + (a + \bar{a})bc$ P6(a)

$= ab + \bar{a}c + abc + \bar{a}bc$ P5(b)

$= (ab + abc) + (\bar{a}c + \bar{a}cb)$

$= ab + \bar{a}c$ T4(a)

# 315 การนำทฤษฎีพีชคณิตบูลลีนไปใช้ประโยชน์ในการลดเทอม

## การนำTheorem4 ไปใช้ประโยชน์

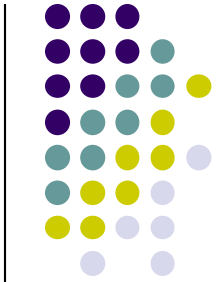
- ตัวอย่างการนำไปใช้งาน

$$(X+Y)+(X+Y)Z = X+Y \qquad T4(a)$$

$$A\bar{B}(A\bar{B}+\bar{B}C) = A\bar{B} \qquad T4(b)$$

$$A\bar{B}C+\bar{B} = \bar{B} \qquad T4(a)$$

# การนำTheorem5 ไปใช้ประโยชน์



- ตัวอย่างการนำไปใช้งาน
  - $B + A\overline{B}\,\overline{C}D = B + A\overline{C}D$ T5(a)
  - $\overline{Y}(X + Y + Z) = \overline{Y}(X + Z)$ T5(b)
  - $(X + Y)\overline{((X + Y) + Z} = (X + Y)Z$ T5(b)
  - $\overline{AB} + (AB)C\,\overline{D} = AB + C\,\overline{D}$ T5(a)

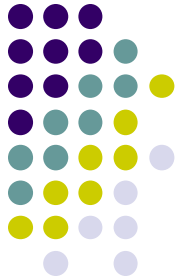
# การนำTheorem6 ไปใช้ประโยชน์

- ตัวอย่างการนำไปใช้งานได้โดยตรง

$$ABC + A\bar{B}C = AC \qquad T6(a)$$

$$(AB + B + C)(AD + \overline{(B + C)}) = AD \qquad T6(b)$$

# การนำTheorem7 ไปใช้ประโยชน์



- การนำไปลดรูปสมการเพื่อใช้งานในการออกแบบ

- $xy + x\bar{y}(\bar{w} + \bar{z}) = xy + x(\bar{w} + \bar{z})$ T7(a)

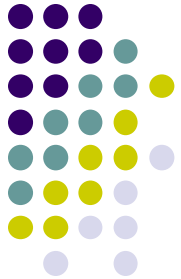
- $(\bar{\bar{x}}\bar{y} + z)(w + \bar{\bar{x}}\bar{y} + \bar{z}) = (\bar{\bar{x}}\bar{y} + z)(w + \bar{\bar{x}}\bar{y})$ T7(b)

- $(\bar{A} + \bar{B} + \bar{C})(\bar{B} + C)(A + \bar{B}) = (\bar{A} + \bar{B})(\bar{B} + C)(A + \bar{B})$ T7(b)

  $= \bar{B}(\bar{B} + C)$ T6(b)

  $= \bar{B}$ T4(b)

# การนำTheorem7 ไปใช้ประโยชน์



ตัวอย่างการนำไปใช้งาน(ต่อ)

- $w\bar{y} + w\bar{x}y + wxyz + wx\bar{z} = w\bar{y} + w\bar{x}y + wxy + wx\bar{z}$ T7(a)

$= w\bar{y} + wy + wx\bar{z}$ T6(a)

$= w + wx\bar{z}$ T6(a)

$= w$ T4(a)

# ตัวอย่างการใช้ในการลดรูปสมการลอจิก

$$(\overline{W} + \overline{X} + \overline{Y} + \overline{Z})(\overline{W} + \overline{X} + Z)$$

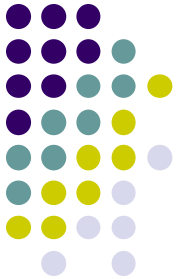
$$= (\overline{W} + \overline{X} + Y + \overline{Z})(\overline{W} + \overline{X} + Y + Z).$$

$$= (\overline{W} + \overline{X} + \overline{Y})(\overline{W} + \overline{X} + Y + \overline{Z})(\overline{W} + \overline{X} + Y + Z) \quad \text{T6(b)}$$

$$= (\overline{W} + \overline{X} + \overline{Y})(\overline{W} + \overline{X} + Y+) \quad \text{T6(b)}$$

$$= (\overline{W} + \overline{X}) \quad \text{T6(b)}$$

# การนำTheorem8 ไปใช้ประโยชน์



ตัวอย่างการนำไปใช้งาน

*Complement the expression* $a + bc.$

$$\overline{a + b.c} = \overline{a + (b.c)}$$

$$= \bar{a}.\overline{(b.c)}$$

$$= \bar{a}.(\bar{b} + \bar{c})$$

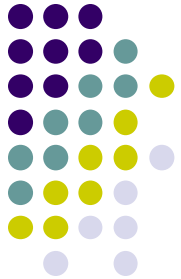
$$= \bar{a}\bar{b} + \bar{a}\bar{c}$$

*Note that :* $\overline{a + b.c} \neq \bar{a}.\bar{b} + \bar{c}$

$$\overline{X + \bar{Y}} = \bar{X}.\bar{\bar{Y}} \qquad \text{T8(a)}$$

$$= \bar{X}.Y \qquad \text{T3}$$

# การนำTheorem8 ไปใช้ประโยชน์



ตัวอย่างการนำไปใช้งาน(ต่อ)

$$\overline{a(b+z(x+\bar{a}))} = \bar{a} + \overline{(b+z(x+\bar{a}))} \qquad \text{T8(b)}$$

$$= \bar{a} + \bar{b}\,\overline{(z(x+\bar{a}))} \qquad \text{T8(a)}$$

$$= \bar{a} + \bar{b}(\bar{z} + \overline{(x+\bar{a})}) \qquad \text{T8(b)}$$

$$= \bar{a} + \bar{b}(\bar{z} + \bar{x}.\bar{\bar{a}}) \qquad \text{T8(a)}$$

$$= \bar{a} + \bar{b}(\bar{z} + \bar{x}a) \qquad \text{T3}$$

$$= \bar{a} + \bar{b}(\bar{z} + \bar{x}) \qquad \text{T5(a)}$$

# การนำTheorem8 ไปใช้ประโยชน์

ตัวอย่างการหาค่าคอมพลีเมนต์ของ $a(b+c)+\bar{a}\,b.$

$$\overline{a(b+c)+\bar{a}\,b} = \overline{a\,b+a\,c+\bar{a}\,b} \qquad \text{P5(b)}$$

$$= \overline{b+a\,c} \qquad \text{T6(a)}$$

$$= \bar{b}(\overline{ac}) \qquad \text{T8(a)}$$

$$= \bar{b}(\bar{a}+\bar{c}) \qquad \text{T8(b)}$$

# การนำTheorem9 ไปใช้ประโยชน์

ตัวอย่างการนำไปใช้งาน

- $AB + \bar{A}CD + BCD = AB + \bar{A}CD$ T9(a)

- $(a+\bar{b})(\bar{a}+c)(\bar{b}+c) = (a+\bar{b})(\bar{a}+c)$ T9(b)

- $ABC + \bar{A}D + \bar{B}D + CD = ABC + (\bar{A}+\bar{B})D + CD$ P5(b)

  $= ABC + \overline{AB}D + CD$ T8(b)

  $= ABC + \overline{AB}D$ T9(a)

  $= ABC + (\bar{A}+\bar{B})D$ T8(b)

  $= ABC + \bar{A}D + \bar{B}D$ P5(b)

# The principal identities of Boolean algebra

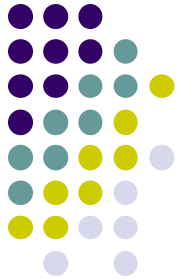
| | | |
|---|---|---|
| $a+b=b+a$ | $ab=ba$ | *Commutativity* |
| $a+(bc)=(a+b)(a+c)$ | $a(b+c)=(ab)+(ac)$ | *Distributivity* |
| $a+(b+c)=(a+b)+c$ $=a+b+c$ | $a(bc)=(ab)c$ $=abc$ | *Associativity* |
| $a+a=a$ | $aa=a$ | *Idempotency* |
| $a+\bar{a}=1$ | $a\bar{a}=0$ | *Complement* |
| $1+a=1$ | $0a=0$ | |
| $0+a=a$ | $1a=a$ | *Identity* |
| $\overline{(\bar{a})}=a$ | | *Involution* |
| $a+ab=a$ | $a(a+b)=a$ | *Absorption* |
| $a+\bar{a}b=a+b$ | $a(\bar{a}+b)=ab$ | *Simplification* |
| $\overline{(a+b)}=\bar{a}\bar{b}$ | $\overline{(ab)}=\bar{a}+\bar{b}$ | *DeMorgan'sLaw* |

# 32 การพิสูจน์ทฤษฎีพีชคณิตบูลลีน

## สามารถทำได้หลายวิธี

- ใช้การแทนด้วยค่าคงที่
- ใช้ตารางความจริง
- ใช้วิธีทางพีชคณิต

# อธิบาย



- ทฤษฎีของบูลลีน เพื่อศึกษาและทำให้เกิดความเข้าใจทางพีชคณิตบูลลีนได้ดียิ่งขึ้นจึงจำเป็นต้องพิสูจน์เพื่อหาค่าที่ได้ว่ามีค่าตรงกับทฤษฎีที่กำหนดไว้
- การพิสูจน์สามารถหาได้หลายวิธี เช่น

  - 1. หาจากการแทนค่าคงที่ลงไปในสมการแล้วหาค่าผลลัพธ์ที่เกิดขึ้น

  - 2. หาจากการใช้ตารางความจริงกำหนดค่าที่ต้องการหา

  - 3. หาจากการใช้พีชคณิตบูลลีน

ในสัปดาห์นี้จะทำการพิสูจน์ให้เห็นทั้ง 3 วิธี เพื่อเป็นแนวทางที่นักศึกษาจะได้นำไปใช้ในการพิสูจน์ทุกทฤษฎี

นักศึกษาลองนำไปทดสอบใช้ทั้ง 3วิธี สรุปได้ว่าวิธีไหนเหมาะที่สุดในการแก้ปัญหาโจทย์ต่างๆ ที่เกิดขึ้นเมื่อนักศึกษานำไปใช้งานจริง

# 321 พิสูจน์โดยการแทนค่าตัวคงที่

## ตัวอย่างการแทนค่าคงที่พิสูจน์ T 1(a)

$a + a = a$

การพิสูจน์ต้องแยกเป็นกรณีดังนี้

**กรณีที่ 1 a = 0**

$a + a = a$

$0 + 0 = 0$

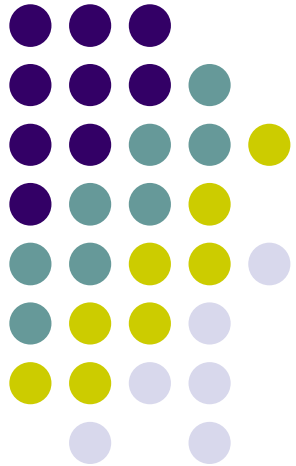
$0 = 0$ ← เท่ากัน

**กรณีที่ 2 a = 1**

$a + a = a$

$1 + 1 = 1$

$1 = 1$ ← เท่ากัน



$a + \bar{a} = 1$ จาก P 6(a)

## การพิสูจน์ต้องแยกเป็นกรณีดังนี้

## กรณีที่ 1 a = 0

$a + \bar{a} = 1$

$0 + \bar{0} = 1$

$0 + 1 = 1$

$1 = 1$ ⟵ เท่ากัน

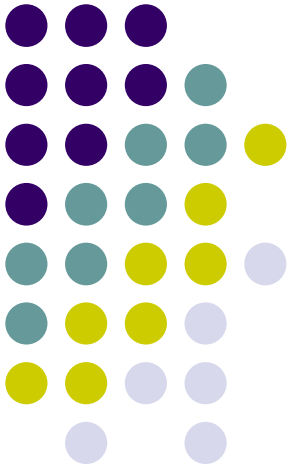
## กรณีที่ 1 a = 1

$a + \bar{a} = 1$

$1 + \bar{1} = 1$

$1 + 0 = 1$

$1 = 1$ ⟵ เท่ากัน



# 322 พิสูจน์โดยใช้ตารางความจริง

$a + ( b + c ) = ( a + b ) + c$

| a | b | c | ( b + c ) | a + ( b + c) | ( a + b) | ( a + b )+c |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

เท่ากัน

# พิสูจน์ทฤษฎีที่ 3(a) โดยการใช้ตารางความจริง

$$a + b \cdot c = (a + b) \cdot (a + c)$$

| a | b | c | b . C | a +b .c | ( a + b ) | ( a +c ) | (a+b).(a+c) |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

เท่ากัน

# พิสูจน์ว่า De Morgan's Theorem

| $A$ | $B$ | $A+B$ | $\overline{A+B}$ | $\overline{A}$ | $\overline{B}$ | $\overline{A}.\overline{B}$ | $\overline{A+B}$ | $\overline{A.B}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

# 323 พิสูจน์โดยใช้พีชคณิต

- จากการพิสูจน์ค่าฟังก์ชันการทำงานของวงจรเกตในทฤษฎีที่ 1-9 ของพีชคณิตบูลลีนถึงการลดรูปวงจรลงได้ เป็นการพิสูจน์โดยทางพีชคณิต

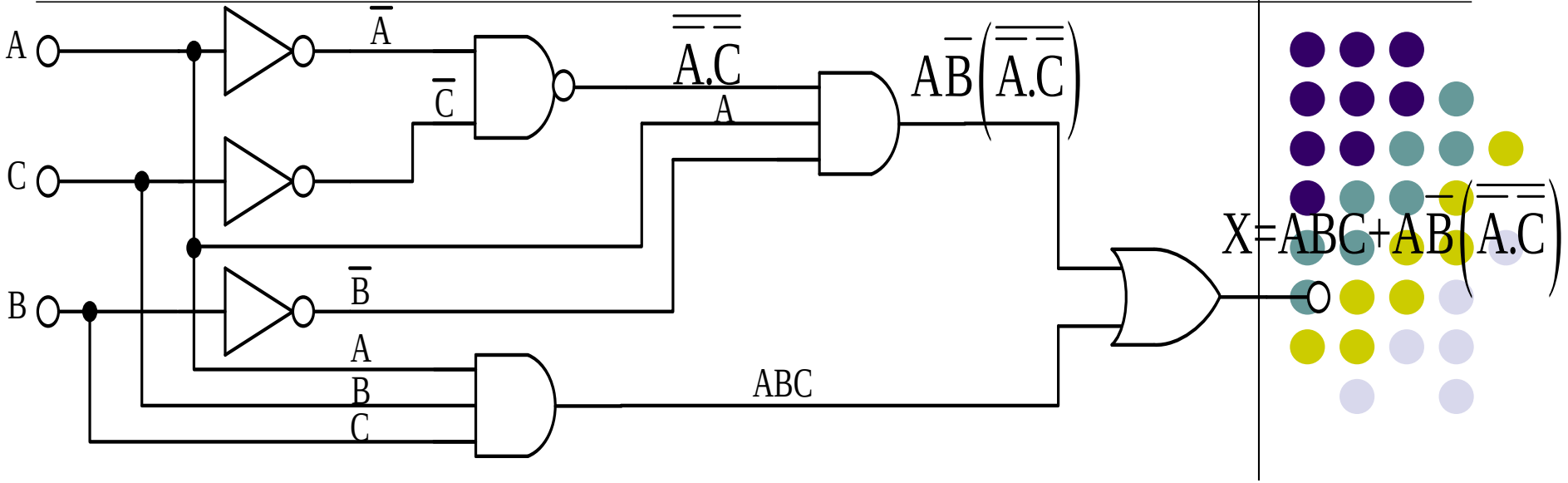
- การนำพีชคณิตบูลลีนมาใช้ในการตรวจสอบค่าสัญญาณทางเอาต์พุตของวงจรเกตที่มีความสลับซับซ้อนให้ง่ายต่อการวิเคราะห์วงจร

- การนำวงจรเกตที่มีรูปแบบต่างๆมาใช้ในการเปลี่ยนรูปแบบตามที่ต้องการได้ เช่นในรูปแบบการทำงานของ NAND Gatesหรือ NOR Gates โดยใช้ทฤษฎี De Morgan ‘s ที่ 8

# 3.3 การลดรูปฟังก์ชัน

**ตัวอย่างที่ 3.1 การใช้พีชคณิตบูลลีนลดรูปสมการลอจิก**

จากรูปวงจรจงใช้พีชคณิตบูลลีน แก้ปัญหาของวงจรลอจิกที่กำหนด ให้อยู่ในรูปที่ง่ายขึ้น เทอมน้อยกว่าเดิม หรือมีตัวแปรน้อยลง โดยมีการทำงานของวงจรลอจิกเหมือนเดิม




(a)

# ตัวอย่างที่ 3.2 การลดรูปโดยใช้ทฤษฎีพีชคณิต

$X = A B C + A \overline{B} (\overline{A} . \overline{C})$

$= A B C + A \overline{B} (\overline{A} + \overline{C})$

[ ใช้ทฤษฎีของ De Morgan ลดรูปก่อน

$= A B C + A \overline{B} ( A + C )$ T3

$= A B C + A \overline{B} A + A \overline{B} C$

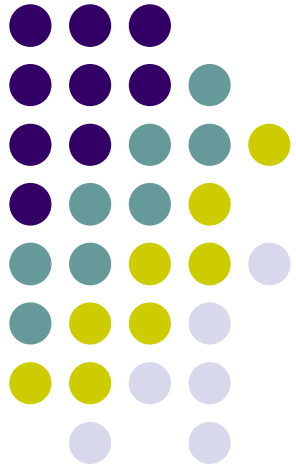
$= A B C + A \overline{B} + A \overline{B} C$ T1

$= A C ( B + \overline{B} ) + A \overline{B}$ T4b

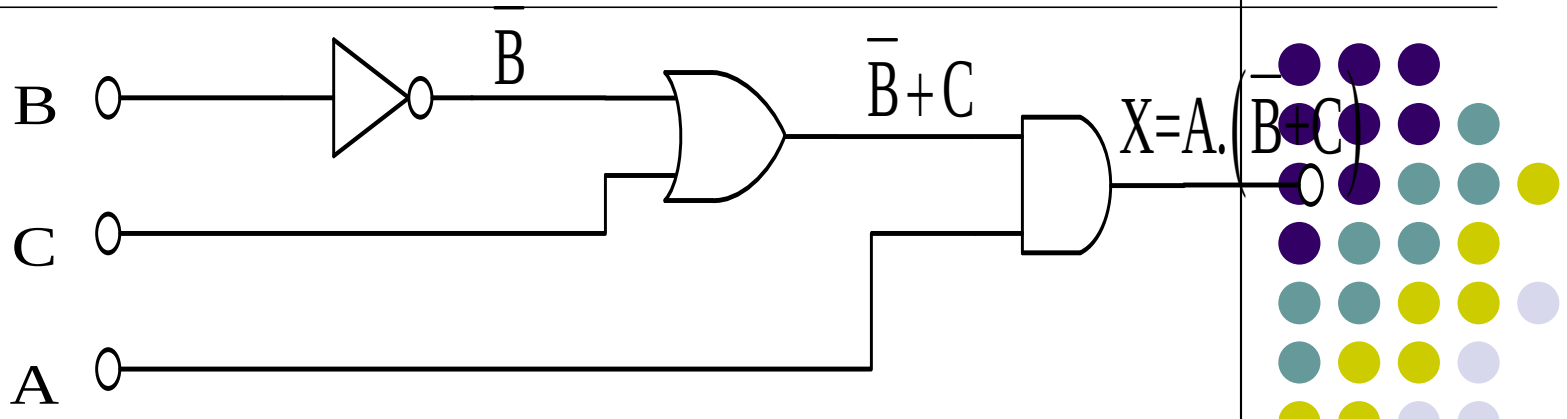
$= A . C . 1 + A \overline{B}$ P6a

$= A C + A \overline{B}$ P2

$X = A(C+\overline{B})$ จะได้ลอจิกดังรูป (b)



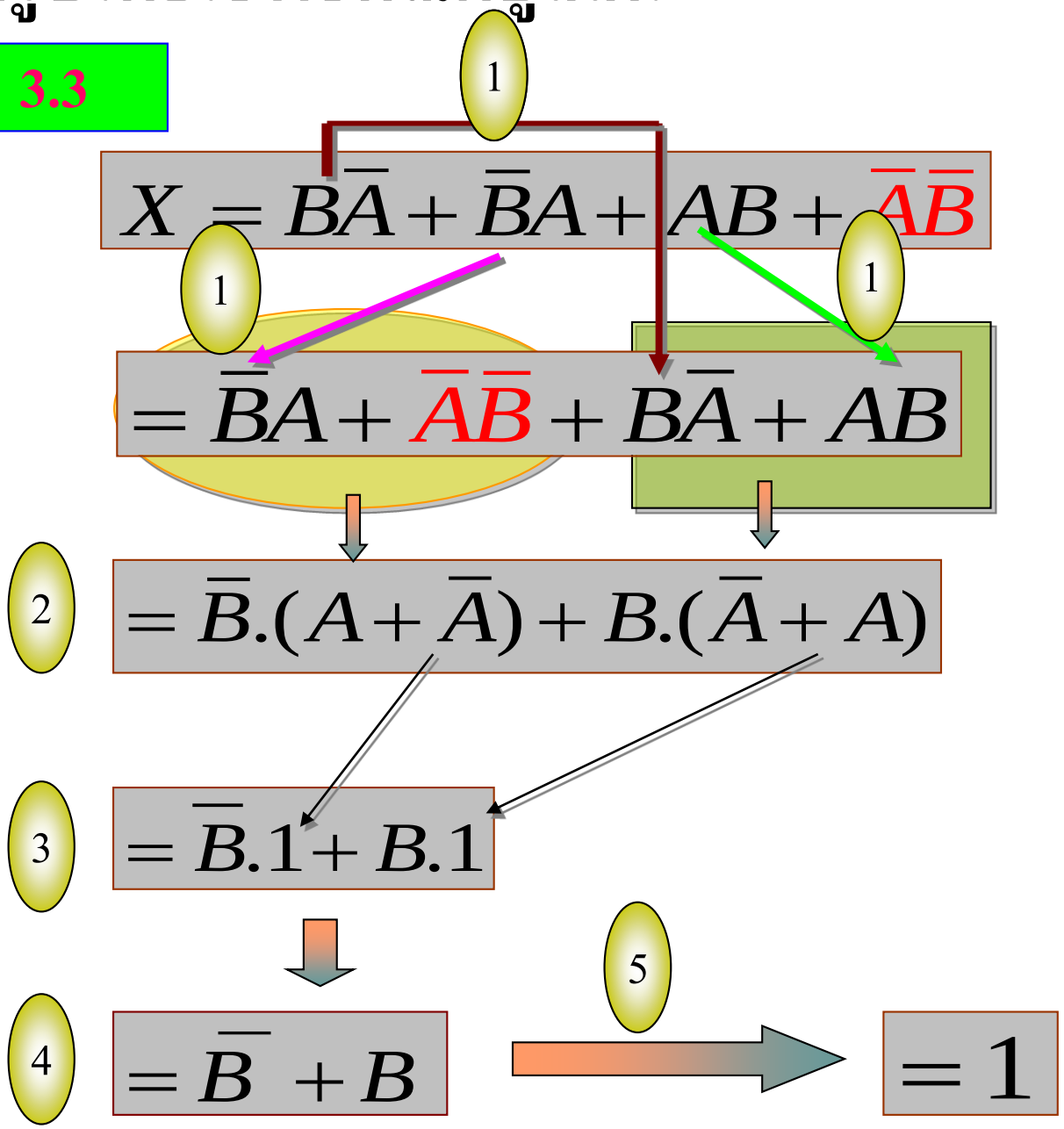
จากสมการ $ABC + \overline{A}\overline{C} + A\overline{B}(\overline{\overline{A}\overline{C}})$ ลดรูปเหลือเพียงวงจรง่ายๆ เป็นดังสมการลอจิก $X = A(C + \overline{B})$ สามารถนำมาเขียนเป็นวงจรที่ให้ค่าเอาต์พุตเหมือนกัน




(b)

# 331 การลดรูปโดยใช้พีชคณิตบูลลีน

**ตัวอย่างที่ 3.3**

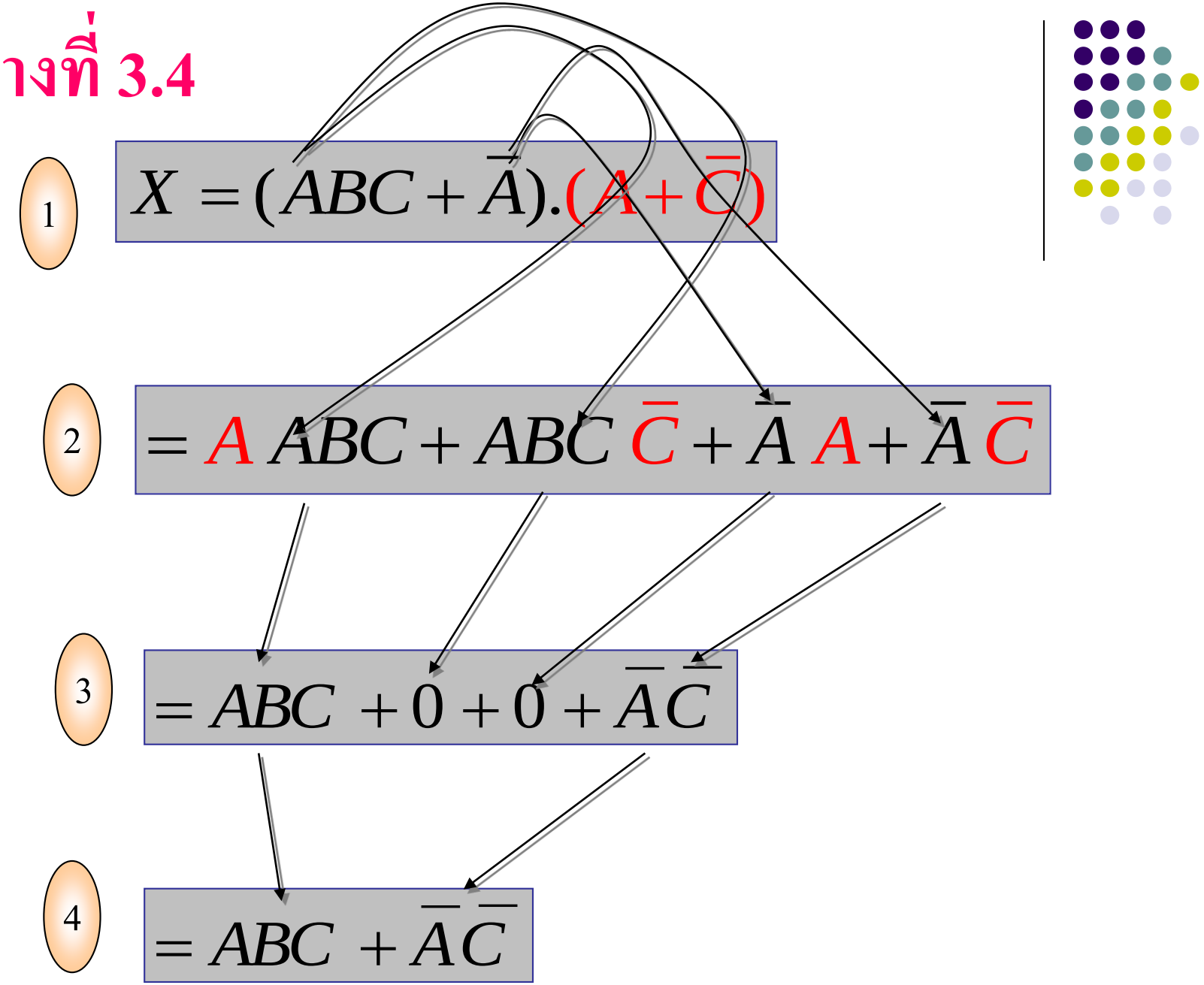


# อธิบาย

- ตัวอย่างที่3.3 การลดรูปแบบใช้พีชคณิตบูลลีน

- ขั้นตอนที่1 สลับที่เทอมต่างๆ เพื่อให้ง่ายต่อการลดรูปโดยใช้กฎสลับที่
- ขั้นตอนที่สอง พิจารณาที่วงกลมทึบ กับสี่เหลี่ยมทึบใต้สมการ ใช้กฎการกระจาย
- ขั้นตอนที่ 3 กับขั้นตอนที่ 4 ใช้ P2(b)
- ขั้นตอนที่ 5 คือคำตอบมีค่าเท่ากับ 1 ใช้ P6(a)

# ตัวอย่างที่ 3.4


1
$X = (ABC + \bar{A}).(A + \bar{C})$
2
$= A\,ABC + AB\bar{C}\,\bar{C} + \bar{A}\,A + \bar{A}\,\bar{C}$
3
$= ABC + 0 + 0 + \bar{A}\bar{C}$
4
$= ABC + \bar{A}\bar{C}$

# อธิบาย

- ตัวอย่างที่ 3.4 การลดรูปโดยใช้พีชคณิตบูลลีน

- ขั้นตอนที่ 1 ใช้การกระจาย P5(b)

- ขั้นตอนที่ 2 ใช้ทฤษฎีที่ 1(a),T1(b)

- ขั้นตอนที่ 3 ใช้ทฤษฎีที่ P6 (b)

- ขั้นตอนที่ 4 สามารถลดรูปได้เป็น 2 เทอม คือ $(ABC + \overline{A}\,\overline{C})$

3.3.2 การลดรูปโดยใช้ **แผนผังคาร์โนห์** (K-Map)

- การลดรูปโดยใช้ k-map ในรูปแบบ minterm

- การลดรูปแบบใช้ k-map ในรูปแบบ Maxterm

# 332 การลดรูปโดยใช้ K-map

| | B | A |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

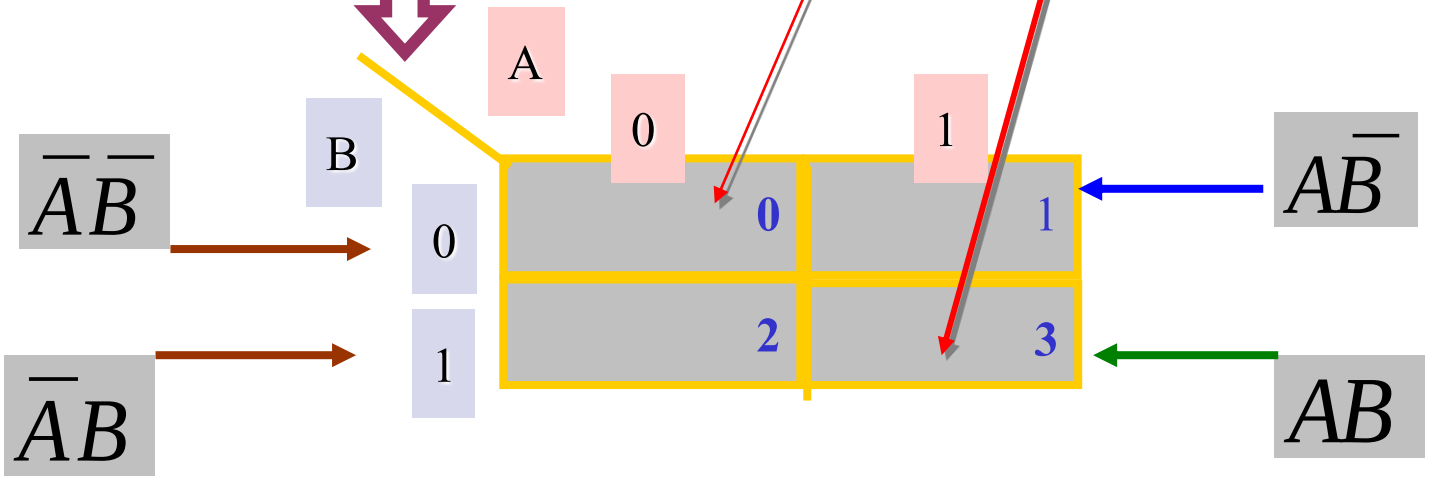
$\overline{A}\,\overline{B}$

$A\overline{B}$

$\overline{A}B$

$AB$

แผนผังคาร์โนห์แบบ minterm



# อธิบาย

- แผนผังคาร์โนห์แบบ minterm ที่ใช้ลดรูปสมการลอจิก 2 ตัวแปร

  - ค่า A กับค่า B เป็น 0,0 เขียนเป็น $\overline{A}\,\overline{B}$

  - ค่า A กับค่า B เป็น 1,0 เขียนเป็น $A\overline{B}$

  - ค่า A กับค่า B เป็น 0,1 เขียนเป็น $\overline{A}B$

  - ค่า A กับค่า B เป็น 1,1 เขียนเป็น $AB$

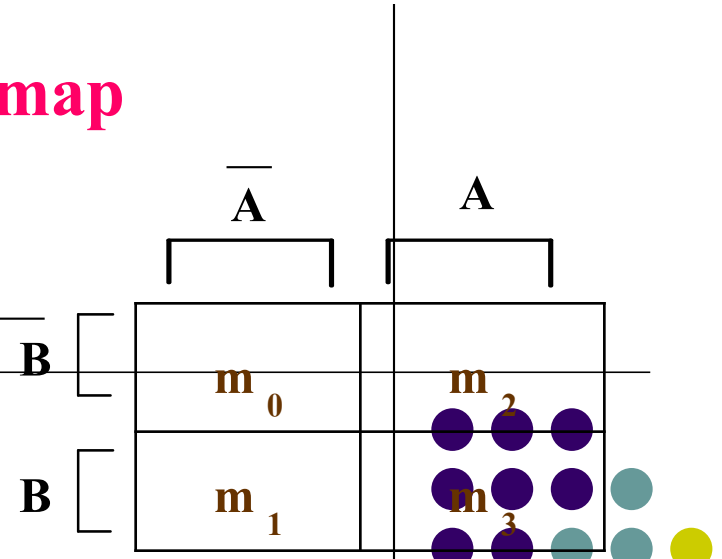
  ตาราง k-map เป็นดังรูป

# การใช้แผนผังคาร์นอลดรูปสมการลอจิก

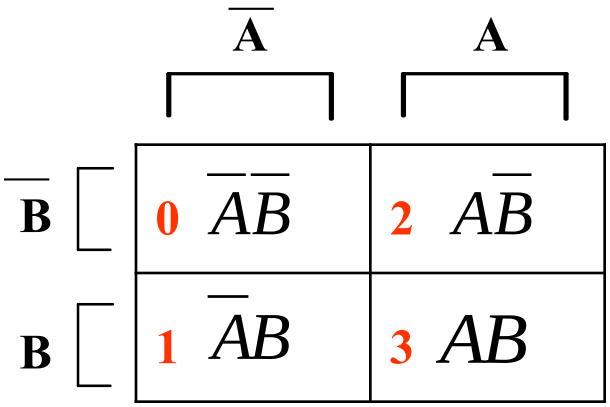
| A | B | X |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

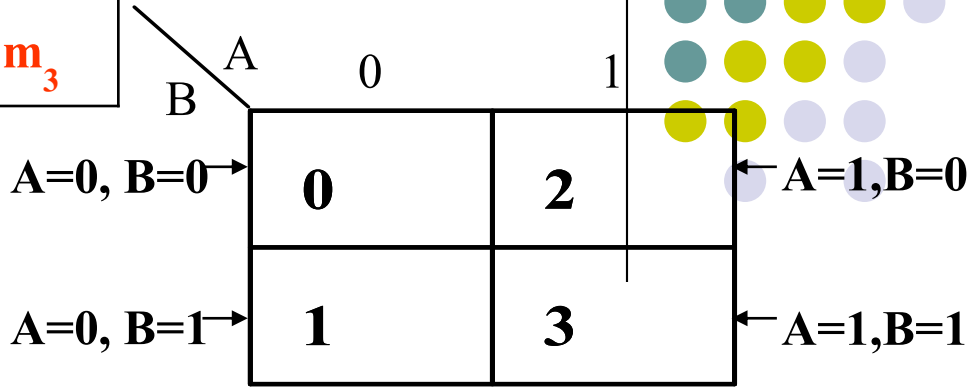
## รูปแบบของ K map

| Min term |
|---|
| $\overline{A}\,\overline{B} = m_0$ |
| $\overline{A}\,B = m_1$ |
| $A\,\overline{B} = m_2$ |
| $A\,B = m_3$ |




K map ชนิด 2 ตัวแปร



# อธิบาย

- ผัง K-map ในรูปแบบ minterm 2 ตัวแปร ตารางมีค่าเท่ากับ $2^n$ n คือตัวแปร

– ค่าที่เกิดขึ้น 4 ค่าที่ไม่ซ้ำกันดังนี้

$$\overline{A}\,\overline{B} = m_0$$

$$\overline{A}B = m_1$$

$$A\overline{B} = m_2$$

$$AB = m_3$$

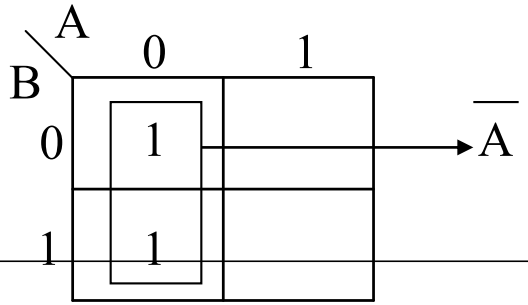
พิจารณาทางแนวตั้งตารางmap เป็นค่า $\overline{A}, A$

พิจารณาทางแนวนอนตารางmap เป็นค่า $\overline{B}, B$

# การลดรูป Switching function โดยใช้ K map ชนิด 2 ตัวแปร

## รูปแบบMinterm

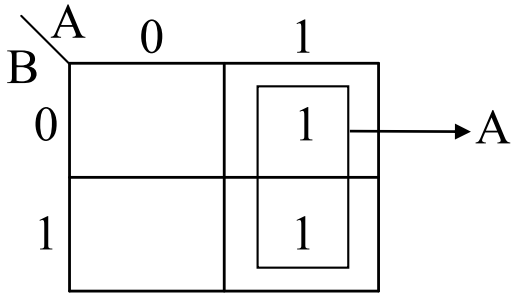
จงลดรูป $f(A,B) = \overline{A}\ \overline{B} + \overline{A}\ B$




$$f(A,B) = \overline{A}\ \overline{B} + \overline{A}\ B$$

$$= \overline{A}$$

จงลดรูป $f(A,B) = A\ \overline{B} + A\ B$




$$f(A,B) = A\ \overline{B} + A\ B$$

$$= A$$

# อธิบาย

- ตัวอย่าง

- ตาราง k-map ที่เขียนจากค่า minterm
- สามารถลดรูปจากการจับ 4 ได้ค่าเอาต์พุต 1
- จับคู่ 2 ได้ค่าเอาต์พุต 1 ตัวแปร
- จับคู่ 1 ได้ค่าเอาต์พุต 2 ตัวแปร
- ตัวแปรที่เลือกจับคู่แล้วสามารถนำใช้ร่วมในการจับคู่ใหม่ได้
- ตารางบนเป็นการเลือกจับคู่ของค่าเลข 0กับค่าเลข 1 เข้าด้วยกัน ค่าเอาต์พุตได้ $\overline{A}$
- ส่วนตารางล่างเป็นการจับคู่ของค่าเลข 2กับค่าเลข 3 เข้าด้วยกัน ค่าเอาต์พุตได้ $A$

## แผนผังคาร์โนห์ 2 ตัวแปรรูปแบบMaxterm

| | B | A |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

$A+B$

$\overline{A}+B$

$A+\overline{B}$

$\overline{A}+\overline{B}$

| B \ A | 0 | 1 |
|---|---|---|
| 0 | $A+B$ | $\overline{A}+B$ |
| 1 | $A+\overline{B}$ | $\overline{A}+\overline{B}$ |

# อธิบาย

3.4 พิสูจน์การลดรูปโดยใช้พีชคณิตเทียบกับ K-Maps

- แผนผังคาร์โนห์แบบ Maxterm

-ค่า A กับค่าB เป็น 0,0 เขียนเป็น $A+B$

-ค่า A กับค่าB เป็น 1,0 เขียนเป็น $\overline{A}+B$

-ค่า A กับค่าB เป็น 0,1 เขียนเป็น $A+\overline{B}$

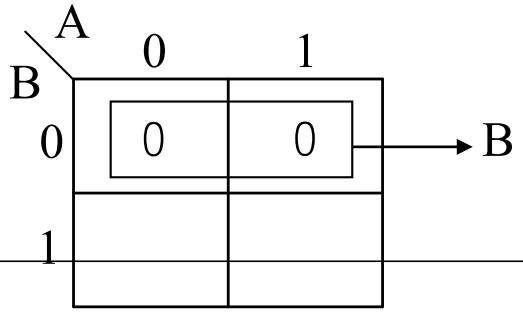
-ค่า A กับค่าB เป็น 1,1 เขียนเป็น $\overline{A}+\overline{B}$

ตารางk-map เป็นดังรูป

# การลดรูป Switching function โดยใช้ K map ชนิด 2 ตัวแปร

## รูปแบบ Maxterm

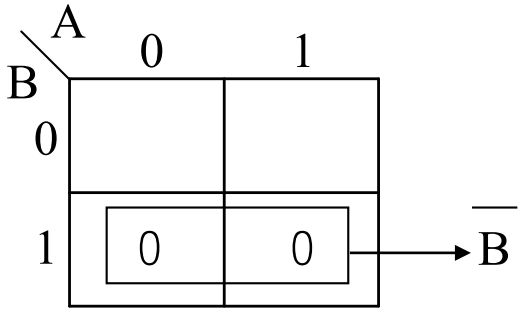
จงลดรูป $f(A, B) = (A + B)(\overline{A} + B)$




$f(A, B) = (A + B)(\overline{A} + B)$

$= B$

จงลดรูป $f(A, B) = (A + \overline{B})(\overline{A} + \overline{B})$




จงลดรูป $f(A, B) = (A + \overline{B})(\overline{A} + \overline{B})$

$= \overline{B}$

# อธิบาย

- ตัวอย่างนี้เป็นการลดรูป Maxterm 2 ตัวแปร ในรูปแบบ k-map

- หลักการจับคู่เหมือนกับการจับแบบ minterm

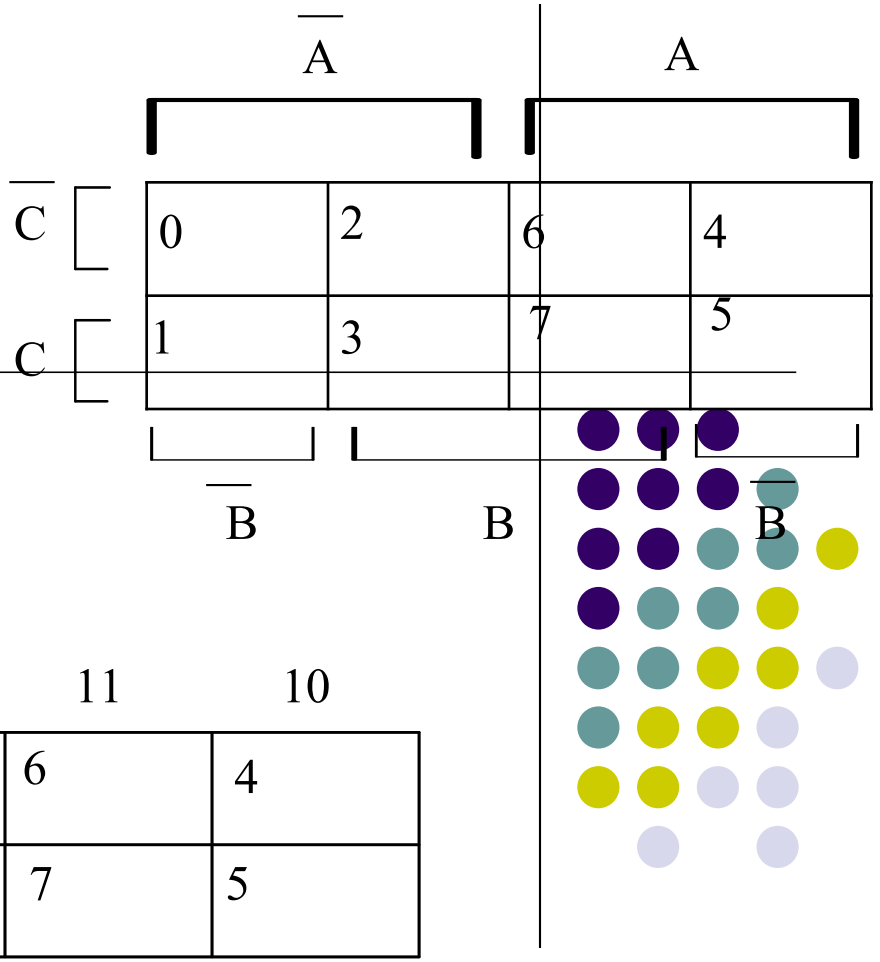
- ค่าเอาต์พุตนั้นเขียนในรูป Maxterm ซึ่งจะมีรูปแบบตรงข้ามกับ minterm

- จากรูปบนแสดงการจับคู่ของค่าตัวเลข 0 กับ 2 ได้ค่า B

- จากรูปล่างแสดงการจับคู่ของค่าตัวเลข 1 กับ 3 ได้ค่า $\overline{B}$

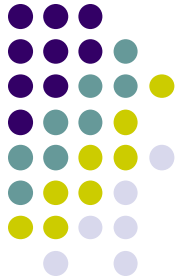
# K map ชนิด 3 ตัวแปร

| | | | | |
|---|---|---|---|---|
| C | m0 | m2 | m6 | m4 |
| $\overline{C}$ | m1 | m3 | m7 | m5 |




| AB / C | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 | 6 | 4 |
| 1 | 1 | 3 | 7 | 5 |

# อธิบาย



- K-map 3ตัวแปร ตารางมีค่าเท่ากับ $2^3$ 8 ค่าตัวแปร

- ค่า $\bar{A}\bar{B}\bar{C}=0, \bar{A}\bar{B}C=1, A\bar{B}C=2, \bar{A}BC=3, A\bar{B}\bar{C}=4, A\bar{B}C+5, AB\bar{C}=6, ABC=7$

- หลักการจับคู่ที่เป็นกลุ่ม สามารถเลือกจับคู่กลุ่มในตารางเป็น 1,2,4,8 โดยเลือกจับให้มากที่สุด เพื่อจะได้ลดรูปฟังก์ชันเหลือน้อยที่สุด

- จับคู่ในตารางให้ครบ 8 จะได้ค่าเอาต์พุตเท่ากับ 1

- จับคู่ในตารางกลุ่มครบ 4 จะได้ค่าเอาต์พุต 1 ตัวแปร

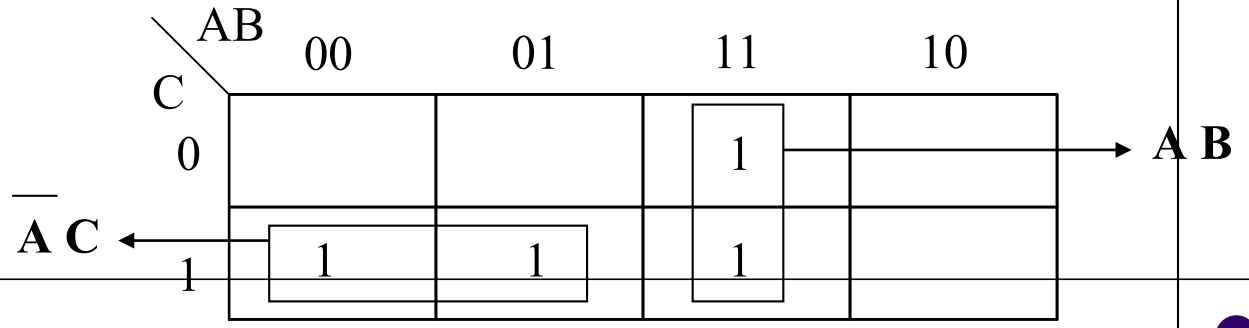
- จับคู่ในตารางกลุ่มครบ 2 จะได้ค่าเอาต์พุต2 ตัวแปร

- จับคู่ในตารางกลุ่ม 1 จะได้ค่าเอาต์พุต 3 ตัวแปร

# การลดรูป Switching function โดยใช้ K map ชนิด 3 ตัวแปร

โดยใช้ minterm

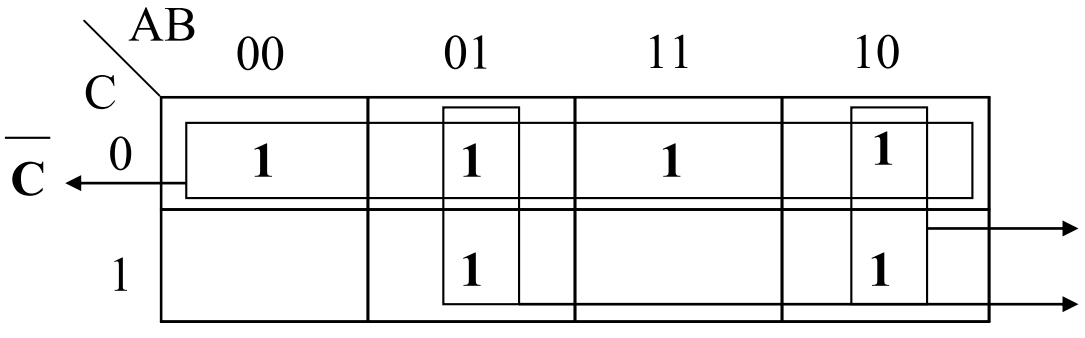
จงลดรูป $f(A,B,C) = \sum m(1,3,6,7)$




$f(A,B,C) = \sum m(1,3,6,7)$

$= AB + \overline{A}C$

จงลดรูป $f(A,B,C) = \sum m(0,2,3,4,5,6)$




$f(A,B,C) = \sum m(0,2,3,4,5,6)$

$= \overline{A}B + A\overline{B} + \overline{C}$

# อธิบาย

- ตัวอย่างการใช้k-map ลดรูปจากฟังก์ชันminterm ค่าตัวแปรที่ให้ค่ามาดังนี้ (1,3,6,7) รูปบน

- พิจารณาการจับคู่ได้เพียง 2 คู่ คือคู่ที่ 1 จับค่าเลข 1กับ 3 ได้ $A\overline{C}$ อีกคู่จับค่าเลข 6กับ 7 ได้ $AB$

อีกตัวอย่างรูปล่างให้ค่า minterm ดังนี้ (0,2,3,4,5,6)

- สามารถจับคู่ (0,2,6,4) ได้ค่าที่ $\overline{C}$

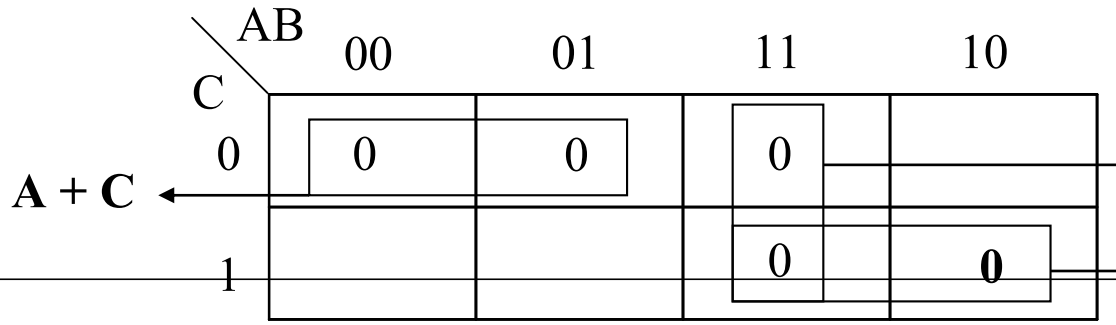
- สามารถจับคู่ (2,3) ได้ค่า $\overline{A}B$

- สามารถจับคู่ (4,5) ได้ค่า $A\overline{B}$

# การลดรูป Switching function โดยใช้ K map ชนิด 3 ตัวแปร

## โดยใช้ Maxterm

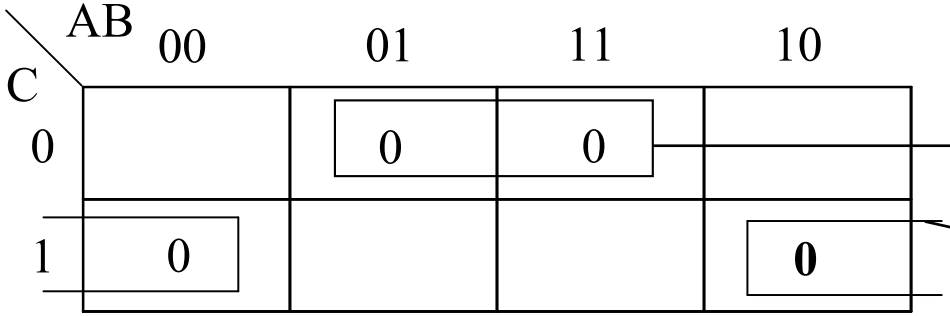
จงลดรูป $f(A,B,C)=\pi M(0,2,5,6,7)$




$f(A,B,C)=\pi M(0,2,5,6,7)$

$=(\overline{A}+\overline{B})(A+C)(\overline{A}+\overline{C})$

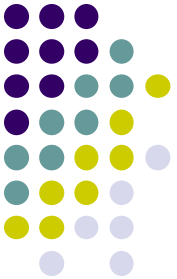
จงลดรูป $f(A,B,C)=\pi M(1,2,5,6)$




$f(A,B,C)=\pi M(1,2,5,6)$

$=(\overline{B}+C)(B+\overline{C})$

# อธิบาย



การลดรูปswitching functionโดยใช้ K-map 3ตัวแปรในรูปแบบ Maxterm ที่กำหนดค่าเป็น (0,2,5,6,7)

- ค่าจับคู่สามารถจับได้คือ 2ค่า เป็นจำนวน 3 เทอม ดังนี้คือ
- ค่า 0กับค่า2 ลดค่าในรูปแบบ maxterm คือ $A+C$
- ค่า 6กับค่า7 ได้ค่า $\overline{A}+\overline{B}$
- ค่า 7กับค่า 5 ได้ค่า $\overline{A}+\overline{C}$
- ได้ค่าเอาต์พุต $f=(A+C)(\overline{A}+\overline{B})(\overline{A}+\overline{C})$

อีกตัวอย่าง กำหนดค่าเป็น (1,2,5,6) รูปแบบMaxterm

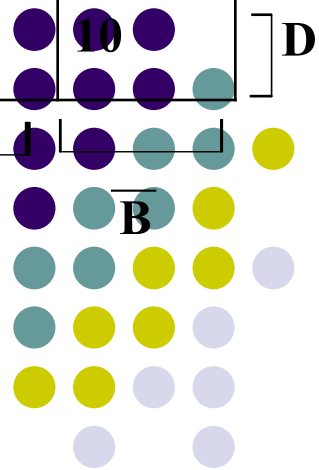
- ค่าที่สามารถจับคู่ได้ 2 ค่า 2เทอม คือค่า 2กับ6 ได้เอาต์พุตเป็น $\overline{B}+C$
- ค่าที่จับ 1กับ 5 ได้เอาต์พุต เป็น $B+\overline{C}$
- ค่าเอาต์พุตที่เกิดขึ้นจากการลดรูปคือ $(\overline{B}+C)(B+\overline{C})$

# K map ชนิด 4 ตัวแปร

| $\overline{C}$ \ $\overline{A}$ | $\overline{A}$ | $A$ | $A$ | $\overline{A}$ |
|---|---|---|---|---|
| | $\overline{B}$ | $B$ | $B$ | $\overline{B}$ |
| $\overline{C}$ | m0 | m4 | m12 | m8 |
| $\overline{C}$ | m1 | m5 | m13 | m9 |
| $C$ | m3 | m7 | m15 | m11 |
| $C$ | m2 | m6 | m14 | m10 |

| $\overline{C}$ \ $\overline{A}$ | $\overline{A}$ | $A$ | $A$ | $\overline{A}$ |
|---|---|---|---|---|
| | $\overline{B}$ | $B$ | $B$ | $\overline{B}$ |
| $\overline{C}$ | 0 | 4 | 12 | 8 |
| $\overline{C}$ | 1 | 5 | 13 | 9 |
| $C$ | 3 | 7 | 15 | 11 |
| $C$ | 2 | 6 | 14 | 10 |

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 4 | 12 | 8 |
| 01 | 1 | 5 | 13 | 9 |
| 11 | 3 | 7 | 15 | 11 |
| 10 | 2 | 6 | 14 | 10 |



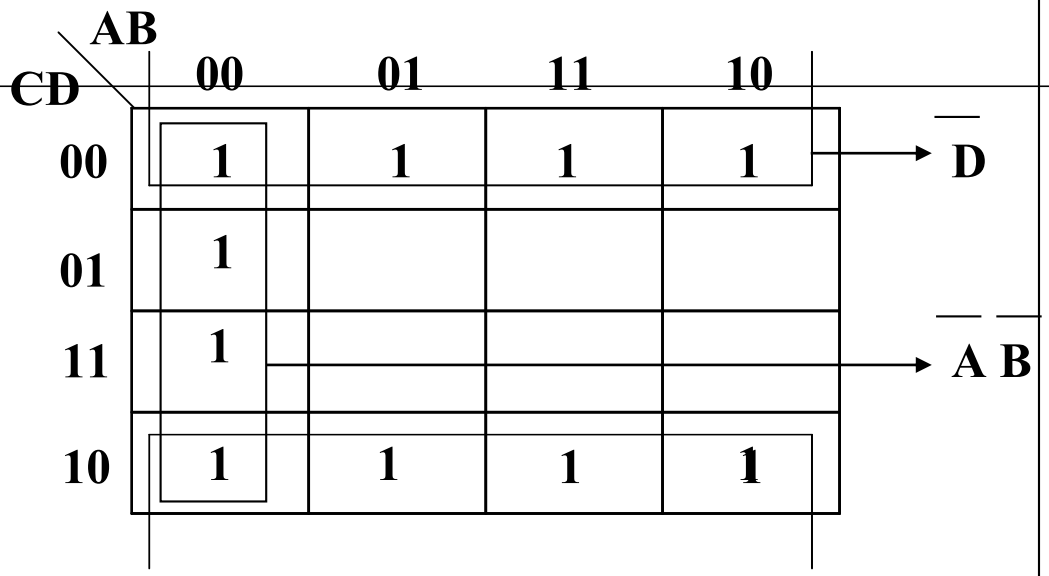
# อธิบาย

- K-map ชนิด4ตัวแปร $2^4$=16 ค่าที่แตกต่างกันคือ $m_0$-$m_{15}$ กำหนดไว้ในตารางพิจารณา
- แนวตั้งจะเป็นตัวแปรค่าทางอินพุต 2ค่า คือ A,B กำหนดความแตกต่างด้านบนเป็นค่า $\overline{A}$ กับค่า $A$ ด้านล่างเป็นค่า $\overline{B}$ ที่กำหนดไว้เพื่อให้map ต่อกันเพื่อให้ง่ายต่อการลดรูป และยังเป็นผลให้ค่าตารางจะไม่ซ้ำค่ากับ $A$
- แนวนอนด้านซ้ายมือจะเป็นค่า $\overline{C}$ 2 ช่องบน ส่วนอีก 2 ช่องล่างเป็นค่า $C$ ไม่ซ้ำกัน
- แนวนอนทางด้านขวามือจะกำหนดให้เป็นค่า $D$ และ $\overline{D}$ ที่มีตารางไม่ซ้ำกับค่า $\overline{C}$

# การลดรูป Switching function โดยใช้ K map ชนิด 4 ตัวแปร

## โดยใช้ minterm

จงลดรูป $f(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 10, 12, 14)$




$f(A, B, C, D) = \sum m(0, 1, 2, 3, 4, 6, 8, 10, 12, 14)$

$= \overline{D} + \overline{A}\,\overline{B}$

# อธิบาย

- การลดรูปk-map แบบ 4 ตัวแปรในรูปแบบ minterm เป็นวิธีที่ง่ายต่อการลดรูปสมการ
- หลักการจับคู่ในmap 16 ช่องได้ค่าเอาต์พุต เป็น 1

  - จับคู่ในmap 8 ช่องได้ค่าเอาต์พุต เป็น 1 ตัวแปร
  - จับคู่ในmap 4 ช่องได้ค่าเอาต์พุต เป็น 2 ตัวแปร
  - จับคู่ในmap 2 ช่องได้ค่าเอาต์พุต เป็น 3 ตัวแปร
  - จับคู่ในmap 1 ช่องได้ค่าเอาต์พุต เป็น 4 ตัวแปร

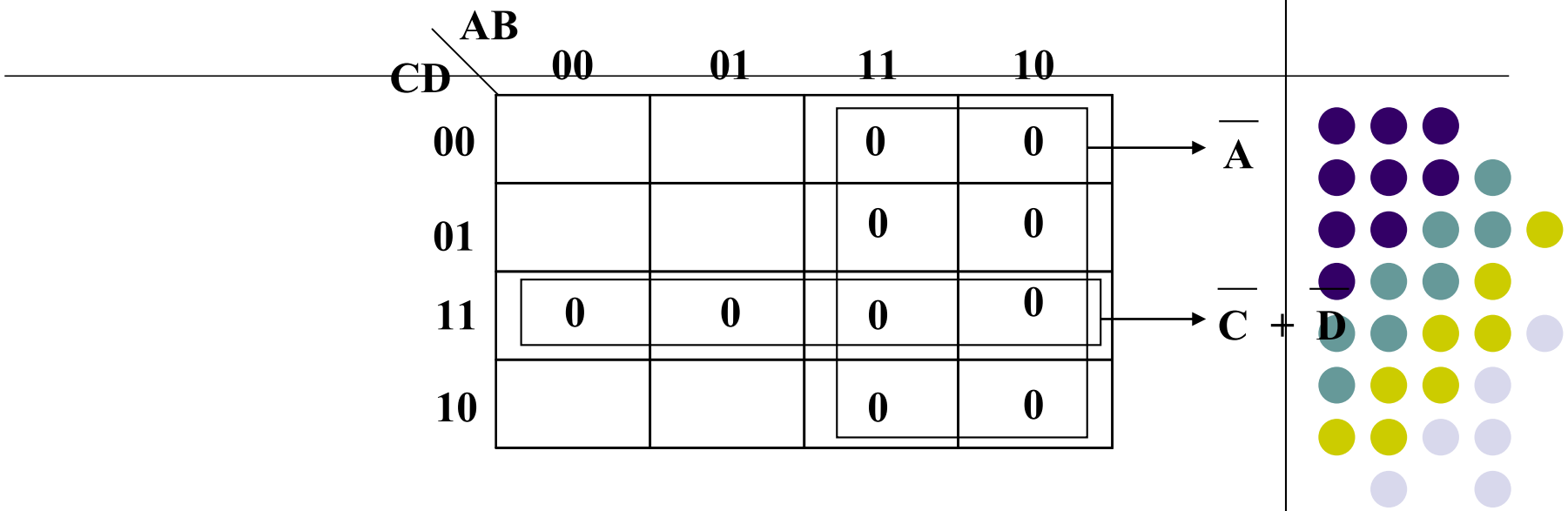
กำหนดค่าเป็น ( 0 , 1 ,2, 3 , 4 , 6 , 8 , 10 , 12 , 14 )

- จับคู่ในmap 8 ช่องคือค่า (0,4,12,8 กับ 2,6,14,10 )ได้ค่าเอาต์พุต เป็น 1 ตัวแปรคือ $\overline{D}$
- จับคู่ในmap 4 ช่องคือค่า (0,1,3,2) ได้ค่าเอาต์พุต $\overline{A}\overline{B}$
- ค่าที่เกิดขึ้นทางเอาต์พุตคือ $\overline{D}\left(\overline{A}\overline{B}\right)$

# การลดรูป Switching function โดยใช้ K map ชนิด 4ตัวแปร

## โดยใช้ Maxterm

จงลดรูป $f(A, B, C) = \pi M(3, 7, 8, 9, 10, 11, 12, 13, 14, 15)$




$f(A, B, C) = \pi m(3, 7, 8, 9, 10, 11, 12, 13, 14, 15)$

$= \overline{A}(\overline{C} + \overline{D})$

# อธิบาย

- การลดรูป k-map แบบ 4 ตัวแปรในรูปแบบ Maxterm เป็นอีกวิธีที่สามารถลดรูปสมการ
- หลักการจับคู่ใน map 16 ช่องได้ค่าเอาต์พุต เป็น 1
  - จับคู่ใน map 8 ช่องได้ค่าเอาต์พุต เป็น 1 ตัวแปร
  - จับคู่ใน map 4 ช่องได้ค่าเอาต์พุต เป็น 2 ตัวแปร
  - จับคู่ใน map 2 ช่องได้ค่าเอาต์พุต เป็น 3 ตัวแปร
  - จับคู่ใน map 1 ช่องได้ค่าเอาต์พุต เป็น 4 ตัวแปร
- หลักการจะเหมือนกับ minterm จะต่างกันที่เลือกจับค่าที่เป็น 0

ตัวอย่าง กำหนดค่าเป็น ¶m ( 3 , 7 , 8 , 9 , 10 , 11 , 12 , 13 , 14 , 15 )

-จับคู่ในmap 8 ช่องคือค่า (12,13,15,14 กับ 8,9,11,10 )ได้ค่าเอาต์พุต เป็น 1 ตัวแปรคือ $\overline{A}$

- จับคู่ในmap 4 ช่องคือค่า (3 , 7 , 15 , 11) ได้ค่าเอาต์พุต $\overline{C}+\overline{D}$
- เมื่อได้ค่าเอาต์พุตให้นำมารวมกันคือ $\overline{A}\left(\overline{C}+\overline{D}\right)$

# ตัวอย่างที่ 3.5 เปรียบเทียบการลดรูปแบบใช้พีชคณิตบูลลีนเทียบกับ k-map

$$x = AB + \bar{A}B + \bar{A}\bar{B}$$

1

$$= B(A + \bar{A}) + \bar{A}\bar{B}$$

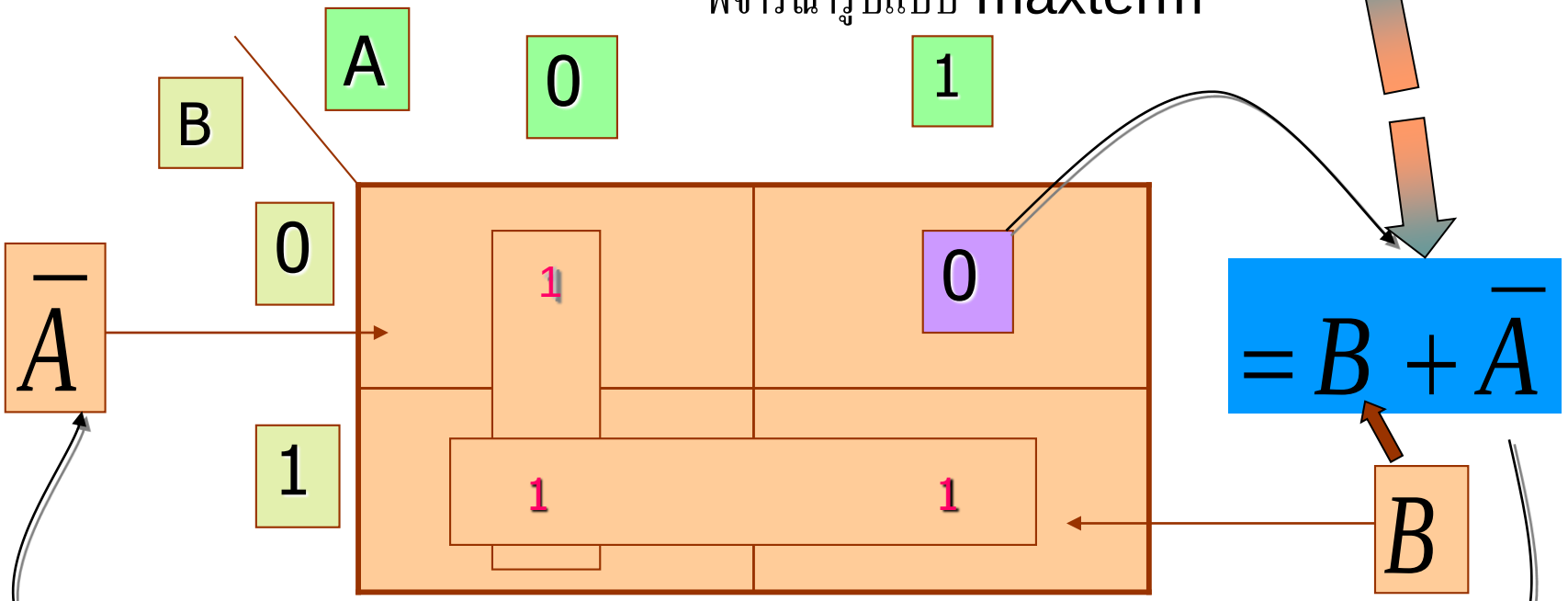
2

$$= B + \bar{A}\bar{B}$$

3

$$= B + \bar{A}$$

พิจารณารูปแบบ maxterm




พิจารณารูปแบบ minterm

# อธิบาย

- ตัวอย่างที่ 3.5 เป็นการเปรียบเทียบวิธีการลดรูป

- แบบใช้พีชคณิตบูลลีนเทียบกับ k-map 2 ตัวแปร

- การใช้พีชคณิตบูลลีนลดรูปได้ดังนี้

- การใช้พีชคณิตทฤษฎีที่11 (a) คอนเซนซัส และทฤษฎีที่9(a)เหลือค่าคือ $B+\overline{A}$

-การใช้ตาราง k-map โดย ให้ค่า $AB+\overline{A}B+\overline{A}\overline{B}$

- สามารถนำค่าลงตารางในช่อง 0,1,3

- จับคู่ระหว่าง 1,3 ได้ค่า B จับคู่ 0,1 ได้ค่า $\overline{A}$

- คำตอบก็คือ $B+\overline{A}$

การหาค่าอีกวิธีหนึ่งสามารถหาค่าได้จากตารางในรูป maxterm

- คำตอบก็คือช่องที่ 2 ของตารางคาร์โนห์ มีค่าเท่ากับ 0 คือค่า $B+\overline{A}$

ตัวอย่างที่ 3.6 การลดรูปสมการ $X = (A+B)(A+\bar{B})(\bar{A}+\bar{B})$

วิธีทำ

$$X = (A+B)(A+\bar{B})(\bar{A}+\bar{B})$$

$$= (A\bar{A} + A\bar{B} + BA + B\bar{B})(\bar{A}+\bar{B})$$

$$= A(1+\bar{B}+B)(\bar{A}+\bar{B})$$

$$= A(1)(\bar{A}+\bar{B})$$

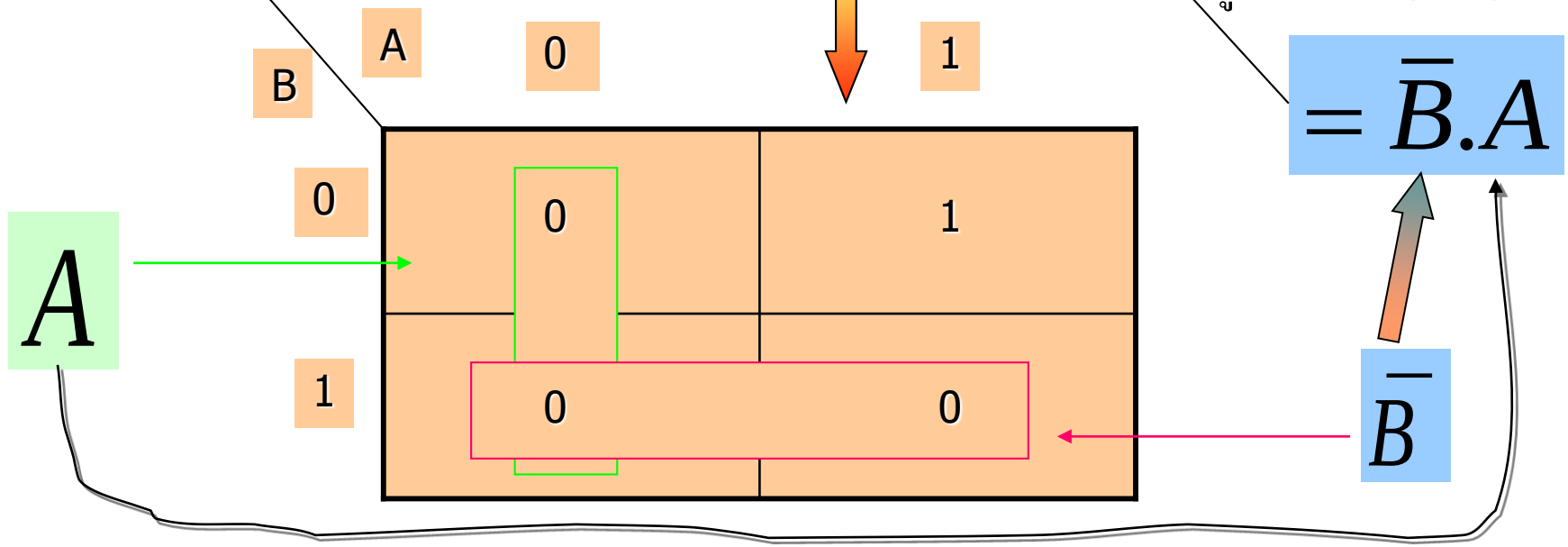
$$= \bar{B}A$$

พิจารณารูปแบบ minterm

$$= \bar{B}A$$

พิจารณารูปแบบ maxterm

$$= \bar{B}.A$$



# อธิบาย

- ตัวอย่างที่ 3.6 เป็นอีกตัวอย่างในการลดรูปสมการ $X = (A+B)(A+\overline{B})(\overline{A}+\overline{B}$

-ในรูปพีชคณิตเทียบกับตาราง K-map เพื่อให้เห็นค่าความสัมพันธ์กันทั้ง 2 รูปแบบ

-การใช้พีชคณิตบูลลีนโดยใช้ทฤษฎีที่ 13(a), ทฤษฎีที่ 7(b)

-ขั้นตอนที่ 1 ใช้ทฤษฎีที่ 3(a) กระจาย 2 เทอมแรกออก

-ขั้นตอนที่ 2 ใช้ทฤษฎีที่ 4(b) กับค่า $A\overline{A} = 0$

-ขั้นตอนที่ 3 ใช้ทฤษฎีที่ 5(b)ลดทอนเหลือค่าผลลัพธ์ $\overline{B}A$

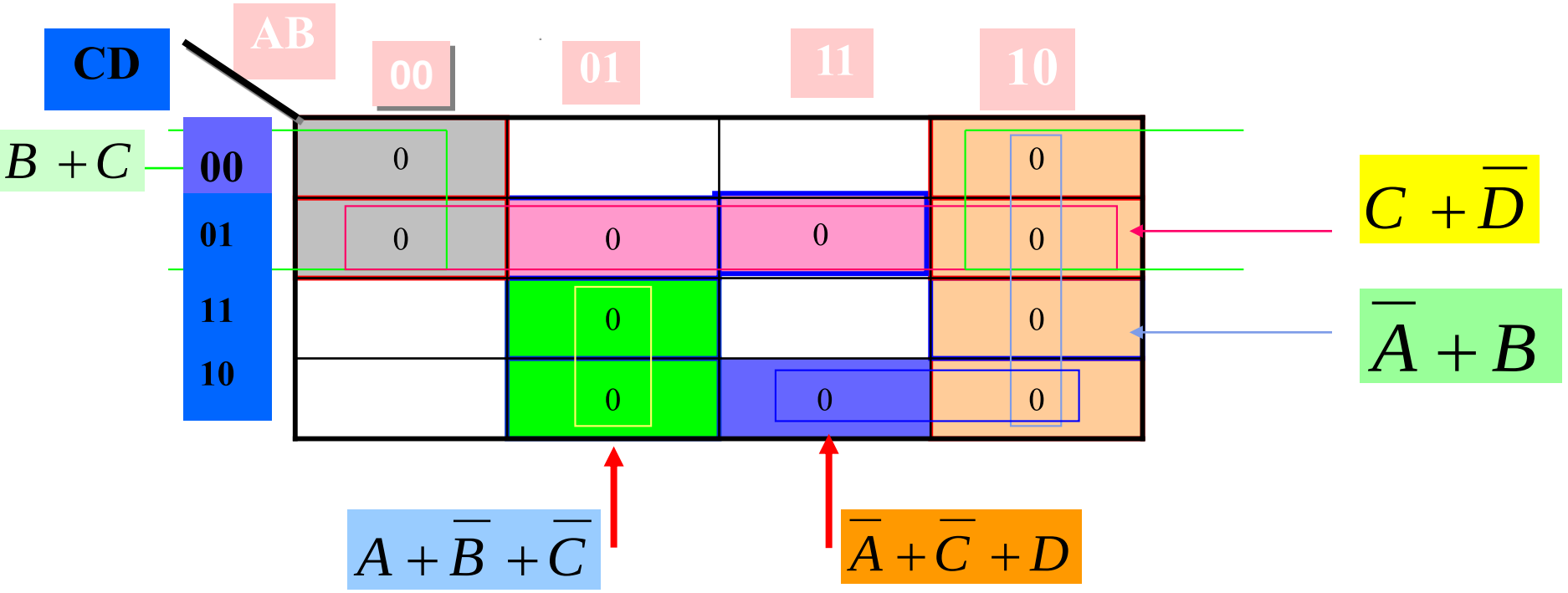
# ตัวอย่างที่ 3.7 ลดรูปสมการ MAX - Term

$$X = (A+B+C+D)\bullet(A+B+C+\overline{D})\bullet(A+\overline{B}+C+\overline{D})\bullet$$

$$(\overline{A}+\overline{B}+C+\overline{D})\bullet(\overline{A}+B+C+\overline{D})\bullet(\overline{A}+B+C+D)\bullet$$

$$(\overline{A}+B+\overline{C}+\overline{D})\bullet(\overline{A}+B+\overline{C}+\overline{D})\bullet(A+\overline{B}+\overline{C}+\overline{D})\bullet$$

$$(A+\overline{B}+\overline{C}+D)\bullet(\overline{A}+\overline{B}+\overline{C}+D)$$




$$X = (B+C)\bullet(\overline{A}+B)\bullet(C+\overline{D})\bullet(A+\overline{B}+\overline{C})\bullet(\overline{A}+\overline{C}+D)$$

# อธิบาย

ตัวอย่างการนำMaxterm มาใช้ในการลดรูป สมการที่กำหนดให้ฟังก์ชัน

(0,1,5,6,7,8,9,10,11,13,14)

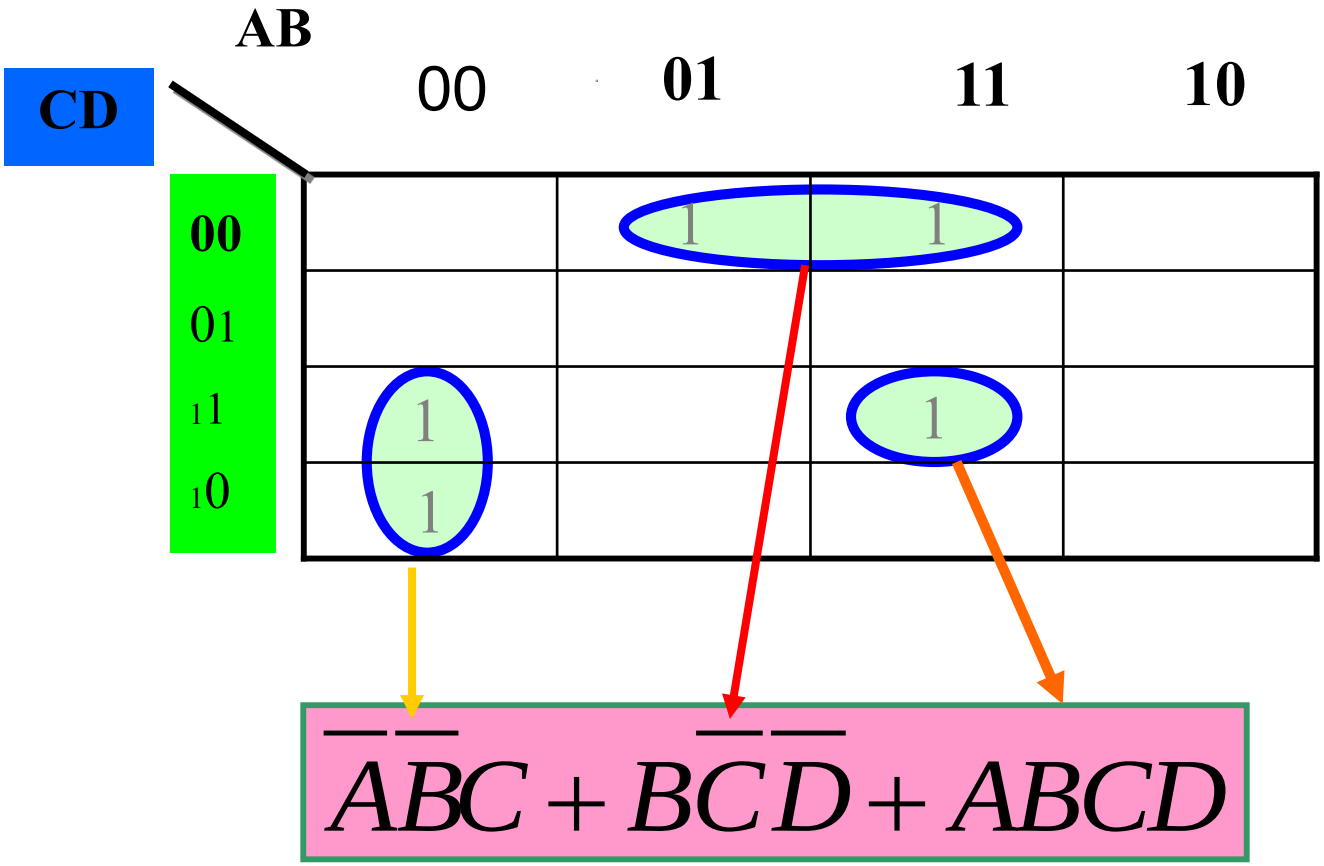
- พิจารณาจากค่าตารางในK-map สามารถจับคู่กลุ่ม 4 , 2 ได้ดังต่อไปนี้
- จับคู่กลุ่ม 4 ได้ 3 เทอม คือ (0,1,8,9)ได้ค่า $(B+C)$
- จับคู่กลุ่ม4 ได้ (8,9,11,10) $(\overline{A}+B)$
- จับคู่กลุ่ม 4 ได้ 3 เทอม คือ (1,5,13,9ได้ค่า $(C+\overline{D})$

ในส่วนที่เหลืออีก 4 เทอมที่ไม่ต่อกันสามารถจับคู่2 ได้ 2 เทอมคือ

- จับคู่ (6,7)ได้ค่า $(A+\overline{B}+\overline{C})$
- จับคู่ (14,10)ได้ค่า $(\overline{A}+\overline{C}+D)$

# ตัวอย่างที่ 3.8 จงลดรูปจากตารางในสมการ

## Min - Term



# อธิบาย

- ตัวอย่างที่ 3.8 แสดงการลดรูปแบบ minterm ค่าที่กำหนดให้ในตารางเป็น (2,3,4,12,15)

- จากตัวอย่างนี้ค่าที่ลงในk-map จะกระจายไม่สามารถจับคู่กลุ่ม 4 ช่องได้

- พิจารณาคู่กลุ่ม 2 ช่องติดต่อกัน เป็น 2เทอม และแยกกลุ่ม1 ช่อง 1 เทอม

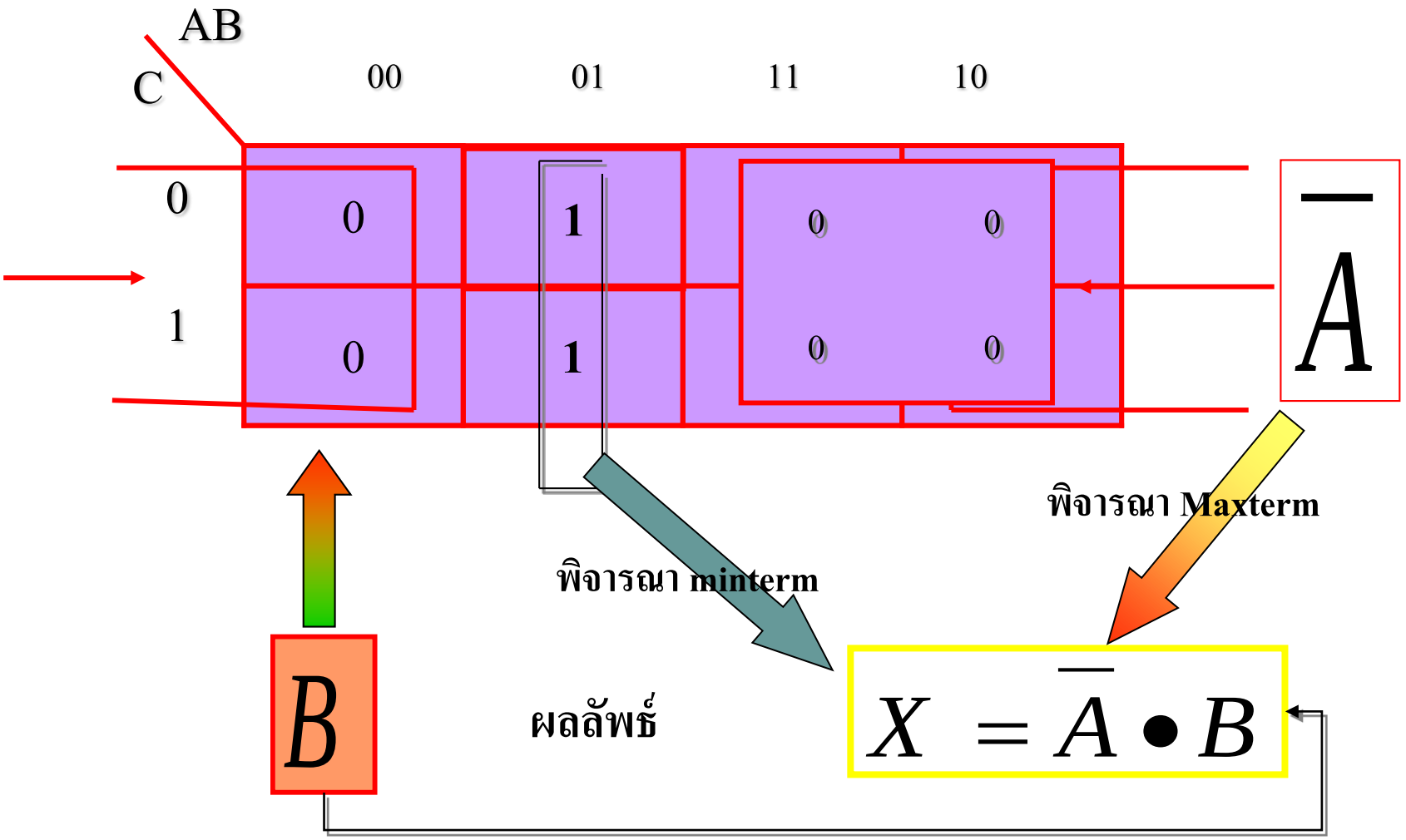
- ค่าตัวแปรที่เกิดจาก 2 ช่องรวมกันคือ ค่าในช่อง(3กับ 2)ได้ค่า $\overline{A}\,\overline{B}\,C$

- และช่องเลข (4 กับ 12) ได้ค่า $B\,\overline{C}\,\overline{D}$

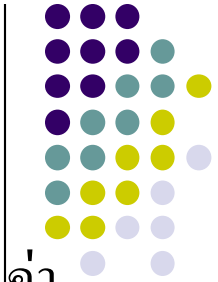
- ค่าเอาต์พุตที่เกิดจาก 1 ช่องที่ไม่สามารถรวมกับกลุ่มอื่นได้คือช่องหมายเลข 15 มีค่าเท่ากับ $ABCD$

# ตัวอย่างที่ 3.9 จงลดรูปสมการพีชคณิตในรูปแบบ K-map

$$X = (\overline{A} + B + C) \bullet (\overline{A} + \overline{B} + C) \bullet (\overline{A} + \overline{B} + \overline{C}) \bullet (\overline{A} + B + \overline{C}) \bullet (A + B + C) \bullet (A + B + \overline{C})$$



# อธิบาย



- ตัวอย่างที่3.9 การลดรูปจากฟังก์ชันทางพีชคณิตที่ให้มาในรูปแบบ Maxterm 6 ค่า
- ค่าที่ให้มา ¶ m ( 0,1,4,5,6,7)

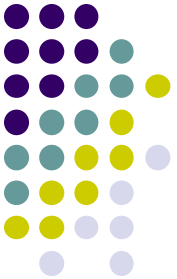
  - ค่าที่เหลือจะอยู่ในรูป minterm ก็คือค่า (2,3)

  - จับคู่ใน map 4 ของ maxterm ได้2 เทอม (0,1,4,5) กับ(6,7,4,5) ได้ค่าคือ $B$ กับ $\overline{A}$

  - เอาต์พุตที่ได้ ค่าที่ควรนำมาพิจารณาจะเห็นว่าจะง่ายอยู่ในรูป minterm เพราะว่ามีเพียงเทอมเดียว $\overline{A}.B$
- ที่นำมาจับคู่ 2

  - ค่าที่ได้คือ $\overline{A}.B$

# ตัวอย่างที่ 3.10 จงลดรูปสมการพีชคณิตเทียบกับตาราง K-map



**วิธีทำ** $X = ABC + \overline{A}\overline{B}C + A\overline{B}C + \overline{A}\overline{B}\overline{C}$

1 $X = ABC + \overline{A}\overline{B}C + A\overline{B}C + \overline{A}\overline{B}\overline{C}$

2 $= AC\left(\overline{B} + B\right) + \overline{A}\overline{B}\left(\overline{C} + C\right)$

3 $= AC + \overline{A}\overline{B}$

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | | | |
| 1 | 1 | | 1 | 1 |

$AC$

$\overline{A}\overline{B}$

$X = AC + \overline{A}\overline{B}$

# อธิบาย

- ตัวอย่างที่ 3.10 ตัวอย่างนี้แสดงการเปรียบเทียบการลดรูปแบบพีชคณิตเทียบกับตารางK-map โดยมีฟังก์ชันการทำงานให้ดังนี้

$$X = ABC + \overline{A}\overline{B}C + A\overline{B}C + \overline{A}\overline{B}\overline{C}$$

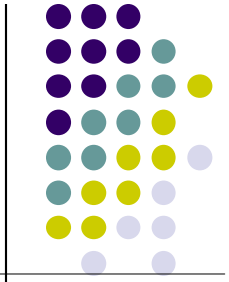
- ขั้นตอนแรกจัดรูปฟังก์ชันให้เหมาะสมเพื่อทำการลดรูปในขั้นตอนสองใช้กฎการสลับที่

- ขั้นตอนที่สอง ใช้กฎการกระจายทั้งสองเทอม

- ขั้นตอนที่สาม ตัวแปรออร์กับค่า ตัวคงที่มีค่าเท่ากับ1 ผลลัพธ์เท่ากับ 1

- ค่าผลลัพธ์ที่ได้ $AC + \overline{A}\overline{B}$

# ทดสอบความรู้สัปดาห์ที่3



**1. จงลดรูปสมการให้อยู่ในรูปง่ายๆ พร้อมเขียนวงจรเกต**

(a) $f(A,B,C,D) = \overline{(A+\overline{C}+D)(\overline{B}+C)(A+\overline{B}+D)(\overline{B}+C)(\overline{B}+C+\overline{D})}$

(b) $f(A,B,C,D) = \overline{AB+\overline{A}\,\overline{D}+B\,\overline{D}+\overline{A}\,B+C\,\overline{D}\,A+\overline{A}\,D+CD+\overline{A}\,\overline{B}\,\overline{D}}$

(c) $f(A,B,C,D) = \overline{A\overline{B}C+AB+\overline{ABC}+A\overline{C}+AB\overline{C}}$

(d) $f(A,B,C) = \overline{(B+\overline{A})(AB+C)+AB\,\overline{A}+\overline{A}\,\overline{B}C+(A+B)(\overline{A}+C)}$

(e) $f(A,B,C) = \overline{(\overline{A}+\overline{B})(A+\overline{A}B)(\overline{A}+\overline{B}+\overline{A}\,\overline{B}C)+(A+B)(\overline{A}+C)}$

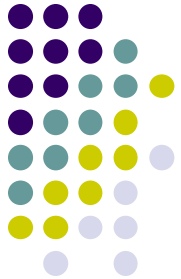
**2. จงหาค่าฟังก์ชันการทำงานดังต่อไปนี้**

(a) $f(A,B,C) = \sum m(1,4,5)$

(b) $f(A,B,C,D) = \prod M(0,2,4,5,8,11,15)$

(c) $f(A,B,C,D) = \sum m(0,2,5,8,9,10,13)$

# ทดสอบความรู้สัปดาห์ที่3 (ต่อ)



**3. ใช้ทฤษฎีของ De Morganes ในการลดรูป:**

(a) $X(Y+\overline{Z}(Q+\overline{R})$

(b) $X+Y(\overline{Z}+Q\overline{R})$

(c) $XY+A\overline{C}+IQ$

(d) $(A+B\overline{C})(\overline{A}+\overline{D}E)$

**4. ใช้ทฤษฎี (consensus) ในการลดรูปตามฟังก์ชันที่ให้มา:**

(a) $QR+\overline{X}Q+RX$

(b) $(X+Y)Z+\overline{X}\,\overline{Y}W+ZW$

(c) $(\overline{X}+Y)WZ+X\overline{Y}V+VWZ$

(d) $(X+Y+Z+\overline{W})(V+X)(\overline{V}+Y+Z+W)$

**5. จงเขียนตารางความจริงจากฟังก์ชันที่ให้มา:**

(a) $f(A,B)=A+B$

(b) $f(A,B,C)=AB+A\overline{C}$

(c) $f(a,b,c)=ab\overline{c}+b\overline{c}$

(d) $f(a,b,c)=a(b+\overline{c})(\overline{b}+c)$

# เนื้อหาคำบรรยายในสัปดาห์ที่ 4

# การออกแบบวงจร Combination Logic Gates

41 วงจรเกตเบื้องต้นกับสัญลักษณ์มาตรฐาน IEEE

411 Truth Table of Logic Gates

412 การนำทฤษฎีของดีมอร์แกน มาใช้งานในรูปแบบ NAND กับ NOR

413 การออกแบบวงจร ลอจิกเกตในเทอมของ minterm , Maxterm

414 Don’t care

42 การเขียนลอจิกเกตจากฟังก์ชันพีชคณิต

421 การเขียน Timing Diagram จากวงจรลอจิกเกต

422 การเขียนลอจิกฟังก์ชันจากสวิทซ์ คอนแทครีเลย์

# อธิบายในสัปดาห์ที่4 บรรยาย

- การออกแบบวงจร Combination logic gates ที่ประกอบด้วยเกตประเภทต่างๆ จาก Switching

-Funtion ใดๆ วงจรออกแบบที่ดีจะใช้วงจรเกตน้อยที่สุด เพื่อลดความยุ่งยากและความประหยัด

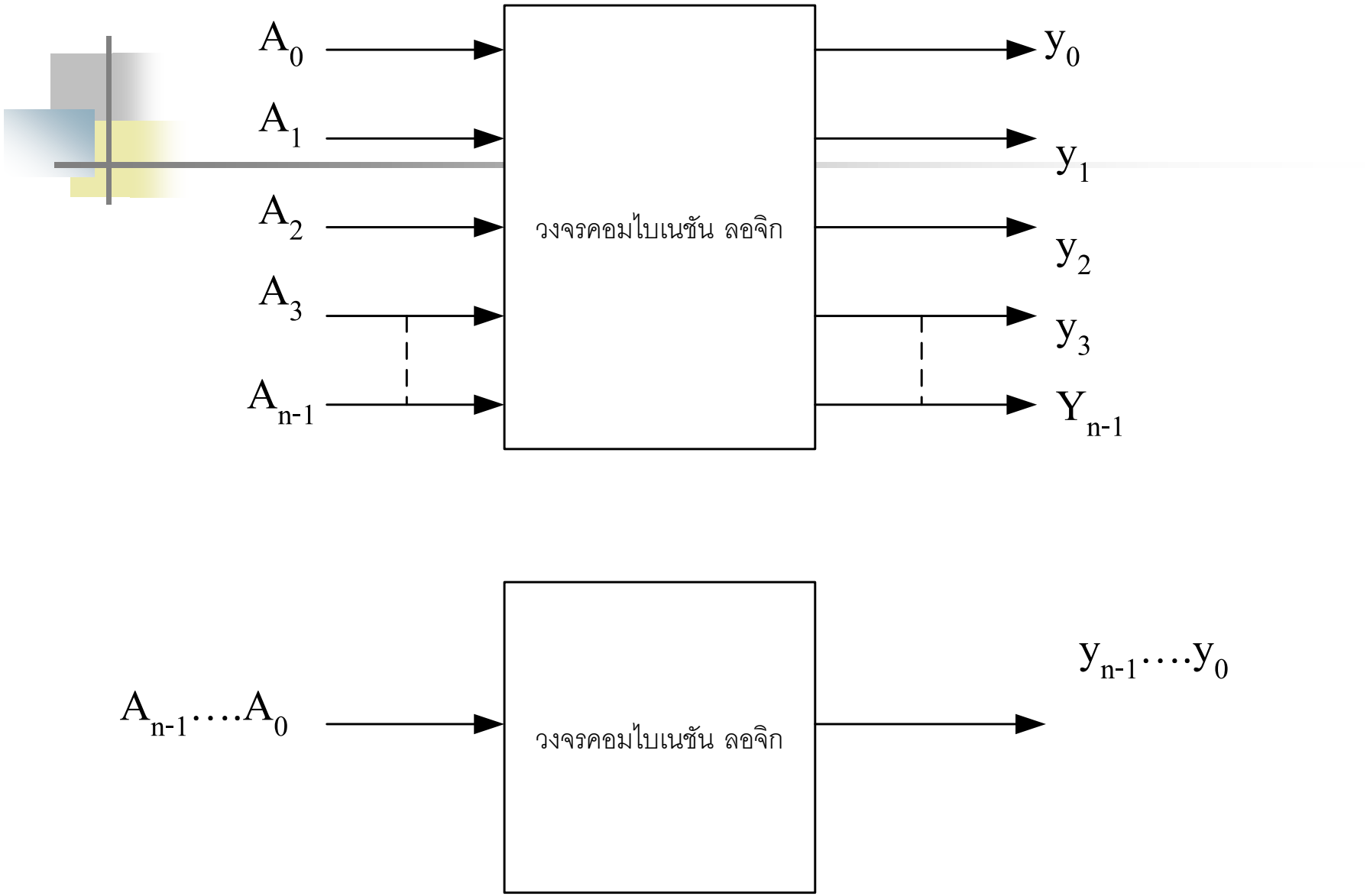
- Switching Function ใดๆ สามารถสร้างจากตารางความจริงได้

- การออกแบบวงจร Combination logic gates ด้วยวงจร Universal gates โดยการใช้ทฤษฎีดีมอร์แกน

- การเขียนฟังก์ชันจากการกำหนดค่าตัวแปรทางอินพุต 3 ตัวแปร

- วิธีการลดรูปฟังก์ชันโดยใช้ K-map

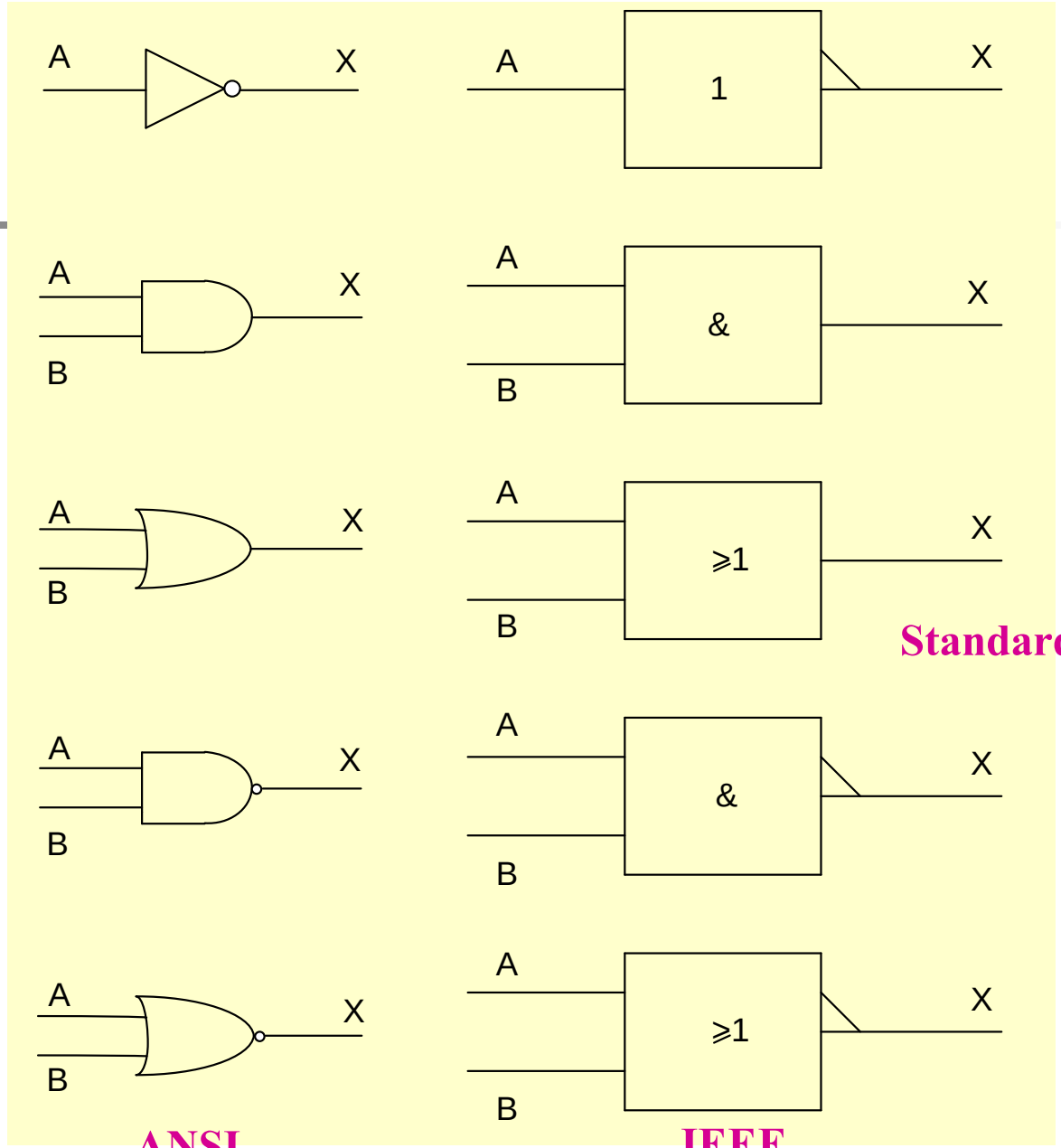
- จากการออกแบบวงจร Combination logic gate นำไปสร้างวงจรเกตควบคุมเอาต์พุต โดยสัญญาณอินพุต 3 ตัวแปร


$A_0$
$A_1$
$A_2$
$A_3$
$A_{n-1}$
วงจรคอมไบเนชัน ลอจิก
$y_0$
$y_1$
$y_2$
$y_3$
$Y_{n-1}$
$A_{n-1}....A_0$
วงจรคอมไบเนชัน ลอจิก
$y_{n-1}....y_0$

# อธิบาย

- วงจรคอมไบเนชัน หรือวงจรรวมอุปกรณ์การทำงานทางลอจิกเข้าด้วยกัน
- วงจรคอมไบเนชันสามารถแบ่งออกเป็น 3 กลุ่มคือ
- กลุ่มวงจรการคำนวณทางคณิตศาสตร์ (Arithmetic Circuit)
- กลุ่มวงจรเข้ารหัส/ถอดรหัส (Encoder/Decoder  Circuit)
- กลุ่มวงจรรับส่ง/ข้อมูล (Multiplexer /Demultiplexer  Circuit)

# 41 วงจรเกตเบื้องต้นพร้อมกับสัญลักษณ์มาตรฐานANSI กับมาตรฐานของIEEE

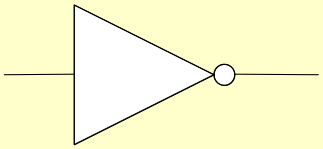



Standard Logic Symbols

# อธิบาย

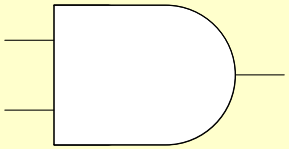
- สัญลักษณ์มาตรฐาน ANSI เทียบกับ IEEE
- ด้านซ้ายมือเป็นมาตรฐานสัญลักษณ์ของเกตที่ใช้ในวงจรการทำงานทั่วไปหรือเป็นมาตรฐานANSI ซึ่งเป็นอักษรย่อมาจาก National Standard Institute
- ส่วนทางด้านขวามือจะเป็นสัญลักษณ์ที่ใช้มาตรฐานIEEE ( Institute of Electrical and Electronic Engineering) จะใช้รูปสี่เหลี่ยมแทนวงจรเกต ส่วนที่แสดงเป็นหน้าที่การทำงานทางฟังก์ชันจะใช้สัญลักษณ์ทางคณิตศาสตร์ แทนค่า 1, &, $\geq 1$
- การทำงานทางอินพุตกับ เอาต์พุต จะใช้สามเหลี่ยมแทนในสภาวะที่เป็นโลว์ดังในรูปแสดง(Active Low) หรือ ทำงานที่สภาวะจากค่า1 ไปเป็น 0

## Inverter



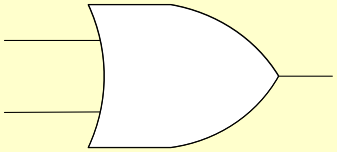
$F = \overline{A}$

### AND Gate



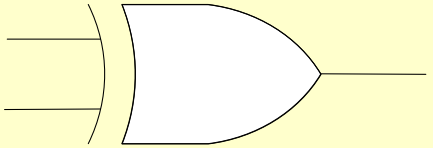
$F = A.B$

### OR Gate



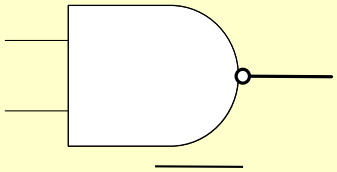
$\boldsymbol{F = A+B}$

### Exclusive OR Gate



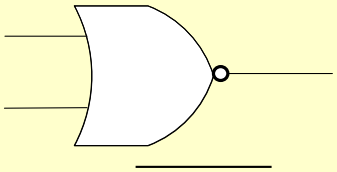
$F = A \oplus B$

### NAND Gate



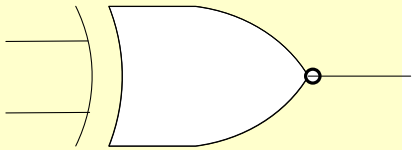
$F = \overline{A.B}$

### NOR Gate



$F = \overline{A + B}$

### Exclusive NOR Gate



$F = \overline{A \oplus B}$

# อธิบาย

การทำงานของเกตที่ใช้ในทางดิจิตอลเบื้องต้น

- ชนิดที่1 เป็นเกตชนิด INVERTER ค่าฟังก์ชันมีค่าเท่ากับ $\bar{A}$
- ชนิดที่2 เป็นเกตชนิด AND ค่าฟังก์ชันมีค่าเท่ากับ A.B
- ชนิดที่3 เป็นเกตชนิด NAND ค่าฟังก์ชันมีค่าเท่ากับ $\overline{A.B}$
- ชนิดที่4 เป็นเกตชนิด OR ค่าฟังก์ชันมีค่าเท่ากับ A + B
- ชนิดที่5 เป็นเกตชนิด NOR ค่าฟังก์ชันมีค่าเท่ากับ $\overline{A + B}$
- ชนิดที่6 เป็นเกตชนิด EX-OR ค่าฟังก์ชันมีค่าเท่ากับ $A \oplus B$
- ชนิดที่7 เป็นเกตชนิด EX-NOR ค่าฟังก์ชันมีค่าเท่ากับ $\overline{A \oplus B}$

# 411 ตารางความจริงการทำงานของเกตต่างๆ

| A | $\overline{A}$ | B | $\overline{B}$ | AND | NAND | OR | NOR | EX-OR | EX-NOR |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

อินพุตต่างกันให้เอาต์พุตเป็น 1

อินพุตเหมือนกันให้เอาต์พุตเป็น 1

# อธิบาย

- **ตารางความจริงของการทำงานของเกตชนิดต่างๆ**

- INVERTER gate ค่าฟังก์ชันจะมีค่าตรงข้ามกับอินพุต

- AND gate จะให้ค่าเอาต์พุตก็ต่อเมื่อค่าอินพุตเป็น 1 ทั้งสองอินพุต

- NAND gate ประกอบด้วยAND กับ INVERTER gate ค่าที่เกิดขึ้นทางเอาต์พุต จะตรงข้ามกับ AND gate

- OR gate ค่าเอาต์พุตจะเป็น 1 ก็ต่อเมื่อทางอินพุตใดๆ เป็น 1 หรืออินพุตทั้งหมดเป็นค่า 1

- NOR gate ประกอบด้วยOR กับ INVERTER gate ค่าเอาต์พุตจะจะเป็น 1 เมื่ออินพุตเป็น 0 ทั้งหมด

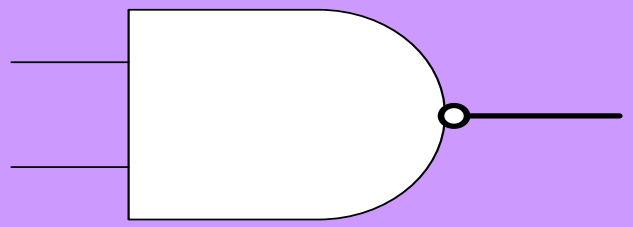
- EXCLUSIVE-OR gate เขียนย่อ EX-OR gate เป็นเกตที่จะให้ค่าเอาต์พุตจะจะเป็น 1 โดยนำอินพุตมาบวกกันแบบไม่คิดตัวทด หรือเอาต์พุตจะเป็น1 เมื่ออินพุตต่างกัน

- EXCLUSIVE-NOR gate เขียนย่อ EX-NOR gate ประกอบด้วย EX-OR กับ INVERTER gate ค่าเอาต์พุตจะจะตรงข้ามกับ EX-OR gate หรือพิจารณาง่ายๆ เอาต์พุตจะจะเป็น 1 ก็ต่อเมื่ออินพุตเข้ามามีค่าเท่ากัน ใช้ในวงจรเปรียบเทียบ

# 412 การนำทฤษฎีของดีมอร์แกนมาใช้งานในรูปแบบ NAND Gate กับ NOR Gate

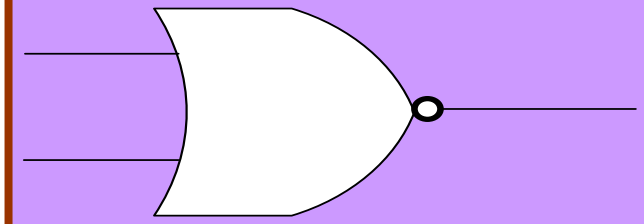
**Universal Gates**

NAND Gate



$$F = \overline{A.B}$$

NOR Gate



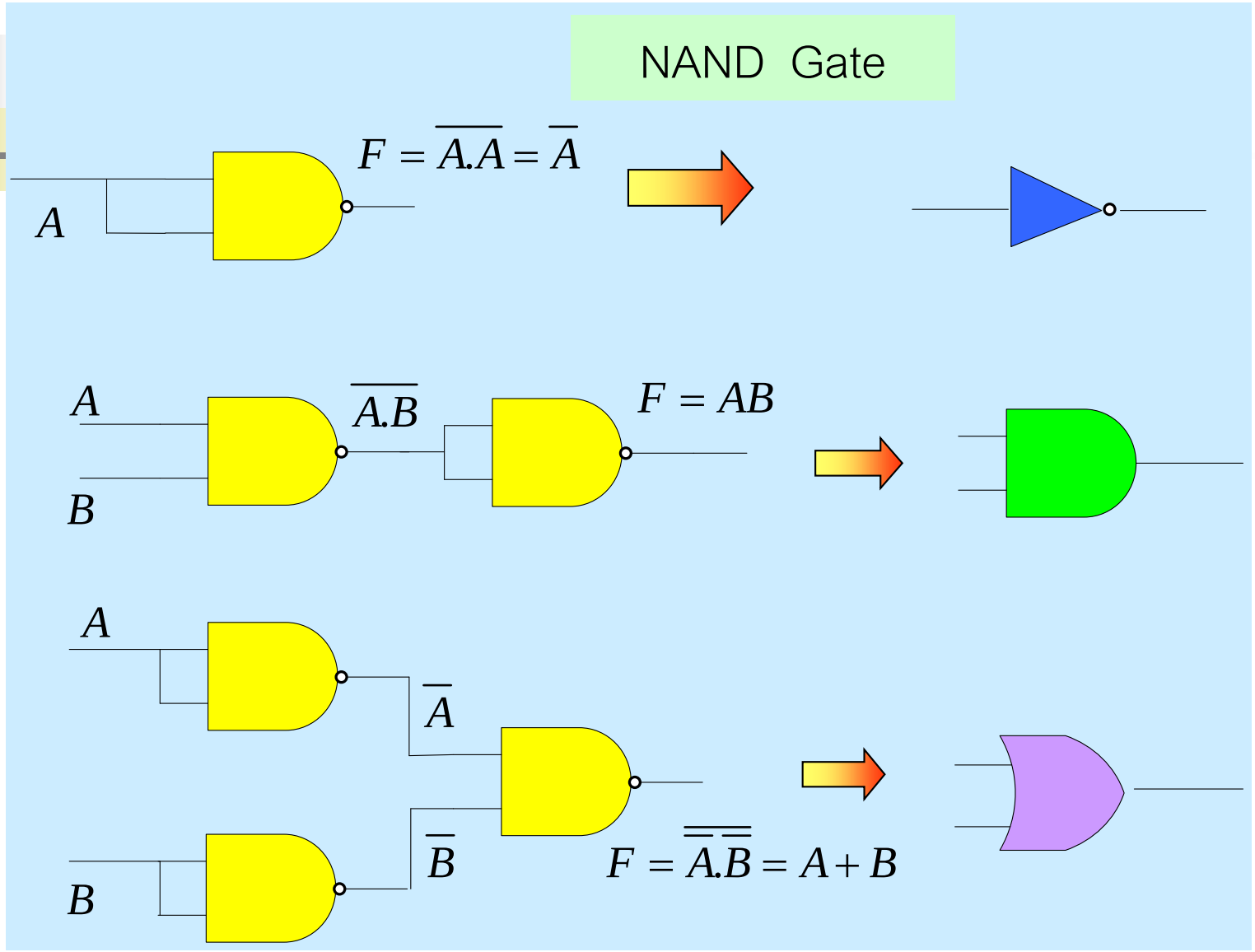
$$F = \overline{A + B}$$

# อธิบาย

- Universal Gates คือลอจิกเกตที่สามารถนำไปใช้ได้ทุกฟังก์ชันทางคณิตศาสตร์ ซึ่งเป็นเกตสากลที่ได้รับความนิยมสูง
- ในทางปฏิบัติ UNIVERSAL gate เป็นชื่อเรียก NAND กับ NOR gates ด้วยความสามารถของเกตทั้งสองชนิด นี้สามารถนำไปแทนการทำงานของเกตที่กล่าวมาทั้งหมด ไม่ว่าจะเป็น OR ,AND , INVERTER, EX-OR ,EX-NOR gates
- หลักการแทนฟังก์ชันโดยการใช้ทฤษฎีของ De Morgan's เข้าช่วยดังนี้คือ

$$A = \overline{\overline{A}}, \quad \overline{A}.\overline{B} = \overline{A+B}, \quad \overline{A}+\overline{B} = \overline{AB}$$

# Universality of NAND gates



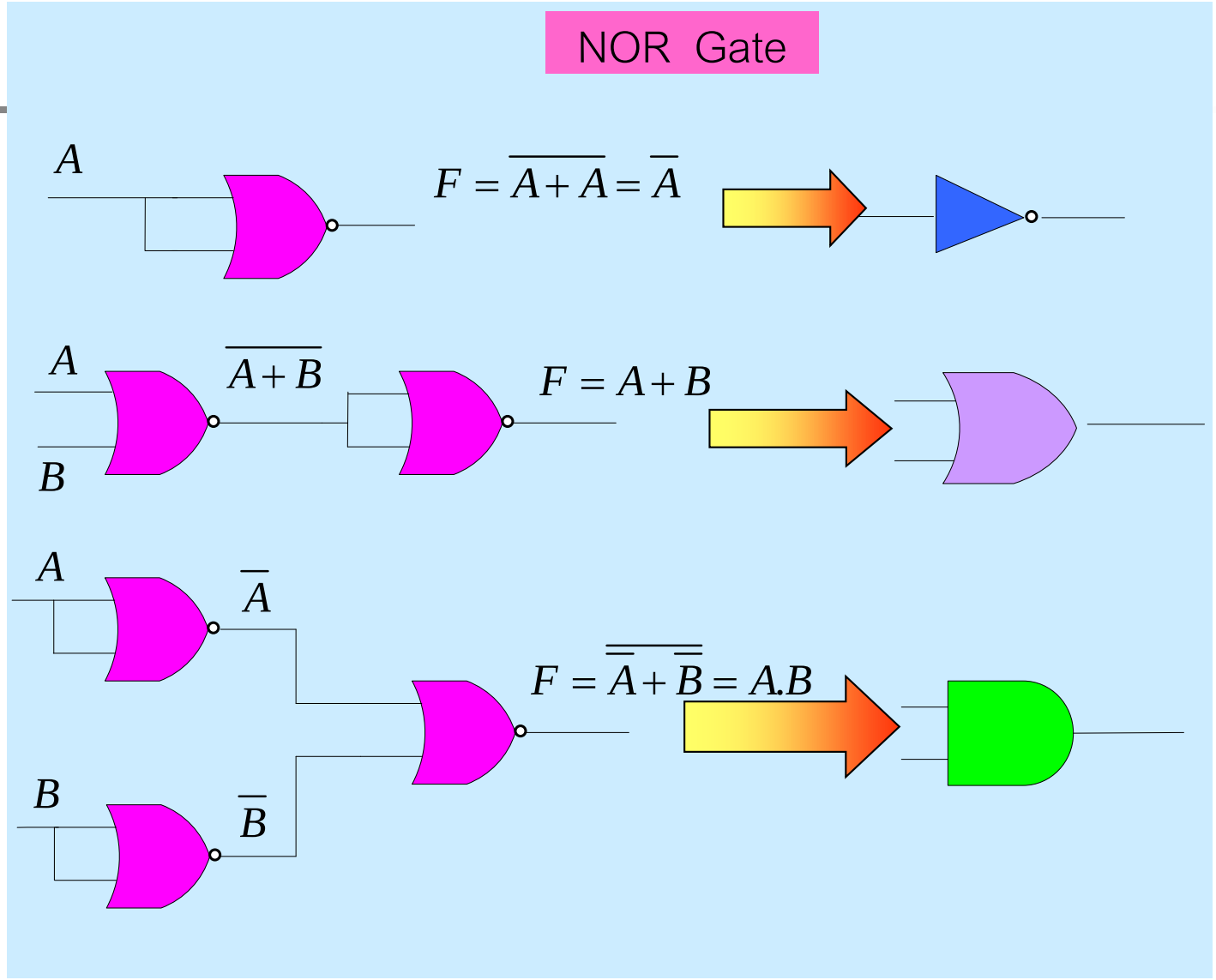
# อธิบาย

- การนำเอาNAND gate ไปใช้แทนค่า INVERTER gate โดยการนำอินพุตทั้งสองของNAND gate ต่อเข้าด้วยกันดังสมการ $F = \overline{A.A} = \overline{A}$ T1(b)

- ฟังก์ชันAND gate สามารถนำNAND gate จำนวนสองตัวมาต่อเข้าด้วยกัน โดยการนำINVERTER gate ของฟังก์ชันที่กล่าวมา ต่อไปยังเอาต์พุตจะของNAND gates ตัวแรก เป็นดังสมการ $F = \overline{A.B}$ ตัวที่สองเป็นอินเวอร์เตอร์ $F = \overline{\overline{A.B}} = AB$

-ฟังก์ชันOR gate จะถูกแทนด้วย NAND gates จำนวน3 ตัว โดยนำอินพุต A,B มาผ่าน INVERTER gates แยกกัน 2 อินพุต ดังรูปจากนั้นนำ เอาต์พุต $\overline{A}.\overline{B}$ มาต่อ NAND gate2 อินพุต ค่า $F = \overline{\overline{A}.\overline{B}} = A + B$

# Universality of NOR gates



# อธิบาย

- การนำเอา NOR gate ไปแทน INVERTER gate โดยการนำอินพุตทั้งสองของ NORมาต่อเข้าด้วยกันค่า $F = \overline{A + A} = A$ T1(a)
- ฟังก์ชัน OR gate จะถูกแทนด้วยวงจร NOR gate จำนวนสองตัวมาต่อเข้าด้วยกันจาก $F = \overline{A + B}$ เป็นเอาต์พุตจะของ NOR gate ถ้านำ INVERTER gate มาต่อเข้าด้วยกันกับเอาต์พุตจะ NOR Gate ก็จะได้
- $F = \overline{\overline{A + B}} = A + B$ T3
- ฟังก์ชัน NOR gate จะถูกแทนด้วย NOR gates จำนวน 3 ตัว โดยนำอินพุต A,B มาต่อแบบ INVERTER gates แบบแยกกันสองอินพุต ดังรูป จากนั้นนำเอาต์พุตจะของ $\overline{A}.\overline{B}$ มาต่อเข้ากับอินพุตของNAND gate 2 อินพุต $F = \overline{\overline{A} + \overline{B}} = A.B$ T8(a),T3

# หลักการออกแบบวงจรลอจิกเกต (Logic combination)

1. ทำการเขียนตารางตามการทำงานหน้าที่ของวงจร โดยมีตัวแปรค่าควบคุมทางอินพุต สัมพันธ์กับทางเอาต์พุต

2. การเขียน function จากตารางความจริงสามารถเขียนได้ 2 รูปแบบคือ SOP, POS

3. การลดรูปลอจิกเกตจาก SOPหรือ POSโดยใช้พีชคณิตบูลลีน หรือ ใช้ K-map

4. นำค่าที่ได้จากการลดรูปมาต่อวงจรเกตเพื่อนำไปใช้ควบคุมหน้าที่การทำงาน

# อธิบาย

- หลักการออกแบบวงจรลอจิกเกต (Logic Combination)
- การเขียนตารางตามการทำงานที่เกิดขึ้นจริง ในรูปแบบ SOP กับPOS
- การลดรูปลอจิกเกตจาก SOP หรือ POS โดยใช้พีชคณิตบูลลีน หรือใช้ K-map เปรียบเทียบผลการทำงาน
- นำผลที่ได้จากการลดรูปมาต่อเป็นวงจรเกตเพื่อนำไปใช้ควบคุมหน้าที่การทำงานที่ถูกกำหนดในตารางความจริง

SOP (sum of product) เขียนจากค่า Out put 1

POS (product of sum) เขียนจากค่า Out put 0

| | A | B | C | Out Put | SOP | POS |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $\bar{A}\bar{B}\bar{C}$ | $A + B + C$ |
| 1 | 0 | 0 | 1 | 1 | $\bar{A}\bar{B}C$ | $A + B + \bar{C}$ |
| 2 | 0 | 1 | 0 | 0 | $\bar{A}B\bar{C}$ | $A + \bar{B} + C$ |
| 3 | 0 | 1 | 1 | 1 | $\bar{A}BC$ | $A + \bar{B} + \bar{C}$ |
| 4 | 1 | 0 | 0 | 1 | $A\bar{B}\bar{C}$ | $\bar{A} + B + C$ |
| 5 | 1 | 0 | 1 | 0 | $A\bar{B}C$ | $\bar{A} + B + \bar{C}$ |
| 6 | 1 | 1 | 0 | 1 | $AB\bar{C}$ | $\bar{A} + \bar{B} + C$ |
| 7 | 1 | 1 | 1 | 0 | $ABC$ | $\bar{A} + \bar{B} + \bar{C}$ |

**ตารางที่ 2**

# อธิบาย

- SOP เป็นคำย่อมาจาก(Canonical  Sum of Product form)
- POS เป็นคำย่อมาจาก(Canonical Product of Sum form)
- พิจารณาจากรูปเทอมของ SOP จะมีค่าตัวแปรทางอินพุต A,B,C เขียนจากเอาต์พุตจะเป็น 1

-มีค่า 0,0,1 รูปแบบทางเอาต์พุตจะเป็น $\overline{A}\,\overline{B}\,C$

0,1,1 รูปแบบทางเอาต์พุตจะเป็น $\overline{A}BC$

1,0,0 รูปแบบทางเอาต์พุตจะเป็น $A\overline{B}\overline{C}$

1,1,0 รูปแบบทางเอาต์พุตจะเป็น $AB\overline{C}$

# อธิบาย

- พิจารณาจากรูปเทอมของ POS จะมีค่าตัวแปรทางอินพุต A,B,C เขียนจากเอาต์พุตจะเป็น 0

  - มีค่า 0,0,0 รูปแบบทางเอาต์พุตจะเป็น $A+B+C$

    0,1,0 รูปแบบทางเอาต์พุตจะเป็น $A+\overline{B}+C$

    1,0,1 รูปแบบทางเอาต์พุตจะเป็น $A+\overline{B}+C$

    1,1,1 รูปแบบทางเอาต์พุตจะเป็น $\overline{A}+\overline{B}+\overline{C}$

- จากตารางจะเห็นว่าเอาต์พุตSOP กับ POS จะมีค่าเป็นComplement ซึ่งกันและกัน
- ข้อสังเกตค่าที่ได้จากการเขียน SOP กับ POS จะมีค่าเท่ากัน แต่ไม่เป็น Complement กัน

# 413 การออกแบบวงจรลอจิกเกตในเทอมของ minterm และ maxterm

เขียนจากตารางความจริงที่ค่าทางเอาต์พุตจะเป็น 1

**ในรูปแบบที่ 1 SOP (SUM OF PRODUCT)**

**Example. การหาค่า** $\sum_m$

$$SOP = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C}$$

$$F\ (A, B, C) = \sum m\ (1,3,4,6)$$

# อธิบาย

- การเขียนฟังก์ชันจากตารางความจริงสามารถเขียนได้ 2 รูปแบบคือ เขียนในรูปแบบ Sum of Product และ Product of Sum
- Sum of Product ค่าเอาต์พุตจะเกิดจากผลรวมแต่ละเทอมที่ถูกกระทำด้วย AND
- ในสไลด์นี้แสดงให้เห็นถึงการเขียนจาก Sum of Product ซึ่งจะพิจารณาค่าจากเอาต์พุตจะเป็น 1 $SOP = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + AB\overline{C}$
- จากสไลด์ที่ผ่านมานำค่าที่ได้มาเขียนในรูปแบบดังนี้
- หรือเขียนในรูปแบบสมการ

$$f(A,B,C) = \sum m(1,3,4,6)$$

# รูปแบบที่2 POS (Product Of Sum)

เขียนจากตารางความจริงที่ค่าทางเอาต์พุตเป็น 0 $\pi_M$

$$F(A, B, C) = \pi M(0, 2, 5, 7)$$

$$POS = (A+B+C)(A+\overline{B}+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+\overline{C})$$

# อธิบาย

- การเขียนฟังก์ชันจากตารางความจริง สามารถเขียนในรูปแบบที่สองคือเขียนในรูปแบบ (Product of sum)
- รูปแบบของ Product of sum จะเขียนจากเทอมที่มีค่าเอาต์พุตจะเป็น 0
- จากตารางค่าความจริงจะได้ค่าเอาต์พุตจะดังนี้

$$POS = (A + B + C)(A + \overline{B} + C)(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + \overline{C})$$

- สามารถเขียนในรูปแบบสมการได้ดังนี้

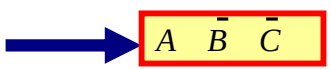
$$f(A, B, C) = \pi(0, 2, 5, 7)$$

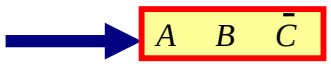
# ความสัมพันธ์ระหว่างตารางความจริงค่าเอาต์พุตที่เป็น 1 กับK-Map ชนิด 3 ตัวแปร

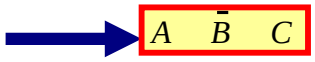
| | C | B | A | *output* |
|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 1 | 1 |
| **2** | 0 | 1 | 0 | 0 |
| **3** | 0 | 1 | 1 | 1 |
| **4** | 1 | 0 | 0 | 0 |
| **5** | 1 | 0 | 1 | 1 |
| **6** | 1 | 1 | 0 | 0 |
| **7** | 1 | 1 | 1 | 1 |

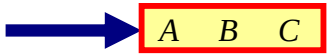
**ตารางความจริงที่3**

min-term



# อธิบาย

- ตารางค่าความสัมพันธ์ระหว่างตารางความจริงที่3 มีค่าเอาต์พุตจะเป็น 1
- การพิจารณาค่าเอาต์พุตจะที่มีค่าเป็น1 เรียกว่าการพิจารณาในรูปแบบ minterm
- ค่าของ $f(A,B,C) = \sum m(1,3,5,7)$
- เขียนฟังก์ชันทางเอาต์พุตจะ ที่มีค่าเป็น 1 คือ

Switching Function = $A\overline{B}\overline{C} + AB\overline{C} + A\overline{B}C + ABC$

# การลดรูปโดยใช้ K-Map ชนิด 3 ตัวแปร

## ¶M (1,3,5,7)

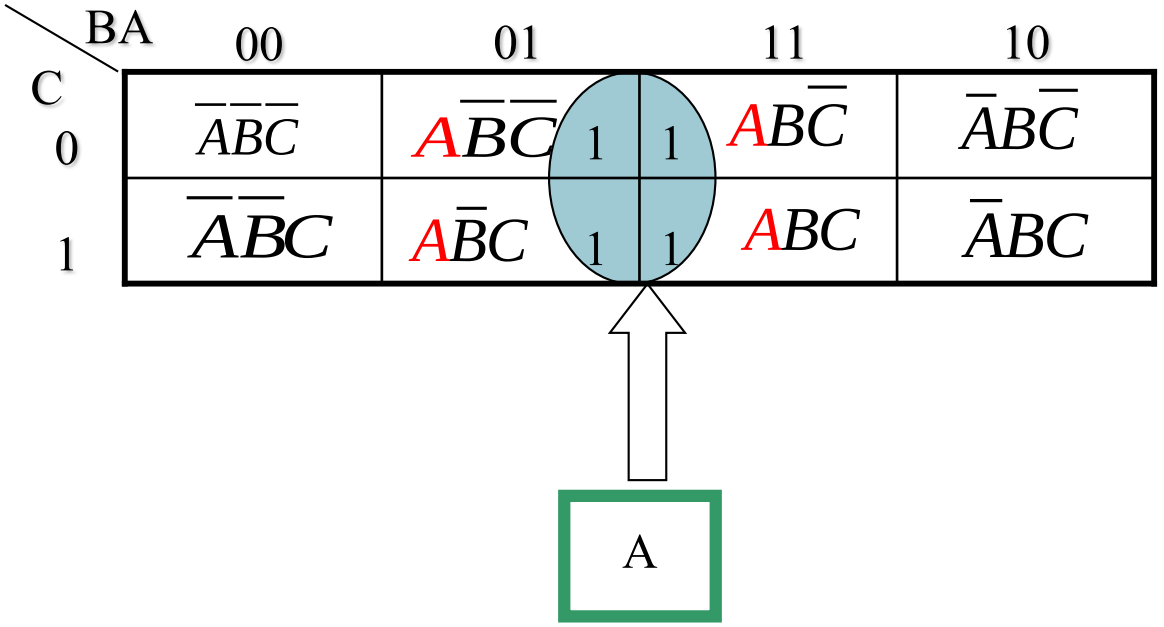
Min-term

1 → $A\,\bar{B}\,\bar{C}$

3 → $A\,B\,\bar{C}$

5 → $A\,\bar{B}\,C$

7 → $A\,B\,C$



# อธิบาย

- นำค่าที่อยู่ในรูป minterm ลงในตาราง k-map 3 ตัวแปร
- นำค่าที่กำหนดคือ 1,3,5,7 ลงในตาราง k-map
- ค่าที่อยู่ในตารางสามารถจับคู่ 4 ได้ค่าเป็น A

# ความสัมพันธ์ระหว่างตารางความจริงกับค่าเอาต์พุตที่เป็น 0 กับK-Map ชนิด 3 ตัวแปร $\sum m$

| C | B | A | output |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

ตารางความจริงที่ 4

$A+B+C$

$A+\bar{B}+C$

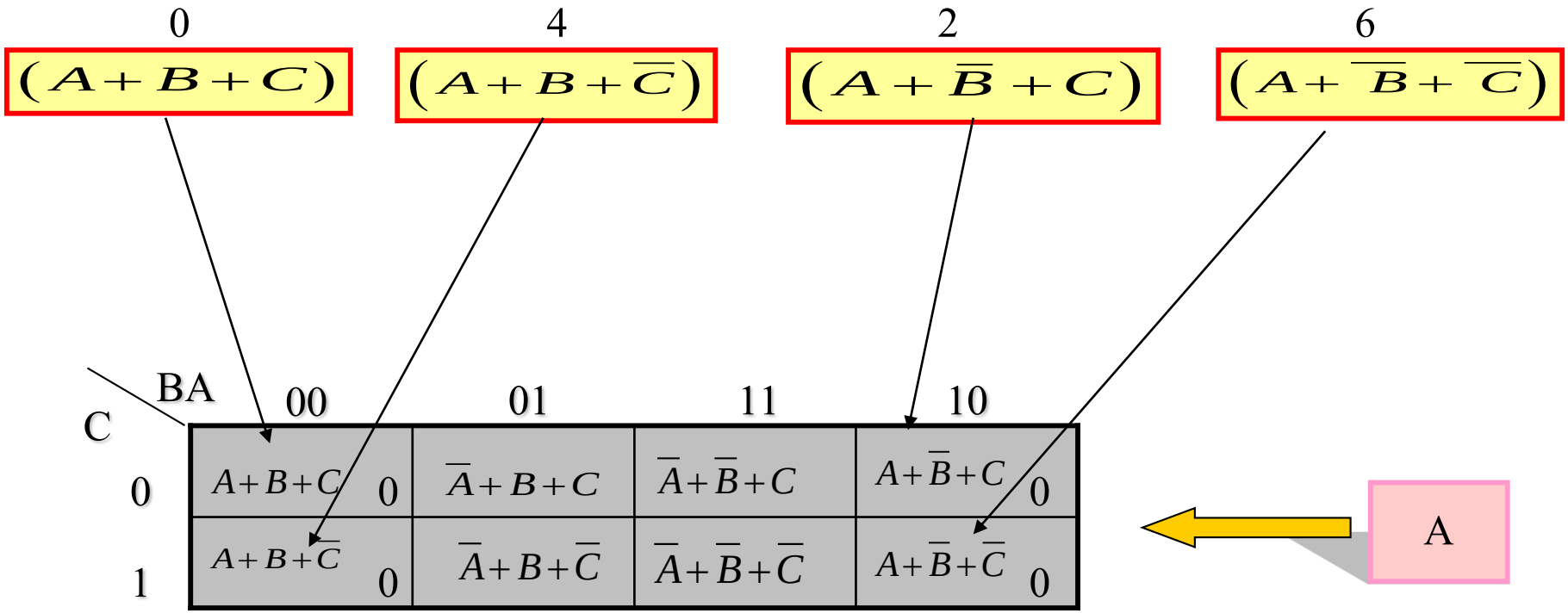
$A+B+\bar{C}$

$A+\bar{B}+\bar{C}$

Max-term

# ความสัมพันธ์ระหว่างตารางความจริงค่าเอาต์พุตที่เป็น 0 กับ K-Map ชนิด 3 ตัวแปร

**Max-term** $\pi m$



# อธิบาย

- การเขียนฟังก์ชันจากตารางความจริงในรูปแบบ POS หรือเขียนจากค่า Maxterm
- รูปแบบของ Maxterm เป็นไปดังตารางโดยพิจารณาจากเอาต์พุตที่มีค่า 0
- จากรูปจับเทอม 0, 4, 2, 6 ในรูป Maxterm ค่าที่ได้จากการจับลดรูปได้เป็น A
- ค่าที่ได้จากตารางเดียวกันสามารถใช้SOP หรือ POS จะได้ค่าผลลัพธ์เท่ากันคือ A
- การพิจารณาที่จะใช้ค่า SOP หรือ POS ให้พิจารณาว่าจำนวนเทอมของ SOP หรือ POS น้อยที่สุดให้เลือกเทอมนั้น จะได้ฟังก์ชันของค่าเอาต์พุตน้อยที่สุด

# การออกแบบวงจรคอมไบเนชัน

- กำหนดตารางความจริงที่4 ตามหน้าที่การทำงานที่ออกแบบไว้โดยมีอินพุตควบคุม 4 ตัวแปรคือ D,C,B,A และมีเอาต์พุต ที่ได้ดังตารางเป็นแสดงทั้งค่า SOP กับ POS คือทั้งค่าที่เป็น 1 กับ 0 ดังนี้คือ

$$\sum m \ (0,1,2,3,4,6,8,10,12,14)$$

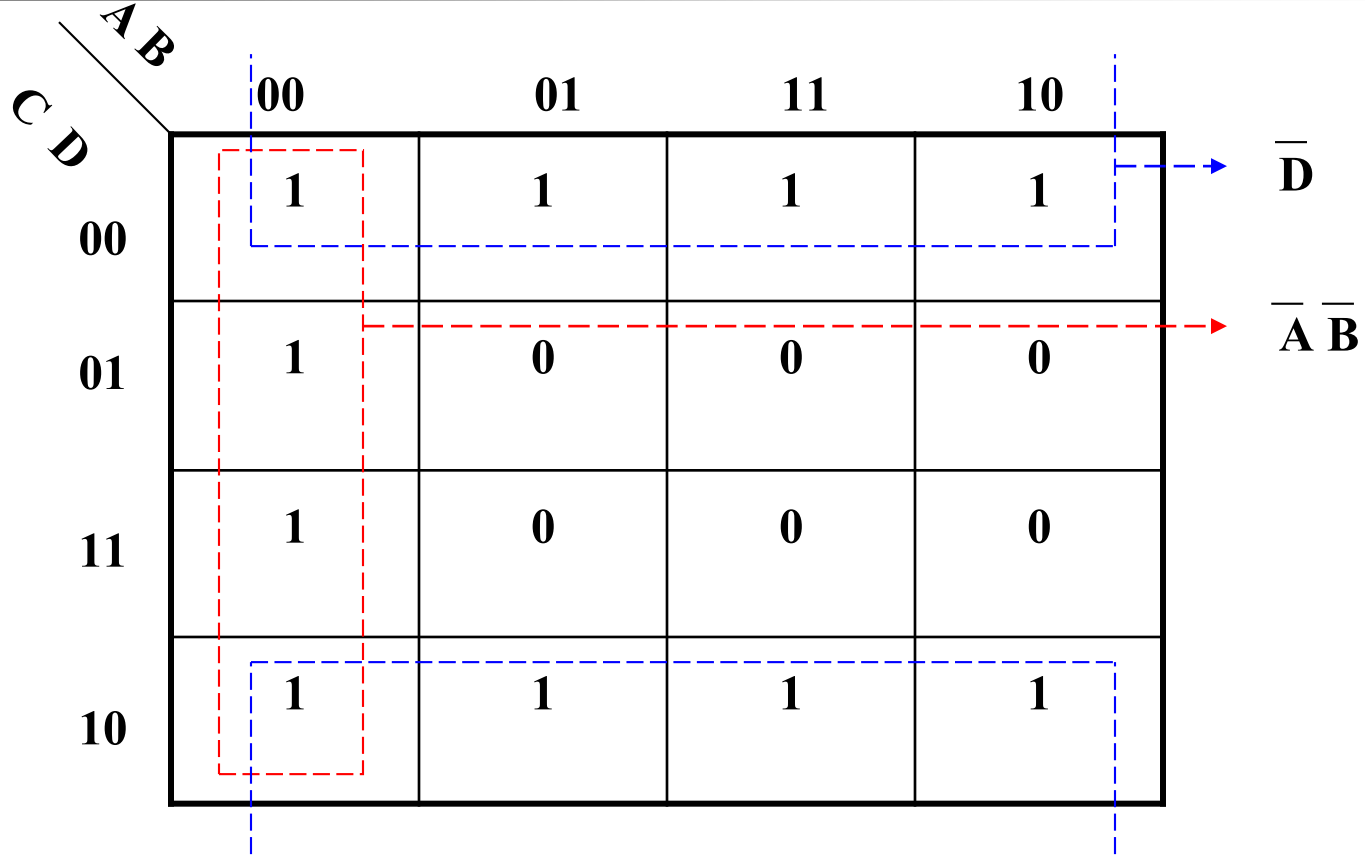
$$\pi M \ (5,7,9,11,13,15)$$

| | D | C | B | A | F | POS | SOP |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | | $\bar{A}\bar{B}\bar{C}\bar{D}$ |
| 1 | 0 | 0 | 0 | 1 | 1 | | $\bar{A}\bar{B}\bar{C}D$ |
| 2 | 0 | 0 | 1 | 0 | 1 | | $\bar{A}\bar{B}C\bar{D}$ |
| 3 | 0 | 0 | 1 | 1 | 1 | | $\bar{A}\bar{B}CD$ |
| 4 | 0 | 1 | 0 | 0 | 1 | | $\bar{A}B\bar{C}\bar{D}$ |
| 5 | 0 | 1 | 0 | 1 | 0 | $\bar{A}+B+\bar{C}+D$ | |
| 6 | 0 | 1 | 1 | 0 | 1 | | $\bar{A}BC\bar{D}$ |
| 7 | 0 | 1 | 1 | 1 | 0 | $\bar{A}+\bar{B}+\bar{C}+D$ | |
| 8 | 1 | 0 | 0 | 0 | 1 | | $A\bar{B}\bar{C}\bar{D}$ |
| 9 | 1 | 0 | 0 | 1 | 0 | $\bar{A}+B+C+\bar{D}$ | |
| 10 | 1 | 0 | 1 | 0 | 1 | | $A\bar{B}C\bar{D}$ |
| 11 | 1 | 0 | 1 | 1 | 0 | $\bar{A}+\bar{B}+C+\bar{D}$ | |
| 12 | 1 | 1 | 0 | 0 | 1 | | $AB\bar{C}\bar{D}$ |
| 13 | 1 | 1 | 0 | 1 | 0 | $\bar{A}+B+\bar{C}+\bar{D}$ | |
| 14 | 1 | 1 | 1 | 0 | 1 | | $\bar{A}BC\bar{D}$ |
| 15 | 1 | 1 | 1 | 1 | 0 | $\bar{A}+\bar{B}+\bar{C}+\bar{D}$ | |

ตารางความจริงที่5

# ตัวอย่างที่ 4.1 จากตารางความจริงที่ 5ได้

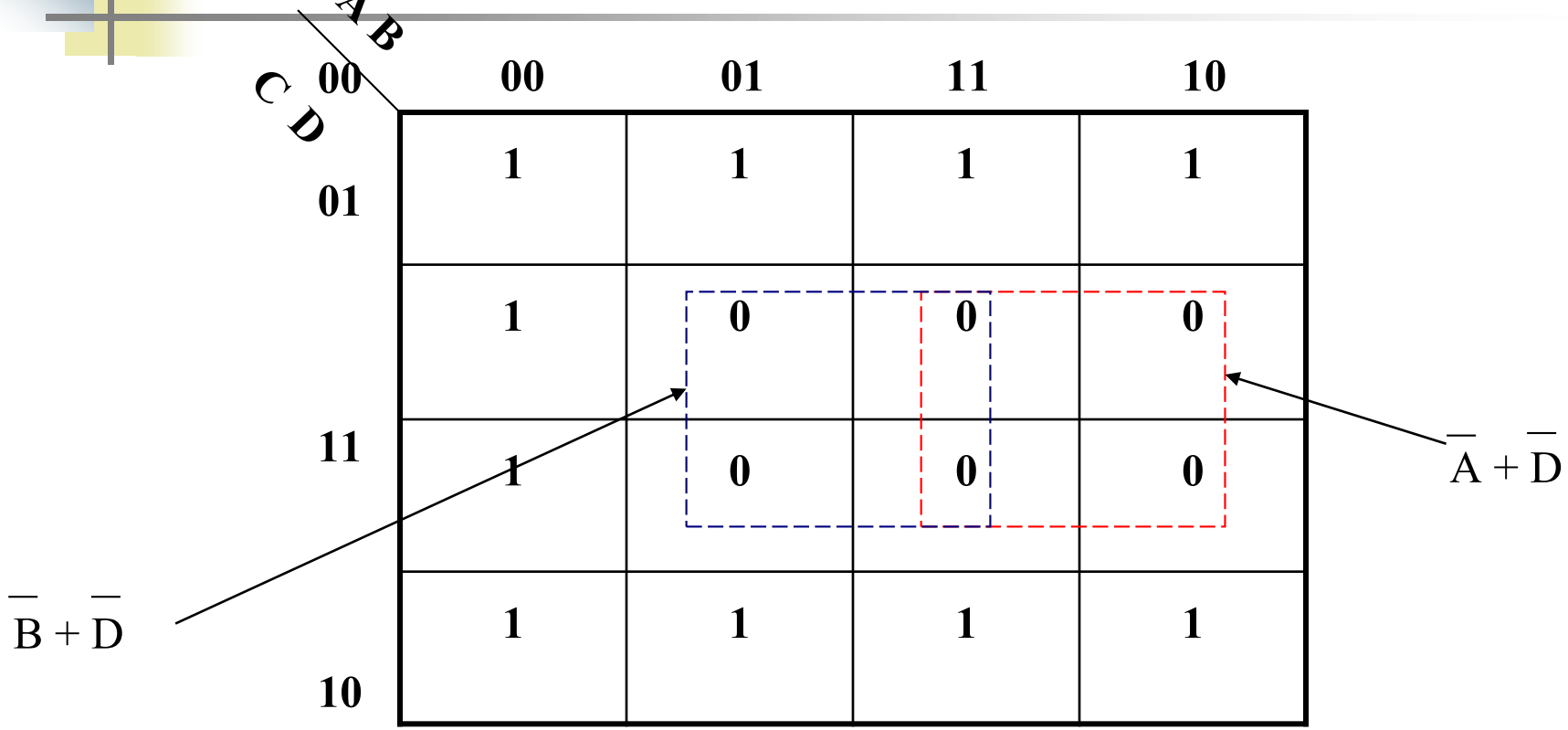
## Function $\Sigma$ m (0,1,2,3,4,6,8,10,12,14)




$$\Sigma m\,(0,1,2,3,4,6,8,10,12,14) = \overline{A}\,\overline{B} + \overline{D}$$

# ตัวอย่างที่ 4.2 จากตารางความจริงที่5 ได้ค่าฟังก์ชัน

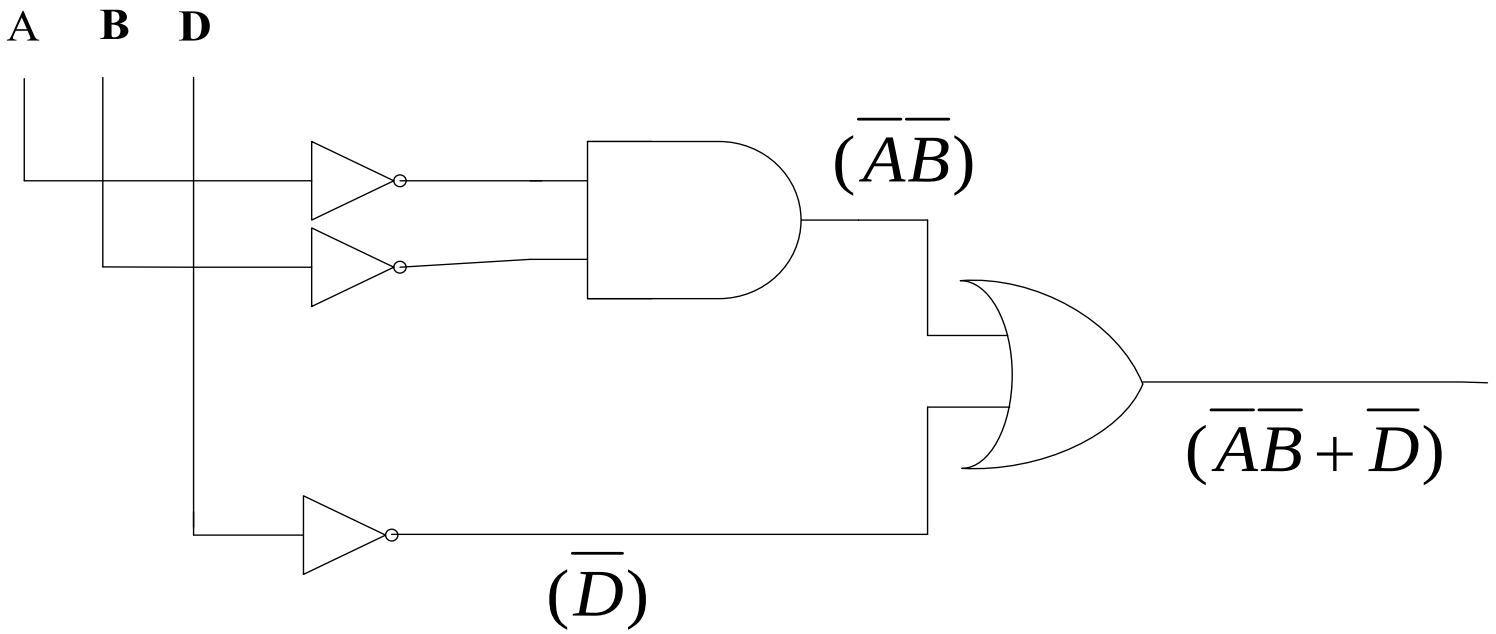
## ¶ M (5,7,9,11,13,15)




$$\P M(5,7,9,11,13,15) = (\overline{B}+\overline{D}).(\overline{A}+\overline{D})$$

$$= \overline{D}+\overline{A}\,\overline{B}$$

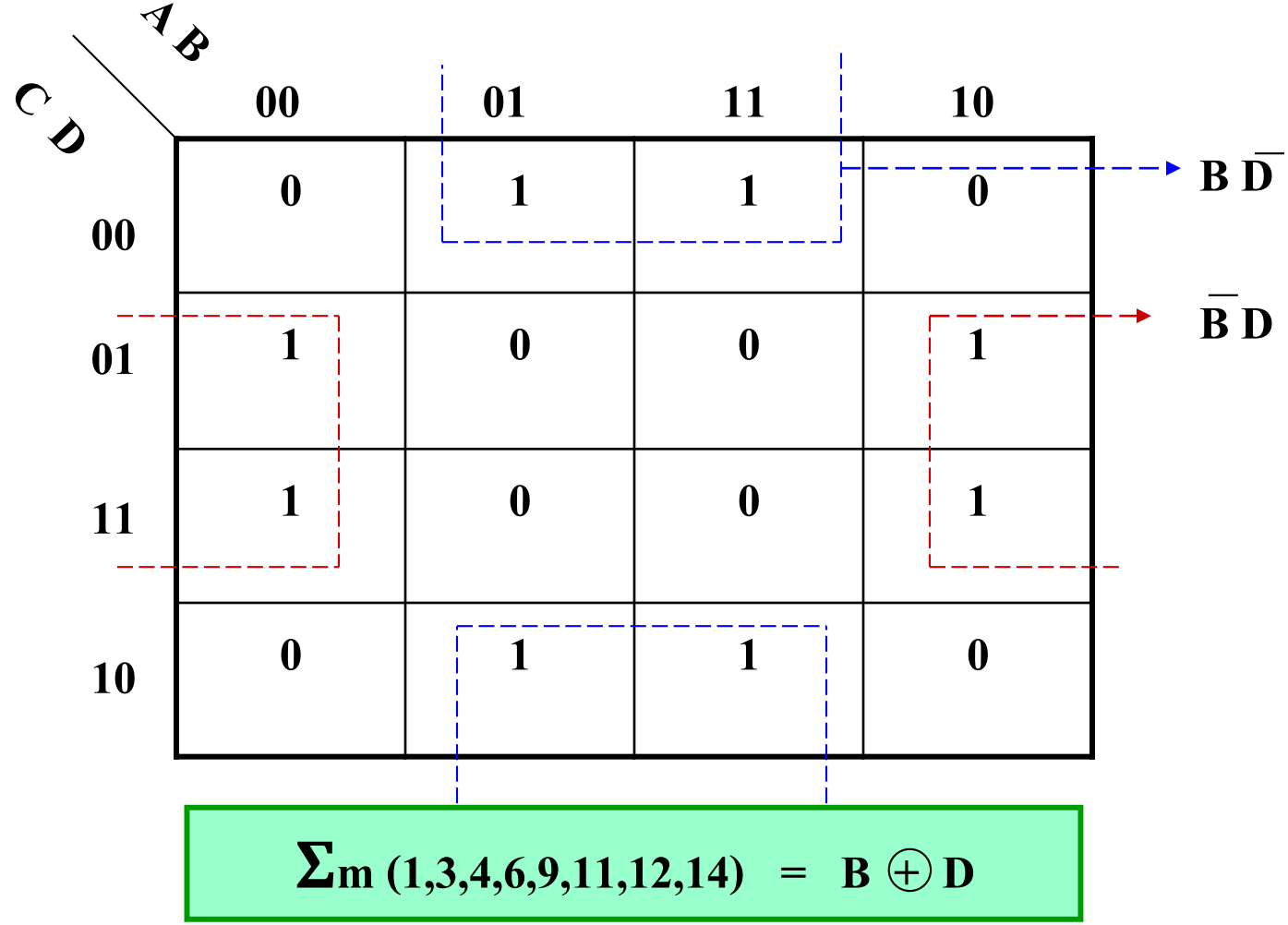
# วงจรลอจิกเกตที่ถูกลดรูปจากตารางK-map จากตัวอย่างที่4.1, 4.2



# อธิบาย

- การออกแบบวงจรคอมไบเนชัน จากตารางความจริงที่ 5ในรูปแบบ SOP , POS
- นำค่า SOP มาเขียนลงในตาราง K-map ทำการลดรูปได้ค่าฟังก์ชันเป็น $\overline{A}\overline{B}+\overline{D}$
- นำค่า POS มาเขียนลงในตาราง K-map ทำการลดรูปได้ค่าฟังก์ชันเป็น $(\overline{B}+\overline{D})(\overline{A}+\overline{D})$ ซึ่งมีค่าเท่ากันกับ SOP
- ได้ค่าแล้วนำไปเขียนวงจรลอจิกเกต $\overline{A}\overline{B}+\overline{D}$

# ตัวอย่างที่ 4.3 การออกแบบวงจรคอมไบเนชัน ที่มี Function $\Sigma m$ (1,3,4,6,9,11,12,14)



# อธิบาย

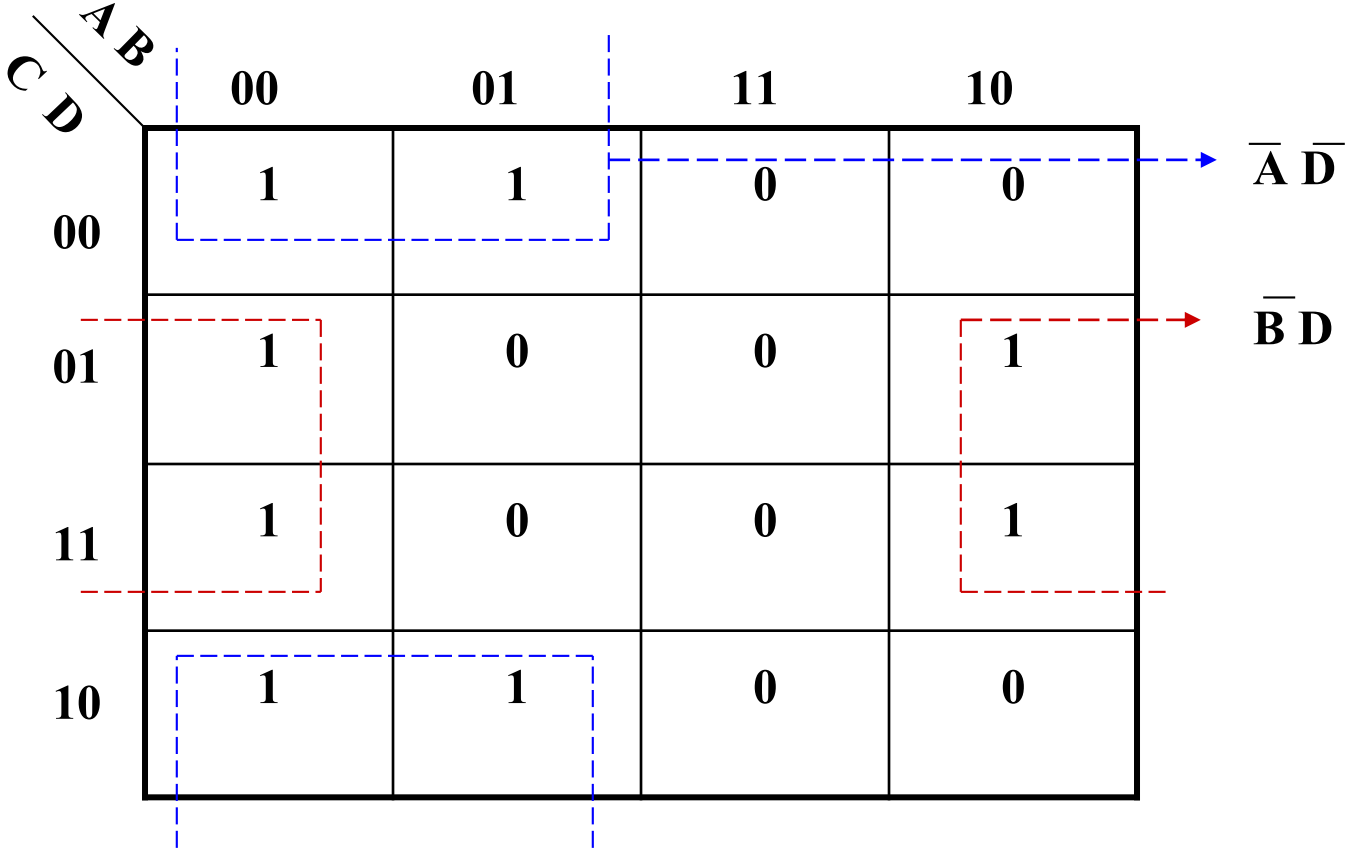
**จากตัวอย่างที่ 4.3**

- ในรูปแบบของ SOP ที่กำหนดค่าเอาต์พุตเป็น 1 ที่ตำแหน่ง 1 , 3 , 4 , 6 , 9 , 11 , 12 , 16
- สามารถเขียนลงในตาราง K-MAP 4 ตัวแปรกำหนดเป็น $\bar{A}B\bar{C}D$
- นำค่าที่เป็น 1 ในตารางมาลดรูปโดยจับ 4 ได้เป็นค่า 2 เทอม เทอมบนจับคู่ 4 ได้ค่าเป็น

  เป็น BD เทอมขวาและซ้ายได้ค่าเป็น BD
- วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-mapเป็นค่าของวงจร EX-OR Gate
- ตัวแปรคือ B EX –OR D ดังรูปตาราง K-map

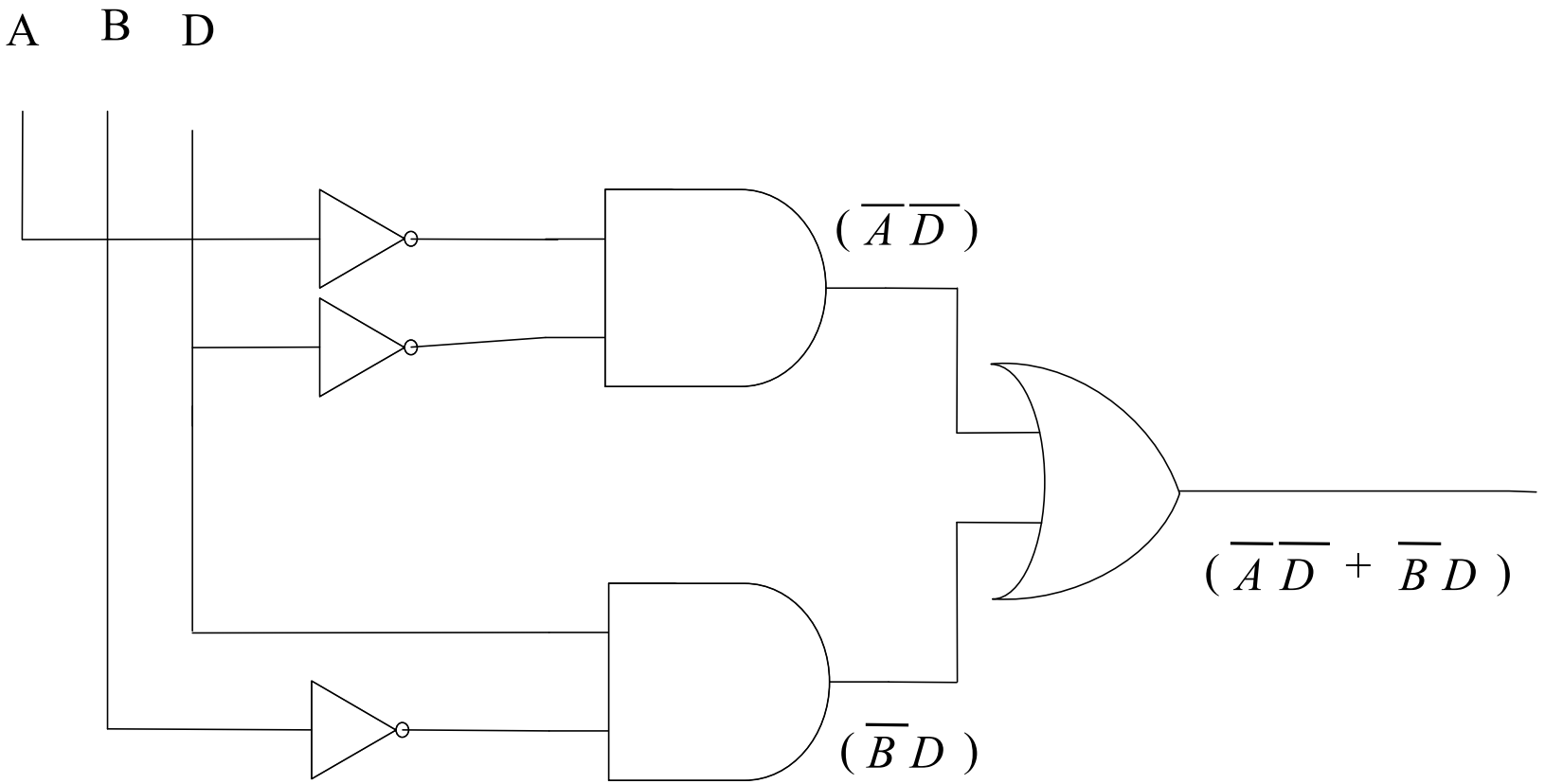
# ตัวอย่างที่ 4.4 การออกแบบวงจรคอมไบเนชันที่มี

## Function $\Sigma m\ (0,1,2,3,4,6,9,11)$

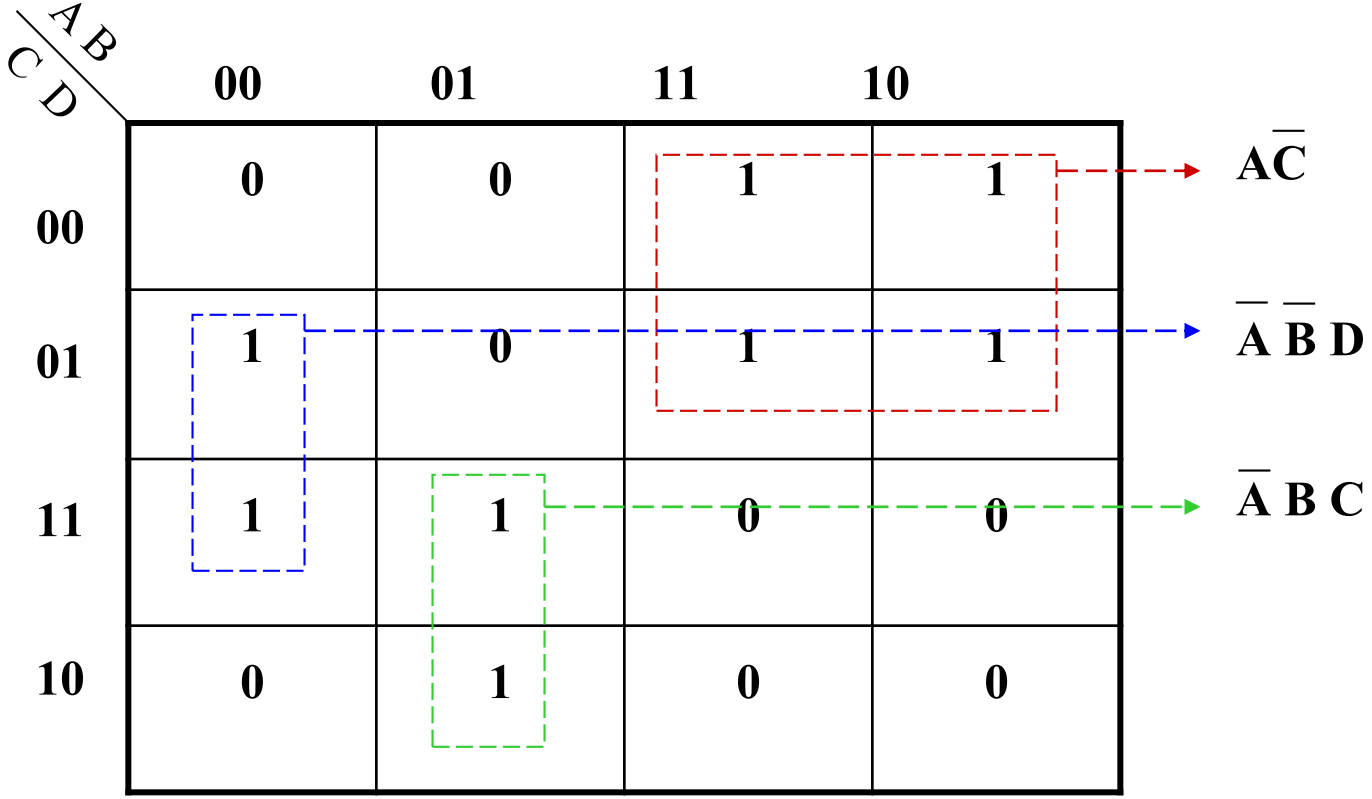



$$\Sigma m\ (0,1,2,3,4,6,9,11) = \bar{A}\,\bar{D} + \bar{B}\,D$$

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.4

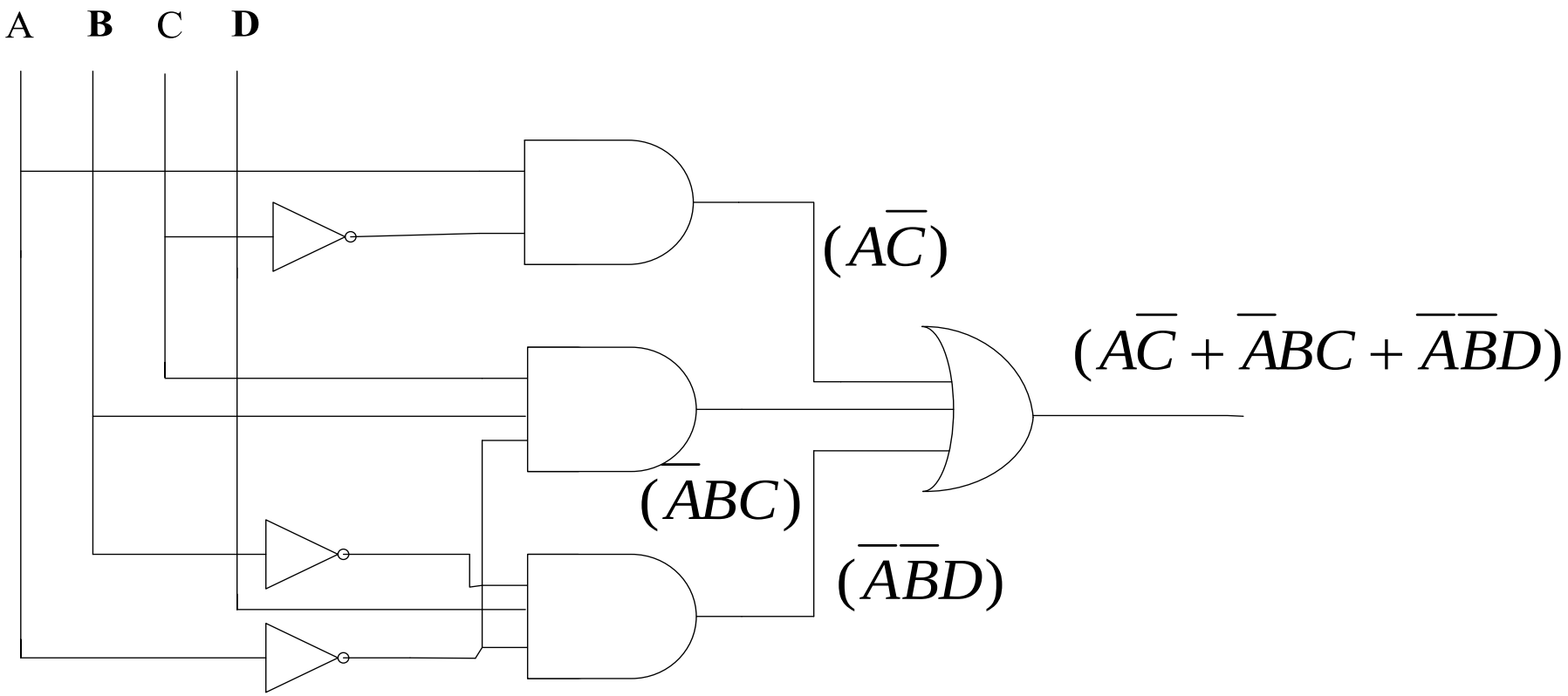


# ตัวอย่างที่ 4.5 การออกแบบวงจรคอมไบเนชัน ที่มี Function $\Sigma$ m (1,3,6,7,8,9,12,13)




$$\Sigma m\,(1,3,6,7,8,9,12,13) = A\,\overline{C} + \overline{A}\,\overline{B}\,D + \overline{A}\,B\,C$$

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.5



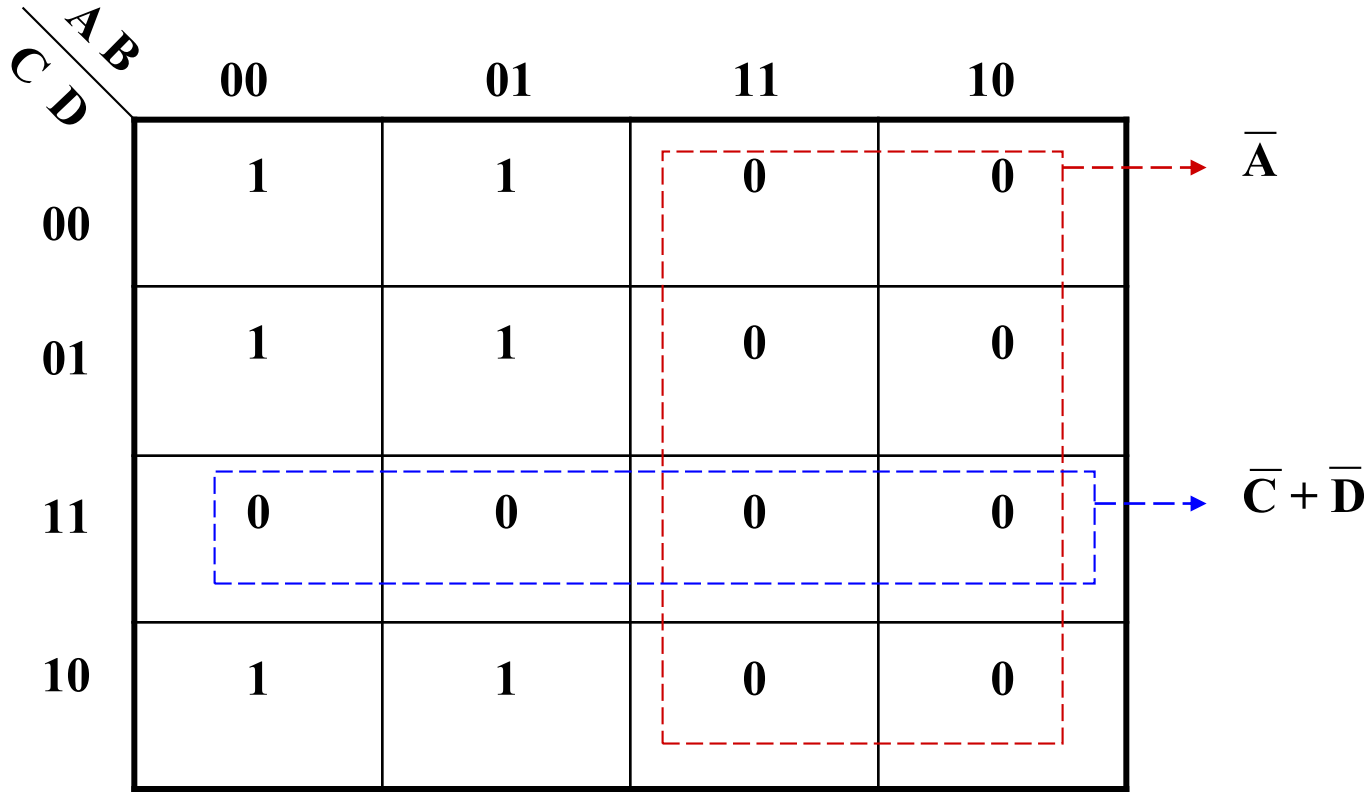
# อธิบาย

- จากตัวอย่างที่4.1 กับ 4.2 แสดงการออกแบบวงจรคอมไบเนชัน และนำไปลดรูปด้วย k-map ทั้งในรูปแบบ minterm และ Maxterm

  เปรียบเทียบให้เห็นถึงค่าเอาต์พุตที่ได้จะมีค่าเท่ากัน
- ส่วนในตัวอย่างที่ 4.3 ถึง 4.5 เป็นตัวอย่างการออกแบบวงจรคอมไบเนชันในรูปแบบ minterm แล้วทำการลดรูปโดยใช้ k-map
- จากนั้นนำไปเขียนวงจรลอจิกเกต

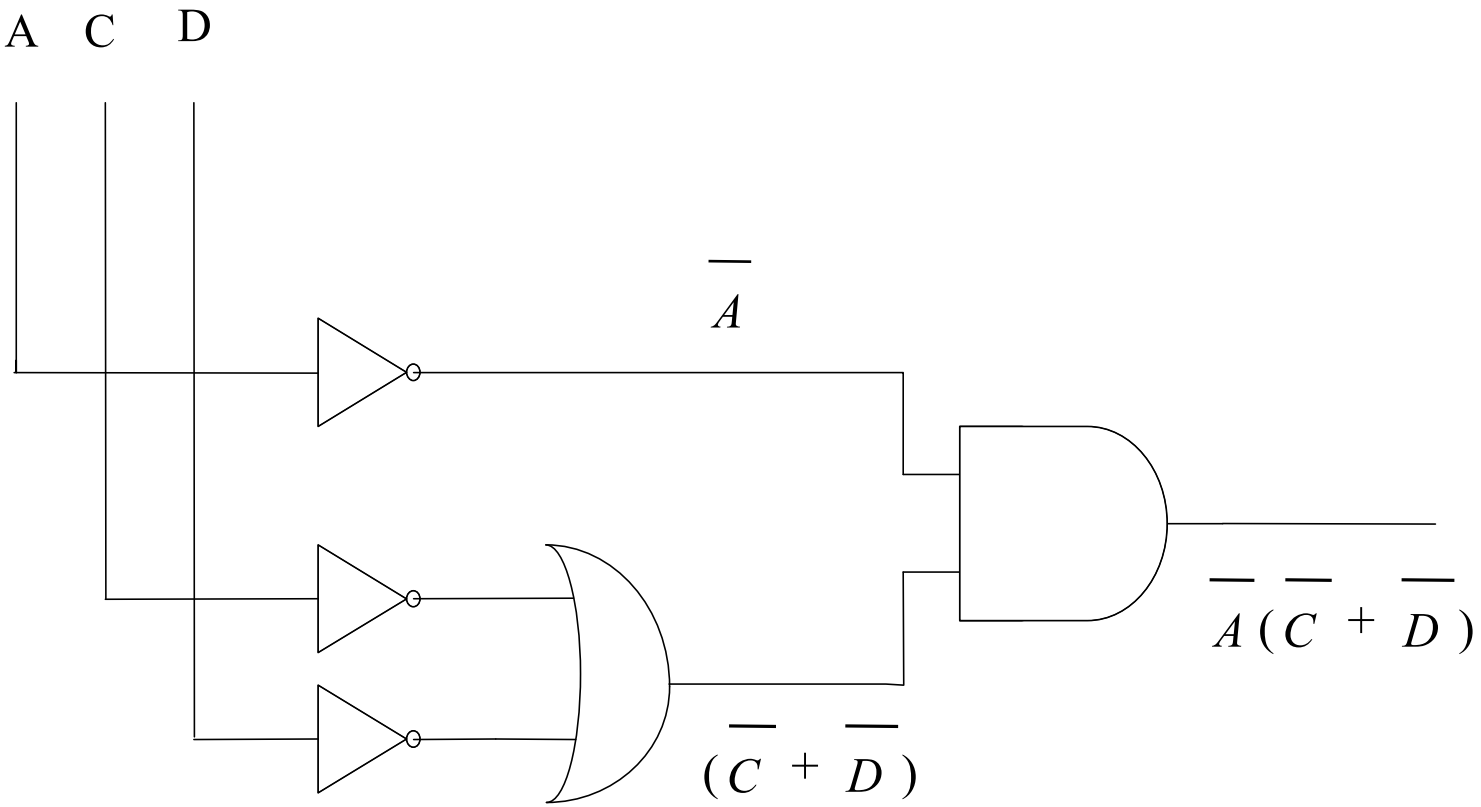
# ตัวอย่างที่ 4.6 การออกแบบวงจรคอมไบเนชันที่มี

**Function ¶ M ( 3,7,8,9,10,11,12,13,14,15 )**



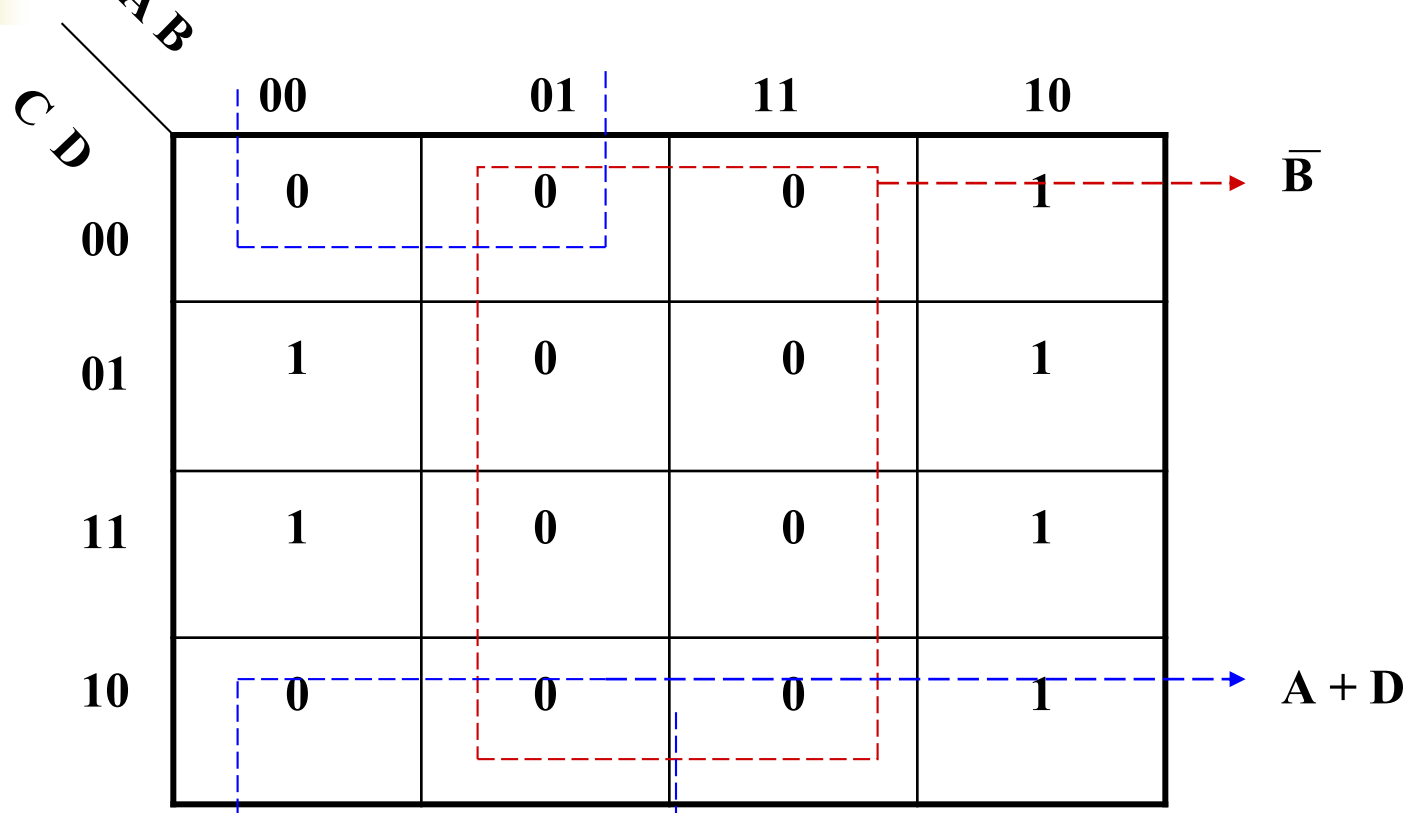

$$\P m\,(3,7,8,9,10,11,12,13,14,15) = \overline{A}(\overline{C} + \overline{D})$$

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.6



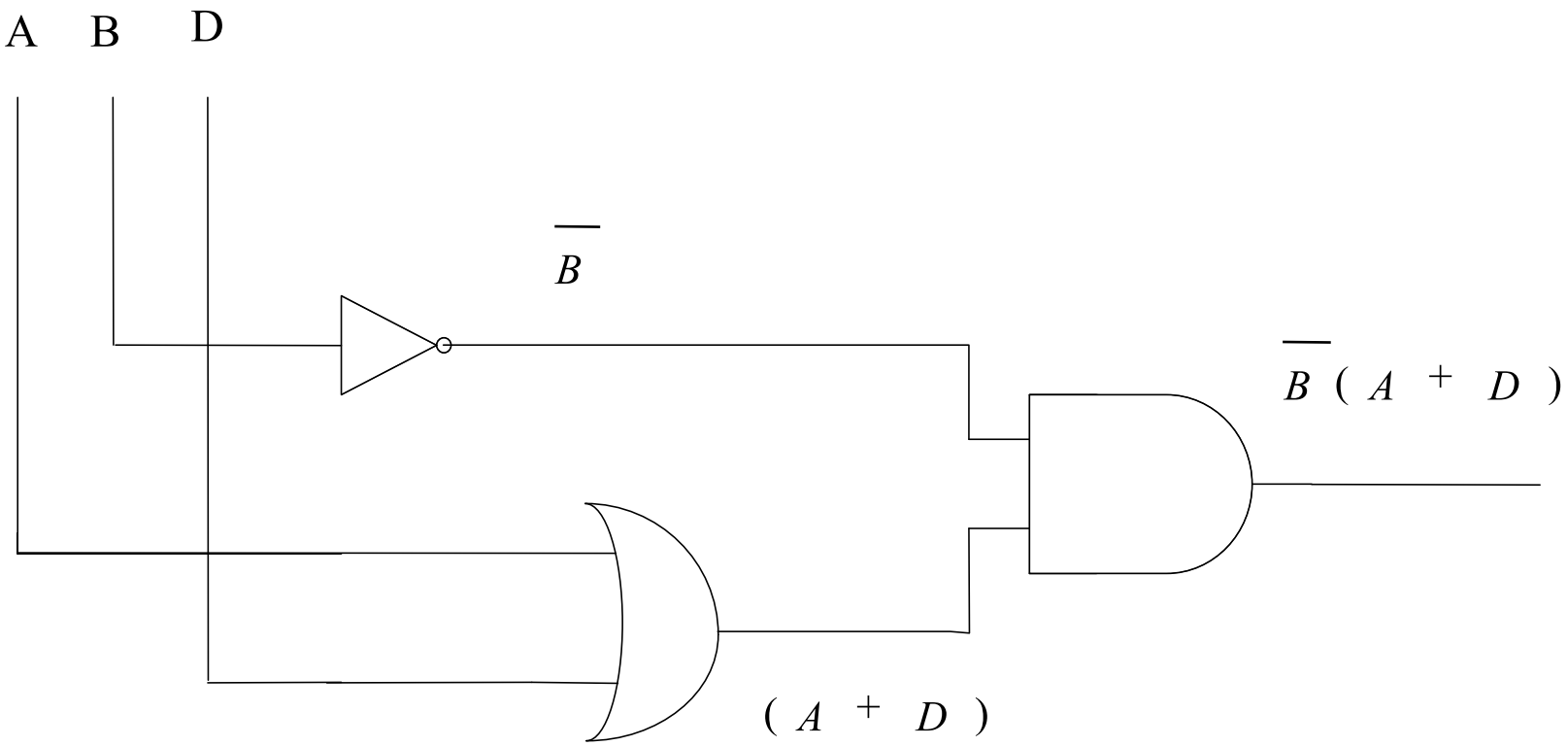
# ตัวอย่างที่ 4.7 การออกแบบวงจรคอมไบเนชันที่มี Function

## ¶M (0,2,4,5,6,7,12,13,14,15)



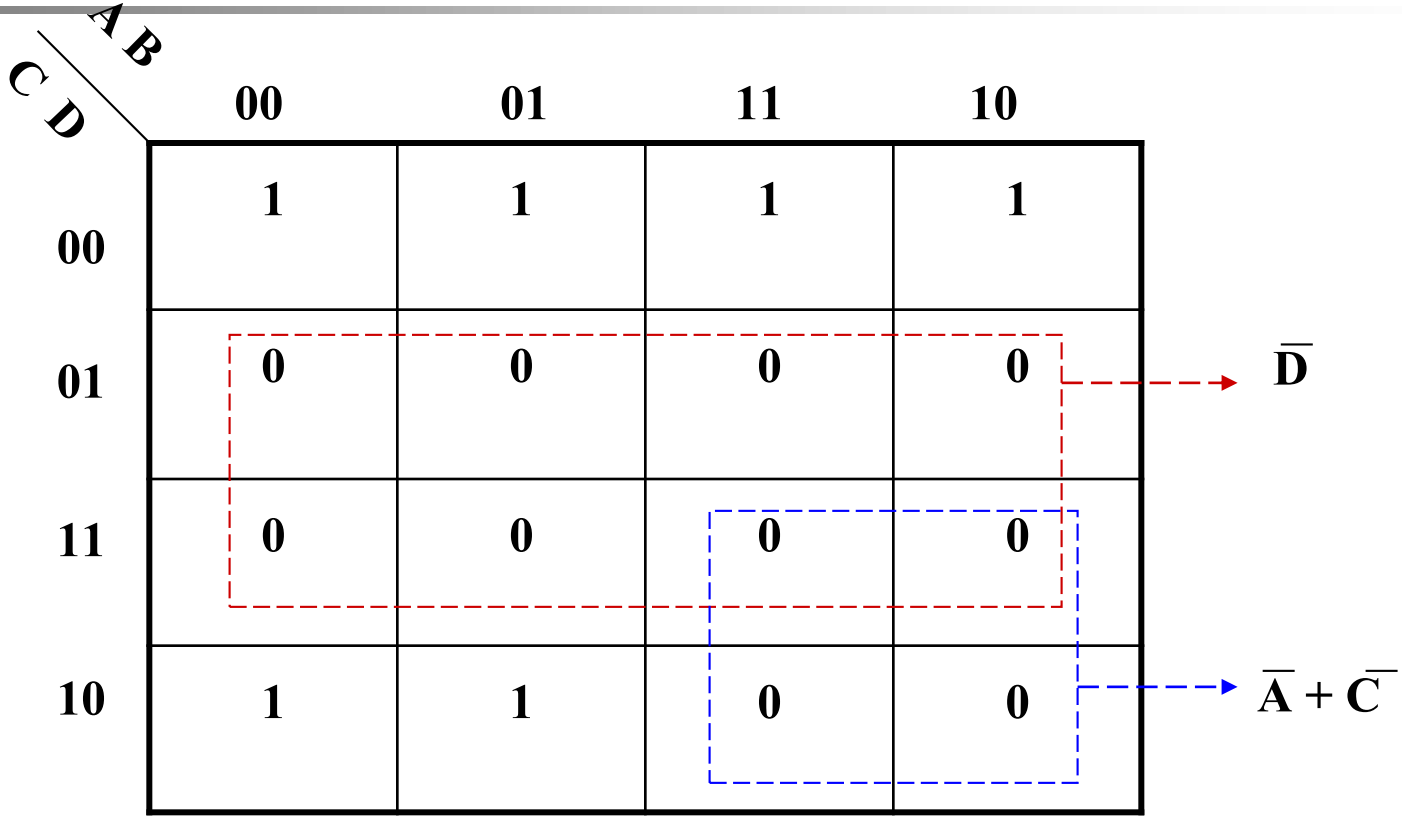

¶m (0,2,4,5,6,7,12,13,14,15) = $\overline{B}$ (A + D)

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.7



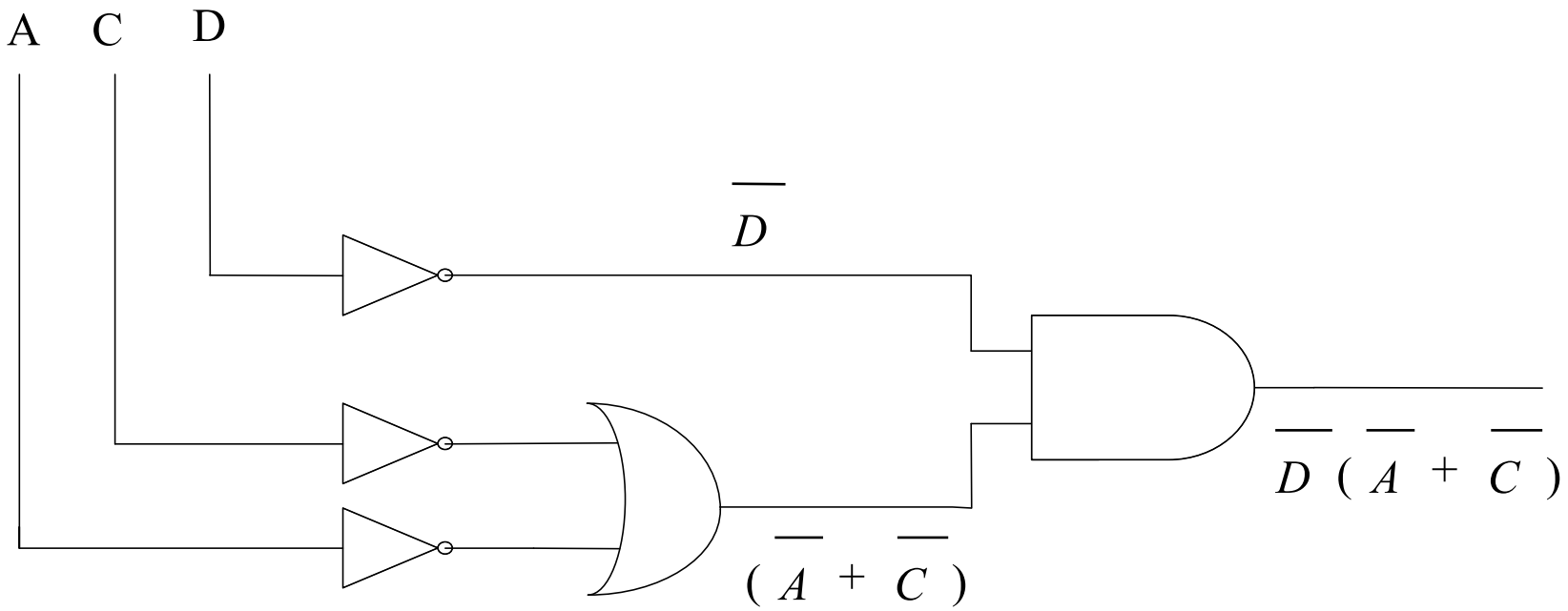
# ตัวอย่างที่ 4.8 การออกแบบวงจรคอมไบเนชันที่มี

## Function ¶M (1,3,5,7,9,10,11,13,14,15)

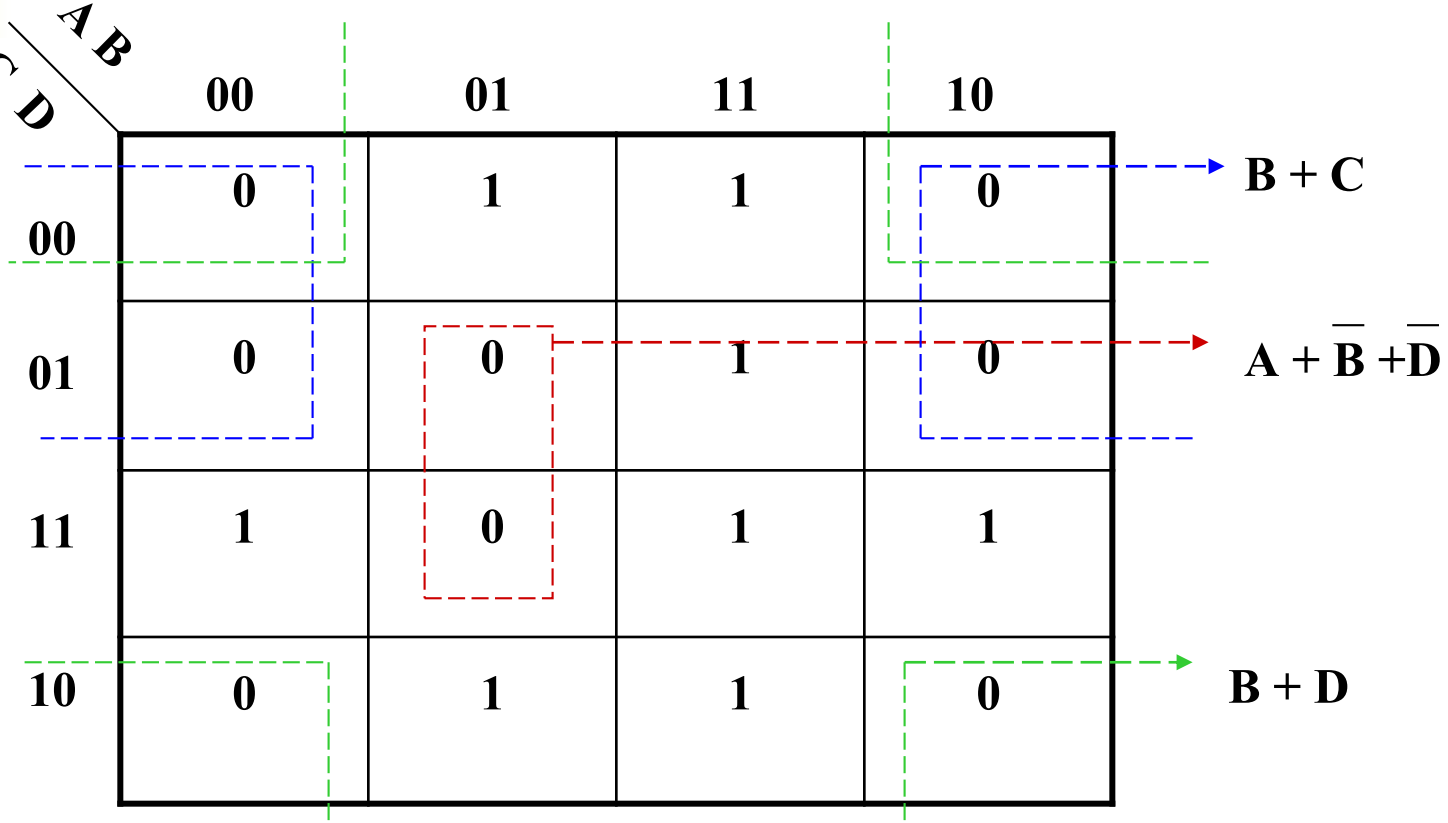



$$¶m\,(1,3,5,7,9,10,11,13,14,15) = \overline{D}(\overline{A} + \overline{C})$$

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.8

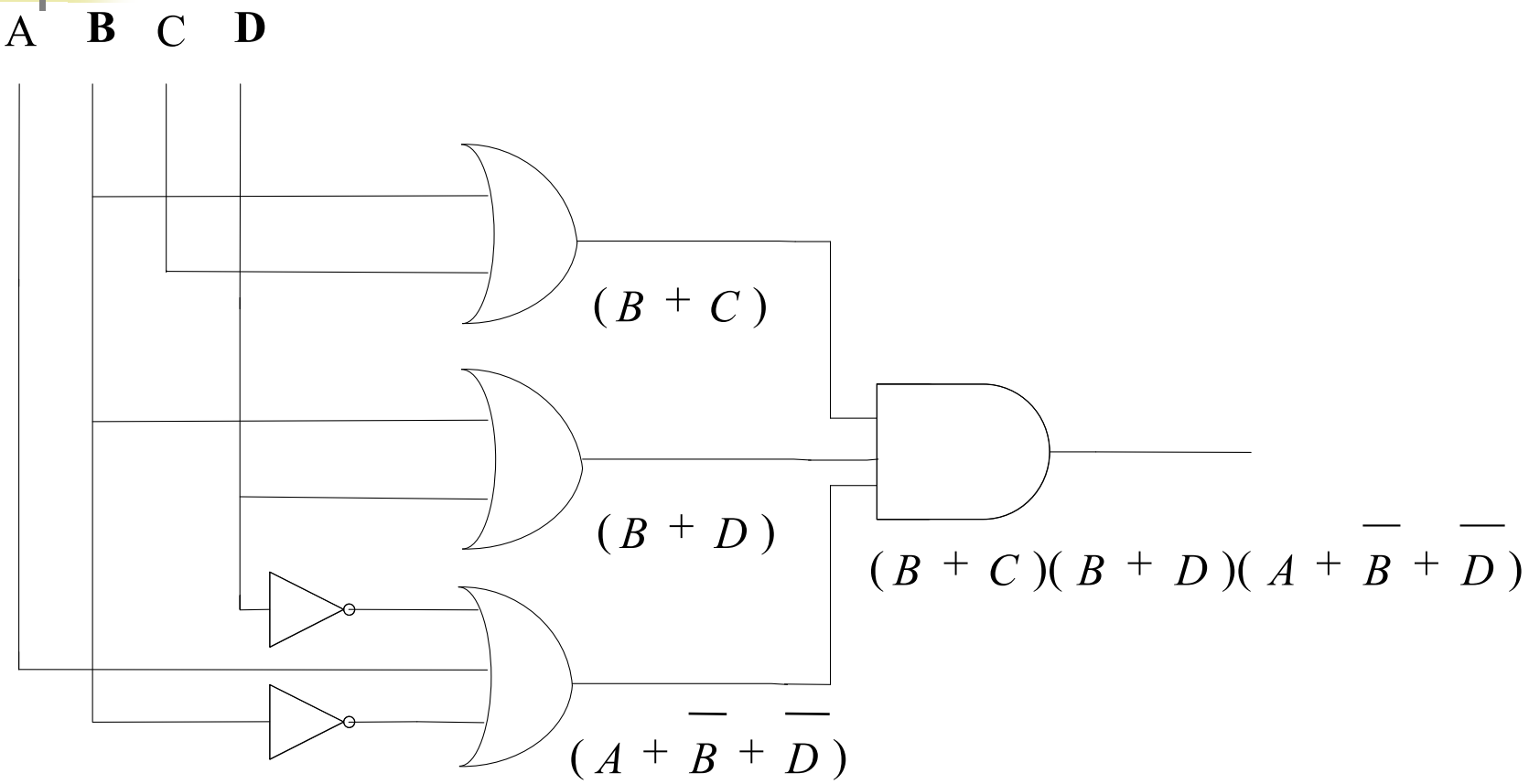


# ตัวอย่างที่ 4.9 การออกแบบวงจรคอมไบเนชัน ที่มี FunctionM (0,1,2,5,7,8,9,10)




¶m (0,1,2,5,7,8,9,10) = (B + C) (B + D) ( A + $\overline{B}$ + $\overline{D}$)

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.9



# อธิบาย

- จากตัวอย่างที่ 4.7-4.10 เป็นตัวอย่างการออกแบบวงจรคอมไบเนชัน จากตารางK-map ที่ให้มาในรูปแบบของ Maxterm
- กำหนดค่าลงในตารางK-map ทำการลดรูปสมการให้เหลือน้อยที่สุด
- นำไปเขียนวงจรเกต 3 เทอมที่กระทำด้วยแอนด์ ดังรูปตัวอย่างที่ 4.9

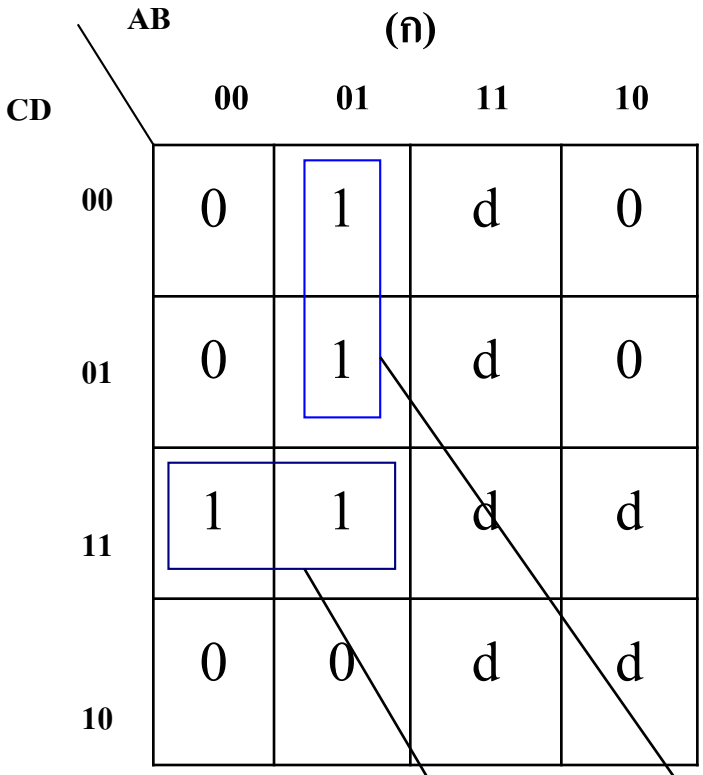
# 414 การใช้ Don’t care terms

- รหัสฐานสองที่มีค่าเท่ากับฐานสิบในรูปแบบBinary บิตที่เกิดจะขึ้นอยู่กับค่าฐานสิบ
- แต่มีรหัสที่มีขนาด 4 บิต เช่นรหัสแบบ BCD-8421, Excess-3 , 2421 ที่มีค่าไม่เกิน4 บิต
- หากค่าเกินจากค่า0-9ในฐานสิบก็จะทดไปด้านซ้ายอีก4บิต เช่นตัวอย่างของรหัสBCD- 8421
- $(9)_{10}$ = (1001)
- $(10)_{10}$ = (0001 0000)
- $(12)_{10}$ = (0001 0010) ซึ่งจะกล่าวละเอียดในสัปดาห์ที่5
- ในสัปดาห์นี้จะกล่าวถึงค่ารหัสที่เกินจากค่า 9 จำนวน 6ค่า คือค่าที่ไม่ใช้งานแต่นำมาใช้ให้เกิดประโยชน์ทางการลดรูปวงจรโดยใช้ K-map
- ในการออกแบบวงจรลอจิกค่าที่เกินจาก9 จำนวน 6 ค่า คือค่า10ถึง15ของฐานสิบ จึงสามารถกำหนดการใช้งานให้เป็น 0 หรือ 1 ก็ได้ ค่านี้เรียกว่าค่า(Don’t Care) ใช้อักษร X แทนค่าดังนี้

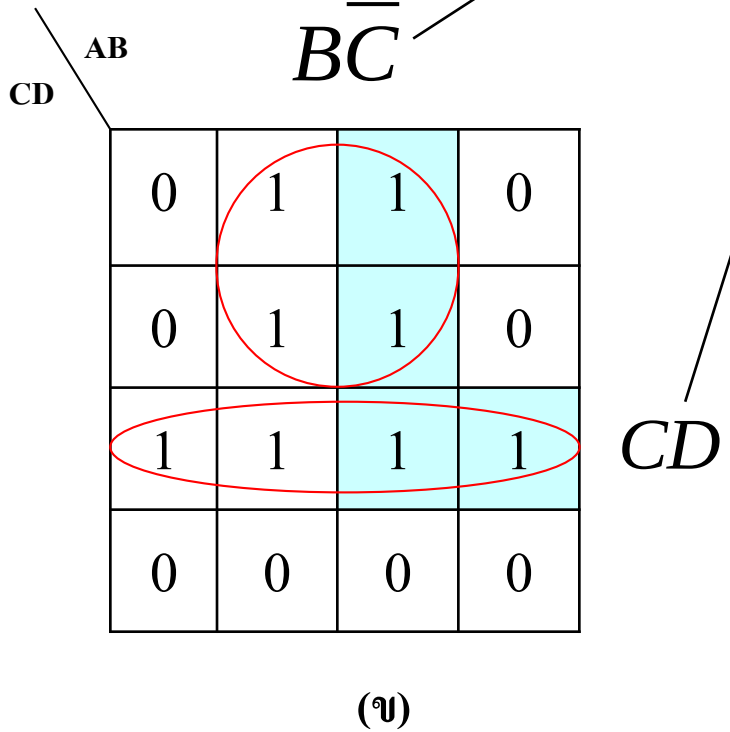
  10 = XXXX , 11 = XXXX , 12 = XXXX ,13 = XXXX , 14 = XXXX , 15 = XXXX
- ใช้อักษรย่อของDon’t Care คือ d ตาราง K-Map
- พิจารณาจากตารางความจริงในรูปแบบ SOP 3,4,5,7

| ฐาน 10 | Input BCD-8421 | | | | เอาต์พุต |
|---|---|---|---|---|---|
| 0-15 | A | B | C | D | X |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 |
| 10 | 1 | 0 | 1 | 0 | d |
| 11 | 1 | 0 | 1 | 1 | d |
| 12 | 1 | 1 | 0 | 0 | d |
| 13 | 1 | 1 | 0 | 1 | d |
| 14 | 1 | 1 | 1 | 0 | d |
| 15 | 1 | 1 | 1 | 1 | d |

# การใช้งานของ Don't Care




$$X = \overline{A}CD + \overline{A}B\overline{C}$$




$$X = B\overline{C} + CD$$

# อธิบาย

-จากตารางความจริงกำหนดค่า SOP 3,4,5,7 ส่วนค่า 10,11,12,13,14,15 เป็นค่าที่ไม่มีการใช้งาน จึงกำหนดเป็น1ก็ได้หรือเป็น0 ก็ได้

-จากตาราง(ก) ค่าที่เกิดจากSOP 3,4,5,7 เมื่อนำลง k-Map ค่าที่ลดรูปได้เป็น

$$X = \overline{A}CD + \overline{A}B\overline{C}$$

-จากตาราง(ข) มีการนำค่าSOP 3,4,5,7 พร้อมกับค่า Don't 11,12,13,15 มาใช้ช่วยในการจับคู่ 4 สามารถลดรูปได้เป็น

$$X = B\overline{C} + CD$$

-การนำd มาใช้งานในK-map สามารถลดค่าคอมพลีเมนต์Aได้ พิจารณาจากเอาต์พุตทั้งสองจะมีค่าทางการฟังก์ชันท่ากัน

-โดยกำหนดค่า d เทอมของ 14,10 กำหนดให้เป็น 0

# 42 การเขียนลอจิกเกตจากฟังก์ชันพีชคณิต

## หลักการเขียนลอจิกเกต

- การเขียนลอจิกเบื้องต้น มีลำดับดังนี้

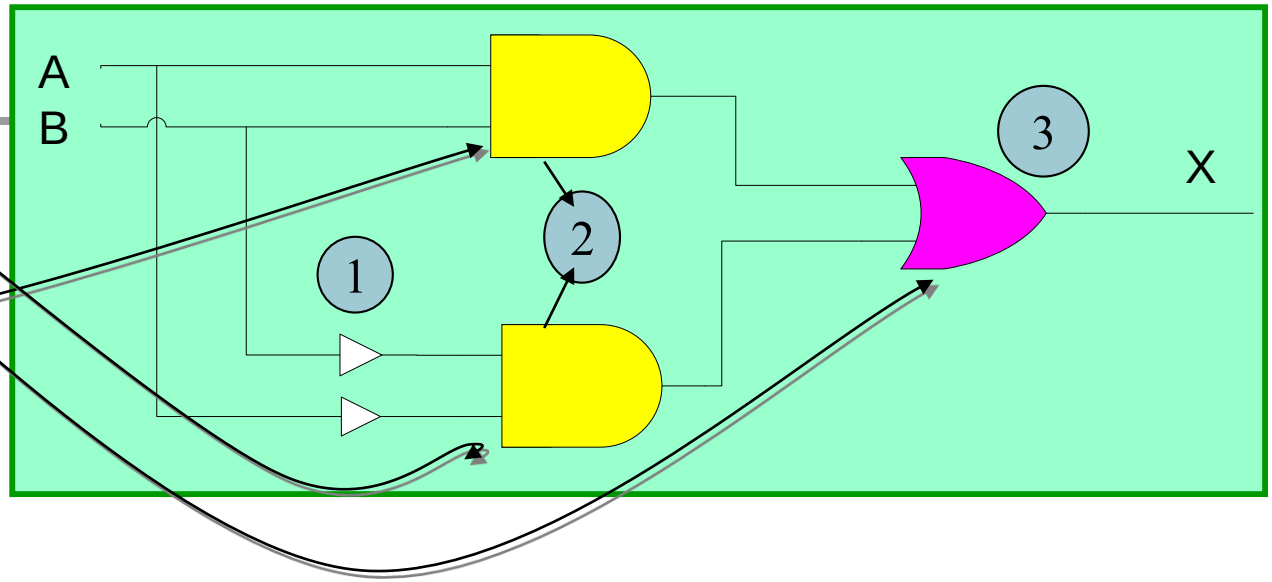
- เทอมที่คูณกันใช้แอนด์เกต
- เทอมที่บวกกันใช้ออร์เกต

- เทอมใดมีบาร์อยู่ให้กระทำก่อน ที่จะไปกระทำเทอมแอนด์ หรือออร์

- ให้กระทำเทอมที่อยู่ในวงเล็บก่อนเป็นเทอมแรก
- ลำดับต่อมาให้กระทำแอนด์เกตก่อน ออร์เกต

# การเขียนลอจิกฟังก์ชันและตารางความจริงจากสมการพีชคณิต

$X = AB + \overline{A}\,\overline{B}$

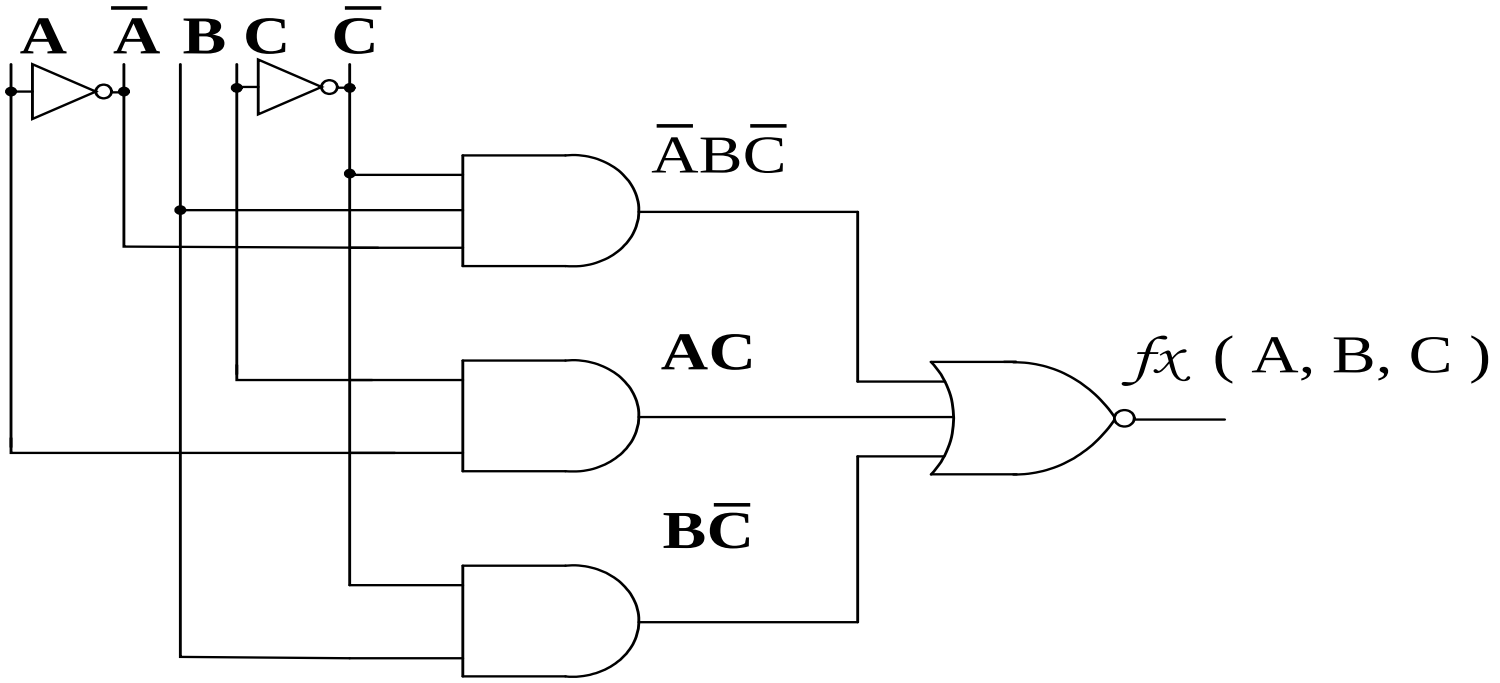



ตารางความจริง

| B | A | A.B | $\overline{A}$ | $\overline{B}$ | $\overline{A.B}$ | X |
|---|---|---|---|---|---|---|
| **0** | **0** | 0 | 1 | 1 | 1 | **1** |
| **0** | **1** | 0 | 0 | 1 | 0 | **0** |
| **1** | **0** | 0 | 1 | 0 | 0 | **0** |
| **1** | **1** | 1 | 0 | 0 | 0 | **1** |

# ตัวอย่างที่ 4.10 การเขียนลอจิกเกตจากฟังก์ชันพีชคณิต

$$fx\ (A, B, C) = \overline{\overline{A}\,\overline{B}C + AC + B\overline{C}}$$




วงจรลอจิกของ $fx\ (A, B, C) = \overline{\overline{A}\,B\,\overline{C} + AC + B\overline{C}}$

# อธิบาย

- ตัวอย่างที่ 4.10 พิจารณาเทอม AND ก่อนแบ่งออกเป็น 3 เทอม ใช้ AND 3 อินพุต 1 ตัว และ 2 อินพุต 2 ตัว NOT เกต 2 ตัวและ NOR เกต 3 อินพุต 1 ตัว
- ขั้นตอนแรกกระทำ NOT เกตที่ A , C ก่อน จากนั้นดำเนินการต่อ
- ขั้นตอนที่สองให้กระทำ AND 3 อินพุตจะถูกต่อเข้ากับ NOT A ,B, NOTC
- ขั้นตอนที่สาม กระทำ AND 2 อินพุตตัวที่ 1 จะถูกต่อกับ A, C
- ขั้นตอนที่สี่ กระทำ AND 2 อินพุตตัวที่ 2 จะถูกต่อกับ B, NOT C
- จากนั้นนำเอาต์พุตทั้งสามของ AND มาต่อกับ NOR เกต 3 อินพุต ได้ค่าเอาต์พุตที่ออกจากนอร์เกต 3อินพุต

# 421 การเขียนไทม์มิ่งไดอะแกรมจากฟังก์ชันพีชคณิต

- จากตัวอย่างที่ 4.11 ให้ฟังก์ชันมา
  - กำหนดให้ AB เป็นค่าตัวแปรที่มีค่าเท่ากับ X
  - กำหนดให้ $B\overline{C}$ เป็นค่าตัวแปรที่มีค่าเท่ากับ Y
  - กำหนดให้ $fx = X \oplus Y = X\overline{Y} + \overline{X}Y$
  - แทนค่า X และ Y ได้ดังนี้

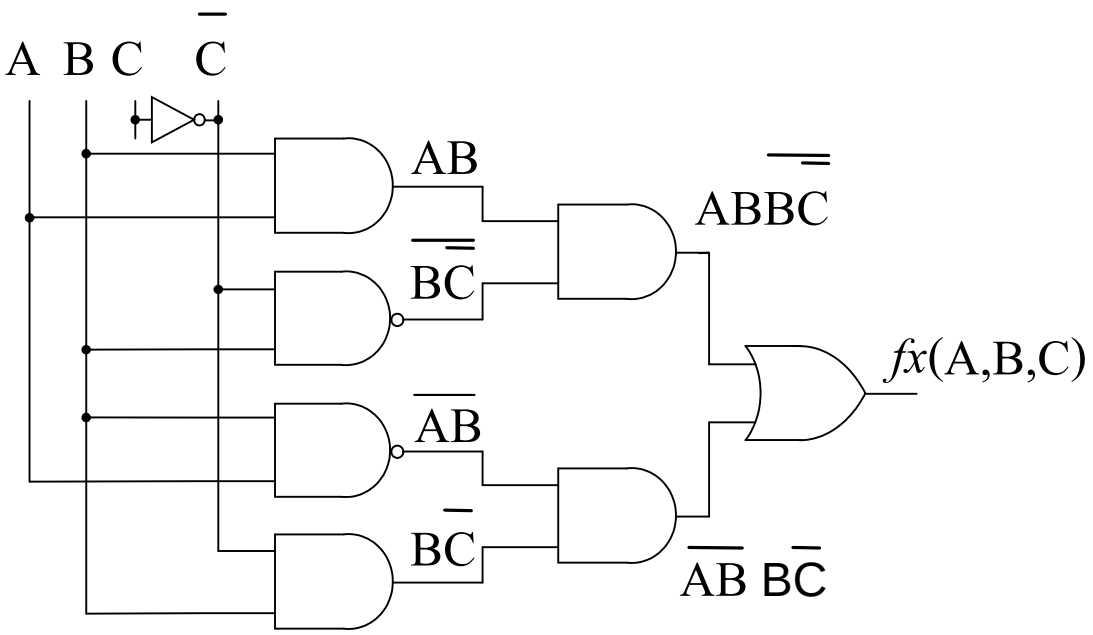
$$fx = AB \oplus B\overline{C}$$

# อธิบายเพิ่ม

- จาก A Ex-or B
- $=\bar{A}B+A\bar{B}$
- ให้ $AB=X$ ,$B\bar{C}=Y$ X Ex-orY $=\bar{X}Y+X\bar{Y}$
- $\bar{A}B\bar{B}\bar{C}+AB\bar{B}\bar{C}$

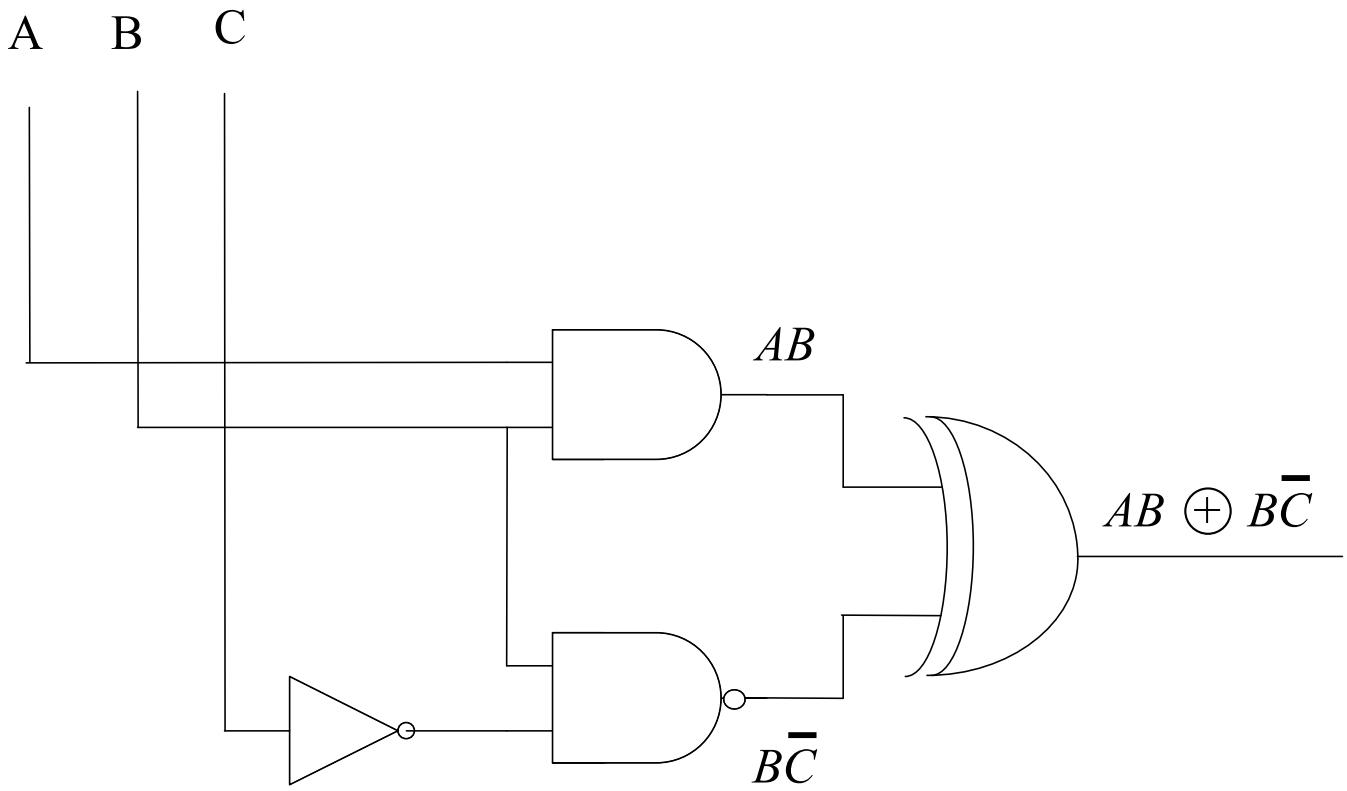
# ตัวอย่างที่ 4.11 การเขียนลอจิกเกตจากฟังก์ชัน

พีชคณิต $fx\ (A, B, C) = AB \oplus B\overline{C} = AB\overline{B\overline{C}} + \overline{AB}\,\overline{B\overline{C}}$

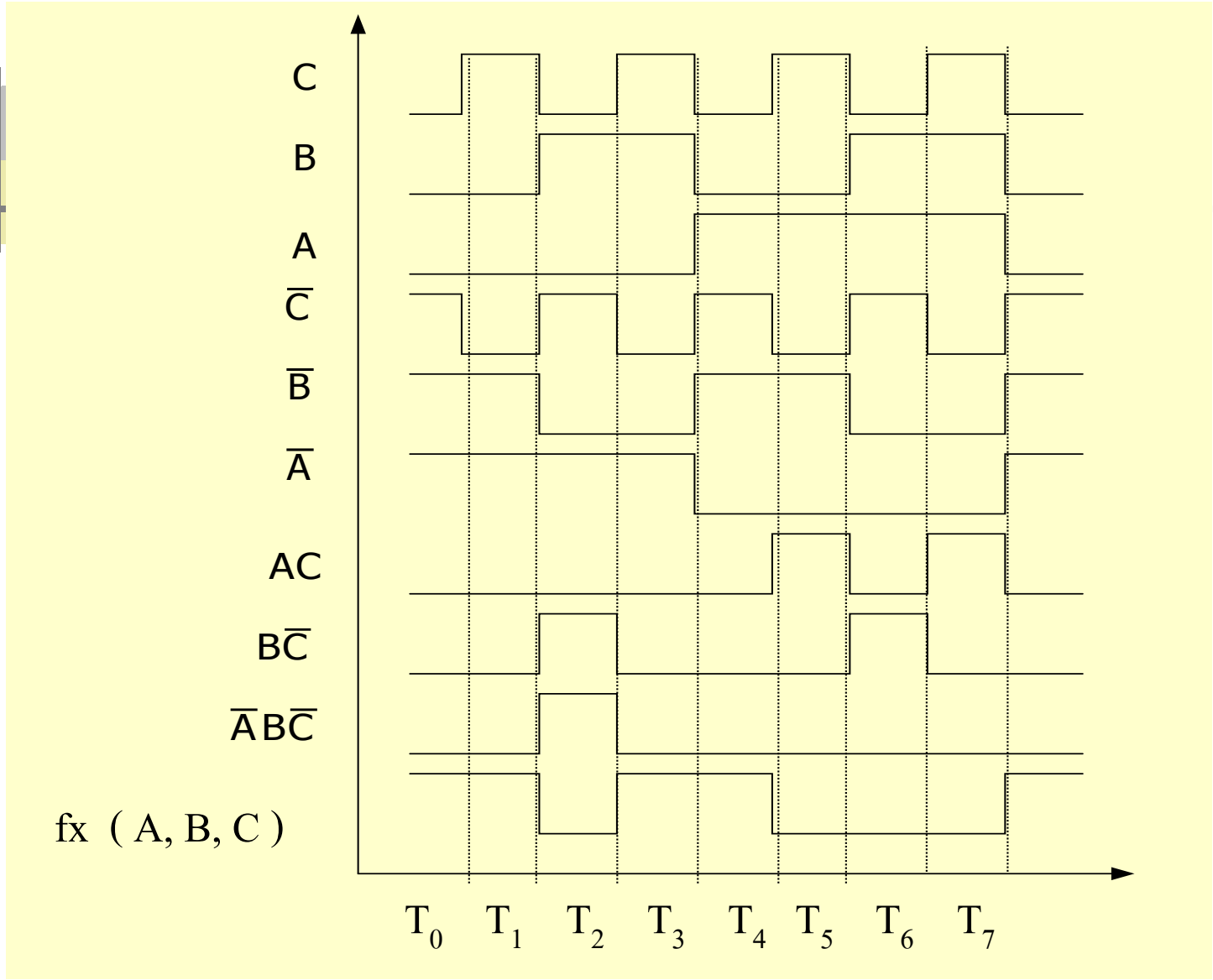



**วงจรลอจิกของ fx ( A, B, C ) = $AB \oplus B\overline{C}$**

# วงจรลอจิกเกตที่ถูกลดรูปจากตาราง K-map จากตัวอย่างที่ 4.11



# ตัวอย่างที่ 4.12 การเขียนไทม์มิ่งไดอะแกรมจากฟังก์ชันพีชคณิต



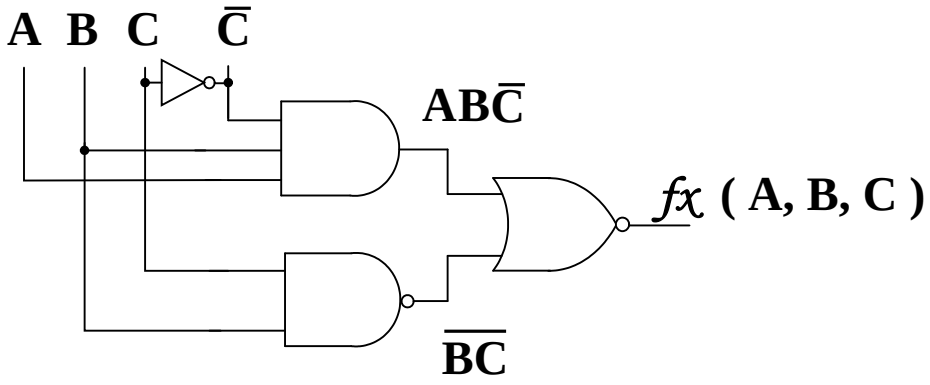

Timing Diagram ของ $fx(A, B, C) = \overline{\overline{A}B\overline{C} + AC + B\overline{C}}$

# อธิบาย

## การเขียนไทม์มิ่งไดอะแกรมจากฟังก์ชันพีชคณิต

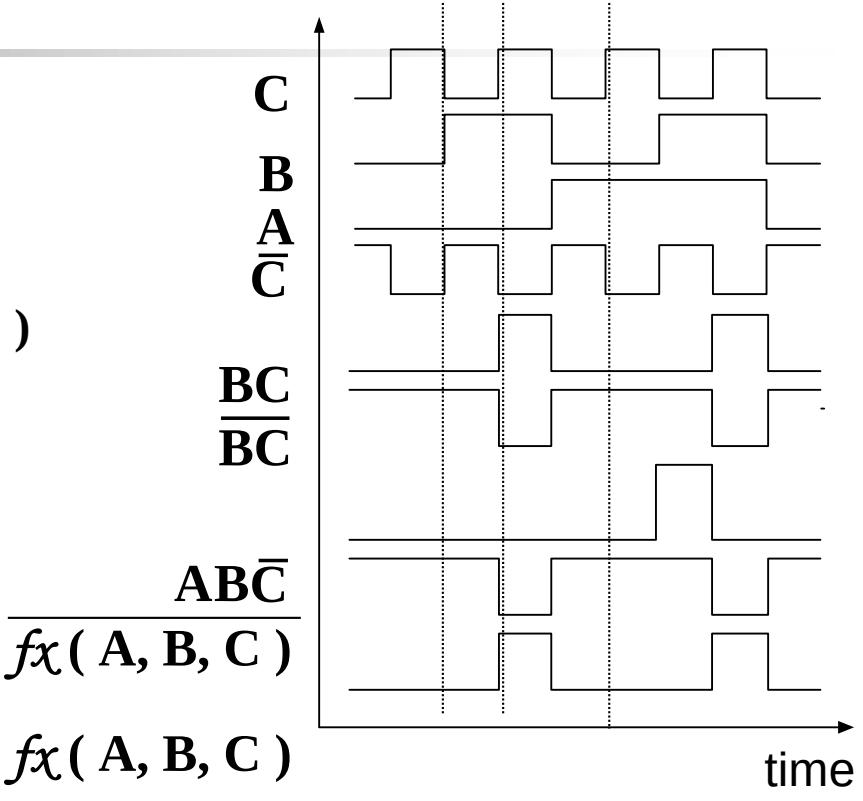
- **ขั้นตอนที่ 1** เขียนค่าไทม์มิ่งของตัวแปร A , B , C เป็นดังตารางความจริงที่ผ่านมา เลขฐานสองโดยมีหลักA เป็น$M_{SB}$ โดยพิจารณาจากค่า เวลาวินาทีที่ที่เริ่มต้นเป็น 0 จนถึงวินาทีที่ 7 ( $T_0$-$T_7$)
- **ขั้นตอนที่ 2** นำค่าไทม์มิ่งไดอะแกรม A , B , C กระทำด้วย NOT Gates ได้ไทม์มิ่งของ NOT A , B , C
- **ขั้นตอนที่3** พิจารณาที่เทอม A.C นำค่าไทม์มิ่งของA กับ C มากระทำด้วย AND โดยพิจารณาที่แต่ละช่วงเวลา ($T_0$-$T_7$) นำค่าไปเขียนเป็นไทม์มิ่งของ A AND กับ C
- **ขั้นตอนที่4** นำค่า $B.\overline{C}$ โดยพิจารณาเหมือนขั้นตอนที่3 จะได้ค่าไทม์มิ่งของ B AND กับ $\overline{C}$
- **ขั้นตอนที่5** นำค่าไทม์มิ่งของ $\overline{A}.B.\overline{C}$ มา AND กันได้เป็นดังรูป
- จากนั้นนำค่าของไทม์มิ่งจากขั้นตอน 3 ถึง 5 มากระทำแบบ NOT OR ก็จะได้ดังไทม์มิ่งสุดท้ายในรูปคือ $f(A, B, C) = \overline{\overline{A}B\overline{C} + AC + B\overline{C}}$

# ตัวอย่างที่ 4.13 ความสัมพันธ์ของฟังก์ชันพีชคณิตกับลอจิกเกตและไทม์มิ่งไดอะแกรม




วงจรลอจิก

$$fx\ (\ A, B, C\ ) = \overline{AB\overline{C}\ +\ \overline{BC}}$$



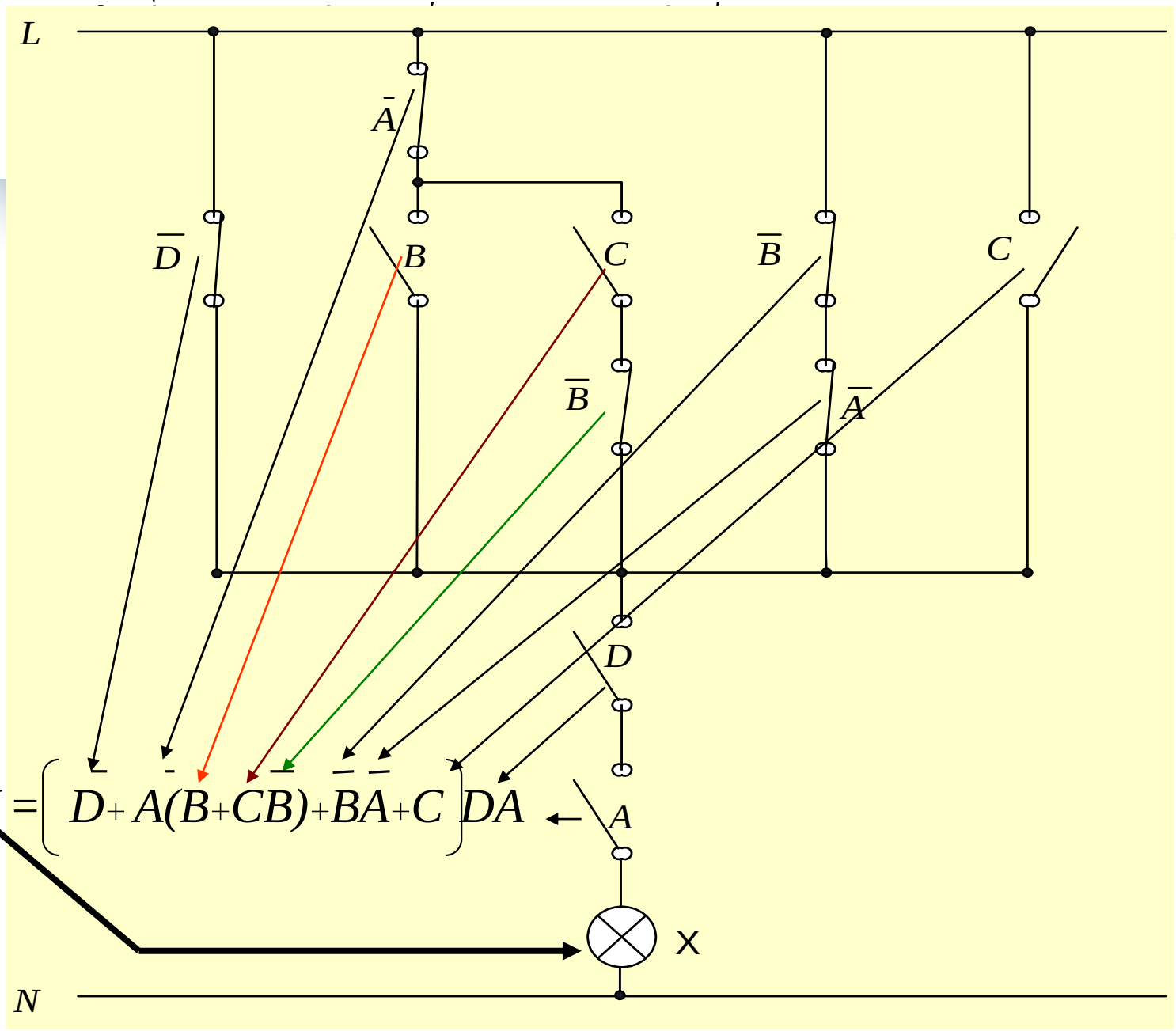

Timing Diagram ของ

$$fx\ (\ A, B, C\ ) = \overline{AB\overline{C}\ +\ \overline{BC}}$$

# อธิบาย

- จากตัวอย่างที่ 4.13 หากกำหนดวงจรเกตมาให้แล้วทำการเขียนไทมมิ่งไดอะแกรม
- ขั้นตอนแรก จากวงจรเกตให้นำมาเขียนฟังก์ชันการทำงาน โดยพิจารณาจากอินพุตไปทางเอาต์พุต ก็จะได้ $f_x(A, B, C) = \overline{AB\overline{C}} + \overline{B\overline{C}}$
- ขั้นตอนที่สองนำค่าที่ได้จากฟังก์ชันมาเขียนเป็นไทมมิ่งไดอะแกรม โดยใช้หลักวิธีที่กล่าวมาในตัวอย่าง 4.12 ,4.13

# 422การเขียนลอจิกฟังก์ชันจากสวิทซ์คอนแทคเรเล่ย์



# อธิบาย

## การเขียนลอจิกฟังก์ชันจากสวิทซ์คอนแทครีเลย์

- จากตัวอย่างที่ 4.14 การเขียนฟังก์ชันจากรีเลย์คอนแทค เพื่อให้เข้าใจขออธิบายจากวงจรที่ซับซ้อนที่ยังมีใช้ในเครื่องจักรรุ่นเก่า
- ในงานระบบไฟฟ้าอุตสาหกรรมหนักแต่เดิมนั้น ส่วนใหญ่จะใช้รีเลย์คอนแทค เป็นตัวกระทำในการควบคุมการทำงานอัตโนมัติ แทนวงจรดิจิตอล
- วงจรดิจิตอลรุ่นก่อน การใช้งานในทางอุตสาหกรรมมักจะประสบปัญหาในเรื่องสภาพแวดล้อมไม่อำนวยต่อการใช้งานเช่นความร้อน การเปลี่ยนแปลงของกระแสอย่างฉับพลัน สัญญาณรบกวนก็มีมาก ส่วนที่ทำให้เกิดการเปลี่ยนแปลงของสภาวะการทำงานของเครื่องจักรมักจะใช้รีเลย์คอนแทคแทนวงจรเกต รีเลย์คอนแทค จึงมีส่วนสำคัญของระบบควบคุมทางอุตสาหกรรมหนัก และเป็นพื้นฐานการทำงานจริงที่ใช้เริ่มแรก ต่อมาการพัฒนาในรูปแบบของวงจรอิเล็กทรอนิกส์ดิจิตอล , วงจรPLC , วงจรไมโครโปรเซสเซอร์,ไปจนถึงการควบคุมด้วยระบบคอมพิวเตอร์ ซึ่งมาจากรากฐานทางตรรกะนี้เป็นสำคัญ นักศึกษาควรเรียนรู้เมื่อประสบกับงานจริงภาคสนาม ทั้งรีเลย์คอนแทค และวงจรเกตในรูปแบบไอซี

# อธิบาย(ต่อ)

- การออกแบบหน้าคอนแทครีเลย์ดังที่แสดงในตัวอย่างที่ 4.14 มีหลักการพิจารณาดังนี้
- พิจารณาได้เหมือนกับการเขียนวงจรลอจิกเกต ได้ดังนี้
- ขั้นตอนที่ 1 พิจารณาในเทอมวงเล็บนำNOTB ไปแอนด์กับC จากนั้นนำไปออร์ กับ B
- ขั้นตอนที่ 2 นำเทอมที่ได้จากขั้นตอนที่ 1 ไปแอนด์กับ NOTA พักค่าไว้เป็นเทอมที่ 1
- ขั้นตอนที่ 3 นำเทอม NOTA ไป แอนด์กับ NOTB พักค่าเอาไว้เป็นเทอมที่2
- ขั้นตอนที่ 4 นำเทอมที่ 1,2 ,NOTD และ C ไปออร์กัน ทั้งหมดจะได้เทอม 3
- ขั้นตอนที่ 5 นำเทอมที่ 3 ไป แอนด์กับ Dและ C จะได้ค่าทางเอาต์พุต

## ตัวอย่างที่ 4.15 นำวงจรสวิทช์ชิงมา เขียนสมการพีชคณิต และลดรูปโดยทฤษฎีของบูลีน

$$X = [\ \overline{A}\ (B + C\overline{B}) + D + \overline{B}\,\overline{A} + C\ ]\ DA$$

$$= [\ \overline{A}B + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B} + C + \overline{D}\ ]\ DA$$

1

$$= [\ \overline{A}B + (1 + C)\ \overline{A}\,\overline{B} + C + \overline{D}\ ]\ DA$$

1

$$= [\ \overline{A}\ (B + \overline{B}) + C + \overline{D}\ ]\ DA$$

$$= [\ \overline{A} + C + \overline{D}\ ]\ DA$$

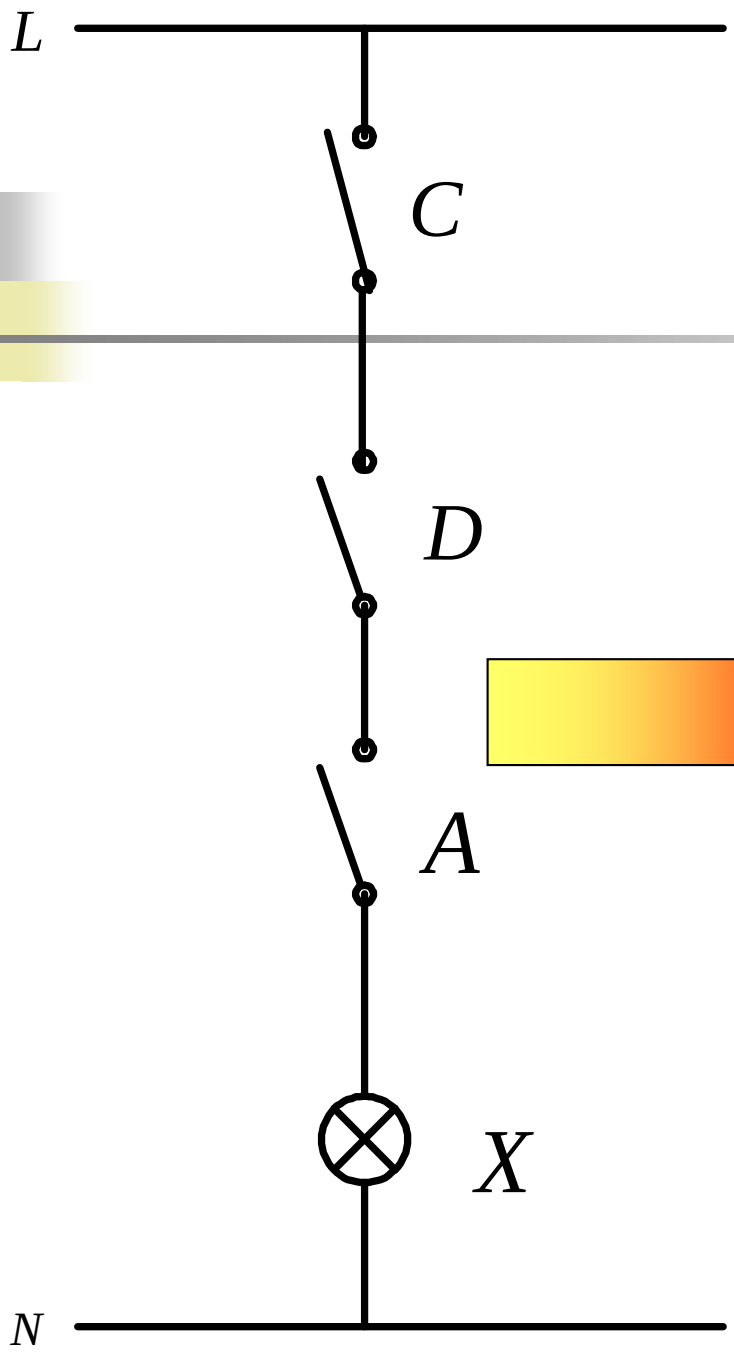
0 0

$$= \overline{A}DA + CDA + \overline{D}DA$$

$$= CDA$$

# อธิบาย

- จากตัวอย่างที่ 4.14 หาค่าได้เป็นตัวอย่างที่ 4.15 สามารถลดค่าสมการที่ยุ่งยากเพื่อให้ง่ายต่อการวิเคราะห์ สามารถลดค่าได้ดังรูปตัวอย่างที่ 4.15 จนเหลือค่าเพียงสมการง่ายๆ เพียงแค่รีเลย์คอนแทค A,D นำไปกระทำกับ C ในฟังก์ชันของ AND ดังรูปสไลด์ถัดไป

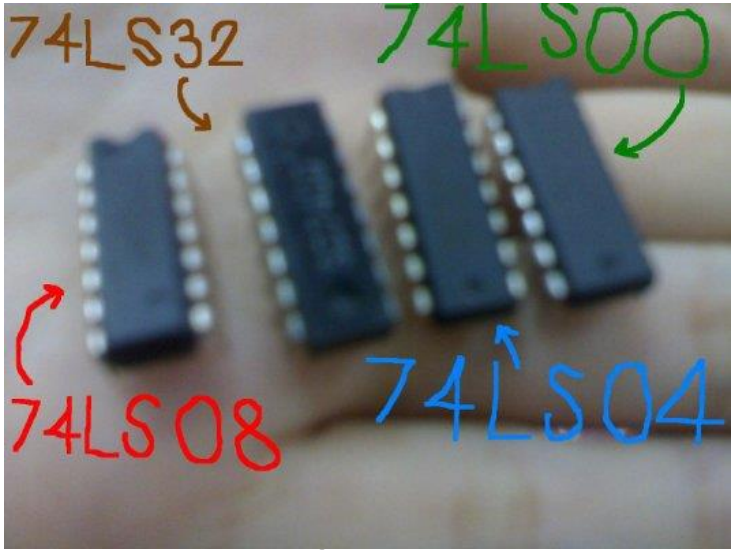


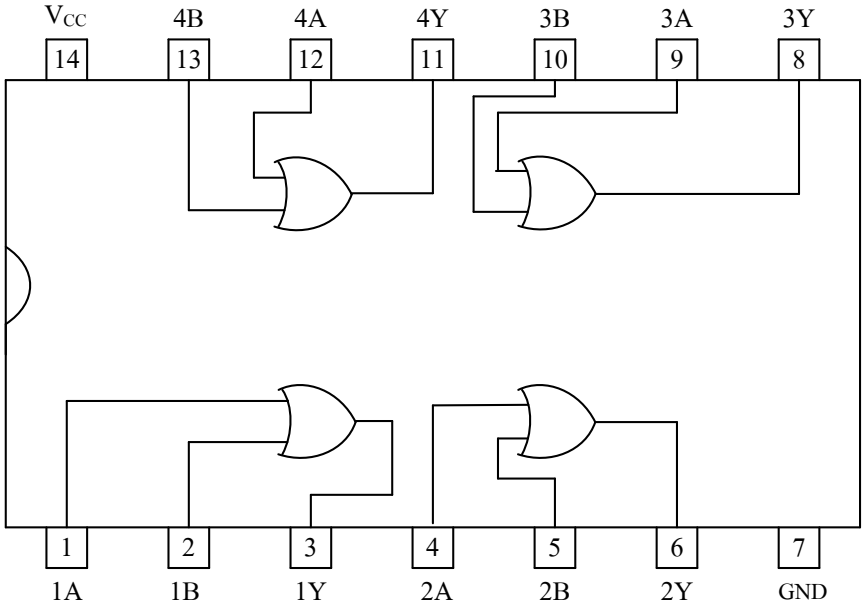

**เขียนวงจรง่าย ๆ**

$$X = CDA$$

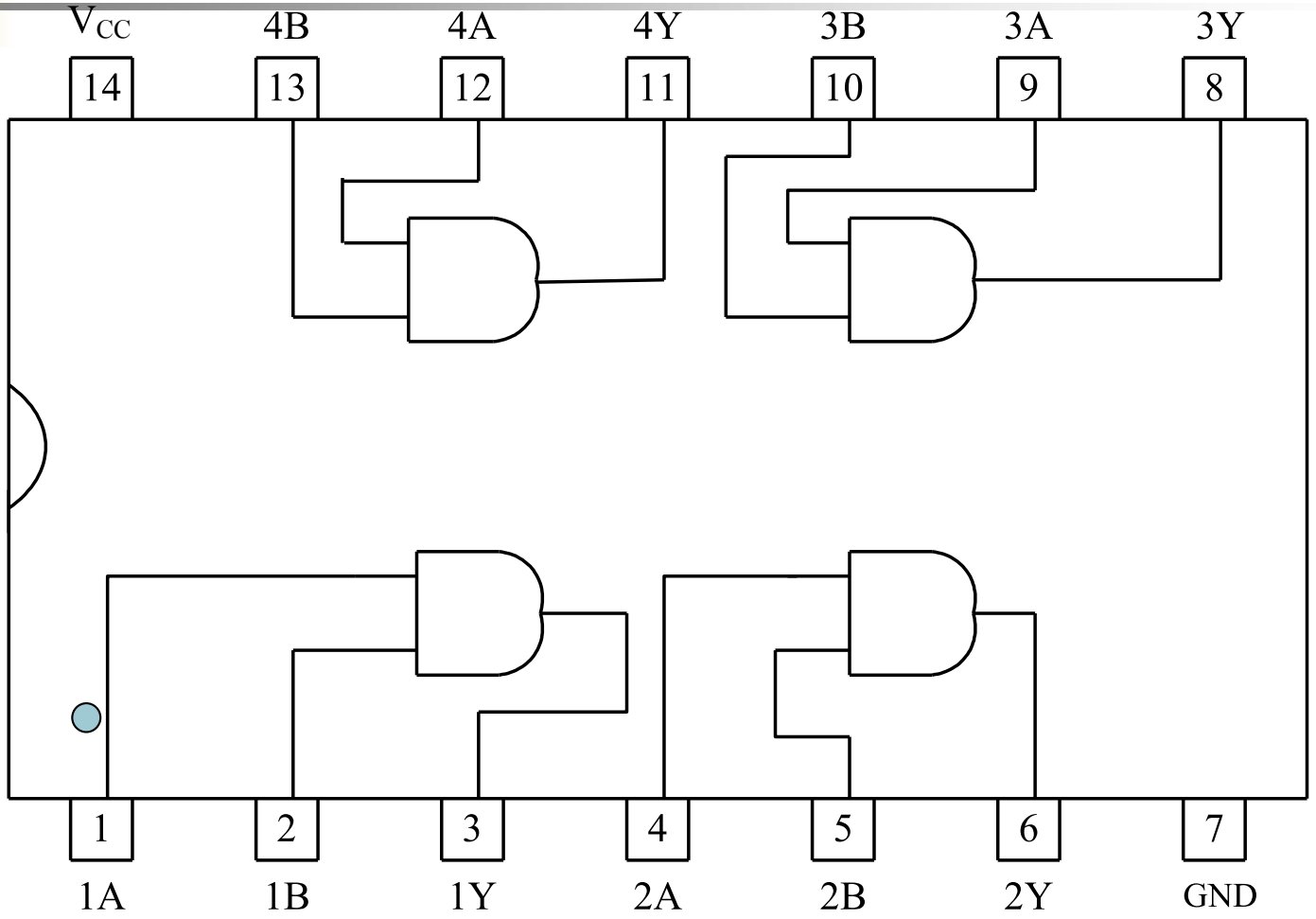
# ตัวอย่างการนำวงจรลอจิกที่อยู่ในรูปไอซีไปใช้งาน

## OR Gate 2อินพุต #74LS32

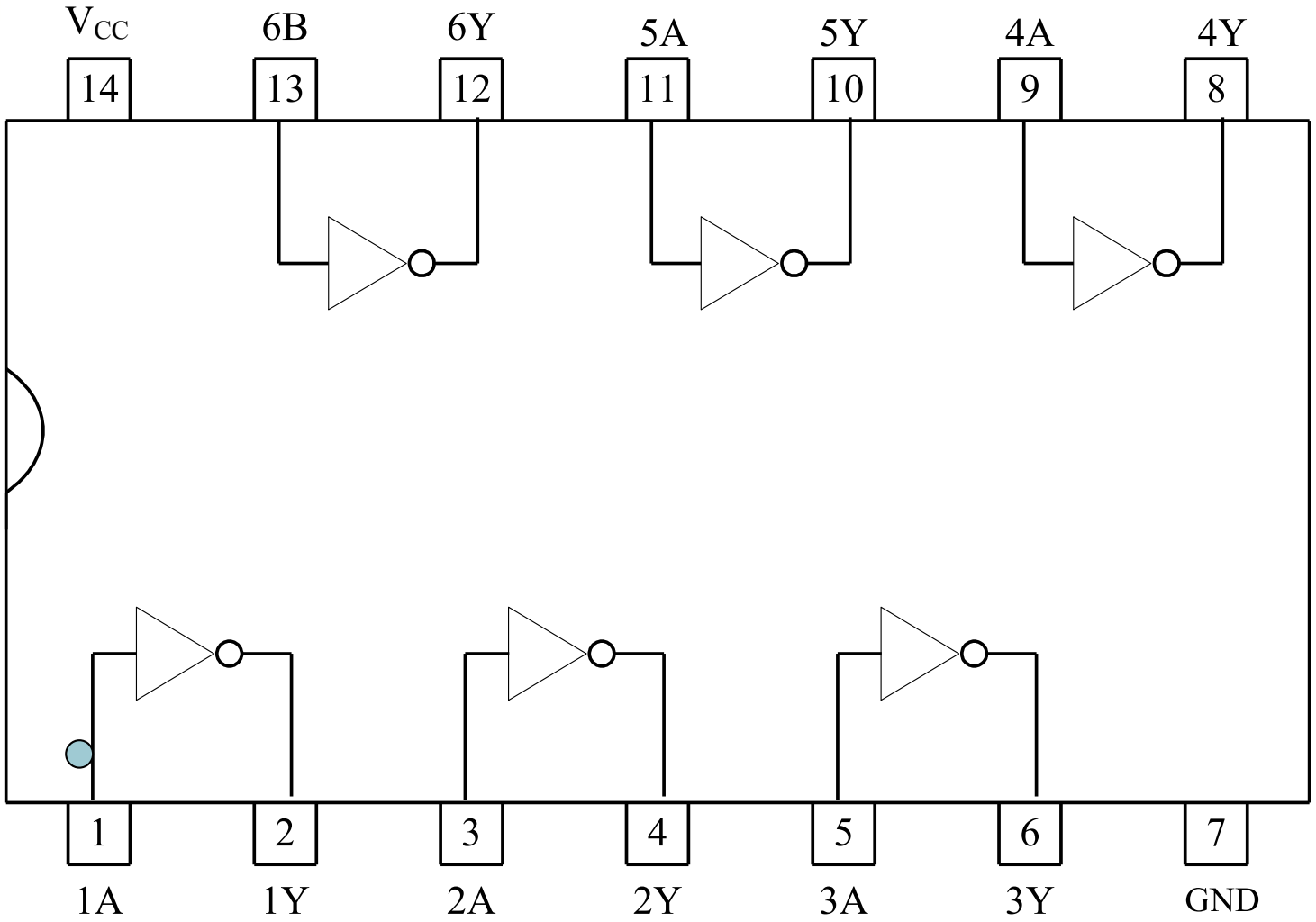


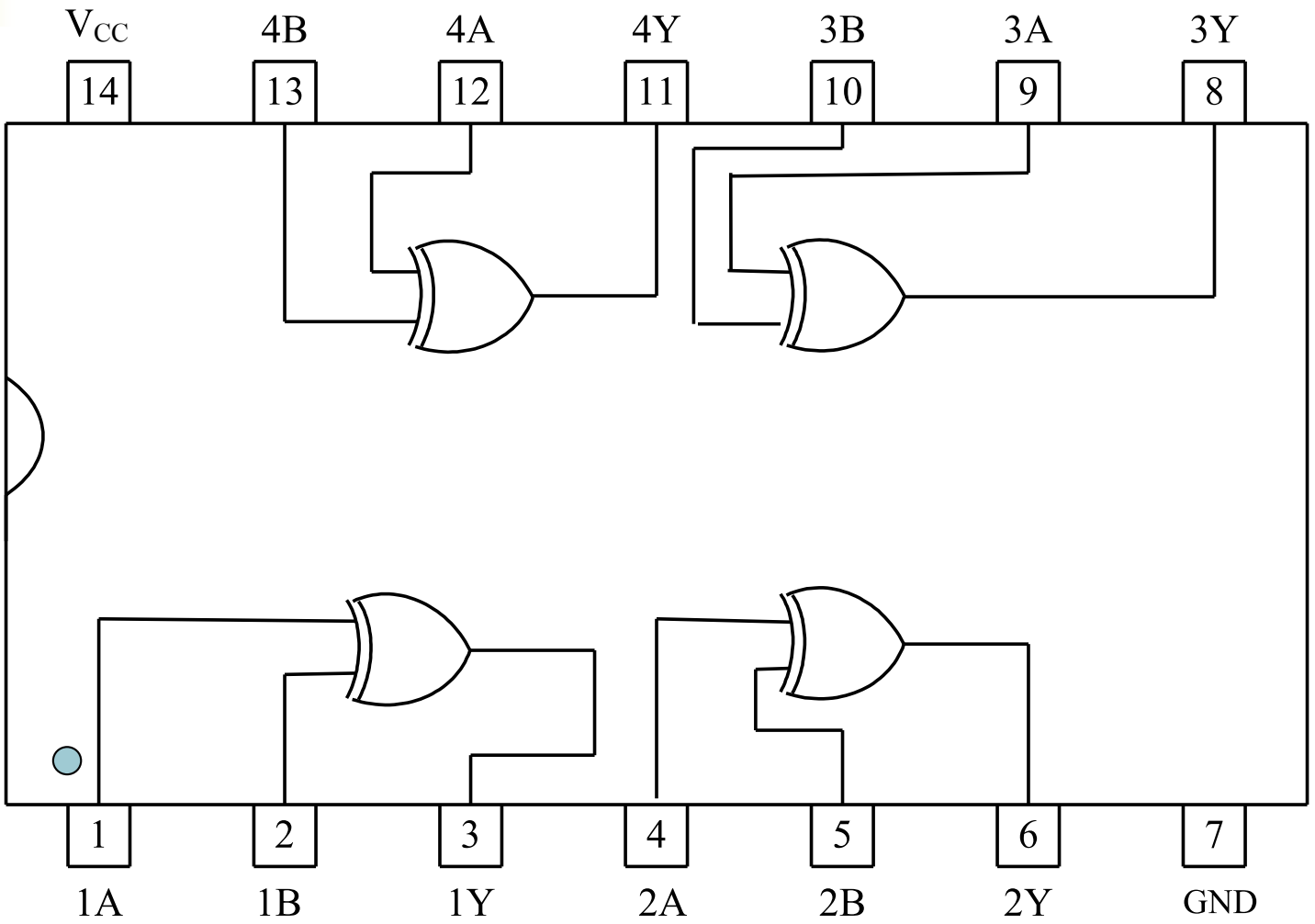
# AND Gate 2 อินพุต # 74LS08



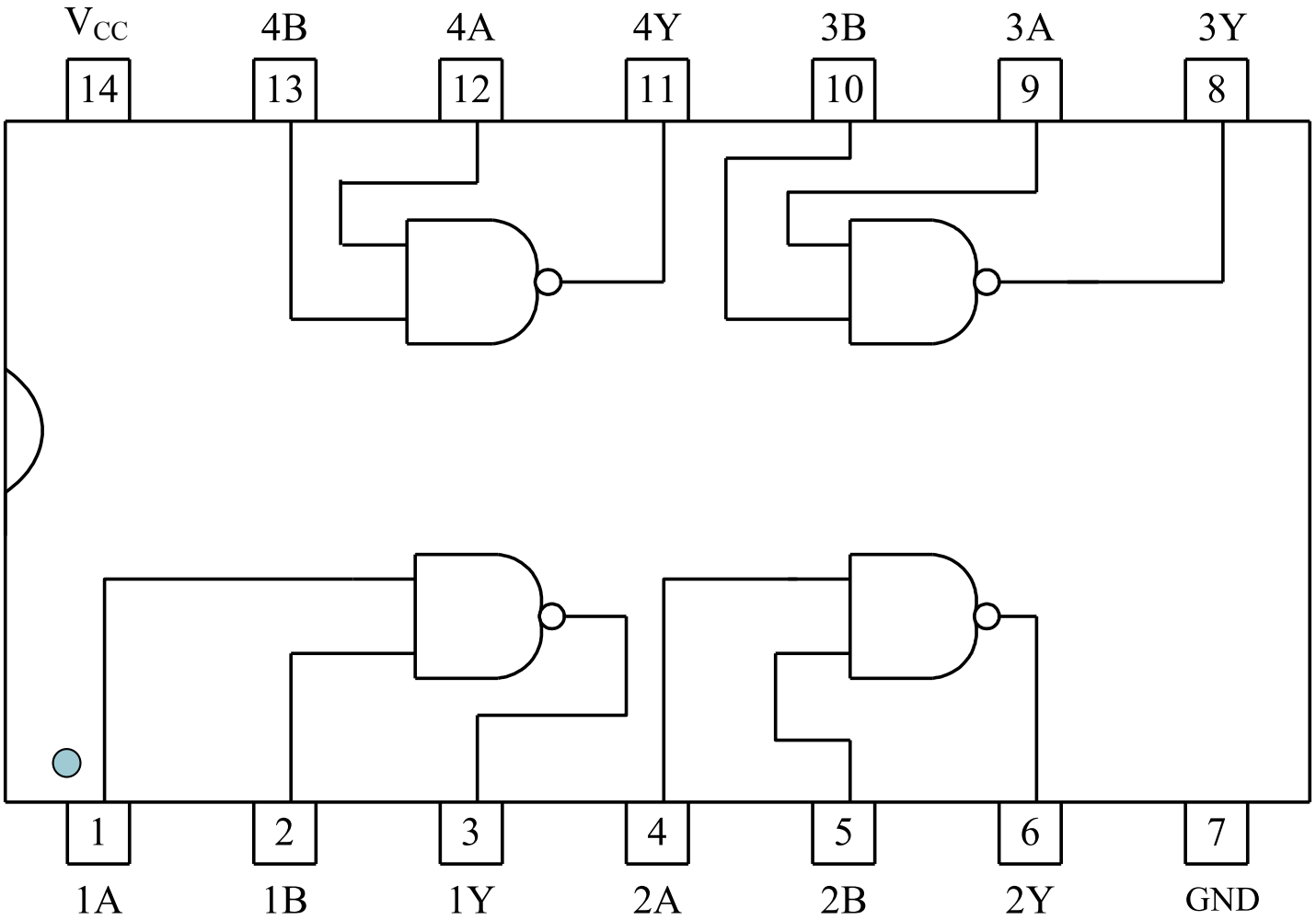
# Inverter Gate # 74LS04



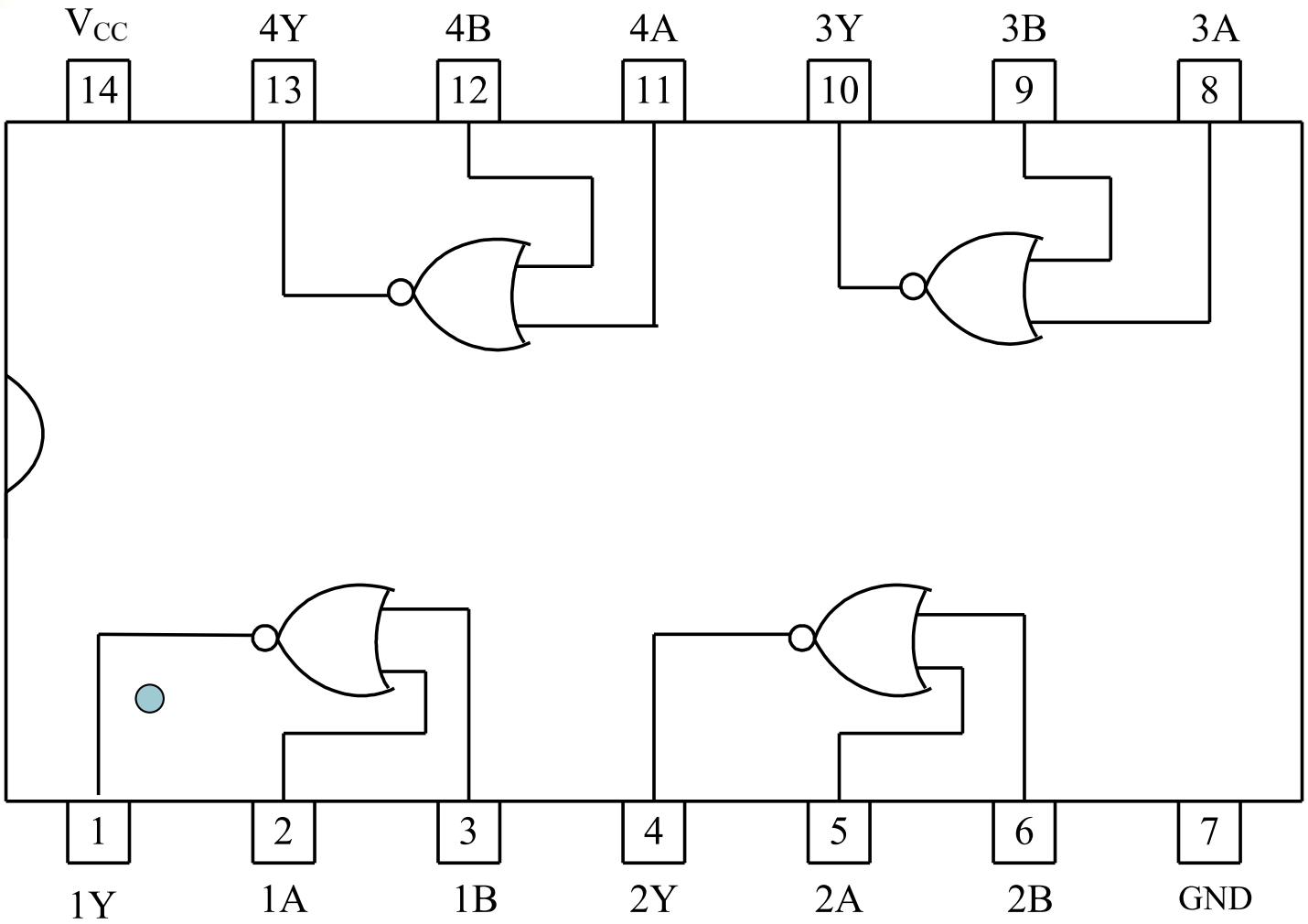
# EX-OR Gate 2 อินพุต #74LS86



# NAND Gate 2อินพุต #74LS00



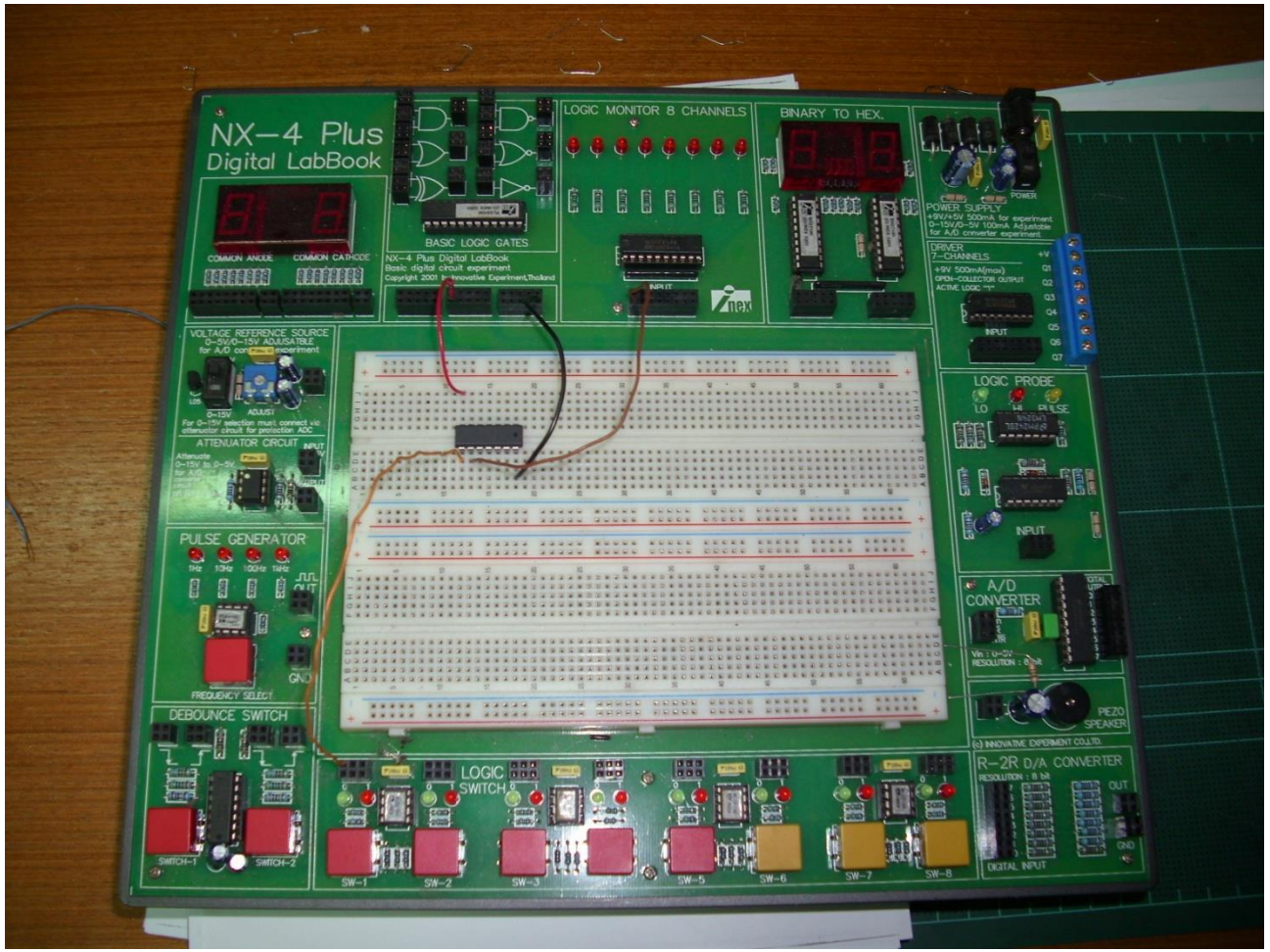
# NOR Gate 2อินพุต #74LS02



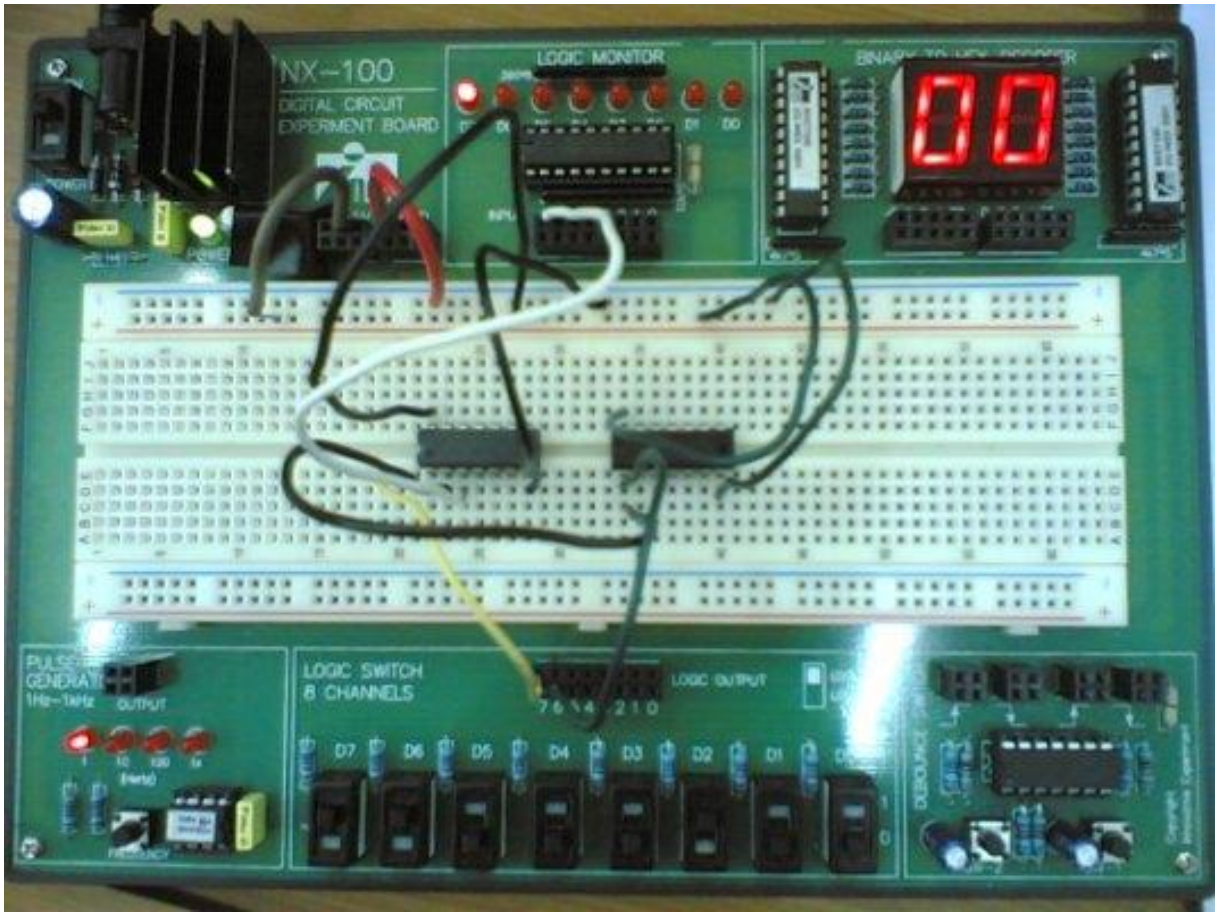
# อธิบาย

- พื้นฐานของไอซีเกตที่มีจำหน่ายในท้องตลาดเป็นไอซีเบอร์ 74และต่อท้ายด้วยตัวเลข ส่วนอักษรแสดงค่าเป็นความเร็ว
- การนับขาจากด้านบนดูที่จุดเป็นขา1,2……. แล้ววนทวนเข็มนาฬิกา
- มาตรฐานของไอซีตระกูล 74LS XX จะเป็น 14 ขา และ 16 ขาเป็นไอซีทำงานตามฟังก์ชัน เช่น Decorder /Demultiplexer เบอร์ 74LS139
- ขา 7 , 8 ต่อลงกราวนด์ หรือ ลอจิก0
- ขา 14 ,16 ต่อกับแหล่งจ่ายไฟบวกขนาด 4.75 Volts ถึง 5.25 โวลต์
- อินพุตระดับ1 ต่อที่ 2.7-5 Volts ระดับ0 ต่อกับ 0-0.8 Volt
- การหารานละเอียดนักศึกษาสามารถดูจากคู่มือ(Data sheet)ของบริษัทที่ผลิตมีอยู่ในห้องสมุด
- ข้อมูลทั่วไปนักศึกษาสามารถหาได้จากเครือข่ายคอมพิวเตอร์โดยใช้ Search engine ที่ทราบกันดี

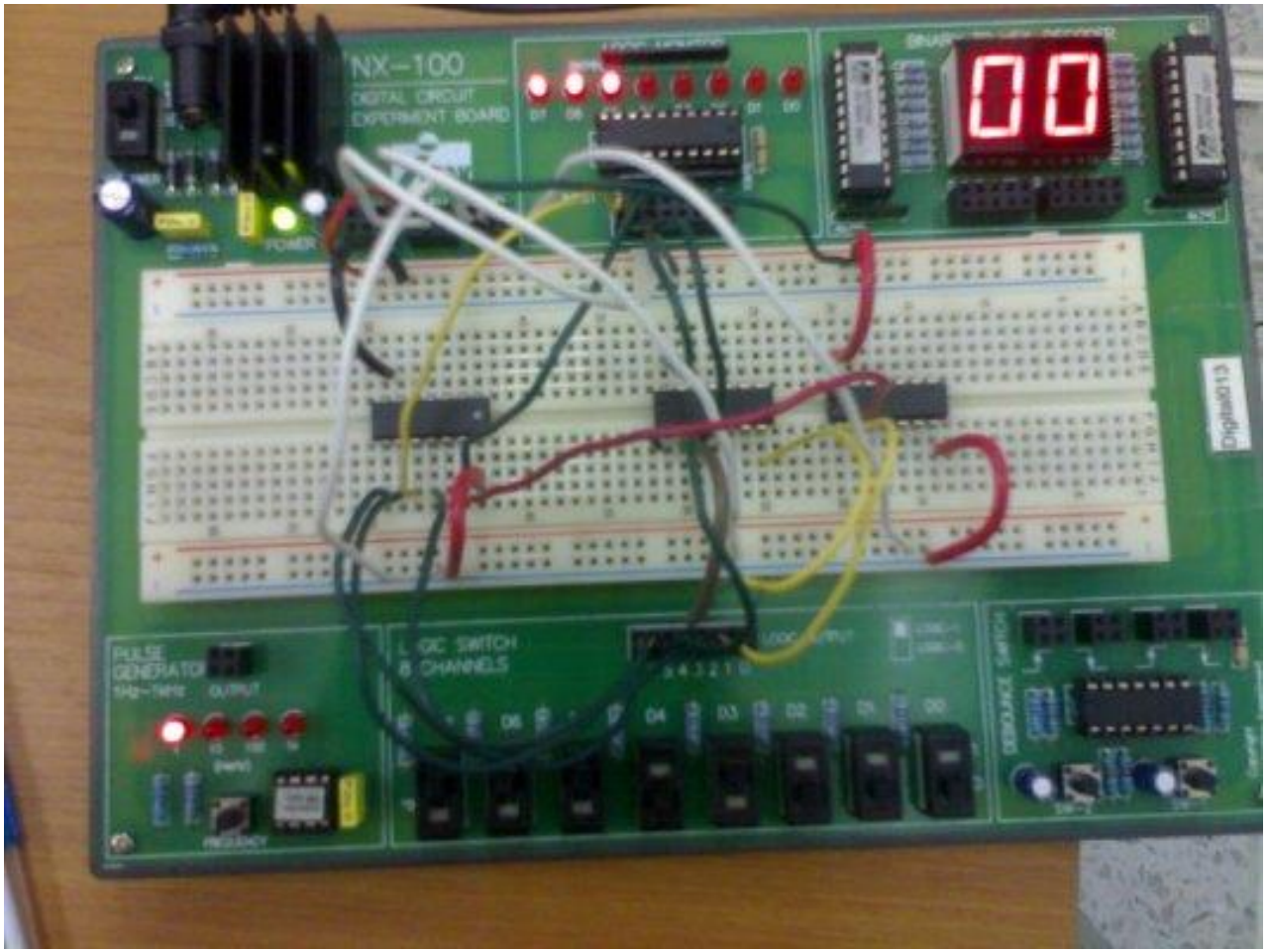
# ตัวอย่างการนำวงจรดิจิตอลที่ออกแบบไปต่อใช้งานจริง



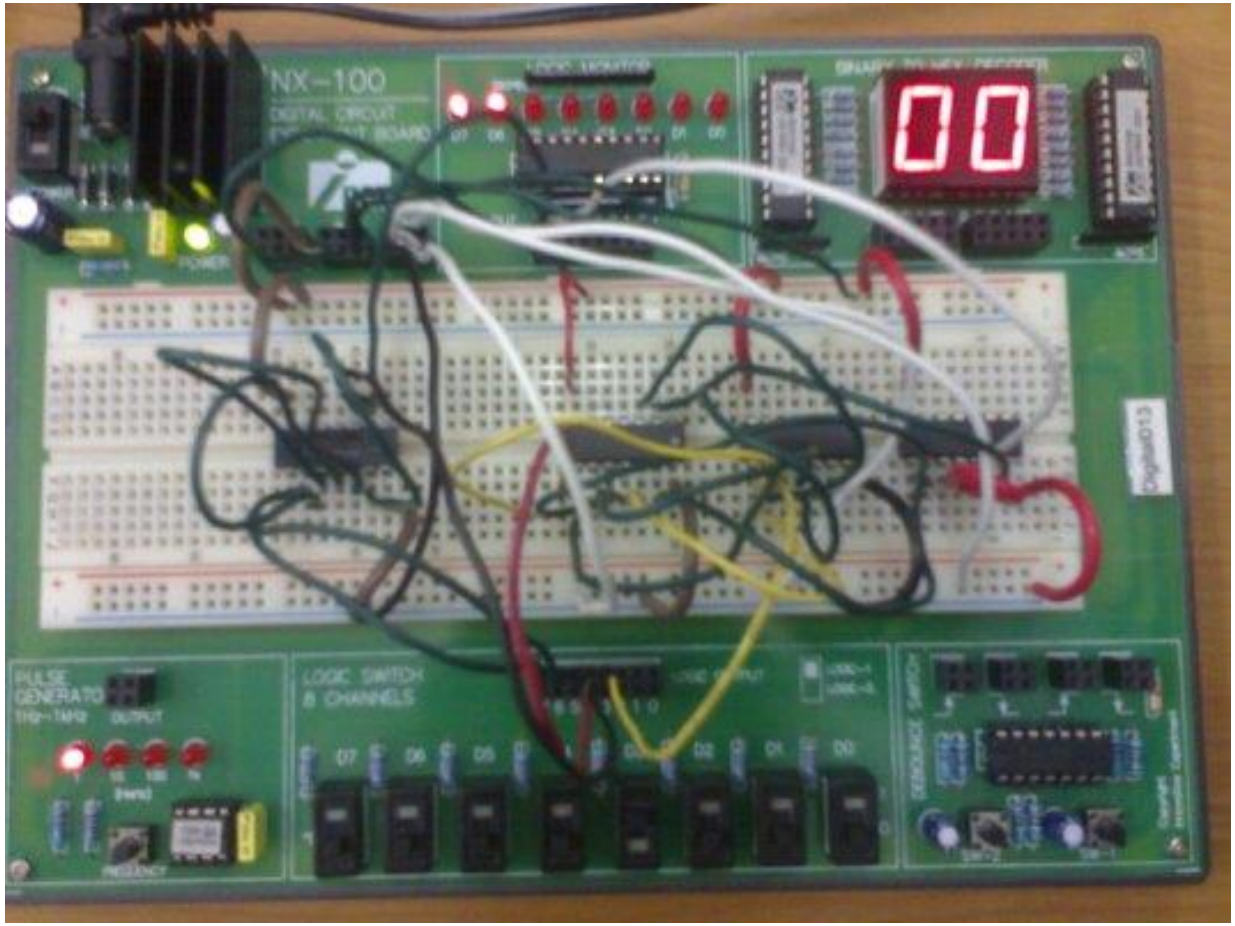
# การนำAND Gate และ OR Gate มาต่อใช้งานจริง



# ตัวอย่างการนำไปใช้งานของ AND,OR, Inverter Gates มาต่อเข้าด้วยกัน



# การนำวงจรเกตที่กล่าวมาไปใช้งาน



# บทสรุป

- สามารถเข้าใจการทำงานของวงจรเกตต่างๆ ได้
- เข้าใจถึงการออกแบบวงจรดิจิตอลได้
- สามารถวิเคราะห์ไทมมิ่งไดอะแกรมจากวงจรดิจิตอลได้
- สามารถเขียนวงจรเกตจากไทมมิ่งไดอะแกรมได้
- สามารถวิเคราะห์ไดอะแกรมการทำงานของรีเลย์ในรูปแบบการทำงานของวงจรดิจิตอล

# ทดสอบความรู้ในสัปดาห์ที่ 4

1. กำหนด Function ให้ทำการเขียนตารางความจริง

a) $f_1(A,B,C,D) = AC + BD + A\bar{B}\bar{D}$

b) $f_2(A,B,C,D) = A\bar{B}\bar{D} + AB + \bar{A}B\bar{C}$

c) $f_3(A,B,C,D) = BD + A\bar{B}\bar{D} + ACD + ABC$

d) $f_4(A,B,C,D) = AC + A\bar{B}\bar{C}\bar{D} + \bar{A}BD + B\bar{C}D$

e) $f_5(A,B,C,D) = (B + \bar{D})(A + B)(A + \bar{C})$

2. ใช้ K-map เพื่อทำให้เป็น Function อย่างง่าย

a) $f(A,B,C) = \sum m(1,5,6,7)$

b) $f(A,B,C) = \sum m(0,1,2,3,4,5)$

3. ใช้ K-map เพื่อทำให้เป็น Function อย่างง่ายพร้อมเขียนไทม์มิ่งการทำงานของฟังก์ชัน

a) $f(A,B,C,D) = \sum m(0,2,5,7,8,10,13,15)$

b) $f(A,B,C,D) = \sum m(1,3,4,5,6,7,9,11,12,13,14,15)$

4. ใช้ K-map เพื่อหา Function อย่างง่ายพร้อมทั้งเขียนเป็นลอจิกเกต และลอจิกคอนแทค

a) $f(A,B,C,D) = \sum m(0,4,5,7,8,10,11,15)$

b) $f(A,B,C,D) = \sum m(1,4,5,6,7,9,11,15)$

5. จงหา Function โดยใช้ POS (Product of sum)

a) $f(A,B,C) = \pi M(0,2,3,4)$

b) $f(A,B,C) = \pi M(0,3,4,7)$

c) $f(A,B,C) = \pi M(0,1,4,5,6)$

d) $f(A,B,C) = \pi M(1,2,3,6)$

e) $f(A,B,C) = \pi M(1,2,5,7)$

f) $f(A,B,C) = \pi M(1,2,3,4)$

# ทดสอบความรู้ในสัปดาห์ที่ 4

1. กำหนด Function ให้ทำการเขียนตารางความจริง

a) $f_1(A,B,C,D) = AC + BD + A\overline{B}\,\overline{D}$

b) $f_3(A,B,C,D) = BD + A\overline{B}\overline{D} + ACD + ABC$

c) $f_5(A,B,C,D) = (B+\overline{D})(A+B)(A+\overline{C})$

2. ใช้ K-map เพื่อทำให้เป็น Function อย่างง่ายพร้อมเขียนไทม์มิ่งการทำงานของฟังก์ชัน ลอจิกเกต และลอจิกคอนแทค

a) $f(A,B,C) = \sum m(1,5,6,7)$

b) $f(A,B,C) = \sum m(0,1,2,3,4,5)$

3. จงหา Function โดยใช้ POS (Product of sum)

a) $f(A,B,C) = \pi M(0,2,3,4)$

b) $f(A,B,C) = \pi M(1,2,3,6)$

c) $f(A,B,C) = \pi M(0,1,4,5,6)$

# เนื้อหาคำบรรยายในสัปดาห์ที่ 5 การแปลงรหัส (Encoder)

## 43 รหัสข้อมูล

431 ชนิดและลักษณะของรหัสแบบต่างๆ

432 รูปแบบรหัสแทนข้อมูล

433 ตารางเลขฐาน, รหัส

434 การออกแบบวงจรเข้ารหัส

435 การออกแบบวงจรเข้ารหัส BCD-8421

436 การออกแบบวงจรสร้างบิตตรวจสอบ

437 รหัสที่ใช้ในเครื่องคอมพิวเตอร์

# 43 รหัสข้อมูล

รหัสแทนข้อมูล คือ รหัสที่ใช้แทนข้อมูล เพื่อส่งเข้าไปบันทึกไว้ในเครื่องที่มีการทำงานในระบบดิจิตอล และให้เครื่องประมวลผล CPU ในรูปแบบที่ไม่ซ้ำกัน และสามารถแยกแยะรายละเอียดของข้อมูลได้ รหัสถูกนำไปสร้างสัญญาณทางดิจิตอลที่หลากหลาย ซึ่งจะวิเคราะห์ถึงการสร้างและการถอดรหัส และการนำไปใช้งาน

# 431 ชนิดและลักษณะของรหัสแทนข้อมูลแบบต่าง ๆ

**รหัสที่ใช้แทนรหัสข้อมูลภายนอก**

– รหัสแทนข้อมูลต่าง ๆ เช่น ตัวเลข, ตัวอักษร, อักขระพิเศษ สำหรับบันทึกลงบนสื่อกลางต่าง ๆ เช่น บัตรเจาะรู, เทป, แผ่นจานแม่เหล็ก

**รหัสที่ใช้แทนรหัสข้อมูลภายใน**

– รหัสแทนข้อมูลต่าง ๆ เช่น ตัวเลข, ตัวอักษร, อักขระพิเศษ สำหรับบันทึกลงบนหน่วยความจำต่าง ๆ

# 432 รูปแบบรหัสแทนข้อมูล

## รหัสแทนข้อมูลมี 3 ประเภท

ประเภทที่ 1. BCD (BINARY CODED DECIMAL)
เป็นรหัสแบบ 6 บิต เป็น 1 ไบต์ แบ่งเป็น 2 ส่วน

-***Zone Bit*** คือ 2 บิตแรก เป็นตัวชี้รหัส

-กลุ่ม, ตัวอักษร, ตัวเลข, อักขระพิเศษ

-***Digit Bit*** คือ4 บิตหลังเป็นเลขฐาน2 จำนวน4ตัว

- ค่าเปลี่ยนตามตำแหน่งตัวอักษร

# BCD แบ่งข้อมูลออกเป็น 3 กลุ่มคือ

**กลุ่มที่ 1** ข้อมูลที่เป็นตัวเลข (0-9) - Zone Bit เป็น 00 เช่น 6 ฐาน10 เป็น BCD 000110

**กลุ่มที่ 2** ข้อมูลที่เป็นตัวอักษร (A-Z) - แบ่งเป็นกลุ่มละ 9 ตัว

- A-I Zone Bit เป็น 11
- J-R Zone Bit เป็น 10
- S-Z Zone Bit เป็น 01

**กลุ่มที่ 3** ข้อมูลที่เป็นอักขระพิเศษ เช่น *,+,-,/,$ เป็นต้น

# ประเภทที่ 2 **EBCDIC** (EXTENTED BINARY CODED DECIMAL INTERCHANGE CODE)

เป็นรหัส 8 บิต เป็น 1 ไบต์ หรือเท่ากับ 2 กำลัง 8 = 256 ตัวอักษร แบ่งเป็น 2 กลุ่ม ๆ ละ 4 บิต เป็นเลขฐาน 2 หรือเลขฐาน 16

**EBCDIC แบ่งข้อมูลเป็น 3 ลักษณะคือ**

ลักษณะที่ 1 ข้อมูลที่เป็นตัวเลข (0-9) - Zone Bit เป็น 1111 หรือ F ในฐาน 16

ลักษณะที่ 2 ข้อมูลที่เป็นตัวอักษร (A-Z) - แบ่งเป็นกลุ่มละ 9 ตัว

- A-I Zone Bit เป็น 1100 = C
- J-R Zone Bit เป็น 1101 = D
- S-Z Zone Bit เป็น 1110 = E

**กลุ่มที่ 3** ข้อมูลที่เป็นอักขระพิเศษ เช่น *,+,-,/,$ เป็นต้น

เช่น COM23

| | | | | | |
|---|---|---|---|---|---|
| **เลขฐาน 2** | 11000011 | 11010110 | 11010100 | 11110010 | 11110011 |
| **เลขฐาน 16** | C3 | D6 | D4 | F2 | F3 |

## ประเภทที่ 3 ASCII (AMERICAN STANDARD CODE FOR INFORMATION)

เป็นรหัส 7 บิต แบ่งเป็นข้อมูลเป็น 3 กลุ่มคือ

**กลุ่มที่ 1** ข้อมูลที่เป็นตัวเลข (0-9) - Zone Bit เป็น 011

**กลุ่มที่ 2** ข้อมูลที่เป็นตัวอักษร (A-Z)

- A-O Zone Bit เป็น 100
- P- Z Zone Bit เป็น 101

**กลุ่มที่ 3** ข้อมูลที่เป็นอักขระพิเศษ เช่น *, -, +, /, $ เป็น 010 เป็นต้น

# ตารางการแปลงเลข ฐานสิบให้อยู่ในรหัสBCD - 8421

| Decimal | BCD – 8421 |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

# อธิบาย

- ตารางการแปลงเลขฐานสิบให้อยู่ในรูป BCD-8421
  - ตัวเลขทางด้านซ้ายถูกกำหนดเป็นเลขฐานสิบ
  - ตัวเลขทางด้านขวามือถูกกำหนดให้เป็นรหัส BCD-8421
- ข้อสังเกต
  - รหัสBCD-8421 จะแสดงค่าเพียง 4 บิต เท่านั้น คือค่า 0000 -1001
  - ถ้าเกินจากเลข 9 ของฐานสิบให้ทำการทดไปอีก 4บิต ดังตัวอย่างถัดไป

# การแปลงเลขฐานสิบเป็นรหัส BCD-8421

## ตัวอย่างCode BCD-8421

| Decimal | BCD - 8421 |
|---|---|
| 5 | 0101 |
| 42 | 0100 0010 |
| 307 | 0011 0000 0111 |
| 1568 | 0001 0101 0110 1000 |
| 10249 | 0001 0000 0010 0100 1001 |

# อธิบาย

- ตัวอย่างการแปลงเลขฐานสิบเป็น BCD
- หลักหน่วยของฐานสิบมีเพียงรหัส BCD 4 บิต เช่นเลข 5
- หลักสิบของเลขฐานสิบทำการทดไปทางซ้าย 4บิต เช่นเลข 42
- หลักร้อยทำการทดไปทางซ้ายอีก 4 บิต เช่นเลข 307 ใช้ 12 บิตของ BCD-8421
- หลักพันทำการทดไปทางซ้ายอีก 4 บิต เป็นค่าเลข 16 บิต
- หลักหมื่น, หลักแสนก็จะทำการทดไปอย่างนี้ แสดงได้ดังตัวอย่าง

# 433 ตารางเลขฐานและ รหัส

| Decimal | Binary | Gray code |
|---|---|---|
| 0 | 0000 | 0000 |
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |
| 3 | 0011 | 0010 |
| 4 | 0100 | 0110 |
| 5 | 0101 | 0111 |
| 6 | 0110 | 0101 |
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |
| 9 | 1001 | 1101 |
| 10 | 1010 | 1111 |
| 11 | 1011 | 1110 |
| 12 | 1100 | 1010 |
| 13 | 1101 | 1011 |
| 14 | 1110 | 1001 |
| 15 | 1111 | 1000 |

Decimal / Binary / Gray

# อธิบาย

- ตารางเปรียบเทียบฐานสิบเป็นรหัสฐานสอง และรหัสเกร์ย (Gray Code)
- ความเหมือนกันของรหัสเกร์ย กับรหัสฐานสอง คือจำนวนบิตไม่ถูกบังคับ จะเป็นจำนวนเท่าใดขึ้นอยู่กับเลขฐานสิบซึ่งจะแตกต่างจากรหัส BCD-8421
- รหัสเกร์ยจะถูกบังคับมาจากรหัสไบนารี่ ซึ่งจะกล่าวต่อไป
- การแปลงฐานใดๆ มาเป็นรหัสเกร์ยจะต้องทำเป็นรหัสไบนารี่ก่อน

# ตาราง Truth table ของ Gray code

| สวิทซ์ | Gray code | | | | |
|---|---|---|---|---|---|
| | D | C | B | A | ST |
| $S_0$ | 0 | 0 | 0 | 0 | 1 |
| $S_1$ | 0 | 0 | 0 | 1 | 1 |
| $S_2$ | 0 | 0 | 1 | 1 | 1 |
| $S_3$ | 0 | 0 | 1 | 0 | 1 |
| $S_4$ | 0 | 1 | 1 | 0 | 1 |
| $S_5$ | 0 | 1 | 1 | 1 | 1 |
| $S_6$ | 0 | 1 | 0 | 1 | 1 |
| $S_7$ | 0 | 1 | 0 | 0 | 1 |
| $S_8$ | 1 | 1 | 0 | 0 | 1 |
| $S_9$ | 1 | 1 | 0 | 1 | 1 |

## พิจารณาเอาต์พุตในตารางGray ที่มีค่าเป็น1 มาเขียนเป็นฟังก์ชัน

**Strobe = 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9**

**$2^0$(A) = 1 + 2 + 5 + 6 +9**

**$2^1$(B) = 2 + 3 + 4+ 5**

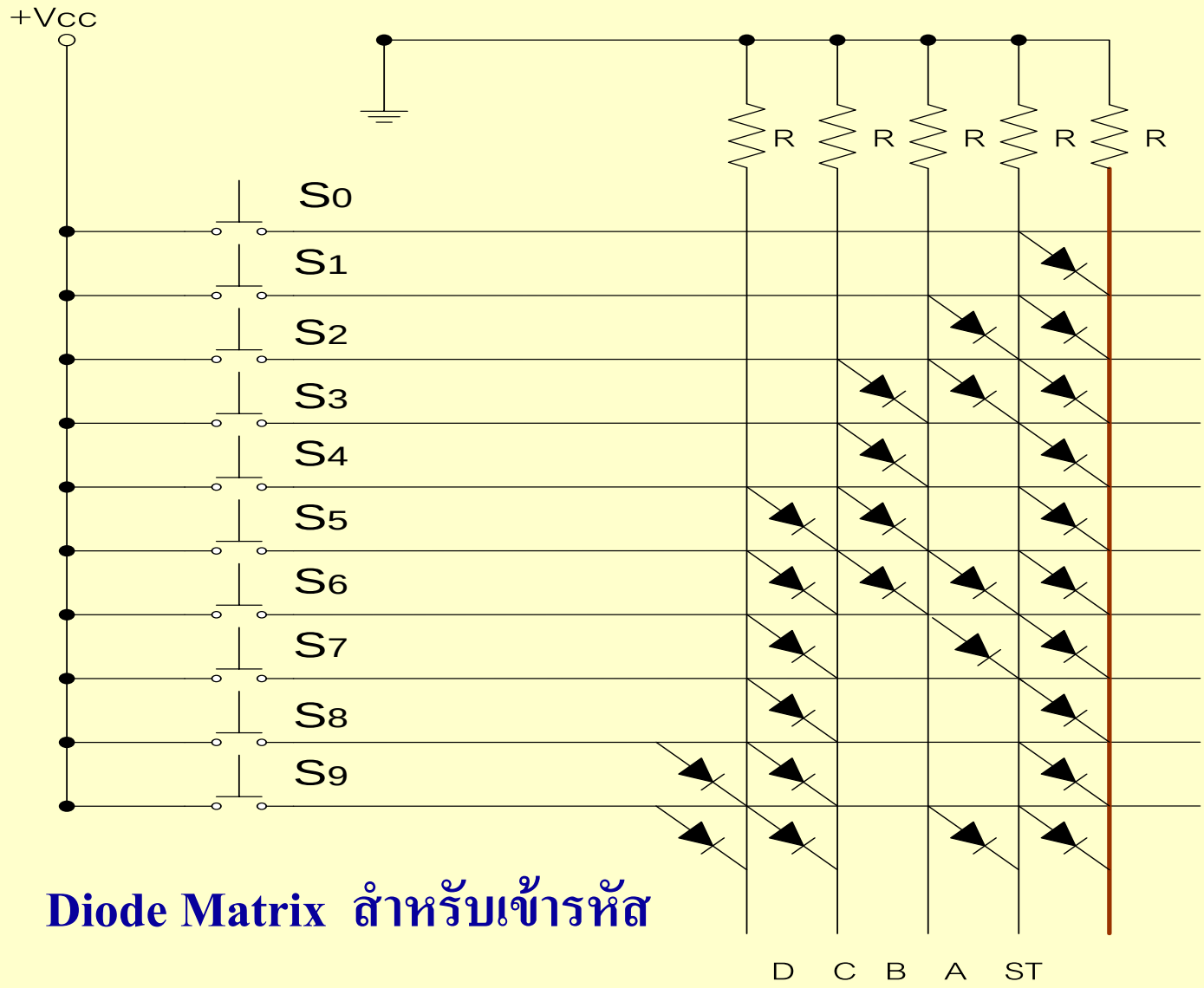
**$2^2$(C) = 4 + 5 + 6 + 7 + 8 + 9**

**$2^3$(D) = 8 + 9**

# อธิบาย

- พิจารณาจากตาราง ค่า Strobe จะทำงานเมื่อมีการกดสวิทซ์ทุกตัว ดังนั้นเอาต์พุตของStrobe จะเป็น1 เมื่อมีการกดสวิทซ์ทางอินพุต
- พิจารณาที่หลักเอาต์พุต $2^0$ หรือตำแหน่ง A จะมีเอาต์พุต ตามตารางรหัสเกร์ยก็ต่อเมื่อมีการกด $S_1$ ,$S_2$, $S_5$, $S_6$, $S_9$
- พิจารณาที่หลักเอาต์พุต $2^1$ หรือตำแหน่ง B จะมีเอาต์พุต ตามตารางรหัสเกร์ยก็ต่อเมื่อมีการกด $S_2$, $S_3$, $S_4$, $S_5$
- พิจารณาที่หลักเอาต์พุต $2^2$ หรือตำแหน่ง C จะมีเอาต์พุต ตามตารางรหัสเกร์ยก็ต่อเมื่อมีการกด $S_4$, $S_5$, $S_6$, $S_7$, $S_8$, $S_9$
- พิจารณาที่หลักเอาต์พุต $2^3$ หรือตำแหน่ง D จะมีเอาต์พุต ตามตารางรหัสเกร์ยก็ต่อเมื่อมีการกด $S_8$, $S_9$

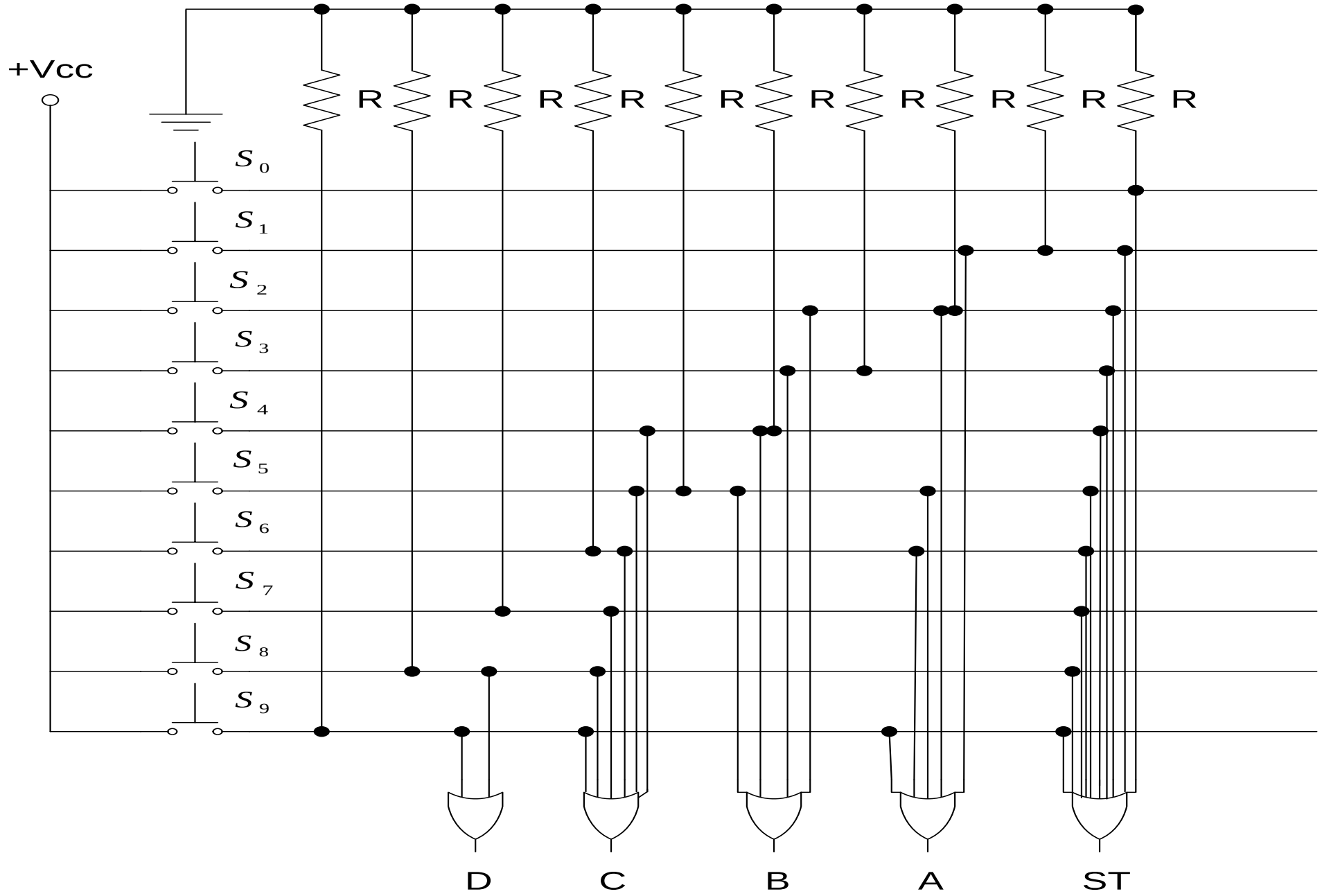
# 434 การออกแบบวงจรเข้ารหัส




**Diode Matrix สำหรับเข้ารหัส**

# อธิบาย

- จากฟังก์ชันที่กล่าวมาสามารถนำมาออกแบบโดยใช้ไดโอดต่อแบบ Matrix ทำงานร่วมกับค่าความต้านทานดังในรูป
- $S_0$-$S_9$ จะต่อร่วมกันทั้งหมดเพื่อไปรับค่าแรงดัน $+V_{CC}$
- $S_0$-$S_9$ อีกปลายของสวิทซ์จะนำไปต่อไดโอดเพื่อสร้างรหัสเกร์ยดังในตารางตามฟังก์ชัน A ,B, C, D
- พิจารณาที่ $S_0$ เมื่อทำการกดสวิทซ์นี้ แรงดัน $+V_{CC}$ นี้จะไปปรากฏที่ขั้วอาโนดของไดโอดที่ต่อกับ ST เกิดแรงดันที่จุดST ดังนี้ $ST=V_{CC}-V_D$ , 5-0.7 =4.3 Volts ค่าแรงดันนี้จะตกคล่อมที่ค่าความต้านทานของจุดต่อ ST
- พิจารณาที่ $S_6$ ค่าแรงดันจะเกิดขึ้นที่ ST, A , B , C , D จะมีค่าเท่ากับ 0101 กับ ST=1


+Vcc
R
R
R
R
R
R
R
R
R
R
$S_0$
$S_1$
$S_2$
$S_3$
$S_4$
$S_5$
$S_6$
$S_7$
$S_8$
$S_9$
D
C
B
A
ST

# อธิบาย

- จากรูปฟังก์ชันที่กล่าวมาเอาต์พุตของStrobe จะเกิดจากการทำงานของสวิทซ์ทุกตัว ดังนั้นสามารถใช้ ออร์เกต ในการสร้างสัญญาณ ST โดยอินพุตของออร์เกต ขนาด 9 อินพุต ไปเชื่อมต่อสวิทซ์ทุกตัวเพื่อรับแรงดัน $+V_{CC}$ โดยมีค่าความต้านทานของแต่ละอินพุตเป็นโหลด ทำให้เกิดแรงดันที่ขั้วของอินพุตออร์เกต
- เอาต์พุตของออร์เกต (A) จะเกิดจากอินพุต $S_1,S_2,S_5,S_6,S_9$
- เอาต์พุตของออร์เกต (B) จะเกิดจากอินพุต $S_2,S_3,S_4,S_5$
- เอาต์พุตของออร์เกต (C) จะเกิดจากอินพุต $S_4,S_5,S_6,S_7,S_8,S_9$
- เอาต์พุตของออร์เกต (D) จะเกิดจากอินพุต $S_8,S_9$

| Decimal number | รหัสเลข BCD | | | |
|---|---|---|---|---|
| | 7421 | 5421 | $2^{*}421$ | $84\bar{2}\bar{1}$ |
| 0 | 0000 | 0000 | 0000 | 0000 |
| 1 | 0001 | 0001 | 0001 | 0111 |
| 2 | 0010 | 0010 | 0010 | 0110 |
| 3 | 0011 | 0011 | 0011 | 0101 |
| 4 | 0100 | 0100 | 0100 | 0100 |
| 5 | 0101 | 1000 | 1011 | 1011 |
| 6 | 0110 | 1001 | 1100 | 1010 |
| 7 | 1000 | 1010 | 1101 | 1001 |
| 8 | 1001 | 1011 | 1110 | 1000 |
| 9 | 1010 | 1100 | 1111 | 1111 |

# อธิบาย

- การแปลงเลขฐานสิบเป็นรหัส BCD
- หลักการของBCD เป็นรหัสที่มีค่าน้ำหนักประจำหลักจำนวน 4บิต
- ที่ผ่านมาส่วนใหญ่มักจะกล่าวถึงรหัสที่เป็นBCD 8421
- BCD มีโค้ดประจำหลักเป็น 7421 ,5421 , 2* 421; 8421
- หลักการของ BCD นี้ จะWeight ค่าของแต่ละตำแหน่งอย่างคงที่ทุกรหัสของโค้ด ดูค่าน้ำหนักได้จากเลขกำกับเช่น BCD 5421,7421
- ค่าที่ควรสังเกตจะมีค่าเป็น 1 ที่หลักของเลข 5-9 ของรหัสBCD เหล่านี้เพื่อที่จะไปรวมให้เกิดน้ำหนักรวมที่ 5-9 ดังกล่าวมา

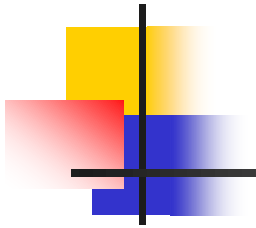
| สวิทซ์ | รหัสเลข BCD | | | | |
|---|---|---|---|---|---|
| | D | C | B | A | ST |
| $Sw_0$ | 0 | 0 | 0 | 0 | 1 |
| $Sw_1$ | 0 | 0 | 0 | 1 | 1 |
| $Sw_2$ | 0 | 0 | 1 | 0 | 1 |
| $Sw_3$ | 0 | 0 | 1 | 1 | 1 |
| $Sw_4$ | 0 | 1 | 0 | 0 | 1 |
| $Sw_5$ | 0 | 1 | 0 | 1 | 1 |
| $Sw_6$ | 0 | 1 | 1 | 0 | 1 |
| $Sw_7$ | 0 | 1 | 1 | 1 | 1 |
| $Sw_8$ | 1 | 0 | 0 | 0 | 1 |
| $Sw_9$ | 1 | 0 | 0 | 1 | 1 |
| $Sw_*$ | 1 | 0 | 1 | 0 | 1 |
| $Sw_\#$ | 1 | 0 | 1 | 1 | 1 |

# 435 การออกแบบวงจรเข้ารหัส BCD-8421

- การสร้างวงจรคอมไบเนชันเข้ารหัส สามารถสร้างได้จากใช้ไดโอดต่อแบบ Matrix หรือใช้วงจรเกตที่กล่าวมา จากตาราง สามารถพิจารณาเอาต์พุตของ Strobe จะมีเอาต์พุตเมื่อทำการกดแป้นรับ
- ส่วนเอาต์พุต D จะเกิดจากอินพุต 1,3,5,7,9,#
- ส่วนเอาต์พุต C จะเกิดจากอินพุต 2,3,6,7,*,#
- ส่วนเอาต์พุต B จะเกิดจากอินพุต 4,5,6,7
- ส่วนเอาต์พุต A จะเกิดจากอินพุต 8,9,*,#

นำเอาต์พุตของตารางมาสร้างวงจรโดยใช้ไดโอดหรือใช้ออร์เกต

## พิจารณาสภาวะเอาต์พุต ของรหัส BCD ที่มีค่าเป็น1 มาเขียนเป็นฟังก์ชัน



**Strobe = 0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + * + #**

**$2^0$ (D) = 1 + 3 + 5 +7+ 9 + #**

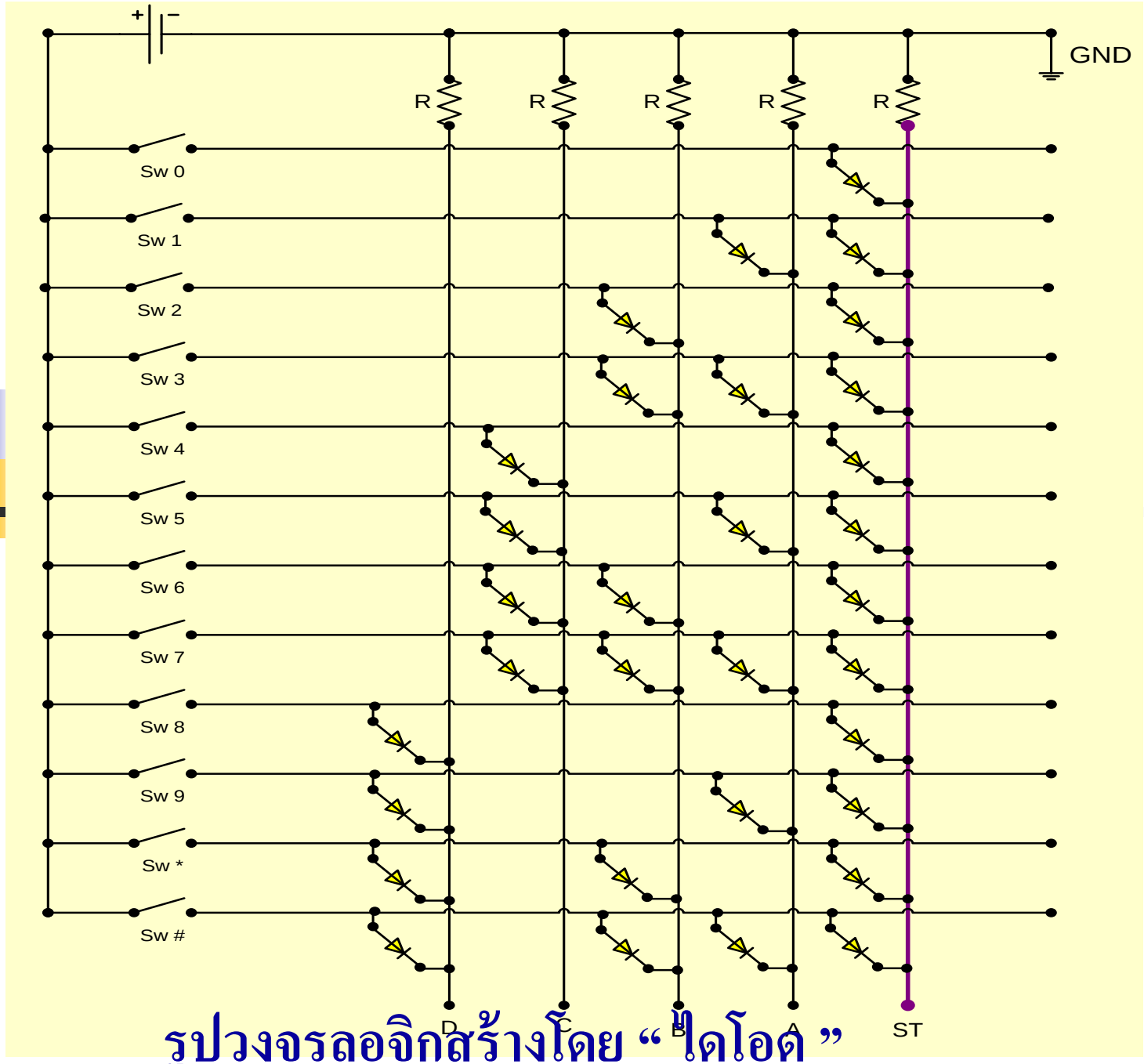
**$2^1$ (C) = 2 + 3 + 6 + 7 + * + #**

**$2^2$ (B) = 4 + 5 + 6 + 7**

**$2^3$ (A) = 8 + 9 + * + #**

# อธิบาย

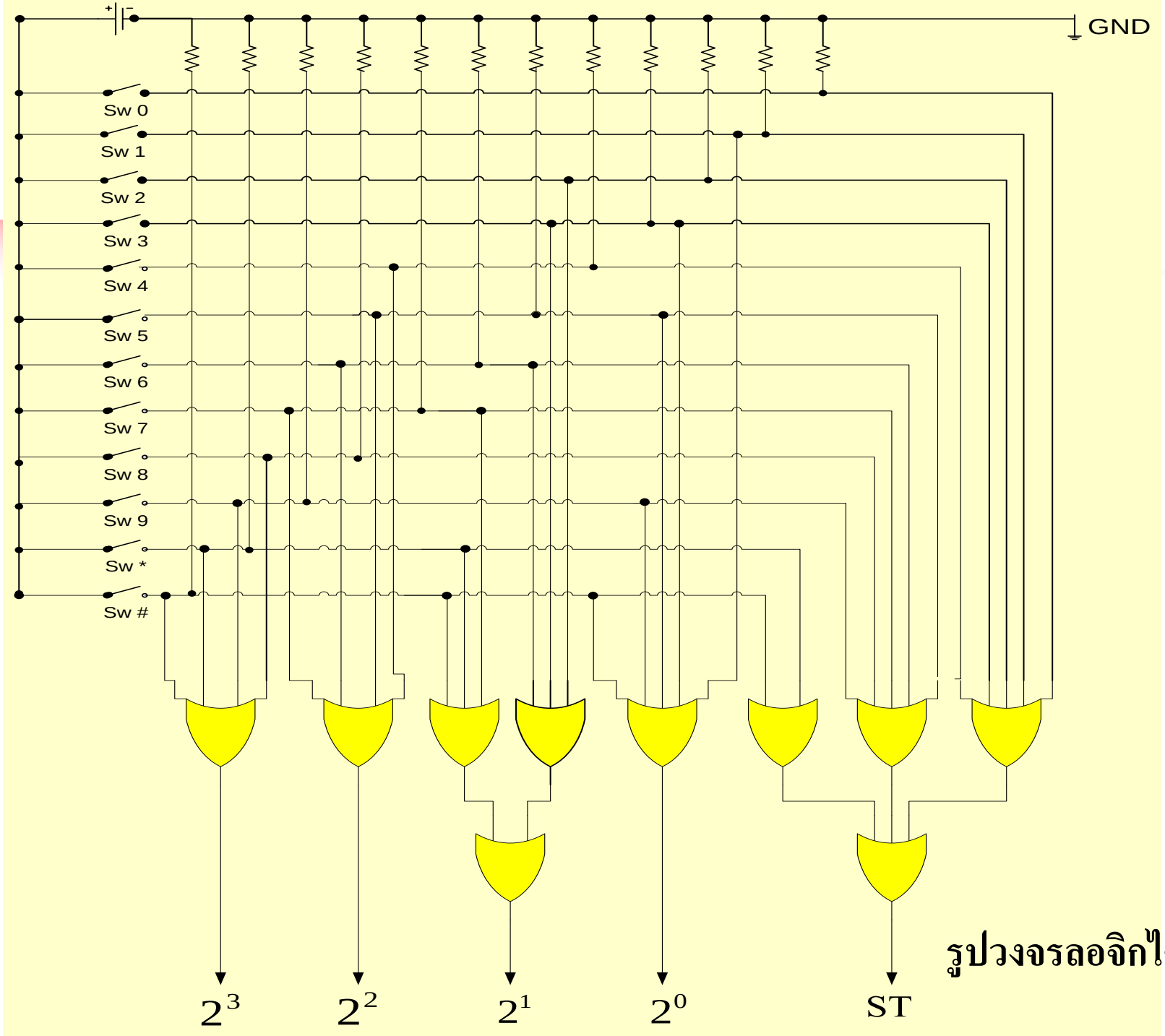
- สภาวะของฟังก์ชัน BCD ที่มีค่า1 นำเขียนได้ดังนี้
- Strobe จะเกิดการทำงานของทุกสวิทซ์ ได้ค่าเป็น 1
- D ค่าเอาต์พุตจะเกิดจาก $Sw_1$,$Sw_2$,$Sw_5$,$Sw_7$,$Sw_9$,$Sw_{\#}$
- C ค่าเอาต์พุตจะเกิดจาก $Sw_2$,$Sw_3$,$Sw_6$,$Sw_7$,$Sw_*$,$Sw_{\#}$
- B ค่าเอาต์พุตจะเกิดจาก $Sw_4$,$Sw_5$, $Sw_6$,$Sw_7$
- A ค่าเอาต์พุตจะเกิดจาก $Sw_8$,$Sw_9$, $Sw_*$, $Sw_{\#}$




# รูปวงจรลอจิกสร้างโดย “ไดโอด”

# อธิบาย

- จากฟังก์ชันเอาต์พุต A ,B ,C ,Dนำมาสร้างวงจรโดยใช้ไดโอดmatrix
- ไดโอดของสวิทซ์แต่ละตัวจะสร้างแรงดันไฟบวกให้แก่เอาต์พุตที่เป็นไปตามรหัส BCD
- เช่น $Sw_4$จะมีไดโอดต่อครบวงจรที่หลัก B ร่วมกับ Strobe ดังนั้นจะสร้างสัญญาณเมื่อกดสวิทซ์ 4 เป็นรหัส BCD คือ 0100 กับ ST=1
- $Sw_7$จะมีไดโอดต่อครบวงจรที่หลัก C, B, A ร่วมกับ Strobe ยกเว้นหลัก D ดังนั้นจะสร้าง สัญญาณเมื่อกดสวิทซ์ 7 เป็นรหัส BCD คือ 0111 กับ ST=1 เป็นต้น




**รูปวงจรลอจิกไดอะแกรม**

# อธิบาย

- ในอีกรูปแบบใช้วงจรคอมไบเนชันในการออกแบบวงจรดังรูป
- เอาต์พุต $2^3$ หรือ D เกิดจากอินพุตดูตารางประกอบ คือ $Sw_8$,$Sw_9$,$Sw_*$,$Sw_\#$
- เอาต์พุต $2^2$ หรือ C เกิดจากอินพุตดูตารางประกอบ คือ $Sw_4$,$Sw_9$,$Sw_*$,$Sw_\#$
- เอาต์พุต $2^1$ หรือ B เกิดจากอินพุตดูตารางประกอบ คือ $Sw_2$,$Sw_3$,$Sw_6$,$Sw_7$,$Sw_*$,$Sw_\#$
- เอาต์พุต $2^0$ หรือ A เกิดจากอินพุตดูตารางประกอบ คือ $Sw_1$,$Sw_3$, $Sw_5$,$Sw_7$,$Sw_9$ ,$Sw_\#$
- ส่วนเอาต์พุตของStrobe จะทำงานเมื่อมีการกดสวิทซ์ทุกฟังก์ชันของสวิทซ์

# การเปรียบเทียบตารางความจริงของ BCD-8421 กับรหัสเกิน 3 ( Excess - 3 code )

| Decimal | BCD – 8421 | Excess - 3 |
|---|---|---|
| 0 | 0000 | 0011 |
| 1 | 0001 | 0100 |
| 2 | 0010 | 0101 |
| 3 | 0011 | 0110 |
| 4 | 0100 | 0111 |
| 5 | 0101 | 1000 |
| 6 | 0110 | 1001 |
| 7 | 0111 | 1010 |
| 8 | 1000 | 1011 |
| 9 | 1001 | 1100 |

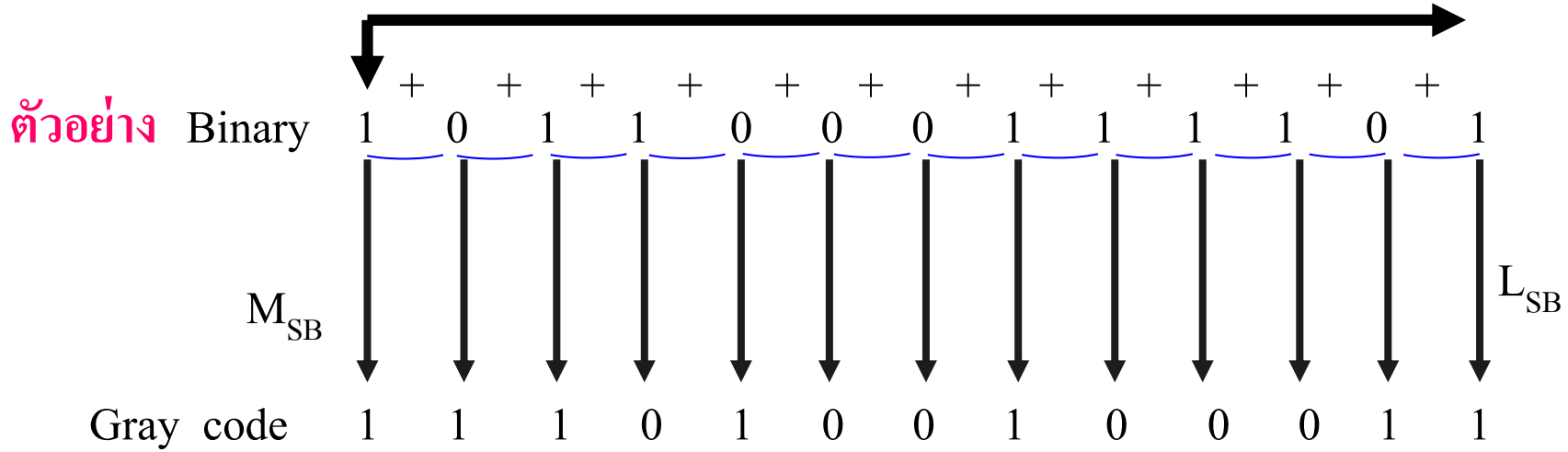
ตารางเปรียบเทียบระหว่าง BCD – 8421 กับรหัสเกิน 3

# อธิบาย

- การเปรียบเทียบตารางความจริงของ BCD-8421 กับ Excess-3
- ค่ารหัสของ Excess-3 จะมีค่ามากกว่ารหัส BCD-8421 อยู่ที่ค่า 3 คือ 0011
- จากตารางจะเห็น BCD-8421 กับ Excess-3 ทุกๆค่าจะต่างกันอยู่ 0011
- ตัวอย่างเลข 2,5,9
- 2 BCD มีค่าเท่ากับ 0010 Exess-3 จะมีค่าเท่ากับ 5 คือ 0101
- 5 BCD มีค่าเท่ากับ 0101 Exess-3 จะมีค่าเท่ากับ 8 คือ 1000
- 8 BCD มีค่าเท่ากับ 1001 Exess-3 จะมีค่าเท่ากับ 1100

# การออกแบบวงจรการแปลงเลข Binary เป็น Gray Code

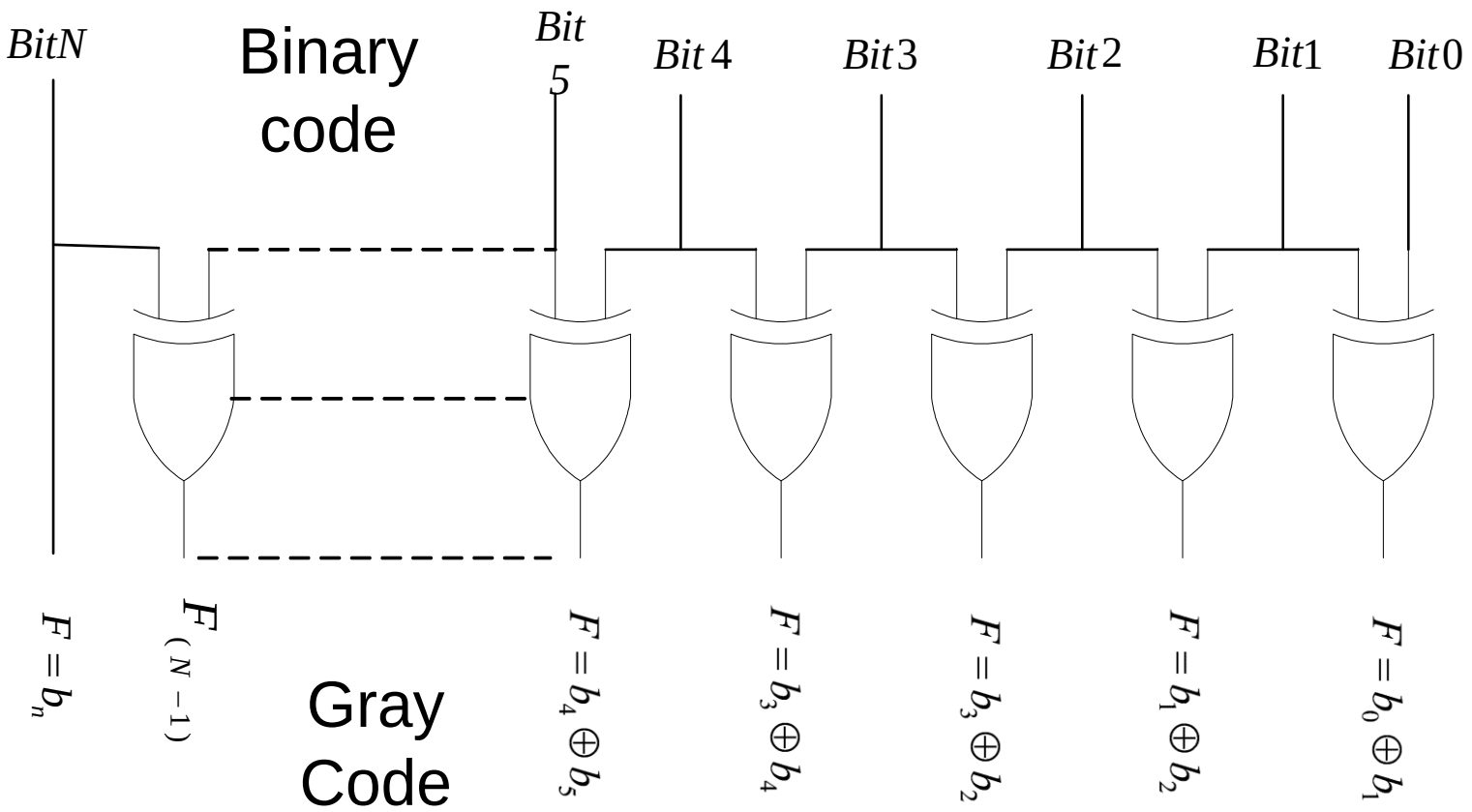
วิธีคิด การแปลงเลขฐานสองเป็นรหัสเกร์ย



# อธิบาย

- หลักการแปลงเลขไบนารี่เป็นรหัสเกร์ย
- ข้อสังเกตบิตที่เกิดจากการแปลงไบนารี่เป็นเกร์ยจะมีจำนวนเท่ากัน
- พิจารณาจากบิตสูงทางซ้ายมือบิตแรกของไบนารี่แปลงเป็นรหัสเกร์ยจะถูกแทนด้วยรหัสเดียวกัน จากนั้นให้นำบิตถัดมาบวกกับบิตแรกทางขวามือของเลขไบนารี่โดยไม่คิดตัวทด ได้เป็นคำตอบของรหัสเกร์ย
- จากนั้นก็ทำการบวกไปในทิศทางขวามือครั้งละ2บิตที่ติดต่อกันได้เป็นผลลัพธ์ของรหัสเกร์ยไปจนถึงบิตที่มีค่าน้อยที่สุด แสดงดังทิศทางในรูป
- การบวกให้คิดถึงผลลัพธ์เพียงอย่างเดียว ตัวทดให้ตัดทิ้ง เช่น 1+1 เท่ากับ 0
- หลักการนี้สามารถนำไปออกแบบวงจรโดยใช้เกตที่เรียนผ่านมา

# วงจรแปลงรหัส Binary เป็นรหัส Gray n-bit



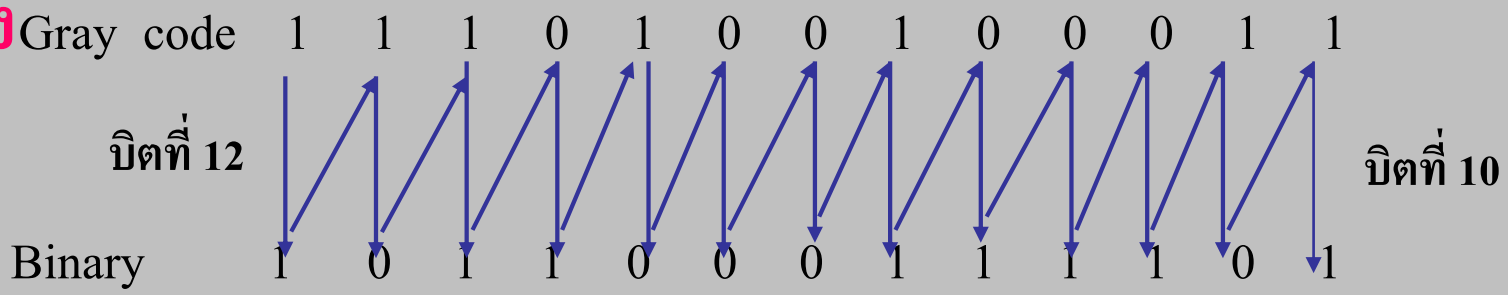
# อธิบาย

- วงจรแปลงรหัสไบนารี่เป็นรหัสเกร์ยจำนวน n บิต
- จากหลักการที่กล่าวมาสามารถนำมาออกแบบเป็นวงจรแปลงรหัสเกร์ยจากไบนารี่แสดงดังรูป สามารถนำ EX-OR เกตมาออกแบบ
- บิตแรกทางซ้ายเหมือนกันถูกดึงลงมา จากนั้นนำบิต n+(n-1) ได้ผลลัพธ์F
- นำบิต (n-1)+(n-2) ได้ผลลัพธ์$F_{(n-1)}$,$F_{(n-2)}$,$F_{(n-3)}$,….$F_0$ การบวกของต่อไปเรื่อยๆ จนถึงบิต 0
- ค่าที่ได้นำผลลัพธ์ของ F ทั้งหมดนำมารวมกันได้เป็นรหัสเกร์ย

# การแปลงเลข Gray Code เป็น Binary

## หลักการแปลงGray code เป็น Binary จำนวน 13 บิต

**ตัวอย่าง**

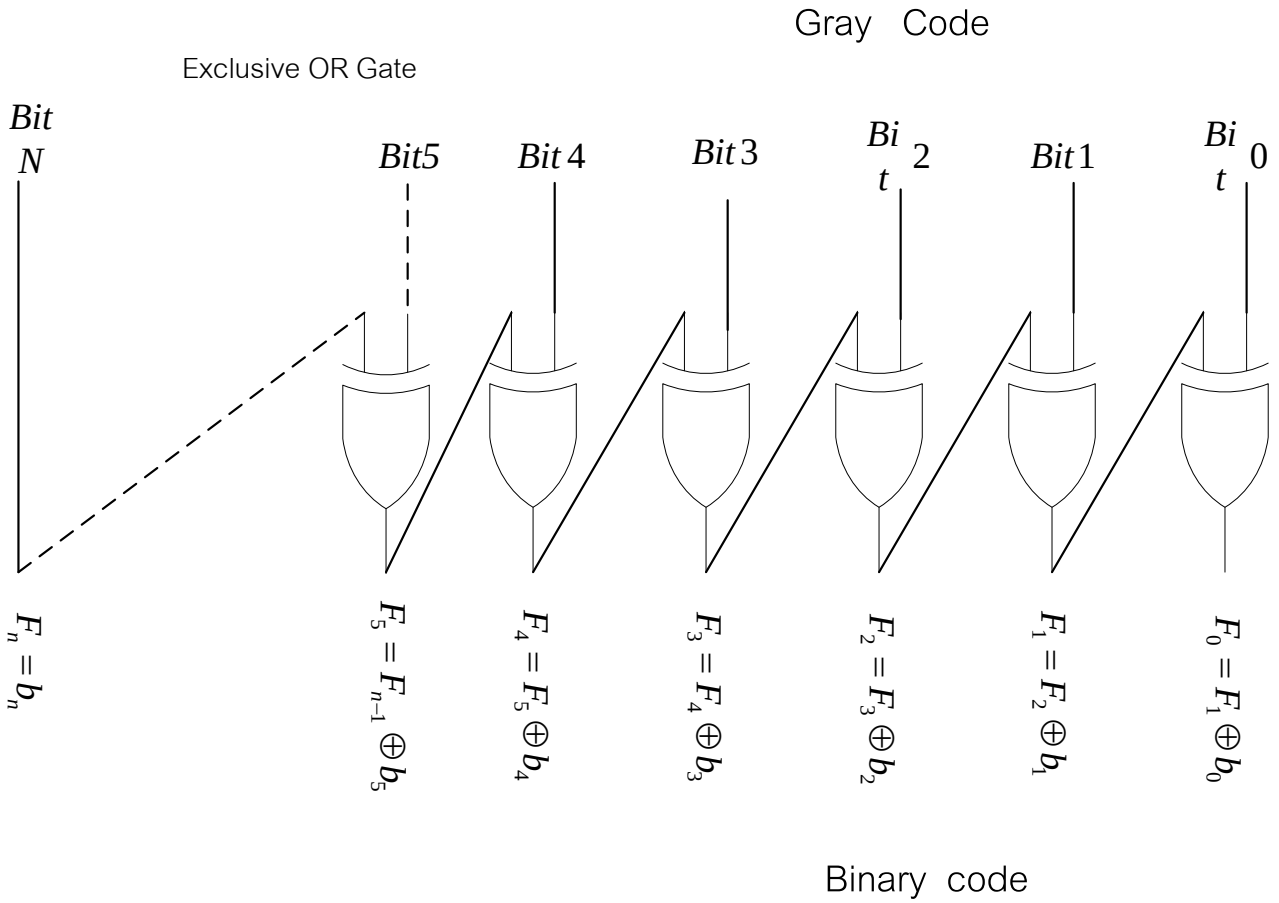


# อธิบาย

- หลักการแปลงรหัสเกร์ยเป็นไบนารี่จำนวน 13 บิต
- หลักการโดยนำค่ารหัสเกร์ยบิตที่ 12 แทนบิตที่12 ของรหัสไบนารี่ตำแหน่งเดียวกัน
- ลำดับสองนำรหัสไบนารี่บิตที่ 12 ไปบวกกับบิตที่ 11 ของรหัสเกร์ยไม่คิดตัวทด ได้ผลลัพธ์เป็นบิตที่11 ของรหัสไบนารี่
- ลำดับสามนำรหัสไบนารี่บิตที่ 11 ไปบวกกับบิตที่ 10 ของรหัสเกร์ยไม่คิดตัวทด ได้ผลลัพธ์เป็นบิตที่10 ของรหัสไบนารี่
- ลำดับสองนำรหัสไบนารี่บิตที่10 ไปบวกกับบิตที่9 ของรหัสเกร์ยไม่คิดตัวทด ได้ผลลัพธ์เป็นบิตที่9 ของรหัสไบนารี่
- ปฏิบัติการอย่างนี้ไปจนถึงบิต 0 ก็จะได้ค่าไบนารี่ครบจนถึง 13 บิตเป็นการแปลงรหัสไบนารี่อย่างสมบูรณ์

# การออกแบบวงจรแปลงรหัสเกร์ยเป็นไบนารี่จำนวน n บิต

## วงจรแปลงรหัส Gray เป็นรหัส Binary n-bit



# อธิบาย

- การออกแบบวงจรรหัสเกร์ยเป็นไบนารี่ จำนวน n บิต
- จากหลักการที่ผ่านมาโดยการนำบิตต่างๆ ที่เกิดขึ้นในรหัสเกร์ยมารวมกับบิตที่เกิดขึ้นในไบนารี่ ก็จะได้ดังแสดงในรูปโดยมี EX-OR เกตเป็นตัวแปลงรหัส
- บิตทางซ้ายมือบิตแรกจะเป็นค่าเดียวกันทั้งรหัสเกร์ยและไบนารี่
- นำบิตของไบนารี่ไปบวกกับบิตเกร์ยในรูปคือบิตที่ 5 ได้ผลลัพธ์บิตที่ 5 ของรหัสไบนารี่ นำบิตที่ 5 ไปบวกกับบิตที่ 4 ของรหัสเกร์ยได้ผลลัพธ์เป็นรหัสไบนารี่ที่4
- บิตถัดไปก็ใช้หลักการเดียวกันจนครบ จนถึงบิต 0 จะได้เป็นรหัสไบนารี่

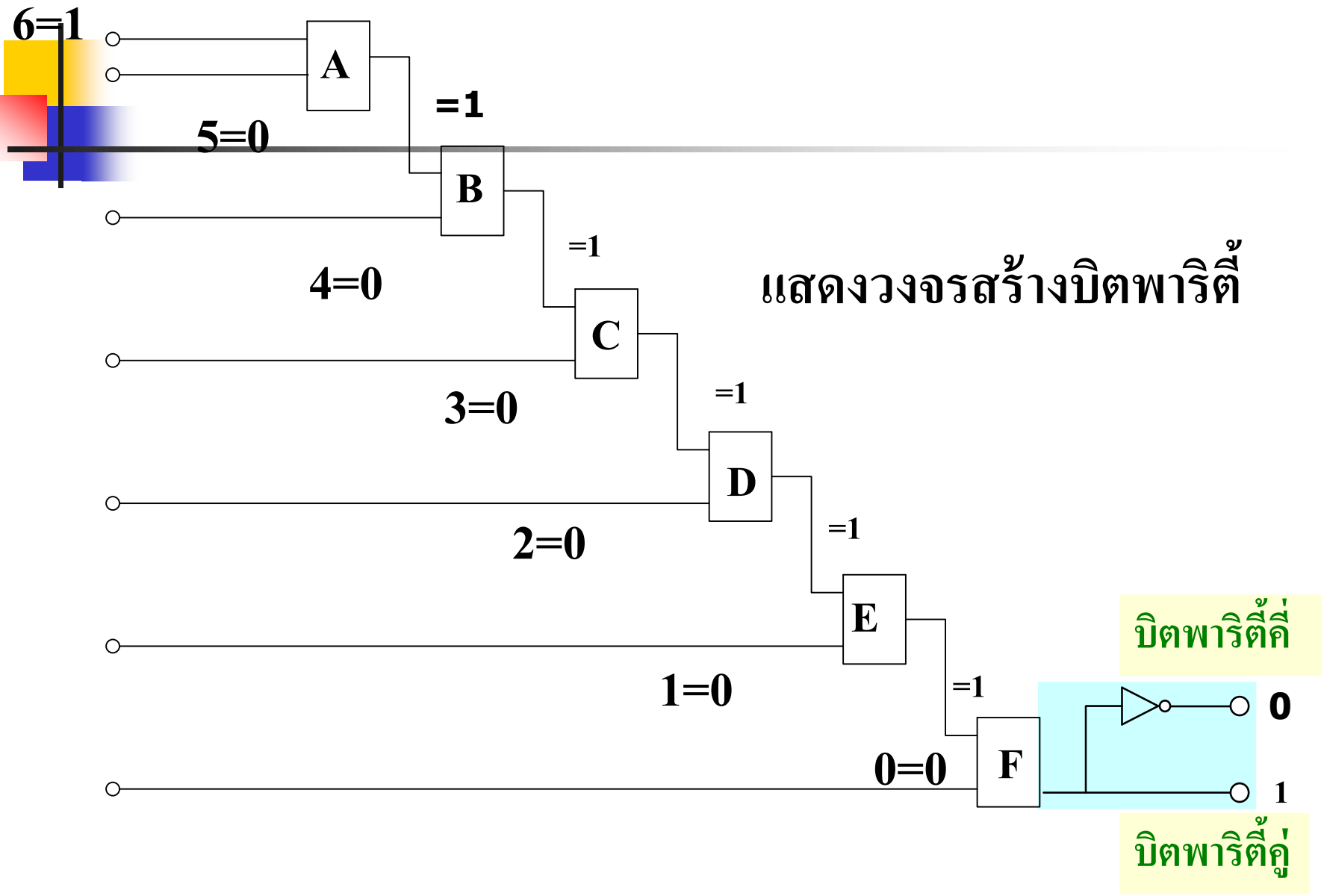
# ตารางแสดงรหัส BCDกับ Excess-3 พร้อมด้วยบิตพาริตี้

| ตัวเลขฐาน 10 | BCD | Odd-parity | Even-parity | Excess-3 | Odd-parity | Even-parity |
|---|---|---|---|---|---|---|
| 0 | 0000 | 1 | 0 | 0011 | 1 | 0 |
| 1 | 0001 | 0 | 1 | 0100 | 0 | 1 |
| 2 | 0010 | 0 | 1 | 0101 | 1 | 0 |
| 3 | 0011 | 1 | 0 | 0110 | 1 | 0 |
| 4 | 0100 | 0 | 1 | 0111 | 0 | 1 |
| 5 | 0101 | 1 | 0 | 1000 | 0 | 1 |
| 6 | 0110 | 1 | 0 | 1001 | 1 | 0 |
| 7 | 0111 | 0 | 1 | 1010 | 1 | 0 |
| 8 | 1000 | 0 | 1 | 1011 | 0 | 1 |
| 9 | 1001 | 1 | 0 | 1100 | 1 | 0 |

# อธิบาย

- ตารางแสดงรหัส BCD กับ Excess-3 พร้อมบิตพาริตี้
- พิจารณาที่ตารางจากเลขฐาน10 จะมีรหัส BCD พร้อมกับบิตพาริตี้ที่เป็นคี่ หมายถึงว่าจำนวนใดๆ ที่แปลงเป็นรหัส BCDในตำแหน่งใดๆ แล้วมีจำนวนเลข 1 เป็นเลขคี่1,3 บิตพาริตี้คี่ (Odd Parity) จะเป็น 0 แต่ถ้ารหัส BCDในตำแหน่งใดๆ มีจำนวน 1 เป็นจำนวนคู่ 2,4 บิตพาริตี้คี่ (Odd Parity) จะเป็น 1
- ในทำนองเดียวกันบิตพาริตี้คู่ (Even Parity) ถ้ารหัส BCDในตำแหน่งใดๆ มีจำนวนเลข 1 เป็นเลขคี่1,3 บิตพาริตี้คู่ก็จะมีค่าเท่ากับ1 ถ้าแปลงรหัส BCDในตำแหน่งใดๆ มีจำนวนเลข1 เป็นเลขคู่ 2,4 บิตพาริตี้คู่ก็จะมีค่าเท่ากับ0
- รหัสอื่นๆในตารางก็ใช้เทคนิคเดียวกัน

# 436 การออกแบบวงจรสร้างบิตตรวจสอบ

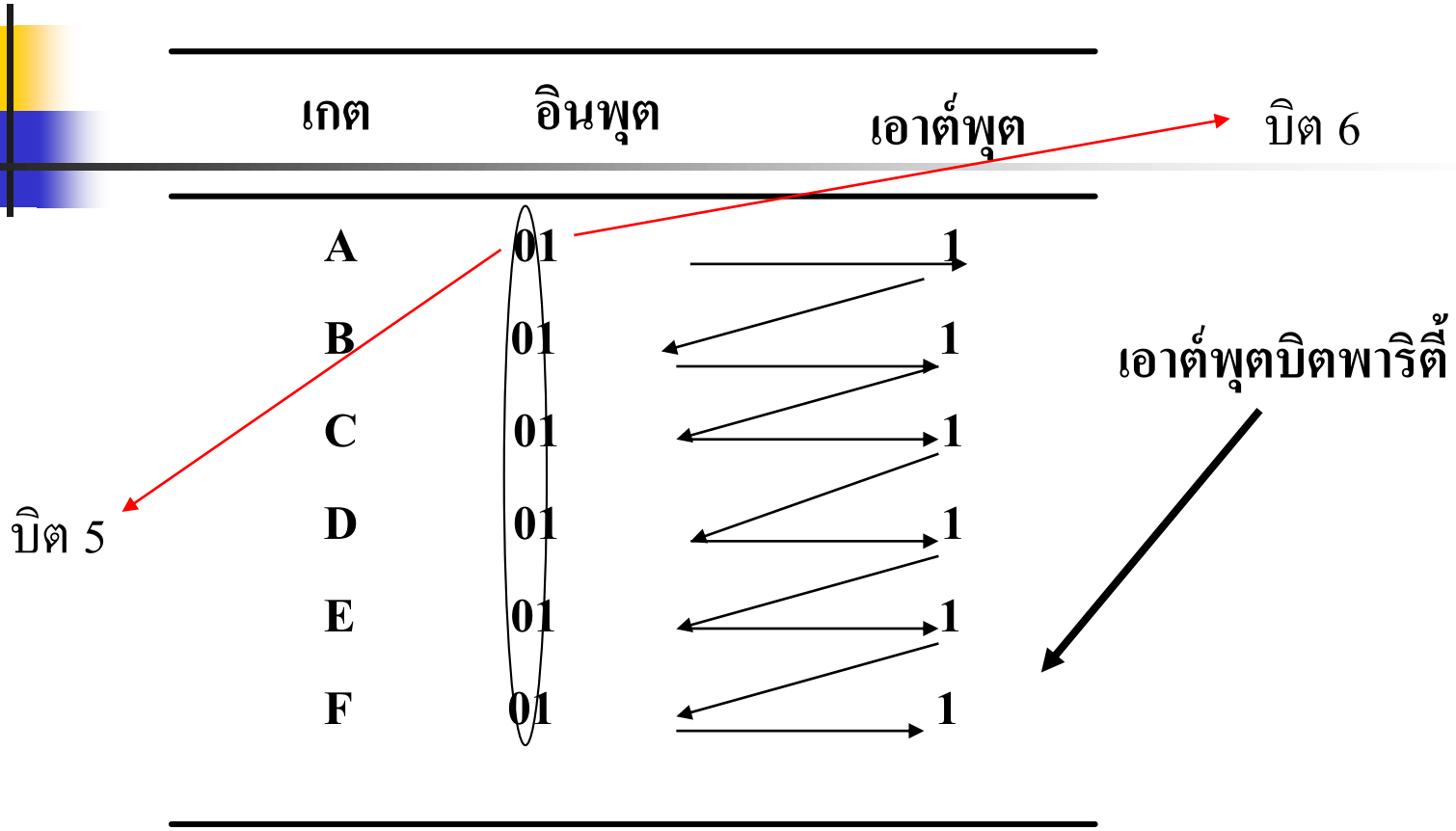



แสดงวงจรสร้างบิตพาริตี้

# อธิบาย

- การออกแบบวงจรสร้างบิตตรวจสอบค่าความผิดพลาด
- ตามมาตรฐาน CCITT (V6) กำหนดให้บิตพาริตี้วางไว้ที่ตำแหน่งบิตที่ 8
- วิธีการโดยใช้ หลักการบวกที่ไม่คิดตัวทด ข้อมูลมีจำนวน 7 บิต
- ใช้ตัวบวก (EX-OR)ไม่คิดตัวทดจำนวน 6 ตัว
- นำค่าแต่ละบิตของข้อมูลมาบวกกันจากบิต 0 บวกกับบิต 1โดยใช้ Block A ได้ค่าเอาต์พุต แล้วนำค่าเอาต์พุตไปบวกกับอินพุตบิต 2 โดยBlock B ได้ค่าเอาต์พุต แล้วนำค่าไปบวกกับบิตต่อไปจนถึงถึงบิตที่ 6 Block F เป็นจำนวนทั้งสิ้น 7 บิต
- ถ้าค่าที่ได้จากการบวกของบิตที่7 มีจำนวนเป็น1 ก็แสดงว่าบิตพาริตี้แบบคู่จะต้องมีค่า 1
- ถ้าค่าที่ได้จากการบวกของบิตที่7 มีจำนวนเป็น0 ก็แสดงว่าบิตพาริตี้แบบคู่จะต้องมีค่า 0
- ในทำนองกลับกันบิตพาริตี้คี่จะตรงกันข้ามกับบิตพาริตี้คู่

# ผลลัพธ์ของวงจรสร้างบิตพาริตี้คู่ ได้ดังนี้



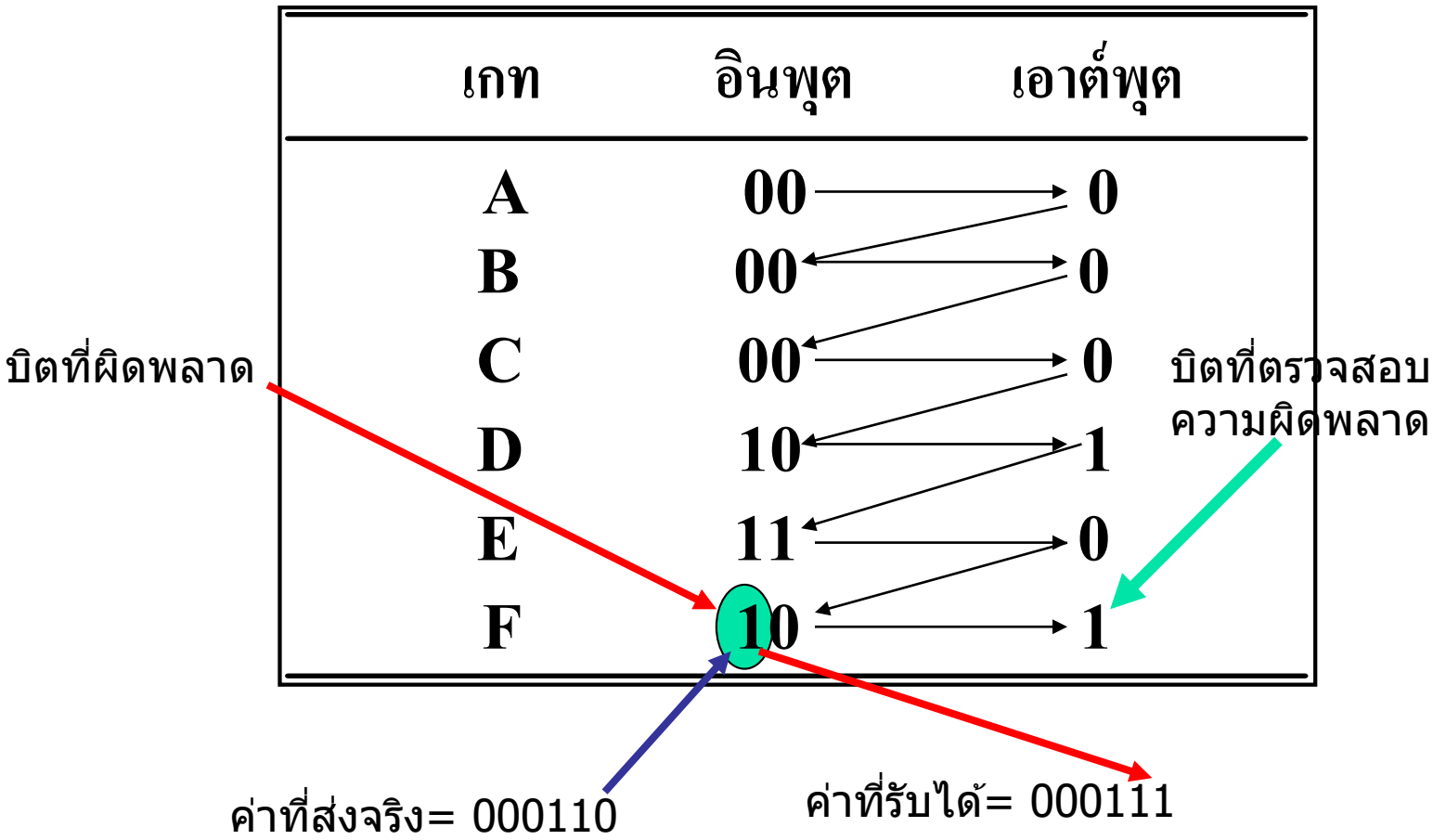

**ค่าเอาต์พุตของเกท F มีค่า1 แสดงถึงต้องการบิตพาริตี้คู่**

# อธิบาย

- การออกแบบวงจรสร้างบิตตรวจสอบ(Parity Bit)
- รหัสที่ทำการสร้างขนาด 7 บิต รวมพาริตี้อีก 1 บิต เป็นรหัส 8 บิต
- หลักการ วิธีตรวจสอบนำบิตทั้งหมดมาเปรียบเทียบ
- โดยเทียบค่าถ้าแตกต่างกันให้ค่า1 ถ้าเหมือนกันจะให้ค่า0
- นำผลที่ได้ไปเปรียบเทียบกับหลักต่อไปจนครบ7 บิต
- ค่าที่เกิดจากการเปรียบเทียบบิตสุดท้ายจะเป็นค่า บิตพาริตี้แบบคู่(Even Parity)
- ค่าที่กำหนด 1000000
- นำค่าบิต7=1 เปรียบเทียบกับบิต6=0 โดยวงจรA ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุต A ไปเปรียบเทียบกับบิต 5=0 โดยวงจรB ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุต B ไปเปรียบเทียบกับบิต 4=0 โดยวงจรC ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุตC ไปเปรียบเทียบกับบิต 3=0 โดยวงจรD ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุตD ไปเปรียบเทียบกับบิต 2=0 โดยวงจรE ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุต E ไปเปรียบเทียบกับบิต 1=0 โดยวงจรF ได้ค่าเอาต์พุตเท่ากับ 1
- นำผลที่ได้จากเอาต์พุต F ไปเป็นบิตพาริตี้ในที่นี้จะเป็นพาริตี้แบบคู่คือค่า 1
- ถ้าต้องการทำเป็นบิตพาริตี้คี่ โดยผ่านวงจรNOTเกต ก็จะได้เป็นวงจรสร้างบิตพาริตี้คี่ มีค่า0

# ตัวอย่างการใช้บิตพาริตี้ตรวจสอบ

กรณีที่มีบิตเพียง หนึ่งบิต ผิดพลาด ทำให้วงจร ด้านรับแสดงค่าลอจิก 1 เพื่อแจ้งความผิดพลาดของข้อมูลขึ้น เราแสดงสถานะบอกแต่ละเกต ดังนี้



# อธิบาย

- อีกตัวอย่าง ค่าทางอินพุตเป็น 0000110 ค่าบิตตรวจสอบคู่ควรจะเป็น 0 เมื่อส่งผ่านไปในวงจรต่างๆแล้วหากเกิดความผิดพลาด เป็น 0000111 ค่าบิตพาริตี้คู่ที่ได้เมื่อมีการผิดพลาดเพียง 1 บิต เป็นค่า 1 ดังนั้นวงจร

- ทำการตรวจสอบแล้วส่งผลว่ามีการผิดพลาด ก็บอกไปยังวงจรที่มีการส่งข้อมูลให้ทำการส่งข้อมูลให้ใหม่ เมื่อมีการรับผิดพลาดแสดงให้เห็นดังตารางการเปรียบเทียบ การผิดพลาดที่บิต 0 ส่งค่า 0 แต่การรับเป็นค่า 1

# กรณีที่มีบิตผิดสองบิตวิธีบิตพาริตี้ ไม่สามารถตรวจพบได้ (นับเป็นข้อเสียอย่างหนึ่ง)

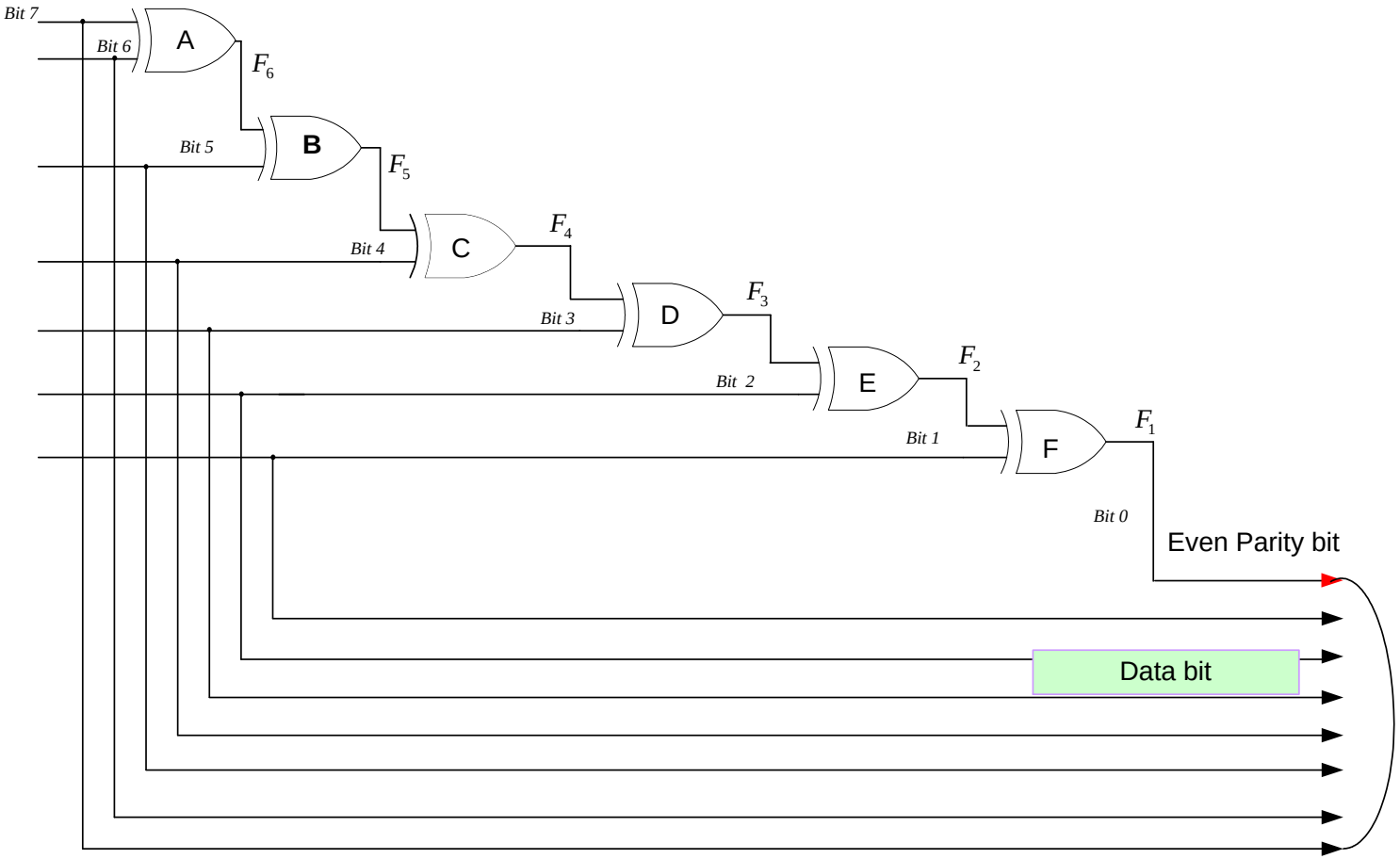
กรณีที่เกิดความผิดพลาด 2 บิตที่ B,F

| เกท | อินพุต | เอาต์พุต |
|---|---|---|
| A | 00 | 0 |
| B | 10 | 1 |
| C | 01 | 1 |
| D | 11 | 0 |
| E | 10 | 1 |
| F | 11 | 0 |

# อธิบาย

- กรณีมีการผิดพลาด 2 บิต ตัวอย่างเดิมถ้าส่งค่า 0000110 บิตตรวจสอบคู่เป็น 0 ในกรณีนี้ค่าบิตที่เกิดเป็น 0010111 มีการผิดพลาดถึง 2 บิต ที่บิต0 , 4 ผลการตรวจสอบได้ค่าเป็น 0 การตรวจสอบถือว่าถูกต้องจึงไม่มีการแจ้งเตือนไปยังวงจรส่ง ค่าที่นำไปใช้มีการผิดพลาด 2 บิต นับได้ว่าเป็นข้อผิดพลาดอย่างหนึ่ง การแก้ไขข้อผิดพลาดอย่างนี้มีการส่งรหัส แก้ไขโดยใช้หลักความน่าจะเป็นในการแก้ไขเมื่อมีการผิดพลาดถึง 2 บิต รหัสนี้จะใช้ในการควบคุมระยะไกลที่มีความแม่นยำและแน่นอนสูงจะทำการตรวจสอบและแก้ไขข้อมูลให้ถูกต้องเหมือนกับต้นฉบับเดิม ซึ่งจะนำไปใช้ในการตวบคุมจรวด, ดาวเทียม, วิถีนำทางไกลม,ยานอวกาศ ซึ่งจะไม่กล่าวในที่นี้

# การสร้างบิตตรวจสอบแบบคู่



# อธิบาย

- การสร้างวงจรบิตตรวจสอบจากหลักการที่กล่าวมาทั้งหมด
- สามารถสร้างค่าจากเอาต์พุตโดยใช้วงจร EX-OR เกต
- ค่าที่ได้จากเอาต์พุต EX-OR เกต นำไปเปรียบเทียบกับบิตถัดไป
- จำนวน 7 บิตสามารถใช้ EX-OR เกต 6 ตัว A-F
- เอาต์พุตของ EX-OR เกต บิตที่0 คือค่า บิตพาริตี้คู่
- บิตข้อมูล (Data bit) สามารถนำมาต่อใช้งานได้โดยตรงเป็นจำนวน 7 บิต

# 437 รหัสที่ใช้ในเครื่องคอมพิวเตอร์

## ตาราง รหัสแอสคี

| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| | | | | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | | | | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | | | 0 | 0 | 0 | 0 | 0 | NUL | (TC7)DLE | SP | 0 | @ | P | . | p |
| | | | 0 | 0 | 0 | 1 | 1 | (TC1)SOH | DC1 | I | 1 | A | Q | a | q |
| | | | 0 | 0 | 1 | 0 | 2 | (TC2)STX | DC2 | “ | 2 | B | R | b | r |
| | | | 0 | 0 | 1 | 1 | 3 | (TC3)ETX | DC3 | # | 3 | C | S | c | s |
| | | | 0 | 1 | 0 | 0 | 4 | (TC4)EOT | DC4 | $ | 4 | D | T | d | t |
| | | | 0 | 1 | 0 | 1 | 5 | (TC5)ENQ | (TC8)NAK | % | 5 | E | U | e | u |
| | | | 0 | 1 | 1 | 0 | 6 | (TC6)ACK | (TC9)SYN) | & | 6 | F | V | f | v |
| | | | 0 | 1 | 1 | 1 | 7 | BEL | (TC10)ETR | * | 7 | G | W | g | w |
| | | | 1 | 0 | 0 | 0 | 8 | FE0(BS) | CAN | ( | 8 | H | X | h | x |
| | | | 1 | 0 | 0 | 1 | 9 | FE(LF) | EM | ) | 9 | I | Y | i | y |
| | | | 1 | 0 | 1 | 0 | 10 | FE2(HT) | SUB | * | : | J | Z | j | z |
| | | | 1 | 0 | 1 | 1 | 11 | FE3(LF) | ESC | + | ; | K | [ | k | : |
| | | | 1 | 1 | 0 | 0 | 12 | FE4(FF) | IS4(FS) | * | < | L | \ | l | ; |
| | | | 1 | 1 | 0 | 1 | 13 | FE5(CR) | IS3(GS) | - | = | M | ] | m | $b_1$ |
| | | | 1 | 1 | 1 | 0 | 14 | SO | IS2(RS) | * | > | N | ^ | n | |
| | | | 1 | 1 | 1 | 1 | 15 | SI | IS1(US) | / | ? | 0 | - | o | _ |

# อธิบาย

- รหัสที่ใช้ในเครื่องคอมพิวเตอร์(ASCII Code) มีมาตรฐาน7 บิต กับ 8 บิต แทนด้วย 1อักษร สามารถแทนค่าได้ 128 ค่า
- ASCII (American Standard Code for Information Interchange) เป็นรหัสที่รับจากคีย์บอร์ด แล้วทำการแปลงเป็นเลขฐาน 2 ในรูปแบบรหัสแอสคีซึ่งมีจำนวน 7 บิต แบ่งเป็นสองส่วน ดังตารางแสดงค่า ถ้าเป็นขนาด8 บิต บิตซ้ายสุดคือค่าตรวจสอบความผิดพลาดตั้งค่า 0
- ตัวอย่าง ข้อมูล A_CAB. แทนด้วย 8 บิต ASCII ได้ดังนี้

| A | blank | C | A | B | . |
|---|---|---|---|---|---|
| 01000001 | 00100000 | 01000011 | 01000001 | 01000010 | 00101110 |

- ในตารางสามารถแยกออกเป็น 2 ส่วน Row แถวประกอบด้วย $b_1$-$b_4$ แสดงค่าตำแหน่งได้ 16 ตำแหน่ง คือค่า 0-15
- ส่วนของคอลัมน์มีค่า 3 บิต ประกอบด้วย b5-b7 แสดงได้ 8คอลัมน์ ไม่ซ้ำกัน
- นำส่วนRow มาตัดColumn จะได้ค่าเป็น ตัวอักษร ส่วนรหัส ASCII ได้จากค่า $b_0$-$b_7$

# ทดสอบความรู้สัปดาห์ที่5

1. จงอธิบายรหัสที่ใช้ในระบบคอมพิวเตอร์ ถึงจำนวนบิต และการทำงานของรหัส
2. จงสร้างรหัสส่วนตัวที่สามารถนำไปสร้างวงจรเข้ารหัสและพร้อมกับบิตที่ทำการตรวจสอบโดยกำหนดเงื่อนไขของเอาต์พุตตามความต้องการที่นักศึกษาต้องการ
3. บอกถึงแนวคิดของการออกแบบวงจรเข้ารหัส
4. เขียนตารางความจริงวงจรเข้ารหัส และแนวทางในการสร้างวงจรถอดรหัส
5. เขียนวงจรที่ออกแบบเข้ารหัส

# เนื้อหาคำบรรยายในสัปดาห์ที่ 6 การถอดรหัส ( Decoder)

## 44 การถอดรหัสทางดิจิตอลของเลขฐานสอง

441 การออกแบบวงจรถอดรหัสฐานสองขนาด 2 บิต

442 การออกแบบวงจรถอดรหัสฐานสองขนาด 3 บิต

443 การออกแบบวงจรถอดรหัสฐานสองขนาด 4 บิตแบบ Tree

444 การออกแบบวงจรถอดรหัสฐานสองขนาด 14 บิตแบบTree

445 การออกแบบวงจรถอดรหัสฐานสองขนาด n บิต

# เนื้อหาคำบรรยายในสัปดาห์ที่ 6 การถอดรหัส ( Decoder) (ต่อ)

**45 การออกแบบวงจรถอดรหัส BCD -8421 แสดงผล แบบแอลอีดี seven segment**

451 การออกแบบวงจรถอดรหัส BCD-8421 แสดงผล แบบแอลอีดี seven segment มีคาโถดต่อร่วมกัน (Common Cathoad)

452 การออกแบบวงจรถอดรหัสBCD-8421 แสดงผล แบบแอลอีดี seven segment มีอาโถดต่อร่วมกัน(Common Anode)

453 การออกแบบวงจรถอดรหัสฐานสอง แบบแสดงผลเอาต์พุตเดียว

454 การออกแบบวงจรถอดรหัสฐานสองเป็นรหัสเกร์ย

**46 การประยุกต์ใช้วงจรถอดรหัสในการออกแบบวงจรลอจิกเกต**

461 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Minterm

462 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูป Maxterm

# การเข้ารหัส และ ถอดรหัส (Encoder and Decoder)

- **การเข้ารหัส (Encoder)** ในสัปดาห์ที่ 5 กล่าวถึงเรื่องของวงจร Logic หมายความว่า การเปลี่ยนแปลงระดับของ Logic จากสวิทช์มาเป็นสัญญาณ Logic ตามรหัสที่เราต้องการ เช่น ถ้ามีสวิทช์อยู่ 10 ตัว กดสวิทช์เลขหมายใดก็ให้ Output ของวงจรเป็นสัญญาณ Logic ของรหัส Binary เป็นต้น ซึ่งเราจะให้ Output ของวงจรเป็นรหัสอะไรก็ขึ้นอยู่กับความต้องการของผู้ใช้ ในทำนองเดียวกัน

- **การถอดรหัส (Decoder)** ก็หมายความถึง วงจรซึ่งเปลี่ยนรหัสทาง Input เพื่อเป็นรหัสอื่นตามต้องการ เช่น เปลี่ยนรหัส BCD เป็นแรงดันของ เลขฐานสิบเป็นต้น วงจรเข้ารหัสและ ถอดรหัสนี้ สามารถสร้างขึ้นมาจาก Diode ซึ่งต่อกันแบบ Matrix หรือ จากวงจร Gate Combination ก็ได้ ซึ่งทั้งสองชนิดขึ้นอยู่กับความเหมาะสมของผู้ใช้ในแง่ของขนาดความยุ่งยาก ราคา ฯลฯ

-**วงจร Gate combination** สำหรับวงจรถอดรหัส BCD -8421 เป็นเลขฐานสิบจากวงจรดังกล่าว ถ้าเราเอา LED มาต่อที่ Output LED จะสว่างเฉพาะ Output ใด Output หนึ่งตามรหัส BCD - 8421 ของ Input

- **หากต้องการแสดงผลออกมาในรูปของ LED 7 ส่วน (segment )** ซึ่งจะทำให้อ่านการแสดงผลได้ง่ายขึ้นวงจรถอดรหัสก็ต้องเป็นวงจรถอดรหัสสำหรับจุด LED 7 ชิ้นส่วนที่กล่าวมาแล้วทั้งหมด

- **ตัวอย่างต่อไปเป็นตารางถอดรหัสจาก BCD เป็นรหัส GRAY** ค่าเอาต์พุตจะนำไปออกแบบวงจรเกตโดยพิจารณาทีละขั้นตอนดังตาราง k-map

# 44 การถอดรหัสทางดิจิตอลของเลขฐานสอง

- ถอดรหัสในรูปแบบตำแหน่ง แสดงผลในเลขฐาน10 โดยใช้ LED แสดงตำแหน่ง 0,1,2,3,4,5,6,7,8,9.

- ถอดรหัสในรูปแบบแสดงผลเป็นเลขฐาน 10 แสดงผลในรูปแบบ Seven segment

- ถอดรหัสเลขฐานสองในรูปแบบรหัสเกร์ย

# 441 การออกแบบวงจรถอดรหัส

**การออกแบบวงจรถอดรหัสฐานสองขนาด 2 บิตแบบแสดงโดยตำแหน่ง LED**

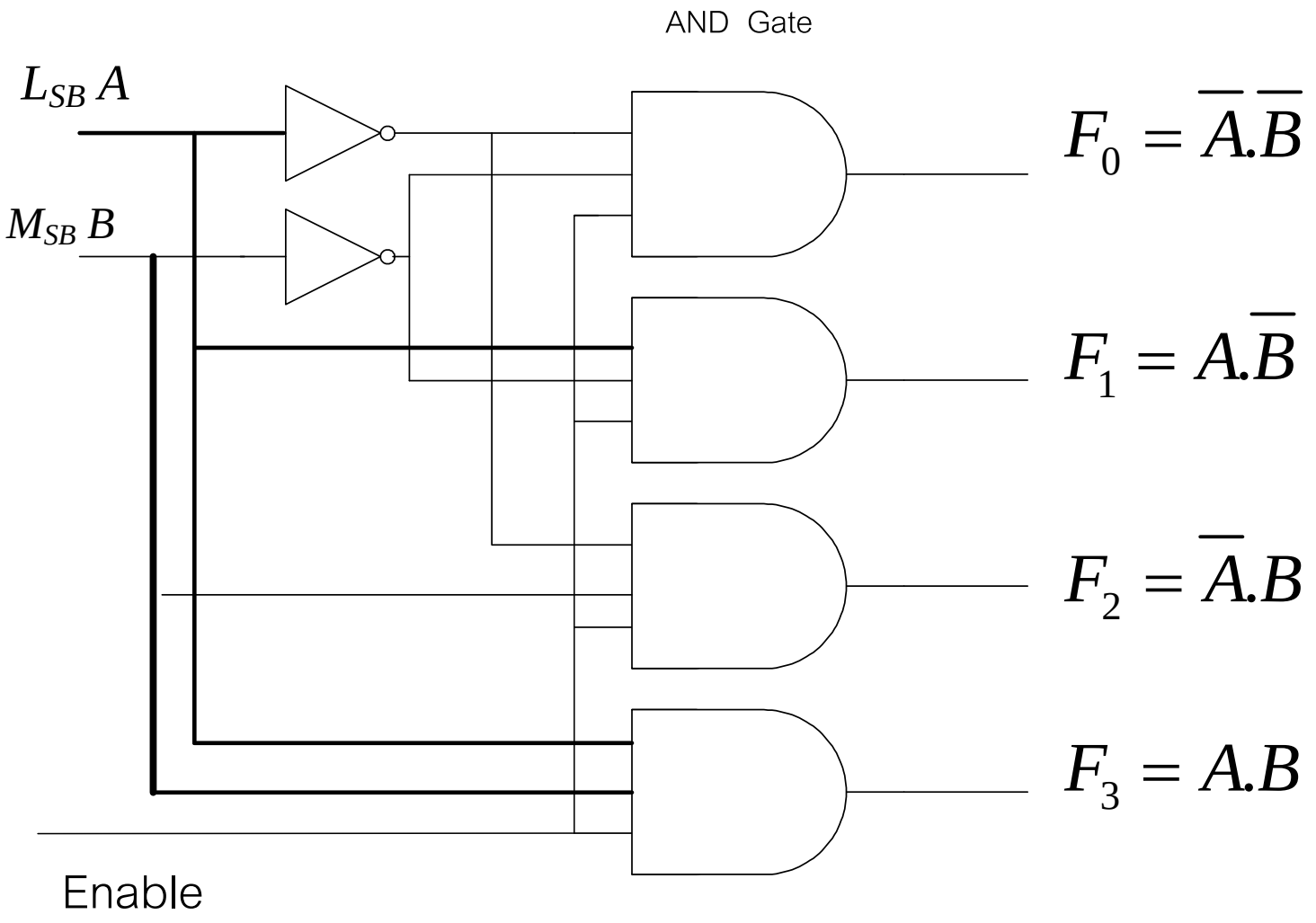
| B | A | $F_0$ | $F_1$ | $F_2$ | $F_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

ตารางแสดงค่าของวงจรถอดรหัส

# อธิบาย

- การออกแบบวงจรถอดรหัสฐานสอง ขนาด 2 บิต
- พิจารณาจากตารางแสดงค่าของวงจรถอดรหัส(Decorder Circuit)
- ค่าที่ได้ทางเอาต์พุตจะได้ค่าที่แตกต่างกัน 4 ค่า โดยพิจารณาแต่ละเอาต์พุตในเทอมของ minterm ได้ดังต่อไปนี้
  - ค่า $F_0=\bar{A}\bar{B}$
  - ค่า $F_1=A\bar{B}$
  - ค่า $F_2=\bar{A}B$
  - ค่า $F_3=AB$
- สามารถเขียนเป็นวงจรถอดรหัสพร้อมด้วยสัญญาณควบคุม (Enable) ของวงจร ดังรูปแสดงถัดไป
- การนำไปใช้งานเช่นแสดงสภาวะของลิฟท์ขนของที่ จำนวนไม่เกิน 7 ชั้นว่าขณะนี้ลิฟท์กำลังเคลื่อนอยู่ชั้นใด

# 2 to 4 Decoder with Enable



# 442 การออกแบบวงจรถอดรหัสฐานสองขนาด 3 บิต

| C | B | A | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

ตารางแสดงค่าวงจรถอดรหัส 3 บิต

# อธิบาย

- พิจารณาจากตารางความจริงของวงจรถอดรหัสขนาด 3 บิต
- สัญญาณทางอินพุตที่จะนำไปถอดรหัสคือ A,B,C
- สัญญาณทางเอาต์พุต ที่ได้ออกมาในรูป minterm มีดังต่อไปนี้

$-F_0 = \bar{A}\bar{B}\bar{C}$

$-F_1 = A\bar{B}\bar{C}$

$-F_2 = \bar{A}B\bar{C}$

$-F_3 = \bar{A}BC$

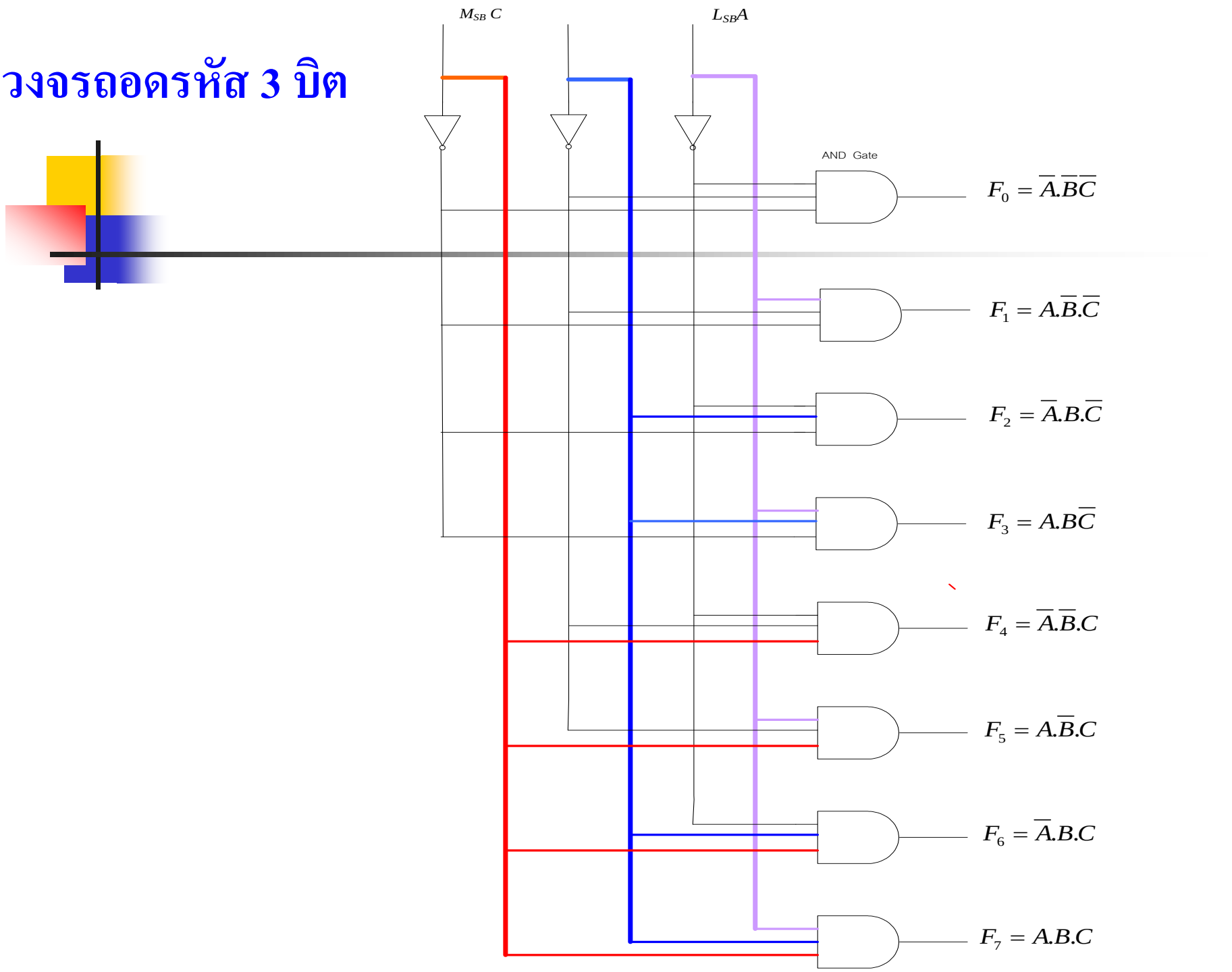
$-F_4 = A\bar{B}\bar{C}$

$-F_5 = A\bar{B}C$

$-F_6 = AB\bar{C}$

$-F_7 = ABC$

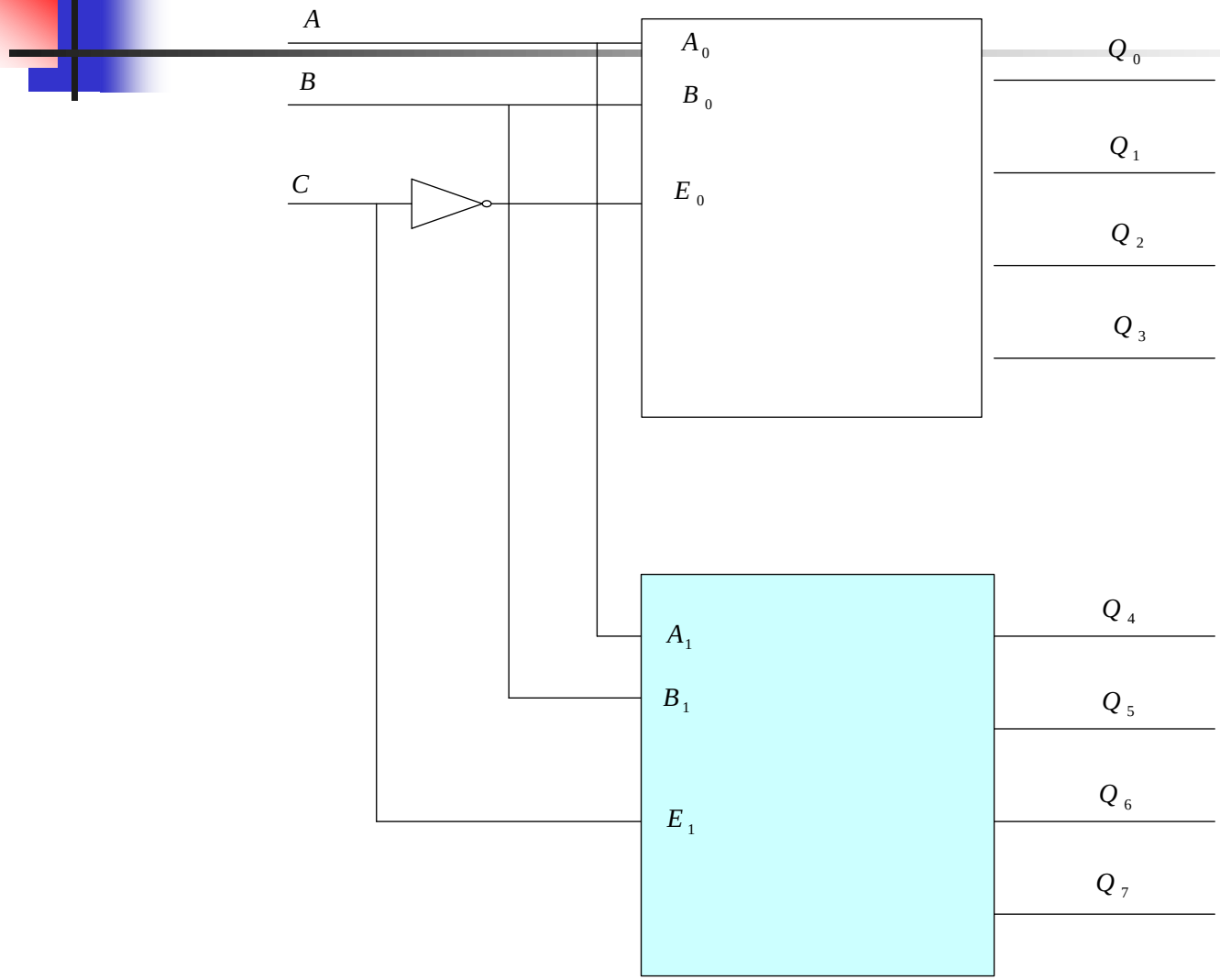
# วงจรถอดรหัส 3 บิต



# อธิบาย

- จากค่าที่ minterm ในตารางดังกล่าวสามารถนำค่าที่ได้จากเอาต์พุตมาเขียนเป็นวงจร
- ค่าเอาต์พุตของ $F_0$ จะเป็น 1 ก็ต่อเมื่ออินพุต A=0, B=0, C=0 วงจรจะทำการถอดรหัสในตำแหน่ง $F_0$ โดยนำ NOT A ไปAND กับ NOT B และ NOT C
- ค่าเอาต์พุตของ $F_1$ จะเป็น 1 ก็ต่อเมื่ออินพุต A=1, B=0, C=0 วงจรจะทำการถอดรหัสในตำแหน่ง $F_1$ โดยนำ A ไป AND กับ NOT B และ NOT C
- ค่าเอาต์พุตของ $F_2$ จะเป็น 1 ก็ต่อเมื่ออินพุต A=0, B=1, C=0 วงจรจะทำการถอดรหัสในตำแหน่ง $F_1$ โดยนำ NOT A ไป AND กับ B และ NOT C
- ในค่าตำแหน่งที่ $F_3$-$F_7$ก็พิจารณาดังที่กล่าวมา
- ค่าเอาต์พุตที่แสดงขนาด 3 บิต จะไม่ซ้ำกัน 8 ตำแหน่ง

# การเพิ่มบิตควบคุมเพื่อเพิ่มขนาดของวงจรถอดรหัส



# อธิบาย

- การสร้างวงจรถอดรหัสขนาด 3 บิต โดยการนำวงจรถอดรหัสขนาด 2 บิตมาใช้งาน
- บิต $A_0$ , $B_0$ จะทำงานที่ตำแหน่ง 0-3 และซ้ำใหม่ 4-7 โดยมีบิต C เป็นบิตตัวแปร
- วงจรถอดรหัส ขนาด 3บิต พิจารณาบิต $M_{SB}$ หรือบิต C จะมีค่าเท่ากับ 0 ในตำแหน่งของเลขฐานสอง ในตำแหน่งเลขฐานสิบเท่ากับ 0 - 3 และทีบิต C มีค่าเท่ากับ 1 ในตำแหน่งเลข 4-7 จึงนำมาเป็นบิตควบคุมการทำงานของวงจรถอดรหัส 2 บิตได้ดังรูป
- การทำงานของวงจรถอดรหัสแบบนี้ จะใช้วงจรถอดรหัส 2บิต ที่มีตัวควบคุม Enable ใช้บิต C ทำงานร่วมกับตัวอินเวอร์เตอร์ทางด้านอินพุต
- $A_0$,$B_0$ จะต่อขนานกับ$A_1$,$B_1$ เป็นตัวเลือกตำแหน่งเอาต์พุต จะมีค่าตั้งแต่ 00-11
- $E_0$และ$E_1$จะทำหน้าที่เลือกวงจรถอดรหัส $Q_0$-$Q_3$ เมื่อบิต C มีค่าเท่ากับ 0 ถ้าบิตC มีค่าเท่ากับ 1 วงจรถอดรหัสจะเลือกเอาต์พุต $Q_4$-$Q_7$

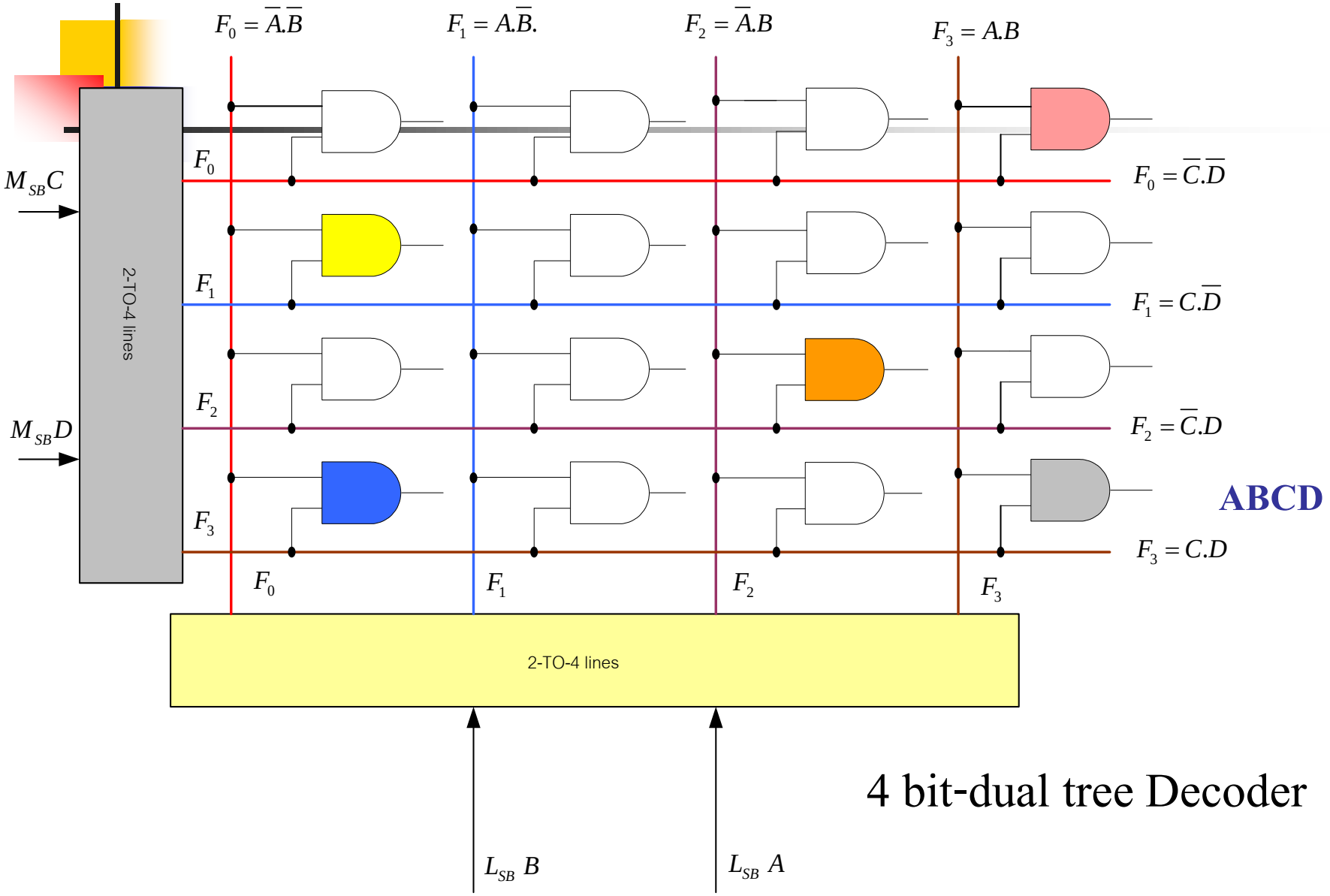
# 443 การถอดรหัสเลขฐานสองขนาด 4 บิตแบบ Tree ออกแบบวงจร

| DC | BA | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ | $F_{10}$ | $F_{11}$ | $F_{12}$ | $F_{13}$ | $F_{14}$ | $F_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **00** | 00 | $\bar{A}\bar{B}\bar{C}\bar{D}$ | | | | | | | | | | | | | | | |
| **00** | 01 | | $\bar{A}\bar{B}\bar{C}D$ | | | | | | | | | | | | | | |
| **00** | 10 | | | $\bar{A}\bar{B}C\bar{D}$ | | | | | | | | | | | | | |
| **00** | 11 | | | | $\bar{A}\bar{B}CD$ | | | | | | | | | | | | |
| *01* | 00 | | | | | $\bar{A}B\bar{C}\bar{D}$ | | | | | | | | | | | |
| *01* | 01 | | | | | | $\bar{A}B\bar{C}D$ | | | | | | | | | | |
| *01* | 10 | | | | | | | $\bar{A}BC\bar{D}$ | | | | | | | | | |
| *01* | 11 | | | | | | | | $\bar{A}BCD$ | | | | | | | | |
| 10 | 00 | | | | | | | | | $A\bar{B}\bar{C}\bar{D}$ | | | | | | | |
| 10 | 01 | | | | | | | | | | $A\bar{B}\bar{C}D$ | | | | | | |
| 10 | 10 | | | | | | | | | | | $A\bar{B}C\bar{D}$ | | | | | |
| 10 | 11 | | | | | | | | | | | | $A\bar{B}CD$ | | | | |
| 11 | 00 | | | | | | | | | | | | | $AB\bar{C}\bar{D}$ | | | |
| 11 | 01 | | | | | | | | | | | | | | $AB\bar{C}D$ | | |
| 11 | 10 | | | | | | | | | | | | | | | $ABC\bar{D}$ | |
| 11 | 11 | | | | | | | | | | | | | | | | $ABCD$ |

# อธิบาย

- พิจารณาจากตารางแสดงถึงค่าเอาต์พุตของวงจรถอดรหัสขนาด 4 บิต
  ทางด้านอินพุต จะแบ่งเป็นสัญญาณขนาด 4 บิต คือ ABCD
  ทางด้านเอาต์พุต จะแบ่งเป็น $F_0$-$F_{15}$ หรือ มีค่าเอาต์พุตจำนวน 16 ค่าที่ไม่ซ้ำกันดังตาราง
- พิจารณาทางด้านอินพุต หากแบ่งค่าเป็น 2 ส่วน คือบิตทาง $L_{SB}$ , บิตทาง $M_{SB}$
- บิต $M_{SB}$ ทางอินพุต D,C ที่มีค่าเป็น 00 จะแสดงค่าในเลขฐานสอง คือ 0-3
- บิต $M_{SB}$ ทางอินพุต D,C ที่มีค่าเป็น 01 จะแสดงค่าในเลขฐานสอง คือ 4-7
- บิต $M_{SB}$ ทางอินพุต D,C ที่มีค่าเป็น 10 จะแสดงค่าในเลขฐานสอง คือ 8-11
- บิต $M_{SB}$ ทางอินพุต D,C ที่มีค่าเป็น 11 จะแสดงค่าในเลขฐานสอง คือ 12-15
- ส่วนบิตทาง $L_{SB}$ ทางอินพุต B,A จะเปลี่ยนค่าจาก 00 -11 ซ้ำดังค่าที่แสดงในตาราง
- ในส่วนของบิต $M_{SB}$ จะใช้ถอดรหัสขนาด 2 บิต
- ในส่วนของบิต $L_{SB}$ จะใช้ถอดรหัสขนาด 2 บิต
- ลักษณะการต่อแบบทรี มีการเชื่อม $M_{SB}$ , $L_{SB}$ ด้วยแอนด์เกต

# การออกแบบวงจรถอดรหัสเลขฐานสองขนาด 4 บิตแบบ Tree




4 bit-dual tree Decoder

# อธิบาย

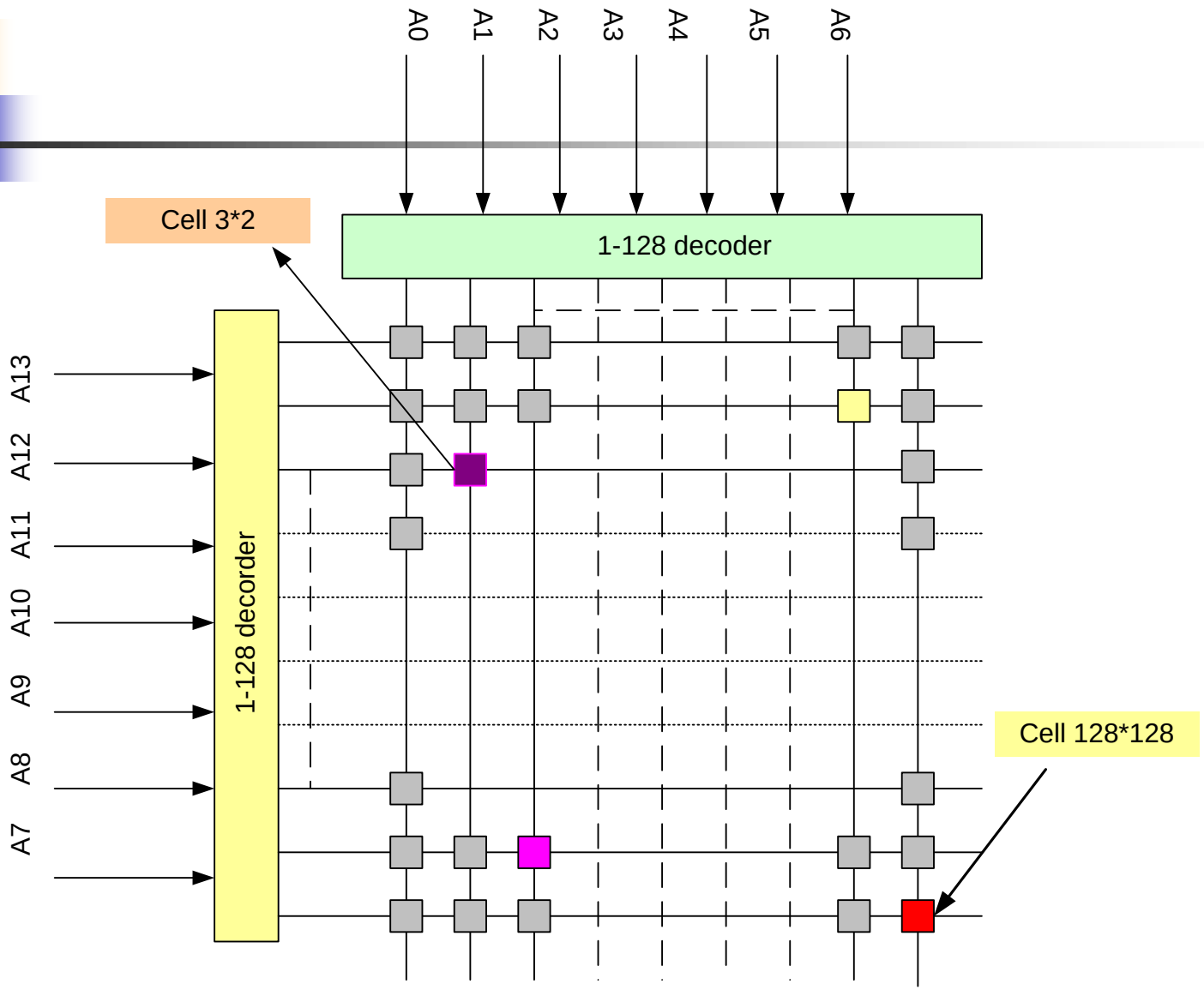
- การสร้างวงจรถอดรหัสขนาด 4 บิตแบบต้นไม้(Tree) โดยใช้วงจรถอดรหัส 2 บิต 2 วงจร มาใช้งานร่วมกับ AND เกต ดังรูป
- กำหนดบิตCD ทางค่าสูงทำหน้าที่เลือกแถว (Row) ซึ่งจะมีค่าเท่ากับ 00-11
- บิตAB ทางค่าต่ำทำหน้าที่เลือกคอลัมน์(Column)ที่จะแสดงค่าตำแหน่งทางเอาต์พุต
- ตัวอย่างที่กำหนดค่าเป็น DC BA

  - ค่า DC จะแสดงค่าในตำแหน่งแถวที่ 4 หรือ $F_3$

  - ค่า AB จะแสดงค่าในตำแหน่งคอลัมน์ที่ 4 หรือ $F_3$

  - ค่าที่ได้จะเข้าทางอินพุตของวงจร AND เกต แสดงถึงเอาต์พุตของค่าที่ถูกถอดรหัสในตำแหน่ง DC BA ดังแสดงในรูป

# 444 การออกแบบวงจรถอดรหัสฐานสองขนาด 14 บิตแบบ Tree



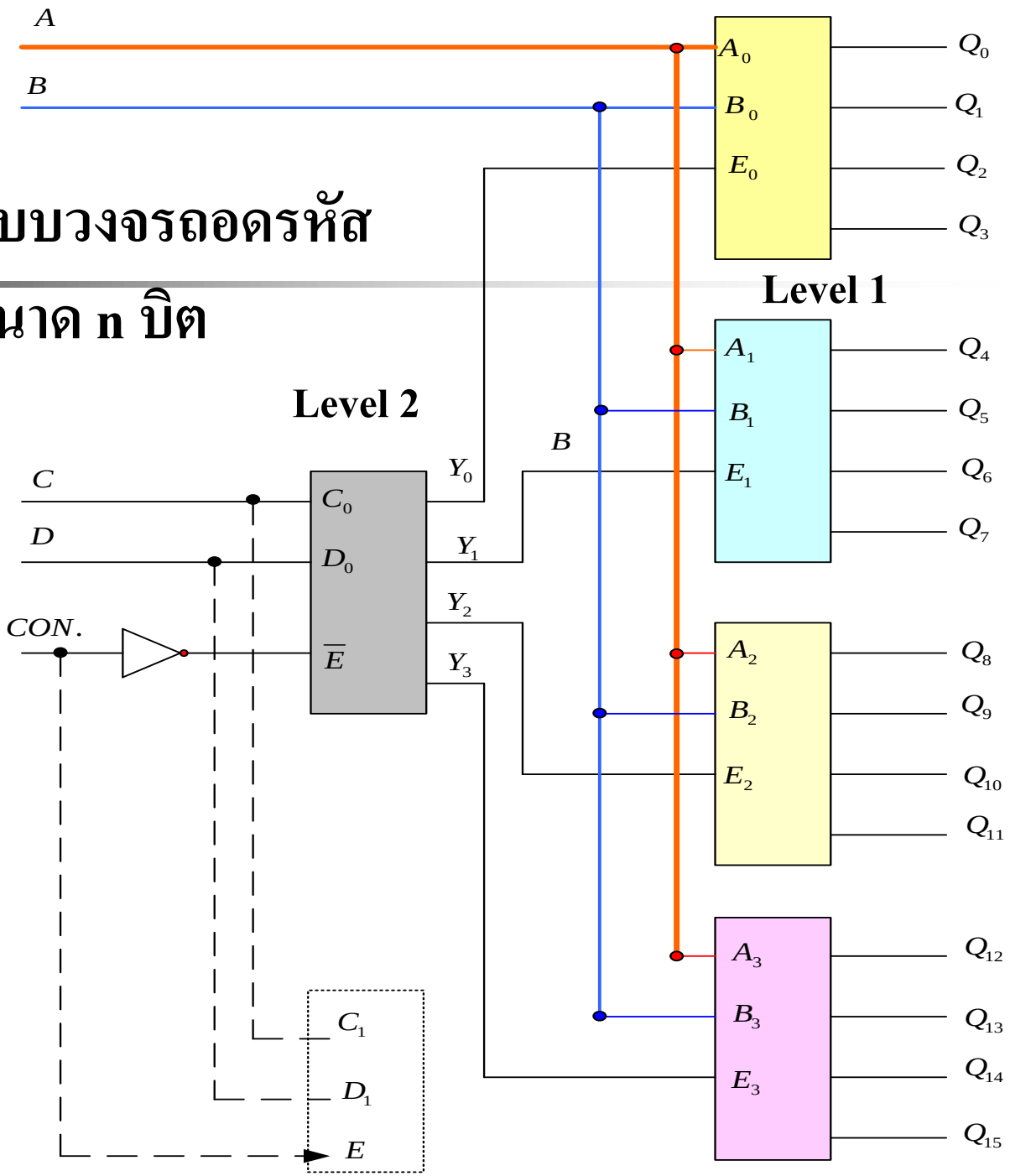
# อธิบาย

- การนำวงจรถอดรหัสขนาด 14 บิต แบบ Dual tree มาใช้งาน
- อินพุตของทางแถวจะแบ่งออกเป็น 7 บิต $A_7$-$A_{13}$ แบ่งออกเป็น 128 แถว
- อินพุตของคอลัมน์จะแบ่งเป็น 7 บิตเช่นกัน $A_0$-$A_6$ แบ่งออกเป็น 128 คอลัมน์
- เอาต์พุตเป็นขนาด 128 * 128 Cell ที่ไม่ซ้ำกัน
- จากตัวอย่างการเลือกตำแหน่งของเซลล์ จะเลือกจากแถวคูณกับคอลัมน์ ในที่นี้จะได้แถวที่3 คูณกับคอลัมน์ 2

# 445 การออกแบบวงจรถอดรหัสเลขฐานสองขนาด n บิต

**Decoder**

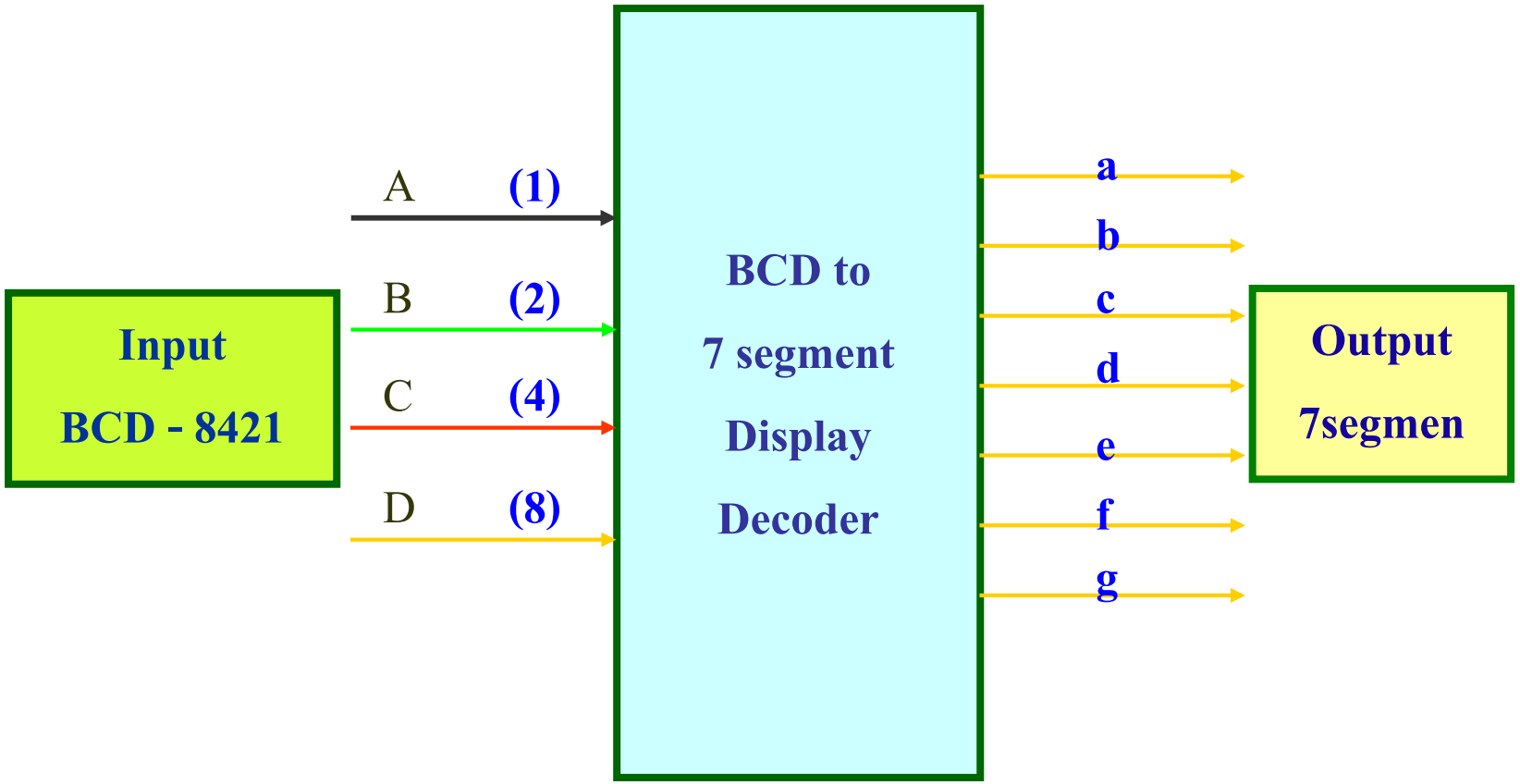
**nBit**



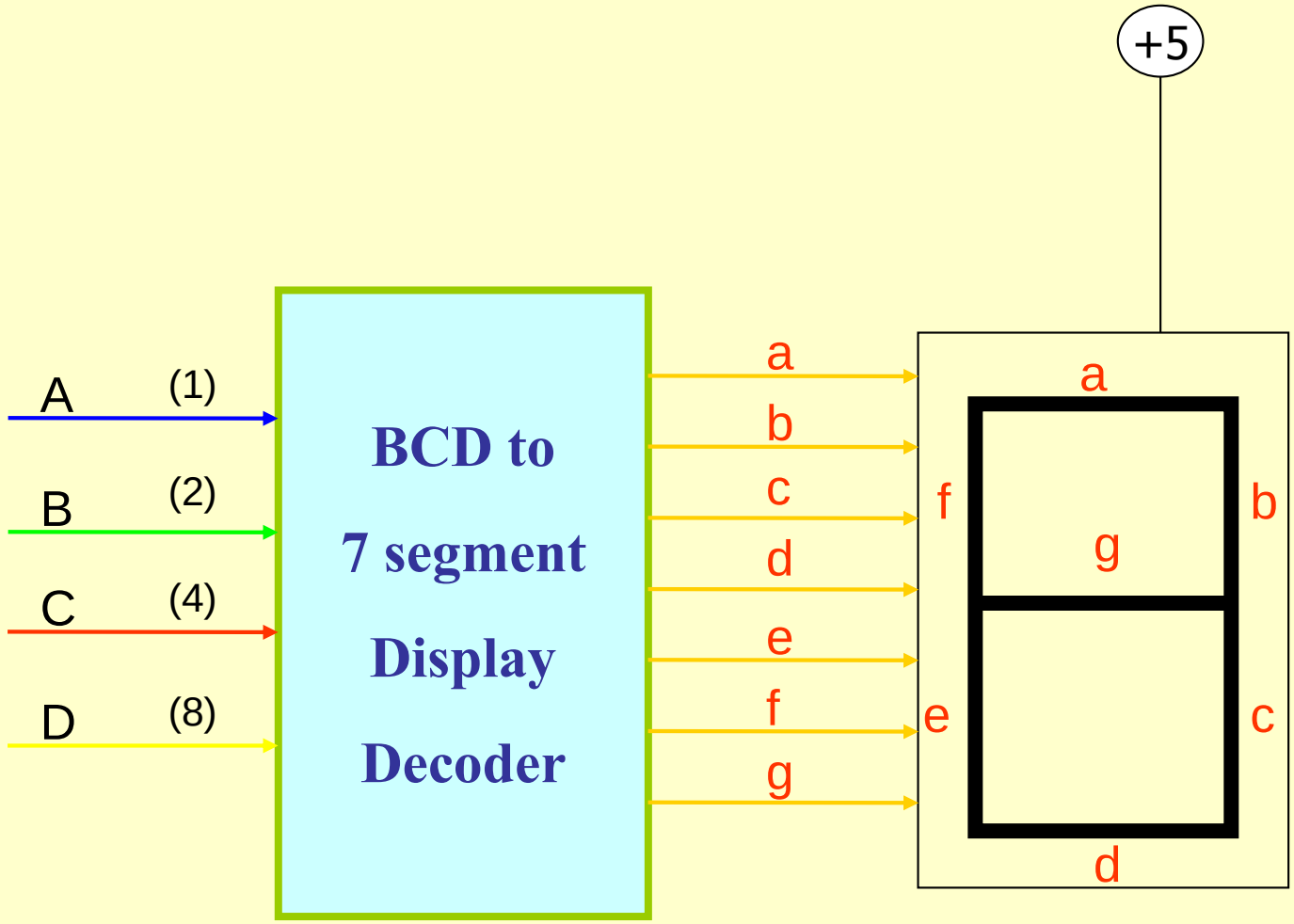
# อธิบาย

- การออกแบบวงจรถอดรหัสฐานสองขนาด n บิต
- วิธีการนำวงจรถอดรหัสขนาด 2 บิต มาต่อกันแบบขนาน 2 ระดับ
- นำค่าบิต $M_{SB}$ มาควบคุมการทำงานของวงจรถอดรหัสในระดับต่างๆ
- จากการออกแบบทางอินพุตจำนวน 5 บิต A-E
- นำค่าบิต $E_1$ บิตมาควบคุมสัญญาณEnable ของวงจรถอดรหัสระดับ 2 สามารถควบคุมได้ถึง 32 ตำแหน่ง
- สัญญาณที่เอาต์พุตของแต่ละตำแหน่งเกิดจากบิต A-B สามารถควบคุมได้ 4 ตำแหน่ง
- โดยมีบิตC-DควบคุมสัญญาณEnable ระดับ 2 ได้ถึง 4 ตำแหน่ง
- การขยายบิต $M_{SB}$ ออกไป ก็จะควบคุมในระดับ 3-n ต่อไป

# 45 การออกแบบวงจรถอดรหัส

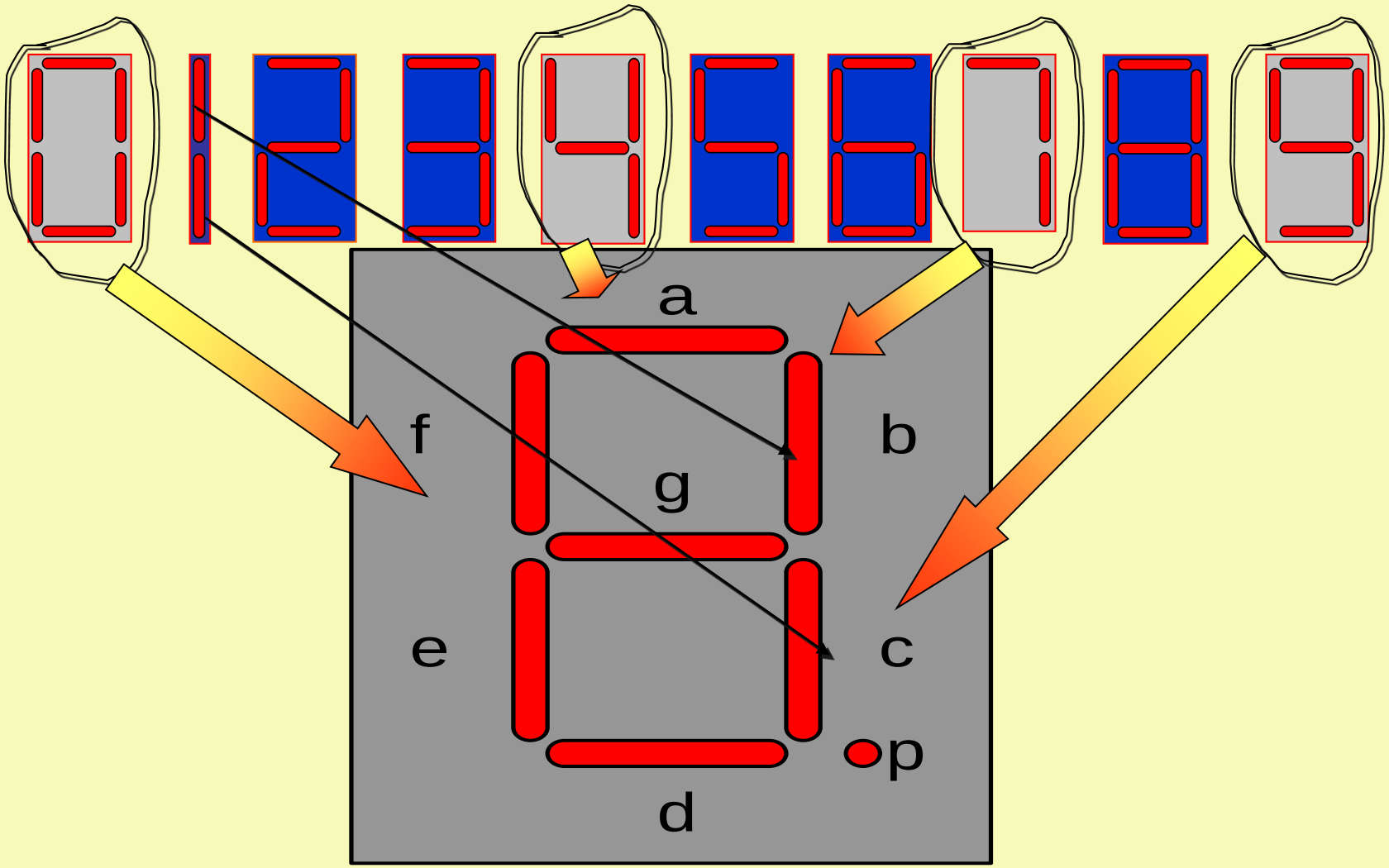



**การออกแบบวงจรถอดรหัส BCD-8421 แสดงผลแบบใช้แอลอีดี (LED) 7 Segment**


+5
A
(1)
B
(2)
C
(4)
D
(8)
BCD to
7 segment
Display
Decoder
a
b
c
d
e
f
g
a
f
b
g
e
c
d

# วงจรถอดรหัส LED 7 ส่วน

แอลอีดี 7 ส่วนประกอบด้วยแอลอีดี 7 ตัวซึ่งทำเป็นเลขอาราบิคจาก0-9 ได้แอลอีดีแต่ละส่วนจะใช้อักษร a-g เรียกชื่อเป็นตำแหน่งแทน



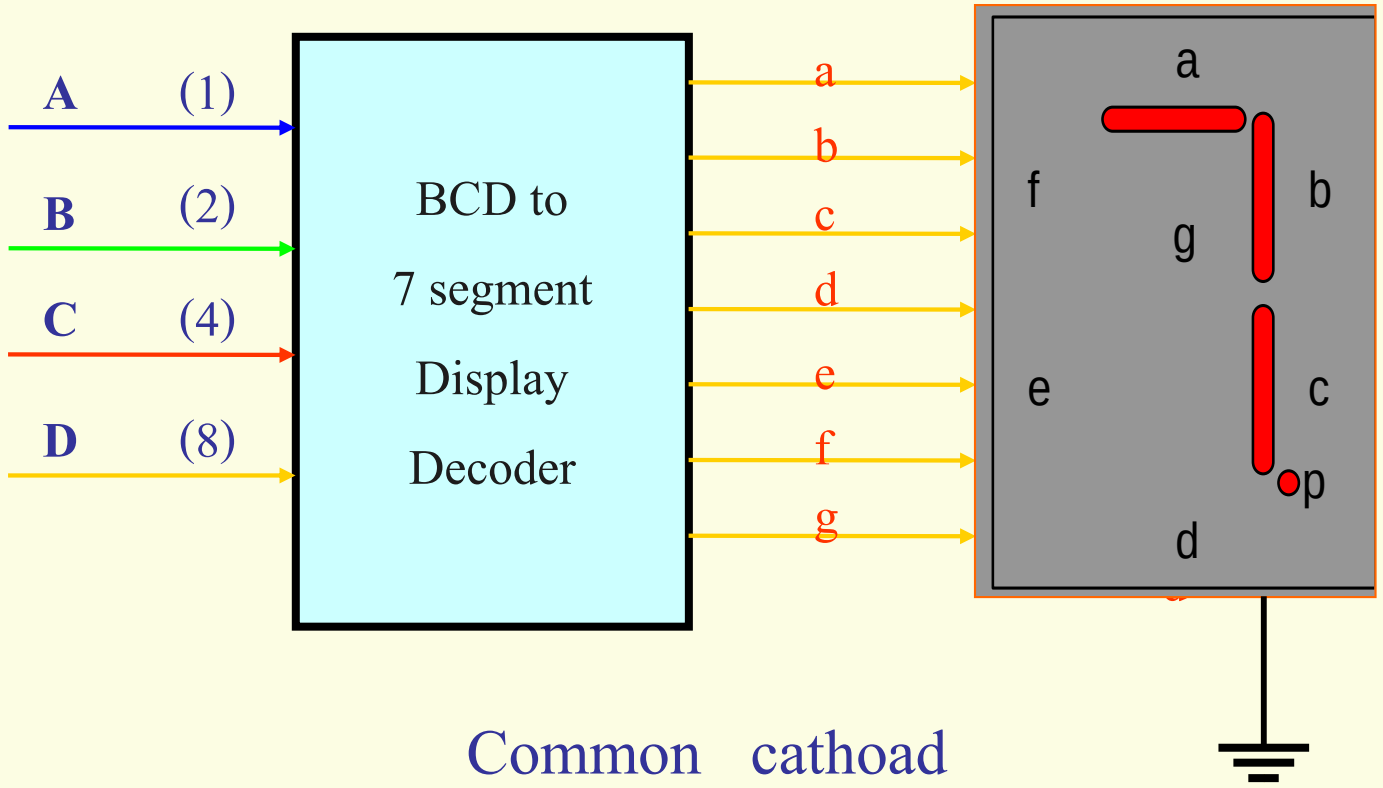
# อธิบาย

- การออกแบบวงจรถอดรหัส BCD เป็นเลขฐานสิบที่สามารถแสดงผลในรูปไดโอดเปล่งแสง
- สัญญาณอินพุตในรหัสBCD โดยมี A,B,C,D เป็นอินพุต
- วงจร BCD To Seven segment Display Decorder จะทำหน้าที่แปลงรหัส BCD ออกเป็นเอาต์พุต 7 ส่วน เพื่อนำไปขับ LED ให้ทำงานตามที่สัญญาณอินพุตป้อนเข้ามา
- จากรูปตัวอย่าง Seven-Segment เป็นไดโอดเปล่งแสงที่นำมาต่อกันในรูปตัวเลขอาราบิค สามารถแสดงผลเป็นเลข 0-9 แทน LED แต่ละส่วน คือ a,b,c,d,e,f,g
- เอาต์พุตของเลขที่แสดง 0 จะมีเอาต์พุต ที่ a-f โดยเว้น g
- เอาต์พุตของเลขที่แสดง 1 จะมีเอาต์พุต ที่ b,c
- เอาต์พุตของเลขที่แสดง 4 จะมีเอาต์พุต ที่ b,c,f,g
- เอาต์พุตของเลขที่แสดง 7 จะมีเอาต์พุต ที่ a,b,c
- เอาต์พุตของเลขที่แสดง 9 จะมีเอาต์พุต ที่ a-f โดยเว้น e

## วงจรถอดรหัสที่ใช้ Seven Segment สามารถออกแบบดังนี้

- แบบ Common cathode รูปแบบใช้ Seven Segment ที่มีคาโถดต่อร่วมกัน แล้วจุดร่วมไปต่อกับศักย์ที่ต่ำกว่าในทางปฏิบัตินำไปต่อกับ เช่น ground
- แบบ Common Anode รูปแบบใช้ Seven Segment ที่มีอาโนดต่อร่วมกัน แล้วนำจุดร่วมไปต่อกับศักย์ที่สูงกว่าในทางปฏิบัตินำไปต่อกับแหล่งจ่ายไฟบวก

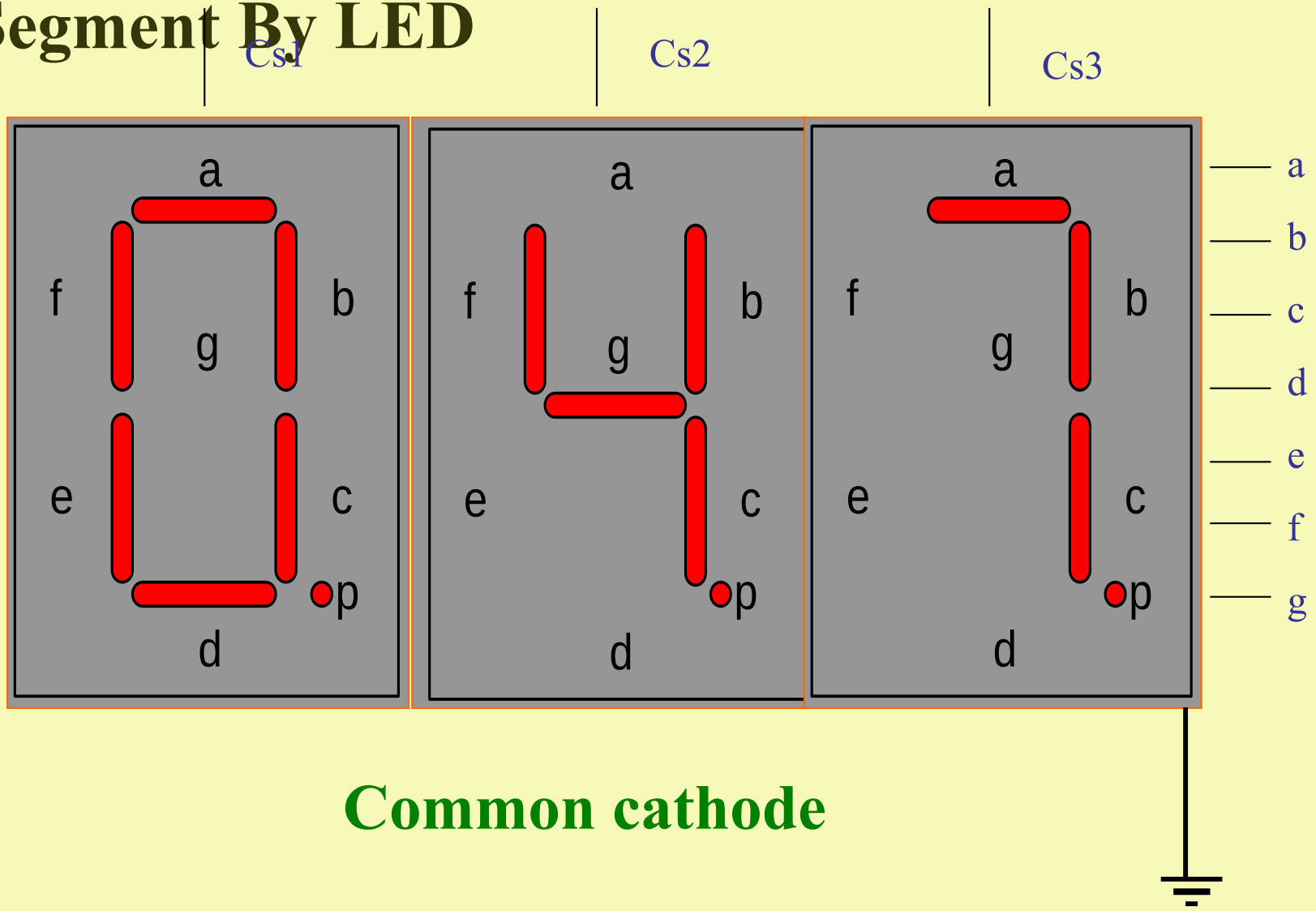
# 451 การออกแบบวงจรถอดรหัส BCD-8421 แสดงผลแบบใช้แอลอีดี 7 ส่วนมีคาโถดต่อร่วมกัน



## ตารางแสดงเอาต์พุตแบบคอมมอนคาโถด

| D | C | B | A | เลขฐาน 10 | g | f | e | d | c | b | a |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 3 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 4 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 5 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 6 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 7 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 9 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

# การใช้งานของ Seven Segment By LED



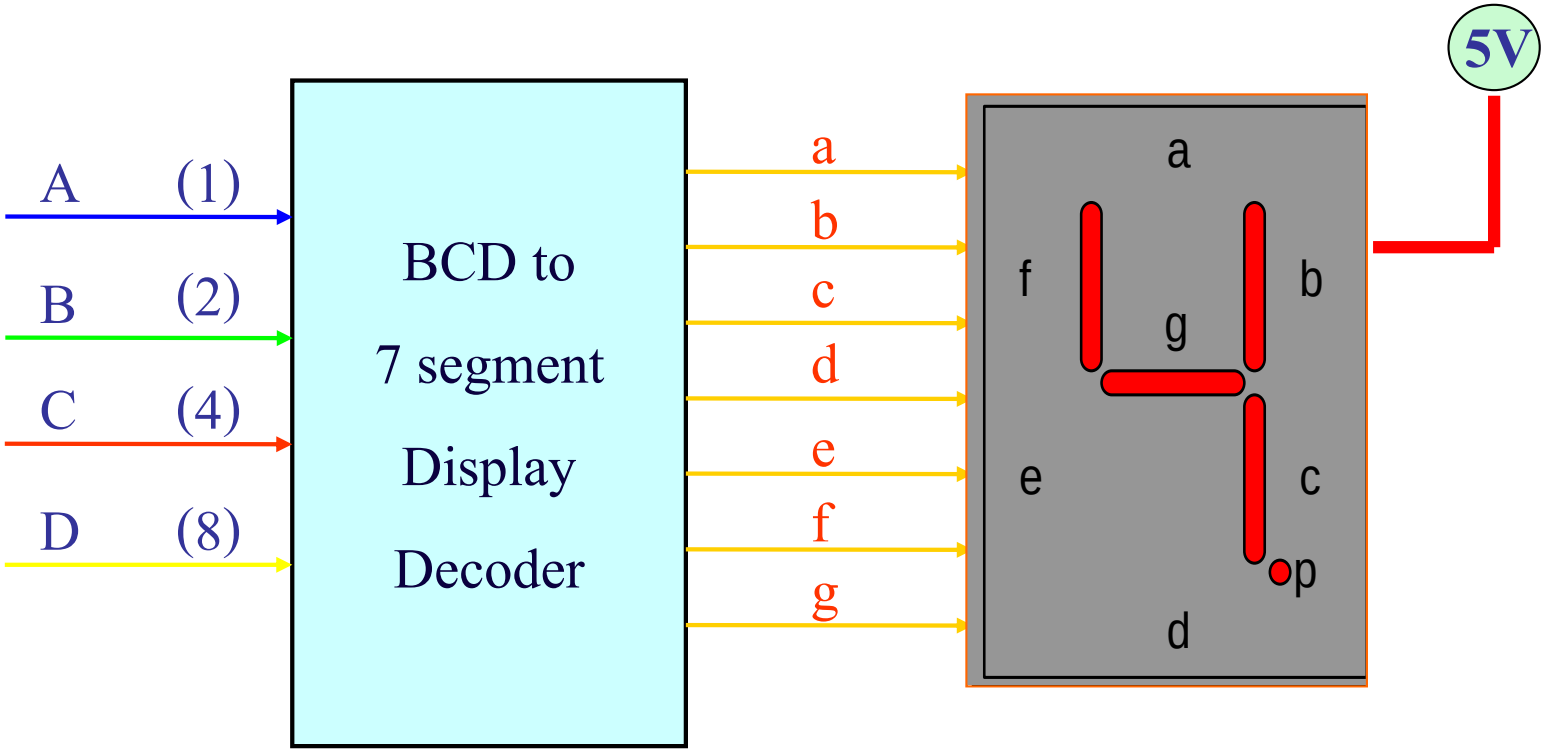

**Common cathode**

# อธิบาย

- แบบ Common cathode รูปแบบใช้ Seven Segment ที่มีคาโถดต่อร่วมกัน แล้วจุดร่วมไปต่อกับศักย์ที่ต่ำกว่าในทางปฏิบัตินำไปต่อกับ ground
- สัญญาณอินพุตเป็นรหัส BCD ผ่านวงจรถอดรหัส ได้เอาต์พุตขนาด 7 บิต ให้ค่าระดับสัญญาณเป็น+5 Volts ดังตารางการถอดรหัส
- เลขแสดง 047 หมายความว่าช่วงเวลาที่ 1 แสดงเลข 0 , ช่วงเวลาที่ 2 แสดงเลข 4 , ช่วงเวลาที่ 3 แสดงเลข 7
- อินพุตในช่วงเวลาที่ 1 D, C, B, A เป็น (0000) ช่วงเวลาที่ 2 เป็น (0100) ช่วงเวลาที่ 3 เป็น (0111)
- สัญญาณเอาต์พุตของ g,f,e,d,c,b,a ช่วงเวลาที่1 เป็น (0111111) ช่วงเวลาที่ 2 เป็น (1100110) ช่วงเวลาที่3 เป็น (0000111)
- แต่ละช่วงเวลาของการแสดงผล จะสัมพันธ์กับการสวิทซ์ขา (Cs) เพื่อเลือก Seven Segment ทำงาน ซึ่งหลักการนี้มีการใช้ร่วมกับวงจรมัลติเพล็กซ์

# 452 การออกแบบวงจรถอดรหัส BCD-8421

## แสดงผลแบบใช้แอลอีดี 7 ส่วนมีอาโนดต่อร่วมกัน

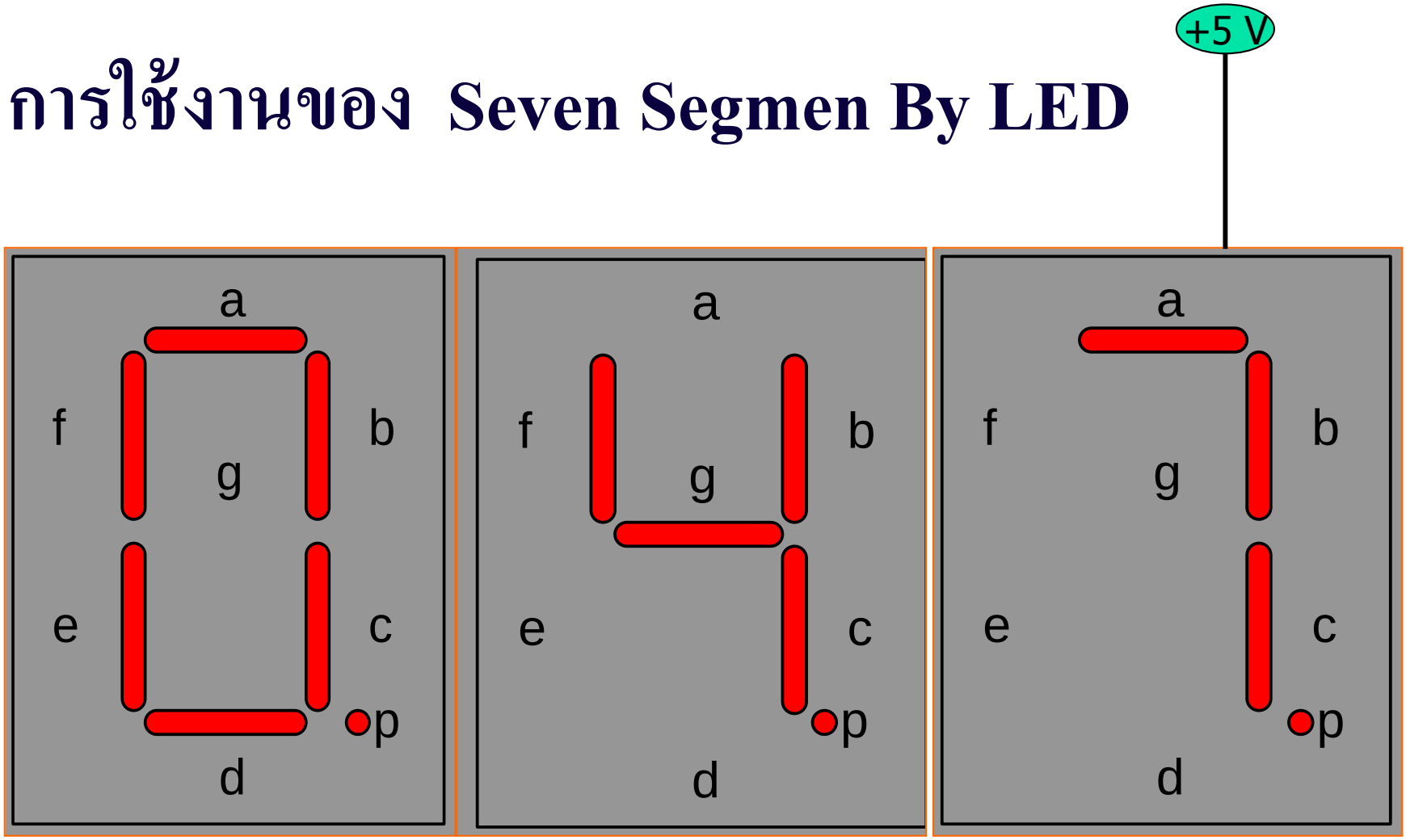



Common anode

# ตารางแสดงเอาต์พุตแบบคอมมอนอาโนด

| D | C | B | A | เลขฐาน 10 | g | f | e | d | c | b | a |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 5 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 7 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 9 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

# การใช้งานของ Seven Segmen By LED



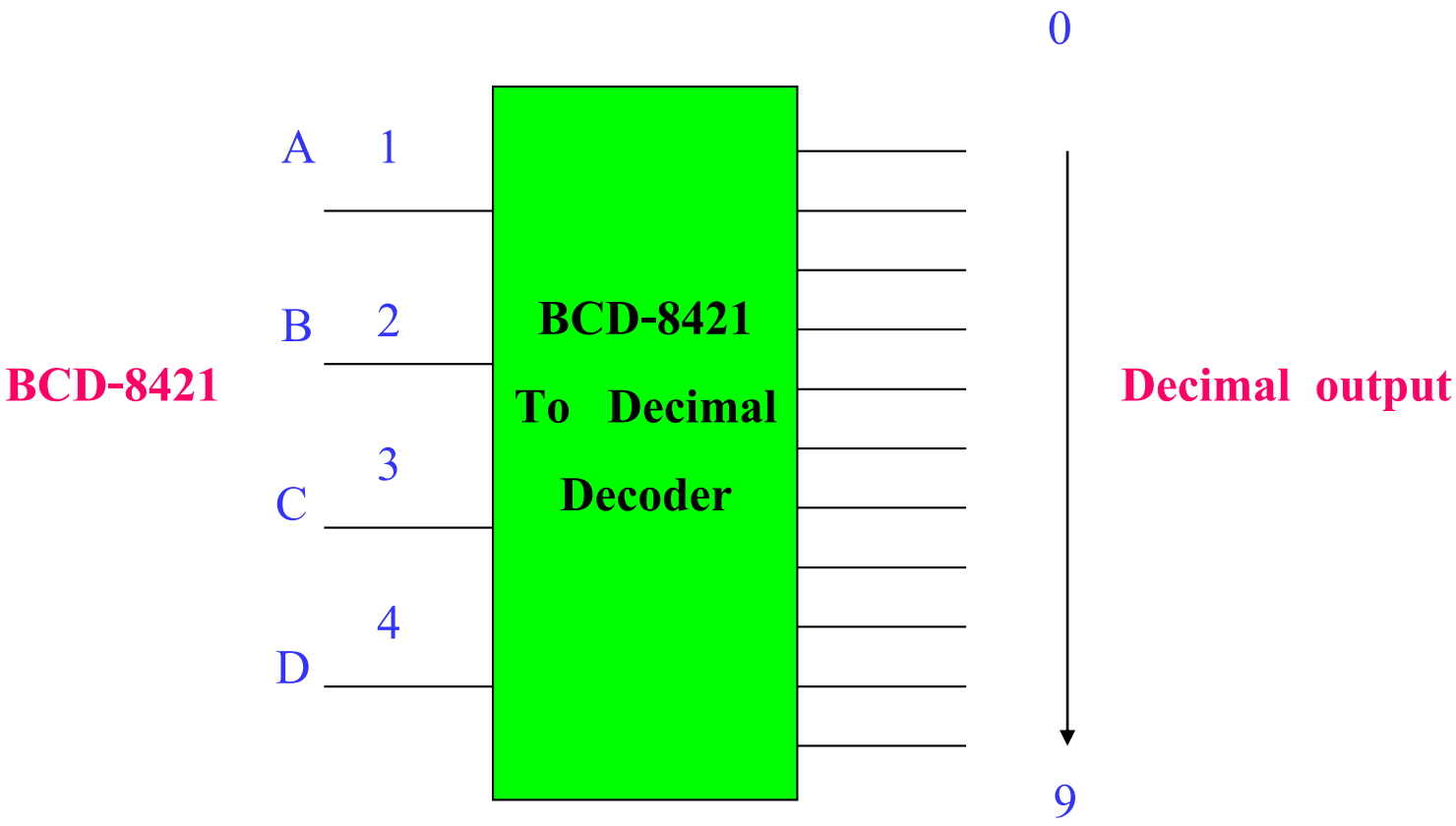

**Common anode**

# อธิบาย

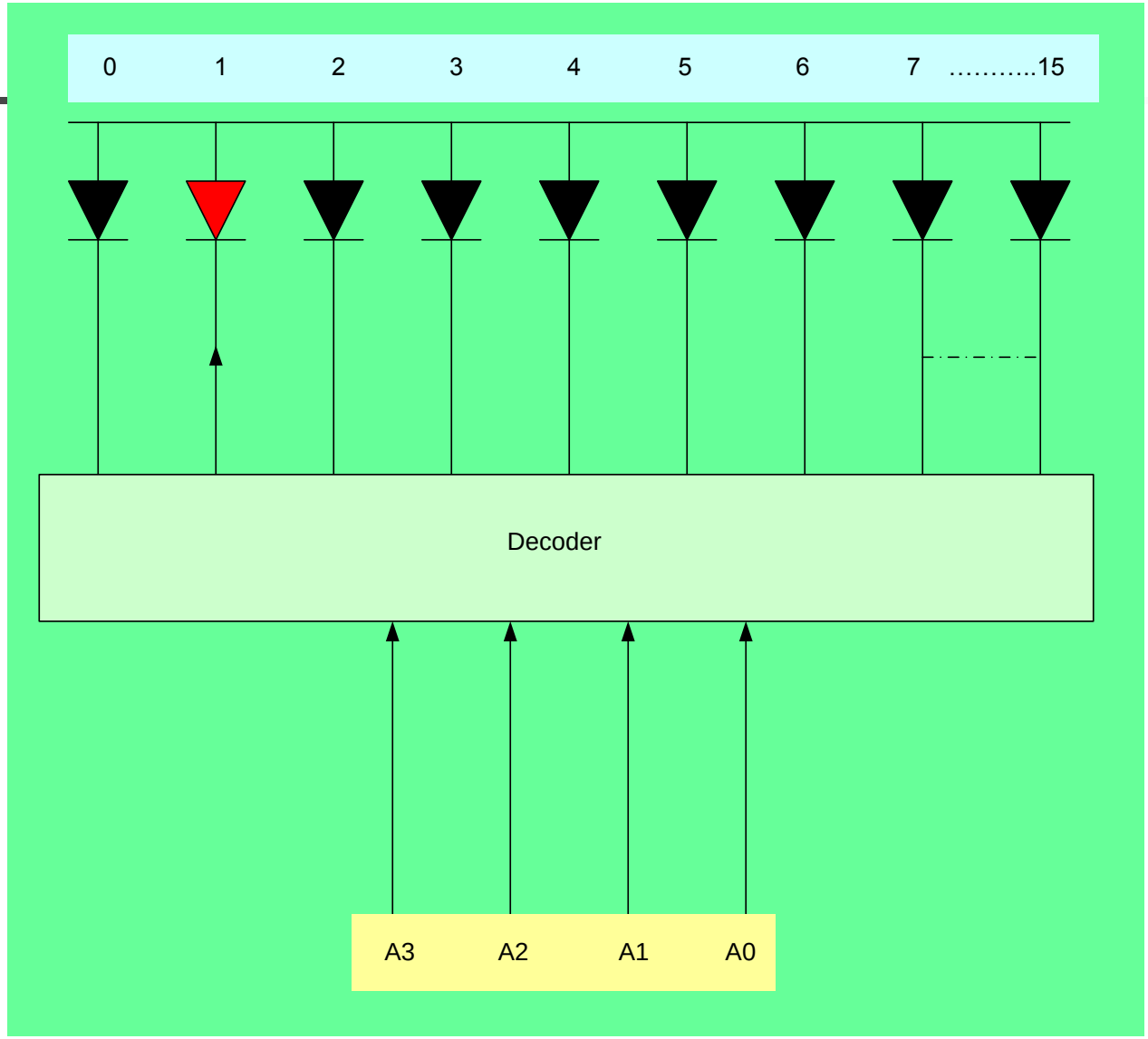
- แบบ Common Anode รูปแบบใช้ Seven Segment ที่มีแอโนดต่อร่วมกัน แล้วนำจุดร่วมไปต่อกับศักย์ที่สูงกว่าในทางปฏิบัตินำไปต่อกับแหล่งจ่ายไฟบวก
- สัญญาณอินพุตเป็นรหัส BCD ผ่านวงจรถอดรหัส ได้เอาต์พุตขนาด 7 บิต ให้ค่าระดับสัญญาณเป็น 0 Volts หรือต่อลงกราวนด์ ดังตารางการถอดรหัส
- เลขแสดง 047 หมายความว่าช่วงเวลาที่ 1 แสดงเลข 0 , ช่วงเวลาที่ 2 แสดงเลข 4, ช่วงเวลาที่ 3 แสดงเลข 7
- อินพุตในช่วงเวลาที่1 D, C, B, A เป็น (0000) ช่วงเวลาที่ 2 เป็น(0100) ช่วงเวลาที่ 3 เป็น (0111)
- สัญญาณเอาต์พุตของ g,f,e,d,c,b,a ช่วงเวลาที่ 1 เป็น (1000000) ช่วงเวลาที่ 2 เป็น (0011001) ช่วงเวลาที่3 เป็น (1111000)

# 453 การออกแบบวงจรถอดรหัสแบบแสดงผลเอาต์พุตเดียว

## Diode matrix สำหรับวงจรถอดรหัสจาก BCD – 8421 เป็น Decimal number

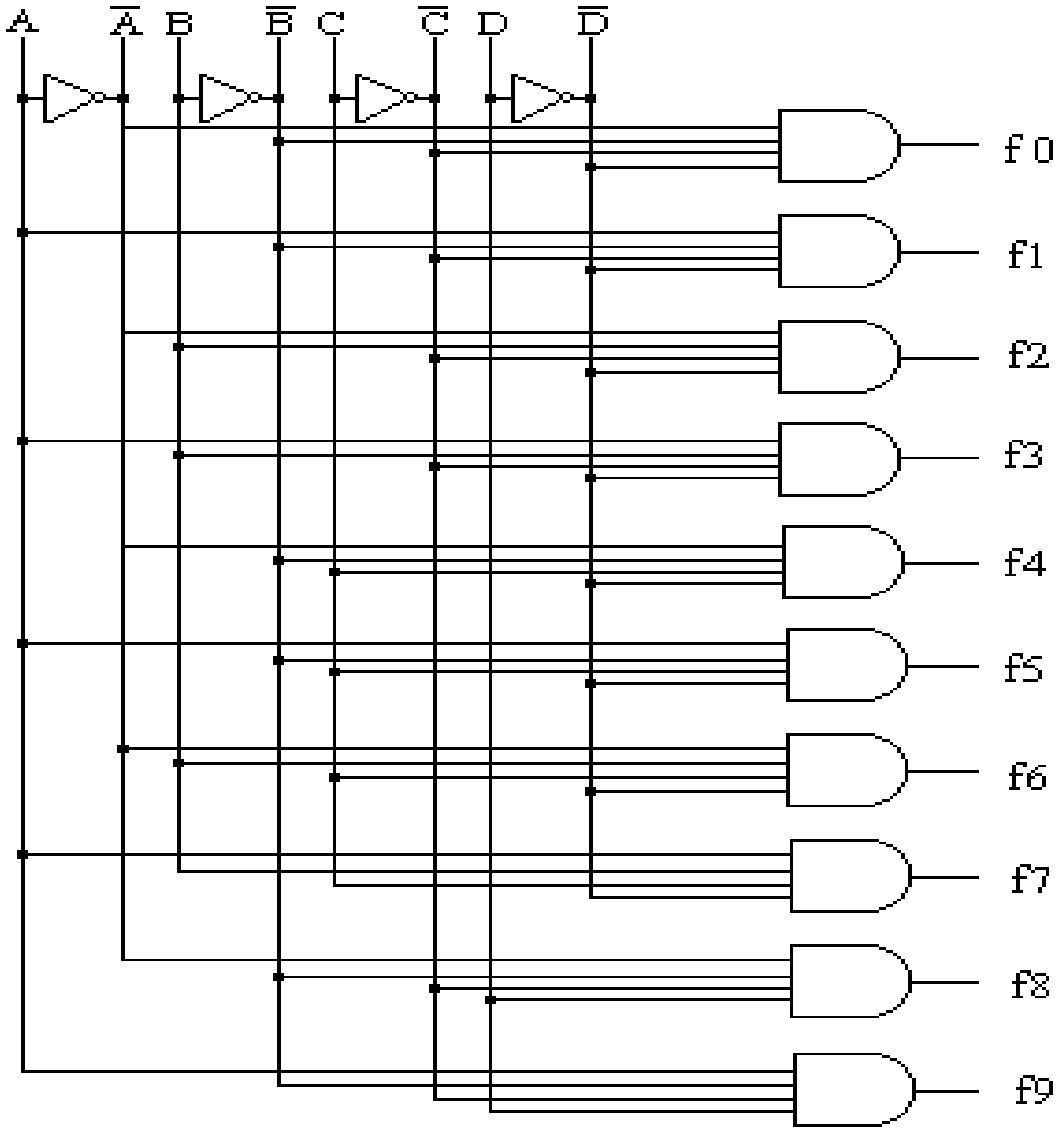


# แสดงเอาต์พุตถอดรหัส 10 ค่า



| Input (BCD-8421) | | | | Output ( Decimal number) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

# วงจรถอดรหัสแบบเอาต์พุตแสดงตำแหน่งของเลขฐานสิบโดยใช้ไดโอดเปล่งแสงแสดงค่า



# อธิบาย

- การออกแบบวงจรถอดรหัสแบบให้ค่าเอาต์พุตเป็นเลขฐานสิบ ที่มีค่าเอาต์พุต 0-9 เอาต์พุตแบบนี้มักแสดงโดยไดโอดเปล่งแสงจำนวน 9 ตัวต่อกับแหล่งจ่ายไฟบวก หรือคอมมอนอาโนด ดังรูปแสดงในการแสดงการทำงานของระดับ 9ตำแหน่ง
- ตารางความจริงสามารถกำหนดค่าอินพุตเป็น A, B, C, D และค่าเอาต์พุตเป็น$F_0$, $F_1$, $F_2$, $F_3$, $F_4$, $F_5$, $F_6$, $F_7$, $F_8$, $F_9$ การทำงานของตารางความจริงเอาต์พุตจะได้เป็น1 เพียงเอาต์พุตเดียว เมื่อสัญญาณทางอินพุตป้อนเข้าตามตารางความจริง
- ตัวอย่าง อินพุตป้อนเข้ามาเป็น 0101 มีค่าเท่ากับ 5 เอาต์พุตที่ตำแหน่ง F5 จะมีค่าเท่ากับ 1 เอาต์พุตอื่นจะมีค่าเท่ากับ0 หมด
- จากตารางความจริงสามารถนำค่า minterm ของแต่ละเอาต์พุต มาเขียนเป็นวงจรเกตได้ ดังรูป$F_0 = \bar{A}\bar{B}\bar{C}\bar{D}$ , $F_1 = \bar{A}\bar{B}\bar{C}D$ , $F_2 = \bar{A}\bar{B}C\bar{D}$, $F_3 = \bar{A}\bar{B}CD$, $F_4 = \bar{A}B\bar{C}\bar{D}$…$F_5 = \bar{A}B\bar{C}D$

# 454 การออกแบบวงจรถอดรหัส BCD เป็นรหัสเกร์ย

ตัวอย่างต่อไปเป็นตารางถอดรหัสจาก BCD เป็นรหัส GRAY ค่าเอาต์พุต จะนำไปออกแบบวงจรเกตโดยพิจารณาทีละขั้นตอนดังตาราง k-map

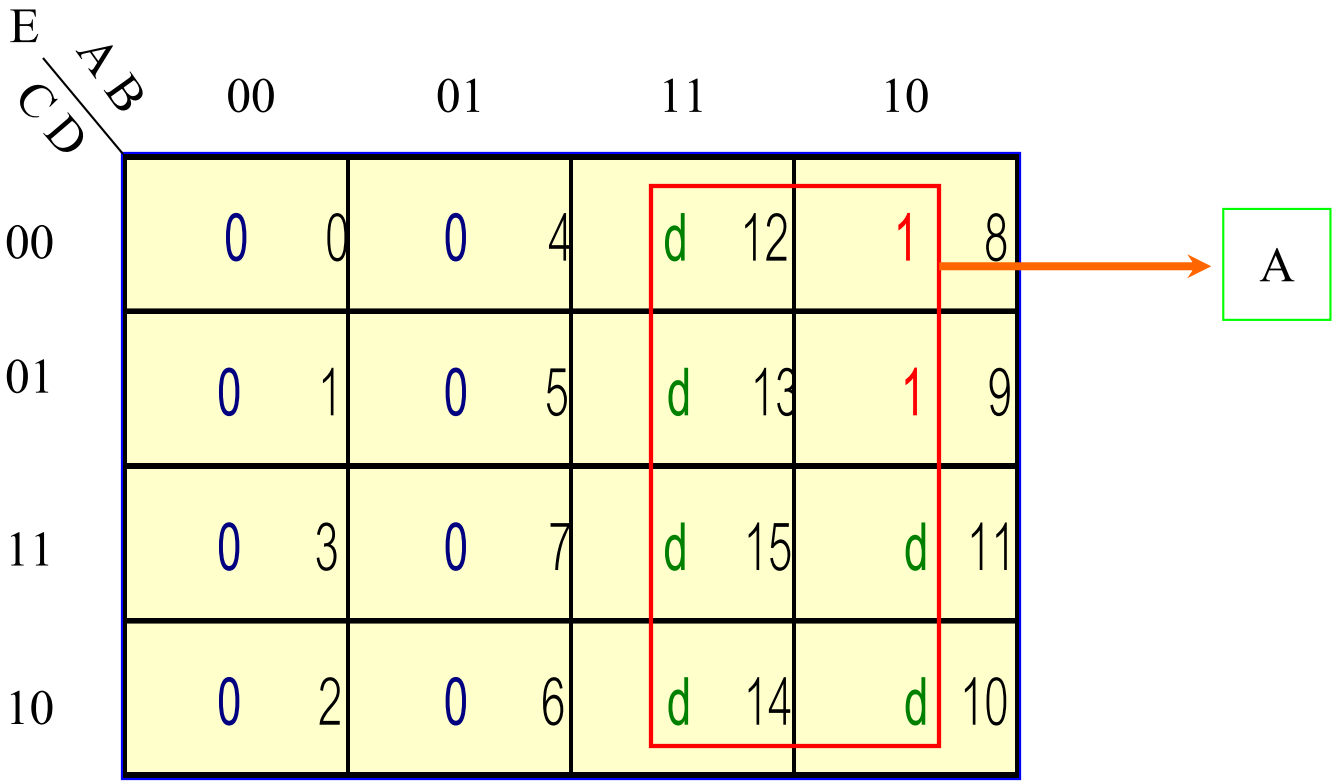
| รหัส BCD | | | | รหัส Gray | | | |
|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H |
| *0* | *0* | *0* | *0* | 0 | 0 | 0 | 0 |
| *0* | *0* | *0* | *1* | 0 | 0 | 0 | 1 |
| *0* | *0* | *1* | *0* | 0 | 0 | 1 | 1 |
| *0* | *0* | *1* | *1* | 0 | 0 | 1 | 0 |
| *0* | *1* | *0* | *0* | 0 | 1 | 1 | 0 |
| *0* | *1* | *0* | *1* | 0 | 1 | 1 | 1 |
| *0* | *1* | *1* | *0* | 0 | 1 | 0 | 1 |
| *0* | *1* | *1* | *1* | 0 | 1 | 0 | 0 |
| *1* | *0* | *0* | *0* | 1 | 1 | 0 | 0 |
| *1* | *0* | *0* | *1* | 1 | 1 | 0 | 1 |

# อธิบาย

- การออกแบบวงจรถอดรหัส BCD เป็นรหัสเกร์ย
- จากสัปดาห์ที่ 5 นำค่าที่ได้จากการแปลงรหัสBCD เป็นรหัสเกร์ย มาเขียนตารางความจริง
- ค่าทางอินพุตของรหัส BCD คือ A,B,C,D ซึ่งจะมีค่าเป็น 0-9 เกินจากค่า 9-15 ให้มีค่าเป็นอะไรก็ได้ (Don't care) คือเป็น 0 หรือ 1 ก็ได้
- ค่าทางเอาต์พุตคือค่ารหัส เกร์ย แยกเป็นเอาต์พุต 4ค่า คือ E,F,G,H
- พิจารณาเอาต์พุต E เพียงเอาต์พุตเดียว นำค่าที่ได้ไปเขียนลงในตาราง K-map เพื่อทำการลดรูปสมการ ดังที่กล่าวมาแล้วในบทที่4
- พิจารณาเอาต์พุต F เพียงเอาต์พุตเดียว นำค่าที่ได้ไปเขียนลงในตาราง K-map เพื่อทำการลดรูปสมการ
- พิจารณาเอาต์พุต G เพียงเอาต์พุตเดียว นำค่าที่ได้ไปเขียนลงในตาราง K-map เพื่อทำการลดรูปสมการ
- พิจารณาเอาต์พุต H เพียงเอาต์พุตเดียว นำค่าที่ได้ไปเขียนลงในตาราง K-map เพื่อทำการลดรูปสมการ

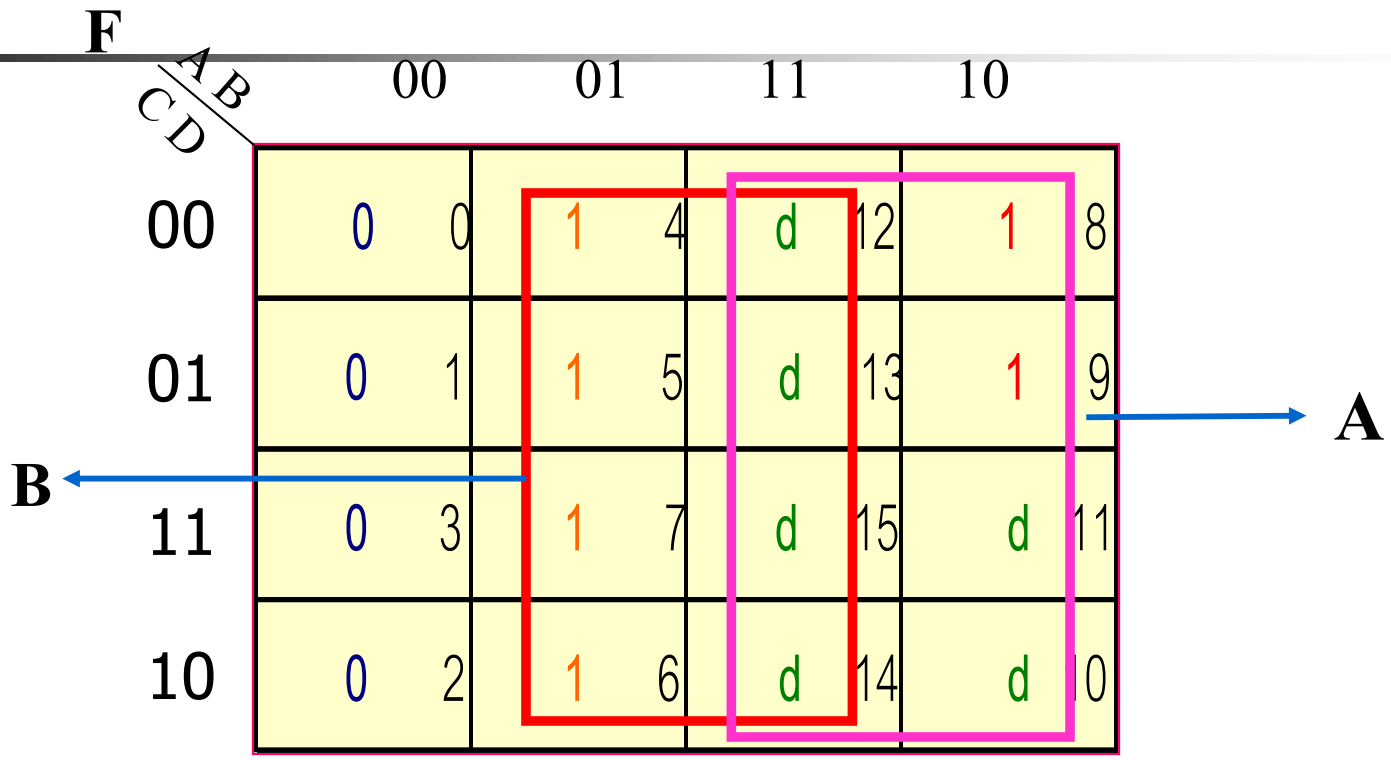
# ในการออกแบบในที่นี้ใช้ Min Term ออกแบบ

เราจะพิจารณาที่เอาต์พุตของฟังก์ชัน E ซึ่งจะเห็นว่ามีสภาวะเป็น ”1” ที่ตำแหน่ง 8 และ 9 จาก 10-15 เป็นค่าอะไรก็ได้จึงทำการจับคู่แล้วลดรูปในที่นี้ทำการจับคู่กับ d ด้วย



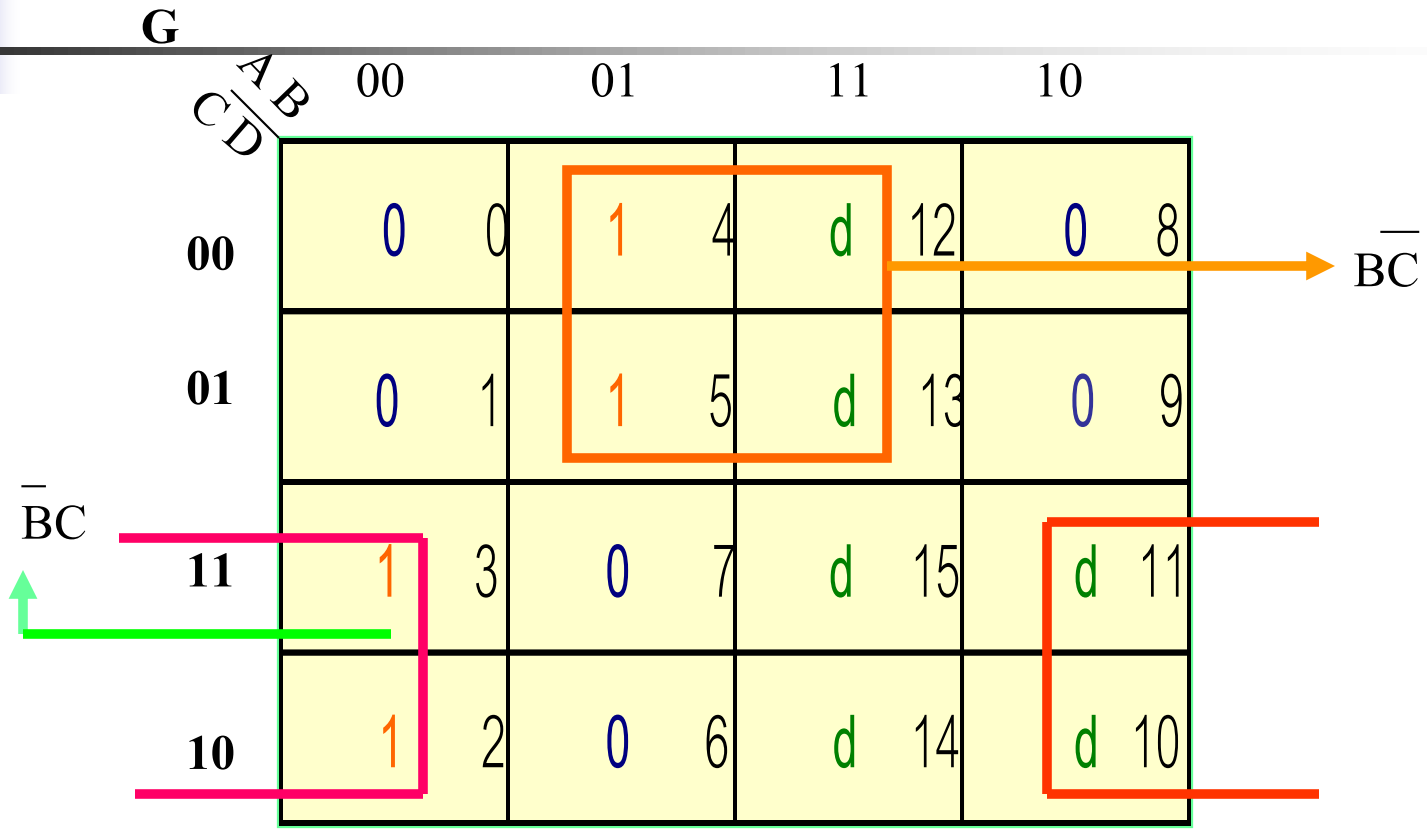

A เป็นฟังก์ชันของ E

พิจารณาที่เอาต์พุตของฟังก์ชัน F ซึ่งจะเห็นว่ามีสภาวะเป็น ”1” ที่ตำแหน่ง 4,5,7,6,8,9 จึงทำการจับคู่แล้วลดรูปในที่นี้ทำการจับคู่กับ d ด้วย



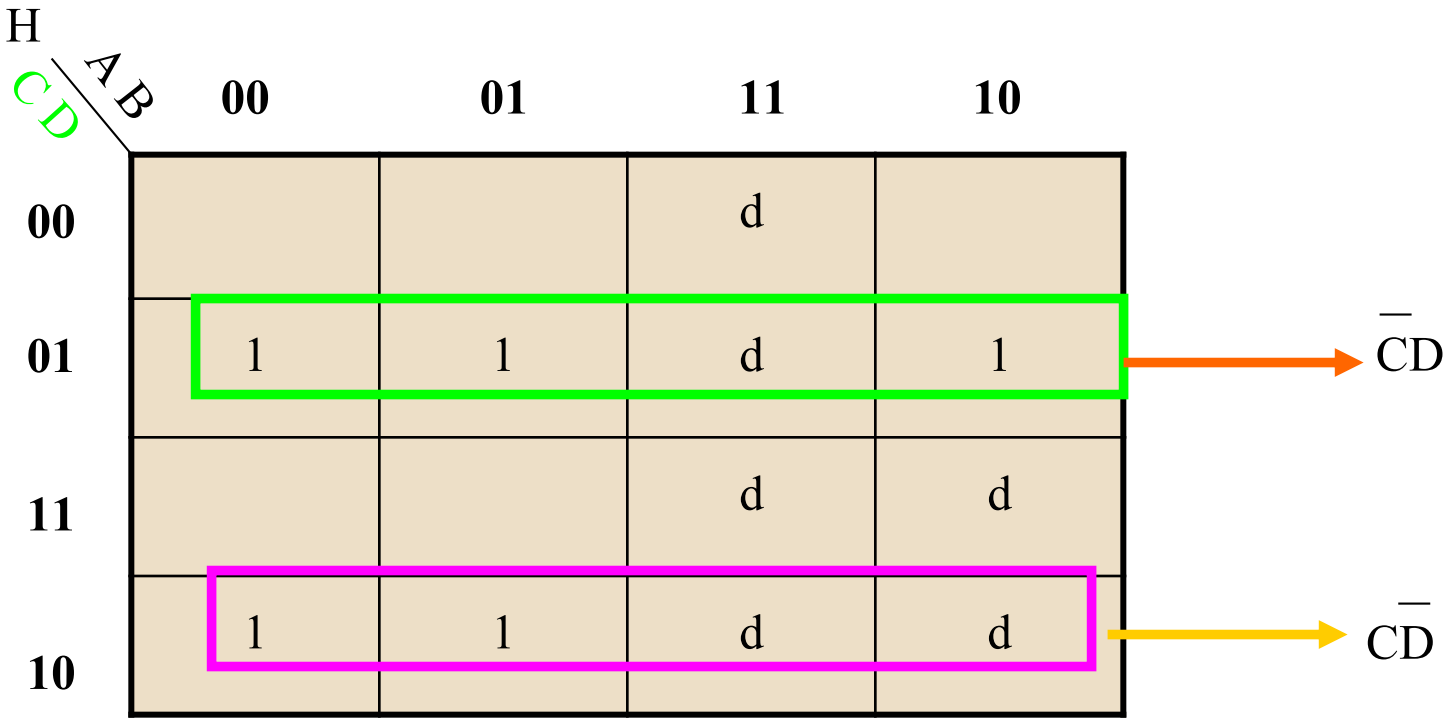

A+B เป็นฟังก์ชันของ F

พิจารณาที่เอาต์พุตของฟังก์ชัน G ซึ่งจะเห็นว่ามีสภาวะเป็น1 ที่ตำแหน่ง 2,3,4,5 จึงทำการจับคู่แล้วลดรูปในที่นี้ทำการจับคู่กับ d ด้วย

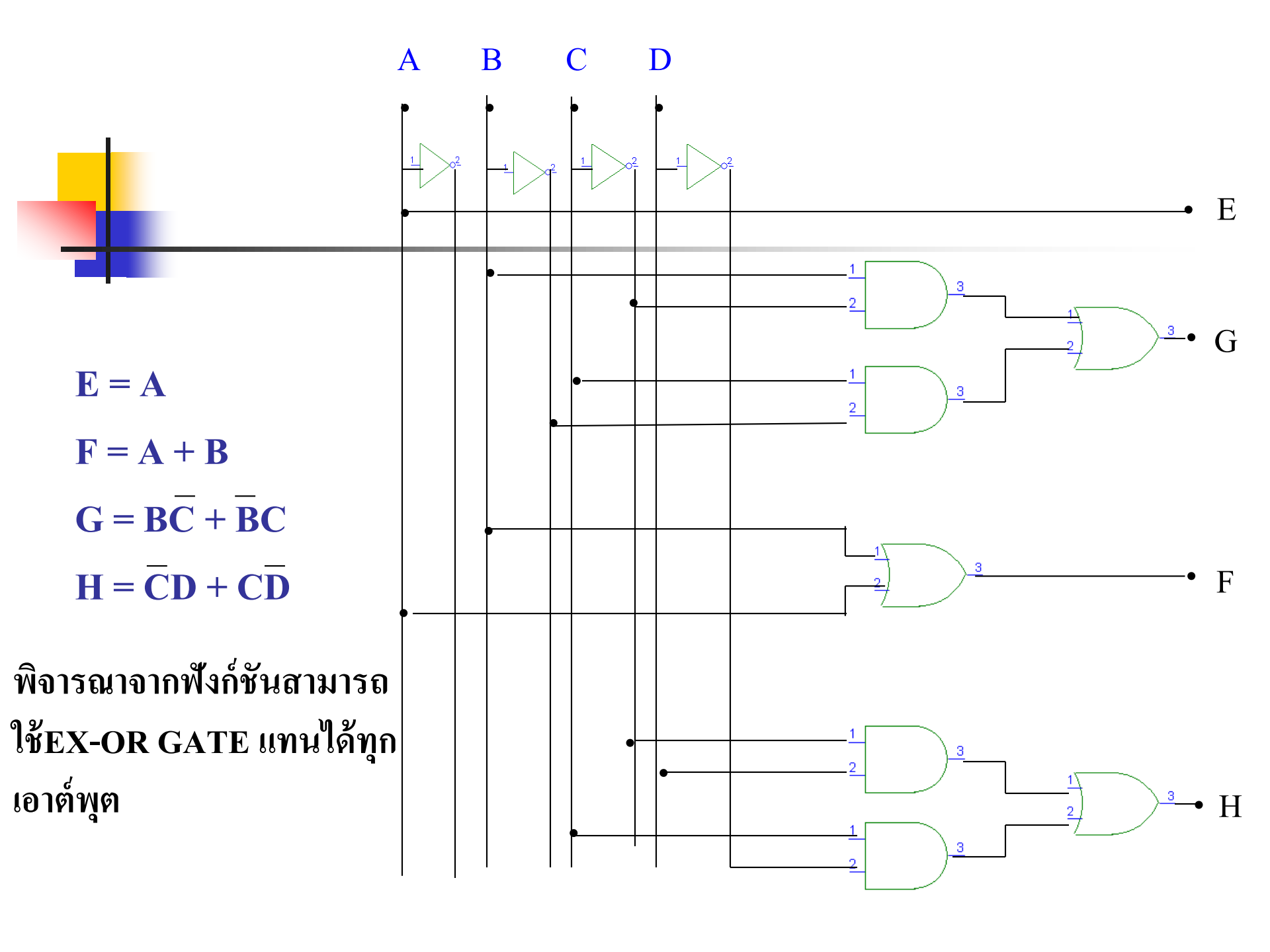



Output G = $B\overline{C}+\overline{B}C$ เป็นฟังก์ชันของ G

# พิจารณาที่ฟังก์ชัน H จะได้




**Output = $\overline{C}D+C\overline{D}$**




$E = A$

$F = A + B$

$G = B\bar{C} + \bar{B}C$

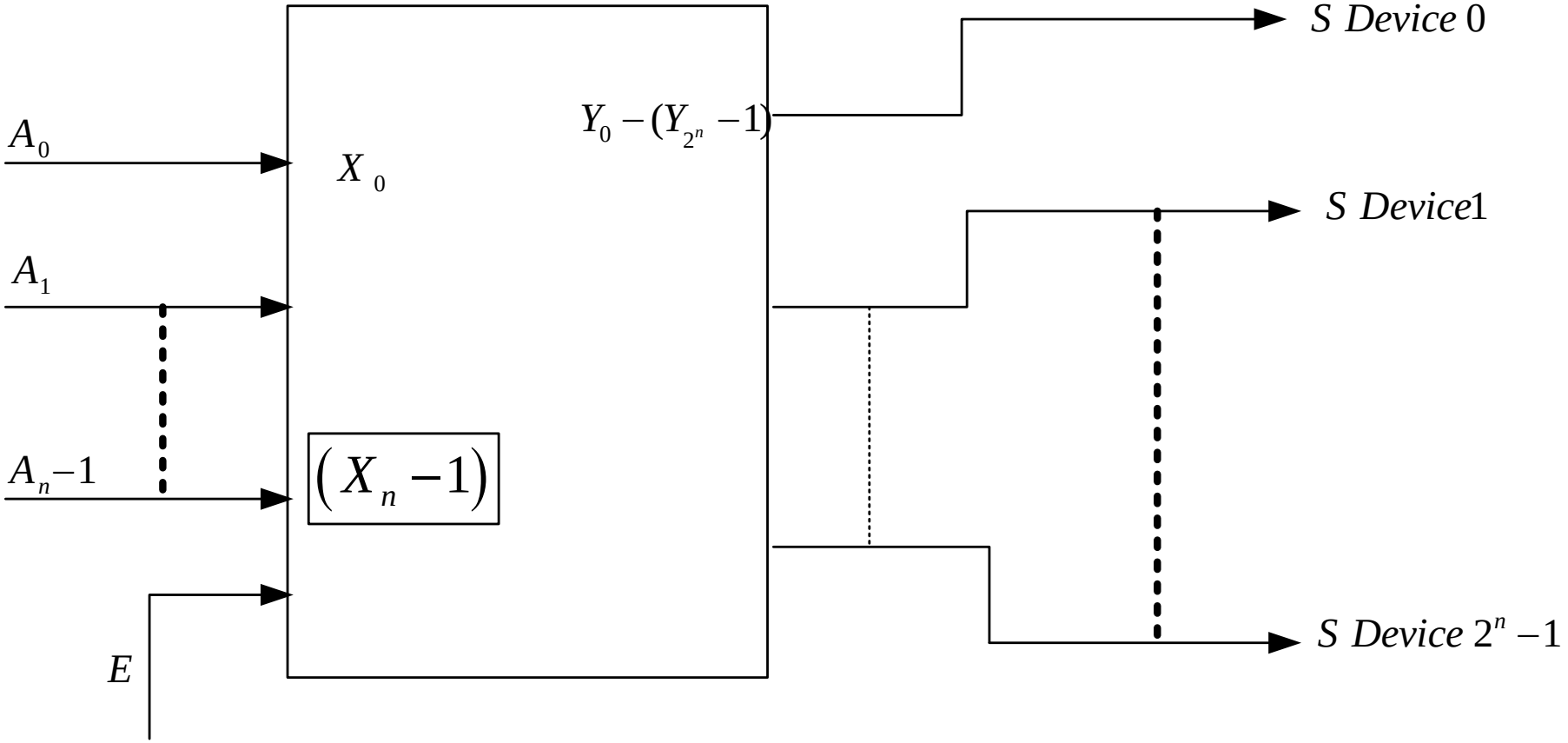
$H = \bar{C}D + C\bar{D}$

พิจารณาจากฟังก์ชันสามารถใช้EX-OR GATE แทนได้ทุกเอาต์พุต

# อธิบาย

- จากการลดรูปสมการ โดย k-map ค่าเอาต์พุตที่ได้ดังนี้
- เอาต์พุต E = A
- เอาต์พุต F เท่ากับนำค่าอินพุต A กับ B มาบวกกัน ไม่คิดตัวทด
- เอาต์พุต G เท่ากับนำค่าอินพุต B กับ C มาบวกกัน ไม่คิดตัวทด
- เอาต์พุต H เท่ากับนำค่าอินพุต C กับ D มาบวกกัน ไม่คิดตัวทด
- จากเอาต์พุต ที่เกิดขึ้น จะเป็นดังรูปวงจรที่ออกแบบในรูปแบบวงจรใช้คอมไบเนชันแบบแอนด์เกตและ ออร์เกต
- อีกวิธีหนึ่งสามารถใช้ EX-OR เกตทำหน้าที่แทนฟังก์ชันการบวกได้เหมือนกัน

# 46การประยุกต์ใช้วงจรถอดรหัสในการออกแบบวงจรลอจิกเกต



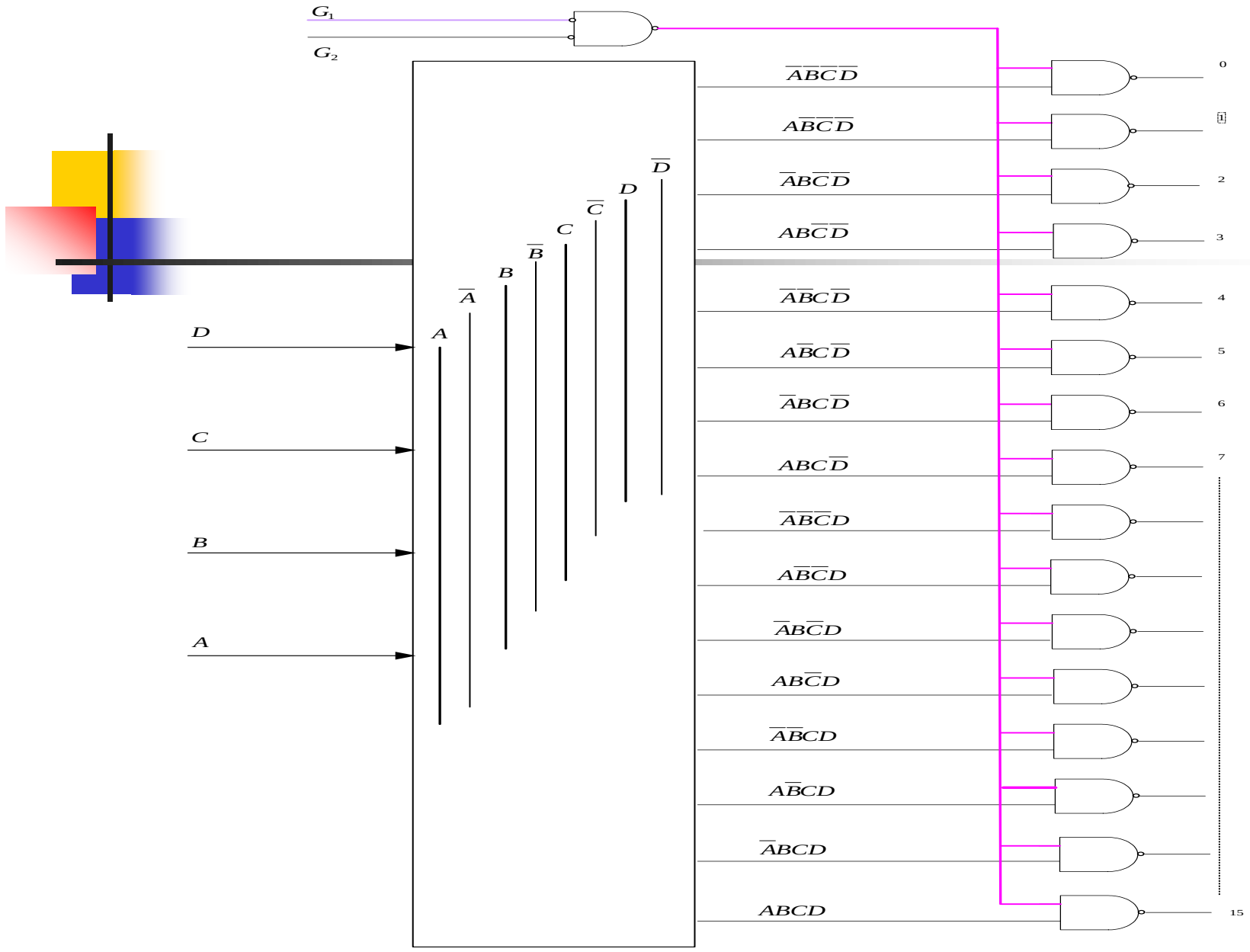
| G1 | G2 | D | C | B | A | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L | L | L | L | L | L | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| L | L | L | L | L | H | H | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| L | L | L | L | H | L | H | H | L | H | H | H | H | H | H | H | H | H | H | H | H | H |
| L | L | L | L | H | H | H | H | H | L | H | H | H | H | H | H | H | H | H | H | H | H |
| L | L | L | H | L | L | H | H | H | H | L | H | H | H | H | H | H | H | H | H | H | H |
| L | L | L | H | L | H | H | H | H | H | H | L | H | H | H | H | H | H | H | H | H | H |
| L | L | L | H | H | L | H | H | H | H | H | H | L | H | H | H | H | H | H | H | H | H |
| L | L | L | H | H | H | H | H | H | H | H | H | H | L | H | H | H | H | H | H | H | H |
| L | L | H | L | L | L | H | H | H | H | H | H | H | H | L | H | H | H | H | H | H | H |
| L | L | H | L | L | H | H | H | H | H | H | H | H | H | H | L | H | H | H | H | H | H |
| L | L | H | L | H | L | H | H | H | H | H | H | H | H | H | H | L | H | H | H | H | H |
| L | L | H | L | H | H | H | H | H | H | H | H | H | H | H | H | H | L | H | H | H | H |
| L | L | H | H | L | L | H | H | H | H | H | H | H | H | H | H | H | H | L | H | H | H |
| L | L | H | H | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H | L | H | H |
| L | L | H | H | H | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | L |
| H | L | d | d | d | d | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| L | H | d | d | d | d | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |
| H | H | d | d | d | d | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H | H |

# อธิบาย

- $A_0$-($A_n$-1) เป็นบิตแอดเดรสเพื่อบอกตำแหน่งทางเอาต์พุต
- E เป็น Device access control signal เป็นตัวเลือก Chip ให้ทำงาน
- เอาต์พุตสามารถควบคุมการทำงานของ Device ได้ถึง n ขึ้นอยู่กับบิตแอดเดรส
- $G_1$,$G_2$ เป็นบิตควบคุมการทำงานของวงจรถอดรหัส

# อธิบาย

- จากตารางแสดงค่าทางเอาต์พุตจะแสดงในรูปแบบของ Maxterm เอาต์พุตจะมีค่าเป็น 0 เมื่อมีรหัสทางอินพุตเข้ามาเป็น (A+B+C+D=0) แสดงในรูปแบบของตารางความจริง
- จากโมดุลวงจรถอดรหัส เพื่อให้เป็นไปตามฟังก์ชัน และเป็นมาตรฐานการใช้งานจริง จึงนำวงจรถอดรหัส เบอร์ 74154 มาอธิบายการทำงานเพื่อให้เข้าใจพิจารณาตารางความจริง
- เอาต์พุตของวงจรถอดรหัส เบอร์ 74154 มีจำนวน 16 เอาต์พุต (0-15)
- จำนวนอินพุตมี 6 อินพุตประกอบด้วย $G_1$,$G_2$, A,B,C,D
- จากตารางความจริงสัญญาณ $G_1$,$G_2$ จะต้องมีค่าเป็น 0 ทั้งคู่ วงจรจึงจะทำงาน
- สัญญาณควบคุมการทำงานของวงจรจะมีค่า 0000-1111 เพื่อเลือกตำแหน่งของเอาต์พุตทั้ง 16 ค่า

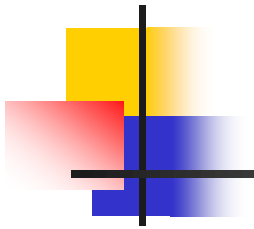



# การนำวงจรถอดรหัสไปออกแบบวงจรลอจิก

# อธิบาย

- จากฟังก์ชันที่กำหนดโดยตารางความจริงสามารถนำมาเขียนวงจรลอจิกเกต ในรูปคอมไบเนชันได้ดังนี้
- ในบล็อกสร้างรหัสที่เอาต์พุต $\bar{A}\bar{B}\bar{C}\bar{D}$ จนถึง ABCD จำนวน 16 ตำแหน่ง ให้ค่าเอาต์พุตส่งไปยังแนนด์เกต
- ในส่วนควบคุมขา $G_1$,$G_2$ จะป้อนอินพุตเป็น 0 ทั้งคู่เพื่อควบคุมการทำงาน ของโมดุลคอมไบเนชันลอจิก
- การทำงานของโมดุลนี้ ถ้าพิจารณาที่อินพุตและเอาต์พุตของแนนด์เกต จะเห็นว่าค่าทั้งสองจะเป็น Complement ซึ่งกันและกันนั่นหมายความว่า รูปแบบอินพุตเป็น Minterm แต่เอาต์พุตเป็น Maxterm

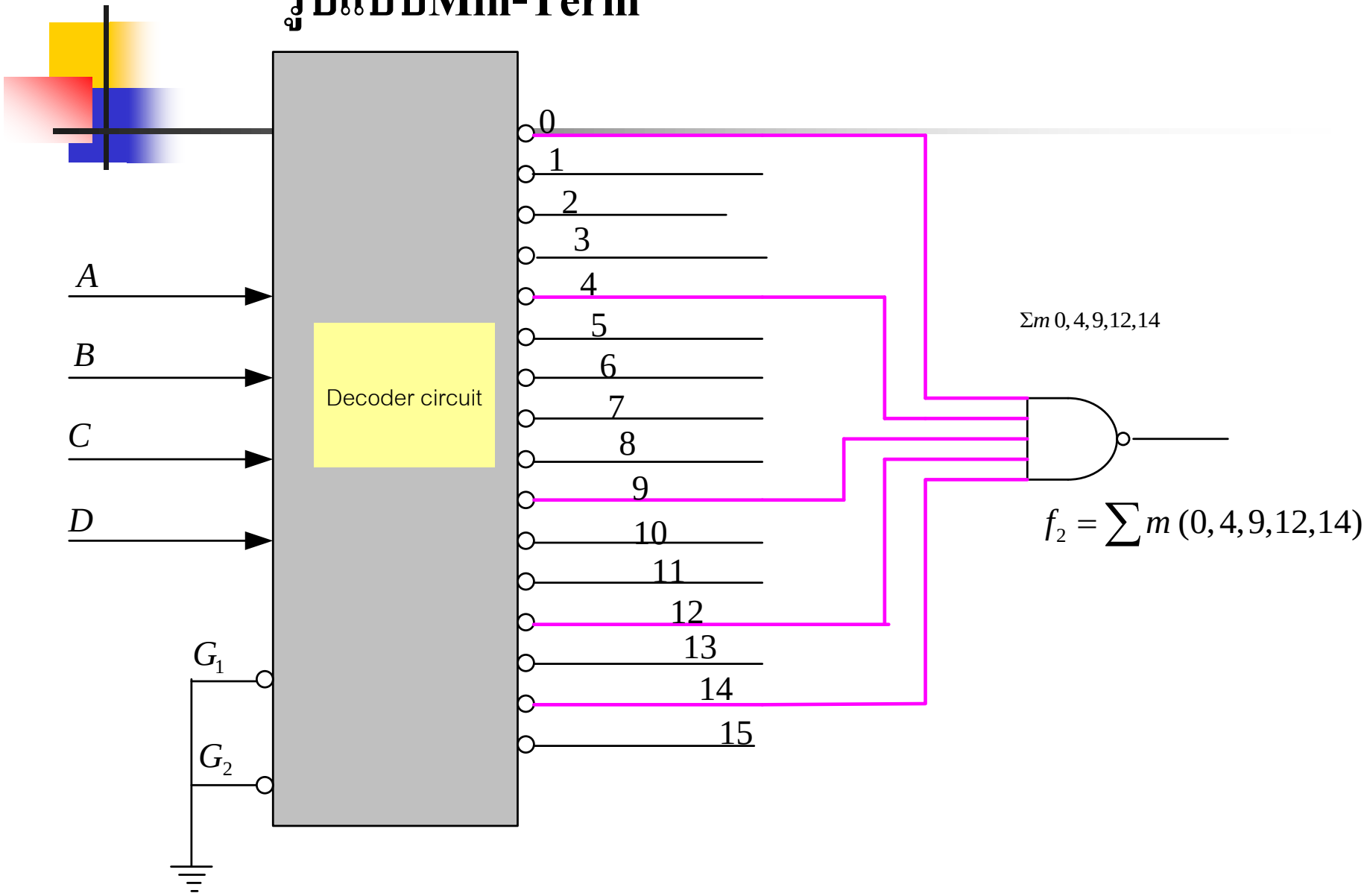
# ตัวอย่างการออกแบบวงจรเกตโดยใช้วงจรถอดรหัส



1. $\Sigma\, m$ 0, 4, 9, 12, 14

2. $\pi\, M$ 2, 3, 6

# 461 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูปแบบMin-Term



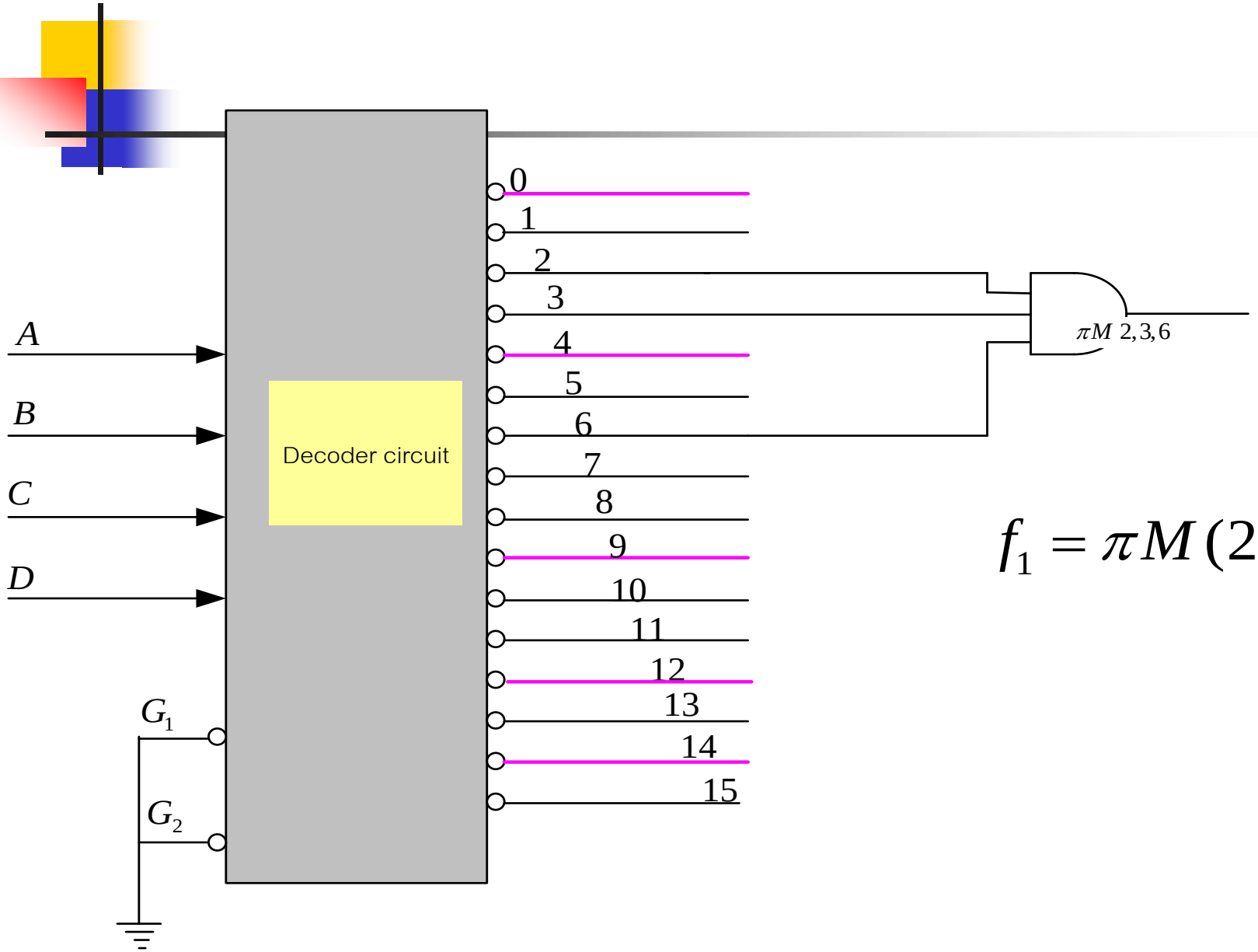
# อธิบาย

- การออกแบบวงจรในรูปแบบ Minterm นี้ขั้นแรกจะต้องทำการเปลี่ยนค่าเอาต์พุตที่เกิดขึ้นในรูปแบบMaxterm ให้อยู่ในรูปแบบ minterm โดยการใช้แนนด์เกตกระทำ
- ดังตัวอย่าง $0 = A + B + C + D$ , $4 = A + \bar{B} + C + D$ , $9 = \bar{A} + B + C + \bar{D}$ ,

  $12 = \bar{A} + \bar{B} + C + D$ , $14 = \bar{A} + \bar{B} + \bar{C} + D$
- นำค่าทั้งหมดมา NOT AND สามารถเปลี่ยนให้อยู่ในรูป Minterm โดยใช้ทฤษฎี De'Morgan ช่วยในการพิสูจน์สมการได้ดังนี้

$$f_2 = \overline{\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}}$$

# 462 การนำวงจรถอดรหัสไปออกแบบลอจิกเกตในรูปแบบMax-Term




$$f_1 = \pi M(2,3,6)$$

# อธิบาย

- เอาต์พุตที่ออกจากโมดุลคอมไบเนชัน แบบของวงจรถอดรหัสซึ่งค่าเอาต์พุตที่เกิดขึ้นจะอยู่ในรูปแบบ Maxterm ดังตารางที่นำเสนอไว้
- สามารถนำค่าเอาต์พุตที่เกิดขึ้นจากวงจรถอดรหัสมาแอนด์กันโดยตรง
- ค่าที่กำหนด ¶M(2,3,6)
- ค่าที่ได้ทางเอาต์พุตคือ$(A+\overline{B}+C+D)(\overline{A}+\overline{B}+C+D)(A+B+\overline{C}+\overline{D})$

# ทดสอบความรู้สัปดาห์ที่ 6

1. จงอธิบายถึงความสัมพันธ์ระหว่าง SOP กับการ Decoder ว่าสามารถนำมาเกี่ยวโยงกันอย่างไร
2. จงออกแบบด้วยวงจร Decoder จากตารางความจริงที่มีตัวแปร 4 ตัวแปร ที่มีค่าเอาต์พุตเป็น1 ดังนี้ 2, 4, 6, 8, 10, 12, 14
3. จงอธิบายการทำงานของการวงจร Decoder ที่แสดงผลทางเอาต์พุตเป็นแบบ Seven Segment ทั้ง Common Cathoad , Common Anode พร้อมเขียนตารางความจริงประกอบคำอธิบาย
4. จากการออกแบบวงจรเข้ารหัสที่นักศึกษาได้ออกแบบ จงนำรหัสที่ได้มาทำการออกแบบวงจรถอดรหัสให้เหมือนเดิม
5. จากคุณสมบัติของวงจร 74154 นำไปประยุกต์ใช้ในการออกแบบวงจร Combination อย่างไร

# เนื้อหาคำบรรยายใน สัปดาห์ที่ 7

# การมัลติเพล็กซ์(Multiplex) และดีมัลติเพล็กซ์(Demultiplex)

# สัปดาห์ที่ 7 การมัลติเพล็กซ์ และดีมัลติเพล็กซ์

47 หลักการพื้นฐานของมัลติเพล็กซ์(Multiplex) และดีมัลติเพล็กซ์(Demultiplex)

471 หลักการเบื้องต้นของการมัลติเพล็กซ์ขนาด 2 ช่องข้อมูล เป็น 1 ช่องข้อมูล

472 หลักการมัลติเพล็กซ์ขนาด 4 ช่องข้อมูลเป็น 1 ช่องข้อมูล

473 การออกแบบวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูล โดยใช้วงจรมัลติเพล็กซ์ 4 อินพุต 2 ภาค

474 การนำวงจรมัลติเพล็กซ์ไปใช้งาน

48 หลักการดีมัลติเพล็กซ์ (Demultiplex)

481 หลักการดีมัลติเพล็กซ์ขนาด 1 ช่องข้อมูลเป็น 2 ช่องข้อมูล

482 หลักการออกแบบวงจรดีมัลติเพล็กซ์ขนาด 1 ช่องเป็น 4 ช่องข้อมูล

483 การออกแบบวงจรดีมัลติเพล็กซ์ขนาด 4 เอาต์พุต

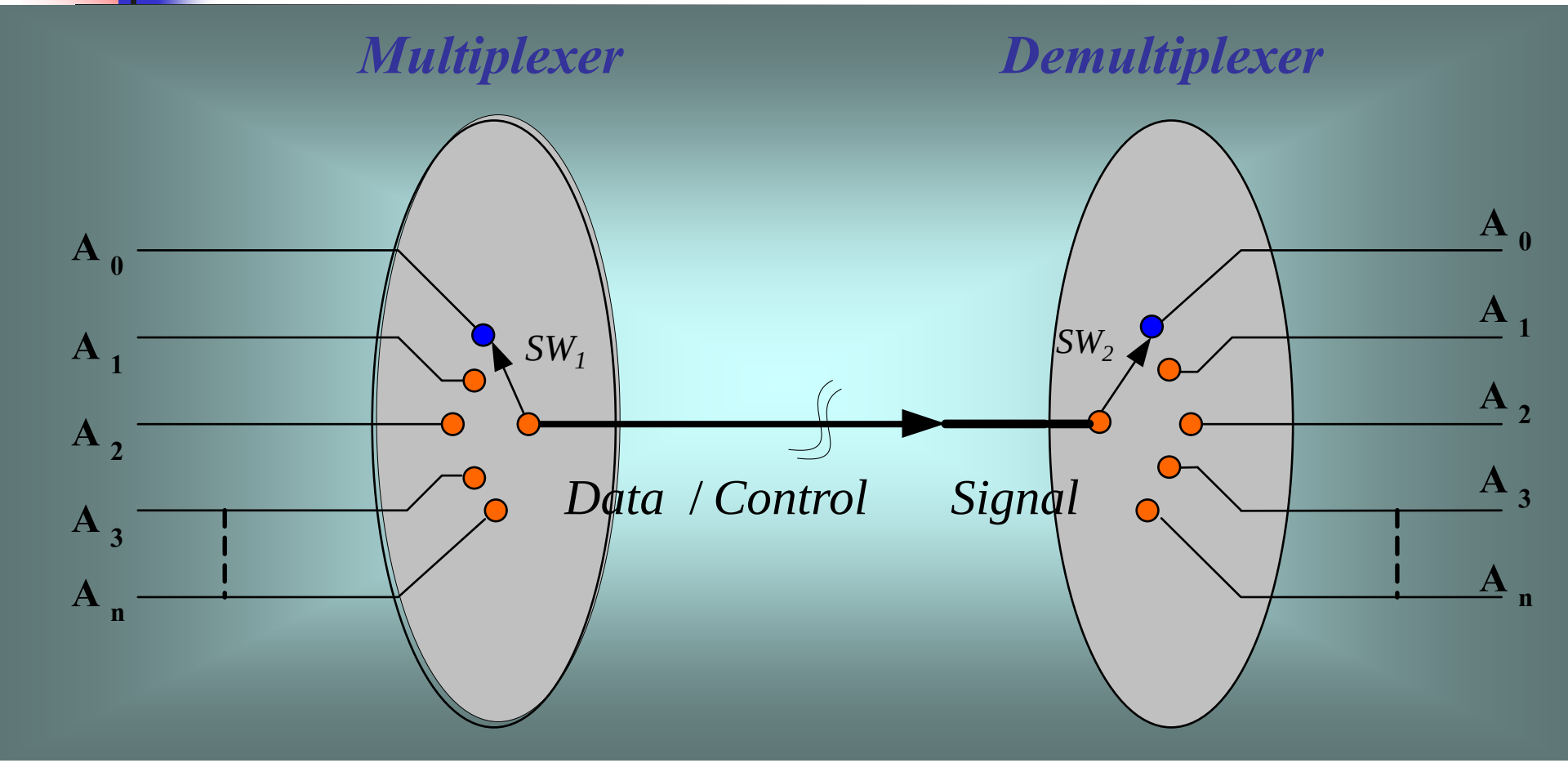
484 การออกแบบวงจรดีมัลติเพล็กซ์ขนาด n เอาต์พุต

49 การนำวงจรมัลติเพล็กซ์ และดีมัลติเพล็กซ์มาใช้งาน

491 การนำวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์มาต่อใช้งาน

492 การส่งข้อมูล และกระจายข้อมูลขนาด n บิต

# 47 หลักการพื้นฐานของมัลติเพล็กซ์/ดีมัลติเพล็กซ์



# อธิบาย

- หลักการเบื้องต้นของการมัลติเพล็กซ์ / ดีมัลติเพล็กซ์
- การมัลติเพล็กซ์หมายถึงการผสมผสานข้อมูลในการส่ง ข้อมูลที่ผสมผสานสามารถแยกออกจากกันเหมือนข้อมูลเบื้องต้น โดยการดีมัลติเพล็กซ์
- พิจารณาจากรูป $sw_1$ กับ $sw_2$ จะทำงานแบบซิงโครนัสกัน (Synchronous)
- ข้อมูลทางผู้ส่งกับข้อมูลทางผู้รับจะเป็นแบบขนานที่จะนำไปใช้งาน แต่ข้อมูลที่ส่งผ่านช่องสัญญาณจะเป็นแบบอนุกรม
- ประโยชน์ที่เกิดขึ้นจากการมัลติเพล็กซ์ / ดีมัลติเพล็กซ์ ทำให้

  - ประหยัดช่องสัญญาณในการส่งข้อมูลแบบขนานหลายบิต

  - ประหยัดอุปกรณ์ในการส่งข้อมูล เพราะสามารถใช้ร่วมกันได้ทุกบิต

  - ประหยัดพลังงานไฟฟ้า เพราะใช้เพียงข้อมูลเดียวในการส่ง /รับแต่ต่างกันที่เวลาในการรับส่ง

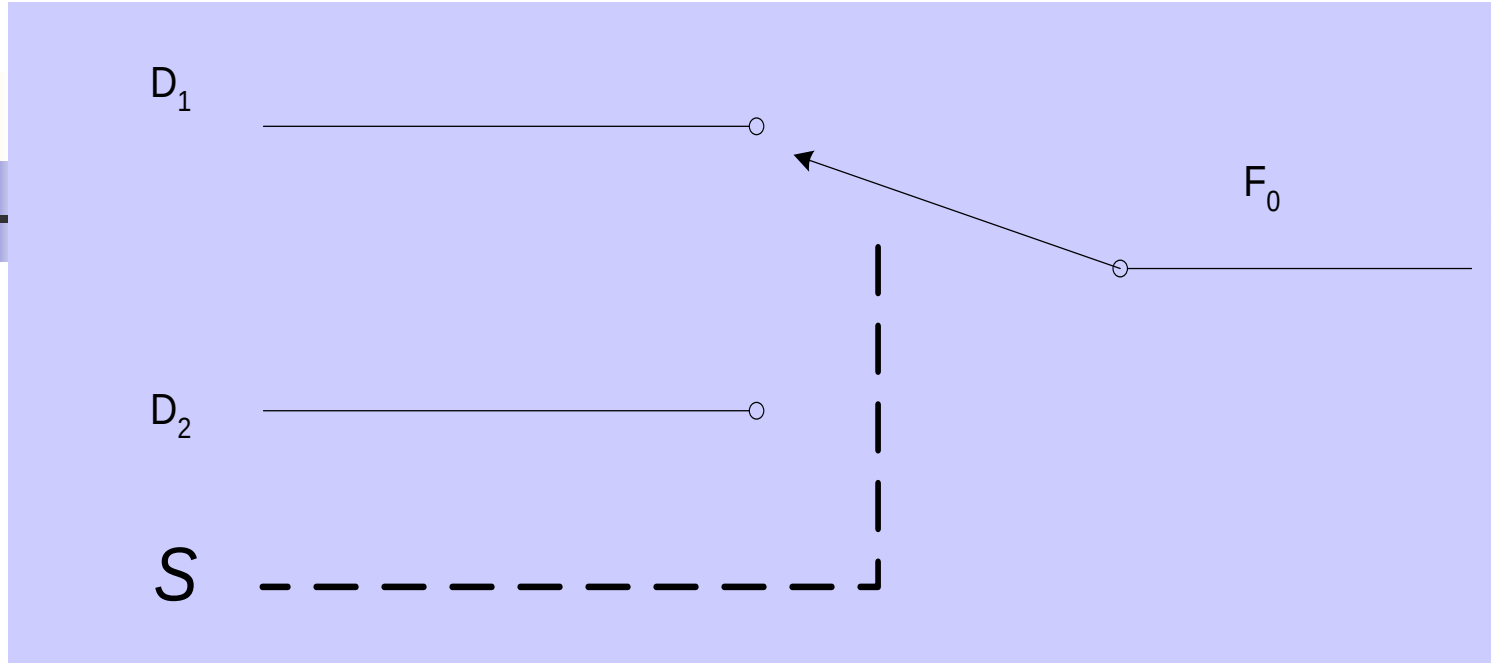
| เวลา | สัญญาณควบคุมสวิทซ์ | | | ตำแหน่งอินพุต | ตำแหน่งเอาต์พุต | ข้อมูลเอาต์พุต |
|---|---|---|---|---|---|---|
| | A | B | C | $SW_1$ | $SW_2$ | In/Out |
| $T_0$ | 0 | 0 | 0 | 1 | 1 | $A_0$ |
| $T_1$ | 0 | 0 | 1 | $\bar{A}\bar{B}C$ | $\bar{A}\bar{B}C$ | $A_1$ |
| $T_2$ | 0 | 1 | 0 | $\bar{A}B\bar{C}$ | $\bar{A}B\bar{C}$ | $A_2$ |
| $T_3$ | 0 | 1 | 1 | $\bar{A}BC$ | $\bar{A}BC$ | $A_3$ |
| $T_4$ | 1 | 0 | 0 | $A\bar{B}\bar{C}$ | $A\bar{B}\bar{C}$ | $A_4$ |
| $T_5$ | 1 | 0 | 1 | $A\bar{B}C$ | $A\bar{B}C$ | $A_5$ |
| $T_6$ | 1 | 1 | 0 | $AB\bar{C}$ | $AB\bar{C}$ | $A_6$ |
| $T_7$ | 1 | 1 | 1 | $ABC$ | $ABC$ | $A_7$ |

ตารางแสดงความสัมพันธ์ของอินพุต/เอาต์พุต ในช่วงเวลาต่างๆ ที่สวิทซ์ควบคุมทำงาน

# อธิบาย

- จากตารางความจริงค่าไทม์มิ่ง(Timing) ของช่วงเวลาเริ่มต้น $T_0$ถึง $T_8$
- จำนวนบิตของสัญญาณควบคุมสามารถหาได้จาก ข้อมูลทางอินพุตว่ามีจำนวนกี่ช่องของข้อมูลในที่นี้มี 8 ช่องข้อมูล สัญญาณควบคุมจึงมีเท่ากับ 3 บิต ($2^3$=8)
- ABC เป็นสัญญาณควบคุมสวิทซ์
- $SW_1$,$SW_2$ จะใช้สัญญาณควบคุมทั้งอินพุตกับเอาต์พุต
- ช่วงเวลา $T_1$ , $SW_1$ กับ $SW_2$ จะมีค่าเท่ากับ $\bar{A}\bar{B}C$=1 สัญญาณควบคุมช่องอื่นเป็น 0 หมด ข้อมูลจะเดินทางจากอินพุตช่อง $A_1$ ถูกส่งไปยังเอาต์พุต $A_1$ เช่นกัน
- ช่วงเวลา $T_3$ $SW_1$ กับ $SW_2$ จะมีค่าเท่ากับ $\bar{A}BC$=1 สัญญาณควบคุมช่องอื่นเป็น 0 หมด ข้อมูลจะเดินทางจากอินพุตช่อง $A_3$ ถูกส่งไปยังเอาต์พุต $A_3$ช่องอื่นๆ ข้อมูลไม่ถูกเลือกจึงไม่สามารถออกไปที่เอาต์พุต

# 471 หลักการเบื้องต้นของการมัลติเพล็กซ์ขนาด2ช่องข้อมูลเป็น1ช่องข้อมูล




**ตารางฟังก์ชันของเอาต์พุต**

| S | $F_0$ |
|---|---|
| 0 | $D_1$ |
| 1 | $D_2$ |

$$F_0 = D_1\bar{S} + D_2S$$

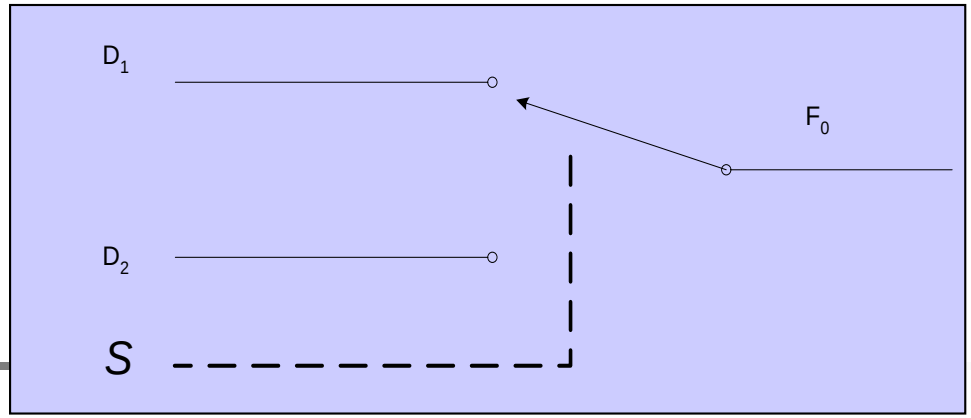
# อธิบาย

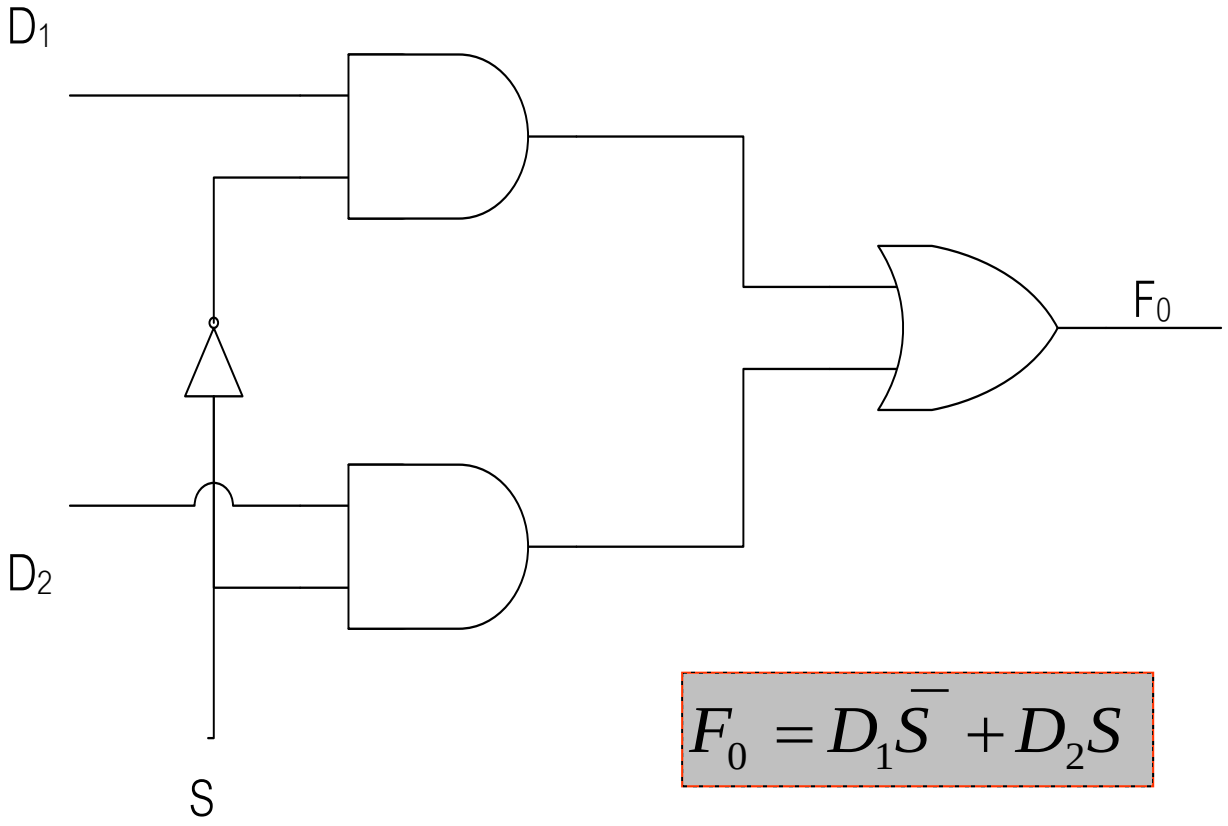
- การส่งข้อมูลทางอินพุต 2 ช่องสัญญาณคือ $D_1$ กับ $D_2$
- ข้อมูลที่ถูกส่งออกที่เอาต์พุตจะถูกควบคุมด้วย S
- พิจารณาจากค่าตารางฟังก์ชันของเอาต์พุต
- สัญญาณที่ออกจากช่องสัญญาณ $D_1$ ไปยังเอาต์พุต$F_0$ ค่า S จะควบคุมด้วย 0
- สัญญาณที่ออกจากช่องสัญญาณ $D_2$ ไปยังเอาต์พุต $F_0$ค่า S จะควบคุมด้วย 1
- สามารถเขียนเป็นฟังก์ชัน โดยมี $F_0$ แทนค่าเอาต์พุตมีค่าเท่ากับ

$$F_0 = D_1\bar{S} + D_2S$$

- จากค่าเอาต์พุตก็นำค่ามาเขียนวงจรลอจิกเกตได้ดังรูปถัดมา

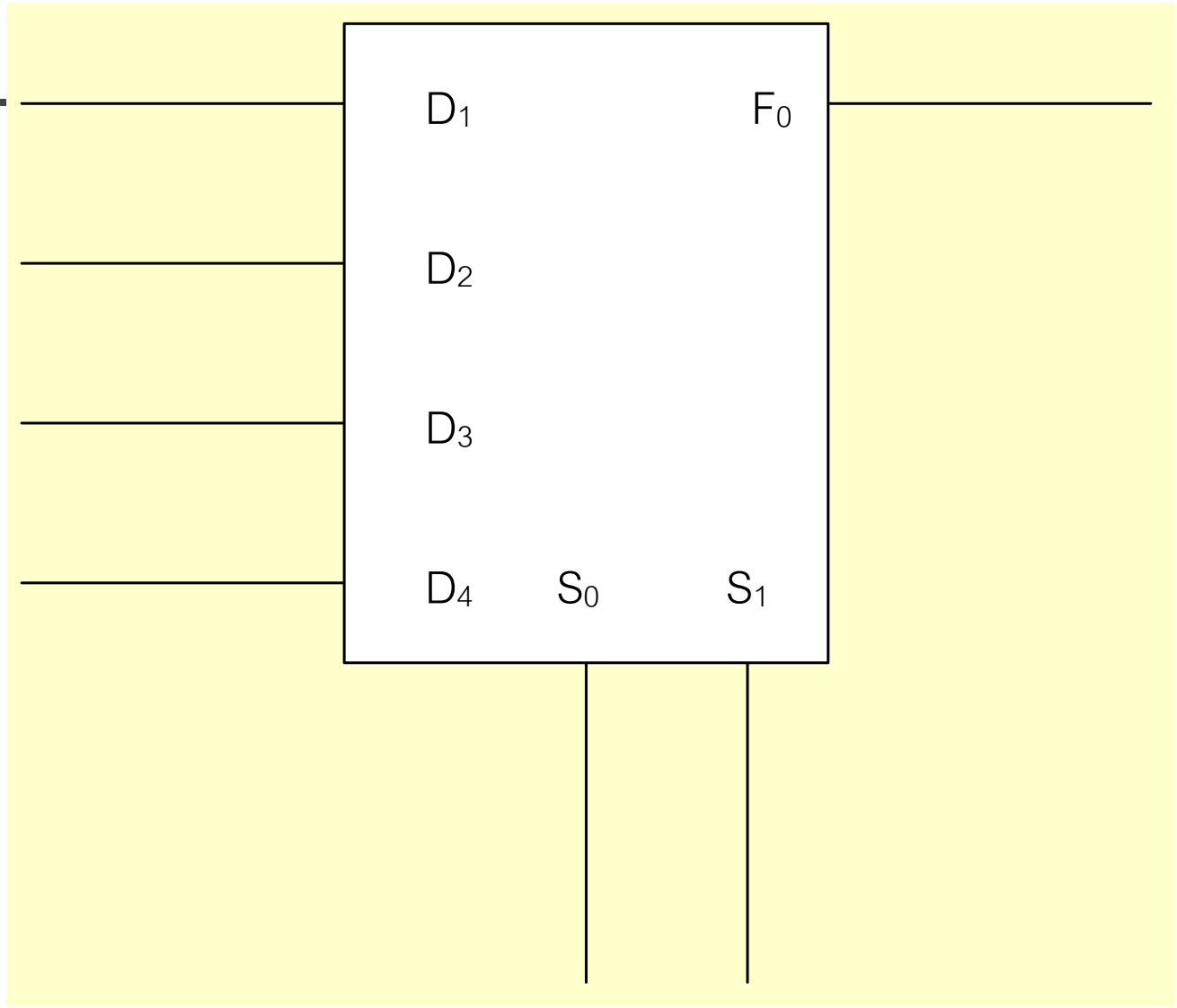
# วงจรมัลติเพล็กซ์




$$F_0 = D_1\overline{S} + D_2S$$

# 472 หลักการมัลติเพล็กซ์ขนาด 4 ช่องข้อมูลเป็น 1ช่องข้อมูล



# อธิบาย

- ข้อมูลทางอินพุตจำนวน 4 ช่องคือ $D_1,D_2,D_3,D_4$ จะถูกส่งไปยังเอาต์พุตในเวลาที่ต่างกันโดยมี S เป็นตัวควบคุม
- ค่า$S_0,S_1$ จะเป็นตัวกำหนดค่าเวลาในการส่งข้อมูลไปยังเอาต์พุต
- โดยกำหนดให้ $S_0,S_1$= 00 ข้อมูล $D_1$ จะถูกส่งออกไปยัง $F_0$
- โดยกำหนดให้ $S_0,S_1$= 01 ข้อมูล $D_2$ จะถูกส่งออกไปยัง $F_0$
- โดยกำหนดให้ $S_0,S_1$= 10 ข้อมูล $D_3$ จะถูกส่งออกไปยัง $F_0$
- โดยกำหนดให้ $S_0,S_1$ = 11 ข้อมูล $D_4$ จะถูกส่งออกไปยัง $F_0$
- เพื่อให้ง่ายต่อการพิจารณาโดยดูจากตารางความจริง

# วงจรลอจิกของ 2 ข้อมูลเป็น 1ข้อมูล

| D | $S_1$ | $S_0$ | $F_0$ |
|---|---|---|---|
| $D_1$ | 0 | 0 | $D_1$ |
| $D_2$ | 0 | 1 | $D_2(1)$ |
| $D_3$ | 1 | 0 | $D_3(2)$ |
| $D_4$ | 1 | 1 | $D_4(3)$ |

## จากตารางเขียนฟังก์ชันของเอาต์พุต

$$F_0 = D_1\overline{S}_0\overline{S}_1 + D_2S_0\overline{S}_1 + D_3\overline{S}_0S_1 + D_4S_0S_1$$

$$= (D_1\overline{S}_1 + D_3S_1)\overline{S}_0 + (D_2\overline{S}_1 + D_4S_1)S_0$$

# อธิบาย

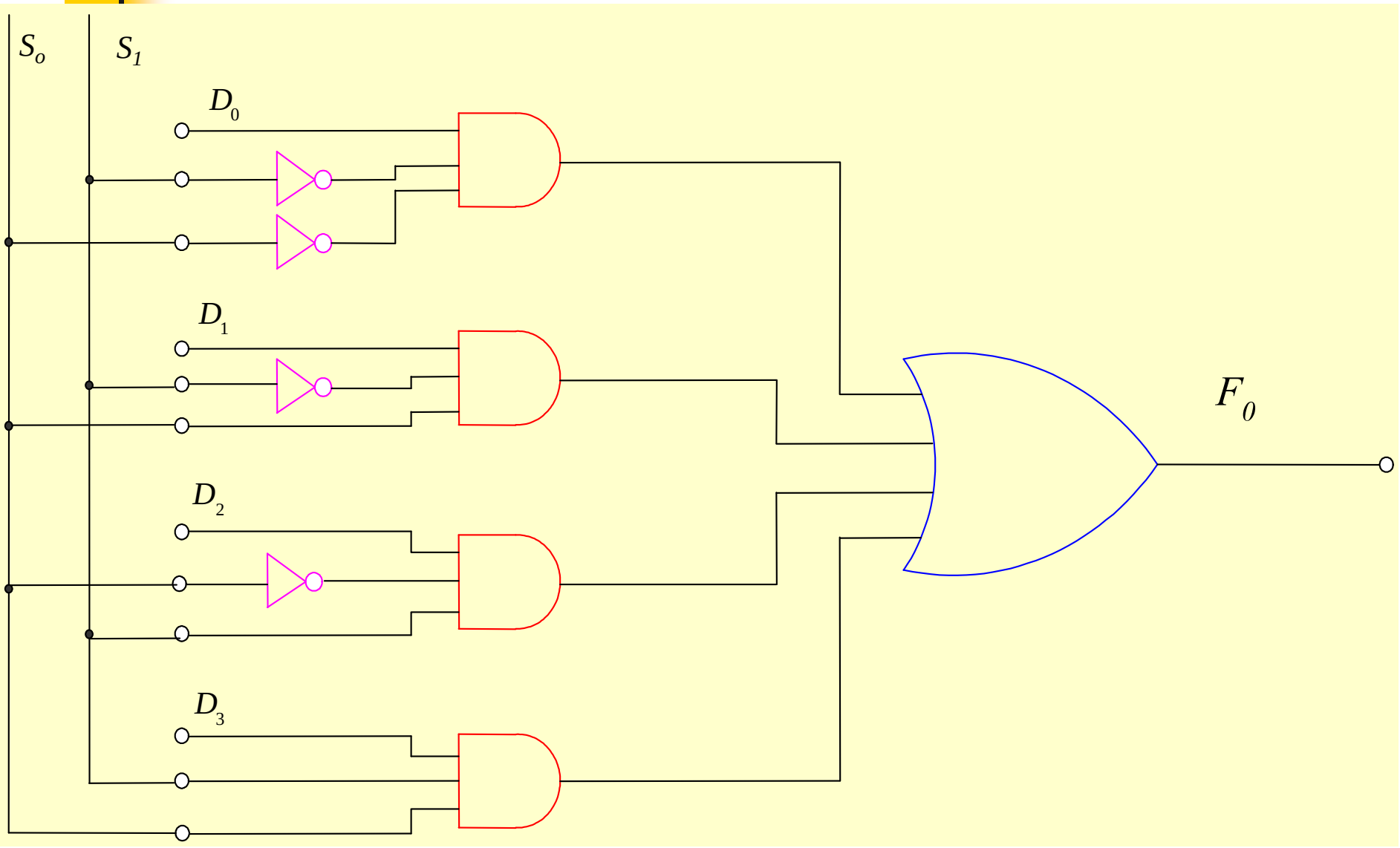
- จากตารางการทำงานของวงจรมัลติเพล็กซ์ขนาด 4 ช่อง
- D เป็นข้อมูลของช่อง1ใช้($D_1$), ช่อง2ใช้($D_2$),ช่อง3ใช้($D_3$),ช่อง4ใช้($D_4$)
- $S_0$เป็นบิตเลือกสัญญาณซึ่งแทนด้วย$L_{SB}$ ( Low Bit)
- $S_1$เป็นบิตเลือกสัญญาณซึ่งแทนด้วย $M_{SB}$ ( High Bit)
- $F_0$ เป็นค่าฟังก์ชันทางเอาต์พุตที่เกิดจากการเลือกสัญญาณทางอินพุตด้วยค่า $S_0$ , $S_1$ สามารถเขียนดังสมการ

$$F_0 = D_1\overline{S_0}\,\overline{S_1} + D_2S_0\overline{S_1} + D_3\overline{S_0}S_1 + D_4S_0S_1 \qquad (1)$$

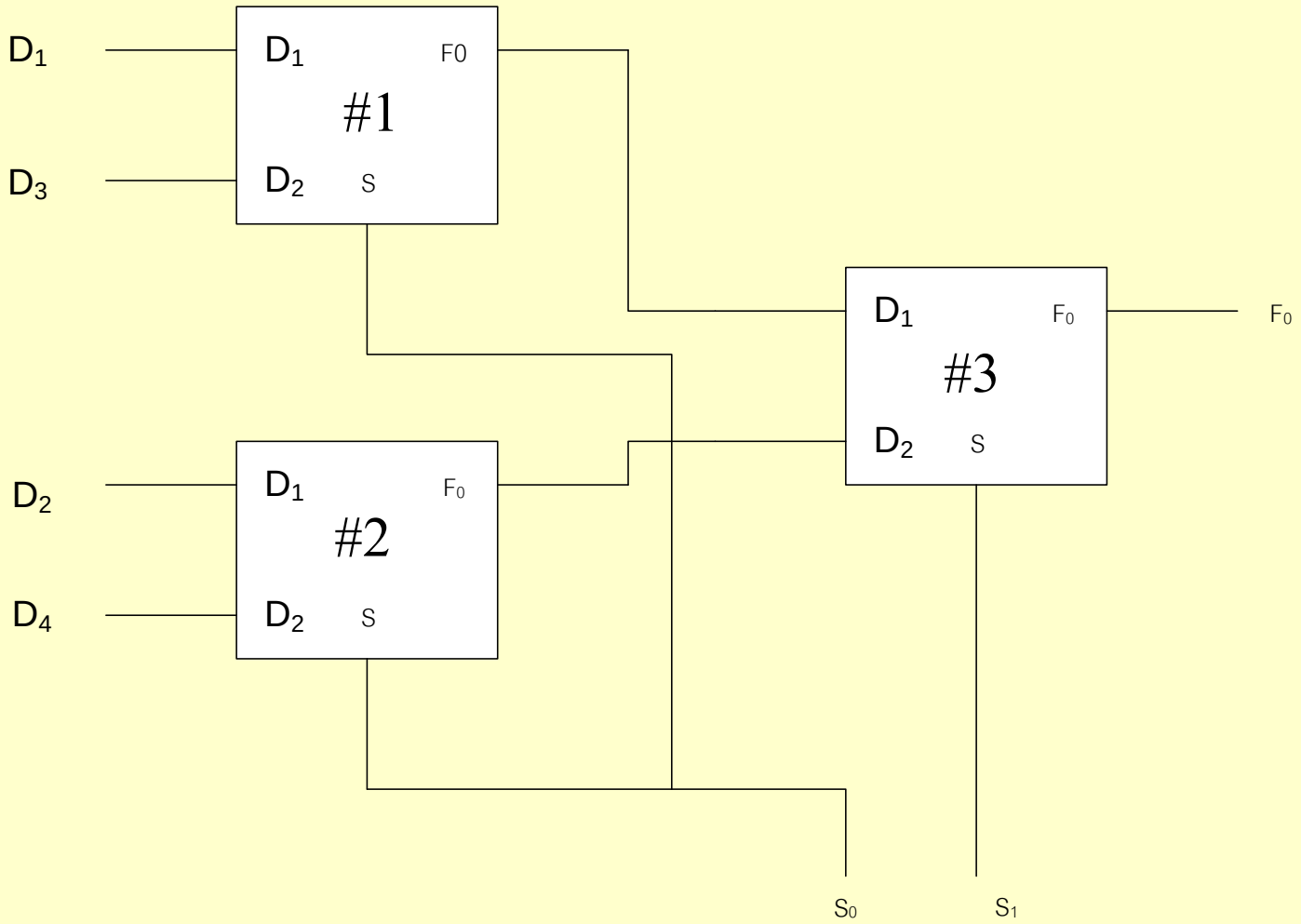
$$F_0 = (D_1\overline{S_1} + D_3S_{1)})\overline{S_0} + (D_3\overline{S_1} + D_4S_1)S_0 \qquad (2)$$

- การสร้างวงจรมัลติเพล็กเซอร์สามารถนำสมการที่ 2 ไปสร้างดังรูปวงจร

# วงจรมัลติเพล็กซ์แบบ 4 บิต



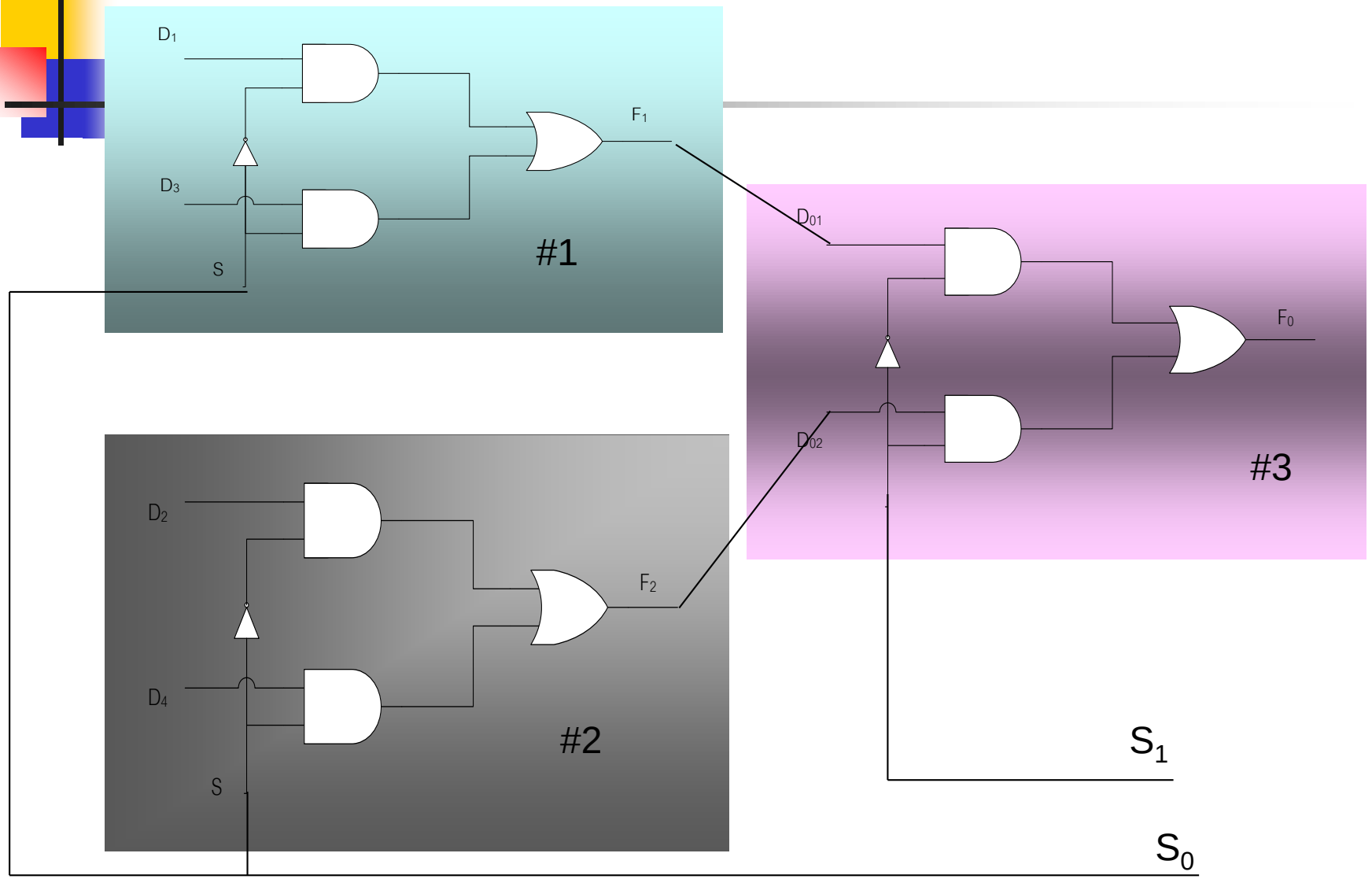
# โครงสร้างวงจร 4 ข้อมูลเป็น 1ข้อมูลแบบมัลติเพล็กซ์



# อธิบาย

- การสร้างวงจรมัลติเพล็กซ์ 4 อินพุต โดยการนำวงจรมัลติเพล็กซ์ที่ออกแบบ 2อินพุตมาต่อกัน2 ภาค(Stage) ดังสมการที่2 มีการทำงานดังนี้
- #1 อินพุตจะป้อนด้วยสัญญาณอินพุต$D_1$กับ$D_3$ ถูกควบคุมด้วย $S_0(L_{SB})$
- #2 อินพุตจะป้อนด้วยสัญญาณอินพุต$D_2$กับ$D_4$ ถูกควบคุมด้วย $S_0(L_{SB})$
- #3 อินพุตจะป้อนด้วยสัญญาณเอาต์พุตของ #1 กับ #2 โดยมีสัญญาณควบคุม$S_1$ซึ่งเป็นบิต $M_{SB}$ เพื่อเลือกสัญญาณที่มาจากเอาต์พุตของ #2 กับ#1ออกเป็น #3
- $D_1$กับ$D_3$จะถูกเลือกเมื่อ $S_1$เป็น 0 $D_2$ กับ$D_4$ จะถูกเลือกเมื่อ$S_1$ เป็น1

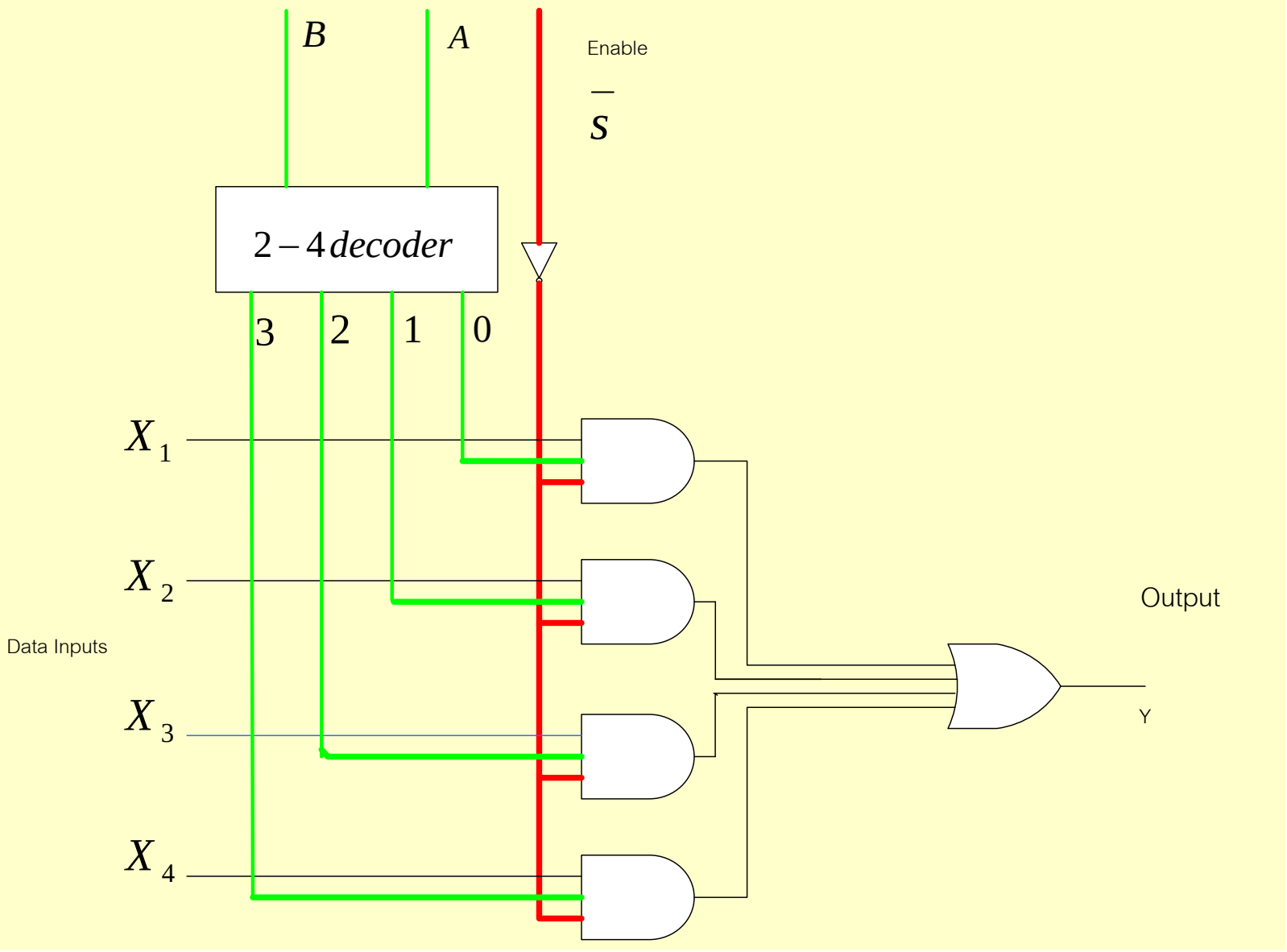
# วงจรมัลติเพล็กซ์ 4 อินพุต



# อธิบาย

- ตัวอย่างการมัลติเพล็กซ์ทางอินพุต 4 ช่องสัญญาณ เพื่อนำส่งออกทางเอาต์พุต 1 ช่องสัญญาณ
- ข้อมูลทางอินพุตคือ $D_1,D_2,D_3,D_4$
- สัญญาณควบคุมข้อมูลออกจะมีค่าเท่ากับ 2 บิต 4 ค่าที่แตกต่างกัน ให้เป็น $S_0,S_1$ ถ้า $S_0$=0 จะมัลติเพล็กซ์ของ #1 $D_1$และ #2 $D_2$ ส่งออกไปยัง $F_1,F_2$ โดยมี $S_1$ เลือก $F_1,F_2$ อีกครั้ง
- ถ้า $S_0$=1 จะมัลติเพล็กซ์ของ #1 $D_3$ และ #2 $D_4$ ส่งออกไปยัง $F_1,F_2$ โดยมี $S_1$ เลือก $F_1,F_2$ อีกครั้ง
- ถ้า $S_1$=0 ข้อมูลที่ถูกเลือกจะเป็น $D_1,D_3$
- ถ้า $S_1$=1 ข้อมูลที่ถูกเลือกจะเป็น $D_2,D_4$

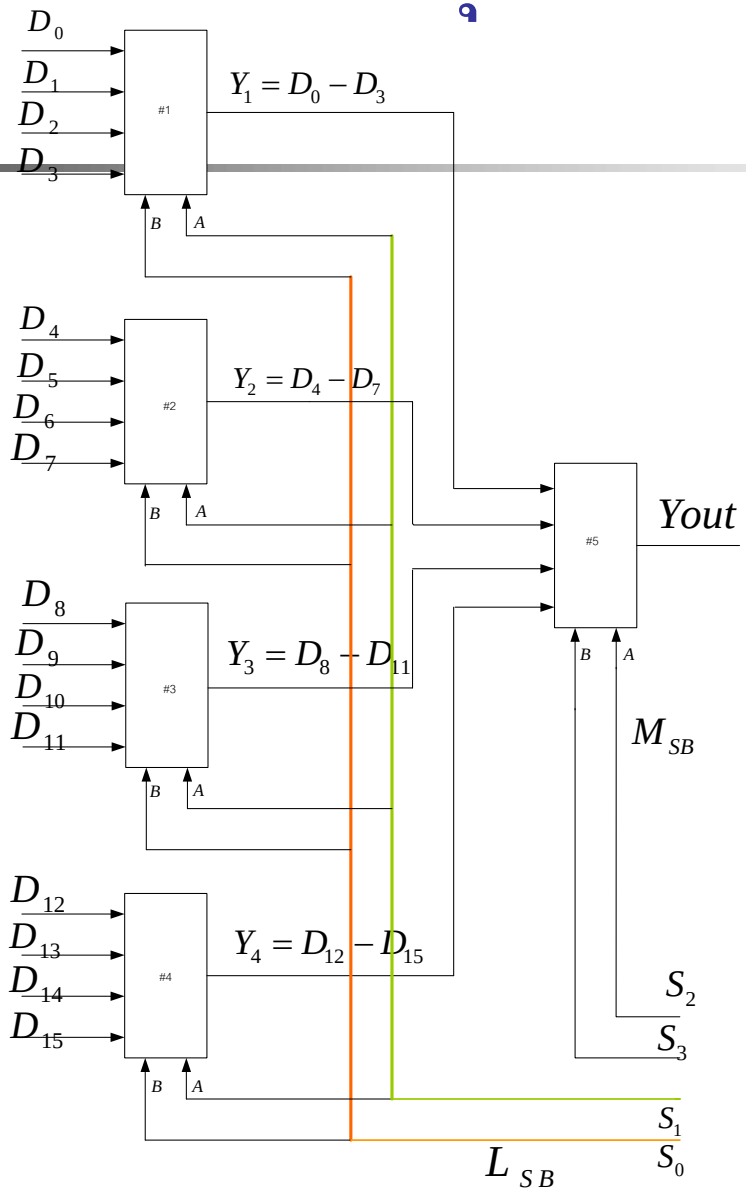
# วงจรมัลติเพล็กซ์แบบส่งข้อมูล 4 ทางโดยใช้วงจรถอดรหัส2 อินพุต



# อธิบาย

- วงจรมัลติเพล็กซ์แบบส่งข้อมูล 4 ทางพร้อมด้วยสัญญาณ Enable เพื่อควบคุมโมดุล
- จากรูปวงจรมัลติเพล็กซ์ที่นำมาใช้งานในวงจรดิจิตอล ประกอบด้วย 3 ส่วน
- ส่วนที่1 นำวงจรดีโค้ดเดอร์มาเป็นตัวมัลติเพล็กซ์ทางอินพุต A , B
- ส่วนที่2 เป็นข้อมูลมาจากอินพุต $X_1$ , $X_2$ , $X_3$ , $X_4$
- ส่วนที่3 ควบคุมโมดุลการทำงาน Enable $\overline{S}$
- ข้อมูลทั้งสามส่วนนี้จะจะถูกผสมผสานอย่างเป็นระเบียบไปยังเอาต์พุตเดียว

# 473 การออกแบบวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูล โดยใช้วงจรมัลติเพล็กซ์ 4 อินพุต 2 ภาค



# อธิบาย

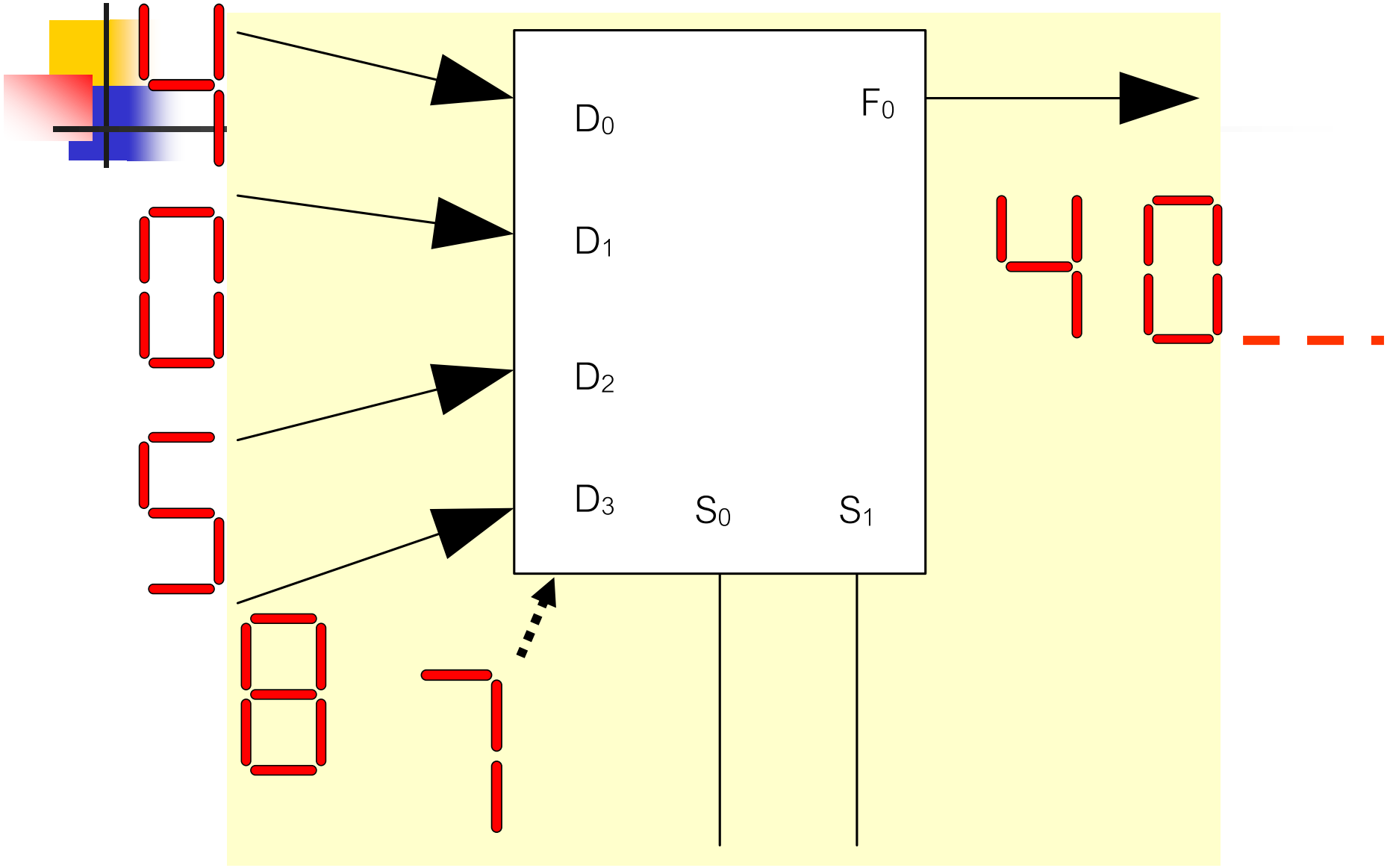
- การออกแบบวงจรมัลติเพล็กซ์ขนาด 16 อินพุต โดยการใช้วงจรมัลติเพล็กซ์ขนาด 4 อินพุต 2 ภาค ต่อกันโดยไม่มีการดัดแปลงวงจรมัลติเพล็กซ์ขนาด 4 อินพุต มีหลักการทำงานดังนี้คือ
- พิจารณาจากตารางความจริง $S_3,S_2$ เป็นบิตควบคุม $M_{SB}$ จะทำการมัลติเพล็กซ์ 4 ข้อมูลที่ถูกแบ่งกลุ่มการมัลติเพล็กซ์ มาป้อนให้อินพุตทั้ง 4 ในภาคหลัง #5 ได้สัญญาณเป็น Yout
- $S_0,S_1$ เป็นบิตควบคุม $L_{SB}$ ของภาคหน้าจำนวน 16 ข้อมูล คือ $D_0$-$D_{15}$
- โดยการแบ่งข้อมูลออกเป็นกลุ่มคือที่ถูกควบคุมด้วยบิต $L_{SB}$ วงจรมัลติเพล็กซ์ #1 จะให้ข้อมูล $Y_1=D_0-D_3$ , วงจรมัลติเพล็กซ์ #2 ให้ $Y_2= D_4-D_7$ , วงจรมัลติเพล็กซ์ #3 ให้ $Y_3=D_8-D_{11}$, วงจรมัลติเพล็กซ์ #4 ให้ $Y_4=D_{12}-D_{15}$
- ข้อมูลจาก $Y_1,Y_2,Y_3,Y_4$ จะถูกบิต$M_{SB}$ ควบคุมโดยใช้วงจรมัลติเพล็กซ์ #5 มัลติเพล็กซ์อย่างละเอียดเพื่อให้ส่งไปยัง Yout สามารถเลือกกลุ่ม 0-3, หรือ 4-7,8-11,12-15 โดย$S_2$-$S_3$ถ้า $S_2$-$S_3$=00เลือกกลุ่ม 0-3 ,01 เลือก4-7, 10เลือก8-11, 11 เลือก 12-15 ดังที่กล่าวมา

| Data | $S_4$ | $S_3$ | $S_2$ | $S_1$ | $Y_1$ | $Y_2$ | $Y_3$ | $Y_4$ | $Y_{out}$ |
|---|---|---|---|---|---|---|---|---|---|
| $D_0$ | 0 | 0 | 0 | 0 | $D_0$ | $D_4$ | $D_8$ | $D_{12}$ | $D_0$ |
| $D_1$ | 0 | 0 | 0 | 1 | $D_1$ | $D_5$ | $D_9$ | $D_{13}$ | $D_1$ |
| $D_2$ | 0 | 0 | 1 | 0 | $D_2$ | $D_6$ | $D_{10}$ | $D_{14}$ | $D_2$ |
| $D_3$ | 0 | 0 | 1 | 1 | $D_3$ | $D_7$ | $D_{11}$ | $D_{15}$ | $D_3$ |
| $D_4$ | 0 | 1 | 0 | 0 | $D_0$ | $D_4$ | $D_8$ | $D_{12}$ | $D_4$ |
| $D_5$ | 0 | 1 | 0 | 1 | $D_1$ | $D_5$ | $D_9$ | $D_{13}$ | $D_5$ |
| $D_6$ | 0 | 1 | 1 | 0 | $D_2$ | $D_6$ | $D_{10}$ | $D_{14}$ | $D_6$ |
| $D_7$ | 0 | 1 | 1 | 1 | $D_3$ | $D_7$ | $D_{11}$ | $D_{15}$ | $D_7$ |
| $D_8$ | 1 | 0 | 0 | 0 | $D_0$ | $D_4$ | $D_8$ | $D_{12}$ | $D_8$ |
| $D_9$ | 1 | 0 | 0 | 1 | $D_1$ | $D_5$ | $D_9$ | $D_{13}$ | $D_9$ |
| $D_{10}$ | 1 | 0 | 1 | 0 | $D_2$ | $D_6$ | $D_{10}$ | $D_{14}$ | $D_{10}$ |
| $D_{11}$ | 1 | 0 | 1 | 1 | $D_3$ | $D_7$ | $D_{11}$ | $D_{15}$ | $D_{11}$ |
| $D_{12}$ | 1 | 1 | 0 | 0 | $D_0$ | $D_4$ | $D_8$ | $D_{12}$ | $D_{12}$ |
| $D_{13}$ | 1 | 1 | 0 | 1 | $D_1$ | $D_5$ | $D_9$ | $D_{13}$ | $D_{13}$ |
| $D_{14}$ | 1 | 1 | 1 | 0 | $D_2$ | $D_6$ | $D_{10}$ | $D_{14}$ | $D_{14}$ |
| $D_{15}$ | 1 | 1 | 1 | 1 | $D_3$ | $D_7$ | $D_{11}$ | $D_{15}$ | $D_{15}$ |

# อธิบาย

- ตารางแสดงค่าความจริงของวงจรมัลติเพล็กซ์ขนาด 16 ช่องข้อมูล
- สามารถแบ่งข้อมูลออกเป็น 4 ช่อง เพื่อให้ง่ายต่อการออกแบบวงจร โดยการนำข้อมูลที่เอาต์พุตของช่อง $Y_1, Y_2, Y_3, Y_4$ ไปรวมด้วยวงจรมัลติเพล็กซ์อีกระดับก็จะได้เป็น $Y_{OUT}$
- ข้อมูลถูกแบ่งเป็นส่วนด้วยการควบคุม2 บิตทางต่ำ$D_0$-$D_3$ ,$D_4$-$D_7$, $D_8$-$D_{11}$,$D_{12}$-$D_{15}$
- ข้อมูลเมื่อถูกควบคุมด้วยบิตทางสูง จำนวน 2 บิต คือ $Y_1, Y_2, Y_3, Y_4$

# 474 การนำวงจรมัลติเพล็กซ์ไปใช้งาน

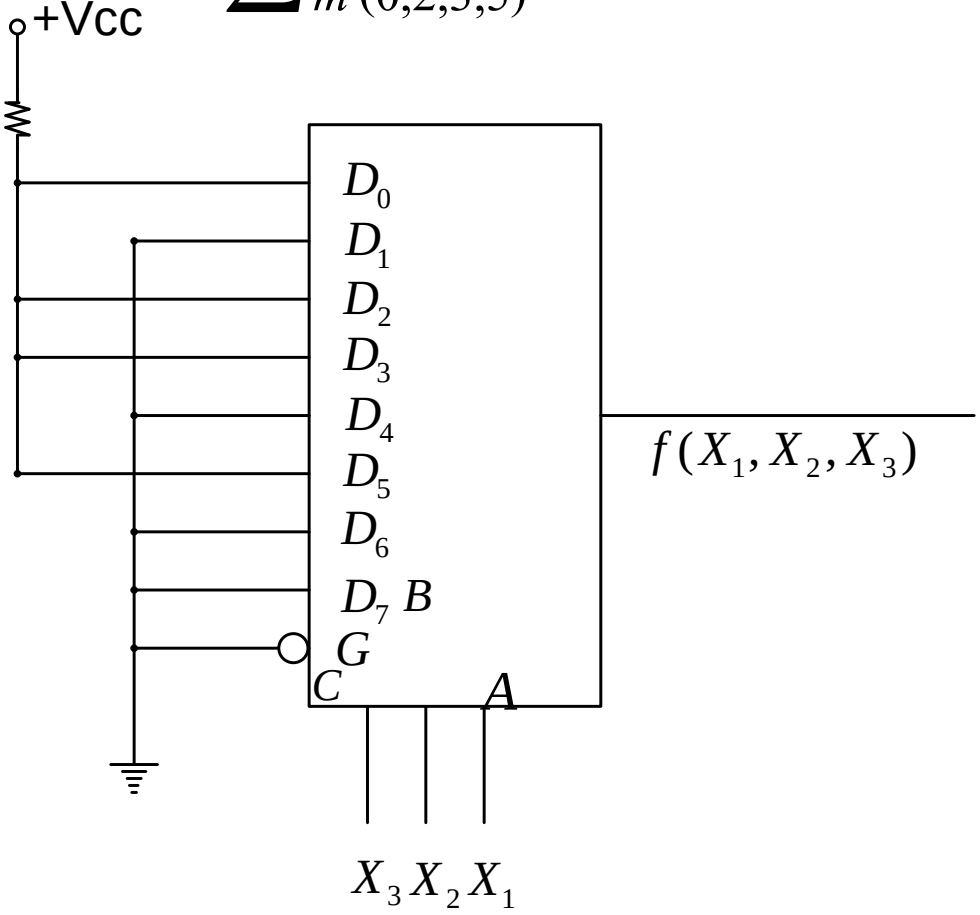


# อธิบาย

- การนำวงจรมัลติเพล็กซ์ไปใช้งาน ด้วยคุณสมบัติเด่นของวงจรมัลติเพล็กซ์ ทำให้ประหยัดในเรื่องการเดินสาย สามารถส่งข้อมูลครั้งละมาก ๆ ข้อมูลที่ส่งสามารถแยกออกจากกันได้ด้วยวงจรดีมัลติเพล็กซ์อย่างง่าย ๆ
- จากตัวอย่างการส่งข้อมูล $D_0 = 4$, $D_1 = 0$, $D_2 = 5$, $D_3 = 8$
- เอาต์พุตใช้สายเพียงชุดเดียว ข้อมูลจะถูกส่งสลับกันในสายที่เวลาต่างกัน
- ประหยัดในเรื่องกำลังงานด้วยกำลังรวมทั้งหมด คือค่ากำลังสูงสุดที่ส่งเพียงข้อมูลเดียวเท่านั้น
- กระแสที่ใช้จะมีเพียงการส่งของแต่ละข้อมูลซึ่งไม่ใช่กระแสรวมของทุกข้อมูล ทำให้เกิดการประหยัดพลังงานในระบบได้
- หลักการนี้จะนำไปใช้ในวงจรแสดงผลข้อมูลแบบ Seven segment ที่มีจำนวนตัวเลขแสดงผลหลายหลัก, วงจร ALU ที่จะกล่าวในสัปดาห์ต่อไป

# การนำวงจรมัลติเพล็กซ์ไปสร้างฟังก์ชันทางลอจิกเกตในรูป minterm

$$\sum m\,(0,2,3,5)$$



# อธิบาย

- ตัวอย่าง การนำวงจรมัลติเพล็กซ์ไปใช้ในรูปแบบ minterm
- ข้อมูลที่ $D_0, D_2, D_3, D_5$ จะถูกต่อไว้กับ +Vcc
- ส่วนข้อมูล $D_1, D_4, D_6, D_7$ จะต่อลง Ground
- สัญญาณควบคุม $X_1 - X_3$ จะถูกเลือกทุกข้อมูลไปยังเอาต์พุตในรูปแบบ minterm ดังนั้นข้อมูลที่ถูกนำไปใช้คือ $D_0+D_2+D_3+D_5$ ดังในตารางที่นำเสนอต่อไป
- ค่าเอาต์พุตที่ได้คือ $\bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C} + A\bar{B}C$

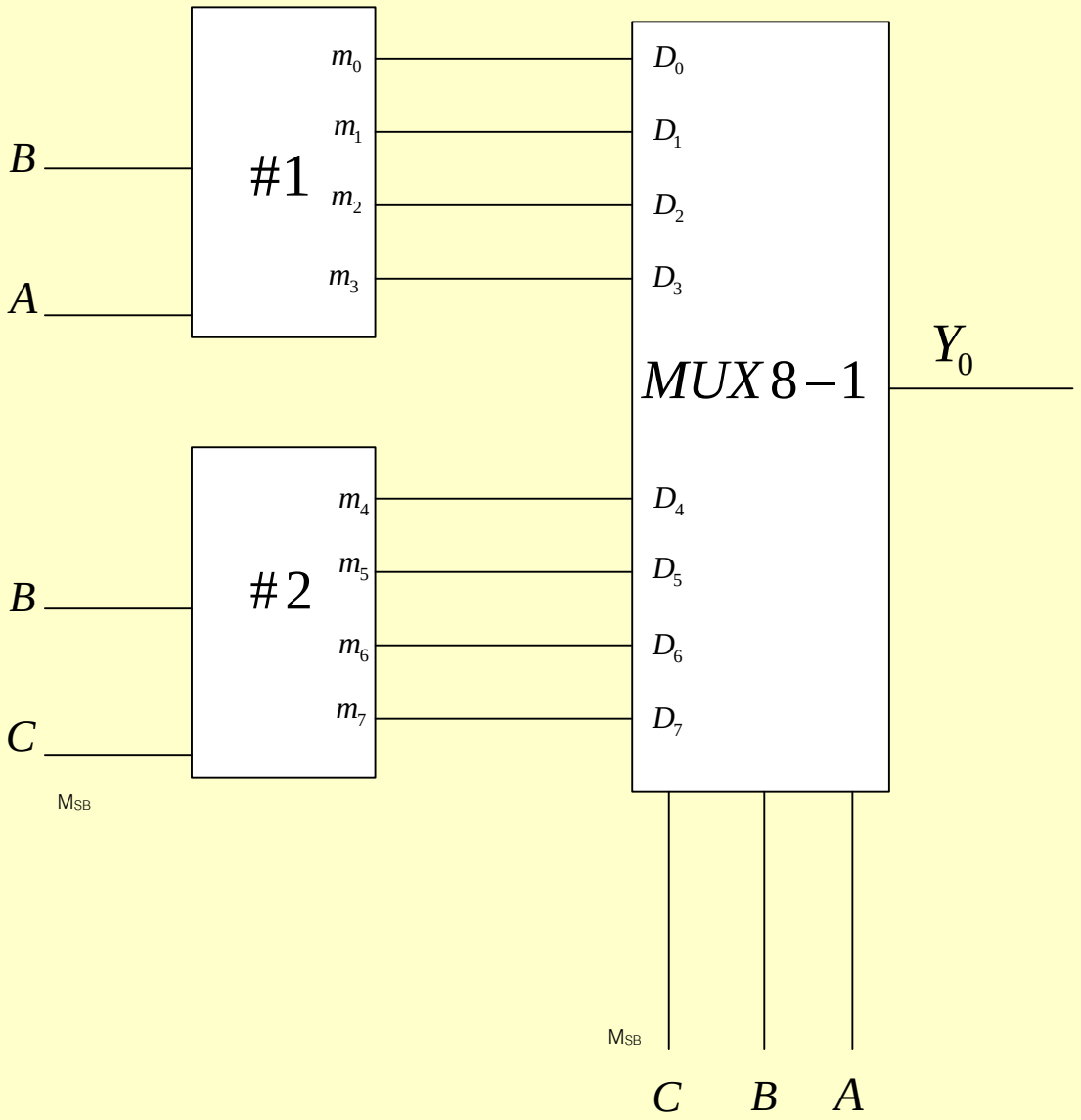
# ตารางการทำงานของวงจรมัลติเพล็กซ์

| i | $C = X_1$ | $B = X_2$ | $A = X_3$ | F = output |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 0 |

# อธิบาย

- จากค่าที่เกิดขึ้นทางเอาต์พุตสามารถนำมาเขียนเป็นตารางความจริงโดยกำหนด
- i เป็นข้อมูลที่จะส่งออกไปยังเอาต์พุต
- C = $X_3$ คือค่าตัวแปรที่ควบคุมการทำงานที่บิต3 ($M_{SB}$)
- B = $X_2$ คือค่าตัวแปรที่ควบคุมการทำงานที่บิต2
- A = $X_1$ คือค่าตัวแปรที่ควบคุมการทำงานที่บิต1 ($L_{SB}$)
- F = Output ที่เกิดจากการควบคุมของ A,B,C ที่มีค่าเท่ากับ 1 จะทำให้เกิดผลที่เอาต์พุต เป็นค่าของ $D_0$, $D_2$, $D_3$, $D_5$

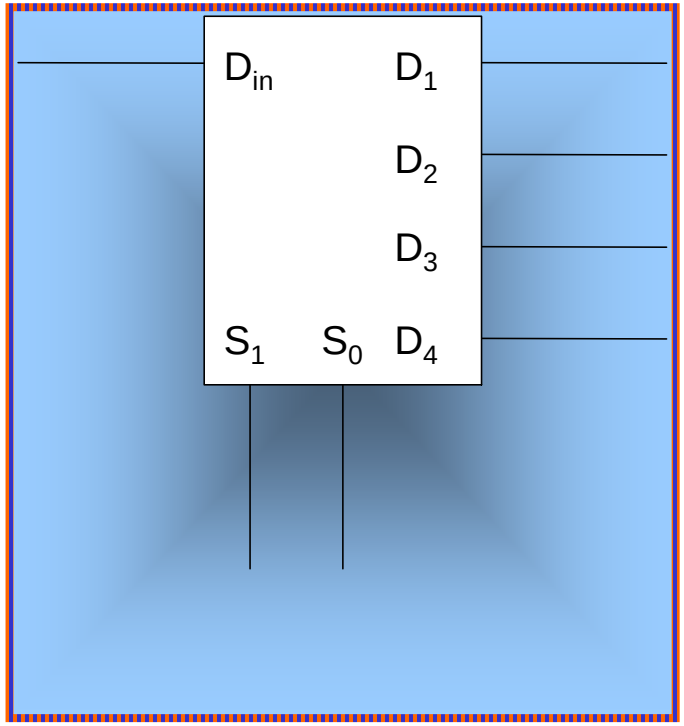
# ตัวอย่างการนำวงจรถอดรหัสทำงานร่วมกับวงจรมัลติเพล็กซ์

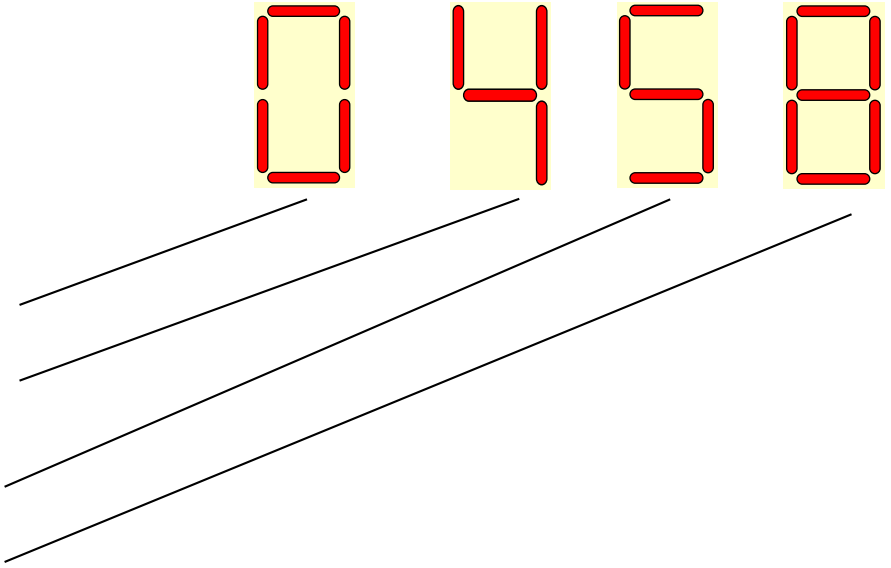


# อธิบาย

- การนำวงจรถอดรหัสที่สร้างสัญญาณทางอินพุตในรูปแบบ minterm
- ตัวอย่างให้วิเคราะห์หาค่า Minterm จากโมดุลที่ให้มา
- #1 เป็นวงจรดีโค้ดเด้อร์ขนาด 4 บิตมี A,B เป็นส่วนควบคุมที่บิตทางต่ำ สร้าง$m_0$-$m_3$
- #2 เป็นวงจรถอดรหัสขนาด 4 บิตมี B,C เป็นส่วนควบคุมที่บิตทางสูง สร้าง $m_4$-$m_7$
- เอาต์พุตของโมดุลทั้งสองส่งไปยังอินพุตของวงจรมัลติเพล็กซ์ขนาด 8 ข้อมูลเป็น1ข้อมูล โดยมีสัญญาณ A, B, C ควบคุมการทำงาน

# 48 หลักการทำงานของดีมัลติเพล็กซ์

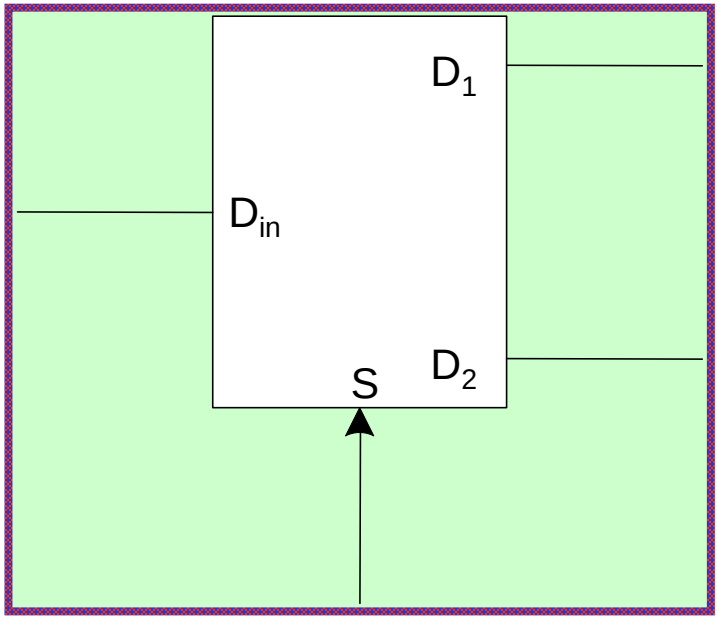


# อธิบาย

- หลักการทำงานของวงจรดีมัลติเพล็กซ์มีหลักการที่ตรงข้ามกับวงจรมัลติเพล็กซ์จึงต้องนำมาใช้งานร่วมกัน
- ข้อมูลทางอินพุตที่ถูกส่งใน $D_{in}$ ที่ผ่านวงจรมัลติเพล็กซ์ส่งมา จะเป็นเลขรหัส BCD มีค่าเท่ากับ 0 4 5 8 หมายถึงในช่วงเวลา $T_1$ จะส่งเลข 0 ช่วงเวลา $T_2$ จะส่งเลข 4 ช่วงเวลา $T_3$ จะส่งเลข 5 ช่วงเวลา $T_4$ ส่งเลข 8 ไปยังวงจรดีมัลติเพล็กซ์
- ข้อมูล $D_{in}$ ซึ่งเป็นรหัส BCD 0 จะถูกส่งไปที่ $D_1$ โดยการเลือกของ$S_1$ , $S_0$ มีค่ากับ 00 ,ช่วงเวลาที่ส่งเลข 4 จะถูกส่งค่าไปที่$D_2$ ค่า$S_1$ , $S_0$จะมีค่าเป็น 01, ช่วงเวลาที่ส่งเลข 5 ค่า $S_1$ , $S_0$ จะมีค่าเป็น 10 , ช่วงเวลาที่ส่งเลข 8 ค่า $S_1$ , $S_0$ จะมีค่าเป็น 11

# 481 หลักการวงจรดีมัลติเพล็กซ์ขนาด 1 ช่องข้อมูลทางอินพุตเป็น 2 ช่องข้อมูลทางเอาต์พุต




| S | $D_1$ | $D_2$ |
|---|---|---|
| 0 | $D_{in}$ | 0 |
| 1 | 0 | $D_{in}$ |

**ตารางแสดงการทำงาน**

$$D_1 = D_{in}\,\overline{S}$$

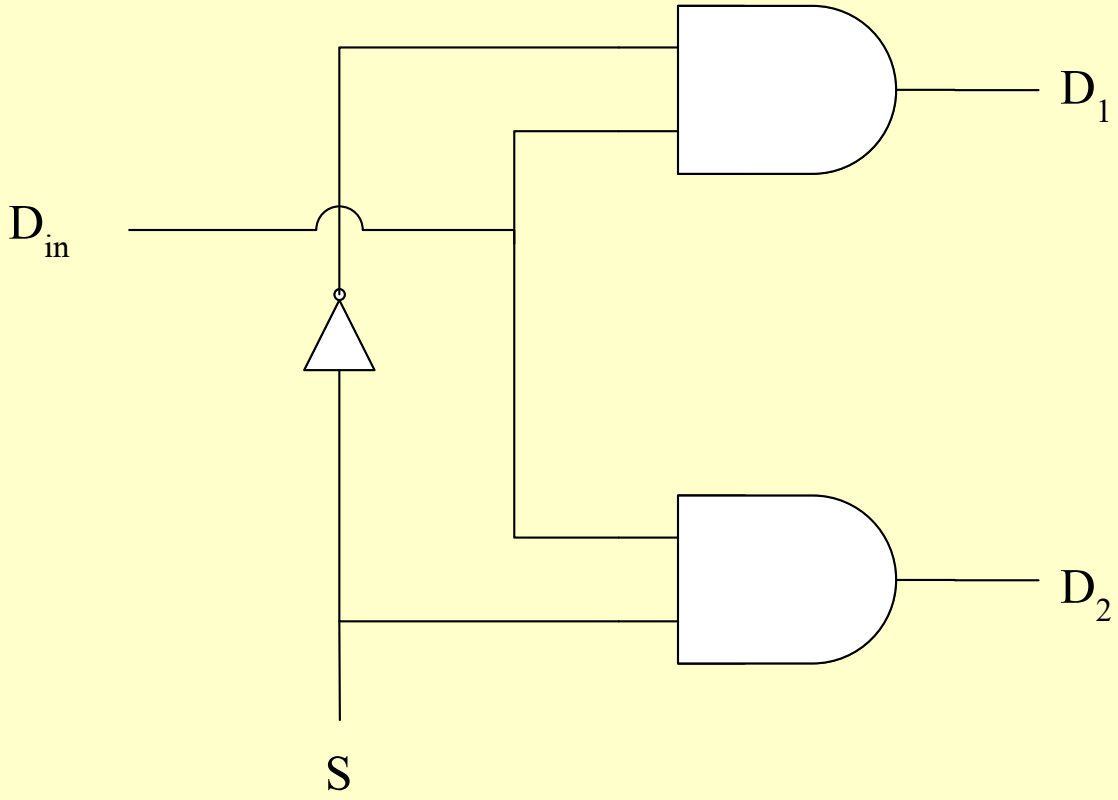
$$D_2 = D_{in}\,S$$

# อธิบาย

ข้อมูลที่ถูกส่งมาทาง $D_{in}$ เป็นข้อมูล $D_1$, $D_2$

- S เป็นตัวเลือกค่าทางเอาต์พุตจะมีค่าเป็น 0,1
- ถ้า S เป็น0 จะเลือกสัญญาณของ $D_1$
- ถ้า S เป็น1 จะเลือกสัญญาณของ $D_2$
- เอาต์พุต $D_1$ จึงมีค่าเป็น $D_{in}$ $\bar{S}$
- เอาต์พุต $D_2$ จึงมีค่าเป็น $D_{in}$ S

# วงจรลอจิก 1 ข้อมูลเป็น 2 ข้อมูล



# อธิบาย

- จากรูปวงจรดีมัลติเพล็กซ์หรือบางตำราเรียกว่าวงจรเลือกสัญญาณขนาด 2 ช่องสัญญาณทางเอาต์พุต
- การควบคุมสัญญาณของข้อมูลที่ส่งไปยังเอาต์พุตใช้ AND เกต จำนวน 2 ตัว
- การมัลติเพล็กซ์2 ค่า สามารถใช้ตัวอินเวอร์เตอร์ทำงานร่วมกับ AND เกต เมื่อสัญญาณ S = 0 ข้อมูลจะถูกเลือกเป็น $D_1$ สัญญาณ $D_{in} = D_1$ เมื่อสัญญาณ S = 1 ข้อมูลจะถูกเลือกเป็น $D_2$ ข้อมูลทางอินพุตจะถูกส่งไปยังเอาต์พุต $D_2$

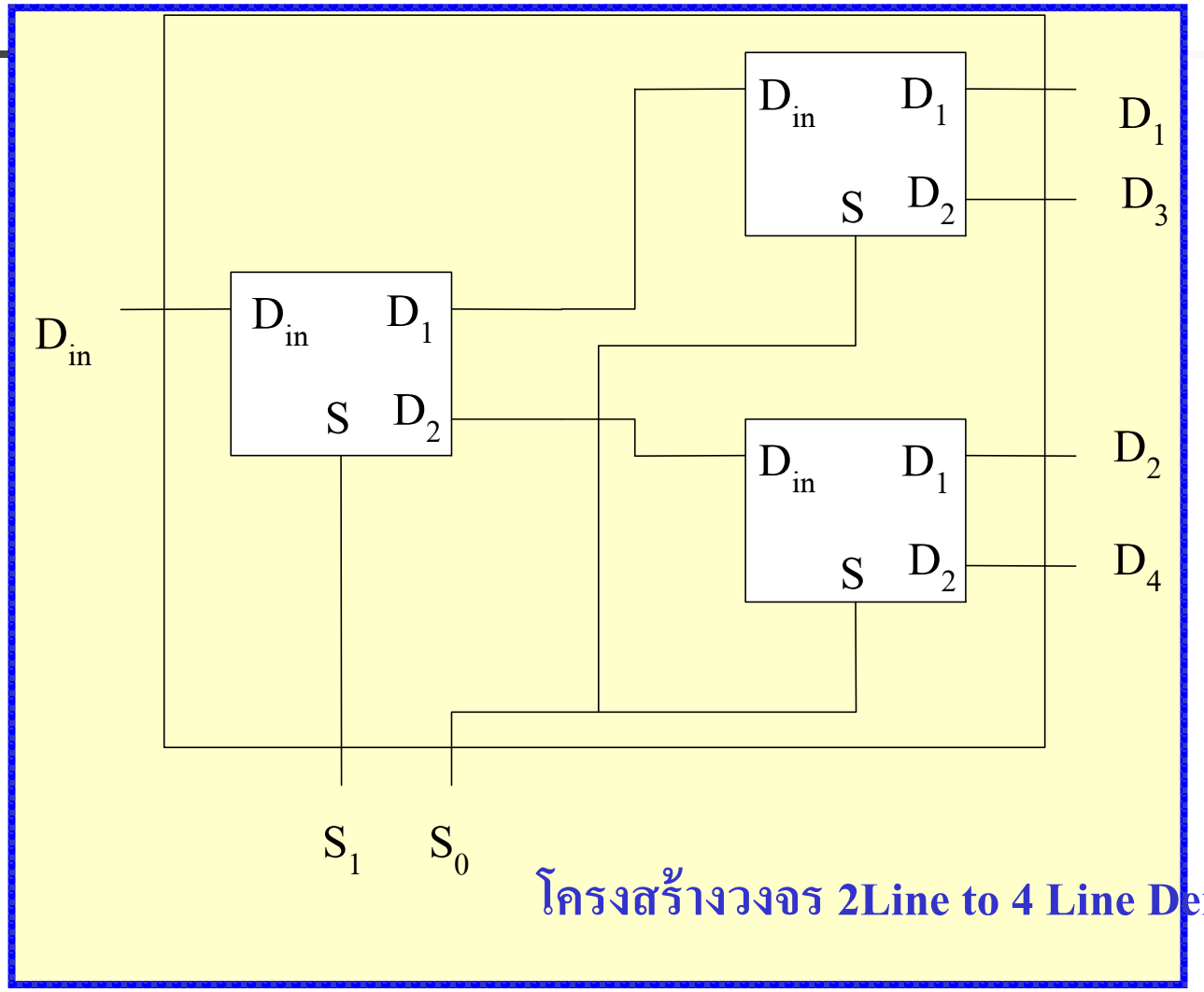
# ตารางการทำงานของวงจรดีมัลติเพล็กซ์ 4 อินพุต

| $S_0$ | $S_1$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ |
|---|---|---|---|---|---|
| 0 | 0 | $D_{in}$ | 0 | 0 | 0 |
| 0 | 1 | 0 | $D_{in}$ | 0 | 0 |
| 1 | 0 | 0 | 0 | $D_{in}$ | 0 |
| 1 | 1 | 0 | 0 | 0 | $D_{in}$ |

# อธิบาย

- จากตารางการทำงานของวงจรดีมัลติเพล็กซ์ 4 เอาต์พุต
- $S_0$, $S_1$ เป็นตัวเลือกสัญญาณทางอินพุต มีค่าเท่ากับ $2^2 = 4$ ค่า นั่นหมายความว่าสามารถเลือกสัญญาณได้ 4 ค่า
- $D_{in}$ เป็นข้อมูลทางอินพุตที่ถูกรวมกันมา 4 ช่องสัญญาณจากวงจร มัลติเพล็กซ์แล้วถูกส่งไปยังวงจรดีมัลติเพล็กซ์ โดยมี $S_0$, $S_1$ เป็นตัวเลือกสัญญาณทางเอาต์พุตให้ตรงกับช่องสัญญาณที่มาจากอินพุต
- $D_1$,$D_2$,$D_3$,$D_4$เป็นข้อมูลทางเอาต์พุต ที่เกิดขึ้นจากการเลือกของ $S_0$, $S_1$ เช่นถ้า $S_0$, $S_1 = 00$ ข้อมูลจะเป็นของ $D_1$ ถ้า$S_0$, $S_1 = 10$ ข้อมูลก็จะเป็นของ $D_2$ เป็นต้น

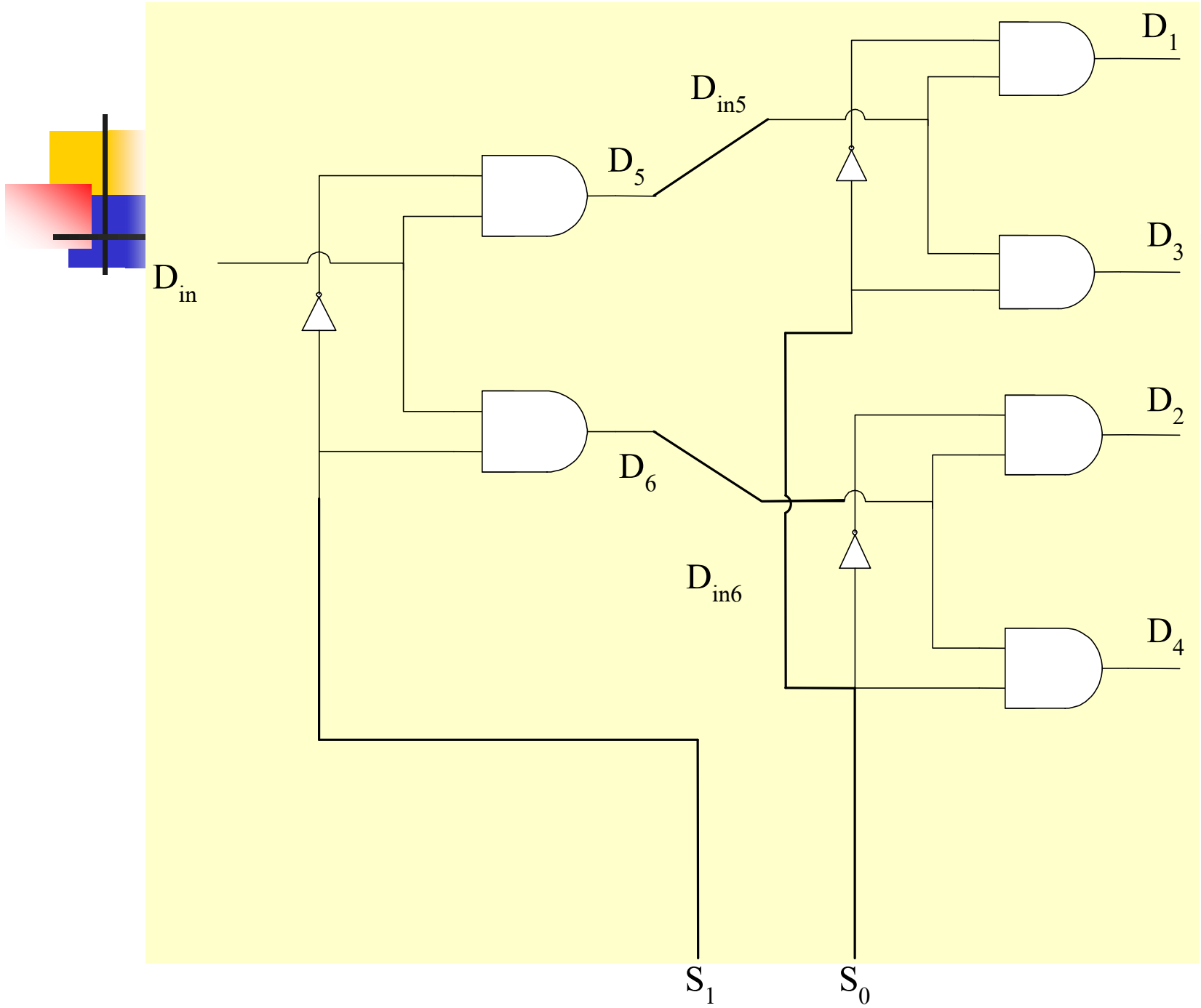
# 482หลักการวงจรดีมัลติเพล็กซ์ขนาด 1 ช่องข้อมูลทางอินพุต เป็น 4 ช่องข้อมูลทางเอาต์พุต




โครงสร้างวงจร 2Line to 4 Line Demultiplex

# อธิบาย

- จากตารางการทำงานของวงจรดีมัลติเพล็กซ์ 4 เอาต์พุต
- สามารถนำวงจรเลือกข้อมูล 2 เอาต์พุต มาต่อกันดังรูป
- $S_0$ ($M_{SB}$) จะมีค่าเป็น 0 จะเลือกส่งข้อมูล $D_1$,$D_2$
- $S_0$ ($M_{SB}$) จะมีค่าเป็น 1 จะเลือกส่งข้อมูล $D_3$,$D_4$
- $S_1$ ($L_{SB}$) จะมีค่าเป็น 0 จะถูกบังคับข้อมูลที่ $D_1$, $D_3$ถ้ามีค่าเป็น 1 จะบังคับข้อมูลที่ $D_2$ , $D_4$
- ถ้า $S_0$,$S_1$= 00 ข้อมูล $D_1$ จะถูกแยกไปยังเอาต์พุต
- ถ้า $S_0$,$S_1$= 01 ข้อมูล $D_2$ จะถูกแยกไปยังเอาต์พุต
- ถ้า $S_0$,$S_1$= 10 ข้อมูล $D_3$จะถูกแยกไปยังเอาต์พุต
- ถ้า $S_0$,$S_1$= 11 ข้อมูล $D_4$ จะถูกแยกไปยังเอาต์พุต


D1
Din5
D5
Din
D3
D2
D6
Din6
D4
S1
S0

# การออกแบบการดีมัลติเพล็กซ์ 4 ข้อมูล

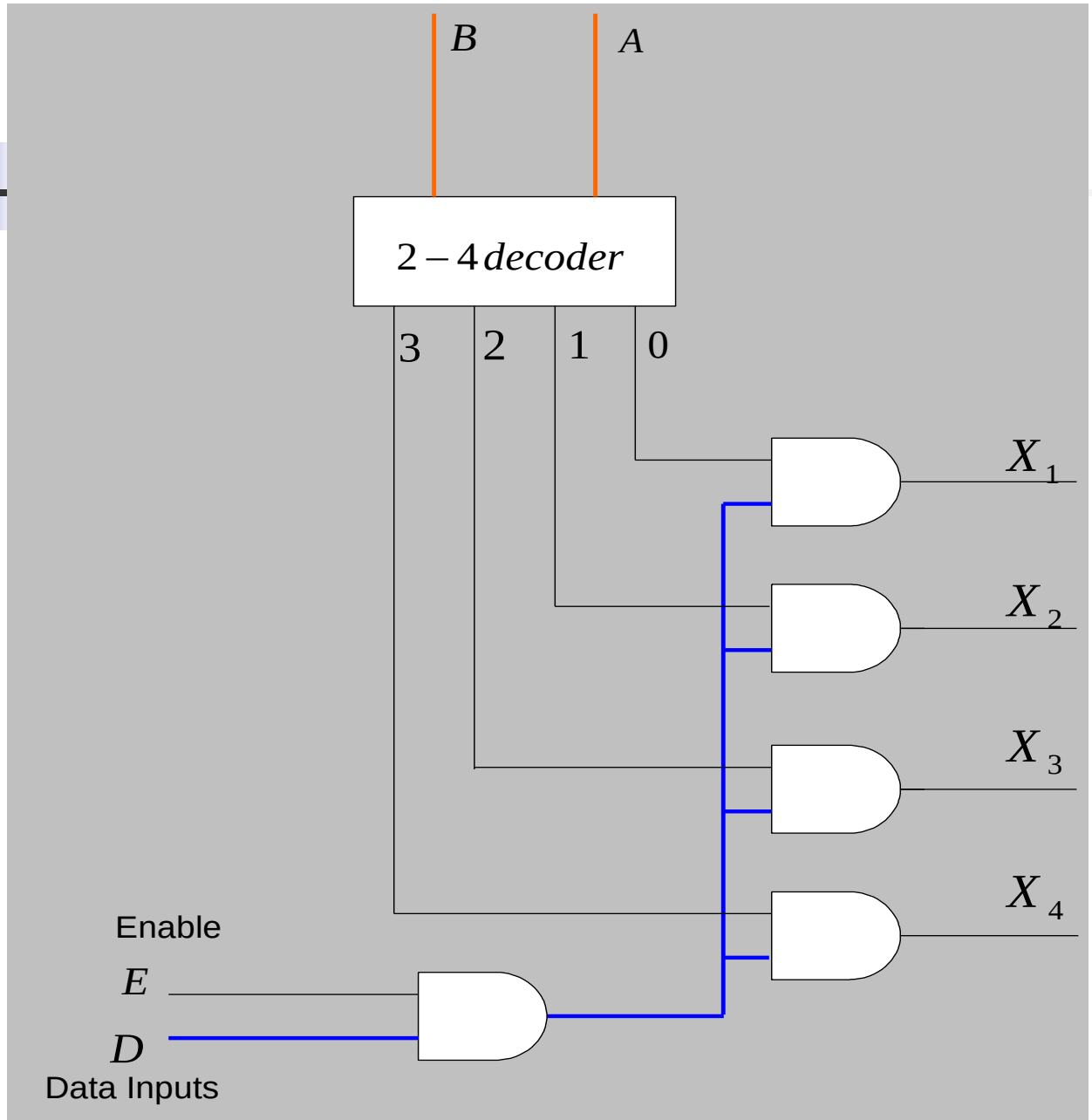
| ตำแหน่งที่ควบคุม | | INPUT | OUTPUT |
|---|---|---|---|
| A | B | DATA | $F_0$ |
| 0 | 0 | $D_0$ | $\bar{A}\bar{B}\ D_0$ |
| 0 | 1 | $D_1$ | $\bar{A}B\ D_1$ |
| 1 | 0 | $D_2$ | $A\bar{B}\ D_2$ |
| 1 | 1 | $D_3$ | $AB\ D_3$ |

# อธิบาย

- กำหนด การควบคุมข้อมูลด้วยตัวแปรทางด้านอินพุตคือ A, B
- ค่า A, B จะถูกเปลี่ยนไปตามเลขฐานสองคือ 00, 01, 10, 11 ตามลำดับ
- ค่าอินพุตของข้อมูลเป็น $D_0, D_1, D_2, D_3$,ค่าเอาต์พุตเป็น $X_1,X_2,X_3,X_4$
- ค่าอินพุตของข้อมูลจะถูกส่งออกไปที่เอาต์พุตเมื่อมีการควบคุมที่ A , B ตามตารางความจริงดังนั้น เอาต์พุตจะได้ดังสมการ

$$F_0 = \overline{A}\,\overline{B}\,D_0 + \overline{A}BD_1 + A\,\overline{B}\,D_2 + ABD_3$$

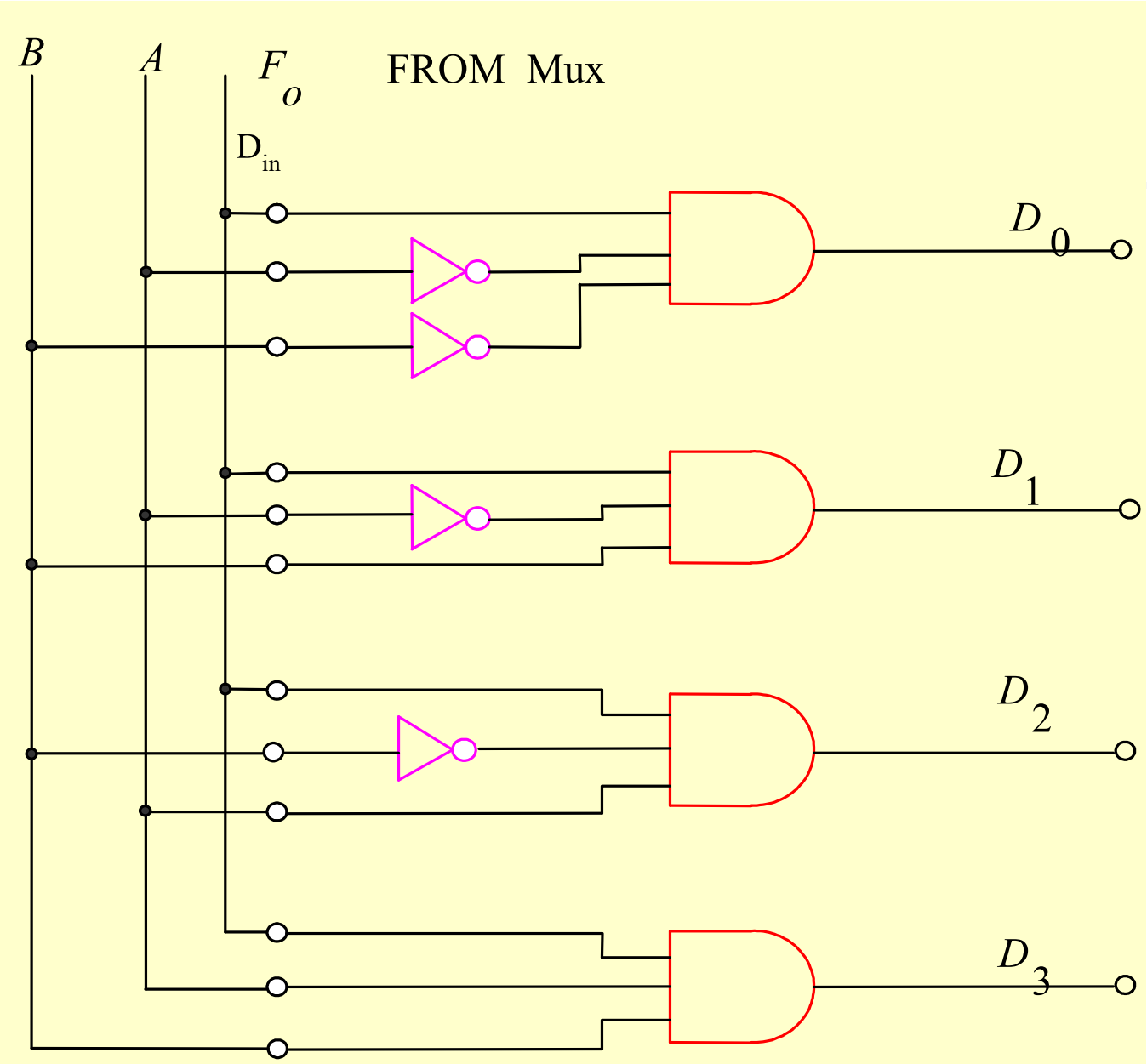
# 483 การออกแบบวงจรดีมัลติเพล็กซ์ขนาด 4 เอาต์พุต



# อธิบาย

- วงจรดีมัลติเพล็กซ์4 ช่องพร้อมด้วยสัญญาณ Enable
- จากรูปวงจรดีมัลติเพล็กซ์แบบนี้ประกอบด้วย 2 ส่วนคือ ส่วนดีโค้ดเดอร์กับส่วนของอินพุตข้อมูลโดยใช้วงจรถอดรหัสเป็นตัวเลือกช่องข้อมูลที่ส่งออกจะทำงานสัมพันธ์กับตัวเลือกข้อมูลทางอินพุต
- Enable คือสัญญาณที่ทำหน้าที่เพื่อควบคุมโมดุลให้ทำงาน
- สัญญาณรวมที่ส่งเป็น Data ทางอินพุตมาจากวงจรเลือกข้อมูล จะถูกเลือกโดยสัญญาณ Enable มีค่าเป็น 1 ข้อมูลจะไปปรากฏที่แอนด์เกต ที่มีสัญญาณจากดีโค้ดเดอร์มาควบคุมเพื่อแยกที่เอาต์พุต

# วงจรดีมัลติเพล็กซ์



# อธิบาย

- อีกรูปแบบของการทำงานของวงจรดีมัลติเพล็กซ์ขนาด 4 ช่องข้อมูล
- พิจารณาจากตารางความจริงของวงจรดีมัลติเพล็กซ์
- $D_0$ จะถูกควบคุมการทำงานโดย $\bar{A}\bar{B}$
- $D_1$จะถูกควบคุมการทำงานโดย $\bar{A}B$
- $D_2$จะถูกควบคุมการทำงานโดย $A\bar{B}$
- $D_3$จะถูกควบคุมการทำงานโดย $AB$
- $F_0$ มาจากค่าเอาต์พุตของวงจรเลือกข้อมูล
- $D_0,D_1,D_2,D_3$ จะถูกแยกจาก$F_0$ โดยค่า A , B

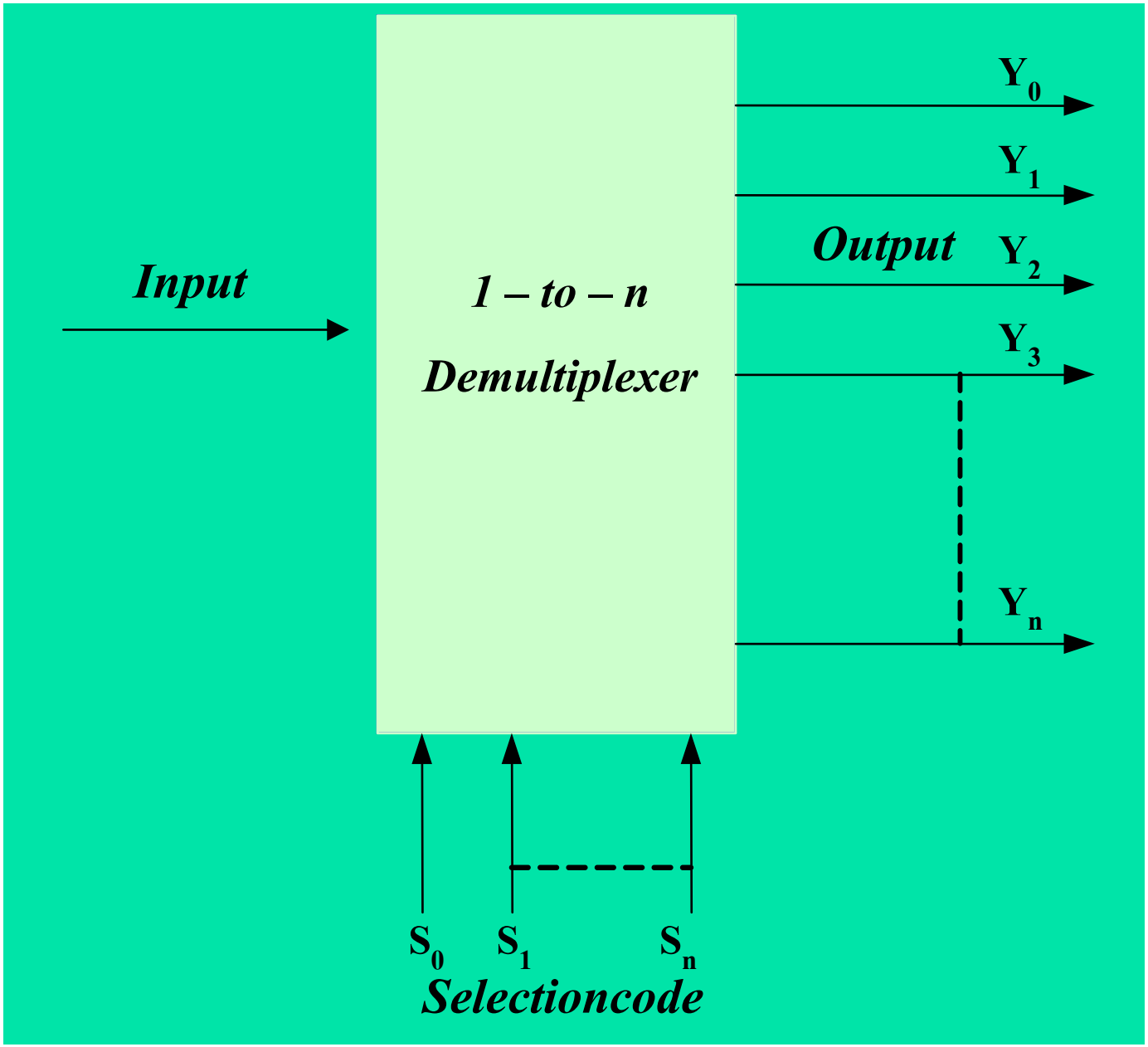
# ตัวอย่างตารางการออกแบบวงจรดีมัลติเพล็กซ์ขนาด 8 บิต

| C | B | A | $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | $Y_0$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | **1** |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | **1** |
| **1** | **0** | 0 | 0 | 0 | 0 | 0 | **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | 1 | 0 | 0 | 0 | 0 | **0** | **1** | **0** | **0** | **1** |
| **1** | **1** | 0 | 0 | 0 | 0 | 0 | **0** | **0** | **1** | **0** | **1** |
| **1** | **1** | 1 | 0 | 0 | 0 | 0 | **0** | **0** | **0** | **1** | **1** |

# อธิบาย

- จากตัวอย่างที่ผ่านมาตารางการทำงานของวงจรดีโค้ดเดอร์และวงจรเลือกข้อมูลและดีมัลติเพล็กซ์ขนาด 8 บิตจะเหมือนกันในส่วนของการถอดรหัสตำแหน่งลองพิจารณาดู
- พิจารณาวงจรถอดรหัส$D_0$-$D_3$ เกิดจากบิตต่ำ A, B มีการเปลี่ยนแปลงเอาต์พุตที่เกิดเป็นดังตารางความจริง
- การ $D_4$-$D_7$ เกิดจากบิตสูง C ที่ทำงานร่วมกับA,B
- วงจรเลือกข้อมูลมีเอาต์พุตเป็น $Y_0$ ถอดรหัส ตำแหน่งเป็นA,B,C เหมือนกับวงจรดีโค้ดเด้อร์ที่ผ่านมา
- วงจรดีมัลติเพล็กซ์ก็จะมีส่วนที่เหมือนกับวงจรดีถอดรหัสเป็นตัวถอดรหัสA,B,C แยกเอาต์พุตเป็น $D_0$-$D_7$
- สามารถประยุกต์การใช้วงจรทั้งสามเข้าด้วยกัน

# 484 การออกแบบวงจรดีมัลติเพล็กซ์ขนาด n' Bit

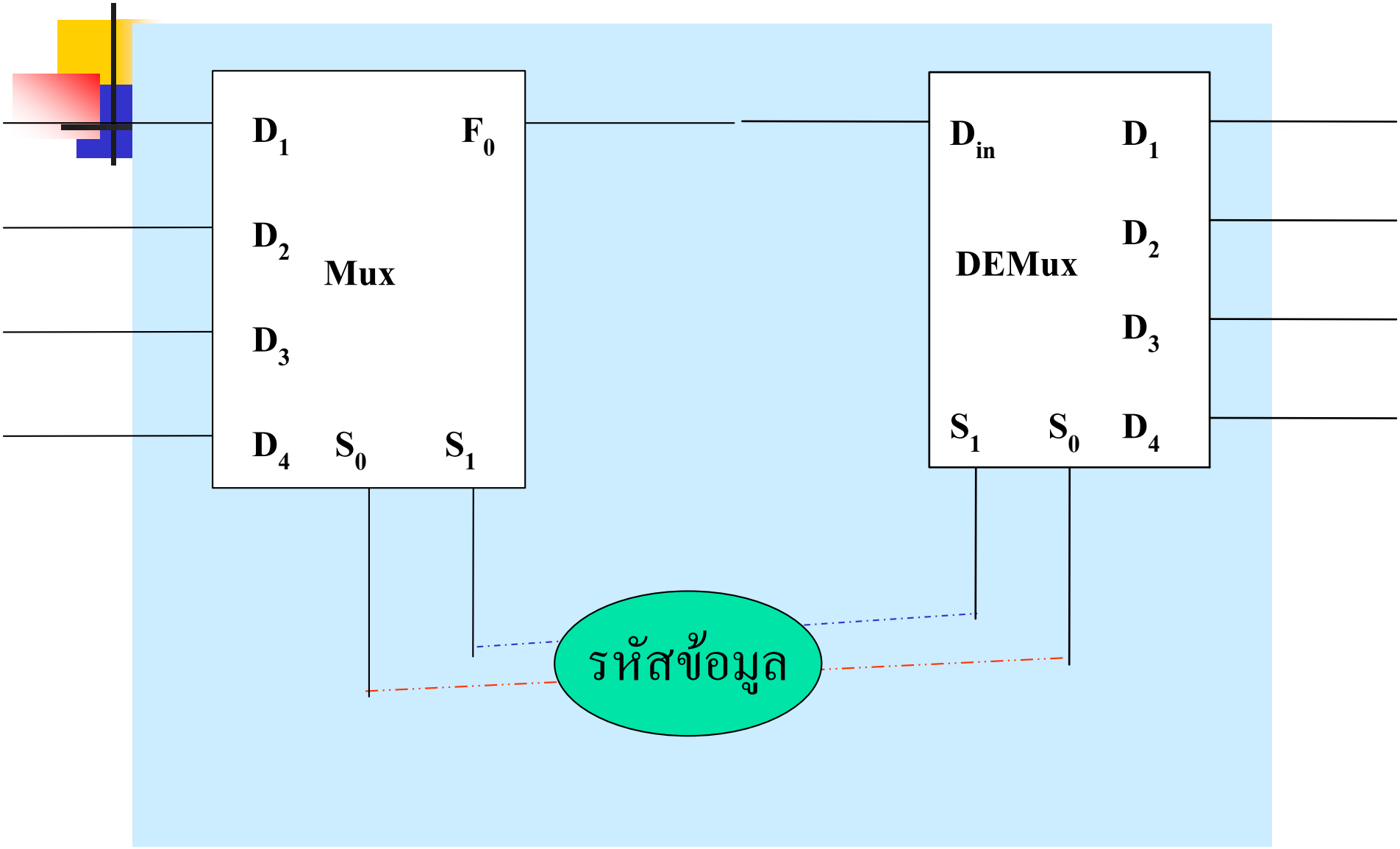


# อธิบาย

- หลักการของดีมัลติเพล็กซ์ขนาด n บิต
- สัญญาณทางอินพุตจะป้อนเข้าวงจรเพียงอินพุตเดียว
- สัญญาณทางเอาต์พุตจะมีค่าเท่ากับ $Y_n$ บิต
- สัญญาณเลือกค่าทางเอาต์พุตจะมีจำนวน $2^n$ บิต เท่ากับจำนวนค่าทางเอาต์พุต

  - ตัวอย่าง $2^n = Y_n$
- ต้องการเอาต์พุต 16 บิต $Y_n=16$ ดังนั้น n จะมีค่า 4 บิต
- นั่นหมายความว่า Selection code ต้องใช้ถึง 4 บิต จึงจะสามารถดีมัลติเพล็กซ์ออกทางเอาต์พุตเป็นจำนวน 16 บิต

# 49 การนำวงจรมัลติเพล็กซ์มาใช้งานร่วมกับวงจรดีมัลติเพล็กซ์



# การนำวงจรมัลติเพล็กซ์มาใช้ร่วมกับวงจรดีมัลติเพล็กซ์

- ด้านซ้ายของวงจรเป็นโมดุลของวงจรมัลติเพล็กซ์ขนาด 4 อินพุต เอาต์พุตส่งออกจำนวน 1 เอาต์พุต ไปยังวงจรถอดรหัส
- ส่วนทางด้านขวามือเป็นโมดุลของวงจรดีมัลติเพล็กซ์ขนาด 1 อินพุต เอาต์พุตดีมัลติเพล็กซ์ตามอินพุตส่งมาเป็น $D_1$, $D_2$, $D_3$, $D_4$ ตามการควบคุมของบิต $S_0$, $S_1$
- การทำงานในส่วนควบคุมการส่งข้อมูลและการรับข้อมูลจะเป็นไปดังนี้

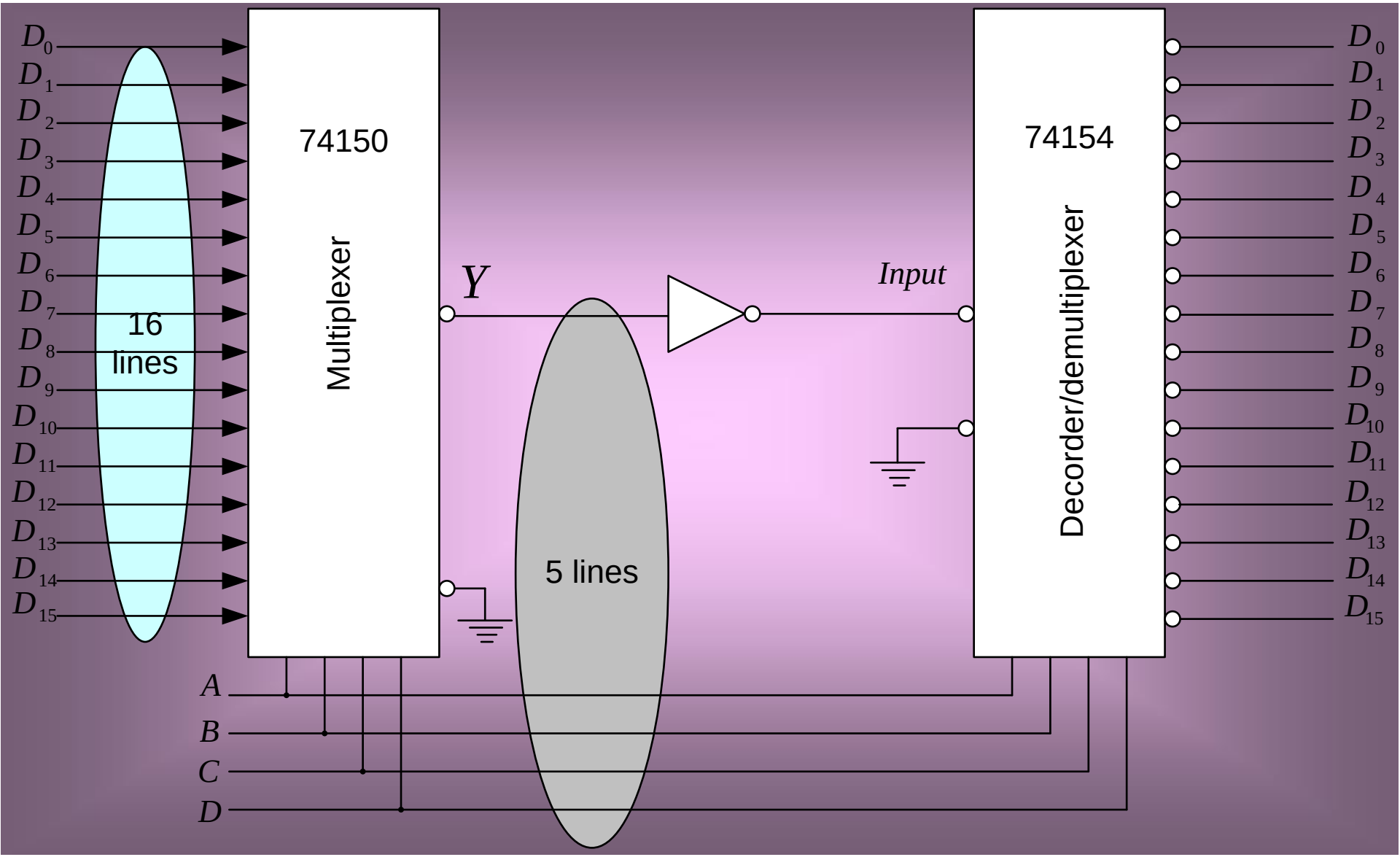
$S_1$, $S_0$ ควบคุมการส่งข้อมูล $D_1$ จากอินพุตไปยังเอาต์พุต เมื่อมีค่าเท่ากับ 00

$S_1$, $S_0$ ควบคุมการส่งข้อมูล $D_2$ จากอินพุตไปยังเอาต์พุต เมื่อมีค่าเท่ากับ 01

$S_1$, $S_0$ ควบคุมการส่งข้อมูล $D_3$ จากอินพุตไปยังเอาต์พุต เมื่อมีค่าเท่ากับ 10

$S_1$, $S_0$ ควบคุมการส่งข้อมูล $D_4$ จากอินพุตไปยังเอาต์พุต เมื่อมีค่าเท่ากับ 11

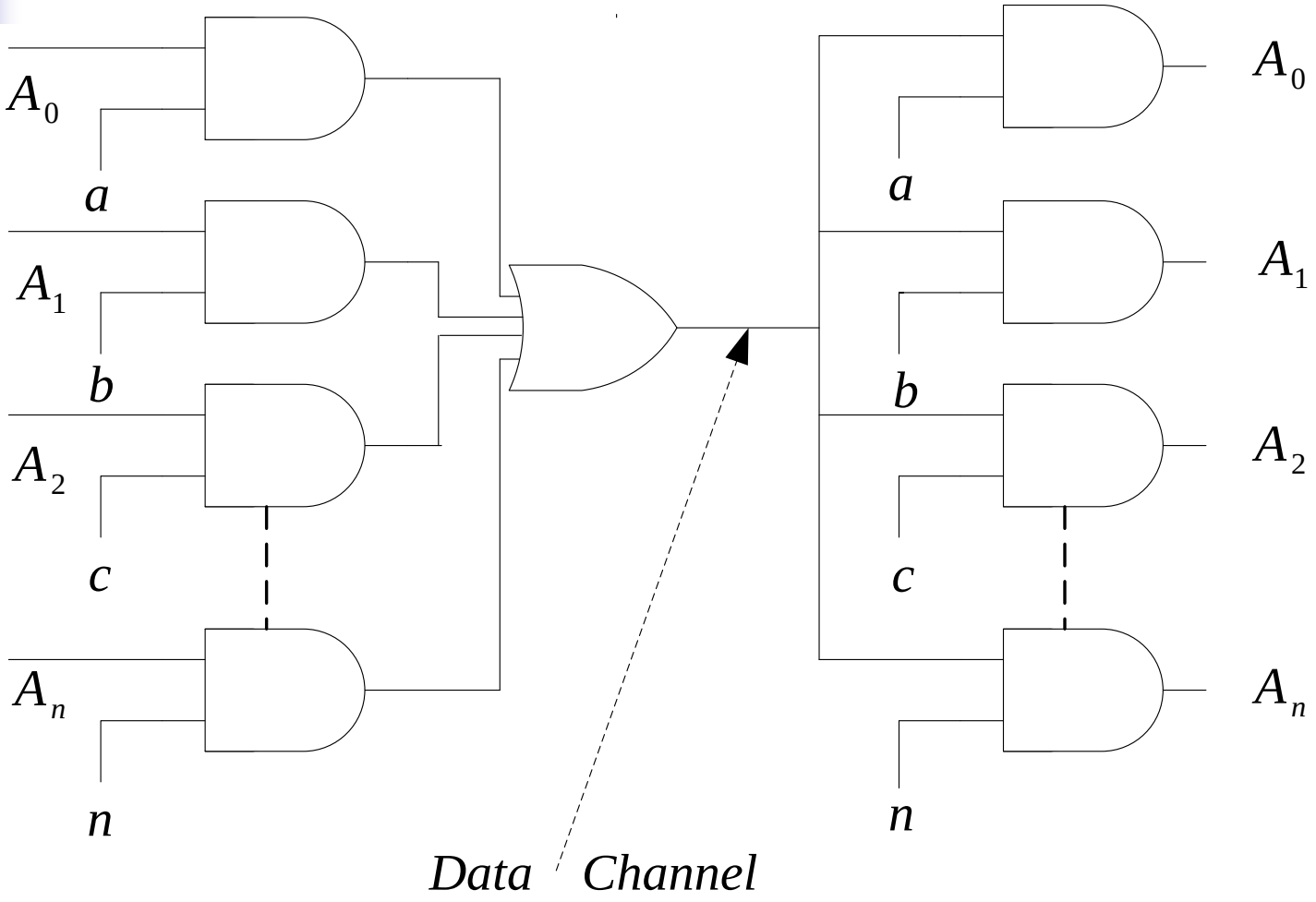
# 491 การนำวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์มาต่อใช้งาน



# อธิบาย

- ตัวอย่างการนำวงจรมัลติเพล็กซ์ และวงจรดีมัลติเพล็กซ์มาใช้งานร่วมกันจำนวน 16 บิต
- สามารถใช้ได้ 16-line to 16-line อินพุต/เอาต์พุต
- ความสามารถของระบบที่สามารถแทนสายเคเบิล 16เส้น
- ระยะทางในการส่งได้ไกลถึง 1000 ฟุต
- รหัสที่ในการดีโค้ดเดอร์สัญญาณเป็น A, B, C, D ขนาด 4 บิต
- สายที่จำเป็นที่ใช้ในระบบจำนวน 5 เส้น ทำให้เกิดการประหยัดสายในการส่ง

# 492การส่งข้อมูลและกระจายข้อมูลขนาด n บิต

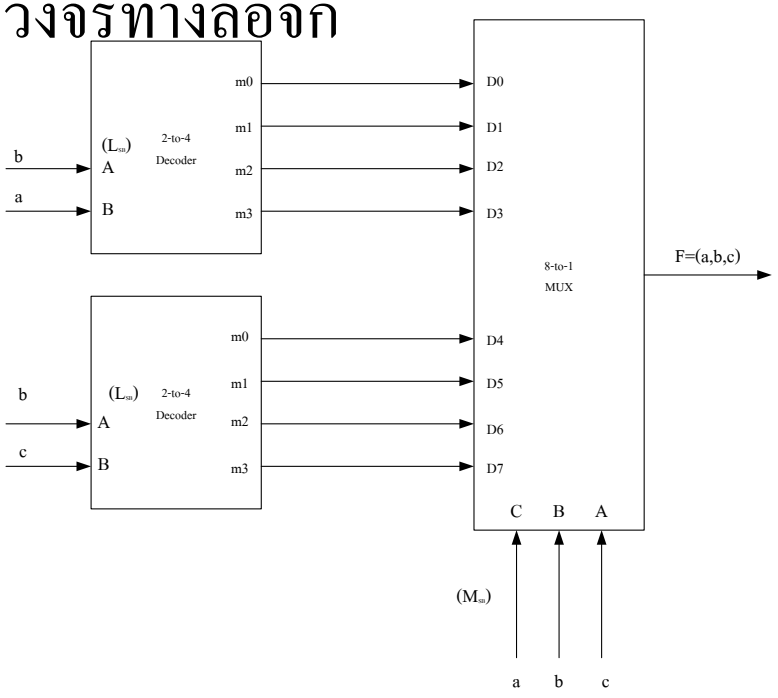


# อธิบาย

- การส่งข้อมูลรวม และการกระจายข้อมูลจำนวน n บิตประกอบด้วย
- วงจรรวมข้อมูลจำนวน n บิต $A_0$-$A_n$ ทำหน้าที่รวมข้อมูลจากบิต 0ถึงบิต n ส่งเข้าออร์เกต เพื่อรวมเป็นเอาต์พุตเดียว แต่ส่งค่าเวลาที่ต่างกันทำให้ข้อมูลสามารถแยกจากกันทางเอาต์พุต ข้อดีนี้ทำให้ประหยัดสายในการส่ง
- ช่องส่งข้อมูล (Data Channel) เป็นช่องทางเดินของข้อมูลทุกบิตโดยใช้สายเพียงหนึ่งเส้นร่วมกับกราวนด์
- วงจรกระจายข้อมูล n บิต $A_0$-$A_n$ ทำหน้าที่รับข้อมูลที่มาจากช่องส่งข้อมูล และทำการดีมัลติเพล็กซ์ออกโดยการควบคุมของ ตัวควบคุมข้อมูล
- ตัวควบคุมข้อมูล a-n ทำหน้าที่ทั้งการรับและการส่งเป็นข้อมูลที่ต้องซิงโครนัสกัน หมายความถึงเมื่อมีการส่ง $A_0$ สัญญาณควบคุมจะเกิดขึ้นที่ a ที่วงจรส่งและวงจรรับ นั่นหมายความว่าข้อมูล $A_0$ ก็จะปรากฏทางเอาต์พุต

# ทดสอบความรู้สัปดาห์ที่ 7

1. อธิบายถึงหลักการวงจรมัลติเพล็กซ์/ดีมัลติเพล็กซ์ สามารถนำไปใช้ในการทำงานในระบบดิจิตอลอย่างไร
2. จงยกตัวอย่างของการนำวงจรมัลติเพล็กซ์ และดีมัลติเพล็กซ์ไปใช้ในการออกแบบระบบคอมพิวเตอร์ในส่วนหน่วยเก็บข้อมูล และหน่วยคำนวณ
3. ให้นักศึกษานำความรู้ที่ได้จากที่เรียนมา ไปออกแบบวงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์ขนาด 16 ช่อง เขียนออกมาเป็นรูปวงจรทางลอจิก
4. จงหาค่า minterm ของฟังก์ชันในรูปที่ให้มา



# เนื้อหาคำบรรยายในสัปดาห์ที่ 8 วงจรเปรียบเทียบ และวงจร ALU

**51 การเปรียบเทียบทางแอนะลอก**

511 การนำวงจรเปรียบเทียบทางแอนะลอกไปใช้งานร่วมกับวงจรดิจิตอลในการควบคุมสัญญาณ

**52 วงจรเปรียบเทียบทางดิจิตอล**

521 การออกแบบวงจรเปรียบเทียบ1 บิต ทางดิจิตอล

522 การออกแบบวงจรเปรียบเทียบ 2 บิต ทางดิจิตอล

523 การออกแบบวงจรเปรียบเทียบ 4 บิต ทางดิจิตอล

524 การออกแบบวงจรเปรียบเทียบ n บิต ทางดิจิตอล

**53 หลักการของ ALU**

531 การทำงานของหน่วยลอจิก (LU)

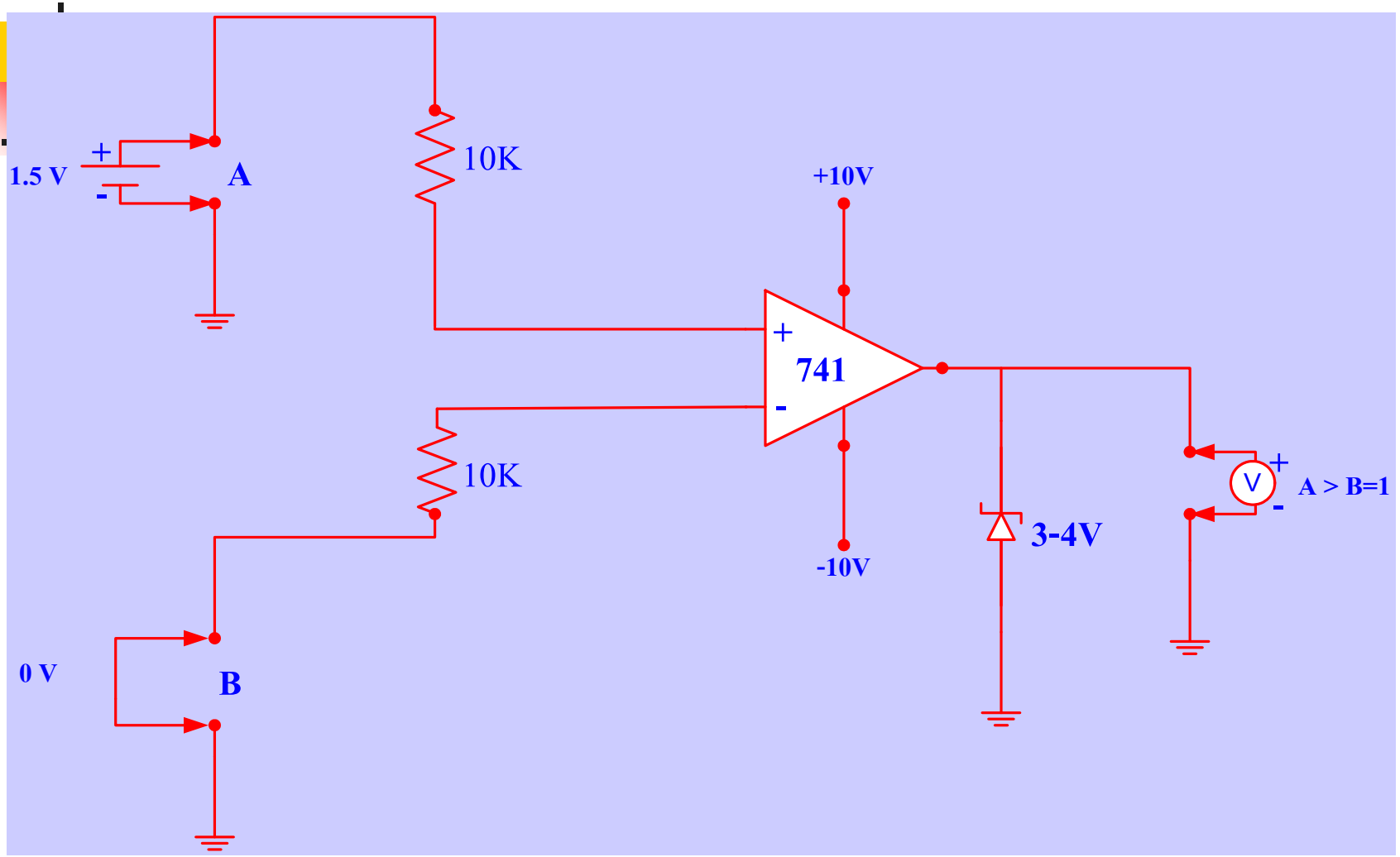
532 การทำงานของหน่วยคำนวณทางคณิตศาสตร์ (AU)

**54 หลักการออกแบบวงจร ALU ขนาด 1 บิต**

541 อธิบายการทำงานของวงจร ALU ขนาด 1 บิต

542 ออกแบบวงจร ALU ขนาด 1 บิต

# 51 การเปรียบเทียบทางแอนะลอก

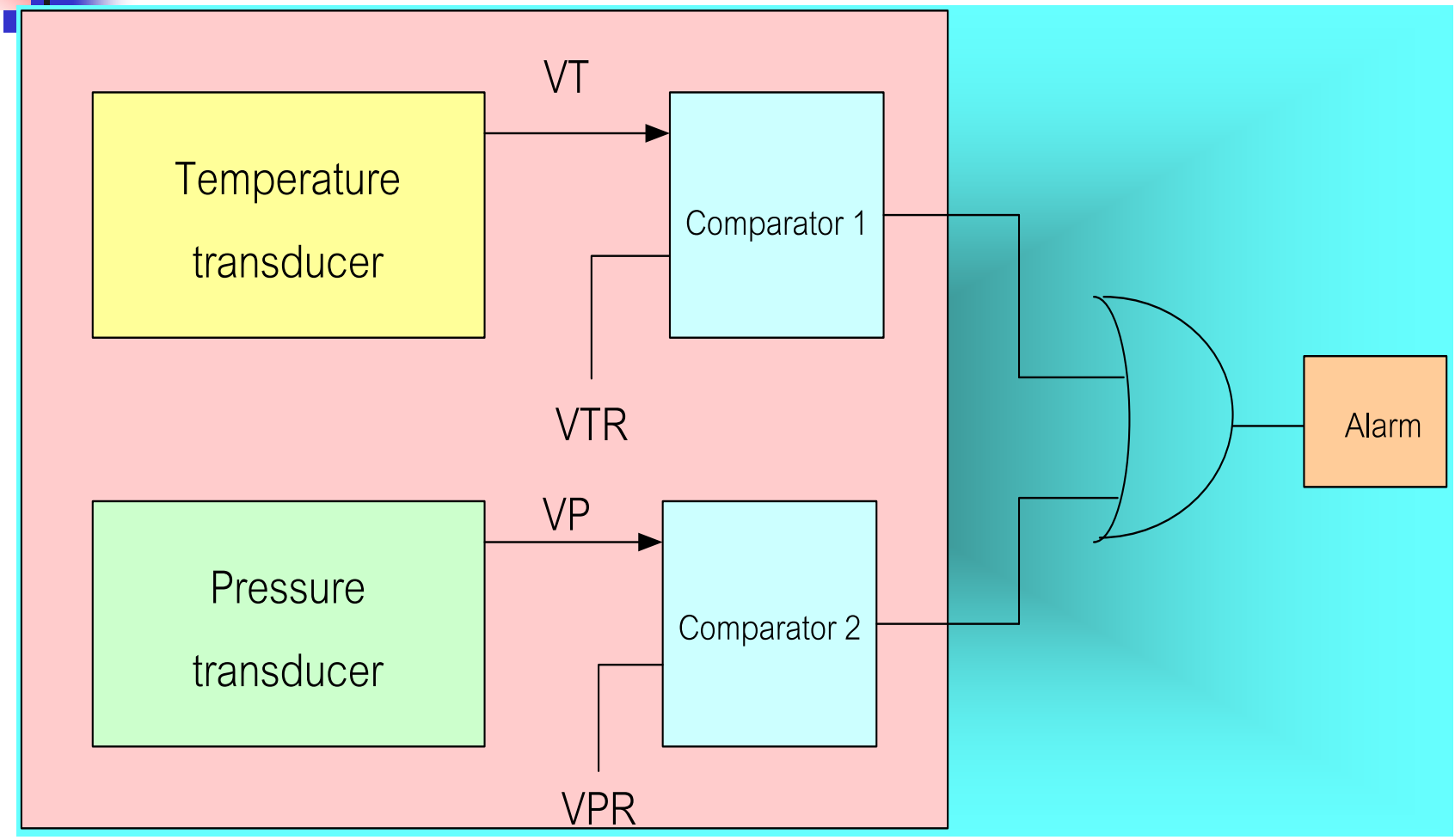



การเปรียบเทียบค่าทางอินพุตแบบแอนะลอก2อินพุตค่าต่างกันจะได้เอาต์พุตมีค่า 1

# อธิบาย

- การเปรียบเทียบแบบแอนะลอกโดยใช้ไอซีออปแอมป์
- หลักการโดยใช้วงจรขยายความแตกต่างทางอินพุตที่อยู่ภายในออปแอมป์เบอร์ 741
- ขั้วต่ออินเวอร์เตอร์(-) ต่อผ่านค่าความต้านทาน10กิโลโอห์มจุดB ลงไปอ้างอิงที่ 0 โวลต์
- ขั้วต่อนอนอินเวอร์เตอร์(+) ต่อผ่านค่าความต้านทาน 10 กิโลโอห์มจุดAไปที่อินพุต ซึ่งมีค่าแรงดันไฟตรง 1.5 โวลต์
- การทำงานของออปแอมป์เมื่อมีอินพุตป้อนมากกว่าจุดอ้างอิง B ค่าแรงดันที่เกิดขึ้นทางเอาต์พุตจะมีค่าเป็นบวกทิศทางเดียวโดยมีค่าซีเนอร์ไดโอดขนาด 3-4 โวลต์จำกัดค่าแรงดันทางเอาต์พุตไว้
- ผลของเอาต์พุตจะเป็น 3-4 โวลต์ ก็ต่อเมื่ออินพุต A ป้อนแรงดันสูงกว่าอินพุต B

# 511 การนำวงจรเปรียบเทียบแบบแอนะลอกไปใช้งานร่วมกับวงจรดิจิตอลในการควบคุมสัญญาณ



# อธิบาย

- การนำวงจรเปรียบเทียบมาใช้งานทางด้านระบบดิจิตอล
- อินพุตมีค่าตัวแปร2 ค่าที่เกิดขึ้น กับระบบนี้คือ ค่าอุณหภูมิ กับค่าแรงดัน
- ตัวทรานส์ดิวเซอร์ ทำหน้าที่ตรวจจับอุณหภูมิ จะเป็นสัญญาณแบบแอนะลอก โดยอุณหภูมิจะแปรค่าเป็นแรงดันที่ขึ้นตรงกับอุณหภูมิ (VT) ส่งไปยังตัวเปรียบเทียบ 1 ซึ่งทำงานแบบแอนะลอกที่กล่าวมา VTR (Voltage Temperature Reference) จะถูกส่งมาที่วงจรเปรียบเทียบ หากผลการเปรียบเทียบมีค่าของVT มากกว่า VTR นั่นหมายความว่าอุณหภูมิที่ตรวจจับสูงกว่าที่ตั้งไว้ comparator 1 จะส่งให้เอาต์พุตมีค่าเท่ากับ 1 ส่งไปยังวงจรสัญญาณเตือน
- ในทำนองเดียวกัน ตัวทรานส์ดิวเซอร์ที่ตรวจจับค่าแรงดันของอุปกรณ์สร้างแรงกด ของอุปกรณ์สร้างแรงดันในระบบค่าแรงดันถูกเปลี่ยนเป็นสัญญาณทางไฟฟ้าในรูปVPR (Voltage Pressure Reference) นั่นหมายความว่า ถ้าแรงดันในระบบมีค่ามากกว่าแรงดันที่ตั้งค่าไว้ วงจรเปรียบเทียบ 2 จะให้ผลทางเอาต์พุตออร์เกตทำงาน ส่งผลไปยังวงจรเตือนสัญญาณ

# 52 การเปรียบเทียบทางดิจิตอล

## ตารางเปรียบเทียบ 2 ค่าทางดิจิตอล

| input | | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|---|
| A | B | A = B | A > B | A < B |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |

จากตารางการเปรียบเทียบนำค่าเอาต์พุต

ของ $f_1, f_2, f_3$ มาเขียนวงจรเกตโดยดูจาก

ค่าminterm ของแต่ละเอาต์พุต

$$f_1 = \bar{A}\bar{B} + AB$$

$$f_2 = A\bar{B}$$

$$f_3 = \bar{A}B$$

**การเปรียบเทียบค่าทางอินพุต 2 ค่าคือ A , B**

**ผลที่เกิดทางด้านเอาต์พุตจะมี 3 ค่า คือ $f_1$ , $f_2$, $f_3$ ดังในตาราง**

# จากตารางเปรียบเทียบ 2 ค่าทางดิจิตอล

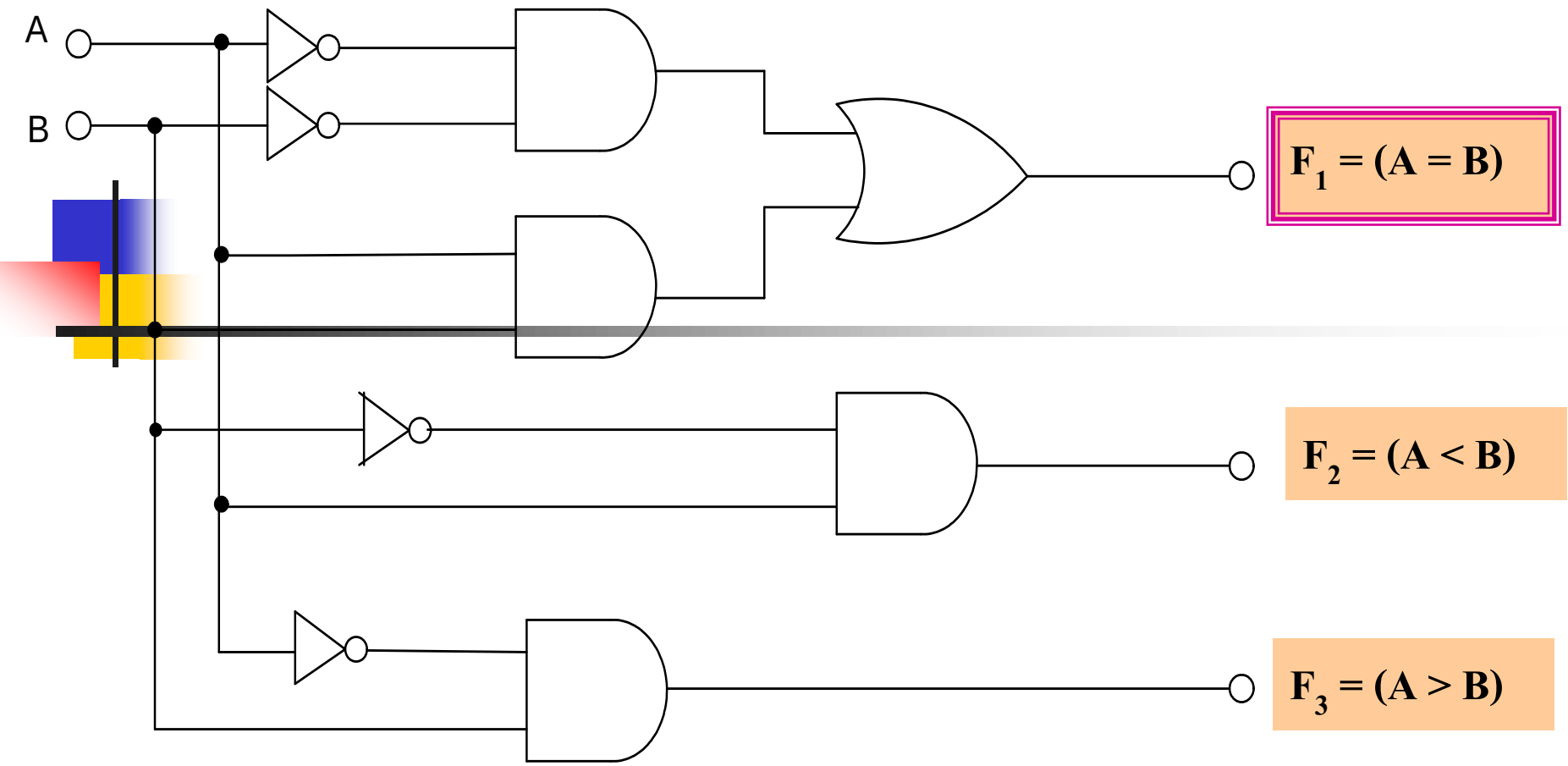
- การเปรียบเทียบค่าอินพุต 2 ค่า แบบดิจิตอล

**เงื่อนไขการเปรียบเทียบ**

- ถ้าค่า A มีค่าเท่ากับค่า B เอาต์พุตจะได้ตามค่า Function $f_1$
- ถ้าค่า A มีค่ามากกว่าค่า B เอาต์พุตจะได้ตามค่า Function $f_2$
- ถ้าค่า A มีค่าน้อยกว่าค่า B เอาต์พุตจะได้ตามค่า Function $f_3$

# 521 การออกแบบวงจรเปรียบเทียบ 1 บิต ทางดิจิตอล

## 1 bit comparator



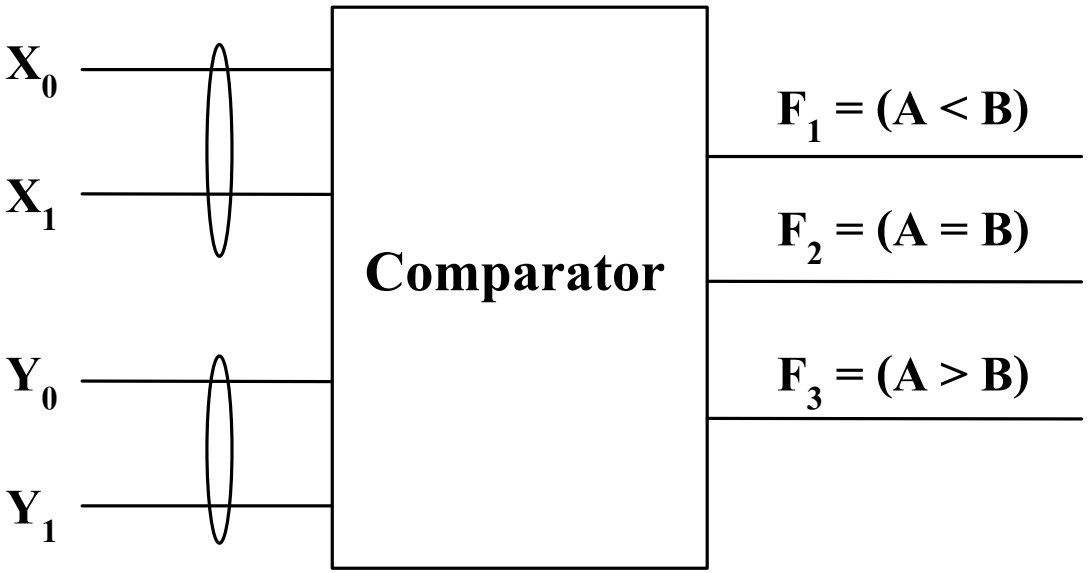
# 522 การออกแบบวงจรเปรียบเทียบ 2 บิต ทางดิจิตอล

จงออกแบบวงจรลอจิกที่มีฟังก์ชันการเปรียบเทียบข้อมูล 2บิต มีการทำงาน ดังนี้

โจทย์กำหนดให้

ข้อมูลที่ใช้ทางอินพุตมีอยู่ 2 ชุด เป็น $(x_1\ x_0)$ ชุดที่1. ,$(y_1\ y_0)$ ชุดที่ 2

เอาต์พุต$F_2$ ได้เป็น High เมื่อเงื่อนไข $X_1 = Y_1$ , $X_0 = Y_0$ ดังตาราง



# อธิบาย

- เอาต์พุต$F_2$ เปรียบเทียบค่าน้อยกว่าเมื่อ X=Y , ($X_1$=$Y_1$) , ($X_0$=$Y_0$)
- เอาต์พุตF1 เปรียบเทียบค่าน้อยกว่าเมื่อ X<Y, ($X_1$<$Y_1$) , ($X_0$<$Y_0$)
- เอาต์พุตF3 เปรียบเทียบค่าน้อยกว่าเมื่อ X>Y, ($X_1$>$Y_1$) , ($X_0$>$Y_0$)

| | $X_1$ | $X_0$ | $Y_1$ | $Y_0$ | $F_1 = (X<Y)$ | $F_2 = (X=Y)$ | $F_3 = (X>Y)$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 15 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

$F_2$ จะเป็นดังสมการที่กล่าวไว้ว่า Out put จะเป็น High ได้ก็ต่อเมื่อเงื่อนไข

$X_1 = Y_1$

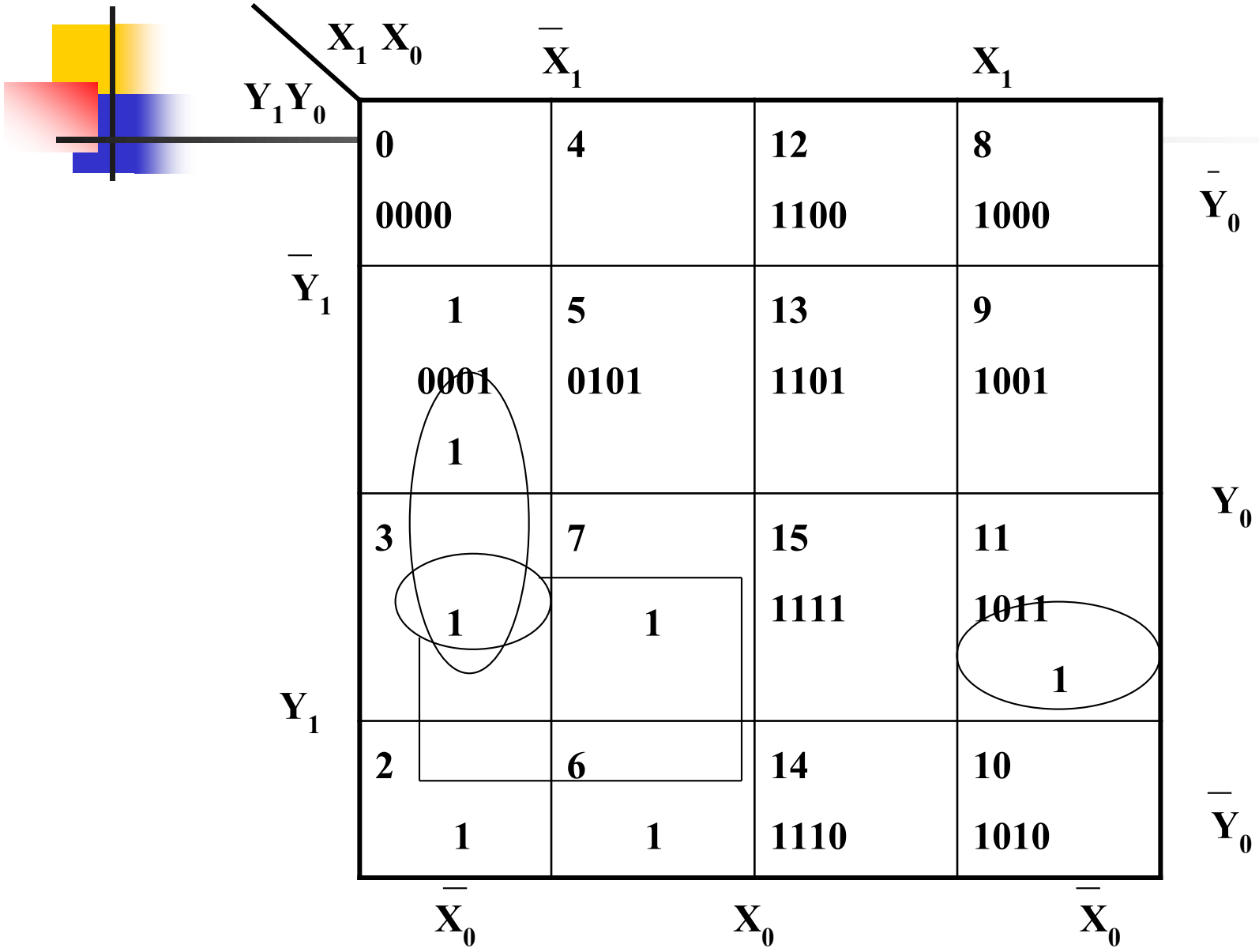
$X_0 = Y_0$

# อธิบาย

- ตารางการเปรียบเทียบ X,Y ขนาด 2 บิต
- $X_0, Y_0$ เป็นบิต $L_{SB}$
- $X_1, Y_1$ เป็นบิต $M_{SB}$
- จากตารางความจริงกำหนดค่าตัวแปร $X_0$, $X_1$ กับ $Y_0, Y_1$ ตามลำดับของเลขฐาน2 ค่า 0-16
- การเปรียบเทียบค่าที่ได้ 3 ส่วน คือ มากกว่า น้อยกว่า และค่าเท่ากัน
  - โดยมี $F_1$ ซึ่งเป็นเอาต์พุตของวงจรเปรียบเทียบ หากค่า x น้อยกว่าค่า y
  - $F_2$ เป็นค่าเอาต์พุตของวงจรเปรียบเทียบ ถ้าค่า x เท่ากับค่า Y
  - $F_3$ เป็นค่าเอาต์พุตของวงจรเปรียบเทียบ ในกรณี x มากกว่า y

  หลักการเปรียบเทียบพิจารณาค่า $L_{SB}$ ก่อนแล้วจึงมาพิจารณาที่บิต $M_{SB}$
  - จากค่า $F_2$ ซึ่งเป็นค่าเปรียบเทียบที่มีค่าระหว่าง X ,Y มีค่าเท่ากัน ในตำแหน่งที่ 0, 10, 15 ของตารางเปรียบเทียบ หลักแรกในการเปรียบเทียบ $X_0 = Y_0$ ก่อน ซึ่งจะมีค่าเท่ากัน จากนั้นก็ไปเปรียบเทียบหลักที่ $X_1 = Y_1$ ในตัวอย่างนี้ก็จะเท่ากันอีก ดังนั้น ที่ตำแหน่ง 0,5,10,15 มีค่าเท่ากับ 1

$(X_1X_0Y_1Y_0)$ Function $F_1$=A<B

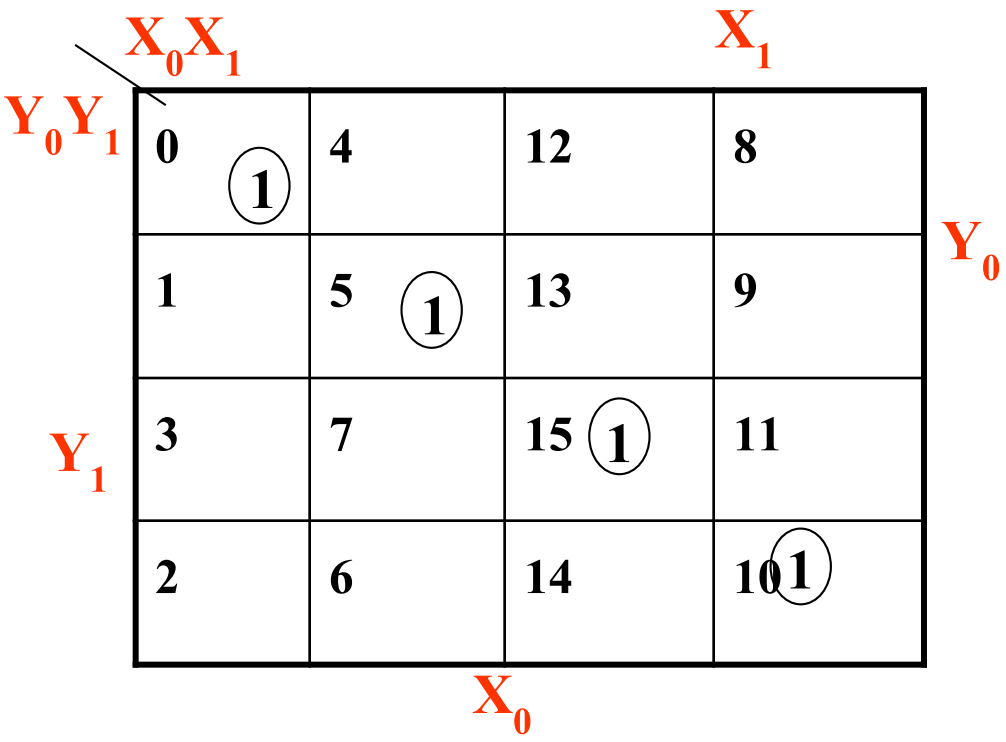


# อธิบาย

- จากตารางความจริงนำฟังก์ชันการเปรียบเทียบค่า $X_1X_0 < Y_1Y_0$ $F_1=X < Y$ ในรูปแบบของ minterm นำค่า 1,2,3,6,7,11
- ใส่ลงในตาราง k-map ขนาด 4 ตัวแปร ดังรูปตารางmap
- สามารถลดรูปในการเลือกจับข้อมูล 4 เทอมติดต่อกันลดรูปได้ 2 ตัวแปร $\bar{X}_1 Y_1$ เลือกจับข้อมูล 2 เทอมติดต่อกันได้ 3 ตัวแปร คือ $\bar{X}_0 X_1 Y_0$ และ $\bar{X}_0 Y_0 Y_1$ ได้ค่าเป็นเอาต์พุตของ $F_1$ คือ

$$F_1 = \overline{X_1}Y_1 + \overline{X_0}\ \overline{X_1}\ Y_0 + \overline{X_0}Y_0Y_1$$

# Function $F_2 = A = B$



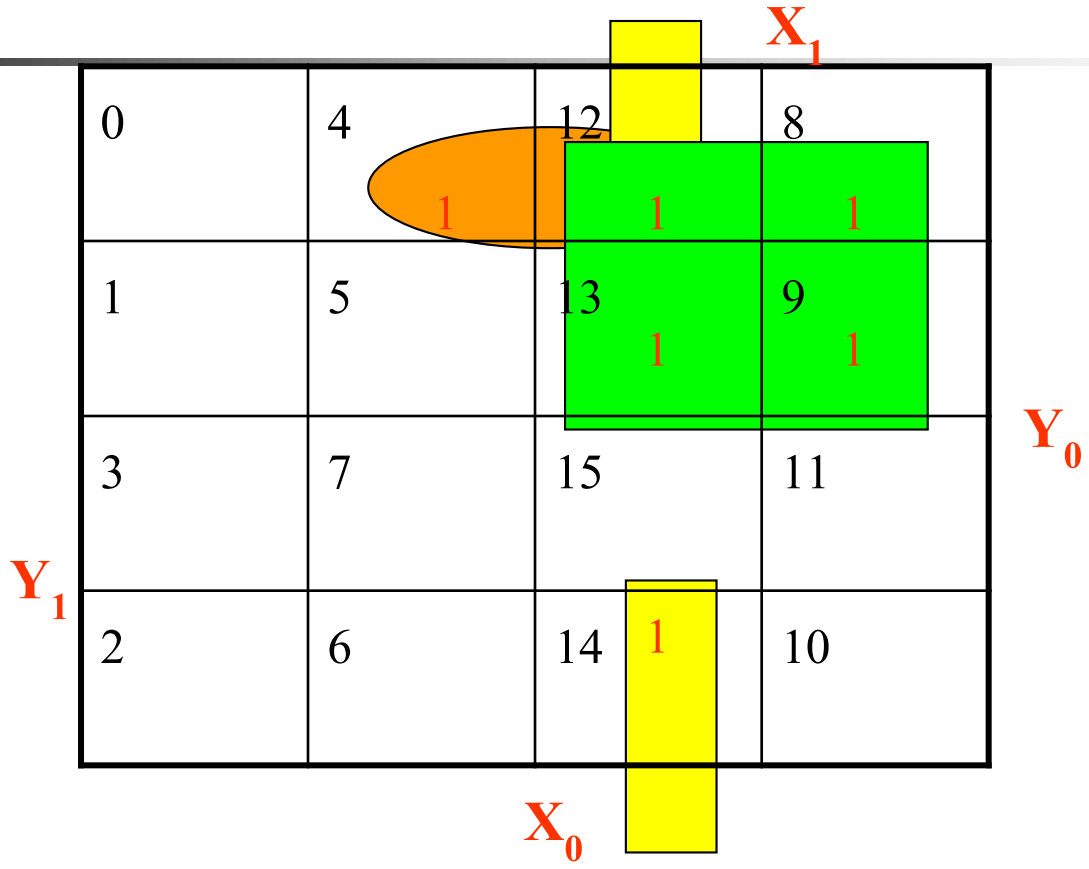

เขียนเป็นฟังก์ชันเมื่อ X = Y

$$X_1 X_0 = Y_1 Y_0$$

$$F_2 = (X_1 X_0 Y_1 \overline{Y}_0 + X_1 X_0 Y_1 Y_0) + (X_1 X_0 Y_1 Y_0 + X_0 X_1 Y_0 Y_1)$$

# Function $F_3 = A > B$

$$X_1 X_0 > Y_1 Y_0$$




เขียนเป็นฟังก์ชันเมื่อ $X > Y$

$$F_3 = X_1 Y_1 + X_0 Y_1 Y_0 + X_1 X_0 Y_0$$

# การนำเกตมาใช้ในวงจรเปรียบเทียบ

- ค่าที่ได้ $F_1, F_2, F_3$ สามารถนำไปสร้างวงจรตามหน้าที่การทำงาน
- การนำเอา EX- NOR Gates มาใช้ในการเปรียบเทียบทำได้ง่ายๆ
- โดยพิจารณาจากตารางความจริงของคุณสมบัติ EX- NOR Gates เงื่อนไขว่าอินพุตเหมือนกันให้ค่า 1 อินพุตต่างกันให้ 0
- จากการนำคุณสมบัตินี้สามารถนำไปสร้างวงจรเปรียบเทียบค่าเท่ากันได้
- เราสามารถนำไปเปรียบเทียบค่าได้ n บิต
- ดังนั้นค่าที่เกิดขึ้นทางเอาต์พุตจะเป็น 0 ก็ต่อเมื่อA >B, B >A

## วิธีทำ

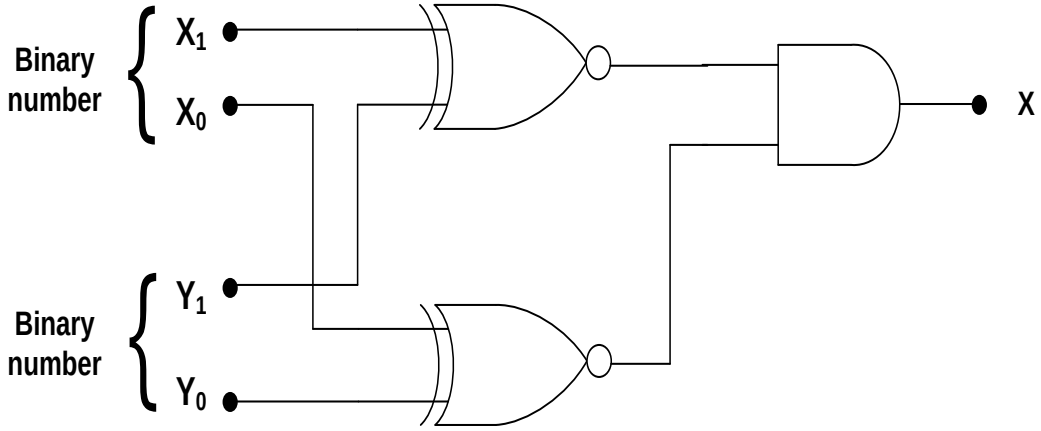
**ขั้นแรก** เขียนตารางค่าความจริงตามเงื่อนไขที่โจทย์ กำหนดมาให้ นั่นคือ X เป็น 1 $X_1 = Y_1$ และ $X_0 = Y_0$ ซึ่งมี 4 กรณีด้วยกัน

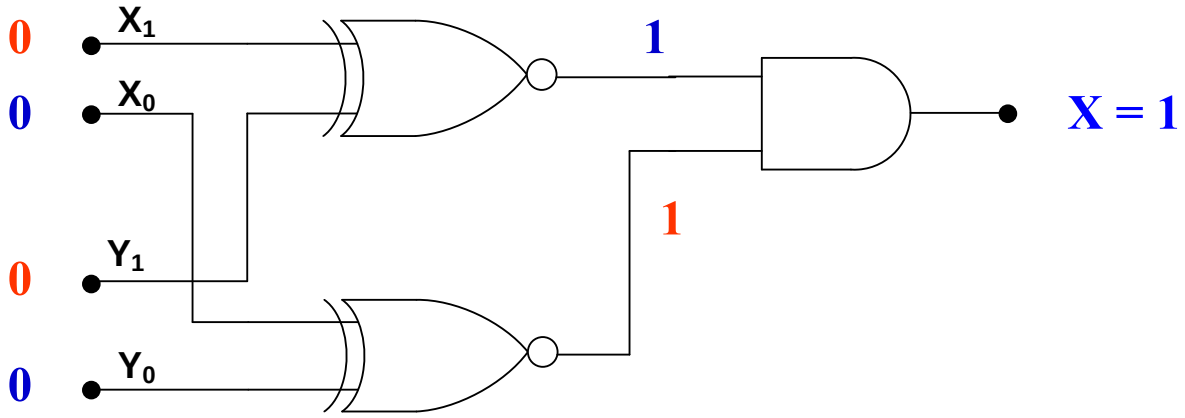
**ขั้นที่สอง** ทำการจัดสมการเอาต์พุตให้อยู่ในรูปแบบการรวมของผลคูณ(minterm) แล้วแก้ปัญหาโดยใช้หลักการพื้นฐานของเกตต่างๆ แล้วจะสามารถแก้ปัญหานี้ได้ง่ายขึ้น

**ขั้นที่สาม** ทำการออกแบบวงจรโดยการนำเอาข้อมูลที่ได้จากที่ได้ทำการจัดสมการเอาต์พุตที่อยู่ในรูปแบบการรวมของผลคูณเรียบร้อยแล้ว ดังนั้น เราไปออกแบบวงจรกันเลย โดยเราจะใช้ EX – NOR Gate กับ AND Gate มาใช้ในวงจรนี้

# รูปวงจรที่ออกแบบเรียบร้อยแล้วโดยใช้

## EX – NOR GATE กับ AND GATE

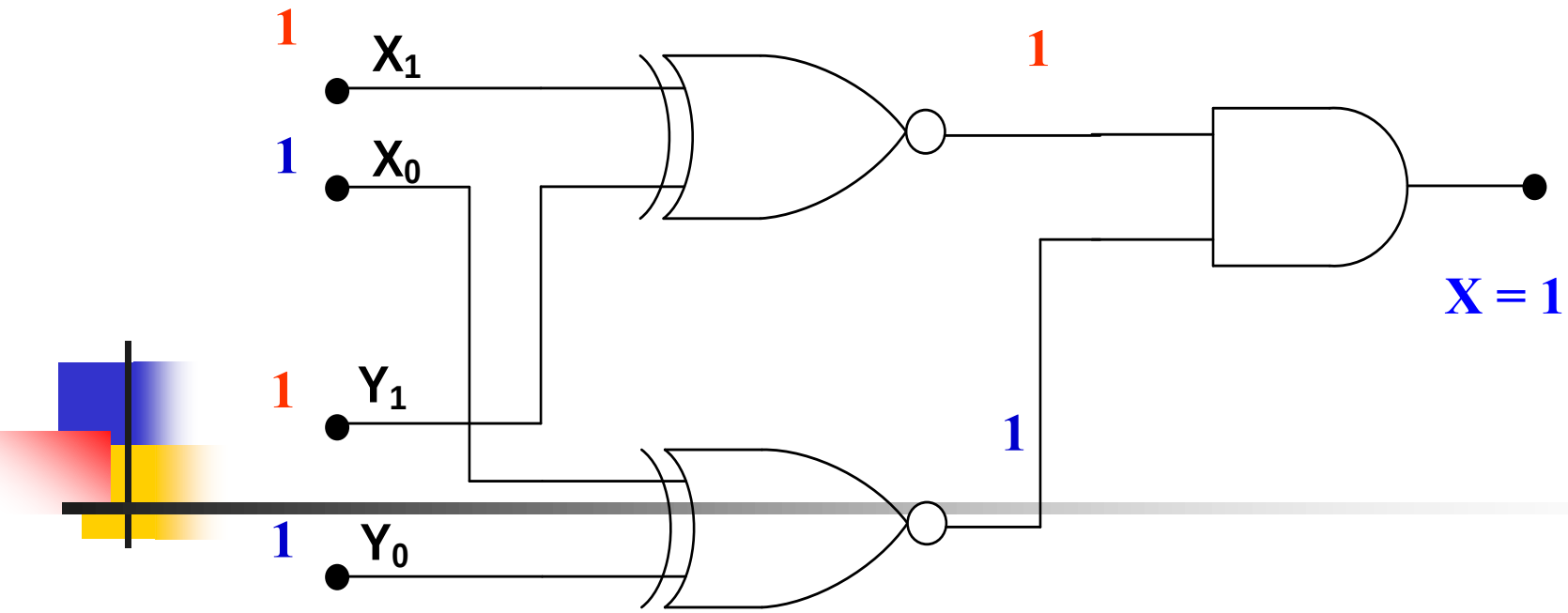


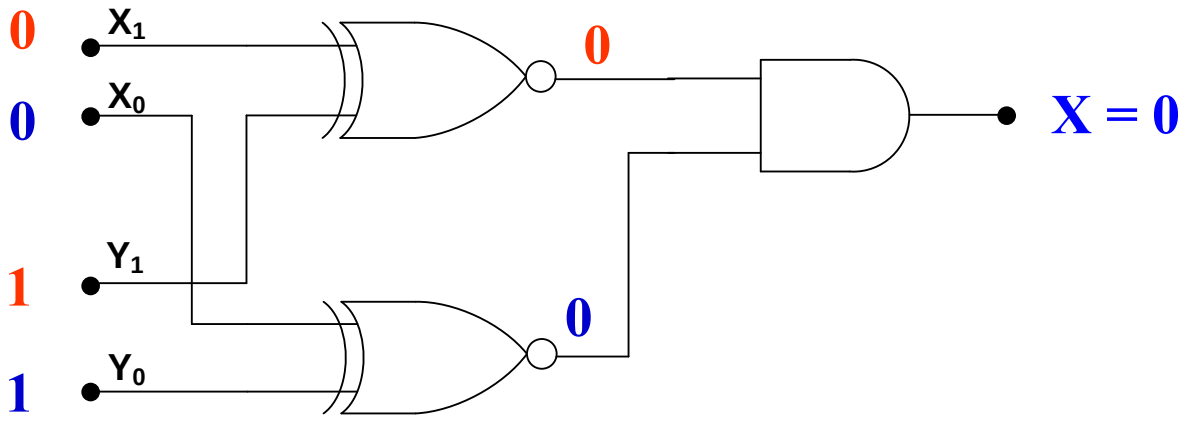
# อธิบาย

ถ้าป้อน X1 และ Y1 ไปยัง EX – NOR ตัวหนึ่งและป้อน X0 และ Y0

ไปยัง EX – NOR อีกตัวหนึ่ง เมื่อ In put ทั้งสองมีระดับ In put เหมือนกันทำให้ out put ของเกต EX – NOR แต่ละตัวเป็น 1 สำหรับกรณีนี้ X0 = Y0 และ X1 = Y1 out put ของ EX – NOR ทั้งสองจึงเป็น “1” ซึ่งเป็นเงื่อนไขที่ต้องการ เพราะเมื่อ In put เป็น “1” ทำให้ out put เป็น “1” out put ของเกต AND เป็น “1”




ถ้าป้อน $X_1 = Y_1$ และ $X_0 = Y_0$ Out put ของEX – NOR ทั้งสองจึงเป็น “ 1 ” ซึ่งเป็นเงื่อนตามไขที่ต้องการเพราะเมื่อ In put $X_1 = 1$ , $Y_1 = 1$ และ $X_0 = 1$ , $Y_0 = 1$ ทำให้Out put เป็น “ 1 ” ทำให้Out put ของเกต AND เป็น “ 1 ”



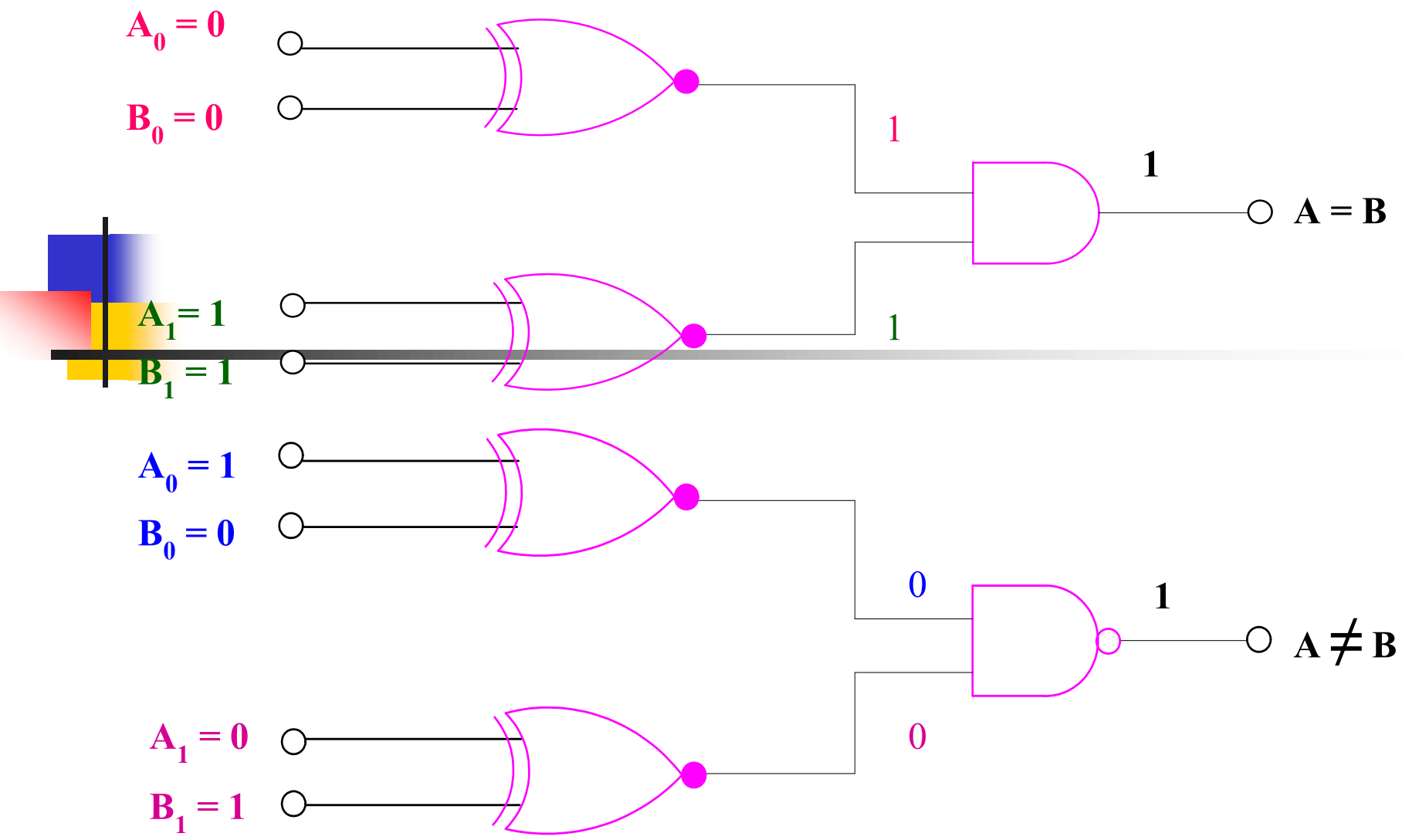

ถ้าป้อน $X_1 < Y_1$ และ $X_0 < Y_0$ out put ของ EX – NOR ทั้งสองจึงเป็น “ 0 ” ซึ่งเป็นตามเงื่อนไขที่ต้องการ เพราะเมื่อ In put $X_1 = 0$ , $Y_1 = 1$ และ $X_0 = 0$ , $Y_0 = 1$ ทำให้ out put เป็น “ 0 ” ทำให้ Out put ของเกต AND เป็น “ 0 ”

**จากตัวอย่างที่ทำการออกแบบ สรุปได้ว่า**

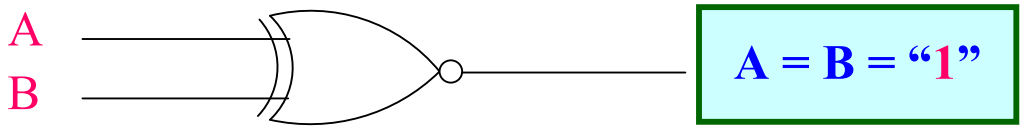
เอาต์พุตที่ออกมาได้เป็น “High” ได้ก็ต่อเมื่อ $X_0 = Y_0$ และ $X_1 = Y_1$ เท่านั้น

ถ้าป้อนอินพุตเป็น $X_0 > Y_0$ และ $X_1 < Y_1$ เอาต์พุตจะได้ “ Low ” เท่านั้น

# ตัวอย่าง จากวงจรเอาต์พุตของวงจรเปรียบเทียบอินพุตดังนี้

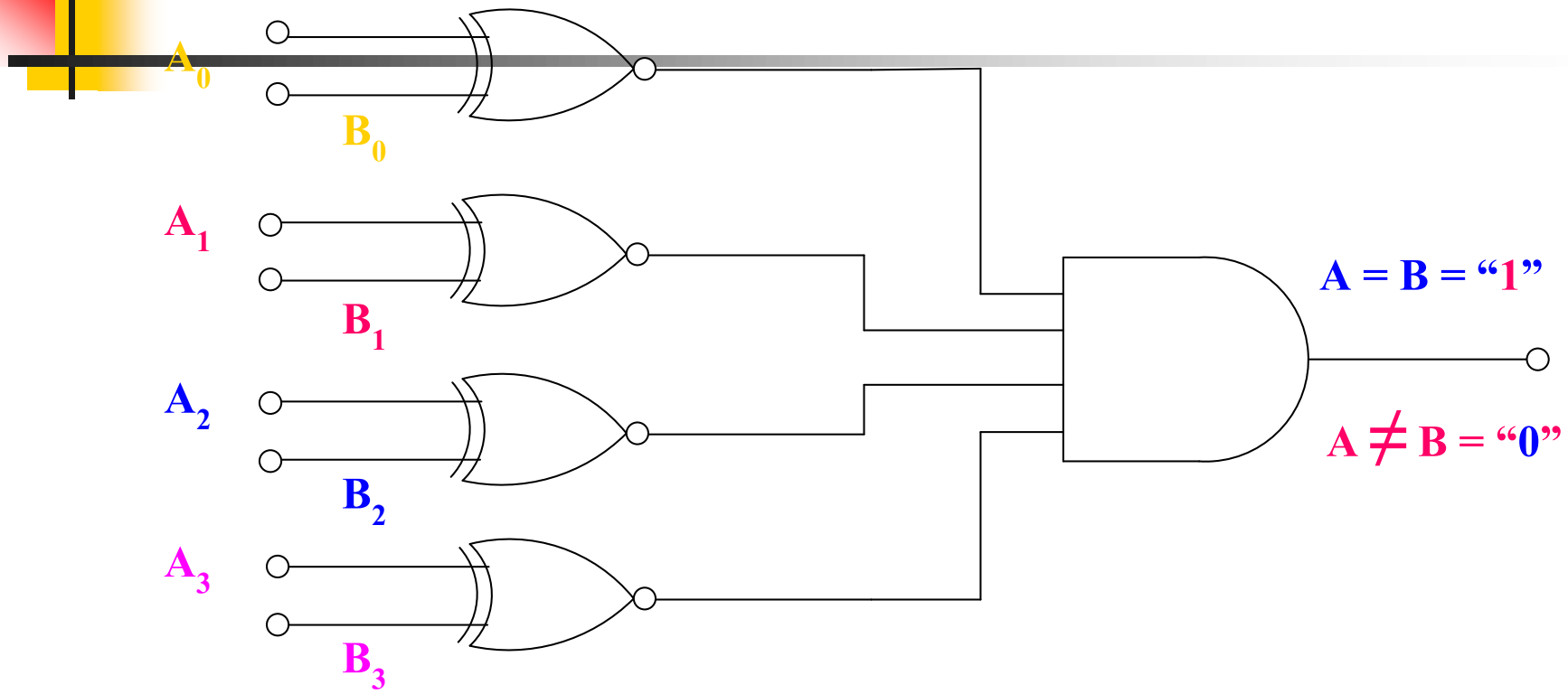


# 523 การออกแบบวงจรเปรียบเทียบ 4 บิต ทางดิจิตอล




จากตาราง $F_1$ จะมีค่าเท่ากับ EX – NOR Gate

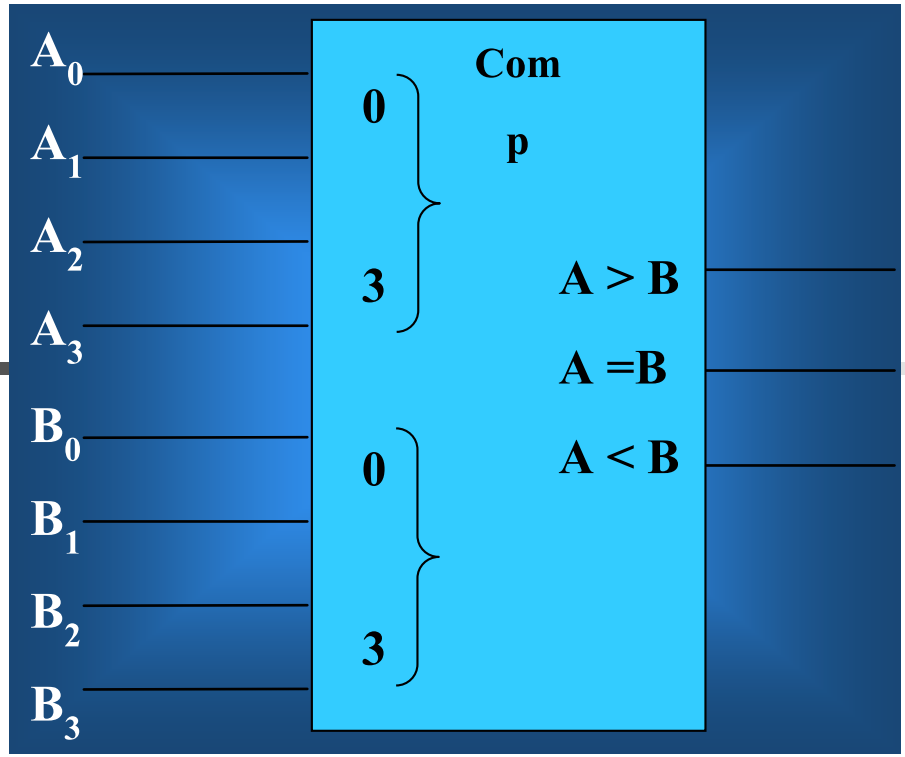
## วงจรเปรียบเทียบ 4 บิต



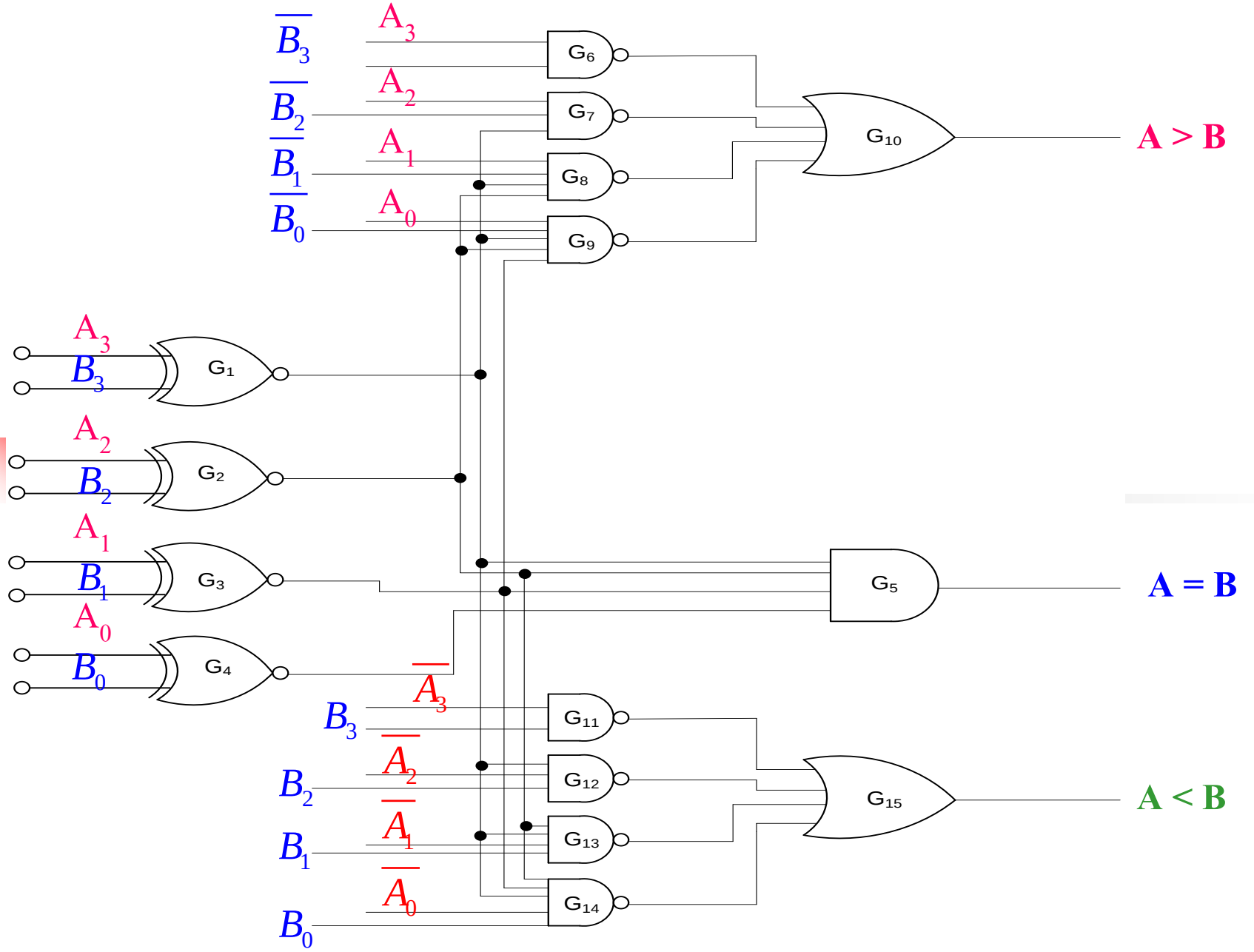
# หลักการเปรียบเทียบบิต

- การนำเอาวงจรเปรียบเทียบในรูปแบบ EX-NOR Gates ไปใช้งานมีข้อดีที่สามารถสร้างวงจรเปรียบเทียบถึง n บิต
- การนำผลที่ได้จากการเปรียบเทียบในกรณีที่มีค่ามากกว่า หรือน้อยกว่า
- โดยการนำเอาค่าผลทางเอาต์พุตที่เกิดขึ้นคือค่า 0 ไปเข้าวงจร NAND Gates ของผลการเปรียบเทียบบิตหลัก $A_0$ กับ $B_0$, $A_1$ กับ $B_1$, $A_2$ กับ $B_2$, $A_3$ กับ $B_3$
- ตั้งหลักเปรียบเทียบจาก ค่า $L_{SB}$ ไปยังหลัก $M_{SB}$
- ผลการเปรียบเทียบจะเปรียบเทียบหลักทั้งหมด

# Integrated Circuit Comparators




**สัญลักษณ์ของ IC Comparators 4 bit**




วงจรเปรียบเทียบแบบ 4 บิต โครงสร้างลอจิกเกต และมีเอาต์พุตออกมา 3 ค่า

# อธิบาย

- การทำงานของวงจรตรวจสอบค่าขนาด 4บิต ประกอบด้วยวงจร 3 วงจรดังนี้คือ วงจรเปรียบเทียบค่าเท่ากัน ค่าน้อยกว่า ค่ามากกว่า
- วงจรที่เปรียบเทียบค่าที่เท่ากันออกแบบโดยใช้คุณสมบัติของ EX-NOR ดังที่กล่าวมา ถ้าอินพุตมีค่าเหมือนกันค่าจะเท่ากัน เอาต์พุตเป็นค่า 1 ถ้าอินพุตต่างกันจะได้ค่า 0
- ถ้าต้องการเปรียบเทียบจำนวน 4 บิต ก็ใช้ EX-NOR ($G_1$-$G_4$) จำนวน 4 ตัว เอาต์พุตต่อไปยังแอนด์เกต 4 อินพุต ($G_5$) ค่าผลการเปรียบเทียบ เมื่ออินพุต 4 บิต $A_0$-$A_3$มีค่าเท่ากับ $B_0$-$B_3$ เอาต์พุต$G_5$ เป็น 1
- การเปรียบเทียบวงจร A>B , A<B มีหลักการโดยนำค่าผลเอาต์พุตของการเปรียบเทียบวงจรเท่ากันมาใช้
- โดยใช้หลักการเปรียบเทียบถ้าค่าเท่ากันอินพุตวงจรเปรียบเทียบค่ามากกว่า และน้อยกว่าจะให้ค่า0 หลัก$M_{SB}$ มาก่อน มีหลักการดังนี้
  - ถ้า $A_3$= 1 ,$B_3$=0 แสดงว่าค่า A>B เอาต์พุตที่ ($G_6$) เป็น 1 (A แอนด์ NOTB )
  - ถ้า $A_3$= 0 ,$B_3$=1 แสดงว่าค่า A<B เอาต์พุตที่ ($G_{11}$)เป็น 1 (NOTB แอนด์ A )
  - ถ้า $A_3$= 1 ,$B_3$=1 แสดงว่าค่า A=B เอาต์พุตที่ ($G_1$) เป็น 1 (EX-NOR )

# อธิบาย

- จากหลักการทั้งสาม สามารถสร้างวงจร เปรียบเทียบค่ามากกว่าโดยใช้เอาต์พุตของการเปรียบเทียบค่าเท่ากันบิตที่3 ($G_1$)ไปควบคุมการเปรียบเทียบบิตที่ 2 ($G_7$) โดยพิจารณาที่อินพุตของวงจรเปรียบเทียบค่ามากกว่าค่าทางอินพุตจะใช้ $A_2$ แอนด์กับComplement $B_2$ ของวงจรบิต 2 กำหนดค่าเป็นค่า A>B ที่บิตที่ 2 ค่า $A_2$ ควรเป็น 1 $B_2$ ควรเป็น 0 เอาต์พุตควบคุมจาก $G_1$ จะมาควบคุมให้มีการเปรียบเทียบกันที่บิต 2 ให้ค่าเอาต์พุตที่วงจรเปรียบเทียบค่ามากเป็น 1 ถ้าหากการเปรียบเทียบที่บิตที่ 2 มีค่าเท่ากันสัญญาณควบคุมจากวงจรเปรียบเทียบค่าเท่ากันจะไปควบคุมที่บิตถัด ไปคือบิตที่ 1 ถ้าค่าที่บิตที่ 1 มีค่าเท่ากันอีกก็จะไปควบคุมบิต 0 จนครบ
- ในกรณีที่ค่า A<B ก็เช่นกัน นำค่า Complement A ไปแอนด์กับ B และนำค่าเอาต์พุตของวงจรเปรียบเทียบค่าเท่ากันไปควบคุมการทำงานอีกครั้ง ถ้า A<B ค่า A ควรเป็น 0 ค่า B ควรเป็น 1 ค่าเอาต์พุตจากการเปรียบเทียบค่าน้อยกว่าควรเป็น 1 ที่ ($G_{11}$) ถ้าการเปรียบเทียบมีค่าเท่ากัน ก็จะไปเปรียบเทียบในหลักถัดไปโดยนำเอาต์พุตของวงจรเปรียบเทียบค่าเท่ากันของ $A_3$ ไปควบคุมบิต 2 ของวงจรน้อยกว่า อย่างนี้จนครบบิต

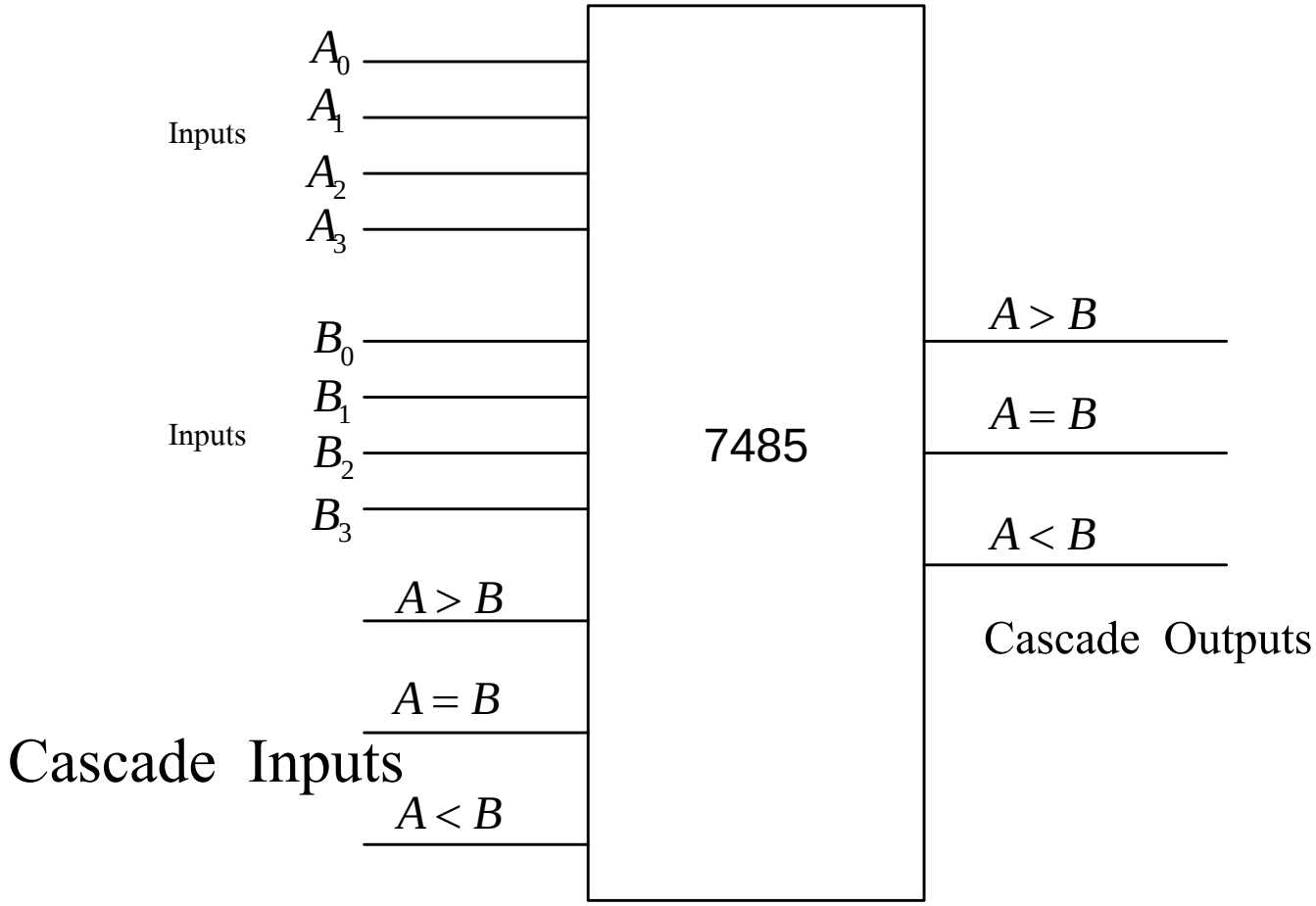
# การออกแบบระบบALU (Arithmetic Logic Unit)ในระบบคอมพิวเตอร์

| Comparing Inputs | | | | Cascade inputs | | | Cascade outputs | | |
|---|---|---|---|---|---|---|---|---|---|
| $A_3 , B_3$ | $A_2 , B_2$ | $A_1 , B_1$ | $A_0 , B_0$ | A>B | A<B | A=B | A>B ($C_3$) | A<B ($C_1$) | A=B($C_2$) |
| $A_3>B_3$ | X | X | X | X | X | X | H | L | L |
| $A_3<B_3$ | X | X | X | X | X | X | L | H | L |
| $A_3=B_3$ | $A_2>B_2$ | X | X | X | X | X | H | L | L |
| $A_3=B_3$ | $A_2<B_2$ | X | X | X | X | X | L | H | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1>B_1$ | X | X | X | X | H | L | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1<B_1$ | X | X | X | X | L | H | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0>B_0$ | X | X | X | H | L | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0<B_0$ | X | X | X | L | H | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | H | L | L | H | L | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | L | H | L | L | H | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | L | L | H | L | L | H |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | X | X | H | L | L | H |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | H | H | L | L | L | L |
| $A_3=B_3$ | $A_2=B_2$ | $A_1=B_1$ | $A_0=B_0$ | L | L | L | H | H | L |

# อธิบาย

- ตารางการเปรียบเทียบ4 บิตของไอซีเบอร์ 7485 ค่าที่เกิดขึ้นตามเงื่อนไขดังนี้
- การเปรียบเทียบจะทำการเปรียบเทียบจาก $M_{SB}$ ไปยัง $L_{SB}$
  ถ้าค่า $A_3>B_3$ แสดงว่าค่าที่เกิดจริงค่า A มีค่ามากกว่าค่า B เอาต์พุตที่เกิดขึ้นจะเป็นค่า Highที่ ($C_3$) แต่ถ้าค่า $A_3=B_3$ ก็จะทำการเปรียบเทียบในหลักถัดไปดังตารางการเปรียบเทียบจนครบจำนวนบิตเปรียบเทียบ จะเห็นว่าการเปรียบเทียบจากค่ามากไปหาค่าน้อยสุดท้ายจะเป็นผลของการเปรียบเทียบ ในกรณีที่ไม่สนใจ Cascade Inputs
- ถ้าต่อกรณีมีการพิจารณาที่ Cascade Inputs ด้วยเราจะพิจาณาที่เหมือนเดิมโดยให้ศักยภาพของCascade Inputs เป็นหลัก $L_{SB}$
- ในกรณีที่ Cascade Inputs เป็นค่า High ที่ A>B , A<B ค่าเอาต์พุต จะเป็น Low ทั้งหมด
- ส่วนในกรณีที่ Cascade Inputs เป็นค่า Low ที่ A>B , A<B , A=B ค่าเอาต์พุต จะเป็น Low ที่ A=B ทั้งหมด

# วงจรเปรียบเทียบขนาด 4 บิต

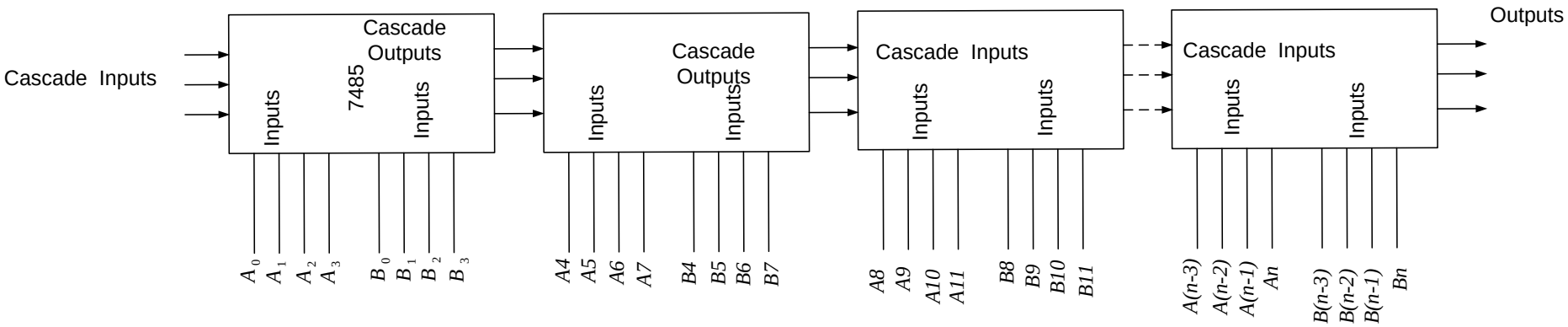


# อธิบาย

- วงจรเปรียบเทียบขนาด 4 บิต ด้วยไอซีสำเร็จรูปเบอร์7485
- การทำงานแบ่งออกเป็น 4 ส่วนที่สัมพันธ์กันคือ
- ส่วนที่ 1 อินพุตขนาด 4บิต ประกอบด้วย $A_0$-$A_3$ เป็นอินพุตที่ 1
- ส่วนที่ 2 อินพุตขนาด 4บิต ประกอบด้วย $B_0$-$B_3$ เป็นอินพุตที่ 2
- ส่วนที่3 Cascade Inputs มีการทำงานในรูปแบบ A>B , A=B , A<B
- ส่วนที่4 Cascade Outputs มีการทำงานในรูปแบบ A>B , A=B , A<B
- การเปรียบเทียบจะนำอินพุตส่วนที่1 คือ $A_0$-$A_3$ ไปเปรียบเทียบกับส่วนที่ 2 $B_0$-$B_3$ ค่าที่เกิดขึ้นจะเป็นส่วนที่4 ขนาด 3บิต เป็นผลทางเอาต์พุตในส่วนที่3 จะนำไปต่อกับไอซี ชนิดเดียวกันเพื่อเพิ่มบิตการทำงาน

# 524 การออกแบบวงจรเปรียบเทียบ n บิต ทางดิจิตอล

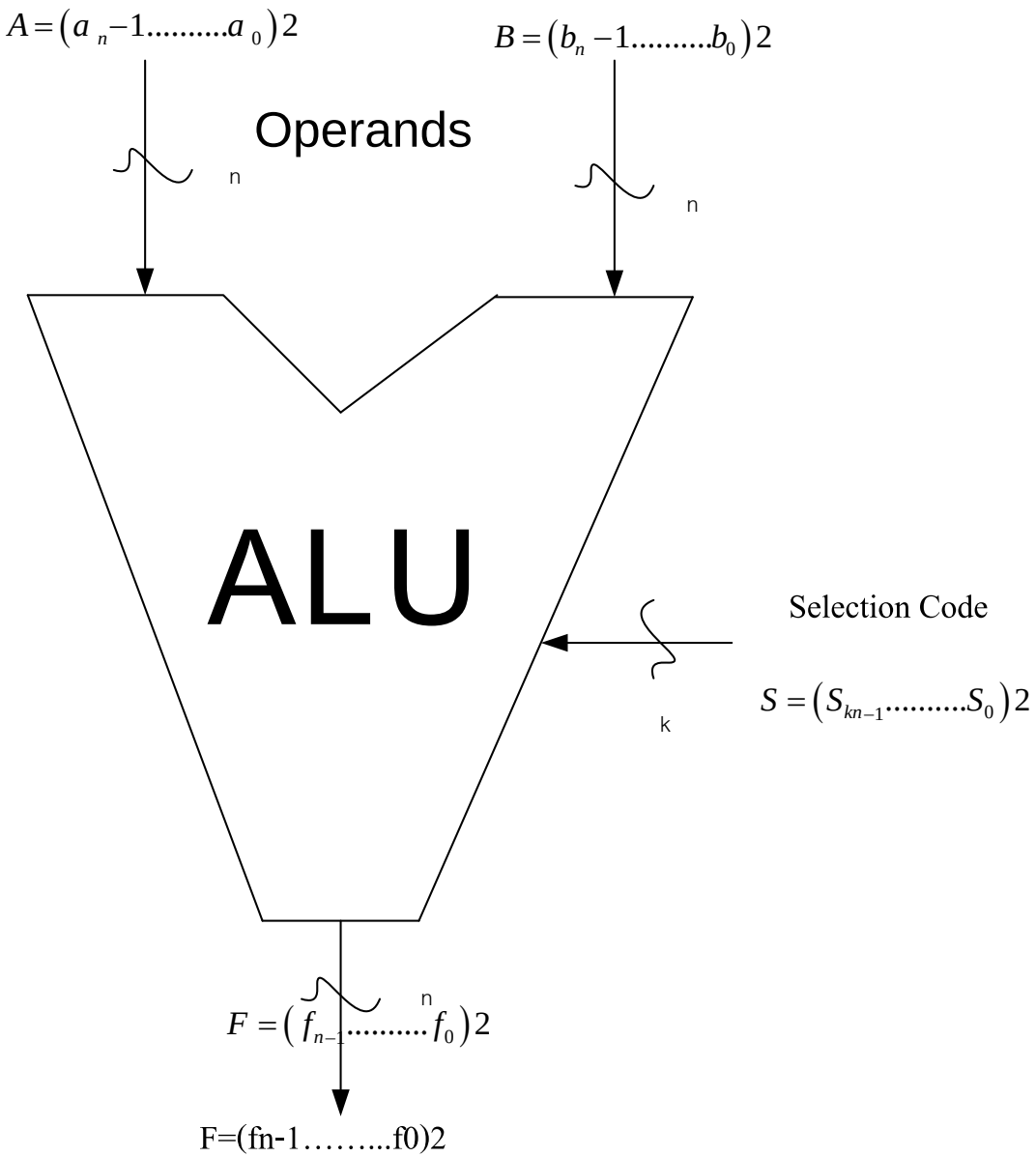
## วงจรเปรียบเทียบขนาด n bits



# อธิบาย

- การสร้างวงจรเปรียบเทียบขนาด n บิต ประกอบด้วยไอซีเบอร์ 7485 จำนวนเท่ากับ n ตัว ดังรูปแสดง
- สัญญาณเอาต์พุตของตัวที่ n จะเป็นเอาต์พุตของผลสุดท้ายจะมีจำนวน 3 บิต A<B , A=B, A>B
- สัญญาณ Cascade Inputs ของตัวที่ n จะนำไปต่อกับเอาต์พุตของ n-1 จนถึง $n_0$ ดังรูปแสดง
- สัญญาณเปรียบเทียบของแต่ละตัวจะป้อนเข้าที่อินพุตของแต่ละตัวเทียบกันตั้งแต่ $A_0$-$A_{n-1}$ ,$B_0$-$B_{n-1}$ ในรูปแบบไบนารี่ ที่ป้อนสัญญาณการเปรียบแต่ละบิต
- สัญญาณ Cascade Inputs ที่จะนำไปป้อนสัญญาณตัวแรกซ้ายสุด จะมีค่าเท่ากับ (0,1,0) เป็นค่าเริ่มต้น

# 53 หลักการของ ALU



# อธิบาย

- ALU ย่อมาจากคำว่า Arithmetic Logic Unit เป็นหน่วยคำนวณทางคณิตศาสตร์ และตรรกศาสตร์
- หน่วยที่ทำหน้าที่ประมวลผลโดยคำนวณทางคณิตศาสตร์มีฟังก์ชัน บวก ลบ คูณ หาร และเปรียบเทียบทางตรรกะ
- สัญลักษณ์ มาตรฐานทางลอจิก แสดงดังรูป ตัวถูกคำนวณเป็น Operands

$$A=\left(a_{n-1}.....a_0\right)^2 \qquad B=\left(b_{n-1}.....b_0\right)^2$$

- การปฏิบัติการโดยการเลือกที่ $S=\left(S_{K-1}.....S_0\right)^2$ k เป็นจำนวนบิต
- โมดุลของ ALU มีมาตรฐานปฏิบัติการคำนวณทางคณิตศาสตร์ 4 ฟังก์ชัน
- หลักการเบื้องต้นของ ALU มี 8 ฟังก์ชัน

# ตารางแสดงหน้าที่การทำงาน ALU

| Selection Code | | | ALU Function | Description |
|---|---|---|---|---|
| $S_2$ | $S_1$ | $S_0$ | | |
| 0 | 0 | 0 | F = A+B | Add |
| 0 | 0 | 1 | F = A-B | Subtract |
| 0 | 1 | 0 | F = A+1 | Increment |
| 0 | 1 | 1 | F = A-1 | Decrement |
| 1 | 0 | 0 | $F = A \cap B$ | AND |
| 1 | 0 | 1 | F = A U B | OR |
| 1 | 1 | 0 | $F = \overline{A}$ | NOT |
| 1 | 1 | 1 | $F = A \oplus B$ | XOR |

# อธิบาย

- ตารางแสดงหน้าที่การทำงานของ ALU 8 ฟังก์ชัน ดังนี้
- Selection Code ทำหน้าที่กำหนดการทำงานของแต่ละฟังก์ชัน
- ALU Function แสดงหน้าที่การทำงานเขียนเป็นฟังก์ชันทางคณิตศาสตร์ และตรรกศาสตร์
- Description บอกลักษณะของการกระทำ เช่นในรูปแบบ บวก ลบ เพิ่มค่า ลดค่า เป็นตัวอย่าง

# อธิบาย

**ตัวอย่าง เมือเลือกโค้ดที่กำหนดหน้าที่**

000 เขียนเป็นฟังก์ชันการทำงานได้ F = A+B ลักษณะการกระทำนำค่า A ไปบวกกับค่า B ได้ค่า F ทางเอาต์พุต

001 ฟังก์ชันเป็น F = A-B ลักษณะการกระทำนำค่า A ไปลบกับค่า B ได้ค่า F ทางเอาต์พุต

010 ฟังก์ชันเป็น F = A+1 ลักษณะการกระทำนำค่า A ไปบวกเพิ่มอีก 1

011 ฟังก์ชันเป็น F = A-1 ลักษณะการกระทำนำค่า A ไปลดค่า 1

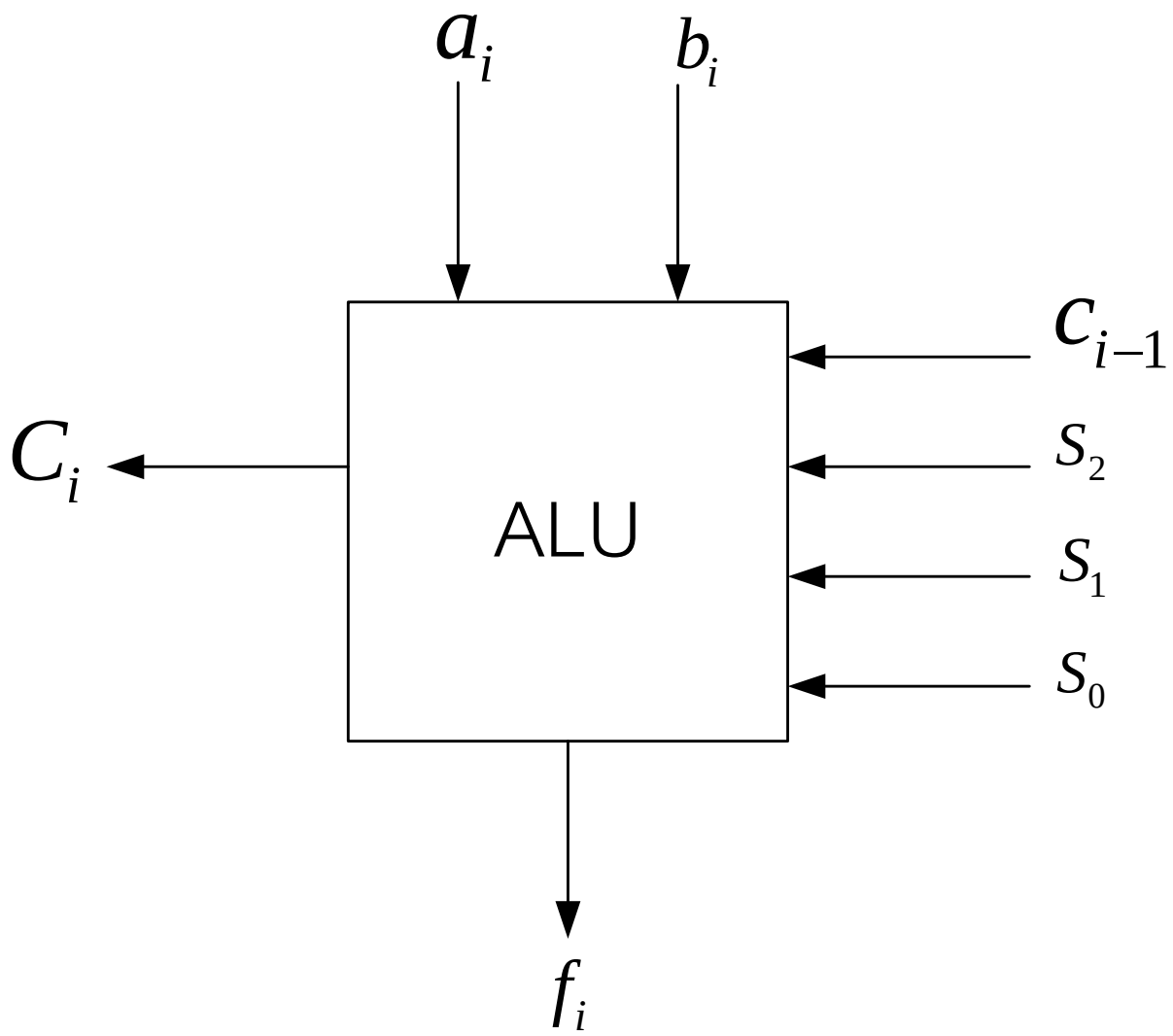
100 ฟังก์ชันเป็น F = A $\cap$ B ลักษณะการกระทำนำค่า A ไปแอนด์กับค่า B

101 ฟังก์ชันเป็น F = $\overline{A}$ U B ลักษณะการกระทำนำค่า A ไปออร์กับค่า B

110 ฟังก์ชันเป็น F = $\overline{A}$ ลักษณะการกระทำนำค่า A ไป NOT

111 ฟังก์ชันเป็น F = $A \oplus B$ ลักษณะการกระทำนำค่า A ไป EX-OR กับค่าB

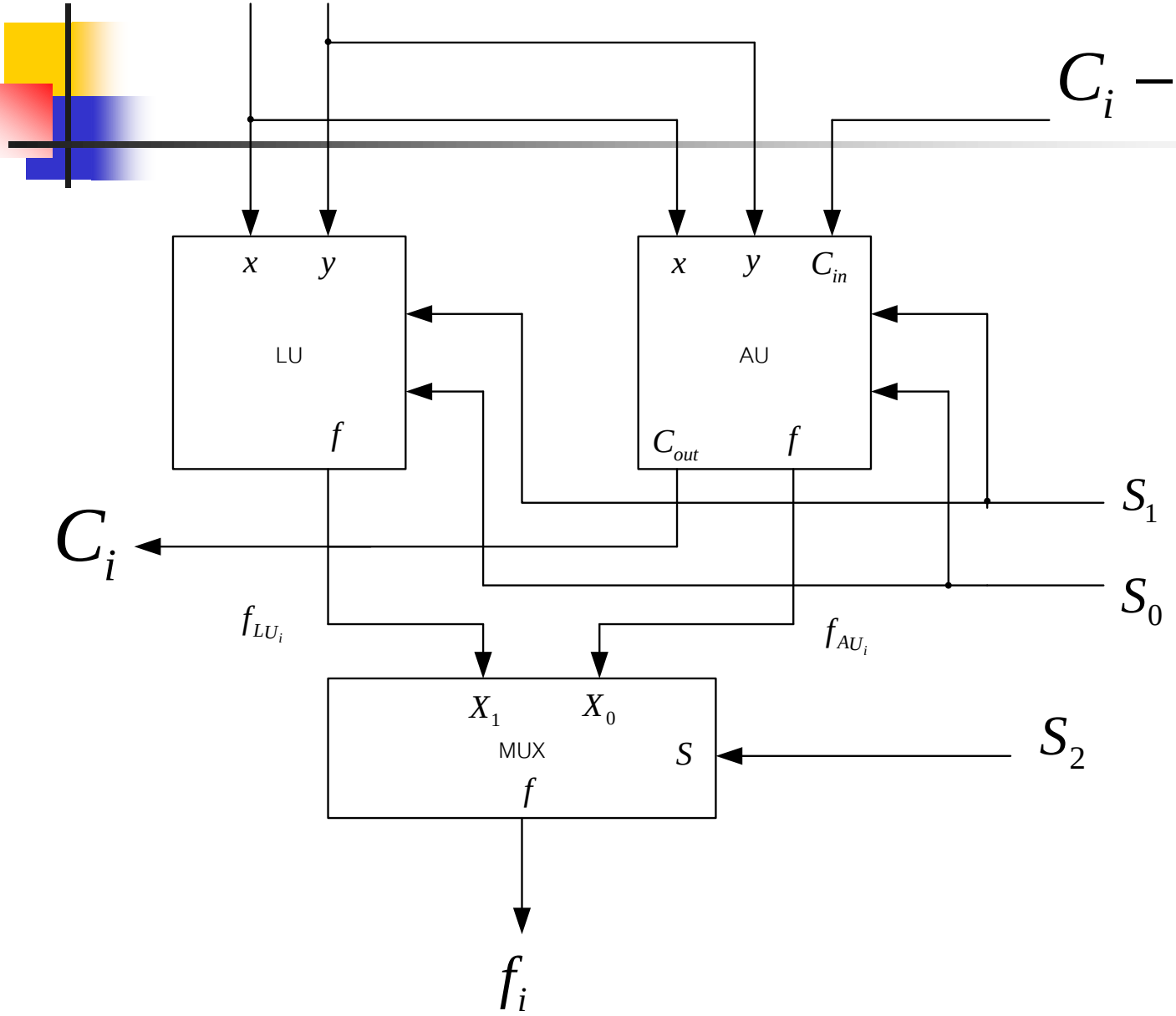
# การออกแบบวงจร ALU



# อธิบาย

- เพื่อให้เข้าใจการทำงานของ ALU อย่างถ่องแท้
- พิจารณาการออกแบบวงจร ALU ขนาด 1 บิต
- หน้าที่หลักของการทำงานของ ALU คือการคำนวณทางคณิตศาสตร์ และคำนวณทางตรรกศาสตร์
- อินพุตขนาด 1 บิตคือ $a_i$ และ $b_i$
- สัญญาณควบคุมแบ่งเป็น 2 ส่วน $S_2$ ควบคุม AL , AU
- สัญญาณควบคุม $S_0$, $S_1$ ควบคุมหน้าที่ย่อยของ AL, AU เช่น AND, OR , EX-OR
- $f_i$ เป็นสัญญาณเอาต์พุตที่เกิดจากการกระทำของ $a_1$,$b_1$
- $C_i$ จะเป็นตัวทด Carry Output ของบิตถัดไป
- $C_{i-1}$ เป็นบิตตัวทดอินพุต Carry Input ของหลัก $L_{SB}$

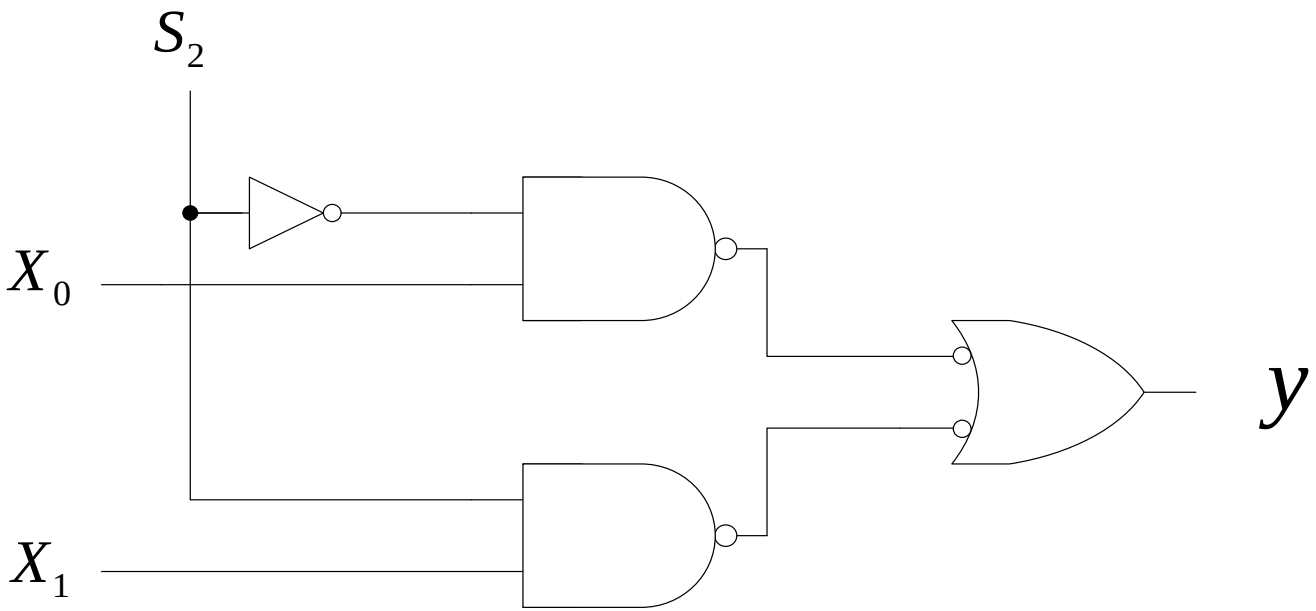
# $a_i$ $b_i$ โครงสร้าง ALU ขนาด 1 บิต



# อธิบาย

- การทำงานของหน่วยคำนวณทางคณิตศาสตร์ 4 ฟังก์ชัน กับการทำงานทางตรรกศาสตร์ 4 ฟังก์ชัน
- สามารถแบ่งส่วนของ ALU ขนาด 1 บิตเป็น 3 โมดุลคือ
  - ส่วนของ AU (Arithmetic Unit)
  - ส่วนของ LU (Logic Unit)
  - ส่วนของเอาต์พุตซึ่งเป็นวงจรมัลติเพล็กซ์ (Multiplex)
- $C_i$,$C_{i-1}$ เป็นการทำงานของตัวทดออกกับ ทดเข้าของวงจรบวก
- เอาต์พุตที่เกิดจาก AU Output ($f_i = f_{AU1}$)
- เอาต์พุตที่เกิดจาก LU Output ($f_i = f_{LU1}$)

# ส่วนเอาต์พุตเป็นวงจรมัลติเพล็กซ์



# อธิบาย

- วงจรมัลติเพล็กซ์เป็นตัวเลือกข้อมูลของฟังก์ชัน AU, LU จะควบคุมสัญญาณทางด้านเอาต์พุตสุดท้ายโดยมีสัญญาณควบคุมคือ $S_2$

  - $X_0$ มาจากเอาต์พุต AU ถ้า $S_2 = 0$ ข้อมูลที่มาจาก AU จะถูกควบคุมโดยแนนด์เกตผ่าน NOTในวงจรMUX ส่งต่อไปยังออร์เกต (NAND Gate ผ่าน NOT มีค่าเท่ากับ AND Gate) ได้สัญญาณทางฟังก์ชันทางการคำนวณทางคณิตศาสตร์จะถูกส่งไปยังเอาต์พุต Y หน้าที่นี้เป็นการทำงานทางคณิตศาสตร์ ผลได้ค่าบวก,ลบ,เพิ่มค่า,ลดค่า

  - $X_1$ มาจากเอาต์พุต LU ถ้า $S_2 = 1$ ข้อมูลที่มาจาก LU จะถูกควบคุมการส่งผ่านคล้ายกับ AU ข้อมูลทางฟังก์ชันทางตรรกศาสตร์จะถูกส่งไปยังเอาต์พุต Y หน้าที่นี้เป็นการทำงานทางตรรกศาสตร์ ผลของเอาต์พุตได้ค่า AND/OR/X-OR/NOT

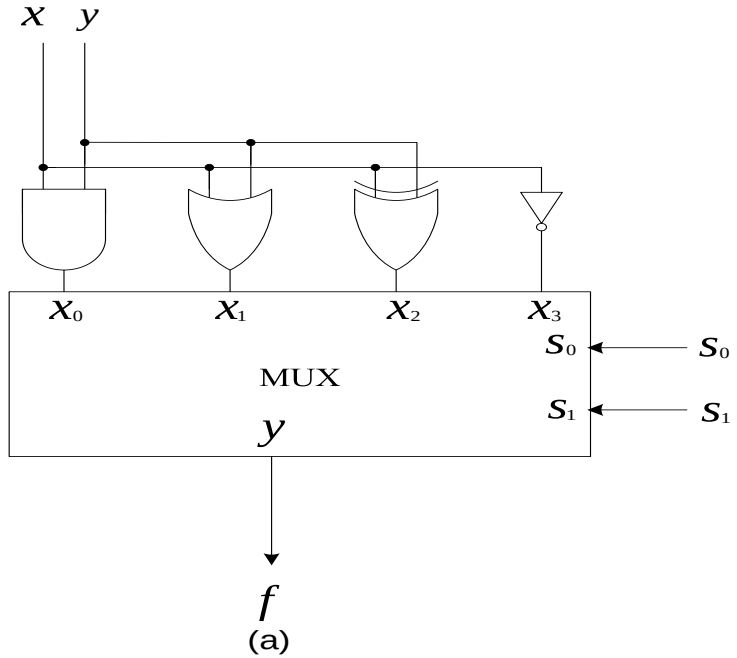
# ตารางหน้าที่ Logic Unit

| Function | | $S_1$ | $S_0$ | $f_{LUi}$ |
|---|---|---|---|---|
| AND : | $F = A \cap B$ | 0 | 0 | $a_i b_i$ |
| OR : | $F = \overline{A} \cup B$ | 0 | 1 | $\overline{a}_i + b_i$ |
| NOT : | $F = A$ | 1 | 0 | $a_i$ |
| XOR : | $F = A \oplus B$ | 1 | 1 | $a_i \oplus b_i$ |

# อธิบาย

- จากตารางหน้าที่ของวงจรคำนวณทางตรรกศาสตร์ แบ่งเป็น 4 ฟังก์ชัน ดังตารางการทำงาน
- หน้าที่ของAND ตัวเลือกหน้าที่ $S_1$=0,$S_0$=0 จะทำหน้าที่นำค่า $a_1$AND $b_1$
- หน้าที่ของOR ตัวเลือกหน้าที่ $S_1$=0,$S_0$=1 จะทำหน้าที่นำค่า $a_1$OR $b_1$
- หน้าที่ของNOT ตัวเลือกหน้าที่ $S_1$=1,$S_0$=0 จะทำหน้าที่นำค่า $a_1$ NOT เป็น $\bar{a}_1$
- หน้าที่ของEX - OR ตัวเลือกหน้าที่ $S_1$=1 , $S_0$=1 จะทำหน้าที่นำค่า $a_1$ บวก $b_1$

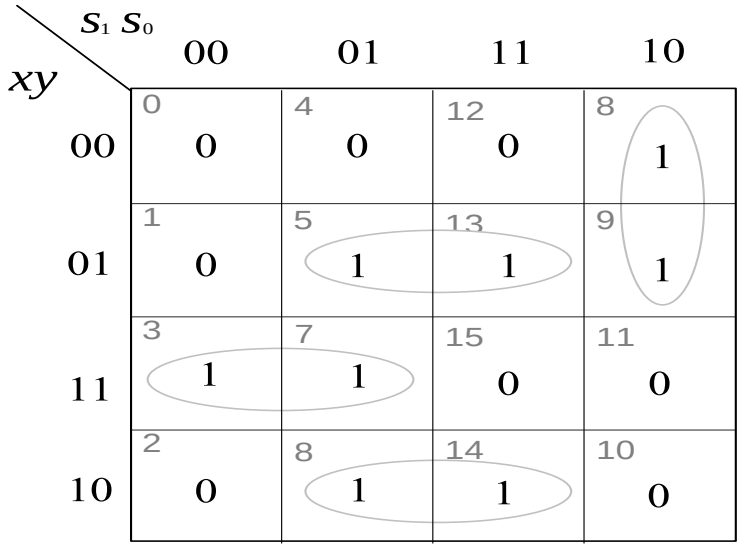
# 531 การทำงานของหน่วยลอจิก(LU)



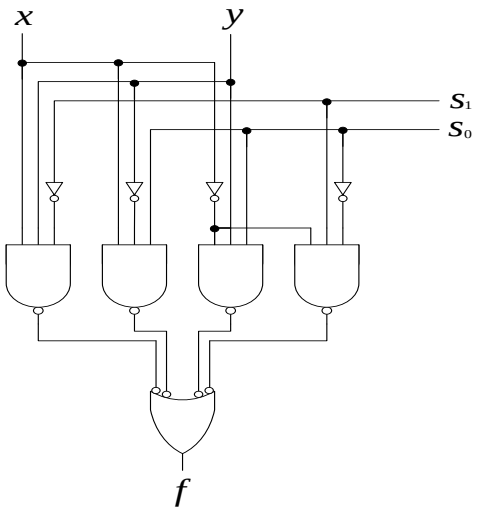

(a)

| $S_1$ | $S_0$ | $x$ | $y$ | $F_{LU}$ | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | $x$ AND $y$ |
| 0 | 0 | 0 | 1 | 0 | |
| 0 | 0 | 1 | 0 | 0 | |
| 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 0 | $x$ OR $y$ |
| 0 | 1 | 0 | 1 | 1 | |
| 0 | 1 | 1 | 0 | 1 | |
| 0 | 1 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | 1 | $x$ NOT $y$ |
| 1 | 0 | 0 | 1 | 1 | |
| 1 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 0 | $x$ XOR $y$ |
| 1 | 1 | 0 | 1 | 1 | |
| 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 1 | 0 | |

(b)




(c)




(d)

# อธิบาย

- จากรูป (a) การปฏิบัติการทางตรรกศาสตร์มีฟังก์ชันที่กล่าวมา

$$X_0 = XY, \quad X_1 = X + Y, \quad X_2 = X \oplus Y, \quad X_3 = \overline{X}$$

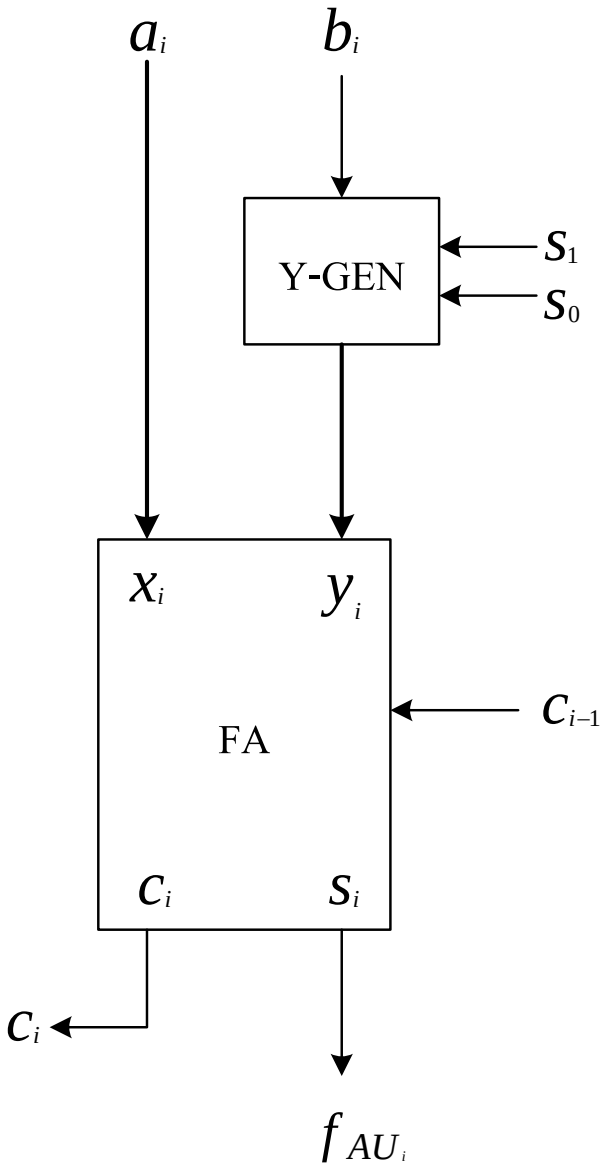
- เอาต์พุตของวงจรมัลติเพล็กซ์สำหรับเลือกข้อมูลโดยมีสัญญาณควบคุมย่อย $S_1$,$S_0$
- ดูตารางประกอบ $S_1$= 0, $S_0$= 0 จะเลือกค่าเอาต์พุต $X_0$ เป็นการกระทำ AND

  $S_1$= 0, $S_0$= 1 จะเลือกค่าเอาต์พุต $X_1$ เป็นการกระทำ OR

  $S_1$= 1, $S_0$= 0 จะเลือกค่าเอาต์พุต $X_2$ เป็นการกระทำ EX-OR

  $S_1$= 1, $S_0$= 1 จะเลือกค่าเอาต์พุต $X_3$ เป็นการกระทำ NOT
- จากรูป (b) เป็นตารางการปฏิบัติการทางตรรกศาสตร์ 4 ฟังก์ชัน
- จากรูป (c) นำค่าจากตารางความจริงที่มีตัวแปร 4 ค่า $S_1$, $S_0$, X, Y ที่ควบคุมการทำงานมาลงค่าในตาราง K-map ได้ค่าเป็น $f_{LU} = \overline{S_1}XY + S_0X\overline{Y} + S_0\overline{X}Y + S_1\overline{S_0}\,\overline{X}$
- จากรูป (d) นำฟังก์ชันจากการลดรูปมาเขียนเป็นวงจรลอจิกเกต ได้เป็น LU ขนาด 1 บิต

# 532 การทำงานของหน่วยคำณวนทางคณิตศาสตร์ (AU)



# อธิบาย

- หน้าที่ของการปฏิบัติการทางคณิตศาสตร์มี 4 หน้าที่การทำงานคือ บวก, ลบ, เพิ่มค่า, ลดค่า ดังตาราง
- วงจรAU ประกอบด้วยวงจรหลักคือ วงจรบวกเลข และวงจรสร้างฟังก์ชันทางคณิตศาสตร์
- หน้าที่หลักของวงจร AU มีการทำงานที่ถูกกำหนดโดย $S_1$ , $S_0$ ที่เป็นส่วนสำคัญ

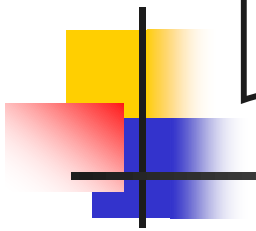
  **ในการกำหนดหน้าที่ทั้ง 4 ที่กล่าวคือ**

  $S_1$ , $S_0$ = 00 กำหนดเป็นฟังก์ชันบวกค่า $a_i$ กับ$b_i$

  $S_1$ , $S_0$ = 01 กำหนดเป็นฟังก์ชันลบค่า $a_i$ ลบกับ $b_i$ ($a_i$+Complement 2 ของ$b_i$)

  $S_1$ , $S_0$ = 10 กำหนดเป็นฟังก์ชันเพิ่มค่าที่ $a_i$

  $S_1$ , $S_0$ = 11 กำหนดเป็นฟังก์ชันลดค่าที่ $a_i$
- โดยมีวงจร Y-Gen. ทำงานร่วมกับวงจรบวกแบบ FA
- AU ที่ประกอบจากวงจรทั้งสองจะสร้างค่าทางเอาต์พุตทั้งหมดที่กล่าวมาเป็นค่า $f_{AUi}$



สัญญาณอินพุตของY-Gen

สัญญาณเอาต์พุตของ Y-Gen

| หน้าที่ | $S_1$ | $S_0$ | $y_1$ | $C_{-1}$ |
|---|---|---|---|---|
| บวก | 0 | 0 | $b_i$ | 0 |
| ลบ | 0 | 1 | $\bar{b}_i$ | 1 |
| เพิ่มค่า | 1 | 0 | 0 | 1 |
| ลดค่า | 1 | 1 | 1 | 0 |

สัญญาณควบคุม

| $S_1$ | $S_0$ | $b_1$ | $y_1$ | $C_{-1}$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 |

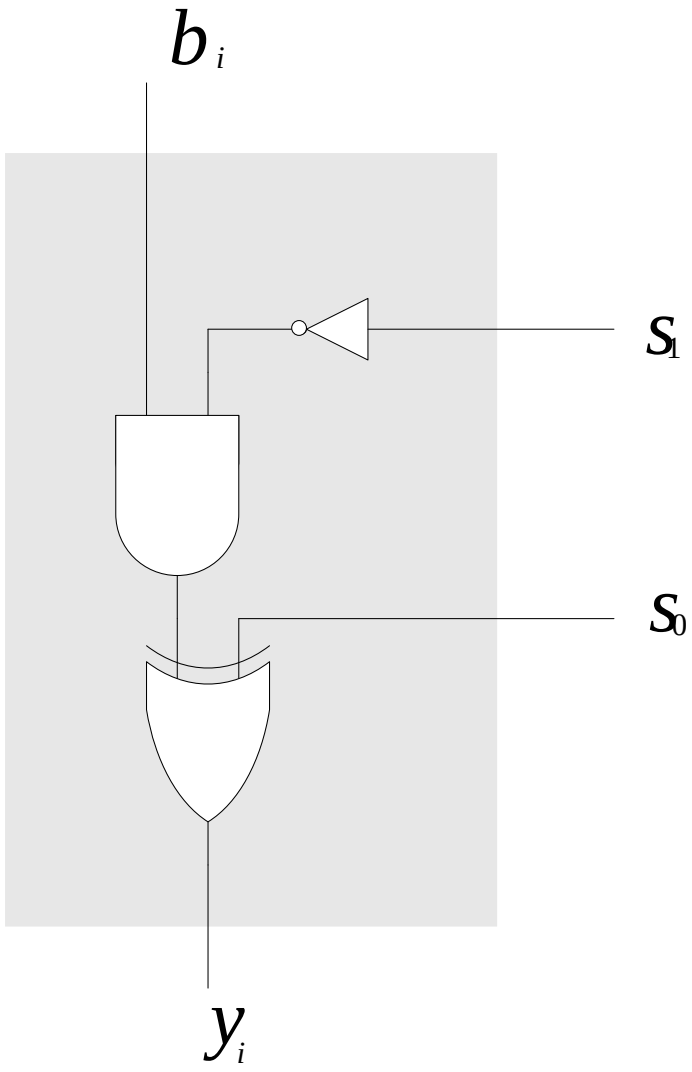
ตารางการทำงานของวงจรกำเนิดฟังก์ชันทางคณิตศาสตร์

# อธิบาย

- จากหลักการหน้าที่ของฟังก์ชันทางคณิตศาสตร์สามารถนำมาเขียนเป็นตารางความจริง
- หน้าที่หลักของวงจร Y-Gen นี้คือการสร้างฟังก์ชันทางคณิตศาสตร์ของ $Y_i$ โดยมีอินพุตควบคุมการทำงาน คือ $S_1$, $S_0$ ที่มีอินพุตตัวแปรคือ $b_i$
- จากตารางจะเห็นว่าค่า $Y_i$ จะเปลี่ยนค่าตามหน้าที่ของการกระทำทางคณิตศาสตร์ดังนี้

  - $Y_i = b_i$ เมื่อเป็นการกระทำที่ $a_i+b_i$ รูปแบบปกติการบวก
  - $Y_i = -b_i$ เมื่อเป็นการกระทำที่ $a_i-b_i$ รูปแบบบวกแบบ2'S
  - $Y_i = 0$ เมื่อเป็นการกระทำที่ $a_i+1$ รูปแบบปกติบวกเพิ่มค่า
  - $Y_i = 1$ เมื่อเป็นการกระทำที่ $a_i-1$ รูปแบบบวกแบบ2'S

- ค่า $C_{-1}$ ก็เช่นกัน จะกำเนิดค่าตามฟังก์ชันทางคณิตศาสตร์ เป็นค่า 1 ก็ต่อเมื่อมีการทำงานเป็นฟังก์ชันทางคณิตศาสตร์ลบ และเพิ่มค่า นักศึกษาลองทำความเข้าใจด้วยการแทนค่าอีกครั้ง




| $b_1$ \ $S_1 S_0$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 2 **1** | 6 **1** | 4 **0** |
| 1 | 1 **1** | 3 **0** | 7 **1** | 5 **0** |

$$y_i = \overline{S_1}\,\overline{S_0} b_i + S_0 \overline{b_i} + S_1 S_0$$

$$y_i = \overline{S_0}\left(\overline{S_1} b_i\right) + S_0\left(\overline{b_i} + S_1\right)$$

$$y_i = S_0 \oplus \overline{S_1} b_i$$

# อธิบาย

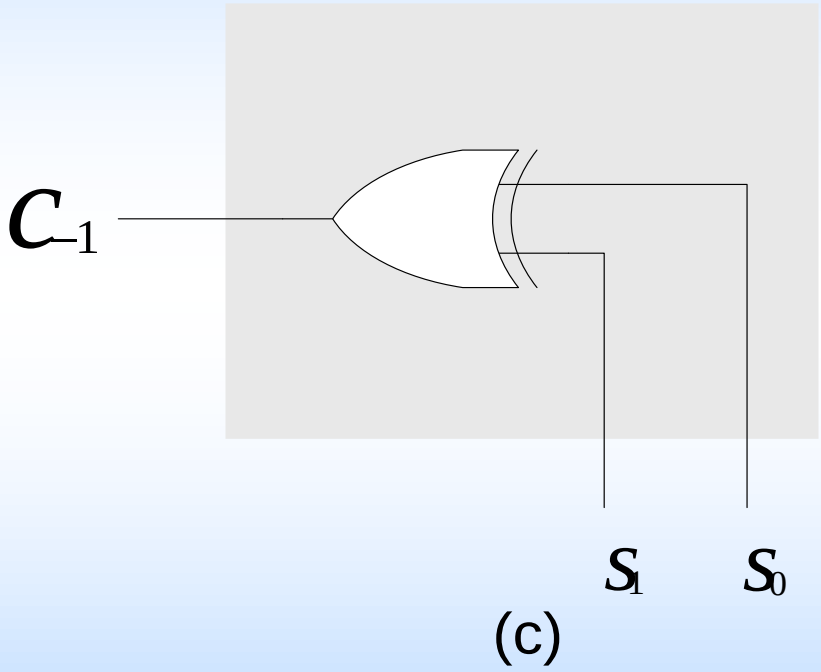
- จากตารางความจริงนำค่า $y_i$ มาเขียนลงใน k-Map เพื่อทำการลดรูปสมการ
- จากสมการที่ทำการลดรูปจัดสมการให้เหมาะแก่การใช้งาน
- สามารถสร้างเป็นวงจรกำเนิดหน้าที่ทางคณิตศาสตร์ให้ค่า $y_i$ กำหนดหน้าที่ดังที่กล่าวมาแล้ว
- วงจร Y-Gen ซึ่งประกอบด้วยวงจร Ex-OR ทำงานร่วมกับ AND และ NOT Gate ดังรูป
- วงจร Y-Gen ทางด้านเอาต์พุตจะนำไปเป็นอินพุตของวงจร FA แสดงดังรูปวงจร ALU

| $S_1$ | $S_0$ | $C_{-1}$ | |
|---|---|---|---|
| 0 | 0 | 0 | Add |
| 0 | 1 | 1 | Subtract |
| 1 | 0 | 1 | Increment |
| 1 | 1 | 0 | Decrement |

(a)

| $S_0$ \ $S_1$ | 0 | 1 |
|---|---|---|
| 0 | 0 | **1** |
| 1 | **1** | 0 |

(b)




(c)

# อธิบาย

- พิจารณาจากตารางความจริงรูป (a) สามารถเขียนค่า $C_{-1}$ จะเป็นค่าที่เกิดจาก$S_1 + S_0$ไม่คิดตัวทดเป็นไปดังการทำงานของ NOR Gate
- อีกวิธีนำค่าในตารางไปลงใน K-Map ดังรูป (b) หาค่าออกมาก็จะได้เป็น

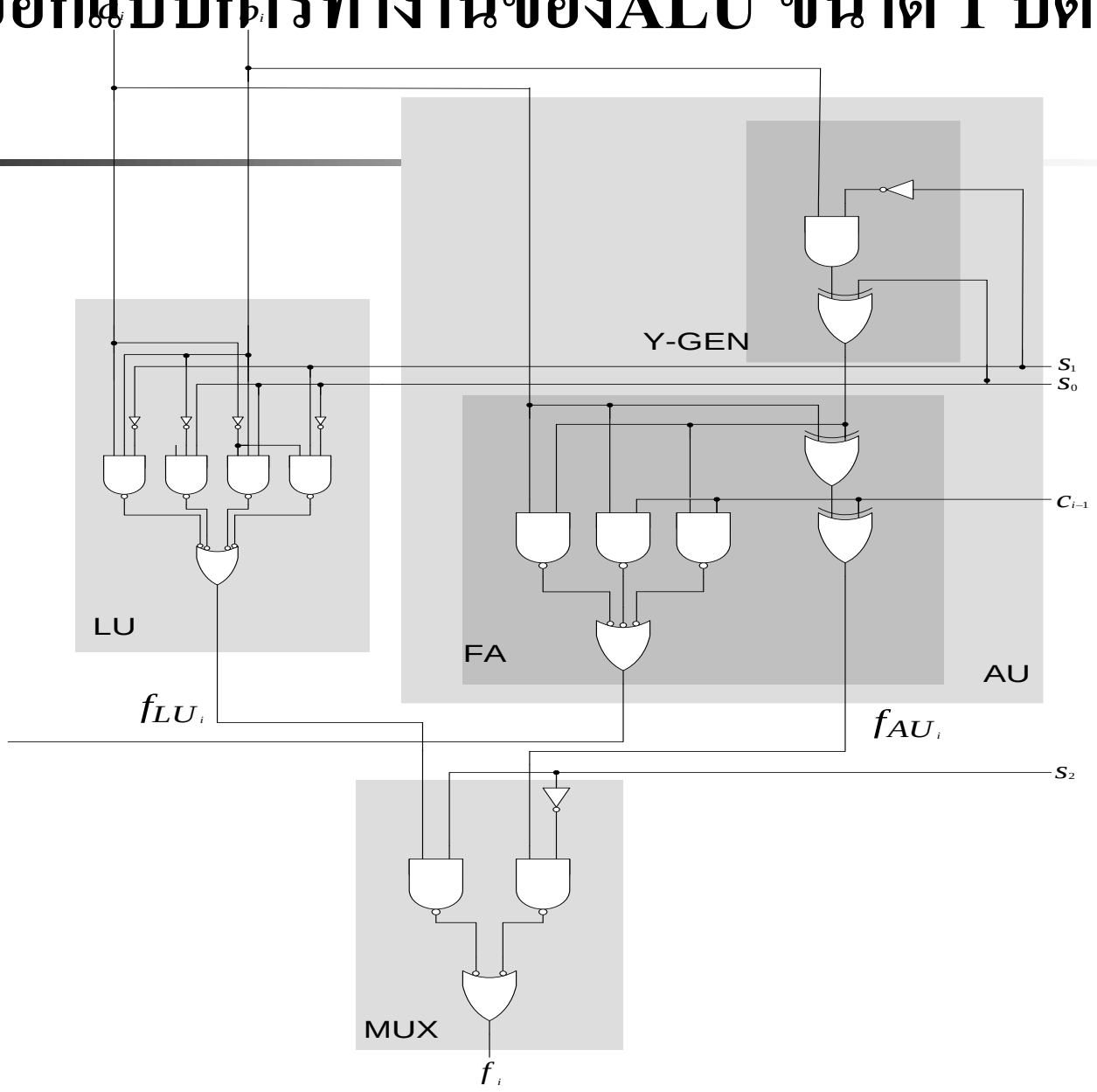
$$C_{-1} = S_1\overline{S_0} + \overline{S_1}S_0$$

$$C_{-1} = S_1 \oplus S_0$$

- สามารถนำไปเขียนเป็นวงจรได้ดังรูป (C)
- จากนั้นนำวงจรไปต่อเข้ากับ FA ดังแสดงรูปถัดไป

# 54 หลักการออกแบบ ALU

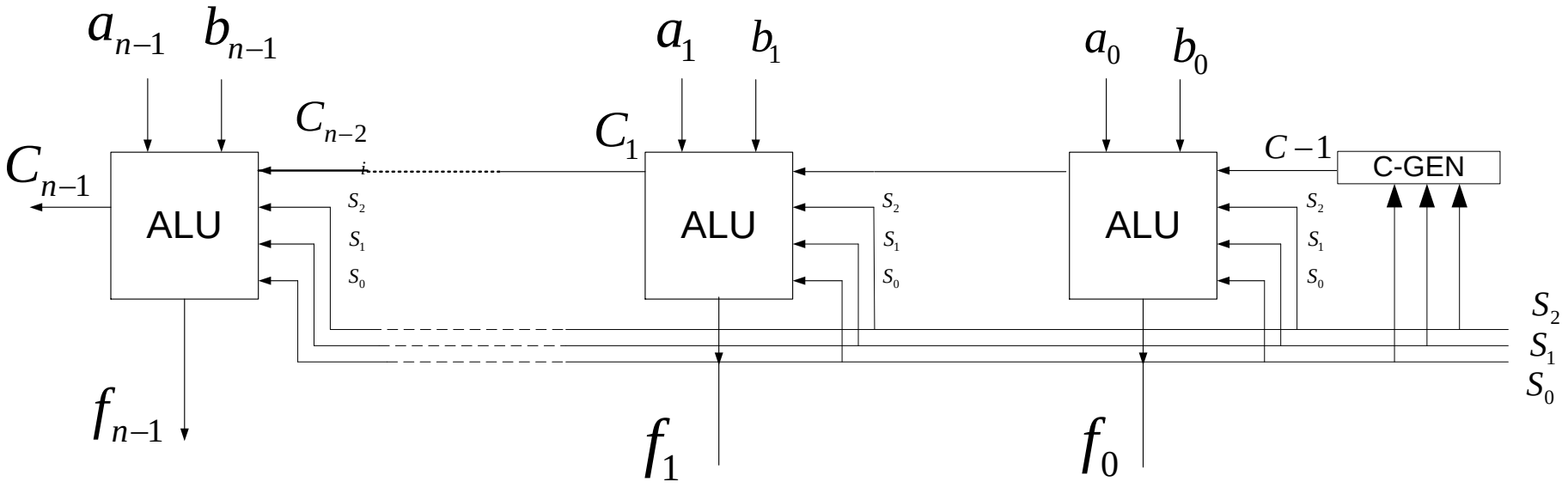
## 541 การออกแบบการทำงานของALU ขนาด 1 บิต



# อธิบาย

- การทำงานของ ALU ขนาด 1 บิตประกอบด้วย 3 ส่วนใหญ่ ๆ คือ LU, AU, MUX
- ส่วนประกอบของการทำงานของวงจร LU ทางด้านซ้ายมือจะทำการปฏิบัติการทางตรรกศาสตร์ มีเอาต์พุตเป็น $f_{LUi}$
- การทำงานของ AU ประกอบด้วยวงจร Full Adder กับวงจร Y-Gen ทางด้านขวามือ จะปฏิบัติการทางคณิตศาสตร์ ให้ค่าทางเอาต์พุตเป็น $f_{AUi}$ การทำงานทางด้านคณิตศาสตร์จำเป็นต้องทดเลขจากหลักน้อยไปหามากจะมีการทดเข้า $C_{i-1}$ และการทดออก $C_i$
- Muxการทำงานสัญญาณทางเอาต์พุตที่มีการทำงาน ด้วย AU หรือ LU จะถูกควบคุมการทำงานด้วยวงจรMuxตัวเลือกสัญญาณทางอินพุตเป็น$f_{LUi}$หรือ $f_{AUi}$ เอาต์พุต โดยมี$S_2$ควบคุมการทำงานในโหมด AU/LUให้ค่าทางด้านเอาต์พุตเป็น $f_i$

# 542 การออกแบบวงจรALU ขนาด n บิต



# อธิบาย

- การนำเอา ALU ขนาดย่อยๆ มาทำการต่อ n บิต ดังรูปแสดง
- นำวงจร ALU ขนาด 1 บิตมาต่อลักษณะ Cascade กัน
- การทำงานของALU แต่ละบิตมีการทำงานอิสระซึ่งกันและกัน
- การควบคุมจะมีการเปรียบเทียบโยงเข้าหากัน โดยบิตสูงสุดจะมีตัวทดเข้า ส่วนตัวทดออกจะมาจากบิตต่ำ เพื่อทำงานทางคณิตศาสตร์ที่มีการเปรียบเทียบค่าตัวเลข
- Control Generator จะควบคุมการทำงานของบิต ต่ำสุดเป็นบิตแรก
- การกำหนดหน้าที่ของ 8 ฟังก์ชันของงานทางคณิตศาสตร์ และตรรกศาสตร์ ควบคุมโดย $S_2, S_1, S_0$ ที่ต่อของแต่ละบิตเป็นแบบขนานกันเพื่อให้มีการทำงานในหน้าที่เดียวกันทุกบิต

# ทดสอบความรู้สัปดาห์ที่8

1. จงอธิบายการเปรียบเทียบทางแอนะลอกกับทางดิจิตอล ว่ามีการทำงานอย่างไร พร้อมข้อดีข้อเสีย
2. จงอธิบายคำจำกัดความของ ALU ที่นักศึกษาได้เรียนผ่านมาให้เข้าใจ
3. หน้าที่ของ Y-Gen ในวงจร ALU มีหน้าที่อย่างไร และนักศึกษา จงออกแบบพื้นฐานการทำงานตามลำดับขั้น พร้อมยกตัวอย่างให้เห็นจริง
4. วงจร LU , AU มีความหมายแตกต่างกันอย่างไร และ จงอธิบายฟังก์ชันของแต่ละวงจรประกอบ

# เนื้อหาคำบรรยายในสัปดาห์ที่ 10

# การออกแบบวงจรบวก ลบ คูณ หาร

**55หลักการวงจรบวก**

551 การออกแบบวงจรบวกแบบไม่คิดตัวทดเข้า(Half Adder)

552 การออกแบบวงจรบวกแบบคิดตัวทดเข้า(Full Adder)

553 วงจรบวกโดยวิธีการมองตัวทดล่วงหน้า(Carry Look Ahead)

554 หลักการบวก4 จำนวนแบบ Carry Save

**56 หลักการวงจรลบ**

561 การออกแบบวงจรลบแบบไม่คิดตัวทดเข้า(Half Subtracter)

562 การออกแบบวงจรลบแบบคิดตัวทดเข้า(Full Subtracter)

563 การออกแบบวงจรบวก/ลบเลขขนาด 2 บิตแบบใช้ 2'Complement

# เนื้อหาคำบรรยายในสัปดาห์ที่ 10

## การออกแบบวงจรบวก ลบ คูณ หาร (ต่อ)

**57 หลักการวงจรคูณ**

571 การออกแบบวงจรคูณ

572 การออกแบบวงจรคูณขนาด 2 บิต แบบใช้โมดุล

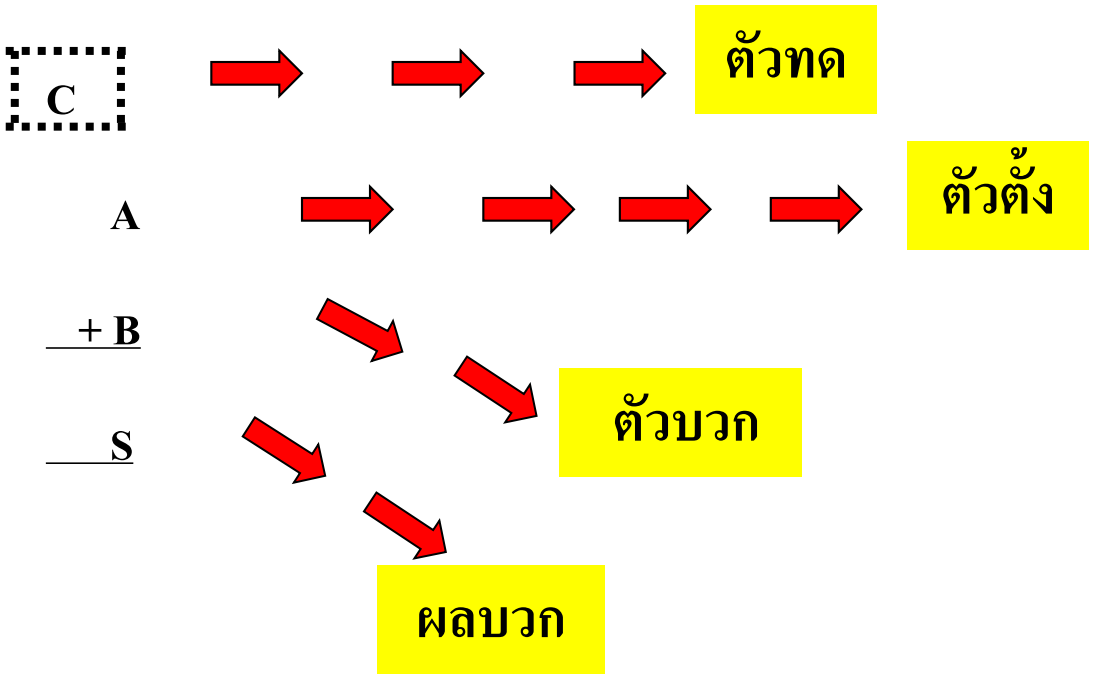
**58 หลักการวงจรหาร**

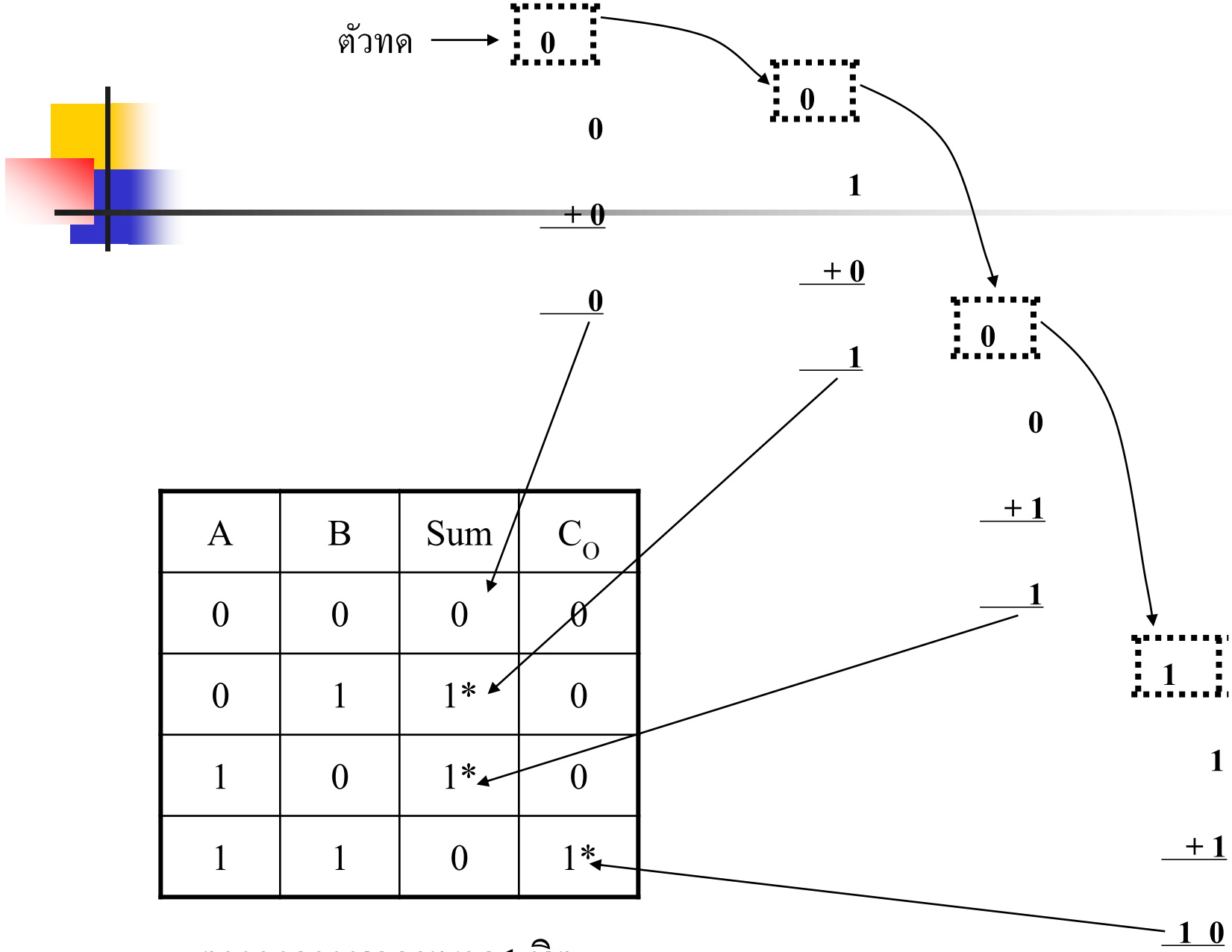
581 การออกแบบวงจรหาร

582 การออกแบบหารเลขแบบ Algorithm ขนาด 2 บิต

# 55 หลักการวงจรบวก

## หลักการของการบวกเลขบวก






| A | B | Sum | $C_O$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1* | 0 |
| 1 | 0 | 1* | 0 |
| 1 | 1 | 0 | 1* |

**ตารางการบวกขนาด1 บิต**

# อธิบาย

**กรณีที่ 1** เมื่อ A = 0 และ B = 1 ซึ่งได้ว่า $\overline{A}\,B$

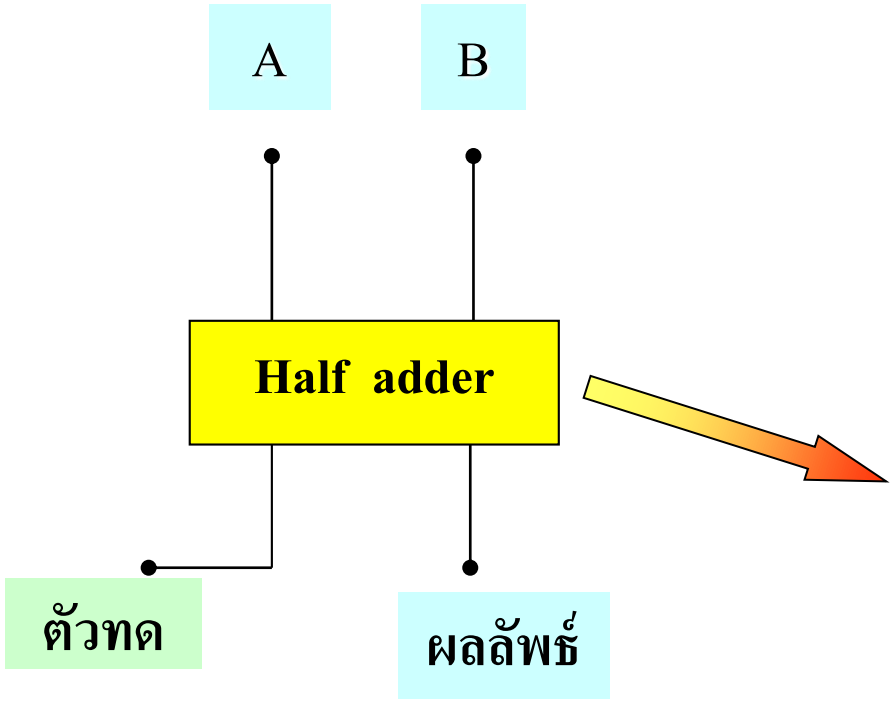
**กรณีที่ 2** เมื่อ A = 1 และ B = 0 ซึ่งได้ว่า $A\,\overline{B}$

ดังนั้นจึงสรุปได้ว่า S เป็น ”1” ได้เมื่อ $\overline{A}\,B$ หรือ $A\,\overline{B}$ ซึ่งก็ คือ $S = \overline{A}\,B + A\,\overline{B}$ ส่วนฟังก์ชันของตัวทดพิจารณากรณีที่ c = 1 ซึ่งมีกรณีเดียว คือ ทั้ง A และ B เป็น 1 ทั้งคู่ ซึ่งก็ได้ว่า C = AB

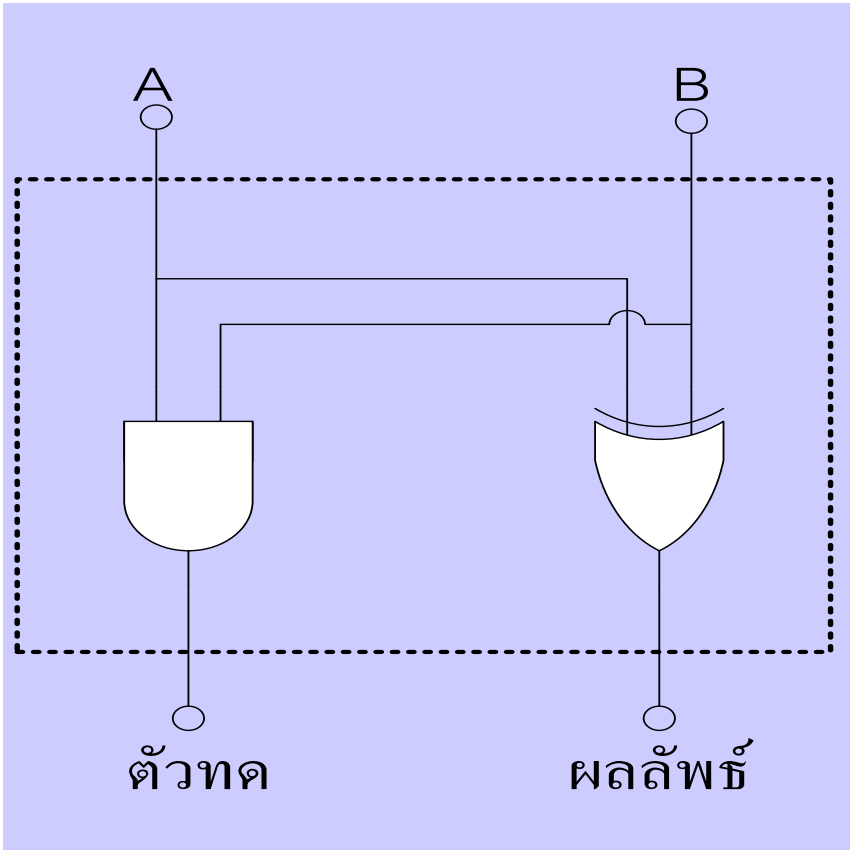
ฟังก์ชันการบวกที่ได้กล่าวมานี้ จะเห็นว่าเป็นการบวกที่ “หลักต่ำสุดซึ่งไม่มีการทดเข้ามาจากหลักต่ำกว่า” มีชื่อเรียกว่า การบวกแบบไม่คิดตัวทดเข้า ( Half Adder หรือ HA)

สามารถสร้างเป็นวงจรเกตทางดิจิตอลได้ดังนี้

# 551 การออกแบบวงจรบวกแบบไม่คิดตัวทดเข้า (Half Adder)




วงจร Half Adder



# ตารางความจริง

| B | A | ผลลัพธ์ | ตัวทด |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

# อธิบาย

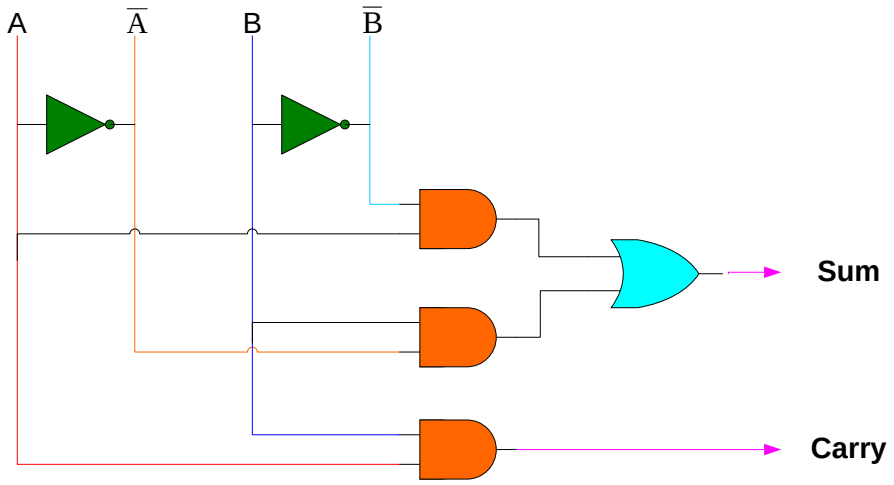
ฟังก์ชันของผลบวก (Sum) และตัวทด ($C_o$) ได้ดังนี้

**ผลบวก** = $S = \bar{A}B + A\bar{B}$

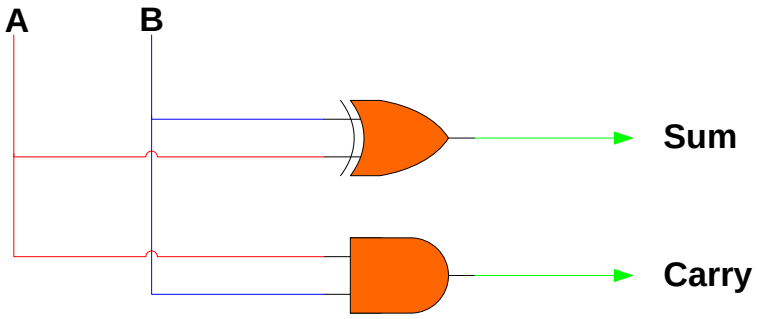
**ตัวทด** = $C_o = AB$

ฟังก์ชันของผลบวกนั้นจะได้มาจากการพิจารณาที่ตารางค่าของ S

โดยพิจารณาว่า S นั้นจะมีสภาวะลอจิกเป็น “ 1 ” ขณะที่อินพุตของ A , B เป็นได้ 2 กรณี




$$Sum = A\overline{B} + \overline{A}B$$




$$Sum = A \oplus B$$

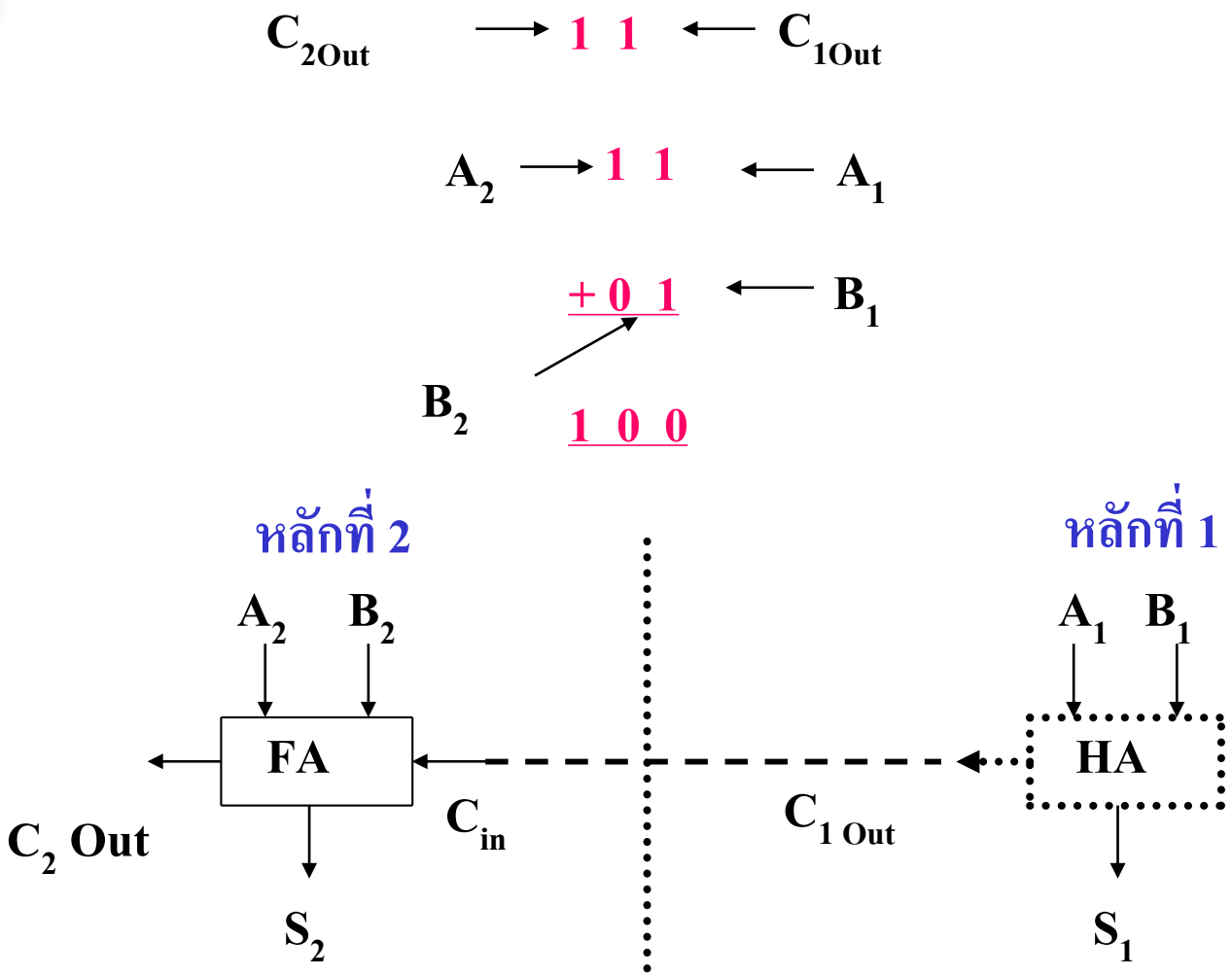
# รูปวงจร Half Adder

# อธิบาย

- จากตารางความจริงสามารถนำค่าผลลัพธ์มาในรูป Minterm ได้ค่า $\bar{A}B+A\bar{B}$
- สามารถออกแบบวงจรง่ายๆ โดยมีค่า A,B ทำการ EX-OR กัน
- วงจรบวกเลขแบบนี้เป็นวงจรบวกขนาด 1 บิต ไม่คิดตัวทดเข้า
- วงจรนี้มักจะใช้เป็นหลัก $L_{SB}$ บิต0 หรือบิต1
- จากตารางเช่นกันค่าตัวทดเกิดจากกรณีที่ค่า A = 1 , B = 1 ค่าเอาต์พุตจะมีค่าเท่ากับ 1 กรณีเดียว ดังนั้นผลของเอาต์พุตเกิดจาก A AND B
- สามารถสร้างเป็นโมดุลอย่างง่ายประกอบด้วย EX-OR กับ AND Gate

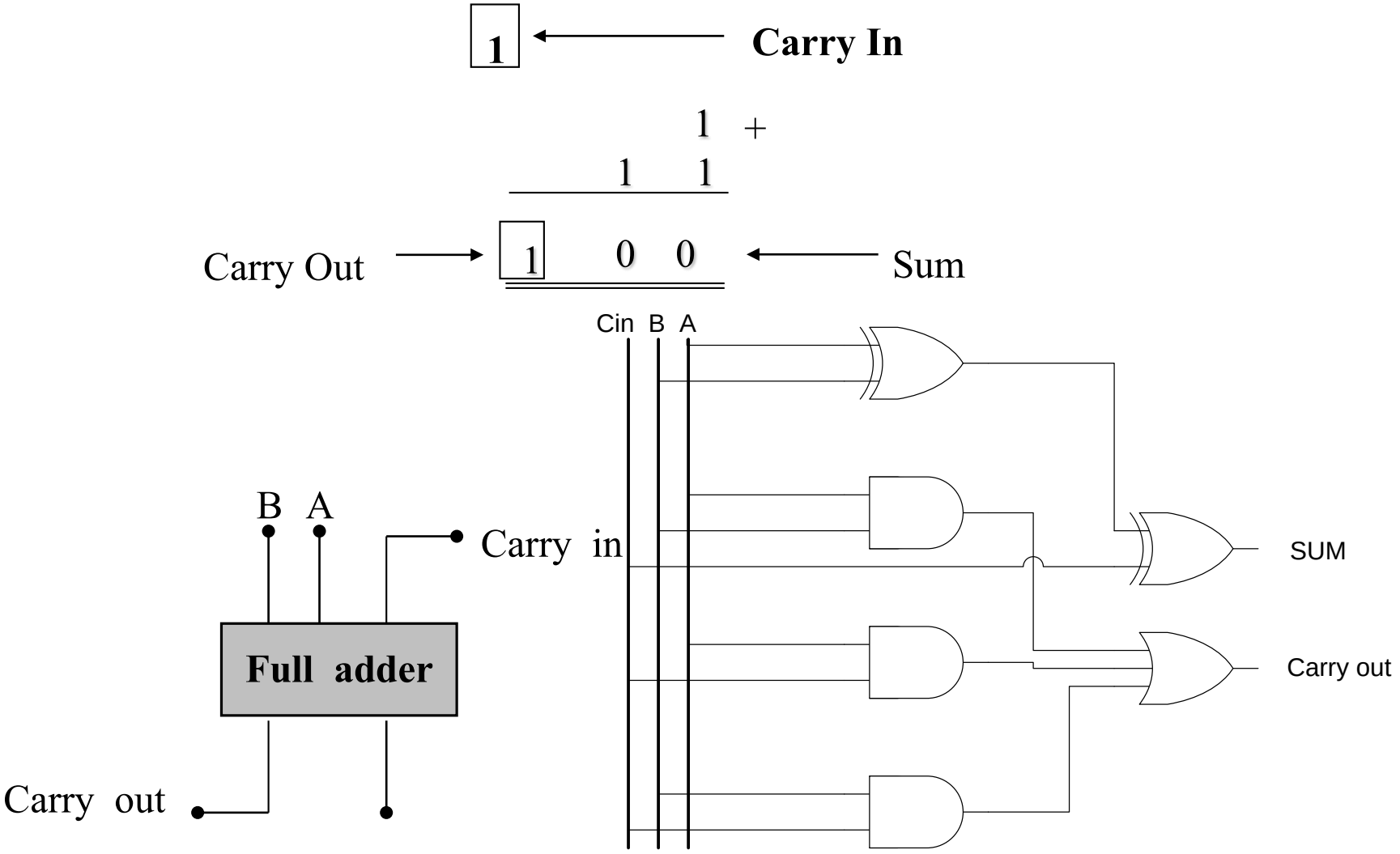
# 552 การออกแบบวงจรบวกแบบคิดตัวทดเข้า (Full Adder)

## วงจรบวกแบบคิดตัวทดเข้าจากหลักต่ำกว่า (Full Adder หรือ FA)



# วงจร FULL ADDER

การบวกเลขฐานสองหลายบิต จะต้องนำเอาวงจร Half Adder มาใช้รวมกันดังขั้นตอนการออกแบบ



# อธิบาย

- การบวกโดยทั่วไปจะเกิดจากการนำบิตทดเข้ามาบวกด้วย ดังนั้นการบวกแบบ Half Adder จึงไม่สามารถทำงานครอบคลุมในบิตต่อไป
- วงจร Full Adder จึงมีความจำเป็นมากสำหรับการบวกในบิตที่ 2 จนถึงบิต n
- ค่าบิตตัวทดออกจากวงจร Half Adder แทนด้วย $C_{1out}$ จะถูกส่งไปบวกกับวงจร Full Adder ในบิตที่ 2 $C_{2in}$
- วงจรบวกขนาดหลาย 2 บิต ค่าที่เกิดขึ้นจะประกอบด้วย $C_{2out}$ , $S_2$ , $S_1$
- จากตัวอย่างค่า ตัวตั้งหลัก A= 11 หลัก B= 01 เมื่อทำการบวกกัน ค่าที่ได้
- $C_{2\,out}=1$ , $S_2=0$ , $S_1=0$

# อธิบาย(ต่อ)

- การบวกแบบ 2บิต $A_2$, $A_1$ เป็นตัวตั้งส่วนค่า $B_2$, $B_1$ เป็นตัวบวก
- ค่าที่เกิดขึ้นจึงบวกจากหลัก $A_1$, $B_1$ ได้ผลลัพธ์กับตัวทดออก (Carry Out)ในที่นี้คือ $C_1$
- นำค่า $C_1$ ไปบวกกับตัวตั้งในหลักถัดไป
- ค่าที่เกิดในหลักที่ $A_2$, $B_2$ นำค่าทั้งสองมาบวกกันได้ผลลัพธ์นำไปบวกกับ $C_1$ จะได้เป็นผลลัพธ์ของหลักที่ 2 กับ $C_2$
- นำค่า $C_2$ ไปบวกกับหลักถัดไปในที่นี้มีเพียง 2 บิต ค่า $C_2$ จะเป็นผลลัพธ์ของบิตหลักที่ 3 ($M_{SB}$)
- วงจร$A_1$,$B_1$ ใช้วงจรบวกแบบไม่คิดตัวทดเข้า (Carry In) ใช้วงจร (Half Adder)
- วงจรที่ทำการบวก $A_2$,$B_2$ ใช้วงจรบวกแบบคิดตัวทดเข้าด้วยในที่นี้ใช้วงจร (Full Adder)
- ดังแสดงในรูปถัดไป

# ตารางความจริง

| | A | B | $C_i$ | S | $C_O$ | |
|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | |
| $\overline{A}\overline{B}Ci$ | 0 | 0 | 1 | 1 | 0 | |
| $\overline{A}B\overline{Ci}$ | 0 | 1 | 0 | 1 | 0 | |
| | 0 | 1 | 1 | 0 | 1 | $\overline{A}BC$ |
| $A\overline{B}\overline{Ci}$ | 1 | 0 | 0 | 1 | 0 | |
| | 1 | 0 | 1 | 0 | 1 | $A\overline{B}Ci$ |
| | 1 | 1 | 0 | 0 | 1 | $AB\overline{Ci}$ |
| $ABCi$ | 1 | 1 | 1 | 1 | 1 | $ABCi$ |

วงจรบวกเลขแบบคิดตัวทดเข้าจากหลักต่ำกว่า

# ตารางความจริงการทำงานของ วงจร Full Adder

| Carry In | B | A | Sum | Carry Out |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**Carry Out**

$$= AB\overline{Cin} + A\overline{B}Cin + \overline{A}BCin + ABCin$$

$$= AB\left(\overline{Cin} + Cin\right) + CinA\overline{B} + Cin\overline{A}B$$

$$= AB + Cin\left(A \oplus B\right)$$

**Sum**

$$= \overline{Cin}\left(A\overline{B} + \overline{A}B\right) + Cin\left(\overline{A\overline{B}} + AB\right)$$

$$= A\overline{B}\,\overline{Cin} + \overline{A}B\,\overline{Cin} + \overline{A\overline{B}}Cin + ABCin$$

$$= \overline{Cin}\left(A \oplus B\right) + Cin\left(\overline{A \oplus B}\right)$$

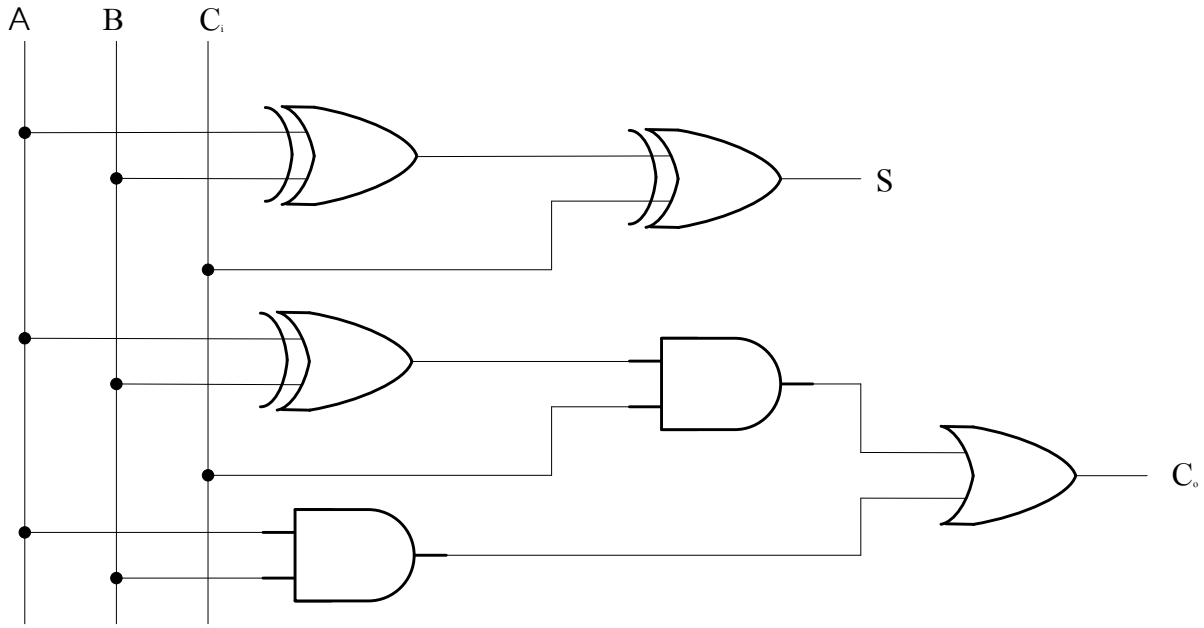
$$= Cin \oplus \left(A \oplus B\right)$$

# อธิบาย

- จากตารางเป็นการกำหนดความจริงในการบวกแบบคิดตัวทดเข้า (Full Adder)
- พิจารณาจากอินพุตที่เกิดการเปลี่ยนแปลงมีตัวทดเข้า Carry In ,ค่าตัวบวกเป็น B , ค่าตัวตั้งเป็น A
- เอาต์พุตที่เกิดขึ้นเป็นผลบวกให้เป็น Sum และตัวทดออก Carry Out
- ผลบวกสามารถหาในรูปแบบMinterm ( 1,2,4,7) ได้ค่า A EX-OR B EX-OR Cin
- ส่วนค่า Carry out สามารถหาในรูปแบบMinterm ( 3,5,6,7) ได้ค่า A AND B OR $C_{in}$ AND( A EX-OR B)
- พิจารณาง่ายโดยนำวงจร Half Adder 2 วงจรมาต่อกันดังรูป

# วงจรบวกเลขแบบคิดตัวทดเข้าจากหลักต่ำกว่า

จะเห็นได้ว่าการลดรูปด้วย Boolean ได้ผลที่น่าสนใจกว่าการลดด้วย K-map กล่าวคือเมื่อนำไปสร้างเป็นวงจรนั้นจะสะดวกกว่า



# อธิบาย

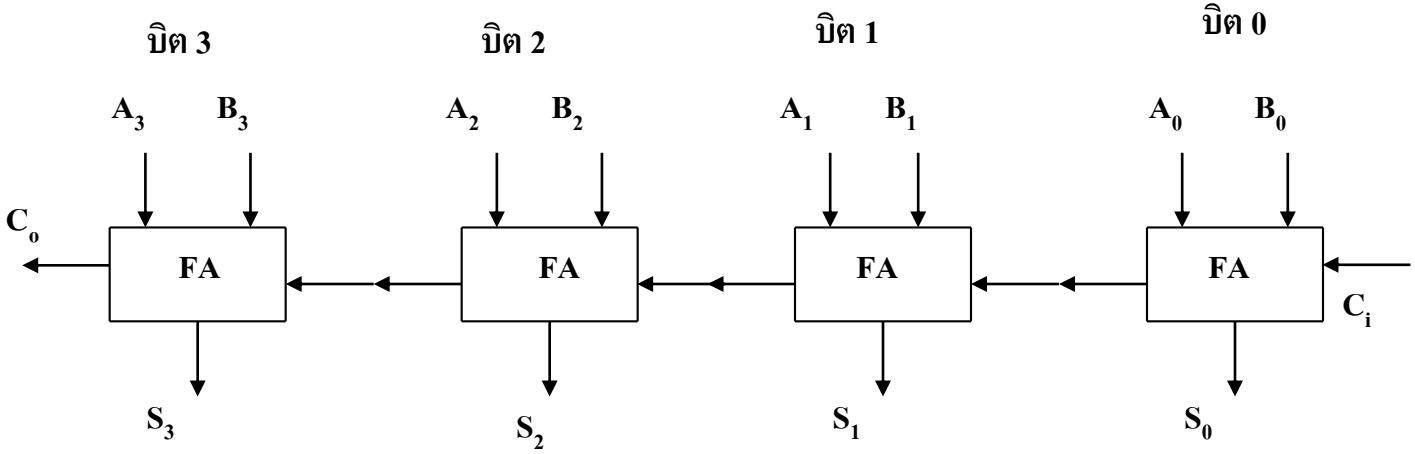
- การบวกในความเป็นจริงแล้วมักจะมีการบวกมากกว่า 1 บิตหรือ 1 หลัก

- ถ้าจะมีการบวกในหลักที่ 2 หรือบิต ที่ 2 จะต้องคำนึงถึง “ผลจากการบวกกันของหลักต่ำกว่าที่อยู่ติดกัน” หมายความว่าเราต้องดูว่าในการบวกของหลักที่ 1 นั้นมีการส่งค่าตัวทด(Carry) ขึ้นมาให้หรือไม่ซึ่งถ้ามีก็ต้องนำมาบวกเข้าไปด้วย ดังนั้นจึงได้เกิดวงจรอีกวงจรหนึ่งที่เรียกว่า

- วงจรบวกแบบคิดตัวทดเข้าจากหลักต่ำกว่า (Full Adder หรือFA)

# วงจรบวกแบบขนาน หรือ Binary Add

วงจรบวกแบบขนานหมายความว่าข้อมูลที่จะนำเข้ามาบวกกัน (ตัวตั้ง, ตัวบวก) เข้ามาแบบขนานกัน เช่นมีข้อมูลที่ต้องการนำมาบวกกันอยู่ 2 ชุด คือ 1011 (ให้เป็นตัวตั้ง) บวกกับ 1000 (ให้เป็นตัวบวก) วงจรบวกที่ใช้บวกก็จะเป็นลักษณะดังนี้

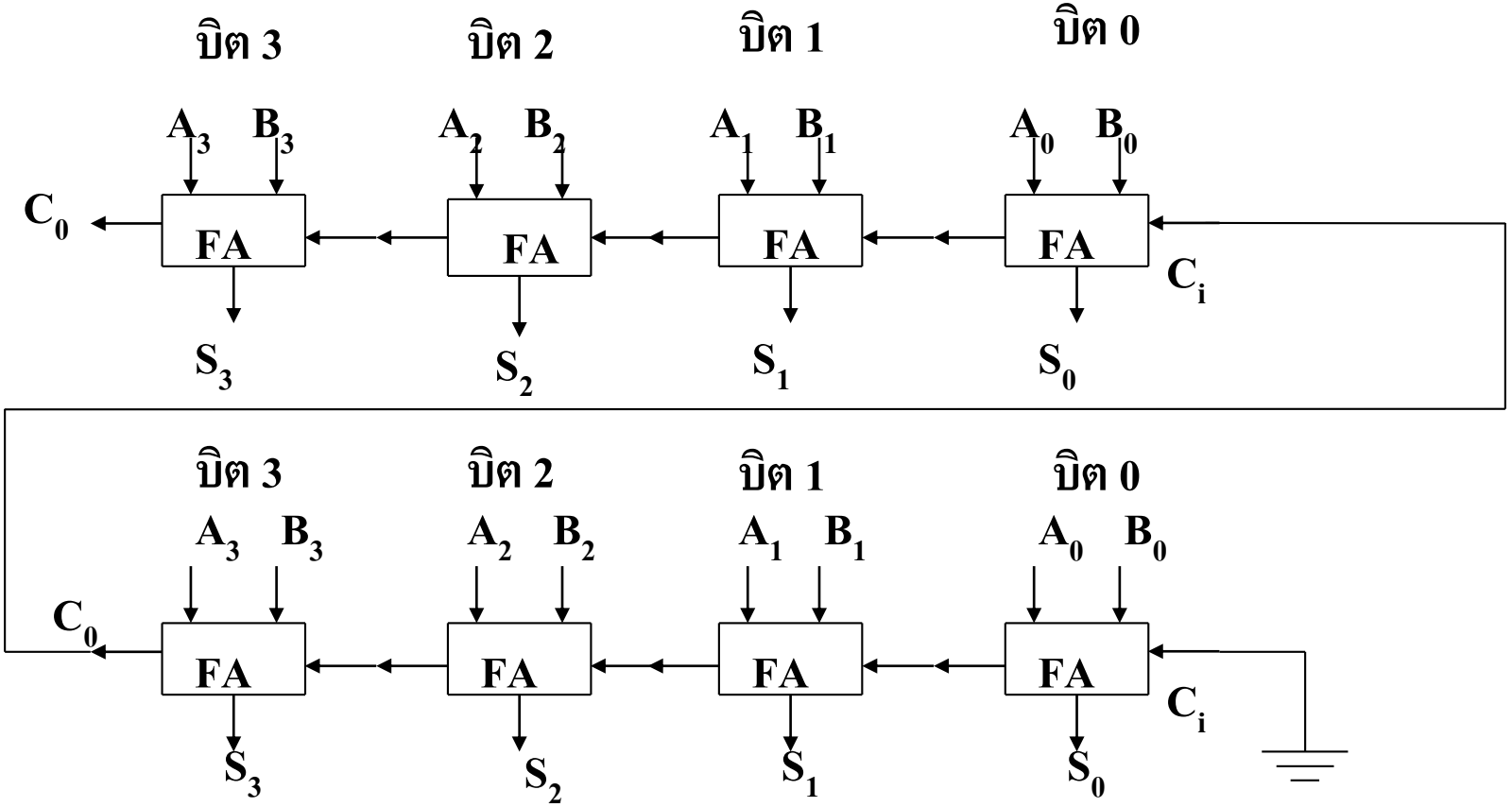



**ไดอะแกรมของวงจรบวกเลขขนาด 4 บิต 2 จำนวน**

# อธิบาย

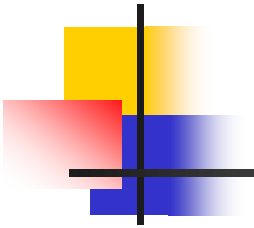
นั่นคือนำวงจรบวกแบบ FA มาต่อๆกัน วงจรบวกเลขขนาด 4 บิต ก็นำ FA มาต่อกัน 4 ชุด จะเห็นว่าทางขวามือสุดนั้นจะเป็น FA ที่มีหน้าที่บวกสำหรับบิตต่ำสุด คือ $A_0$ บวกกับ $B_0$ บวกแล้วได้ผลลัพธ์ออกมาเป็น $S_0$ ถ้ามีตัวทด $C_0$ ก็ทดออกไปเป็น $C_i$ ของหลักสูงกว่าส่วน $C_i$ ของบิต 0 นั้นก็ให้ต่อลงกราวด์ (ลอจิก) ซึ่งจะหมายถึง ไม่มีการทดเข้ามาจากหลักที่ต่ำกว่าบิต**0**

# High - order adder




# Low-order Adder

# 553 วงจรบวกโดยวิธีการมองตัวทดล่วงหน้า (Carry Look Ahead)



ในระบบการทำงานจริงๆ แล้วนั้นวงจรบวกจะถูกสร้างออกมาในลักษณะสำเร็จรูป กล่าวคือจะเป็นวงจรบวกขนาด 4 บิต หรือ 8 บิตไปเลย ในที่นี้เราจะลองใช้กรณีวงจรบวกขนาด 4 บิต มาเป็นตัวอย่างในการอธิบาย

จากวงจรบวกขนาด 4บิต เราจะพบว่าหากเป็นการบวกกันของ “1111” กับ “0001” คำตอบที่ได้ก็คือ “0000” แล้วก็ทดออกไป “1” นั้นคือมี Carry Out ออกไปนั่นเอง Carry Out นี้จะมีประโยชน์ก็คือจะเป็นตัวบอกให้เราทราบว่าผลบวกนั้นเกินค่าสูงสุดที่แสดงได้ (ใน CPU จะเรียกค่านี้ว่า Carry Flag)ซึ่งตัว Carry Flag นี้นั้น ถ้าเป็นจากตัวอย่างจะเห็นว่าเราก็จะรอให้บิต 0 บวกกัน (1+1) แล้วเกิดตัวทดขึ้นมาบิต1 บวกกัน (1+0) บวกกับตัวทด (1+1)ทด 1 ไปข้างหน้า บิต2 บวกกัน(1+0) บวกกับ 1 ที่ทดขึ้นมา (1+1) ได้ 0 ทด 1 ขึ้นไป บิต 3 บวกกัน (1+0)บวกกับ 1 ที่ทดขึ้นมา (1+1) ได้ 0 ทด1 ออกมาเป็น Carry Flag ซึ่งจะเห็นว่าเราต้องรอให้การบวกเทอมนี้สิ้นสุดก่อนจึงจะทราบว่า Carry Flag นี้เป็น 0 หรือ1

ดังนั้น Carry Out ของ บิต 0 หรือ $C_{00}$ จึงเกิดขึ้นได้จากกรณีใดกรณีหนึ่งต่อไปนี้

1. $A_0$ และ $B_0$ เป็น “1” ได้ว่า $C_{oo} = (A_0B_0)$ (เรียกว่า Carry Generate :G)
2. $A_0$ หรือ $B_0$ เป็น “1” และ $C_i$ เป็น “1” ได้ว่า$C_{00}=(A_0+B_0)C_{i0}$

(เรียกว่าcarry propagate :p)

**สรุปเป็นว่า $C_{00}=(A_0B_0)+(A_0+B_0)C_{i0}$** .....1

ทีนี้ลองมาดูของบิต 1 ดูบ้างซึ่งก็เช่นเดียวกันโอกาสเกิด $C_{01}$ นั้นมีได้ 2 กรณีเช่นกัน คือ

$A_1$และ $B_1$ เป็น “1” ได้ว่า $C_{01} = (A_1B_1)$

$A_1$ หรือ $B_1$ เป็น “1” และ $C_{i1}$ เป็น “1” ได้ว่า $C_{01}=(A_1+B_1)C_{i1}$

**สรุปได้ว่า $C_{01}= (A_1B_1)+(A_1+B_1)C_{i1}$** .....2

แต่ที่นี้ถ้าเราสังเกตที่ $C_{i1}$ ในสมการของบิต1 นี้นั้นซึ่งเป็นค่าที่ทดเข้ามาของบิตนี้ก็มาจากค่าที่ทดออก (Carry Out) ของบิต 0นั่นเอง ดังนั้นก็จะเป็นว่า

$C_{i1}= C_{00}= (A_0B_0) + (A_0+B_0) C_{i0}$ (สมการ1) ซึ่งทำให้เขียนสมการที่ 2 ได้ใหม่เป็นดังนี้

**$C_{01}=(A_1B_1)+[(A_1+B_1)((A_0B_0)+(A_0+B_0)C_{i0})]$** .....3

จะเห็นได้ว่าตัวทดออกหรือ $C_0$ ของหลักที่สูงขึ้นไปก็จะได้สมการที่มีตัวแปรเข้ามาเกี่ยวข้องมากขึ้นเรื่อย ๆ สมการก็จะมีขนาดใหญ่ขึ้น ๆ ทำให้ยากในการทำความเข้าใจ ซึ่งได้มีการใช้ตัวแปรเพื่อแทนเทอมของสมการกล่าวคือ ถ้าย้อนกลับไปที่สมการที่ (1) จะเห็นว่าเราแบ่ง $C_{00}$ ออกเป็น 2 เทอมคือ Carry Generate ซึ่งต่อไปจะแทนด้วย G กับเทอมของ Carry Propagate ซึ่งต่อไปจะแทนด้วย P ดังนั้น ถ้าหากเราแทน เทอมทั้งสองด้วยตัวแปร G และ P ก็จะได้เป็น

$$C_{00} = (A_0B_0)+(A_0+B_0)C_{i0} \quad \text{.....4}$$

$$= (G_0) + (P_0)C_{i0}$$

**ดังนั้นในหลักต่อๆ ไปก็จะได้เป็น**

$$C_{01} = (A_1B_1)+(A_1+B_1)C_{i1}$$

$$= (G_1)+(P_1)C_{i1} \quad \text{.....5}$$

$$C_{02} = G_2+P_2C_{i2} \quad \text{.....6}$$

$$C_{03} = G_3+P_3C_{i3} \quad \text{.....7}$$

ในสมการที่(5) นั้น สังเกตที่$C_{i1}$ ก็จะเหมือนกับเมื่อครั้งที่เป็นสมการที่ (3) กล่าวคือ$C_{i1}$ นี้นั่นก็คือ$C_{00}$ นั่นเอง ดังนั้นสมการ(5) นี้ก็จะเขียนใหม่โดยการแทนค่า$C_{i1}$ ด้วย$C_{00}$ (สมการที่4)ได้ดังนี้

$$C_{01} = G_1+P_1C_{i1}$$

$$C_{01} = G_2+P_1(G_0+P_0C_{i0}) \quad \text{....8}$$

สมการ (6) จะเขียนใหม่โดยการแทนค่า $C_{i2}$ ด้วย$C_{01}$ (สมการที่8) ได้เป็น

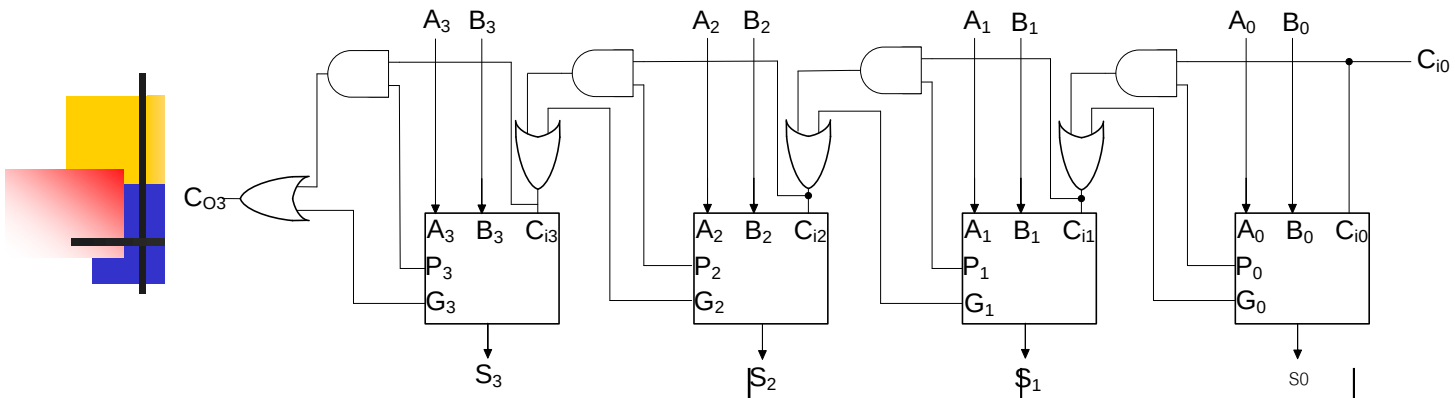
$$C_{02} = G_2+P_2C_{i2}$$

$$C_{02} = G_2+P_2[G_1+P_1(G_0+P_0C_{i0})] \quad \text{....9}$$

สมการ(7) จะเขียนใหม่โดยการแทนค่า$C_{i3}$ด้วย$C_{02}$(สมการที่9)ได้เป็น

$$C_{03} = G_3+P_3C_{i3}$$

$$C_{03} = G_3+P_3\{G_2+P_2[G_1+P_1(G_0+P_0C_{i0})]\} \quad \text{...10}$$



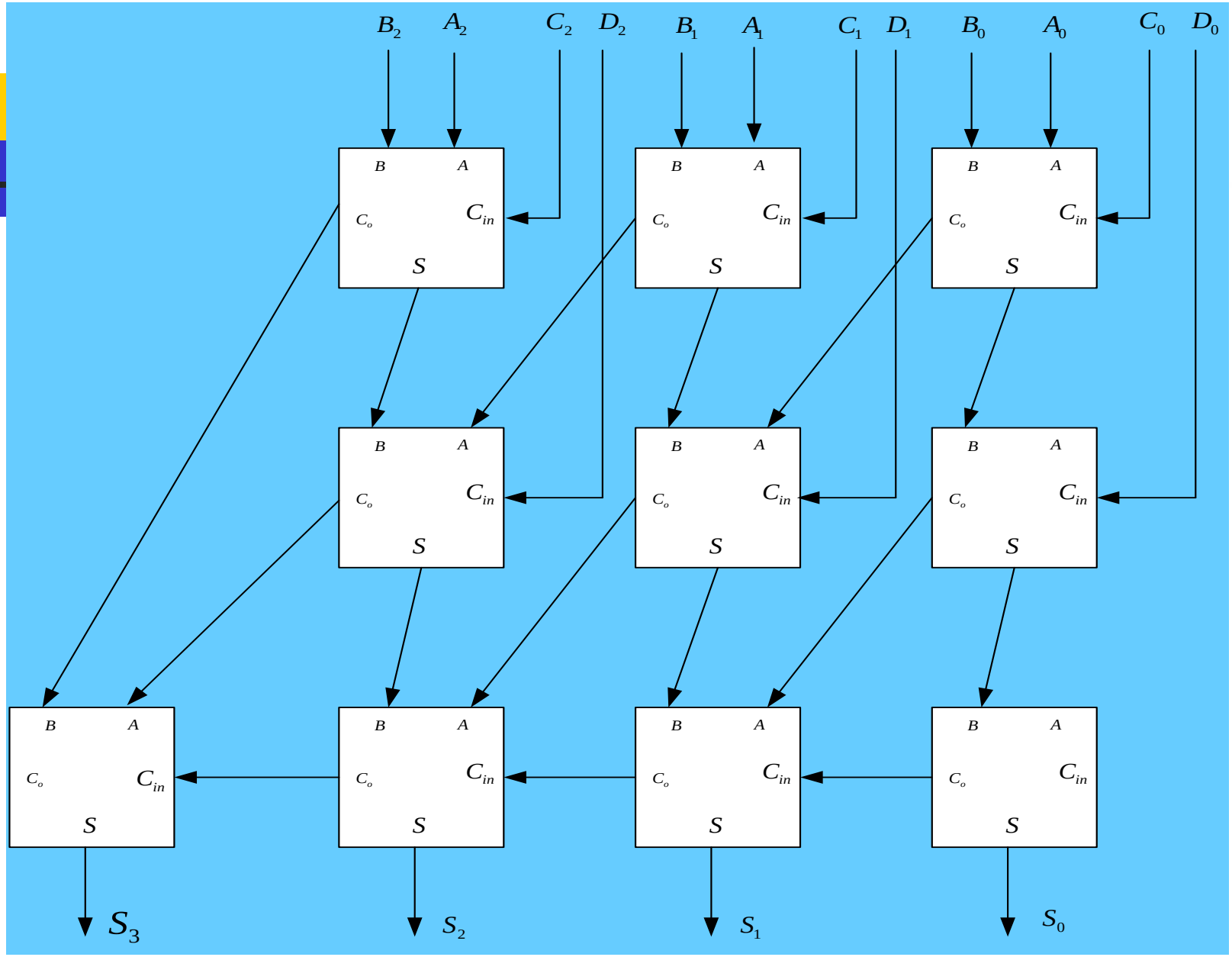

$$C_{03} = G_3+P_3( G_2+P_2[G_1+P_1(G_0+P_0C_{i0})])$$

$$C_{02} = G_2+P_2[G_1+P_1(G_0+P_0C_{i0})]$$

$$C_{01} = G_1+P_1(G_0+P_0C_{i0})$$

$$C_{00} = G_0+P_0C_{i0}$$

# 554 หลักการบวก4 จำนวนแบบ Carry Save



# อธิบาย

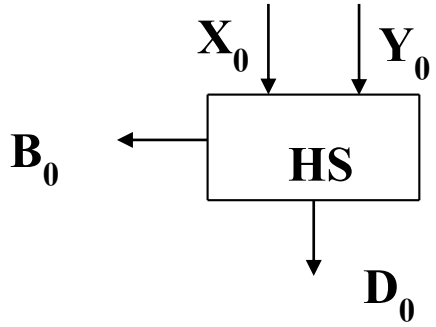
- การบวกที่ไม่จำกัดอยู่ที่ 2 จำนวน
- สามารถทำการบวกได้หลายจำนวนพร้อมกัน
- นำผลของตัวทดไปบวกกับหลักถัดไปได้
- หลักการนี้สามารถนำไปใช้กับการบวกเพิ่มตัวตั้ง n จำนวนได้
- บิตสุดท้าย $M_{SB}$ สามารถนำผลของ Carry Out มาบวกกัน

# 56 หลักการวงจรลบ

## วงจรลบเลขฐานสองแบบไม่คิดตัวยืมจากหลักต่ำกว่า

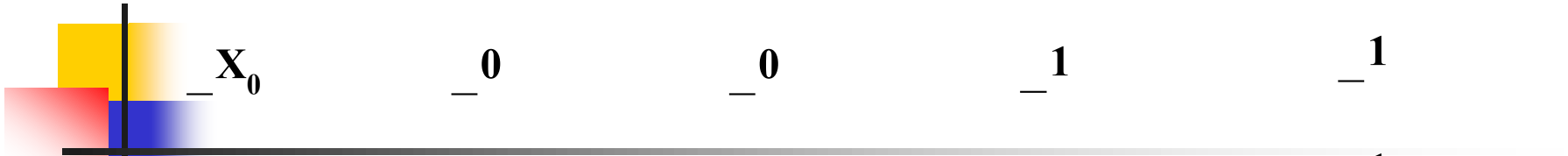
วงจรลบเลขแบบไม่คิดตัวยืม (Half Subtracter หรือ H.S.) เป็นวงจรลบเลขฐานสองขนาด 1 บิต แสดงเป็นบล็อกไดอะแกรมได้ดังนี้

$$\begin{array}{r} - \quad X_0 \\ \quad Y_0 \\ \hline B_0 \; D_0 \end{array}$$




### บล็อกไดอะแกรมของวงจรลบแบบไม่คิดตัวทด

จากรูปจะเห็นว่ามีอินพุตอยู่ 2 ตัวคือ ตัวตั้ง $x_0$ กับตัวลบ $Y_0$ และมีเอาต์พุต2 ตัวเช่นเดียวกัน คือผลต่าง$D_0$(Difference)และตัวยืม$B_0$(Borrow) ซึ่งตัวยืมนี้จะเกิดขึ้นเมื่อตัวตั้งมีค่าน้อยกว่าตัวลบ



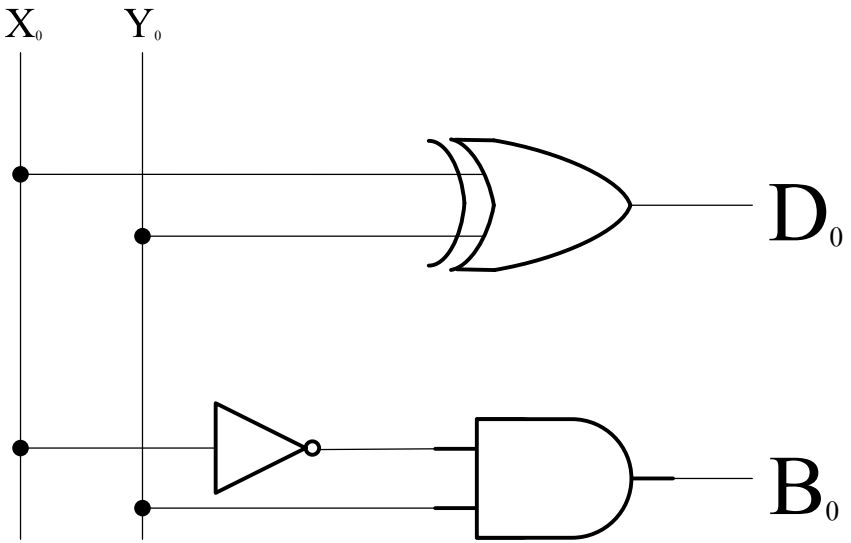
| | | | | |
|---|---|---|---|---|
| $X_0$ | 0 | 0 | 1 | 1 |
| $Y_0$ | 0 | 1 | 0 | 1 |
| $B_0 D_0$ | 0 0 | 1 1 | 0 1 | 0 0 |

| $X_0$ | $Y_0$ | $D_0$ | $B_0$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

กรณีที่น่าสนใจคือกรณีที่ 0-1 กรณีนี้ จะลบไม่ได้ 0 ก็เลยไปยืม เมื่อต้องไปยืม เมื่อต้องไปยืมB0จะเป็น 1 (Borrow Set) ทันทีเมื่อไปยืมมาก็จะได้เป็น 10 ลบด้วย 1 ก็เหลือ 1

จากตารางนำมาสรุปเป็นฟังก์ชันของผลต่าง (หรือผลลบ) และตัวยืมได้ดังนี้คือผลต่าง = $D_0 = X_0Y_0 + X_0Y_0$ หรือ $X_0 + Y_0$

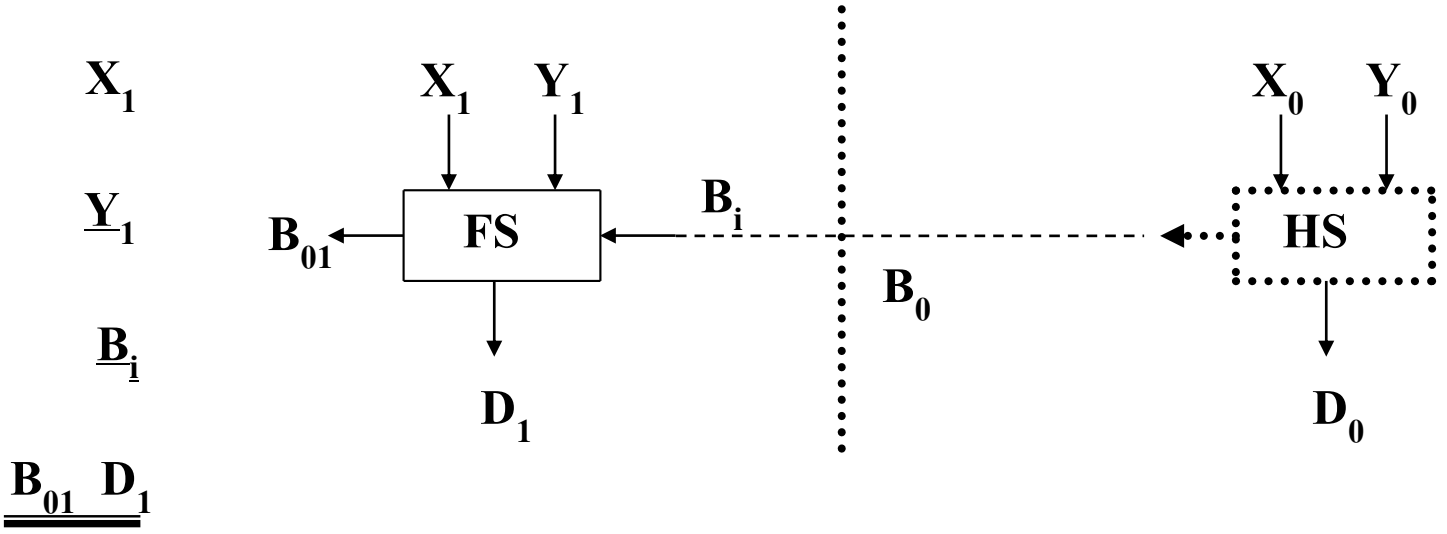
ตัวยืม = $B_0 + X_0Y_0$ เขียนเป็นวงจรได้ดังนี้



# 561 การออกแบบวงจรลบแบบไม่คิดตัวทดเข้า (Half Subtracter)

## วงจรลบเลขแบบคิดตัวยืม

วงจรเลขแบบคิดตัวยืม (Full Subtracter หรือ FS)คือ วงจรเลขฐาน2ขนาด1 บิต ที่ประกอบด้วยตัวตั้ง$X_1$ตัวลบ $Y_1$และตัวยืมจากหลักที่ต่ำกว่า$B_1$ (Borrow In)อีก 1ตัว แสดงได้ด้วยบล็อกไดอะแกรมดังนี้




**บล็อกไดอะแกรมของวงจรลบเลขแบบคิดตัวยืม**

# 562 การออกแบบวงจรลบแบบคิดตัวทดเข้า (Full Subtracter)

## ตารางความจริงการลบเลขโดยคิดตัวทดเข้า (Full Subtracter)

| Input | | | Output | |
|---|---|---|---|---|
| A | B | $B_i$ | Diff | $B_o$ |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

ตัวยืม Input = $B_i$

ตัวยืม Output = $B_o$

$0 - 0 - 0 = 0$ ยืม 0

$0 - 0 - 1 = 1$ ยืม 1

$0 - 1 - 0 = 1$ ยืม 1

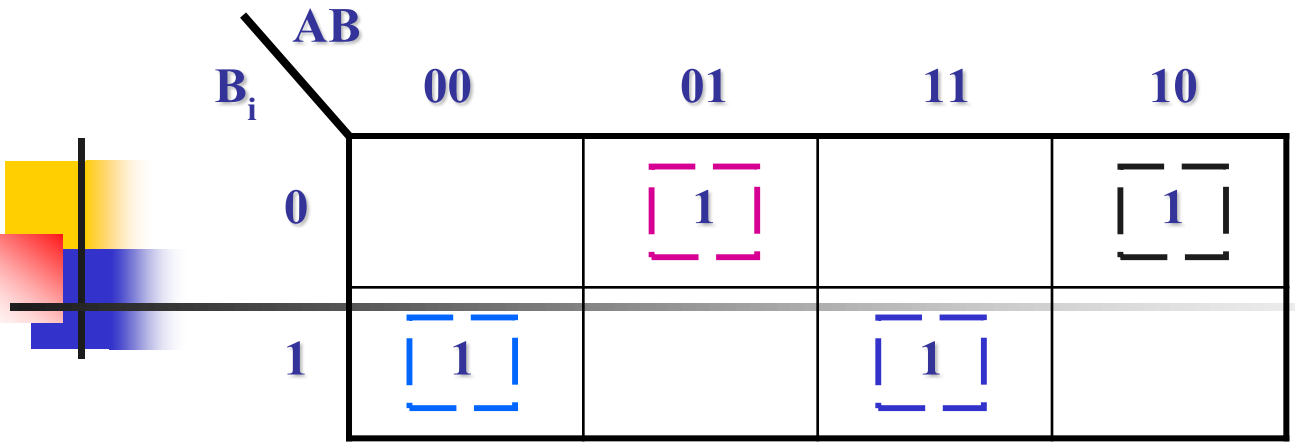
$0 - 1 - 1 = 0$ ยืม 1

$1 - 0 - 0 = 1$ ยืม 0

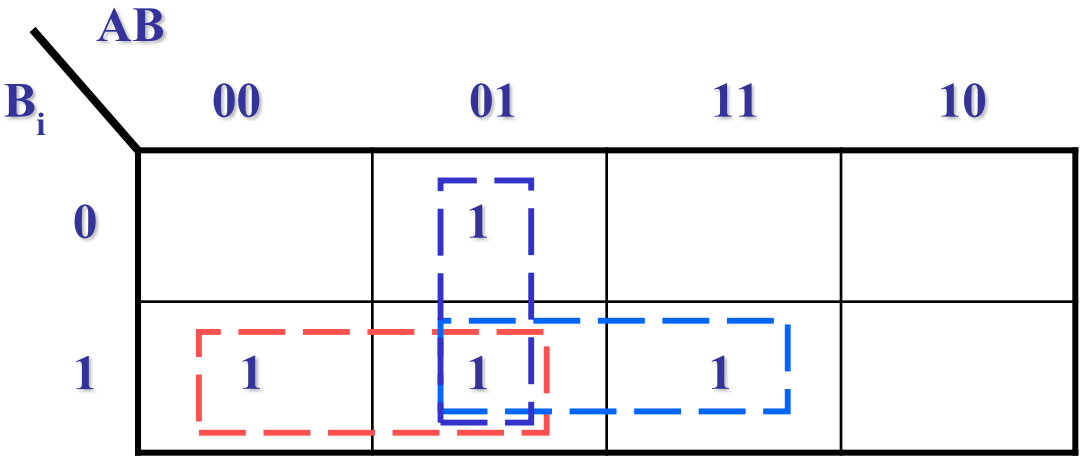
$1 - 0 - 1 = 0$ ยืม 0

$1 - 1 - 0 = 0$ ยืม 0

$1 - 1 - 1 = 1$ ยืม 1




$$\text{Difference} = \overline{A}\,\overline{B}\,B_i + \overline{A}\,B\,\overline{B_i} + A\,B\,B_i + A\,\overline{B}\,\overline{B_i}$$




$$\text{Borrow Out} = \overline{A}\,B_i + \overline{A}\,B + B\,B_i$$

# อธิบาย

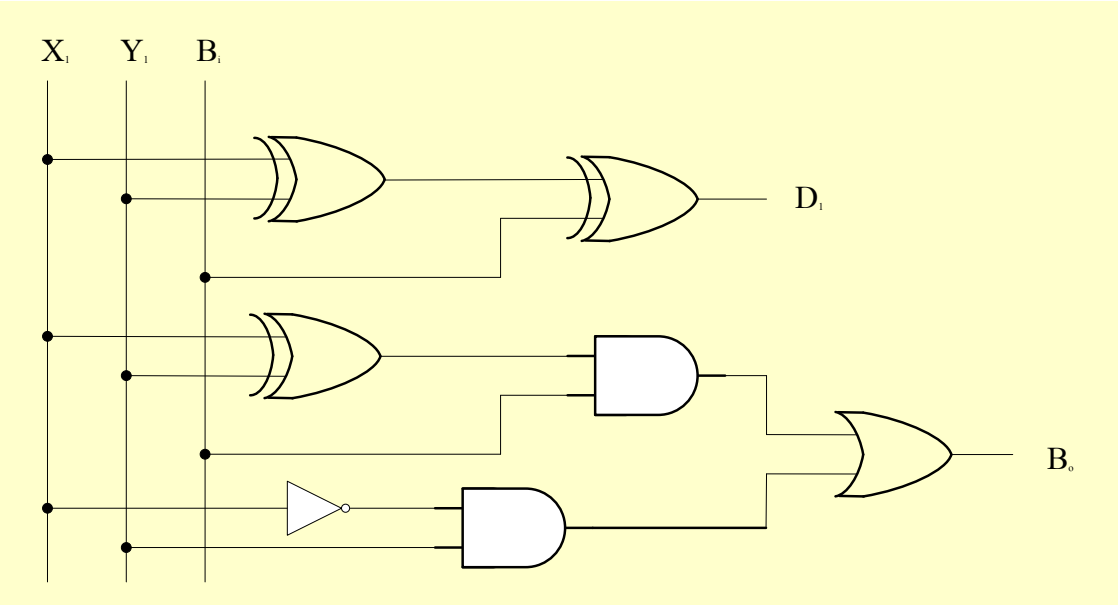
การพิจารณาก็คล้ายๆ กับกรณีของวงจรแบบ HS เพียงแต่ในวงจรแบบ FS นี้ได้เพิ่ม $B_i$ เข้ามาร่วมในการการลบอีกตัวเท่านั้นเองกล่าว คือตัวตั้งก็ยังมีตัวเดียวเหมือนเดิม แต่ตัวบวกจะเพิ่มขึ้นมาเป็น 2 ตัว ดังนั้นเมื่อ ตัวตั้ง $(X_1)$ ลบด้วยตัวลบ $(Y_1)$ แล้วผลเป็นเช่นไร ก็นำมาให้$B_i$ ลบอีกทีหนึ่ง

# จากตารางทำให้เขียนเป็นฟังก์ชันทางพีชคณิต

ผลต่าง = $D_1 = X_1Y_1B_i+X_1Y_1B_i+X_1Y_1B_i+X_1Y_1B_i$

ตัวยืม = $B_{01} = X_1Y_1B_i+X_1Y_1B_i+X_1Y_1B_i+X_1Y_1B_i$

$= B_i(X_1+Y_1) + X_1Y_1$

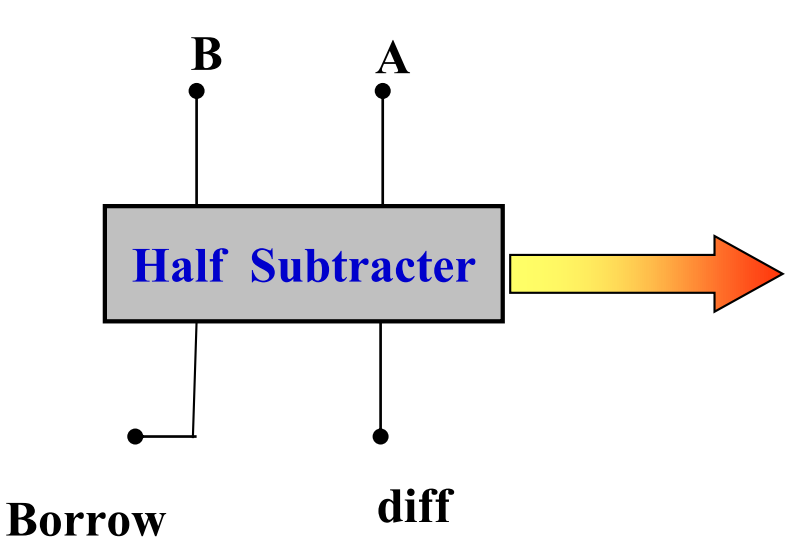


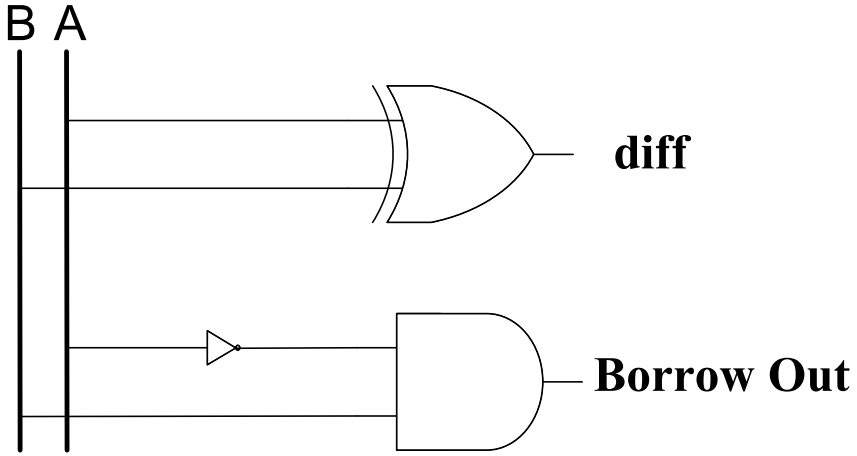

**วงจรลบเลขแบบคิดตัวยืม(Full Subtracter หรือ FS)**

# วงจร Half Subtraction

## การลบเลขฐานสองแบบ 1 บิต มีขั้นตอนดังนี้

| | | |
|---|---|---|
| ตัวตั้ง ( อินพุตA ) | 1 | 0 |
| ตัวลบ ( อินพุตB ) | 0 | 1 |
| ผลต่าง ( Difference ) | 1 | 1 |
| ตัวยืม ( Borrow Out ) | 0 | 1 |



# ตารางความจริง

| B | A | Diff | Borrow Out |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

$0 - 1 = 1$ ยืม 1

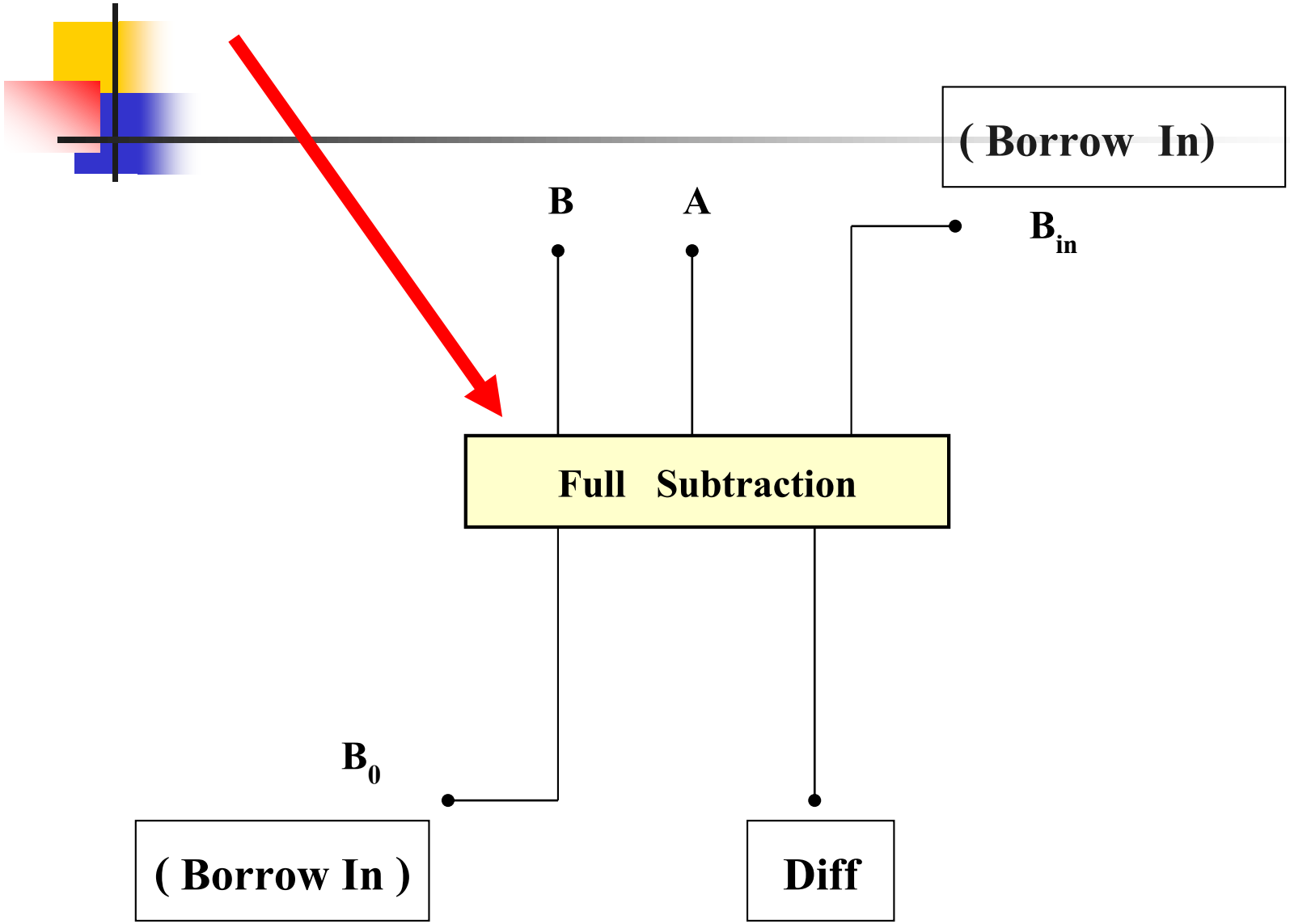
$\text{Diff} = A\overline{B} + \overline{A}B = A \oplus B$

Borrow Out $= \bar{A}B$

# อธิบาย

- การลบเลขขนาด 1 บิตเรียกว่าวงจร Half Subtrater มีตัวตั้งเป็น A ตัวลบเป็น B ผลของเอาต์พุตที่ได้คือ ผลต่าง (Diff ) กับการยืมหลักหน้ามาเป็น (Borrow Out)
- วงจรนี้ประกอบด้วย EX-OR Gate กับ AND Gate เป็นวงจรหลัก
- นำผลของอินพุตกับ เอาต์พุตมาสร้างตารางความจริง
- พิจารณาที่ผลลัพธ์หรือส่วนที่แตกต่าง (Diff) นำไปเขียนในรูปแบบ Minterm คือ ก็จะได้เป็นฟังก์ชันของการลบ $A\overline{B} + \overline{A}B$
- ส่วนค่าที่ทำการยืมจากภายนอกโดยพิจารณาจากตัวตั้งมีค่าน้อยกว่าตัวลบหรือค่า $A < B$ Borrow Out จะมีค่าเท่ากับ 1
- เขียนฟังก์ชันจากตาราง Minterm ได้เป็น $\overline{A}B$

# วงจร Full Subtraction



# ตารางความจริงของการลบเลขแบบมีคิดตัวเลขยืมเข้า

| $B_{in}$ | B | A | Diff | $B_O$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$B_0 = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + \bar{A}BB_{in} + ABB_{in}$$
$$= \bar{A}\bar{B}B_{in} + B_{in}(\bar{A}\bar{B} + \bar{A}B + AB)$$
$$= \bar{A}\bar{B}B_{in} + B_{in}(\bar{A} + B)$$

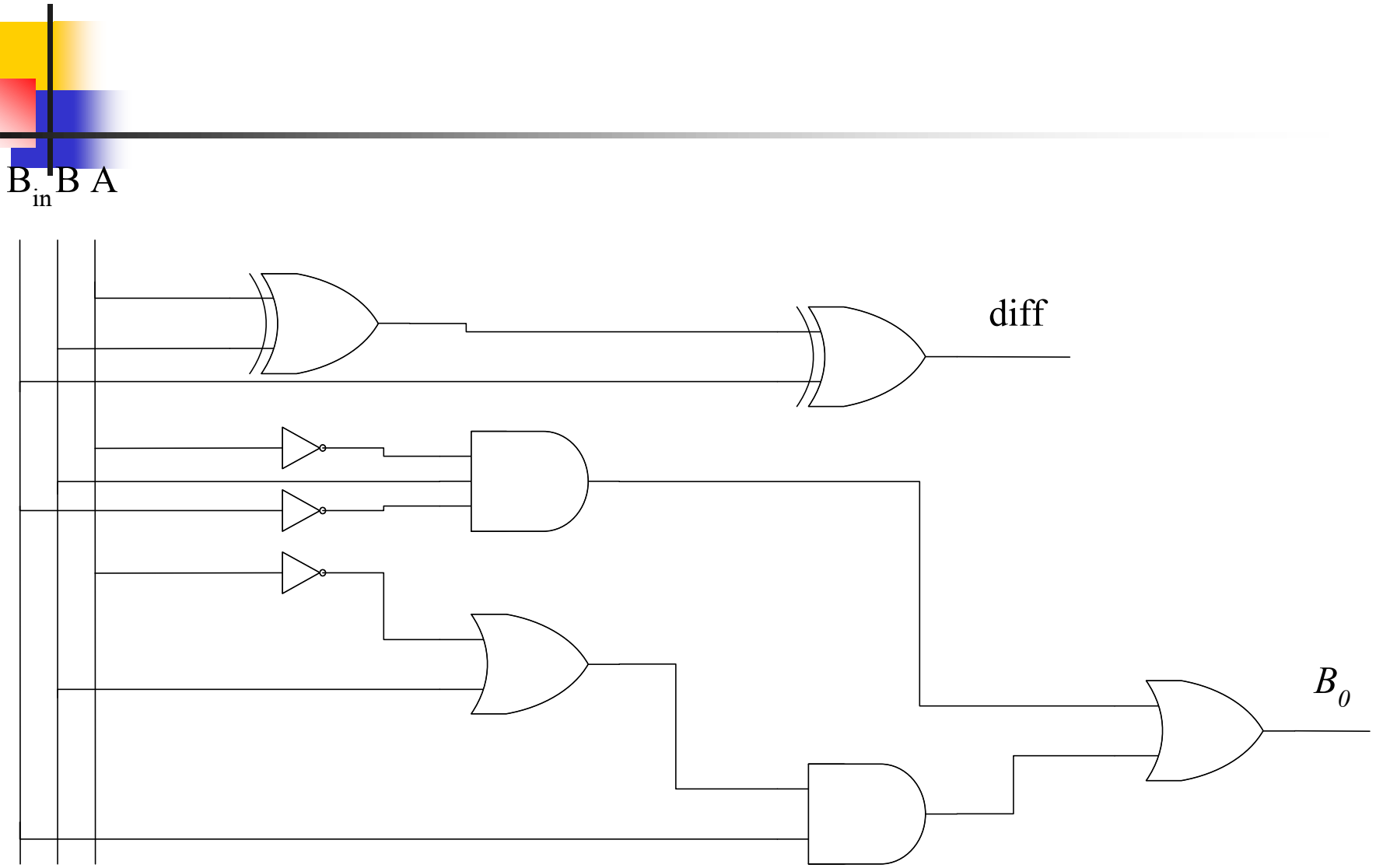
# ตารางความจริงของวงจร Full Subtracter

Diff

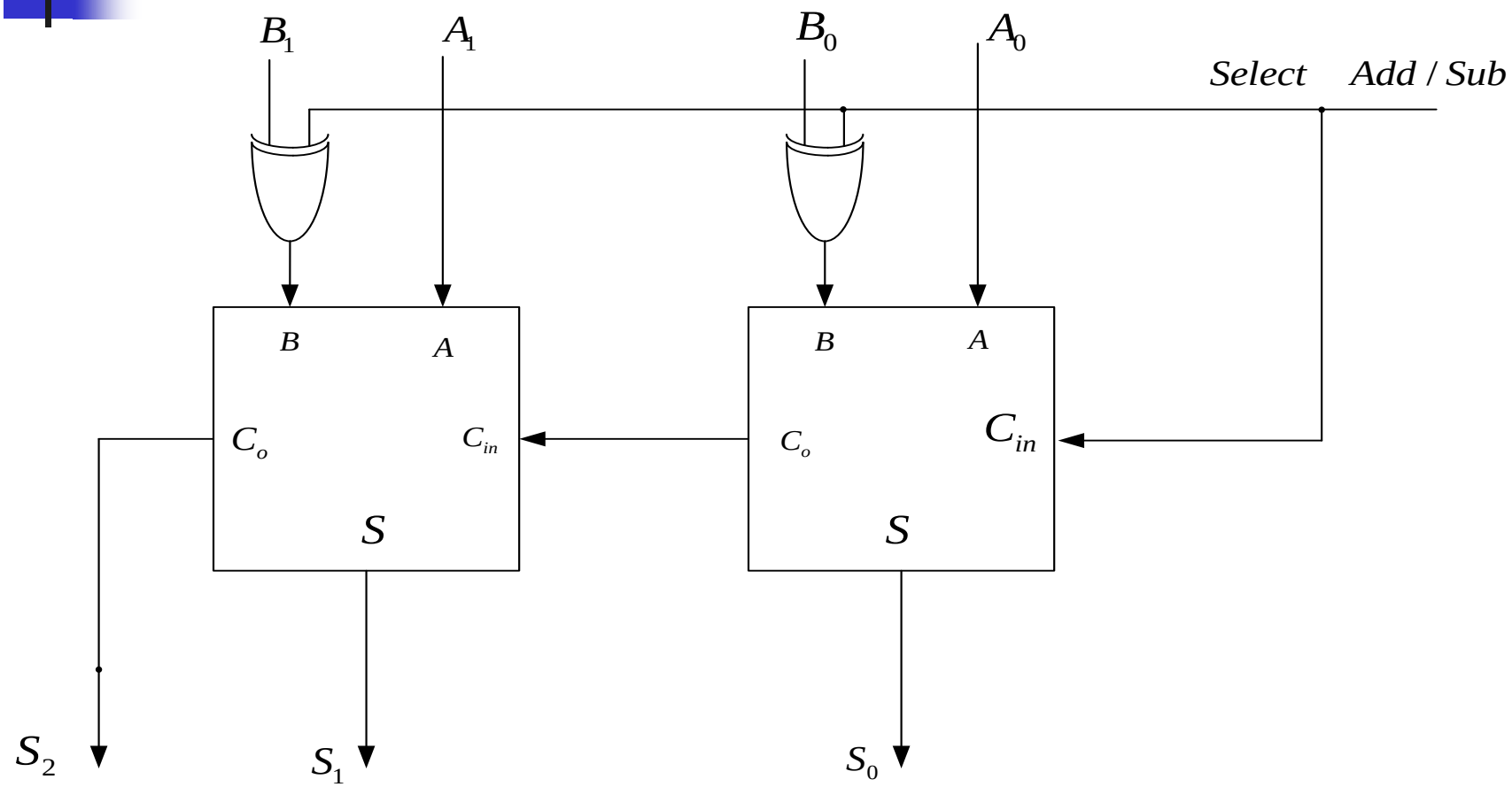
$$= A\bar{B}\bar{B}_{in} + \bar{A}B\bar{B}_{in} + \bar{A}\bar{B}B_{in} + ABB_{in}$$
$$= \bar{B}_{in}(A\bar{B} + \bar{A}B) + B_{in}(\bar{A}\bar{B} + AB)$$
$$= \bar{B}_{in}(A \oplus B) + B_{in}(A + B)$$
$$= B_{in} \oplus (A \oplus B)$$

| $B_{in}$ | B | A | Diff |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# วงจร Full Subtracter



# 563 การออกแบบวงจรบวก/ลบเลขขนาด 2 บิตแบบใช้ 2'Complement



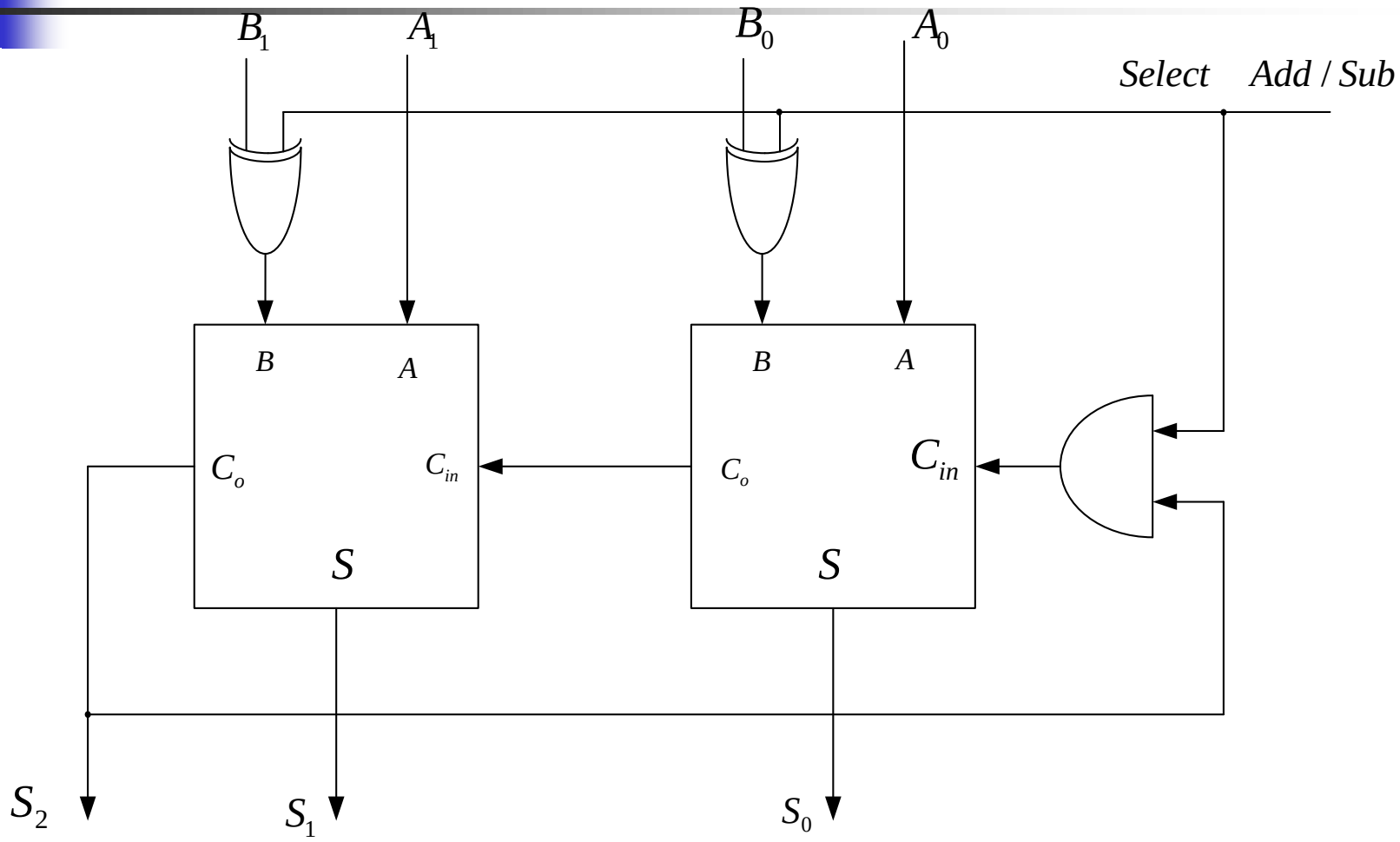
# การบวกและลบโดยใช้วิธี2’Complement

- การลบด้วยวิธี 2’Complement
- โดยใช้หลักการที่กล่าวมาในสัปดาห์ที่ 2
- วงจรที่ออกแบบเป็นวงจรใช้ Full Adder ขนาด 2 บิต
- ตัวตั้งเป็น $A_0$ -$A_1$ ตัวลบเป็นค่า $B_0$- $B_1$
- โดยมีบิตเครื่องหมายกำหนดค่าตัวกระทำ (Selector Bit)

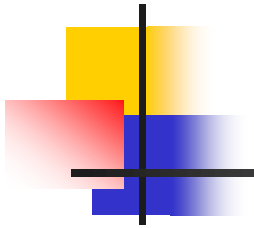
# การบวกและลบโดยใช้วิธี2’Complement (ต่อ)

- ถ้าอินพุตเป็นบวกทั้งตัวตั้งและตัวลบนั่นหมายความว่า (A+B) ตัวเลือกเป็น 0
  - ค่าA เป็นดังอินพุตเข้ามา $A_0$-$A_1$
  - ค่าB กระทำโดยEX-OR Gate ที่ตัวเลือก= 0 $B_0$-$B_1$ =B
- ถ้าอินพุตเป็นบวกที่ตัวตั้งและตัวลบเป็นลบนั่นหมายความว่า(A-B) ตัวเลือกเป็น 1 ค่า
  - ค่าA เป็นดังอินพุตเข้ามา $A_0$-$A_1$
  - ค่าB กระทำโดย EX-OR Gateที่ Sel.= 1 $B_0$-$B_1$เป็น 1’Complement+1 เป็น 2’B
- เมื่อมีโอเวอร์โฟล์จะไม่นำมาบวก
- ค่า A และ B ก็จะกระทำด้วยวงจร Full Adder ได้ผลลัพธ์ เป็น $S_0$-$S_2$

# การลบเลขขนาด 2 บิตแบบใช้ 1'Complement



# การบวกและลบโดยใช้วิธี2'Complement

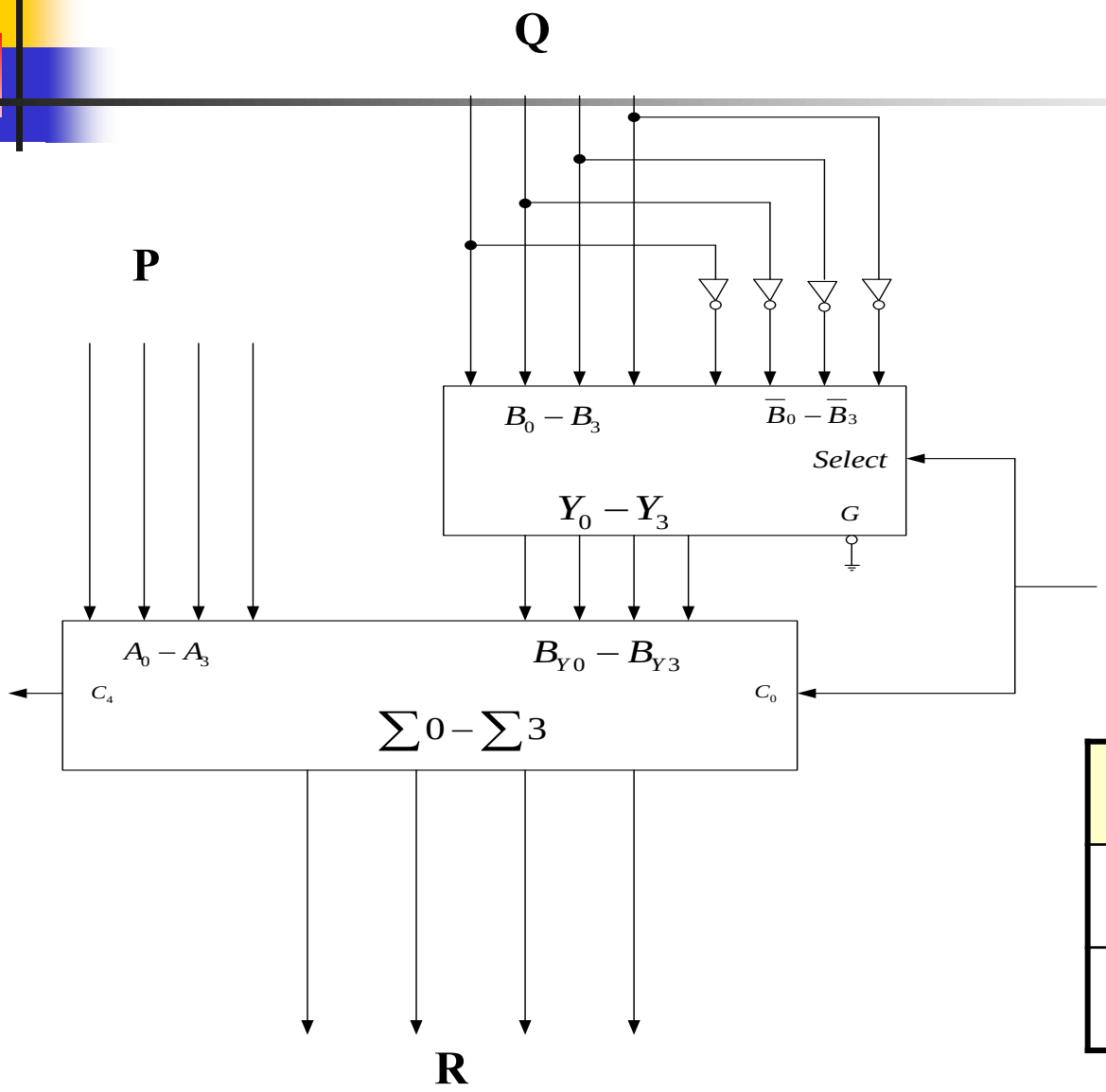


- การลบด้วยวิธี 1'Complement
- โดยใช้หลักการที่กล่าวมาในสัปดาห์ที่ 2
- วงจรที่ออกแบบเป็นวงจรใช้ Full Adder ขนาด 2 บิต
- ตัวตั้งเป็น $A_0$-$A_1$ ตัวลบเป็นค่า $B_0$- $B_1$
- โดยมีบิตเครื่องหมายกำหนดค่าตัวกระทำ (Selector Bit)

# การบวกและลบโดยใช้วิธี2'Complement (ต่อ)

- ถ้าอินพุตเป็นบวกทั้งตัวตั้งและตัวลบนั่นหมายความว่า (A+B) ตัวเลือกเป็น 0
  - ค่าที่ A เป็นดังอินพุตเข้ามา $A_0$-$A_1$
  - ค่าที่ B กระทำโดย EX-OR Gate ที่ตัวเลือก= 0 $B_0$-$B_1$ =B
- ถ้าอินพุตเป็นบวกที่ตัวตั้งและตัวลบเป็นลบนั่นหมายความว่า (A-B) ตัวเลือกเป็น 1 ค่า
  - ค่า A เป็นดังอินพุตเข้ามา $A_0$-$A_1$
  - ค่า B กระทำโดย EX-OR Gate ที่ตัวเลือก = 1 $B_0$-$B_1$ เป็น 1'Complement B
- เมื่อมีโอเวอร์โฟล์จะนำมาบวกกับบิตทางขวามือสุดโดยใช้ AND Gate เป็นตัวช่วย
- ค่า A และ B ก็จะกระทำด้วยวงจร Full Adder ได้ผลลัพธ์ เป็น $S_0$-$S_1$

# หลักการลบโดยวิธีใช้ 2'Complement



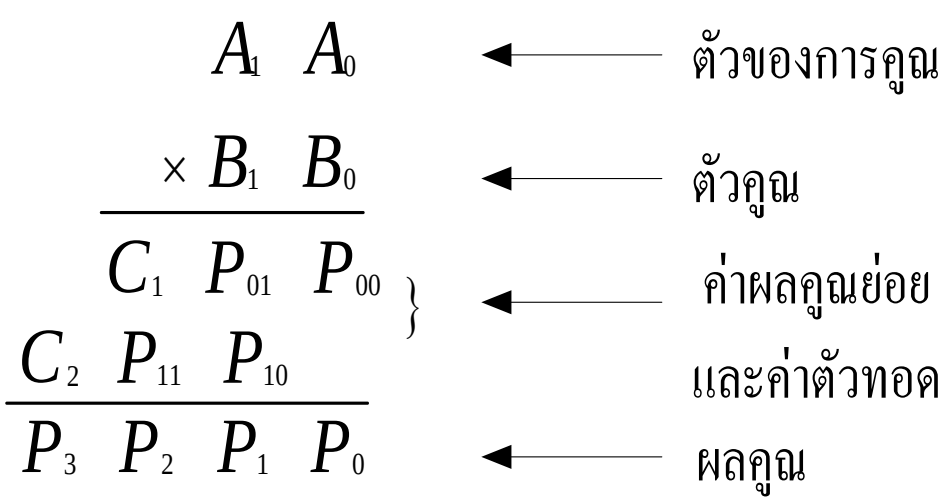

| Select | Function |
|---|---|
| 0 | R=P+Q |
| 1 | R=P+Q+1 |

# อธิบาย

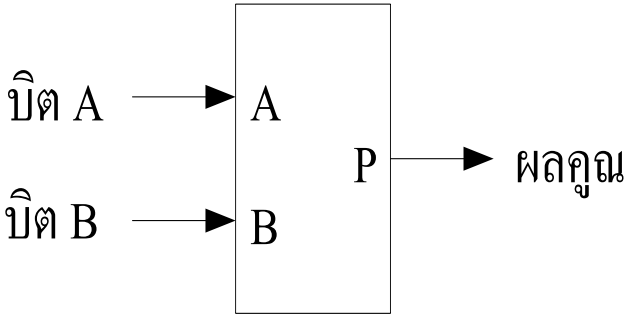
- การสร้างวงจรบวกและลบขนาด n บิต
- ประกอบด้วยวงจร Full Adder เป็นส่วนประกอบหลัก
- วงจรMultiplex ทำหน้าที่เป็นส่วนเลือกการทำงานระหว่างวงจรบวกหรือวงจรลบ
- หลักการของวงจรลบโดยการทำการ2’ Complement
- เลือกฟังก์ชันการทำงานเป็นวงจรบวกให้ค่า S=0 , $C_o$=0 เป็นการบวก$A_0$-$A_3$ กับค่า$B_0$-$B_3$
- เลือกฟังก์ชันกจรทำงานเป็นวงจรลบให้ค่า S=1 , $C_o$=1 เป็นการบวกค่า$A_0$-$A_3$ กับค่า $B_0$-$B_3$ +1 (การทำเป็น 2’ Complement)

# 57 หลักการวงจรคูณ

## หลักการคูณเลขฐานสอง




(ก) ผังการตั้งการคูณ




(ข) บล็อกไดอะแกรมของวงจรคูณ

รูปแบบของการการคูณ และบล็อกไดอะแกรมการคูณ

# อธิบาย

- รูป(ข) แสดงบล็อกไดอะแกรมของการคูณค่าA * B ได้ผลคูณเป็น P
- รูป(ก) เป็นผังของการตัวตั้ง และ ตัวคูณ
- โดยมี$A_1$,$A_0$ เป็นตัวตั้งขนาดสองบิต
- $B_1$, $B_0$ เป็นตัวคูณขนาดสองบิตเท่ากัน
- หลักการคูณที่เคยเรียนผ่านมาได้ค่าผลคูณย่อยได้ดังนี้คือ $P_{00}= B_0A_0$ , $P_{01}= B_0A_1$ , $P_{10}= B_1A_0$ , $P_{11}= B_1A_1$
- ตัวทดย่อยที่เกิดขึ้นคือค่า $C_2$ , $C_1$
- ตำแหน่งของค่าต่างๆ ที่เกิดขึ้นจะเป็นดังรูป
- จากนั้นนำค่าผลคูณย่อยกับตัวทดมาบวกกันได้เป็นผลลัพธ์ ดังนี้คือ $P_0=P_{00}$ , $P_1=P_{01}+ P_{10}$ , $P_2= C_1+P_{11}$ , $P_3= C_2$
- นำหลักการที่กล่าวมาทั้งหมดไปสร้างตารางความจริง

# 571 การออกแบบวงจรคูณ

การคูณเลข Binary มีหลักการเช่นเดียวกันกับการคูณเลขฐานสิบ แต่การคูณเลขBinary กระทำได้ง่ายกว่า เพราะตัวคูณเป็นได้เพียง 0 หรือ 1 เท่านั้น ดังนั้นผลคูณย่อยที่ได้ ถ้าไม่เป็น 0 ก็ต้องเป็นเหมือนตัวตั้ง

**ตัวอย่าง 110101×101**

110101 ×

101

110101

000000

110101

100001001

# ตารางความจริงของการคูณเลขขนาด 2 บิต

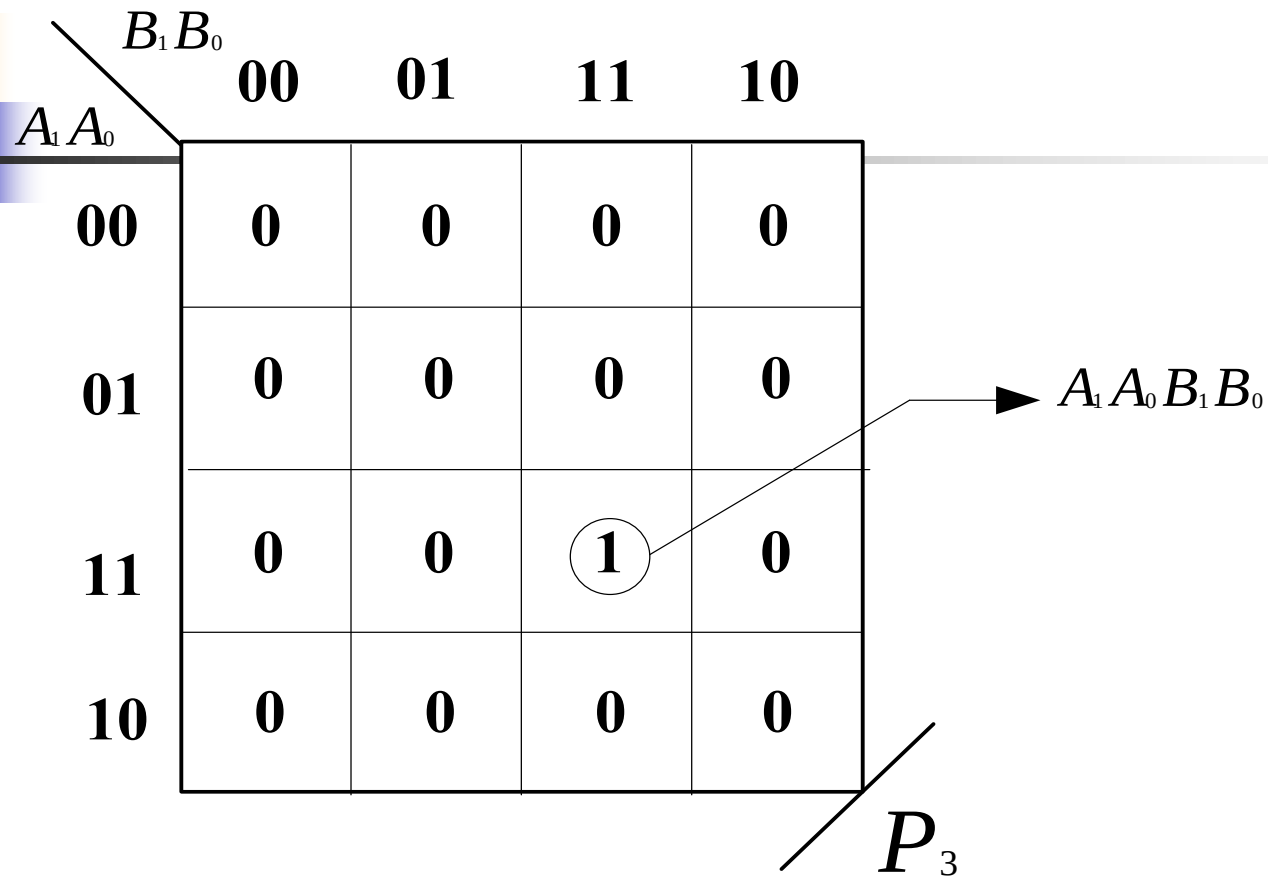
| $B_1 B_0$ | $A_1 A_0$ | $P_3 P_2$ | $P_1 P_0$ | ฐานสิบ |
|---|---|---|---|---|
| 00 | 00 | 00 | 00 | 0 x 0 = 0 |
| 00 | 01 | 00 | 00 | 0 x 1 = 0 |
| 00 | 10 | 00 | 00 | 0 x 2 = 0 |
| 00 | 11 | 00 | 00 | 0 x 3 = 0 |
| 01 | 00 | 00 | 00 | 1 x 0 = 0 |
| 01 | 01 | 00 | 10 | 1 x 1 = 1 |
| 01 | 10 | 00 | 10 | 1 x 2 = 2 |
| 01 | 11 | 00 | 11 | 1 x 3 = 3 |

| $B_1 B_0$ | $A_1 A_0$ | $P_3 P_2$ | $P_1 P_0$ | ฐานสิบ |
|---|---|---|---|---|
| 10 | 00 | 00 | 00 | 2 x 0 = 0 |
| 10 | 01 | 00 | 10 | 2 x 1 = 2 |
| 10 | 10 | 01 | 00 | 2 x 2 = 4 |
| 10 | 11 | 01 | 10 | 2 x 3 = 6 |
| 11 | 00 | 00 | 00 | 3 x 0 = 0 |
| 11 | 01 | 00 | 11 | 3 x 1 = 3 |
| 11 | 10 | 01 | 10 | 3 x 2 = 6 |
| 11 | 11 | 10 | 01 | 3 x 3 = 9 |

# อธิบาย

- การสร้างตารางความจริงของการคูณตัวเลขขนาด 2 บิตที่กล่าวมา
- ตัวแปรทางอินพุตประกอบด้วย $A_1$,$A_0$ กับ$B_1$,$B_0$ถูกแทนค่าด้วยค่า0 กับค่า1 โดยแทนค่าที่ไม่ซ้ำกันขนาด 16 ค่า
- ส่วนทางเอาต์พุตของการคูณประกอบด้วย $P_3$,$P_2$ กับ $P_1$,$P_0$
- ค่าที่เกิดขึ้นของ $P_0$ เกิดจากการนำเอาค่า$A_0$ AND กับ$B_0$
- ส่วนค่าของ$P_1$ เกิดจากการนำค่า$A_1$ AND กับ$B_0$ ไปบวกกับค่า $A_0$ AND กับ $B_1$ ค่าที่เกิดขึ้นจะเป็น$P_1$ กับตัวทด $C_1$
- ค่าของ$P_2$ เกิดจากการนำค่า $A_1$ AND กับ$B_1$ บวกกับค่าตัวทดของ$C_1$ ค่าที่เกิดขึ้นจะเป็น$P_2$ กับตัวทด $C_2$
- นำค่า $C_2$ ไปแทนค่าเป็น $P_3$

# ตารางK-MAP ของ $P_3$

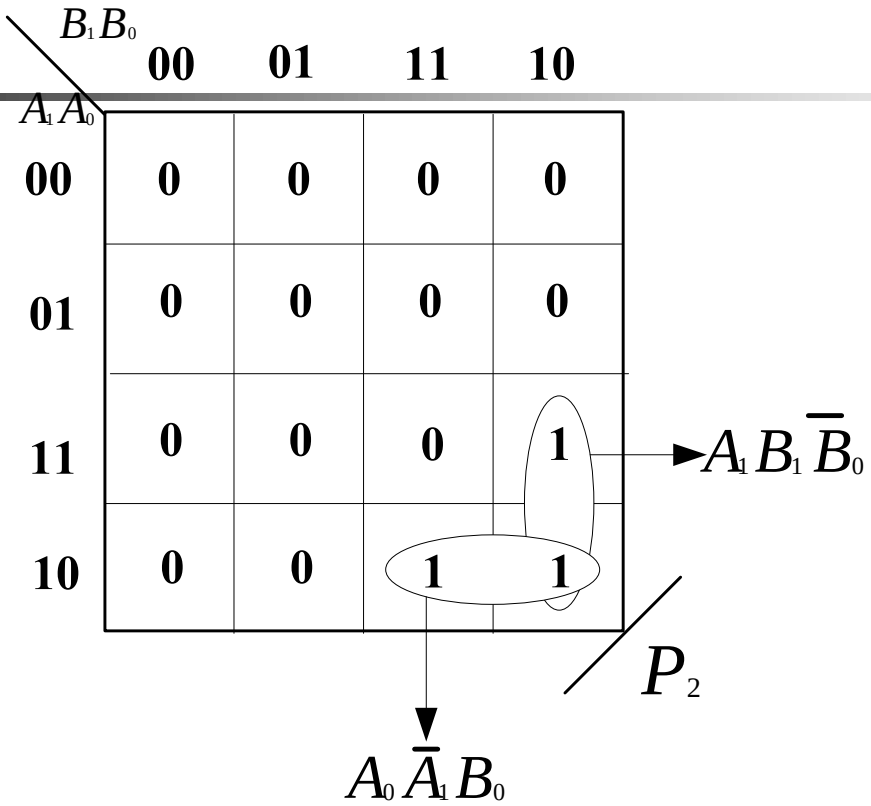



$$P_3 = A_1 A_0 B_1 B_0$$

# อธิบาย

- จากตารางความจริงจะเห็นว่าค่าตัวแปรทางอินพุตเป็น $A_1$,$A_0$ กับ $B_1$,$B_0$ 4ตัวแปร
- ค่าตัวแปรทางเอาต์พุตเป็น $P_3$,$P_2$ กับ $P_1$,$P_0$ 4ตัวแปรเช่นกัน
- ค่า K-Map ที่สร้างขนาด 16 ช่องดูรูปประกอบ
- พิจารณาค่าทางเอาต์พุตที่ละตัวแปร ในกรณีนี้พิจารณาที่ $P_3$ ก่อน
- $P_3$ เป็นค่า 1 เมื่ออินพุต $A_1$,$A_0$ กับ $B_1$,$B_0$ มีค่าเท่ากับ 1
- ลดรูปสมการนี้ได้เป็น $A_1A_0B_1B_0$
- ค่าของ $P_3 = A_1A_0B_1B_0$

# ตารางK-MAP ของ $P_2$

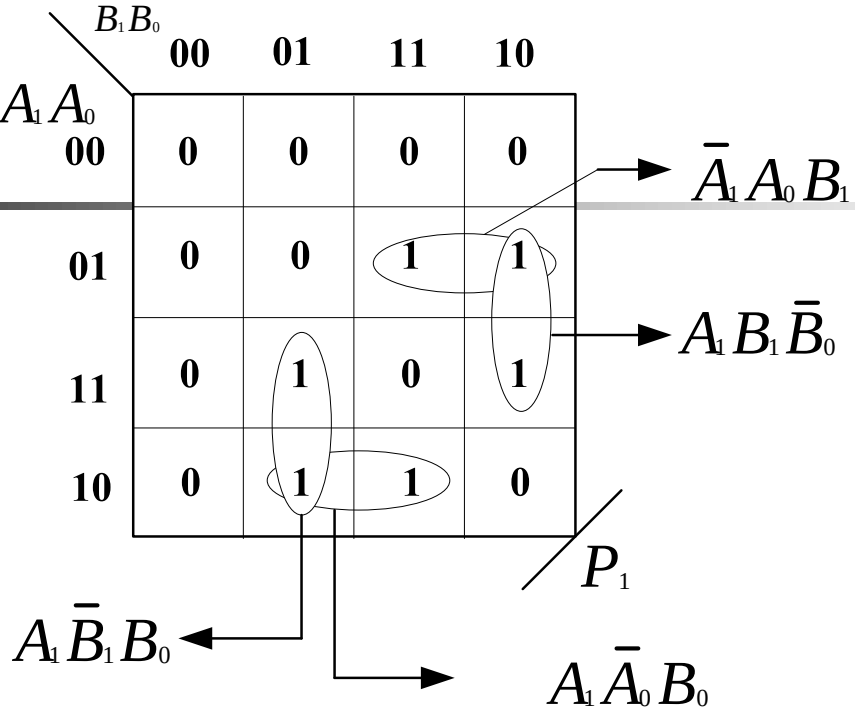



$$P_2 = A_1\overline{B}_1B_0 + A_1\overline{A}_0B_1$$

$$= A_1B_1\left(\overline{B}_0 + \overline{A}_0\right)$$

$$= B_1A_1\left(\overline{B_0A_0}\right)$$

# ตารางK-Map ของ $P_1$




$$P_1 = \bar{A}_1A_0B_1 + A_1B_1\bar{B}_0 + A_1\bar{B}_1B_0 + A_1\bar{A}_0B_0$$

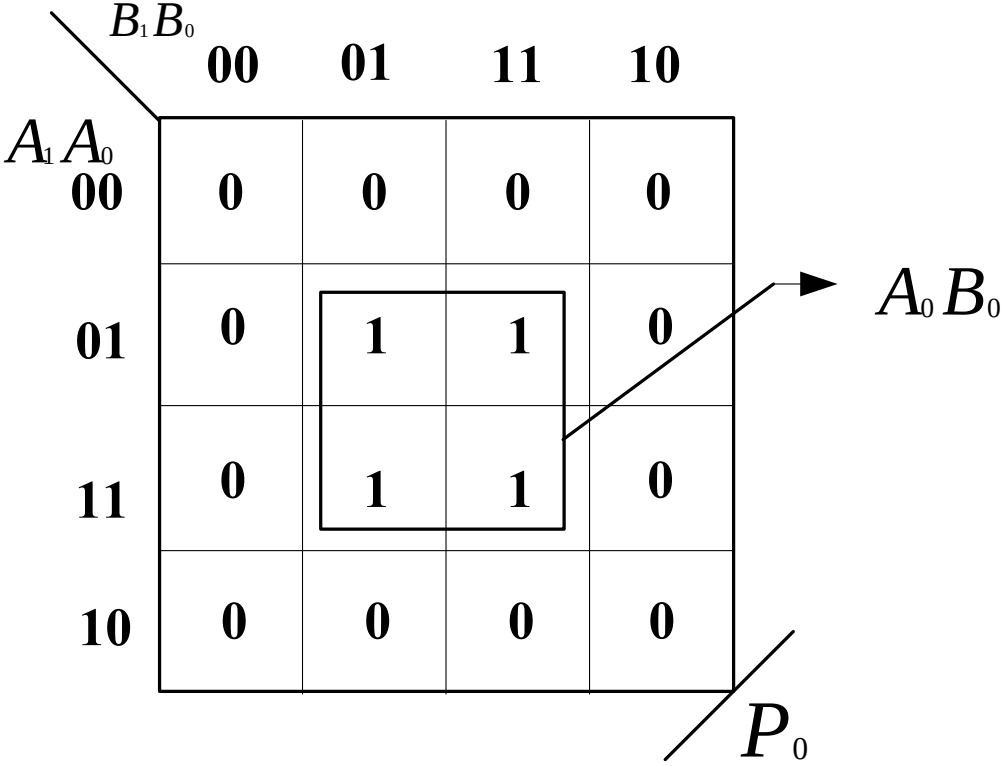
$$= A_1\bar{B}_1B_0 + A_1\bar{A}_0B_0 + A_1B_1\bar{B}_0 + \bar{A}_1A_0B_1$$

$$= A_1B_0(\bar{A}_0 + \bar{B}_1) + A_0B_1(\bar{A}_1 + \bar{B}_0)$$

$$= A_1B_0(\overline{A_0B_1}) + A_0B_1(\overline{A_1B_0})$$

$$= (A_1B_0) \oplus (A_0B_1)$$

# ตารางK-Map ของ $P_0$




$$P_0 = A_0 B_0$$

# อธิบาย

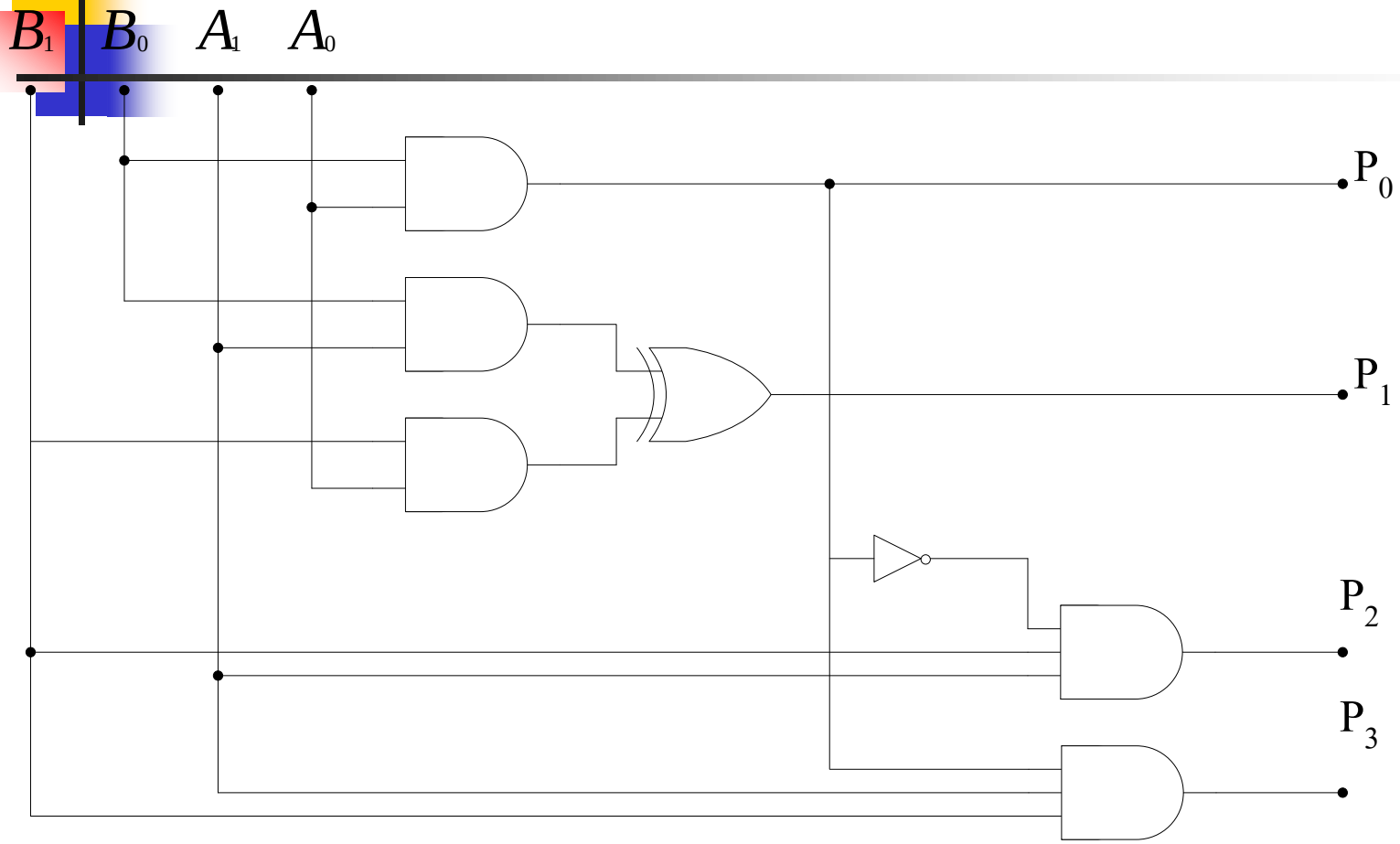
- ในทำนองเดียวกันนักศึกษาได้เรียนวิธีการลดรูป ด้วย K-Map มาจากสัปดาห์ที่4 แล้ว สามารถนำมาใช้ลดรูปได้ $P_2$ กับ $P_1$, $P_0$
- ลองพิจารณาดูจาก K-Map ที่หาค่าต่างๆ จะเห็นได้ว่าค่าที่ได้เป็นดังสมการ

$$P_2 = B_1 A_1 (\overline{B_0 A_0})$$

$$P_1 = (A_1 B_0) \oplus (A_0 B_1)$$

$$P_0 = A_0 B_0$$

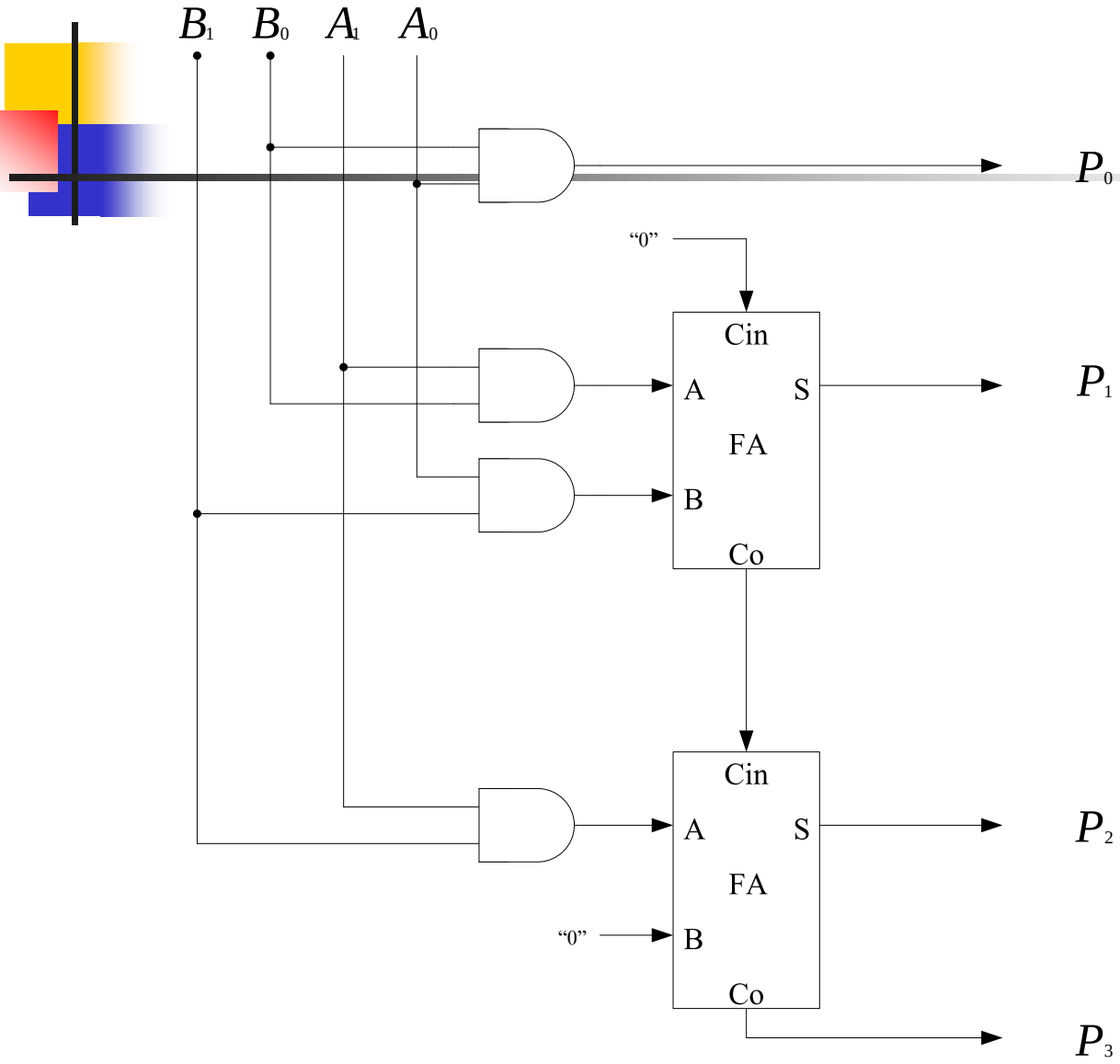
# วงจรคูณ (Multiplier) ขนาด 2 บิต



# อธิบาย

- จากฟังก์ชันของ $P_3,P_2,P_1,P_0$ สามารถนำวงจรเกตมาสร้างเป็นวงจรคูณขนาด 2 จำนวน จำนวนละ 2 บิต
- ข้อเสียของวงจรนี้สามารถทำการคูณได้เพียง 2 บิต
- หากการคูณมีจำนวนบิตมากกว่านี้จะต้องเริ่มการออกแบบตารางความจริงใหม่ และเริ่มลดรูปจากตัวแปรที่มากกว่า 4 ตัวแปร
- ทำให้เกิดความยุ่งยากต่อการสร้างวงจร
- วงจรคูณขนาด 2 บิต จึงสามารถสร้างจากโมดุลของวงจร Full Adder ดังรูปถัดไป

# 572 การออกแบบวงจรคูณขนาด 2 บิต แบบใช้โมดุล

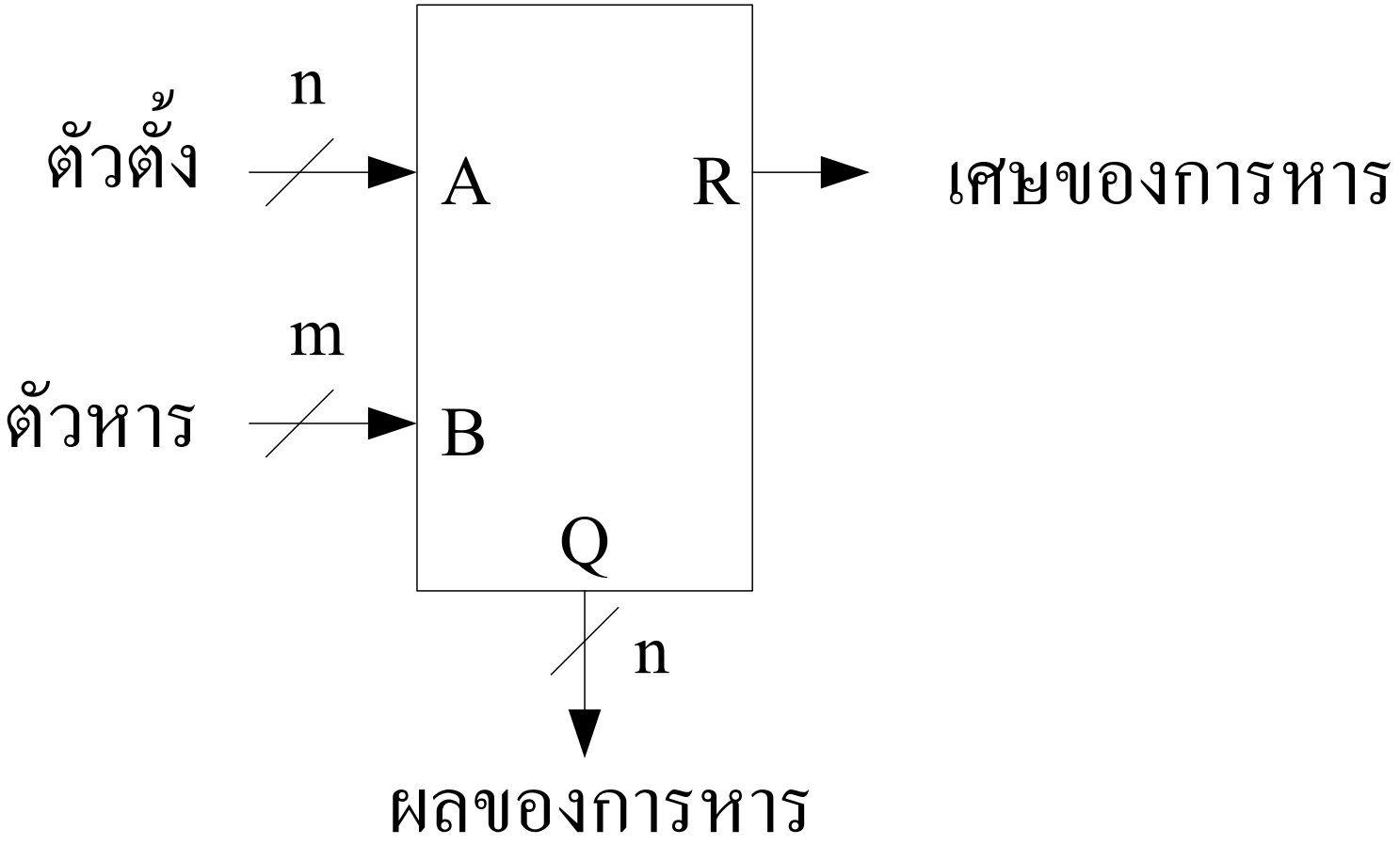


# อธิบาย

- การออกแบบวงจรคูณขนาดหลายบิต
- จากหลักการที่เป็นพื้นฐานของการคูณนำมาออกแบบ โดยการคูณระหว่างตัวตั้งกับตัวคูณให้ใช้ AND Gate การบวกระหว่างบิตจะใช้วงจร Full Adder
- สามารถนำวงจรพื้นฐานของวงจร Full Adder เข้ามาช่วย
- การต่อวงจรต่อลักษณะ Cascade ซึ่งกันและกัน
- มี $C_{in}$ เชื่อมต่อโมดุลของวงจรคูณหลักที่ทดเข้า
- นักศึกษาลองนำแนวคิดเพื่อไปพิจารณาอีกครั้ง

# 58 หลักการวงจรหาร

## วงจรหาร(Divider)



# อธิบาย

- วงจรหารขนาด 2 จำนวน
- กำหนดให้ ตัวตั้ง (Dividend) ให้มีค่าเท่ากับA กำหนดให้มีจำนวน n บิต

  ตัวหาร(Divisor) ให้มีค่าเท่ากับ B กำหนดให้มีจำนวน m บิต

  อินพุตของวงจรหารค่าทั้งสองเป็น m+n

  ทางด้านเอาต์พุตจะเป็นไปตามผลของการหารค่าทั้งสอง

  A/B = Q เป็นผลของการกระทำการหาร เหลือเศษของการหารเป็น R ดังรูปแสดง

# 581 การออกแบบวงจรหาร

## ตารางความจริงของการหาร

| $A \div B$ | $A_1A_0\ B_1B_0$ | $Q_1Q_0\ R_1R_0$ | $QR$ |
|---|---|---|---|
| $0 \div 0$ | 00 00 | 00 00 | 00 |
| $0 \div 1$ | 00 01 | 00 00 | 00 |
| $0 \div 2$ | 00 10 | 00 00 | 00 |
| $0 \div 3$ | 00 11 | 00 00 | 00 |
| $1 \div 0$ | 01 00 | ΘΘ ΘΘ | - |
| $1 \div 1$ | 01 01 | 01 00 | 00 |
| $1 \div 2$ | 01 10 | 00 01 | 00 |
| $1 \div 3$ | 01 11 | 00 01 | 00 |

| $A \div B$ | $A_1A_0\ B_1B_0$ | $Q_1Q_0\ R_1R_0$ | $QR$ |
|---|---|---|---|
| $2 \div 0$ | 10 00 | ΘΘ ΘΘ | - |
| $2 \div 1$ | 10 01 | 00 00 | 20 |
| $2 \div 2$ | 10 10 | 00 00 | 10 |
| $2 \div 3$ | 10 11 | 00 00 | 02 |
| $3 \div 0$ | 11 00 | ΘΘ ΘΘ | - |
| $3 \div 1$ | 11 01 | 01 00 | 30 |
| $3 \div 2$ | 11 10 | 00 01 | 11 |
| $3 \div 3$ | 11 11 | 00 01 | 10 |

# อธิบาย

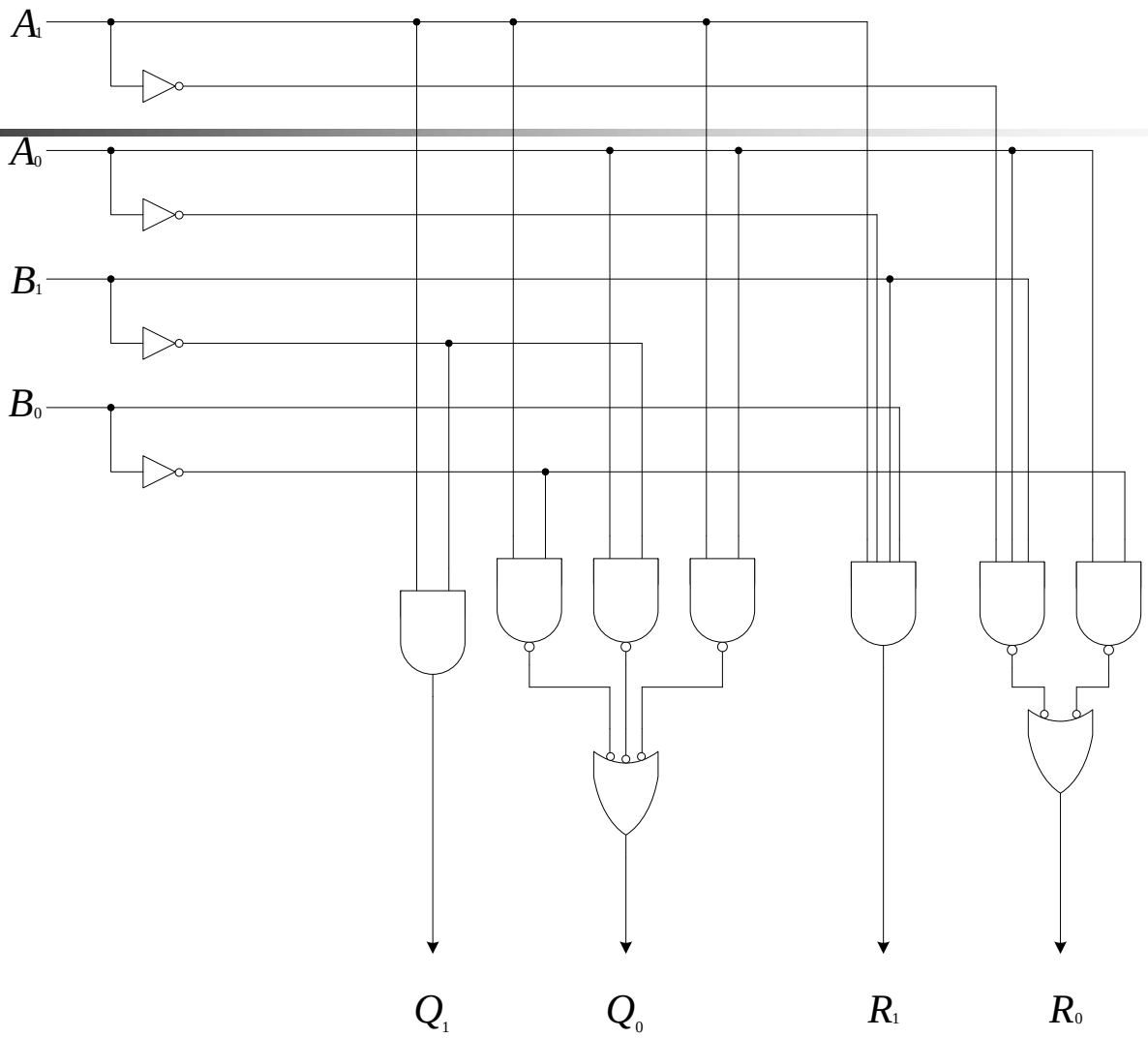
- จากตารางนำค่าทางเอาต์พุตที่สัมพันธ์กับทางอินพุตไปหาค่าฟังก์ชันทางลอจิก
- วิธีการหา ใช้ในรูปแบบของ K-Map หรือวิธีการ ทางพีชคณิตก็ได้ตามที่นักศึกษาได้เรียนผ่านมาแล้ว
- ค่าฟังชันที่หาได้ คือ $Q=A_1\overline{B}_1$

  $Q_0=A_0\overline{B}_1+A_1A_0+A_1\overline{B}_0$

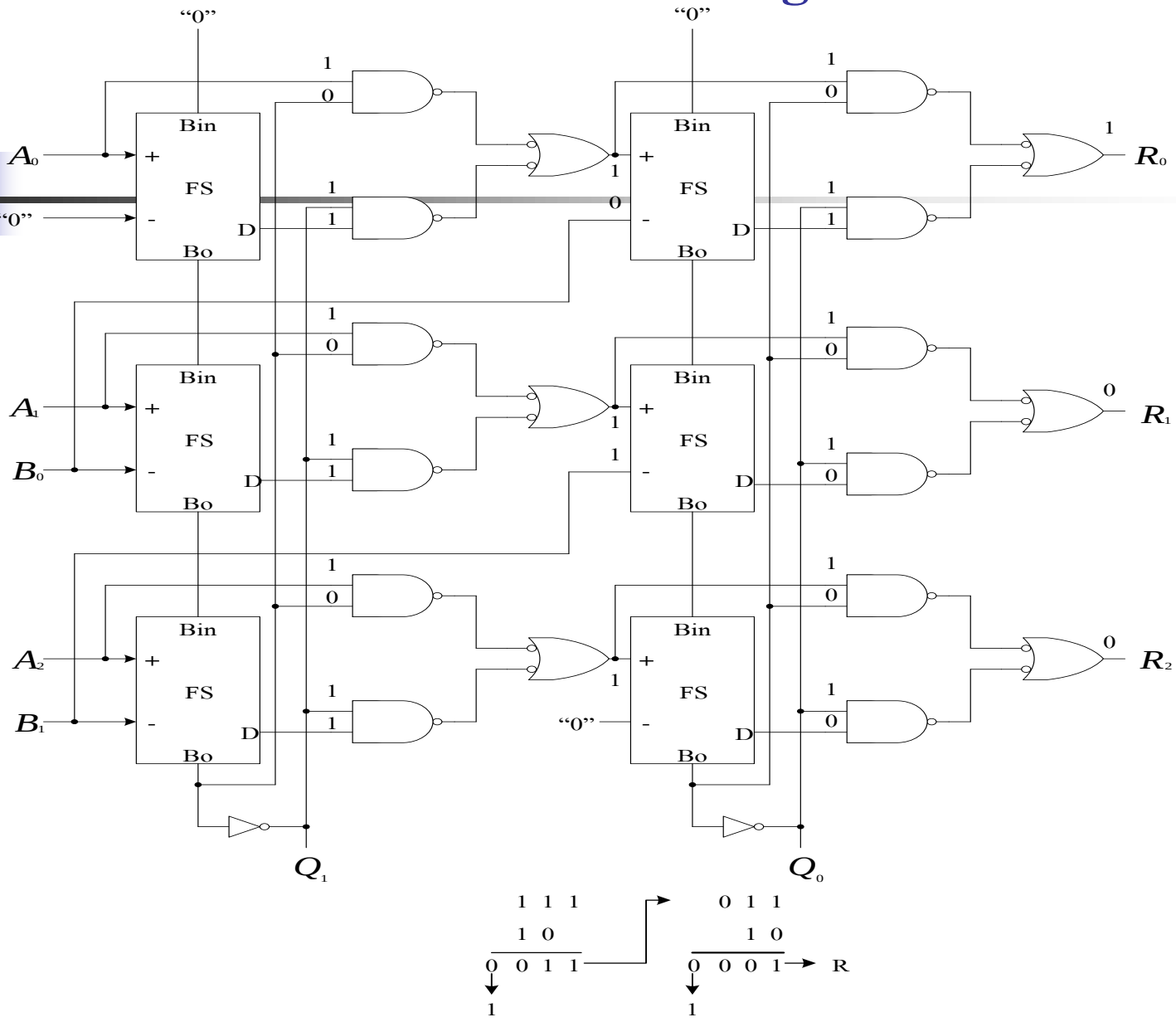
  $R_1=A_1\overline{A}_0B_1B_0$

  $R_0=\overline{A}_1A_0B_1+A_0\overline{B}_0$
- นำค่าที่ได้ไปเขียนเป็นวงจรหารขนาด 2 จำนวนๆ ละ 2 บิต ดังรูปถัดไป

# วงจรหารขนาด 2 บิต

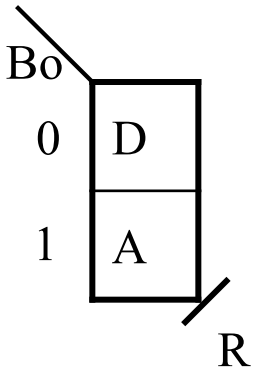


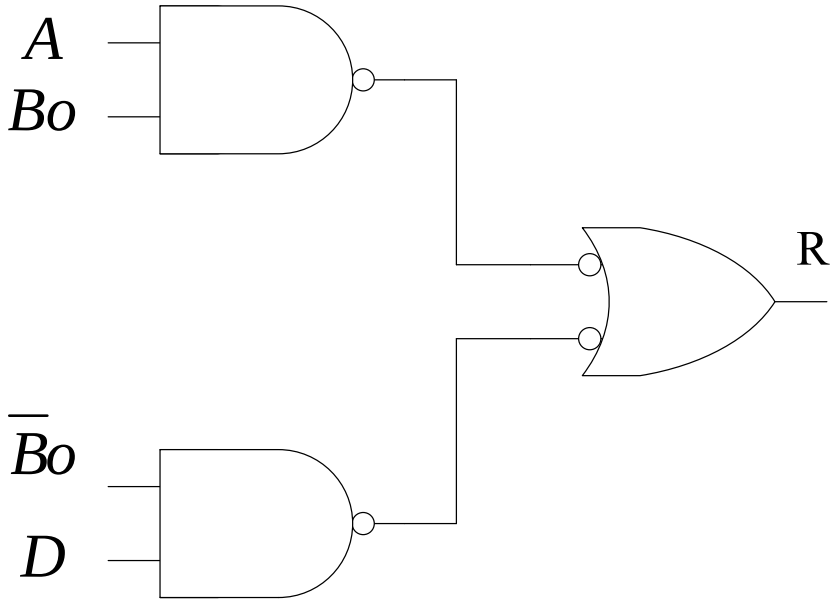
# 582 การออกแบบวงจรหารเลขแบบ Algorithm ขนาด 2 บิต



# ตารางความสัมพันธ์ของเศษหาร

| Bo | R |
|---|---|
| 0 | D |
| 1 | A |



# อธิบาย

- การออกแบบวงหารแบบ Algorithm Division หรือการหารแบบโมดุล
- ขั้นตอนที่1 จัดวางตำแหน่ง $M_{SB}$ ตัวตั้งและค่าตัวหารให้อยู่ตำแหน่งให้ตรงกันเพื่อทำการลบ นำผลไปเลื่อนบิตไปทาง $L_{SB}$ 1บิต
- ขั้นตอนที่2 นำเศษของการหารไปเป็นตัวตั้งของการลบต่อไป $B_o$=0
- R=A , R=เศษของการกระทำการหาร D=ผลลบ A เป็นตัวตั้ง
- ค่าความสัมพันธ์ R หาค่าจากตาราง K-Map
- ขั้นตอนที่ 3 นำค่า$B_o$ มาทำComplement เพื่อเป็นค่าของผลหาร

# ทดสอบความรู้สัปดาห์ที่ 10

- 1. จงอธิบายหน้าที่ความแตกต่างระหว่าง AU กับ LU
- 2. จงออกแบบวงจรบวก/ลบขนาด 4 บิต เป็นแบบโมดุล
- 3. จงออกแบบวงจรคูณ/หารขนาด 4 บิต เป็นแบบโมดุล
- 4. จากข้อ 1-2 ให้นำวงจรที่ทำการออกแบบมาควบคุมฟังก์ชันการทำงานทั้งหมด เป็นโมดุลรวม

# เนื้อหาคำบรรยายในสัปดาห์ที่ 11 การออกแบบวงจรลอจิกซีเควนเชียล

**61 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์**

611 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรขยายความกว้างของพัลส์

612 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรลดความกว้างของพัลส์

613 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรกำจัดพัลส์ที่ไม่ต้องการ

614 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ในรูปแบบไอซีสำเร็จรูป

**62 การทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์**

621 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ทรานซิสเตอร์

622 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซี TTL

623 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้คริสตอลควบคุมความถี่

624 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซี ซีมอส

# เนื้อหาคำบรรยาย การออกแบบวงจรลอจิกซีเควนเชียล(ต่อ)

**63 การนำไอซี 555 ไปใช้งานในรูปแบบต่างๆ**

631 วงจรโมโนสเตเบิลใช้ไอซี 555

632 วงจรอะสเตเบิลใช้ไอซี555

**64 การทำงานของวงจรไบสเตเบิล**

641 R-S Latch

642 R-S ฟลิปฟลอป

643 T ฟลิปฟลอป

644 D ฟลิปฟลอป

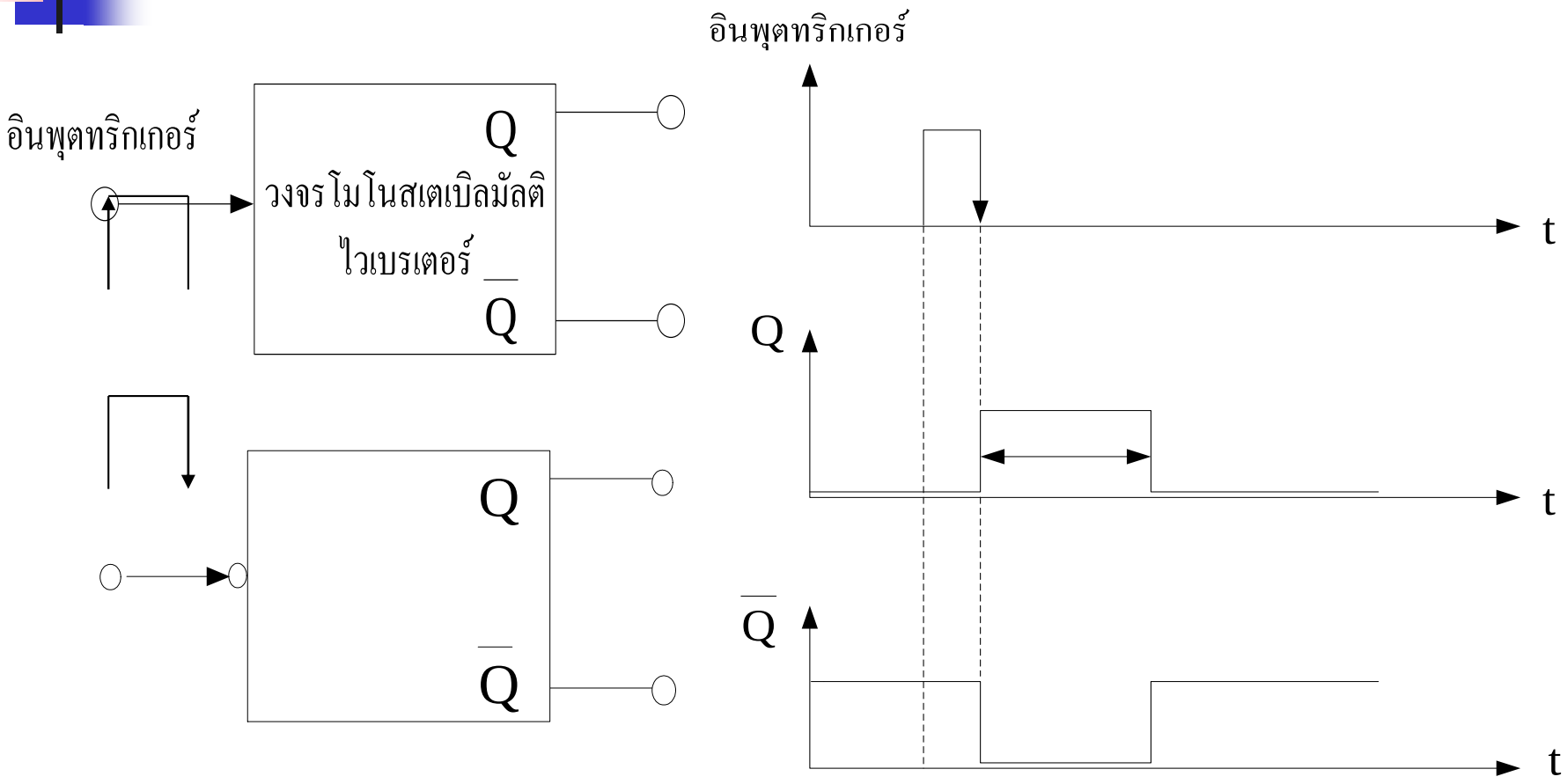
645 J-K ฟลิปฟลอป

646 J-K Master-Slave ฟลิปฟลอป

647 การแปลงฟลิบฟลอบแบบต่างๆ

# 61 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์

จะมีสถานะทางเอาต์พุตตรงกันข้ามกับสถานะปกติ เมื่อมีสัญญาณทริกทางด้านอินพุต



# อธิบาย

- วงจรโมโนสเตเบิลสามารถแบ่งออกเป็นการทำงานดังนี้
- วงจรโมโนสเตเบิลมัลติไวเบรเตอร์แบบทริกได้ครั้งเดียว (Non Trigerable  Monostable Multivibrator) หรือเรียกอีกอย่างหนึ่งว่า One Shot
- วงจรโมโนสเตเบิลมัลติไวเบรเตอร์แบบทริกได้หลายครั้ง (Trigerable Monostable Multivibrator)

# อธิบาย(ต่อ)

การทำงานของวงจร ถ้าไม่มีสัญญาณทริกที่อินพุตวงจรจะคงอยู่ในสถานะคงที่ โดยมี $\bar{Q} = 0$ และ $Q = 1$ แต่ถ้าหากขณะใดมีสัญญาณทริกที่อินพุตจะทำให้เปลี่ยนเป็นสภาวะตรงกันข้าม $Q = 1$ และ $\bar{Q} = 0$

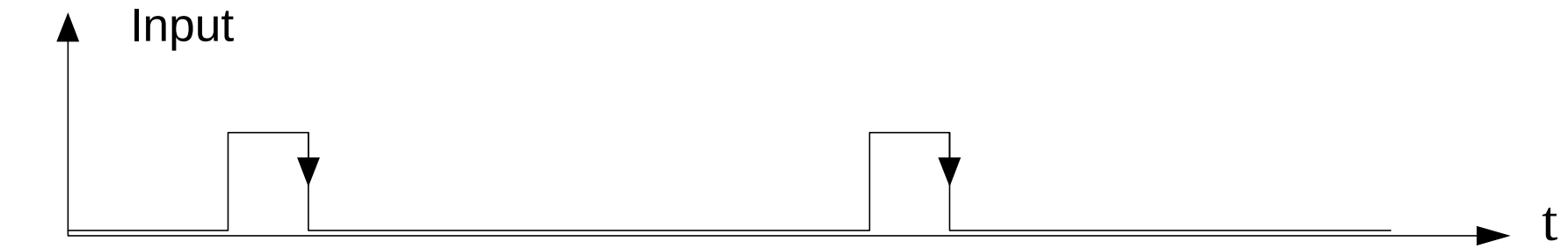
สัญญาณอินพุตควบคุมในช่วงบวก สามารถเขียนได้ดังแสดงในรูป

ส่วนสัญญาณอินพุตที่ควบคุมในช่วงลบจะมีวงกลมปลายลูกศรแสดงในรูป

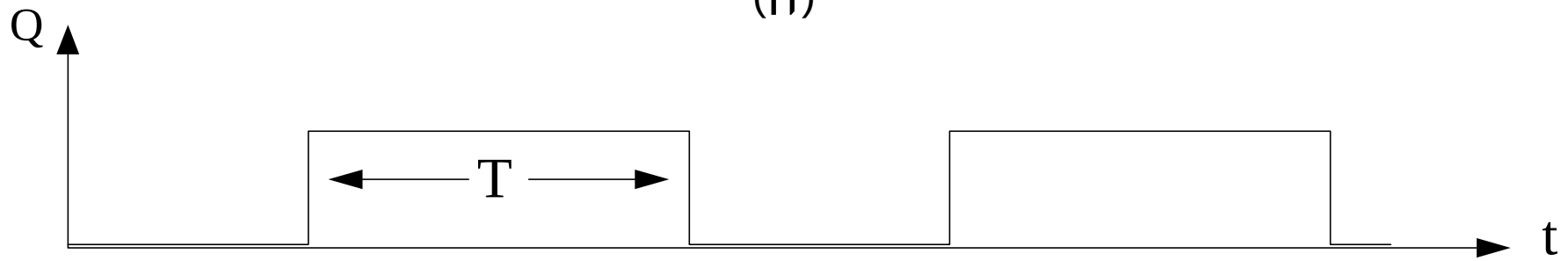
**คุณสมบัติของวงจรโมโนสเตเบิลแบบนี้สามารถนำไปใช้ประโยชน์ได้ดังนี้**

- ใช้เป็นวงจรขยายความกว้างของช่วงพัลส์
- ใช้เป็นวงจรลดความกว้างของสัญญาณพัลส์
- ใช้เป็นวงจรกำจัดพัลส์ที่ไม่ต้องการ

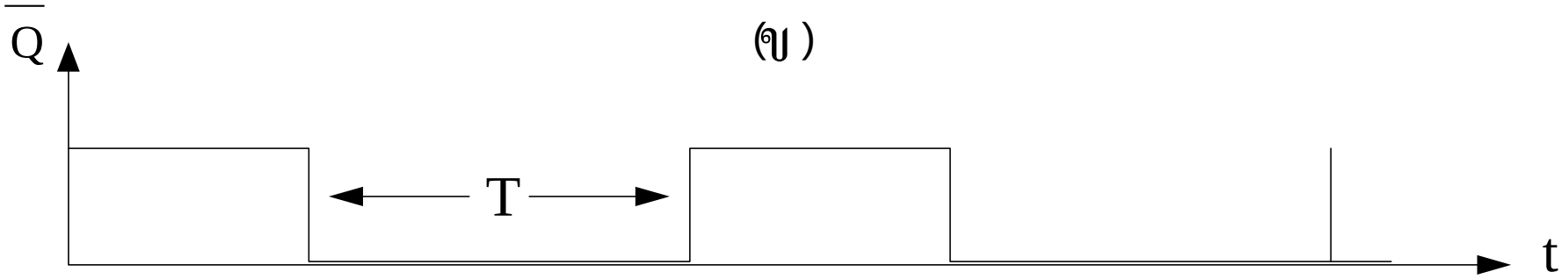
**611 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ไปใช้เป็นวงจรขยายความกว้างของพัลส์**




(ก)




(ข)

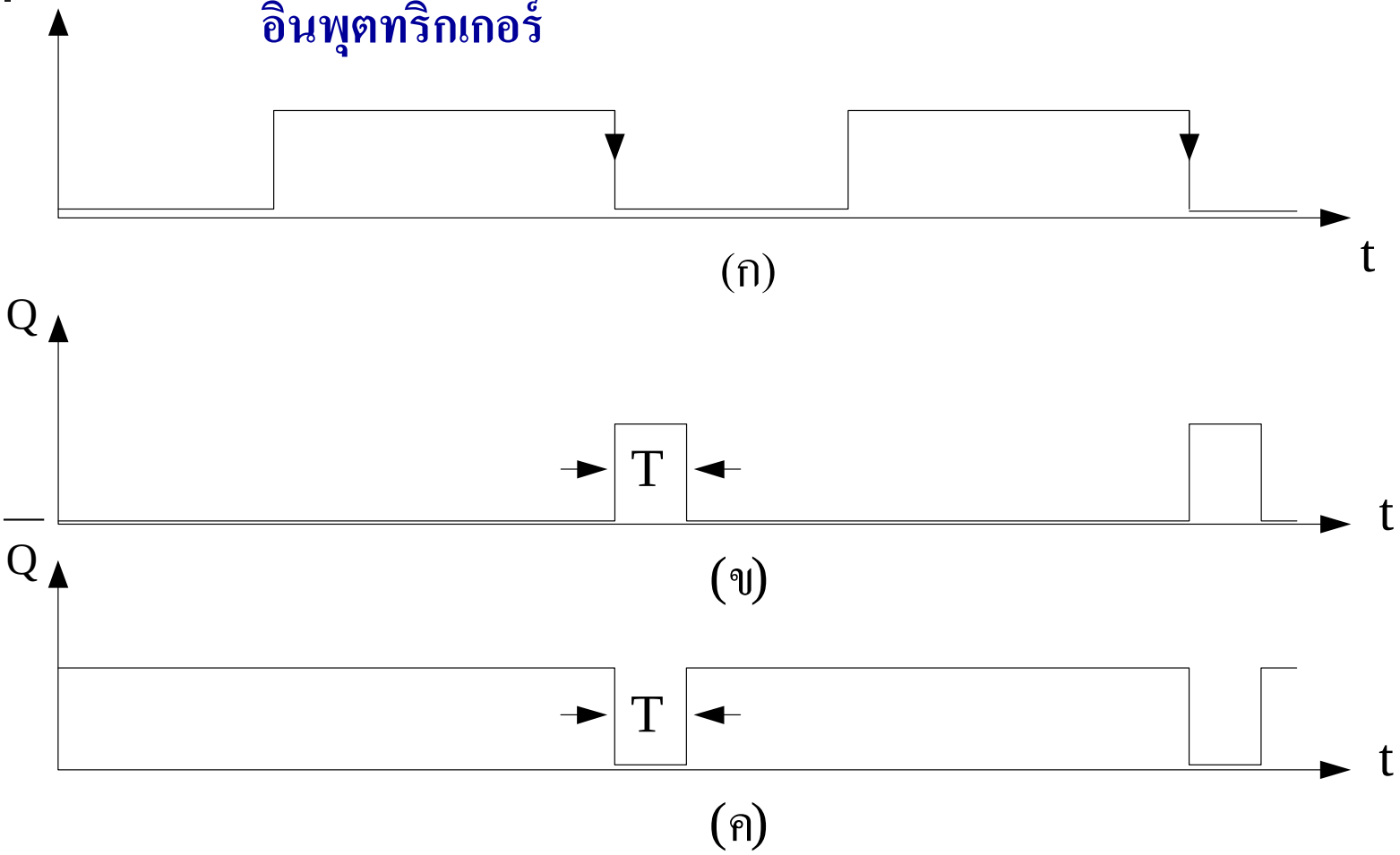



(ค)

# อธิบาย

- การนำวงจรโมโนสเตเบิลไปใช้ควบคุมความกว้างของสัญญาณเอาต์พุต
- สัญญาณที่ป้อนที่อินพุตทริกเกอร์ดังรูป (ก) เป็นสัญญาณควบคุมในช่วงลบจะมีการทำงานเมื่อได้สัญญาณอินพุตเปลี่ยนค่าจากระดับสูงไประดับต่ำ
- ค่าสัญญาณที่เอาต์พุต (Q) จะเกิดการเปลี่ยนแปลงจากค่าระดับต่ำ ไปสูงในช่วงเวลาที่ตั้งค่าไว้ (T) ดังในรูป (ข)
- ในทำนองเดียวกัน ค่าสัญญาณที่เอาต์พุต ($\overline{Q}$) จะเกิดการเปลี่ยนแปลงจากค่าระดับสูงไปต่ำในช่วงเวลาที่ตั้งค่าไว้ (T) ดังในรูป(ค) สัญญาณในรูป (ข), (ค) เป็นสัญญาณที่กลับกัน
- การขยายความกว้างของช่วงพัลส์ เพื่อให้ค่าความกว้างทางเอาต์พุตมีค่าเพียงพอต่อการควบคุมในวงจรดิจิตอล ซึ่งหมายความว่าบางครั้งสัญญาณทางอินพุตอาจจะมีค่าเป็นรูปสัญญาณ สไปค์ที่มีค่าช่วงเวลาแคบมากไม่สามารถนำไปควบคุมการทำงานได้

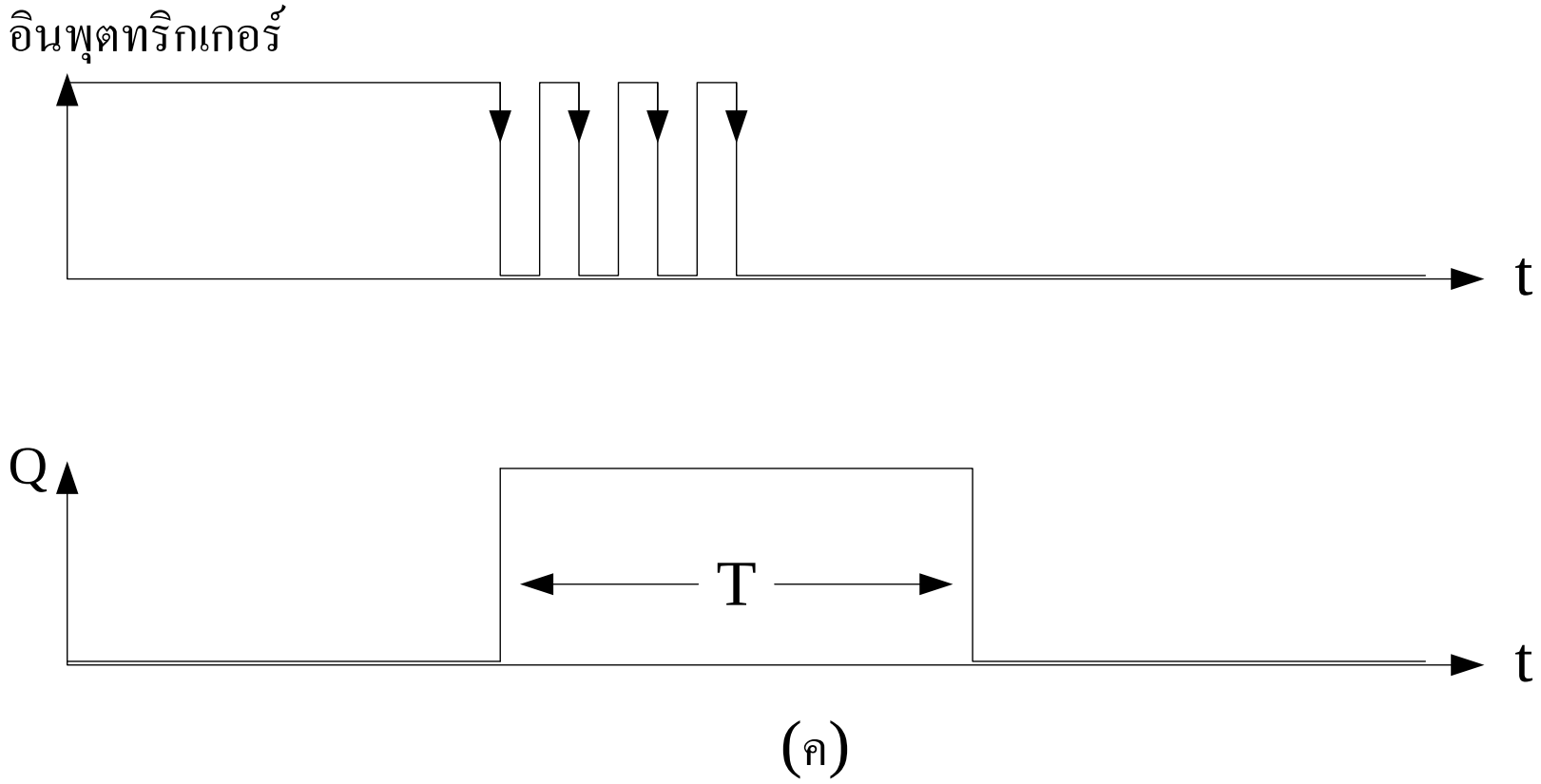
# 612 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรลดความกว้างของสัญญาณพัลส์



# อธิบาย

- การนำวงจรโมโนสเตเบิลไปใช้ลดความกว้างของสัญญาณทางอินพุตให้มีค่าลดลงของสัญญาณเอาต์พุต
- สัญญาณที่ป้อนที่อินพุตทริกเกอร์ดังรูป (ก) เป็นสัญญาณควบคุมในช่วงลบจะมีการทำงานเมื่อได้สัญญาณอินพุตเปลี่ยนค่าจากระดับสูงไประดับต่ำ
- ค่าสัญญาณที่เอาต์พุต (Q) จะเกิดการเปลี่ยนแปลงจากค่าระดับต่ำ ไปสูงในช่วงเวลาที่ตั้งค่าไว้ (T) ดังในรูป (ข) จะเห็นว่าสัญญาณจะมีความกว้างลดลง
- ในทำนองเดียวกัน ค่าสัญญาณที่เอาต์พุต (Q) จะเกิดการเปลี่ยนแปลงจากค่าระดับสูงไปต่ำในช่วงเวลาที่ตั้งค่าไว้ (T) ดังในรูป (ค) ค่าความกว้างทางเอาต์พุตก็มีค่าลดลง
- การลดความกว้างของช่วงพัลส์ เพื่อให้ค่าความกว้างทางเอาต์พุตมีค่าเพียงพอต่อการควบคุมในวงจรดิจิตอล ซึ่งหมายความว่าบางครั้งสัญญาณทางอินพุตอาจจะมีค่าเป็นค่าที่ระดับสัญญาณเปลี่ยนแปลงที่ระดับค่าสูงพร้อมค่ารบกวนมากซึ่งไม่มีความจำเป็นในความกว้างมาก จึงจำเป็นต้องใช้วงจรโมโนสเตเบิลในการลดพัลส์

# 613 การนำวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้เป็นวงจรกำจัดพัลส์ที่ไม่ต้องการ



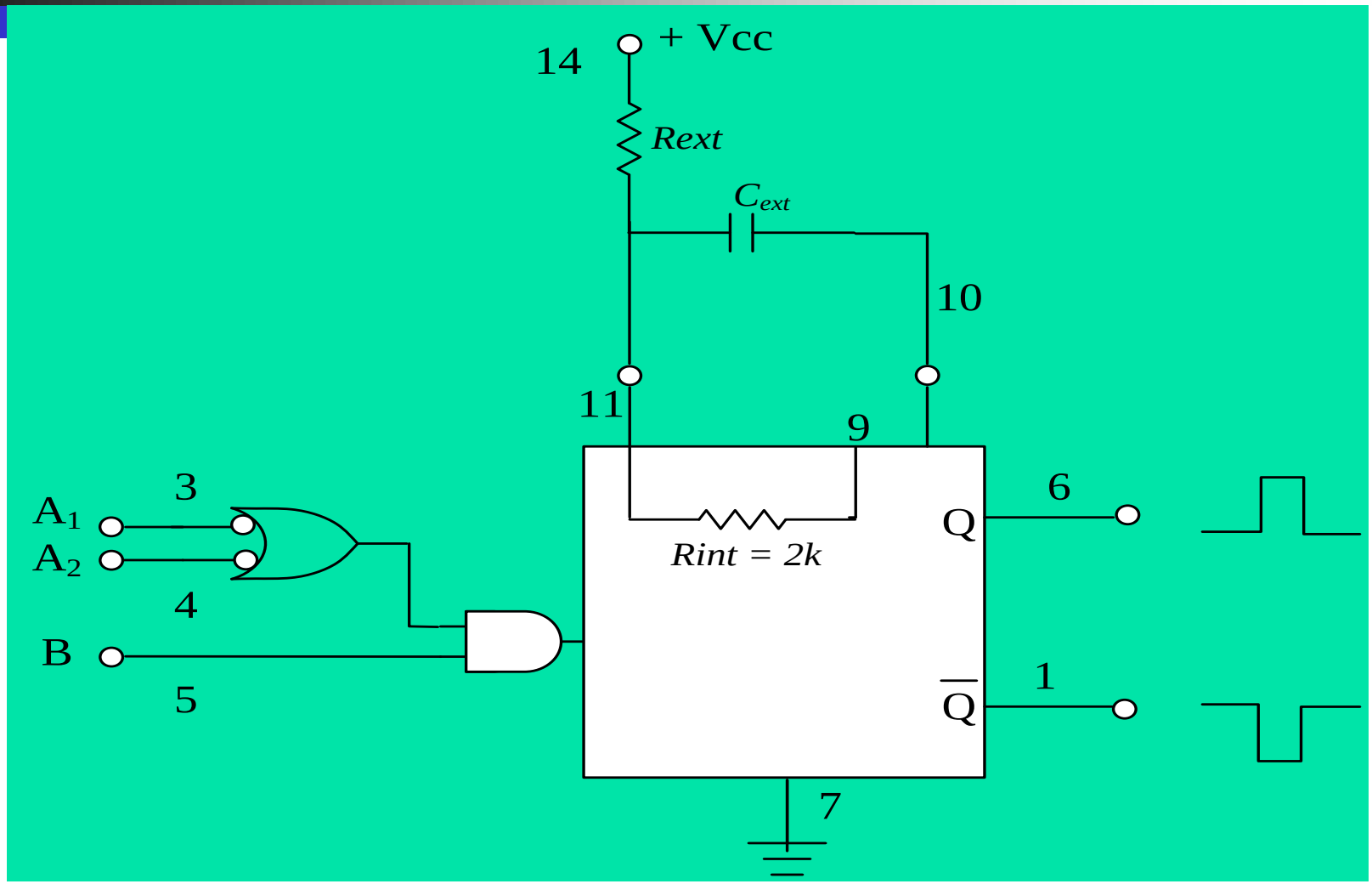

(ค)

# อธิบาย

- การนำวงจรโมโนสเตเบิลไปกำจัดพัลส์ที่ไม่ต้องการ เช่นพัลส์ที่เกิดจากการปิดเปิดสวิทซ์
- ขณะที่สวิทซ์เปลี่ยนตำแหน่งจากสถานะลอจิก 1 เป็น 0 อาจจะทำให้คอนแทคเกิดการปิด-เปิดหลายครั้ง ซึ่งผลทำให้เกิดพัลส์ที่ไม่ต้องการ ดังนั้น
- หากนำสวิทซ์ไปควบคุมการทำงานลอจิก เพื่อป้องกันความผิดพลาดของสวิทซ์ให้ใช้วงจรโมโนสเตเบิลแบบทริกครั้งเดียวควบคุมพัลส์ที่จะไปควบคุมวงจรการทำงานอีกทีหนึ่ง

# 614 การทำงานของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ในรูปแบบไอซีสำเร็จรูป

## วงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้ไอซีเบอร์ 74121



# อธิบาย

- การนำเอาไอซีสำเร็จรูปไปใช้งานในวงจรโมโนสเตเบิลมัลติไวเบรเตอร์ โดยการใช้ RC ภายนอก กำหนดช่วงเวลา T
- วงจรนี้สามารถใช้กับแหล่งจ่ายแรงดันสูงได้
- การทำงานสามารถใช้กับการทริกทั้ง Active High , Active Low
- ค่าความต้านทานภายนอกจะใช้ประมาณ 1.4KOhm-40 KOhm
- ค่าคาปาซิเตอร์ใช้ประมาณ 0.1-1000 μFD
- ค่า Tสามารถหาค่าจาก 0.7RC

# ตารางความจริงของวงจรโมโนสเตเบิลมัลติไวเบรเตอร์

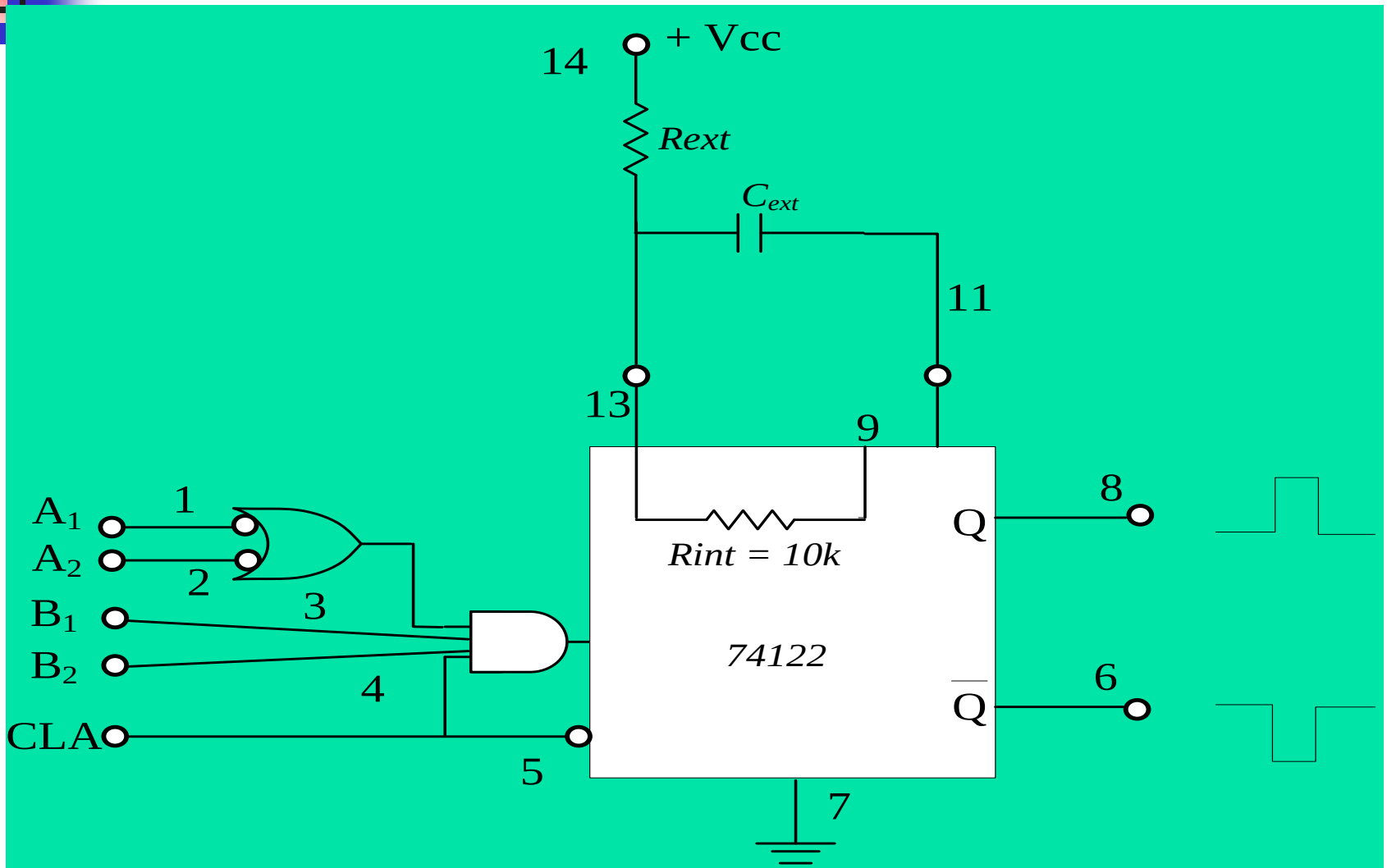
## แบบใช้ไอซีเบอร์74121

| อินพุต | | | เอาต์พุต | |
|---|---|---|---|---|
| $A_1$ | $A_2$ | B | Q | $\overline{Q}$ |
| L | X | H | L | H |
| X | L | H | L | H |
| X | X | L | L | H |
| H | H | X | L | H |
| H | ↓ | H | ⎍ | ⎍ |
| ↓ | H | H | ⎍ | ⎍ |
| ↓ | ↓ | H | ⎍ | ⎍ |
| L | X | ↑ | ⎍ | ⎍ |
| H | L | ↑ | ⎍ | ⎍ |

# อธิบาย

จากตารางความจริงสามารถอธิบายการทำงานของการควบคุมวงจรโมโน-สเตเบิลมัลติไวเบรเตอร์ ใช้ไอซีเบอร์ 74121 คุณสมบัติของวงจรโมโนสเตเบิลไว-เบรเตอร์ทางด้านอินพุตจะควบคุมด้วยสัญญาณ Active Low ของ$A_1$, $A_2$ ส่วนอินพุต B จะถูกควบคุมด้วย Active High จะเห็นว่าอินพุตของส่วน A กับอินพุตของส่วน B จะต้องมีสัญญาณในส่วนใดส่วนหนึ่งเป็น Active แทนระดับ High จึงจะสามารถควบคุมสัญญาณทางเอาต์พุตได้ ดังตารางแสดง Active $A_1$, $A_2$หรือทั้งคู่ของอินพุต A เป็น Active Low นำไปสร้างสัญญาณควบคุมที่ออกจากเอาต์พุต ออร์เกตเป็น Active High นำไปแอนด์กับสัญญาณควบคุม B ที่ระดับ High หรือ Active High สร้างสัญญาณควบคุม IC 74121 เป็นอินพุต Active High จึงจะสร้างสัญญาณทางเอาต์พุตของวงจรโมโนสเตเบิลได้

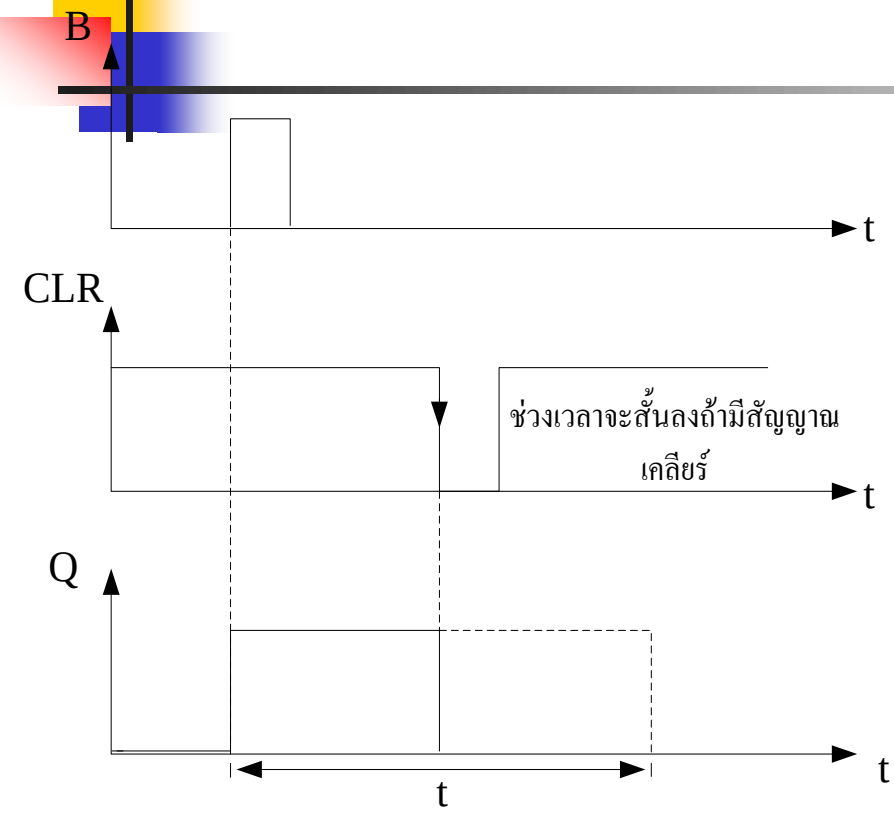
# วงจรโมโนสเตเบิลมัลติไวเบรเตอร์ใช้ไอซีเบอร์ 74122

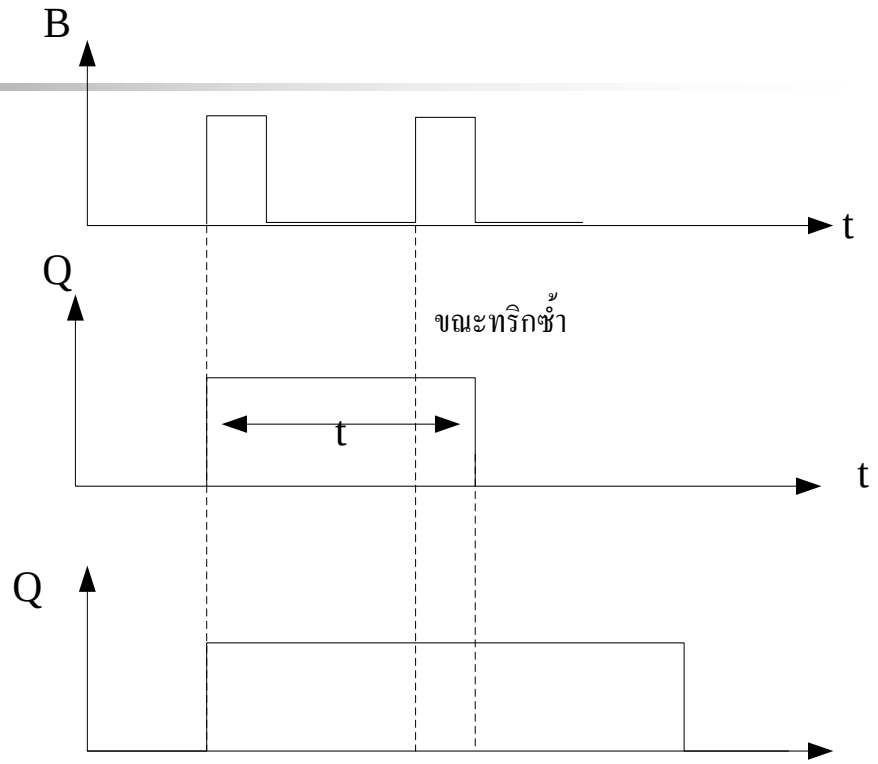


# อธิบาย

- วงจรโมโนสเตเบิลแบบทริกซ้ำได้เบอร์ 74122
- $A_1$, $A_2$ จะทริกด้วย Negative Edge หรือต่อลงกราวนด์
- $B_1$, $B_2$ จะทริกด้วย Positive Edge ต่อกับแหล่งจ่ายไฟบวก หรือปล่อยลอย สภาวะนี้เป็นบวกเช่นกัน แนะนำควรต่อไฟบวกเพื่อความมั่นใจ

# รูปสัญญาณเคลียร์และสัญญาณทริกซ้ำ



# ตารางความจริงของวงจรมัลติไวเบรเตอร์แบบทริกซ้ำได้

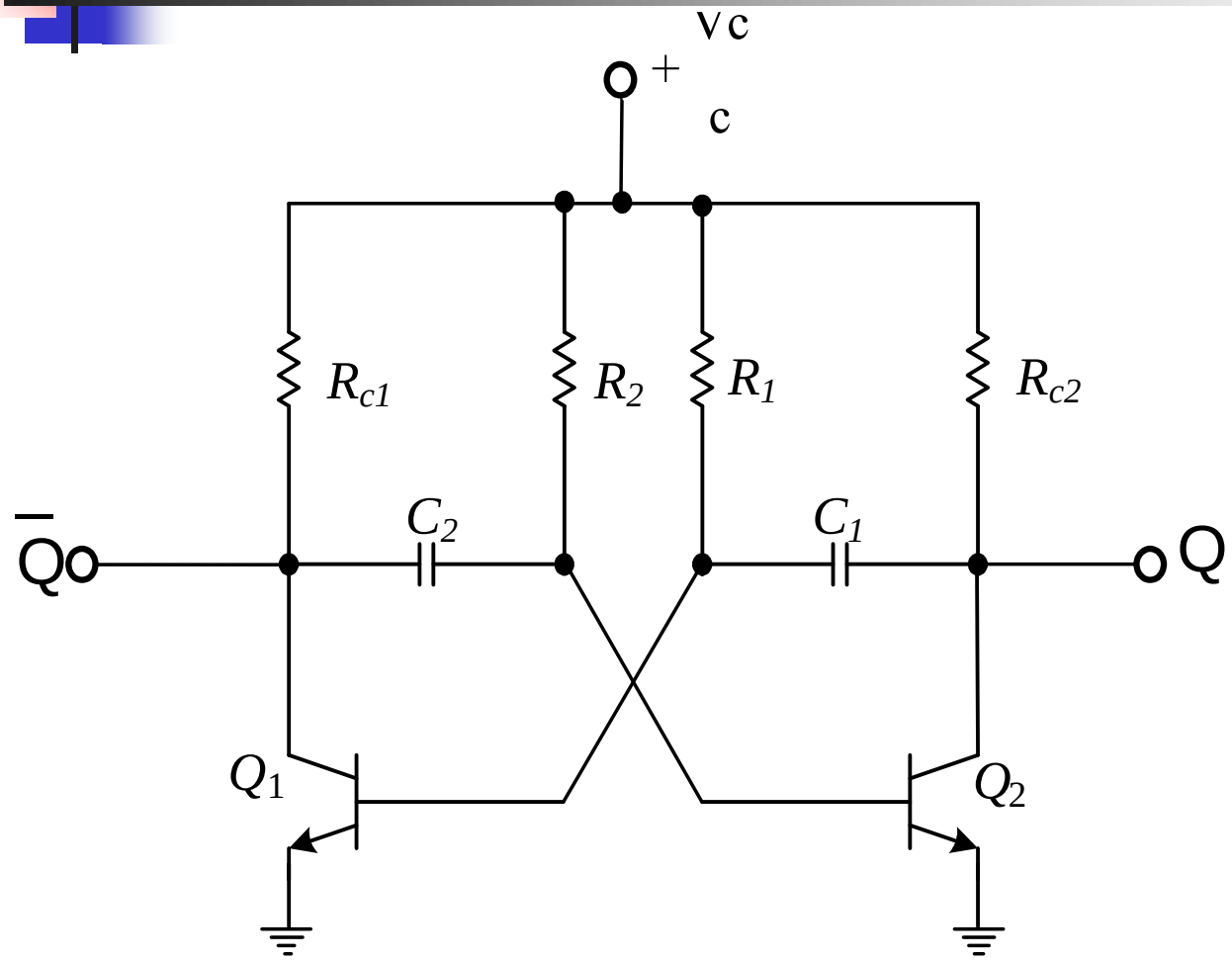
| อินพุต | | | | | เอาต์พุต | |
|---|---|---|---|---|---|---|
| เคลียร์ | $A_1$ | $A_2$ | $B_1$ | $B_2$ | Q | $\overline{Q}$ |
| L | X | X | X | X | L | H |
| X | H | H | X | X | L | H |
| X | X | X | L | X | L | H |
| X | X | X | X | L | L | H |
| X | L | X | H | H | L | H |
| H | L | X | ↑ | H | ⎍ | ⎍ |
| H | L | X | ↑ | ↑ | ⎍ | ⎍ |
| ↓ | X | L | H | H | L | H |
| H | X | L | ↑ | H | ⎍ | ⎍ |
| H | X | L | H | ↑ | ⎍ | ⎍ |
| H | H | ↓ | H | H | ⎍ | ⎍ |
| H | ↓ | ↑ | H | H | ⎍ | ⎍ |
| H | ↓ | H | H | H | ⎍ | ⎍ |
| ↑ | L | X | H | H | ⎍ | ⎍ |
| ↑ | X | L | H | H | ⎍ | ⎍ |

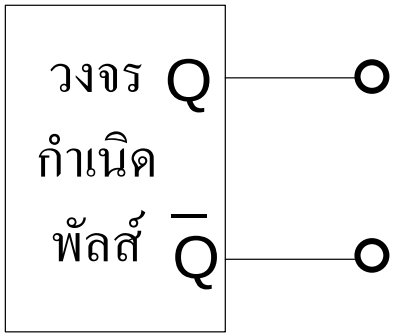
# อธิบาย

- ตารางความจริงของวงจรมัลติไวเบรเตอร์แบบทริกซ้ำได้หลายครั้ง เบอร์ 74122
- การทำงานของ $A_1$ และ $A_2$ เป็นอินพุตที่ทำงานในช่วงลบ ผ่านออร์เกต สามารถทำงานอินพุตใดอินพุตหนึ่งก็ได้
- $B_1$, $B_2$ เป็น Active High จะเห็นว่าค่า $B_1$, $B_2$ จะต้องเป็น High ทั้งคู่ไม่สามารถทำงานเพียงอินพุตเดียว
- สัญญาณ CLR คืออินพุต Clear ของวงจรจะเป็น Active High
- จากการทำงานของวงจรจะเห็นว่าสัญญาณทั้ง 3 ของอินพุตจะต้องสัมพันธ์กันดังนี้ ($A_1$หรือ $A_2$), ($B_1$, $B_2$), (CLA)

# 62 การทำงานของวงจรอะสเตเบิลมัลติไวเบรเตอร์

## 621 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ทรานซิสเตอร์



# อธิบาย

- ค่า $R_2$ , $C_2$ เป็นวงจรประจุให้กับ $Q_2$
- ค่า $R_1$, $C_1$ เป็นวงจรประจุให้กับ $Q_1$
- $R_{C1}$,$R_{C2}$ เป็นโหลด$I_C$ ของทรานซิสเตอร์ทั้งสอง
- การทำงานของโมโนสเตเบิลแบบไบสเตเบิลที่ทำงานด้วยทรานซิสเตอร์ ค่าแรงดันที่ประจุใน C ถ้าค่าแรงดันของ $V_{C1}$ มีค่ามากกว่า $V_{C2}$ จะทำให้ $V_{BE1}$ ของๆ $Q_1$ ทำงาน $Q_2$ ก็จะOFF $C_2$ ก็จะทำการประจุผ่าน $R_2$ ขณะเดียวกัน $C_1$ ก็จะทำการคายประจุผ่าน $V_{BE1}$ ในช่วงที่มีการคายประจุทรานซิสเตอร์ก็จะเปลี่ยนจากON เป็นOFF $C_2$ก็ประจุจนค่าแรงดันเพียงพอต่อการทำงาน $Q_2$ ก็จะทำงาน $Q_1$ ก็จะOFFดังนั้นค่าเอาต์พุต Q จะเปลี่ยนแปลงตามการทำงานของทรานซิสเตอร์ที่ขาคอลเล็คเตอร์ สามารถหาค่าความถี่ได้จาก

**ตัวอย่าง จงออกแบบวงจรไบสเตเบิลที่ออกแบบจากทรานซิสเตอร์ 2 ตัว**

ถ้าวงจรมี $R_1 = R_2 = 10$ กิโลโอห์ม

$C_1 = C_2 = 120$ พิโคฟารัด

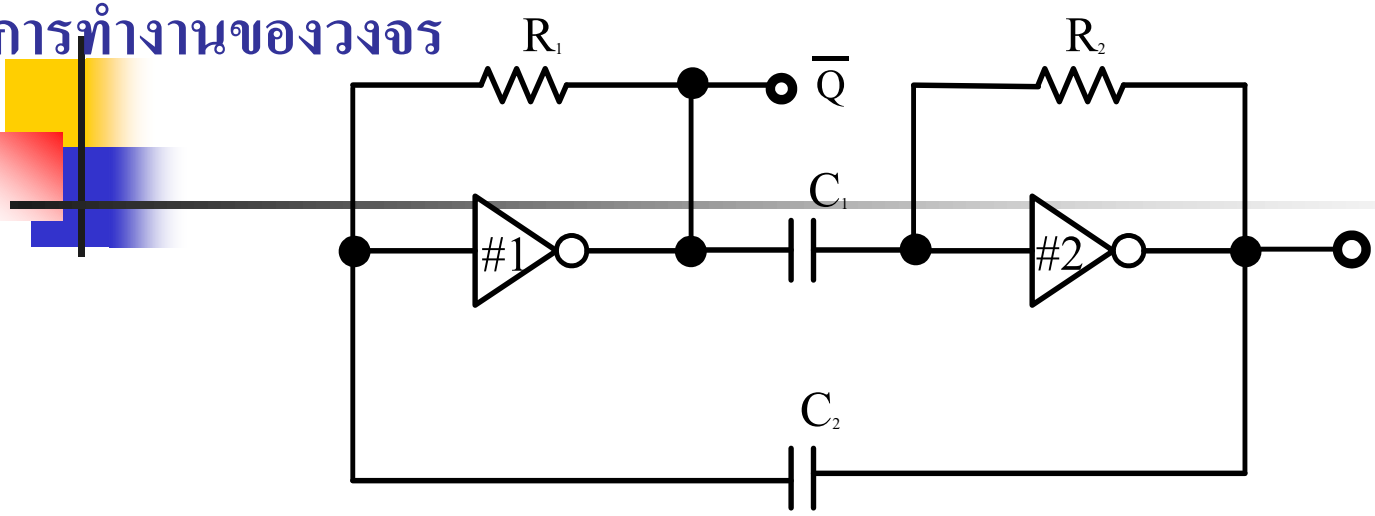
**จงหาความถี่ของวงจร**

**วิธีทำ** $f = \frac{0.7}{RC} = \frac{0.7}{10 \times 10^3 \times 120 \times 10^{-12}} = 0.584 \times 10^6$

$= 584$ **กิโลเฮิรตซ์**

# 622 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีทีทีแอล

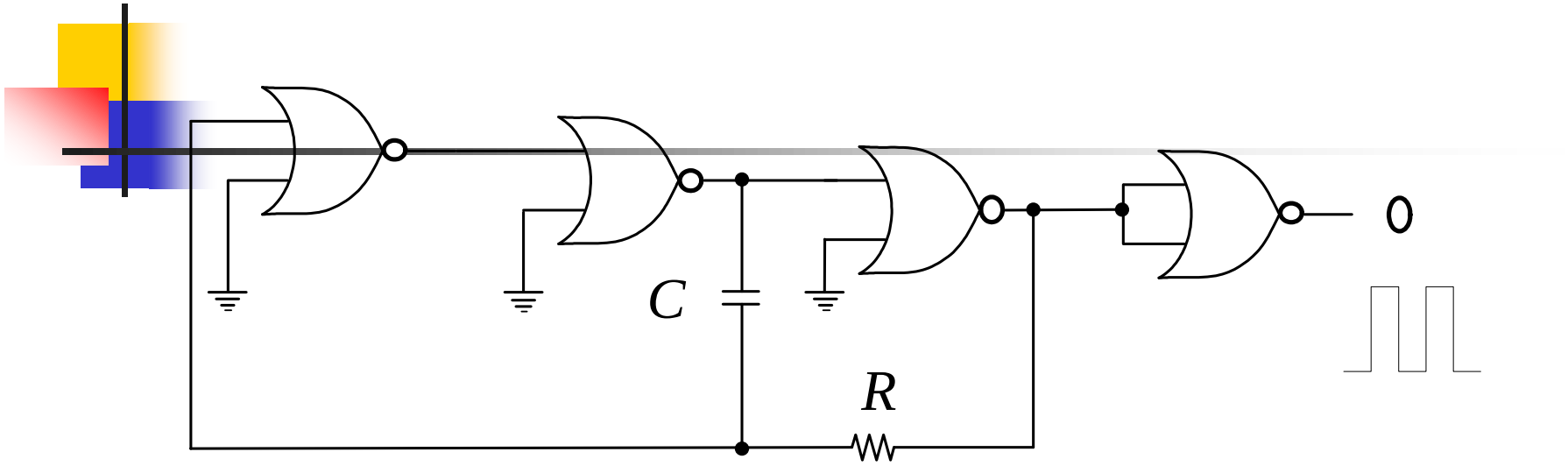
## การทำงานของวงจร




เมื่อจ่ายไฟให้กับวงจร อินเวอร์เตอร์ตัวหนึ่งจะทำงานเป็นวงจรชิงเกิลช็อทจะถูกทริกด้วยอีกตัวหนึ่งหลังจากเก็บประจุ หรือคายประจุ และจะทำให้เอาต์พุตอินเวอร์เตอร์ตัวที่ 2 เปลี่ยนสถานะซึ่งจะผลัดกันทริกไปตลอดเวลาสามารถกำหนดค่าระยะเวลาคายประจุโดยคำณวนจากค่า R และC ดังสูตร

$$f = \frac{0.7}{RC} \qquad R_1 = R_2 , C_1 = C_2$$

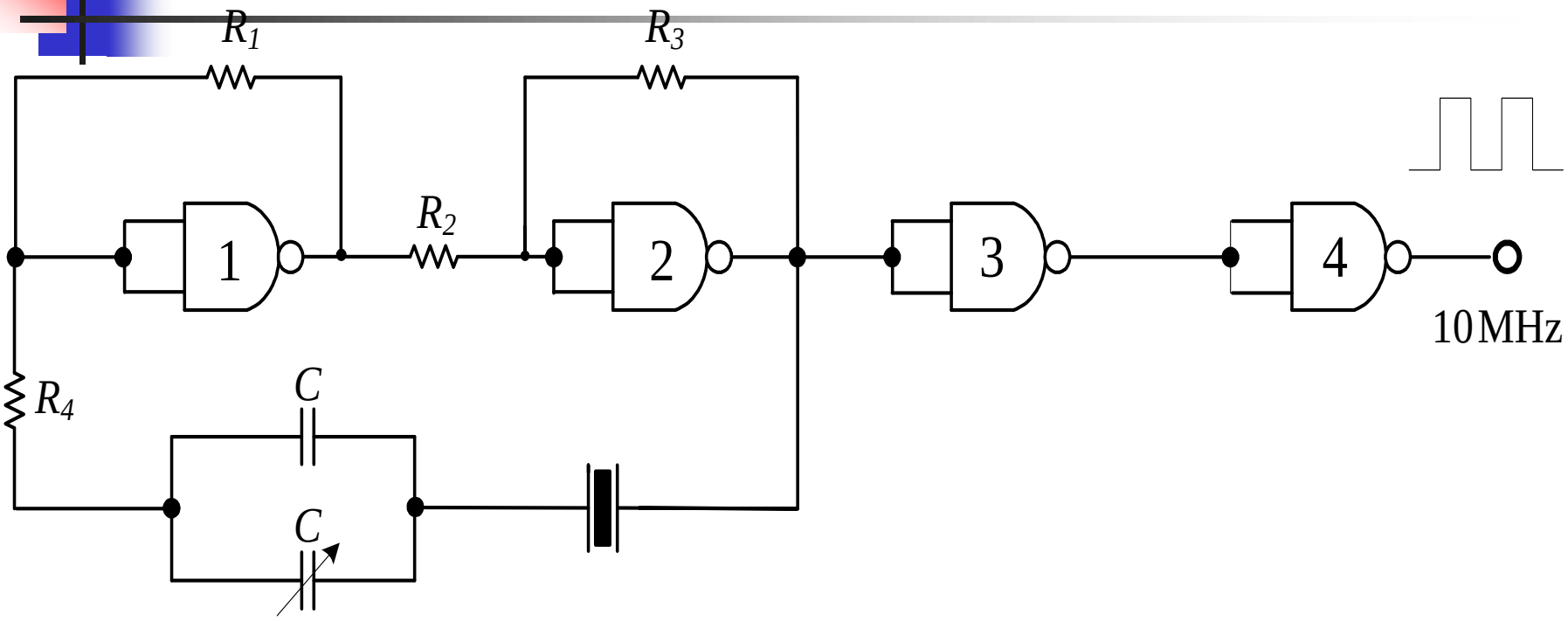
# วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้นอร์เกต




วงจรกำเนิดความถี่ที่นำเอานอร์เกตมาทำงานร่วมกับ R,C ซึ่งเป็นที่นิยมนำไปใช้ในวงจรดิจิตอลมาก สามารถกำเนิดความถี่โดยหาจากสูตร

$$f = \frac{0.455}{RC}$$

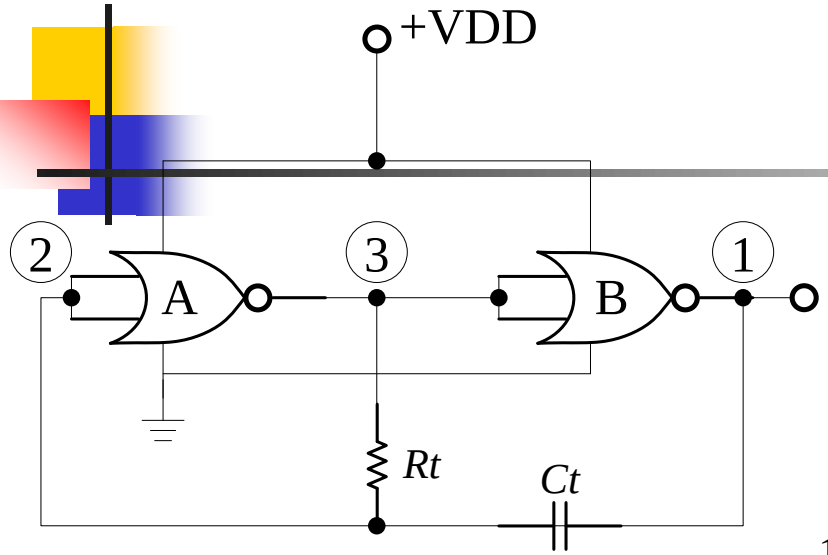
# 623 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้คริสตอลควบคุมความถี่

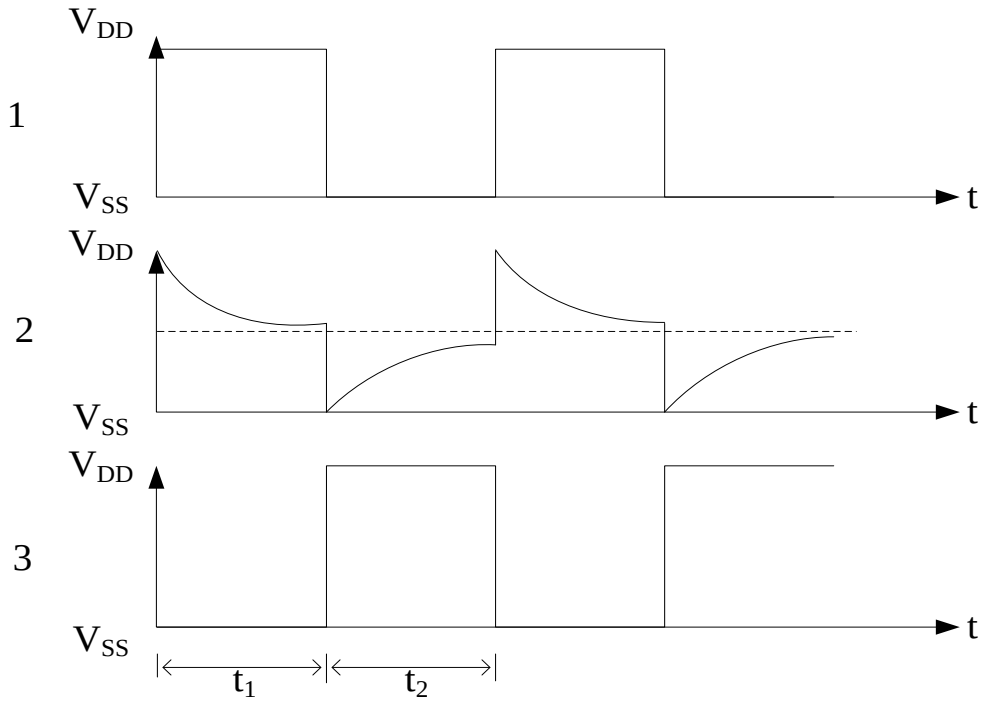


# อธิบาย

- วงจรนี้กำเนิดความถี่สูงที่ต้องการความแน่นอนของสัญญาณ คล็อกเพื่อไปใช้เป็นฐานเวลา โดยการใช้คริสตอลมาควบคุมค่ารีโซแนนซ์
- หลักการทำงานใช้แนนด์เกต 1,2 ทำการกลับสัญญาณทางอินพุตให้มีการป้อนกลับในลักษณะทางเสริม( Positive Feedback)
- ความสามารถของวงจรนี้มีเสถียรภาพต่อความถี่มาก โดยการต่อคาปาซิเตอร์ขนาด 1.7-14 PF อนุกรมเพื่อป้อนกลับให้มีค่ารีโซแนนด์มากที่สุด
- การทำงานของเกต3,4 ทำหน้าที่ปรับรูปสัญญาณทางเอาต์พุตให้เหมาะกับการใช้ในวงจรดิจิตอลคือค่า Risetime และFalltime น้อยที่สุด

# วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีที ที แอล นอร์เกต



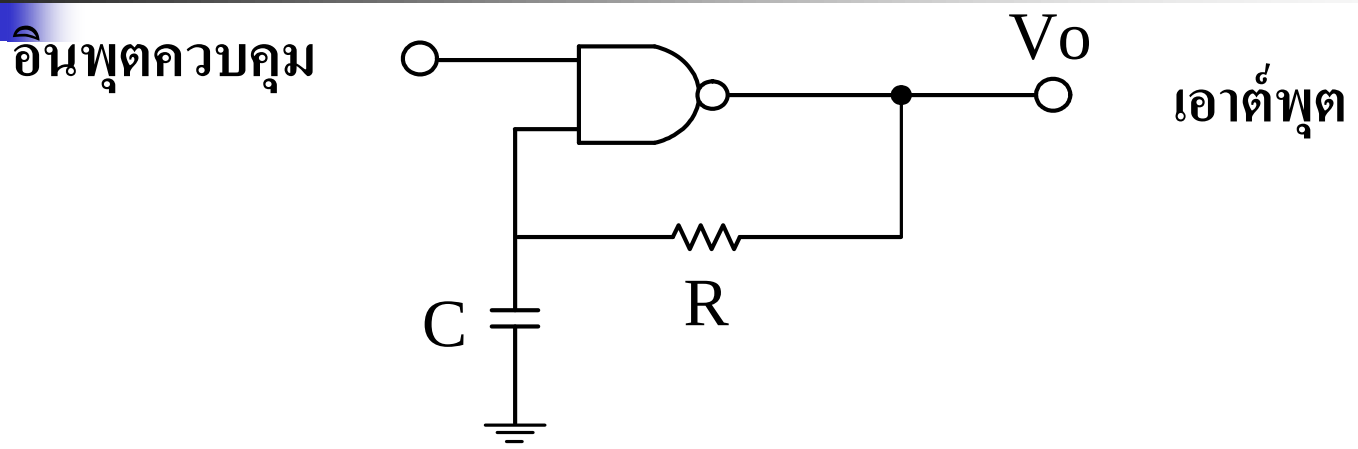



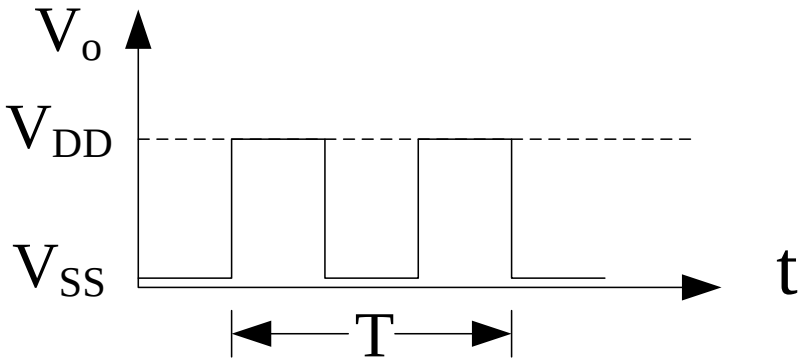
# อธิบาย

- วงจรประกอบด้วยเกต 2 ตัว และค่าRt, Ct
- จากจุด 1 เป็นลอจิก 1 Ct จะคัปปลิ้งค่าระดับ 1 ส่งไปยังจุด 2
- ที่จุด 3 จะถูกนอร์เกตกลับสัญญาณเป็น 0 คาปาซิเตอร์ที่ต่อกับจุด 3 ต่อผ่าน Rt ดังนั้นค่าแรงดันที่จุด 2 จะถูกปรับเข้าหาจุด 3
- โดยจุด 2 จะมีค่าเป็น 0 ทำให้เกต A เปลี่ยนสถานะที่เรียกว่า จุดส่งถ่ายแรงดัน(Transfer Voltage Point)
- เมื่อจุด 2 มีค่าเป็น 0 เอาต์พุตที่จุด 3 เป็น 1 เอาต์พุตที่จุด1 เป็น 0 ค่าที่ประจุอยู่ใน Ct จะคายประจุทำให้ศักย์แรงดันสูงขึ้นเรื่อยๆ
- การทำงานจะวนเวียนอย่างนี้ตลอด

# 624 วงจรอะสเตเบิลมัลติไวเบรเตอร์แบบใช้ไอซีซีมอส

## วงจรอะสเตเบิลมัลติไวเบรเตอร์ใช้ไอซีเบอร์ 4093



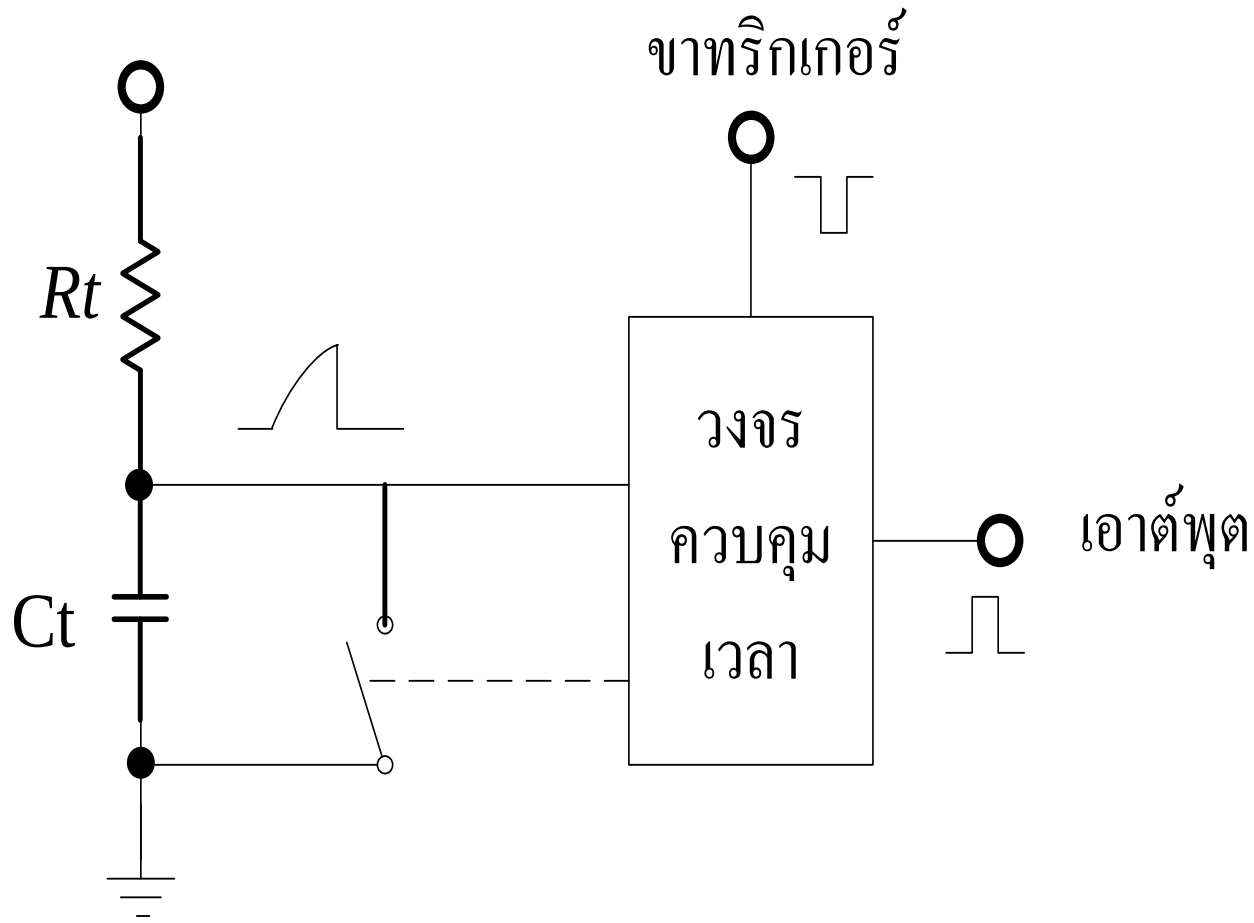
# อธิบาย

- วงจรอะสเตเบิลมัลติไวเบรเตอร์อาจจะออกแบบง่ายๆ โดยใช้วงจรไอซีซีมอสเบอร์ 4093 ต่อลักษณะการป้อนกลับ RC Time constant ก็จะสามารถกำเนิดพัลส์รูปสี่เหลี่ยม
- ค่าความต้านทานที่ใช้อยู่ระหว่าง 50 KOhm – 1MOhm
- ค่าคาปาซิเตอร์จะใช้ประมาณ 100 PF-1μFD
- ค่าความถี่สามารถหาจาก

$$T \cong 2RC\ In\left(\frac{V_p}{V_n}\right),\ f = \frac{1}{T}$$

- $V_p$ = แรงดันทริกเกอร์ด้านสูง
- $V_n$ = แรงดันทริกเกอร์ด้านต่ำ

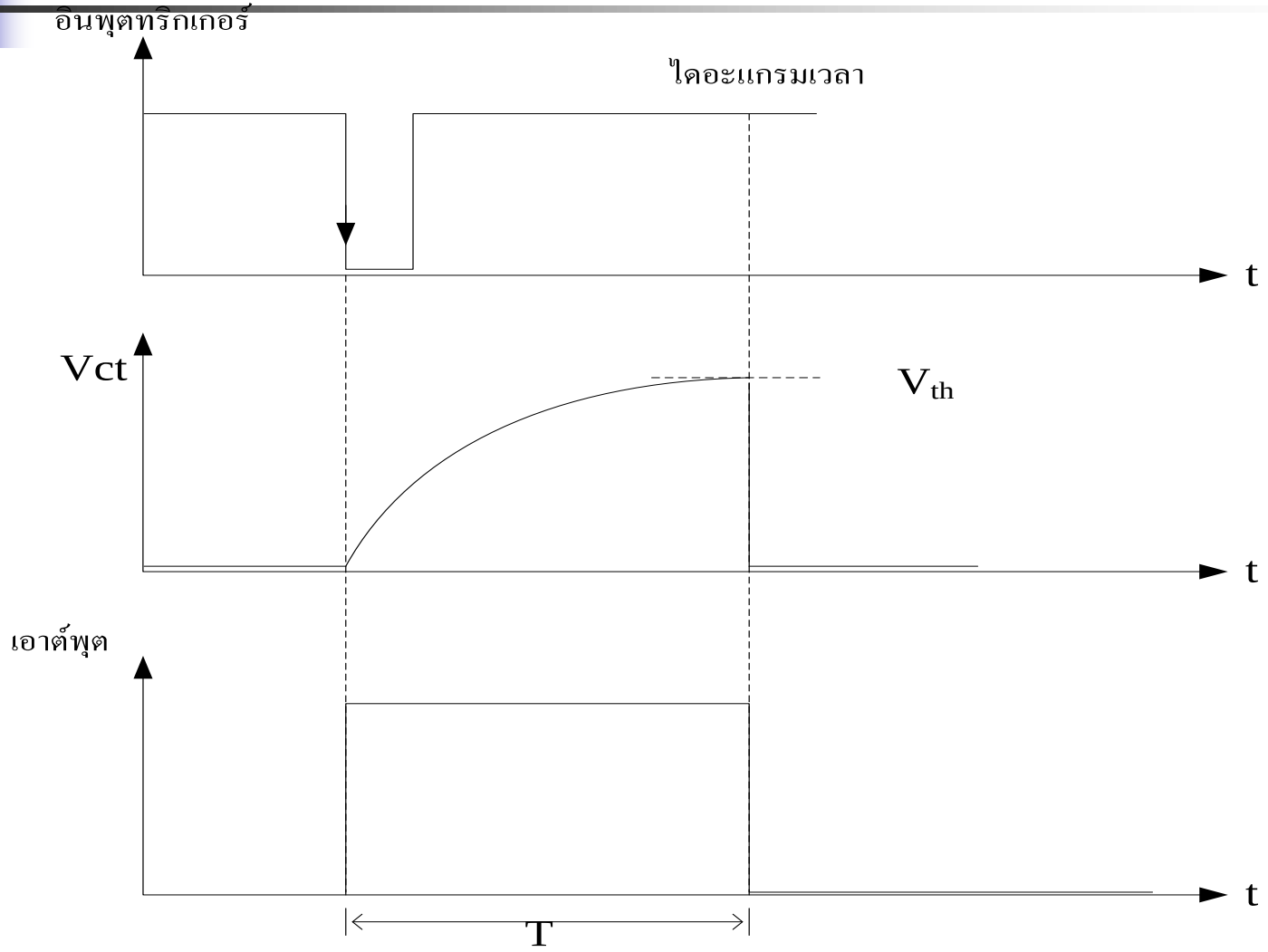
# โมโนสเตเบิลอาร์ซีไทม์เมอร์ (Monostable RC Time)



# อธิบาย

- จากรูปวงจรประกอบด้วยส่วนสำคัญ4 ส่วนคือ Rt ,Ct ,สวิทซ์ ,วงจรควบคุม
- สภาวะปกติ เอาต์พุตมีค่าเป็น 0 สวิทซ์จะทำงานต่อวงจรด้วยการทำงานของวงจรควบคุมเวลา
- สภาวะเมื่อมีการทริกที่วงจรควบคุมเวลา สวิทซ์จะหยุดการทำงานเปิดให้มีการประจุผ่านRt ค่าแรงดันทางเอาต์พุตของวงจรเป็น1
- สภาวะที่สวิทซ์เปิด Ct ประจุได้ค่าเป็น 2/3 +Vcc จะบังคับการทำงานของวงจรควบคุมให้มีการรีเซตทางเอาต์พุตเป็น 0 เหมือนเดิม

# Timing Diagram Monostable RC Time

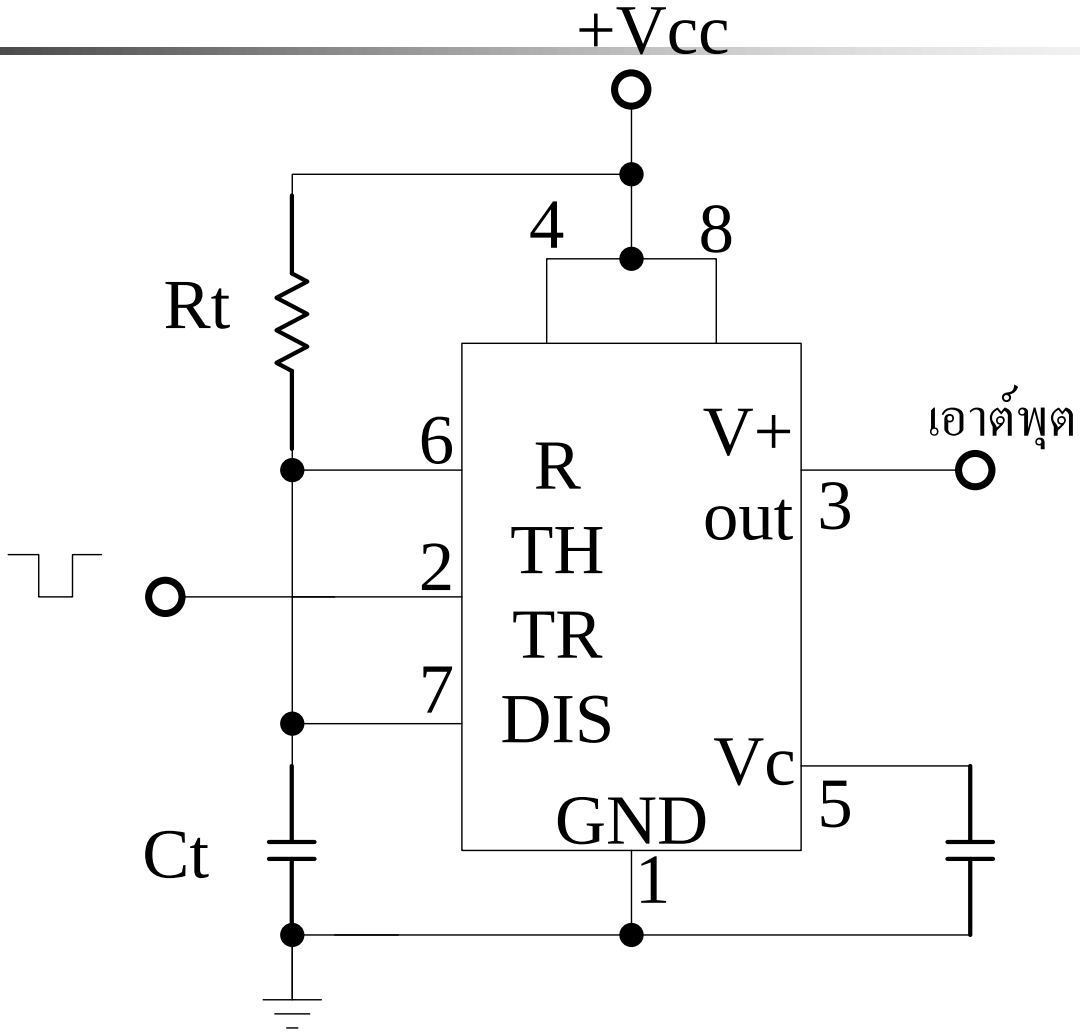


# อธิบาย

- สามารถควบคุมการทำงานโดยคำนวณจากสูตร
- Vi คือค่าแรงดันคล่อม Ct ขณะก่อนเริ่มประจุแรงดันในช่วงเวลาคงที่
- Vth ค่าแรงดันเทรชโฮลด์ ที่บังคับให้วงจรกลับสู่สถานะรีเซต
- T คือค่าช่วงเวลาเอาต์พุตมีสถานะไม่คงที่
- Ct ค่าคาปาซิเตอร์ที่ใช้ประจุแรงดันจนถึงค่า Vth
- Rt ค่าความต้านทานในการประจุแรงดันทำงานร่วมกับCt
- Vcc แหล่งจ่ายแรงดันให้กับวงจร

# 63 การนำไอซี 555ไปใช้งานในรูปแบบต่างๆ

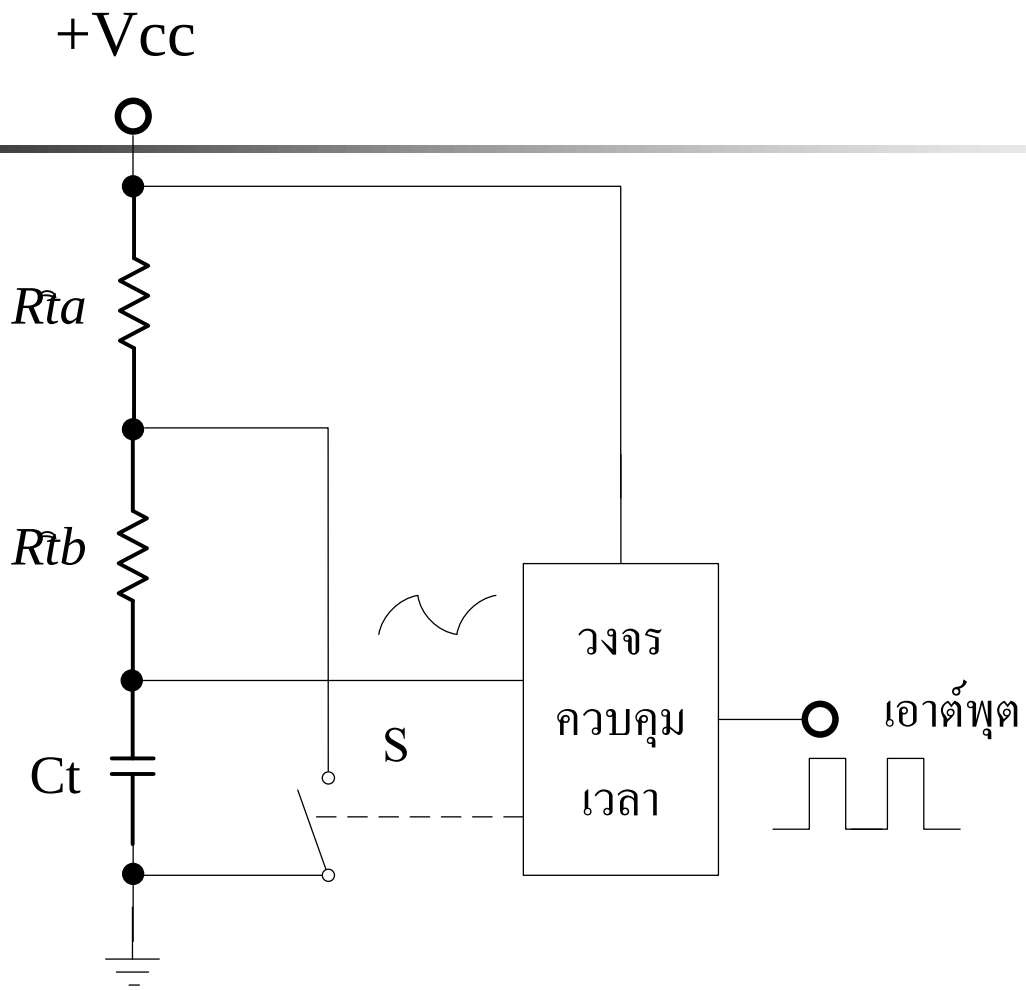
## 631 วงจรโมโนสเตเบิลใช้ไอซีเบอร์ 555



# อธิบาย

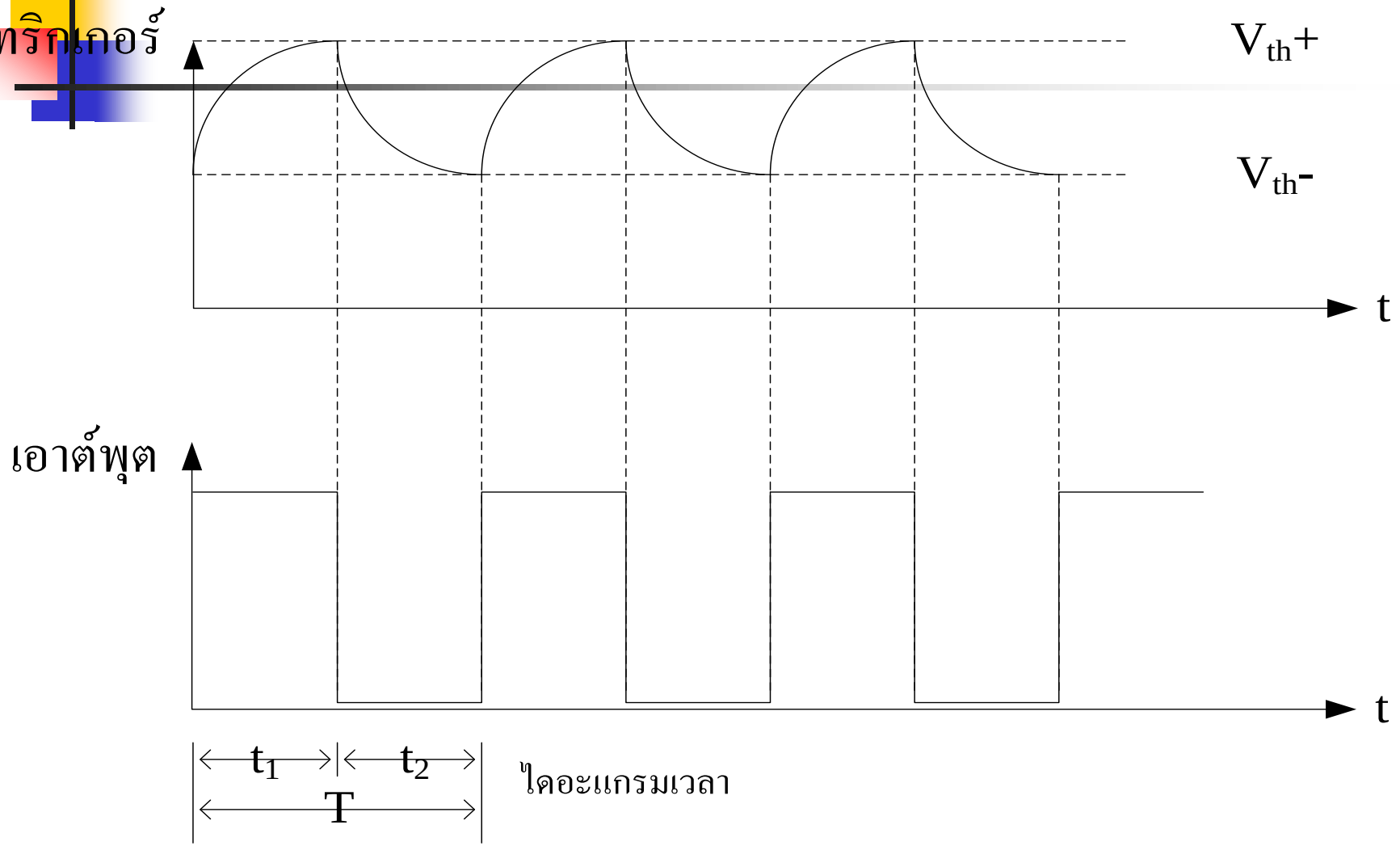
- การนำวงจรโมโนสเตเบิลที่กล่าวมา นำไอซีสำเร็จรูปเบอร์ 555 เป็นวงจรควบคุมเวลา
- ให้นักศึกษาไปหารายละเอียดภายในของวงจร 555 ประกอบการพิจารณาเพื่อให้เข้าใจถ่องแท้
- ขา 2 เป็นขาควบคุมการทริกทางอินพุต
- ขา 7 กับขา 1 ทำหน้าที่เป็นวงจรสวิทซ์อัตโนมัติต่อลงกราวนด์
- ขา 6 เป็นขาอินพุตที่นำค่าแรงดันที่เกิดจากการประจุใน Ct มาควบคุม

# วงจรอะสเตเบิลอาร์ซีไทม์เมอร์




การทำงานของวงจรอะสเตเบิลโดยใช้ RC Timer

# Timing Diagram อะสเตเบิลอาร์ซีไทม์เมอร์



# อธิบาย

- วงจรนี้ทำงานโดย S เปิดและแรงดันเอาต์พุตเป็นลอจิก 1 Ct จะเริ่มประจุแรงดันผ่าน Rta และ Rtb จนแรงดันถึง Vth+ ทำให้เอาต์พุตเป็นลอจิก 0 และเมื่อแรงดันถึง Vth เอาต์พุตจะเปลี่ยนสถานะจากลอจิก 0 เป็น 1 วงจรนี้จะทำงานสลับกันไปมา

- ค่าการควบคุมการทำงานเป็น ไปตามสมการ

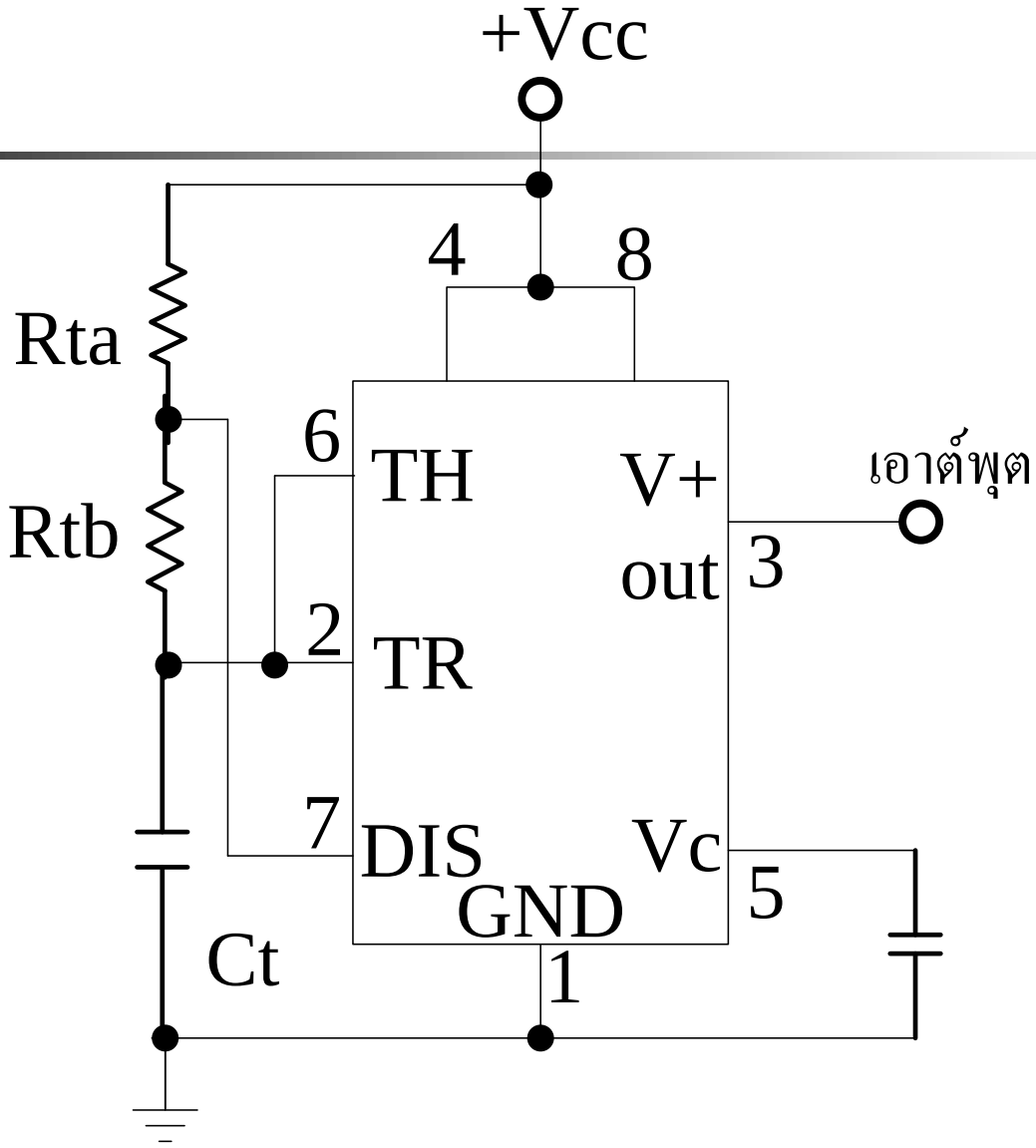
Vc คือ แรงดันตกคร่อม Ct

Vi คือ แรงดันเริ่มประจุของ Ct

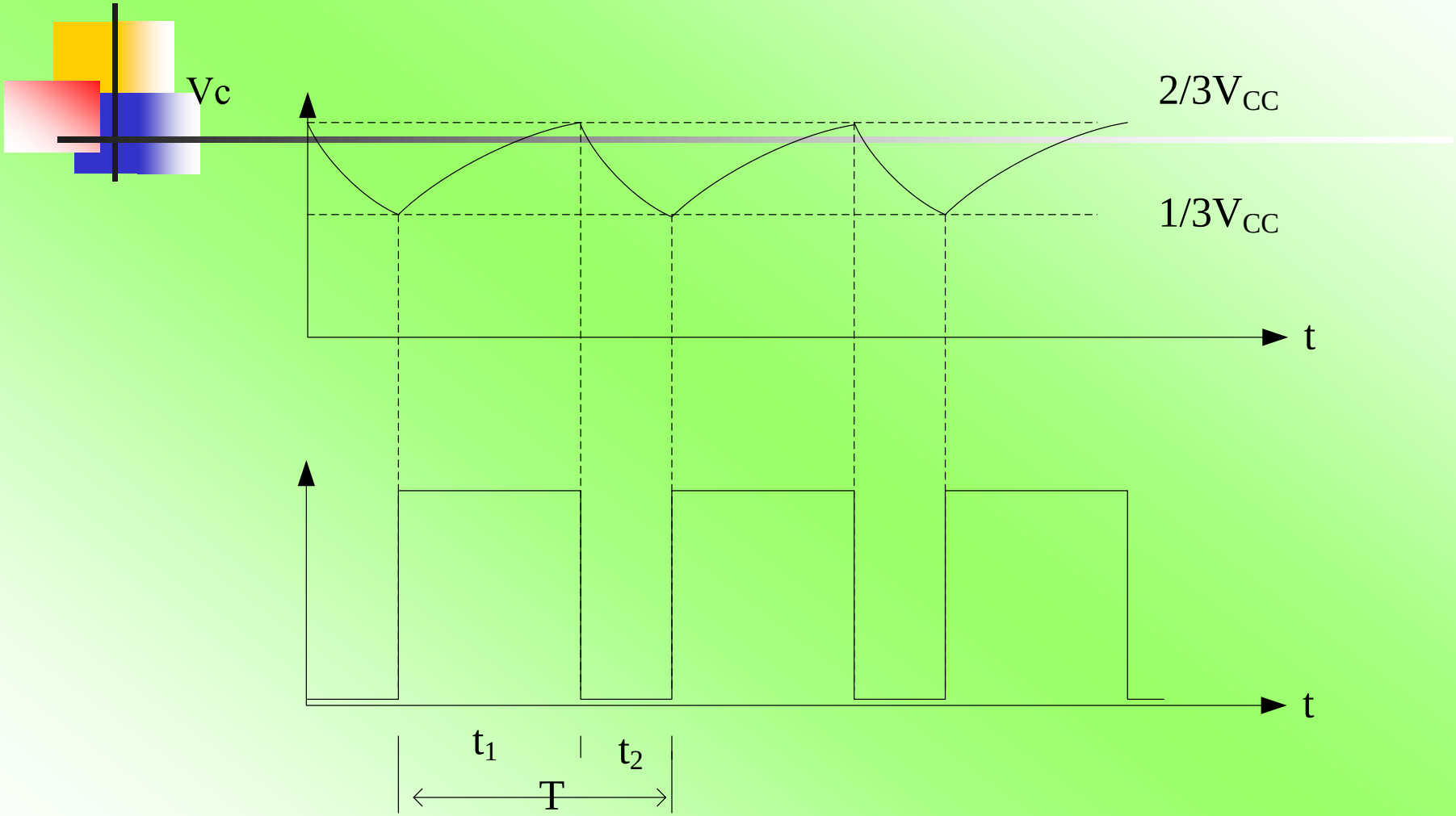
Vth คือ แรงดันควบคุมระดับทริกของวงจร

$$t = RtCt \ln\left(\frac{Vc - vi}{Vc - Vth}\right)$$

# 632วงจรอะสเตเบิลใช้ไอซีเบอร์ 555



# Timing Diagram วงจรอะสเตเบิลใช้ไอซีเบอร์ 555



# อธิบาย

$$t_1 = 0.693(R_{ta} + R_{tb})C_t$$

$$t2 = 0.693R_{tb}Ct$$

$$f = 1.44/(R_{ta} + 2R_{tb})C_t$$

$$DF_{t1} = (R_{ta} + R_{tb})/(R_{ta} + 2R_{tb})$$

$$DF_{t2} = R_{tb}/(R_{ta} + 2R_{tb})$$

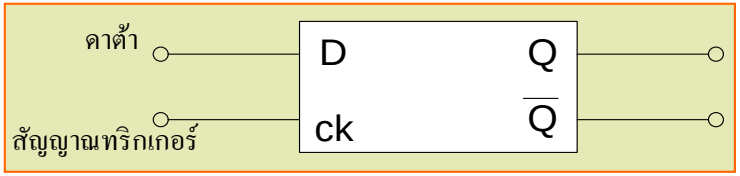
# อธิบาย(ต่อ)

- ขาอินพุตวงจรเปรียบเทียบแรงดัน 2 ชุด ต่อร่วมกัน ขา 2, 6
- $C_t$ ประจุแรงดันเพิ่มขึ้นเป็น 2/3 ของแหล่งจ่าย
- เอาต์พุตขา3 จะเป็นลอจิก 0
- $C_t$ จะคายประจุผ่านทาง $R_{tb}$ และผ่านขา 7 แรงดันจะลดลงเหลือ 1/3 ของแหล่งจ่าย จะบังคับให้ขา 3ทางเอาต์พุตเป็นลอจิก 1
- $C_t$ ก็จะเริ่มประจุแรงดันเพิ่มขึ้นเป็น 2/3 ของแหล่งจ่ายเป็นครั้งที่ 2 และก็จะทำงานเวียนไปเช่นนี้ตลอด
- หากต้องการควบคุมการทำงานจะควบคุมที่ขา 4

# 64 การทำงานของวงจรไบสเตเบิล

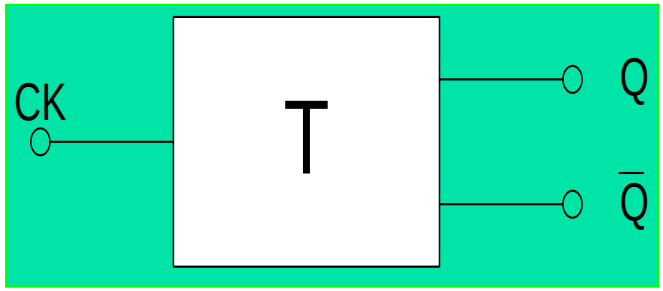
## ฟลิบฟลอบ(Flip-Flop)ชนิดต่างๆ

### D flip-flop

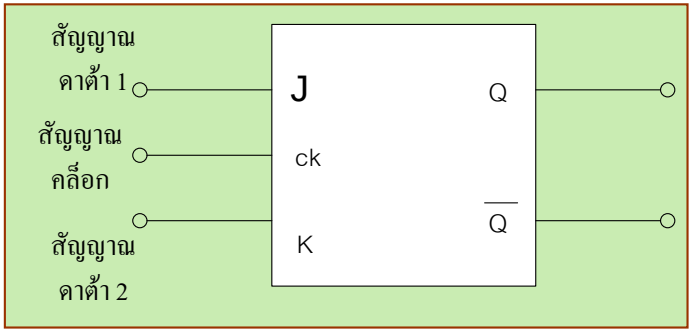



### RS flip-flop

### T flip-flop



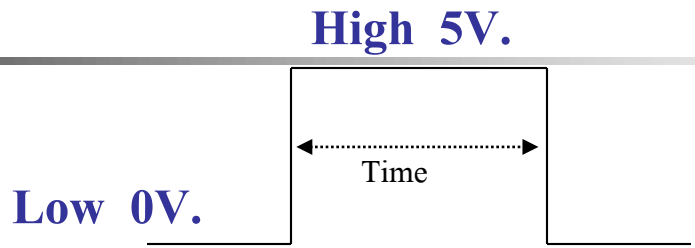

### J-K flip-flop



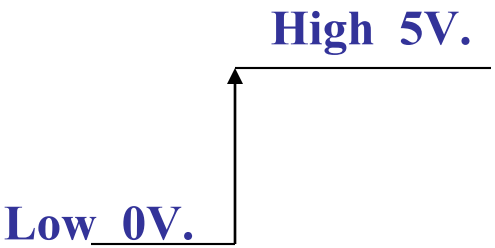
# อธิบาย

- ฟลิบฟลอบที่ใช้ในระบบดิจิตอลมีการทำงาน 4 รูปแบบใหญ่ๆ ตามการทำงานคือ
- D-Flip Flop ใช้ในการเก็บข้อมูล และส่งข้อมูล
- R-S Flip Flop ใช้ทำเป็นสวิทซ์ควบคุมการทำงานทางเอาต์พุต, เก็บข้อมูลใช้สองสภาวะ =Set , Reset
- T Flip Flopใช้ทำหน้าที่หารข้อมูลทางอินพุตหรือเก็บข้อมูล
- J-K Flip Flop นำคุณสมบัติของJ-K กับT Flip Flopมารวมกันแล้วทำหน้าที่คลอบคลุม Flip Flop ทั้งสอง

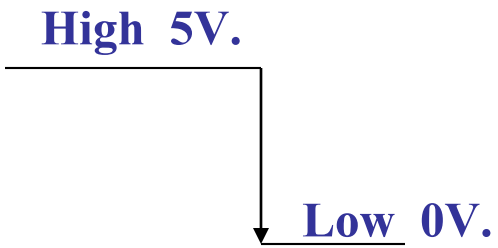
# รูปคลื่น Clock pulse




(ก) Clock pulse




(ข) Leading edge




(ค) Trailing edge

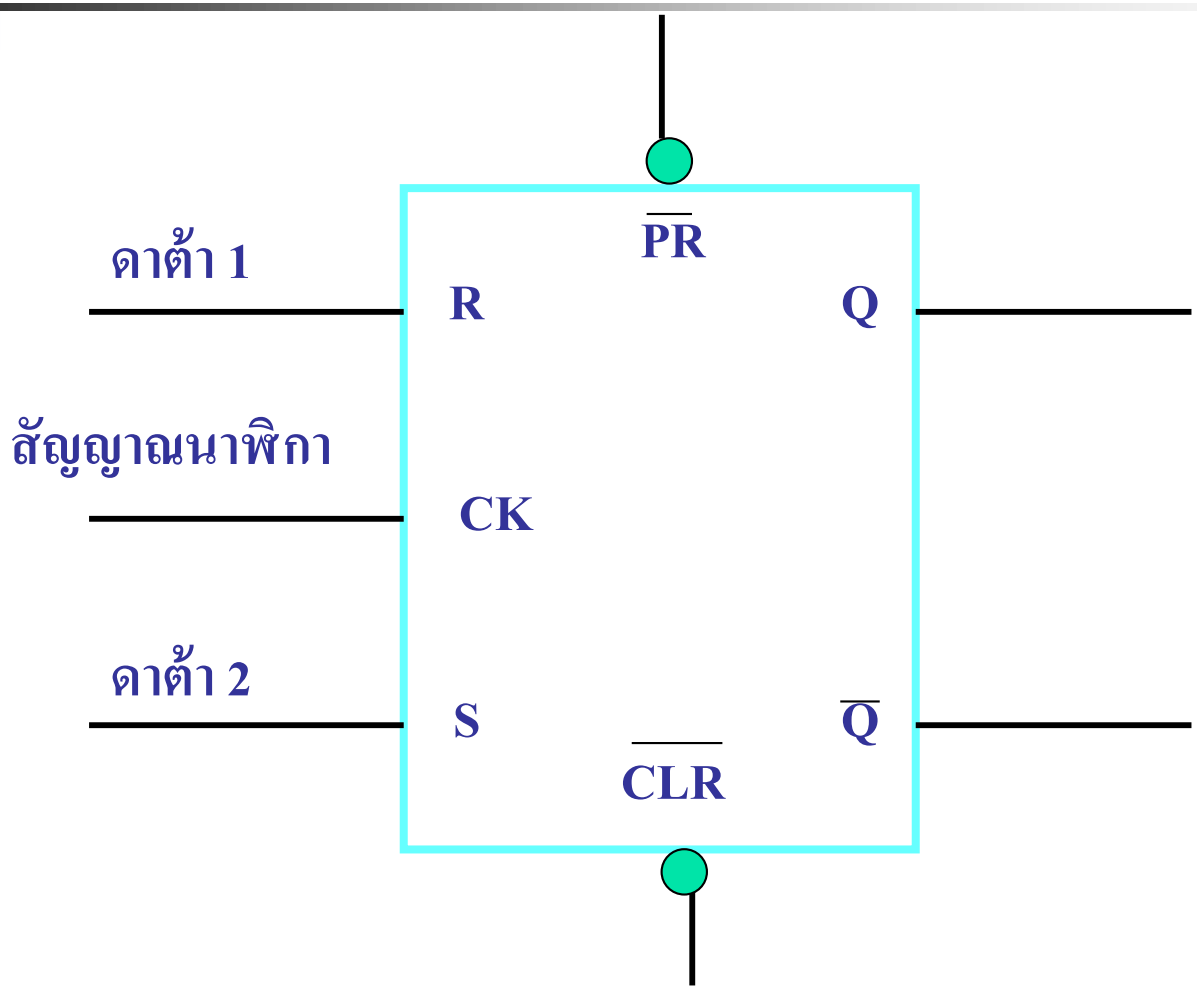
# อธิบาย

รูปสัญญาณที่ใช้ในระบบดิจิตอล จะมีการเปลี่ยนแปลงจาก 0 โวลต์ ไปเป็น 5 โวลต์และคงที่อยู่เป็นช่วงเวลา Time จากนั้นก็จะมีการเปลี่ยนแปลงจากระดับสูงมาระดับต่ำ การเปลี่ยนแปลงนี้จะวนซ้ำกันตลอดสามารถพิจารณาเป็น 2 ส่วนคือ

**ส่วนที่1** ค่าระดับแรงดัน 0โวลต์(Low) เปลี่ยนไปยัง 5 โวลต์ (High) ช่วงนี้เรียกว่าขอบขาขึ้น (Leading Edge) หรือ Active High ดังรูป (ข)

**ส่วนที่2** ค่าระดับแรงดัน 5 โวลต์ เปลี่ยนไปยัง 0โวลต์ช่วงนี้เรียกว่าขอบขาลง(Trailing Edge) หรือ Active Low ดังรูป(ค)

# R – S ฟลิปฟลอป


$\overline{PR}$
ดาต้า 1
R
Q
สัญญาณนาฬิกา
CK
ดาต้า 2
S
$\overline{Q}$
$\overline{CLR}$

# อธิบาย

- หลักการเบื้องต้นของ R-S Flip Flop
- สัญญาณอินพุตประกอบด้วย 3ส่วน สัญญาณควบคุมประกอบด้วย2ส่วน
- Data1 คือสัญญาณควบคุมR (Reset) ทางเอาต์พุต
- Data2 คือสัญญาณควบคุมS (Set) ทางเอาต์พุต
- Ck (Clock) ทำหน้าที่ควบคุมการทำงานเมื่อต่อเข้ากับระบบดิจิตอลให้ทำงานพร้อมกันทั้งระบบ เพื่อให้ระบบมีการทำงานที่สัมพันธ์กันทั้งสาม อินพุตจากรูปวงจรจะทำงานที่ขอบขาขึ้น

# อธิบาย

## ตารางการทำงานของPreset กับClear

| ลำดับควบคุม | Preset $\overline{PR}$ | Clear $\overline{CLR}$ | Reset | Set | Q | $\overline{Q}$ |
|---|---|---|---|---|---|---|
| 1 | 0 | 1 | x | x | 1 | 0 |
| 2 | 1 | 0 | x | x | 0 | 1 |
| 3 | 1 | 1 | x | x | x | x |

# อธิบาย(ต่อ)

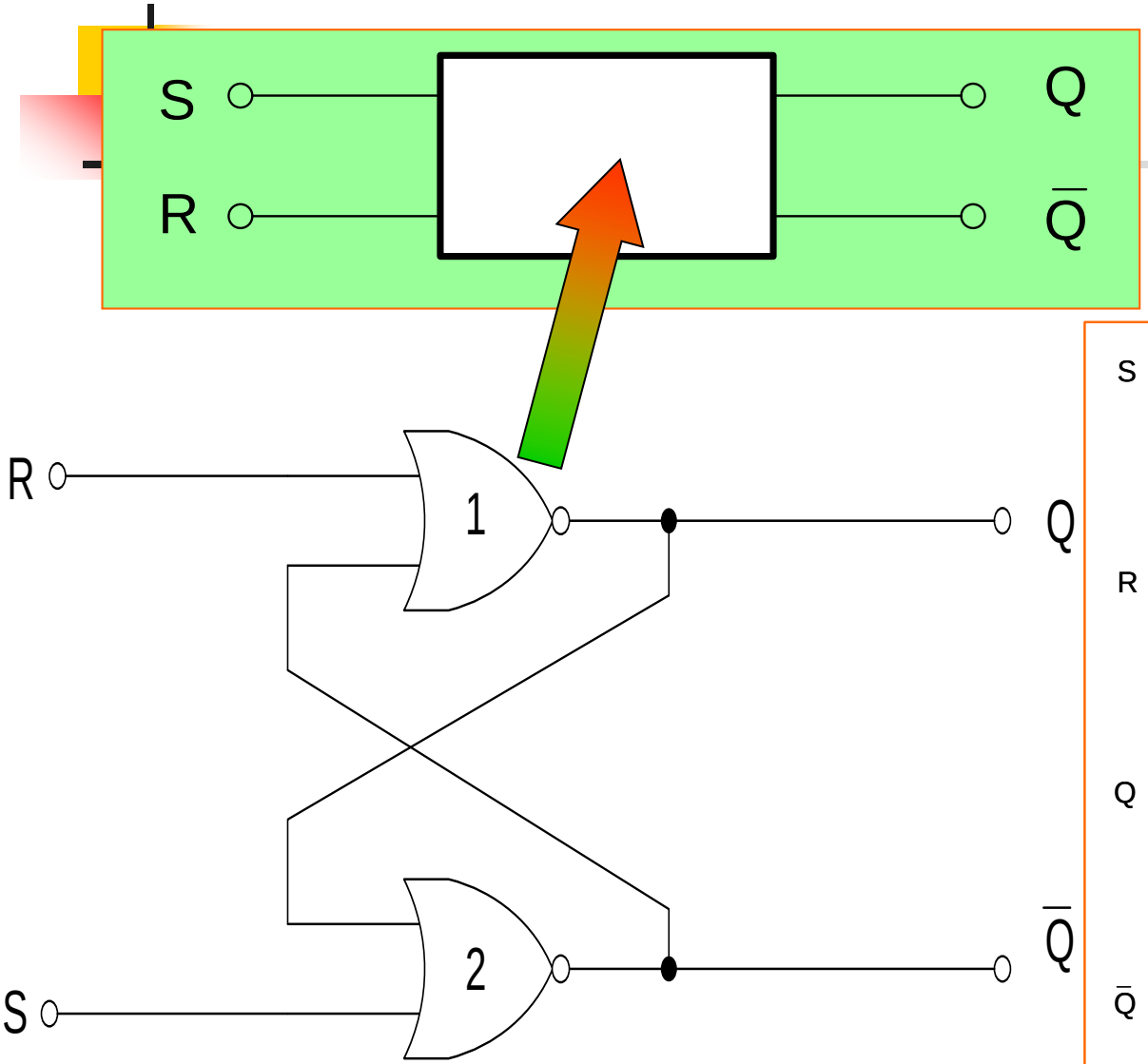
**PR = Preset** เป็นขั้วที่ทำหน้าที่ควบคุมการทำงานของเอาต์พุตโดยตรงให้มีค่าเท่ากับ1 ไม่ขึ้นอยู่กับคล็อก และข้อมูลทางอินพุตจะเป็นอย่างไร ค่าPR จะทำงานเมื่อสัญญาณจุดนี้เท่ากับ 0สังเกตุมีวงกลม(Bubble) ที่ขา PR , CLR

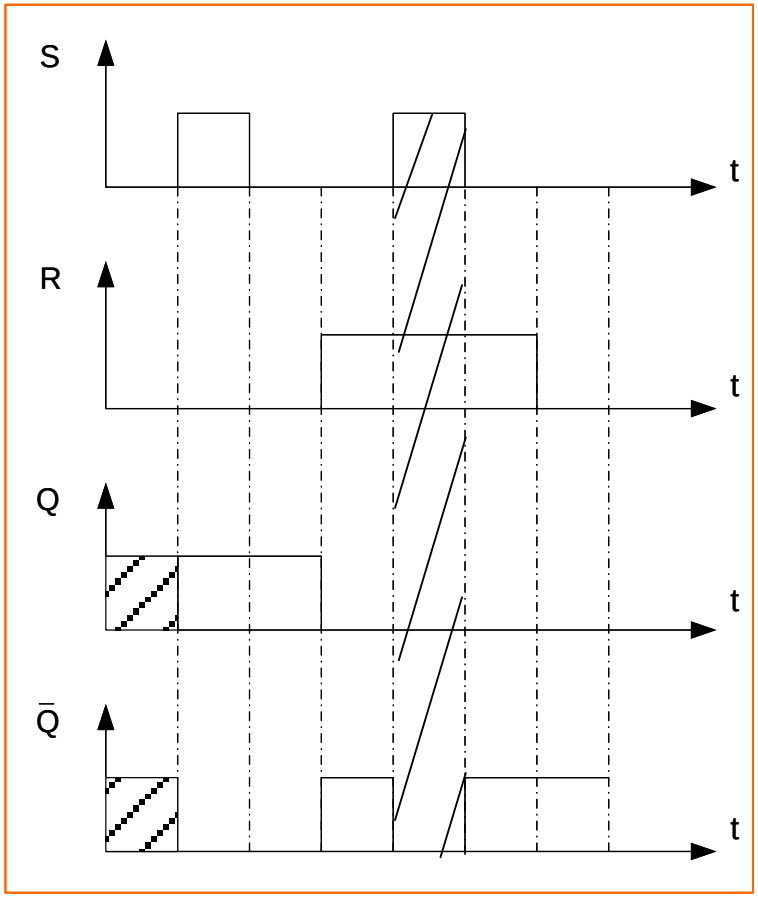
สภาวะปรกติจะต่อเข้ากับ1 ไว้เพื่อความมั่นใจว่า PR จะไม่ทำงานเมื่อไม่มีการควบคุมจริง

**CLR = Clear** เป็นขั้วที่ทำหน้าที่ควบคุมการทำงานของเอาต์พุตโดยตรงให้มีค่าเท่ากับ0 ไม่ขึ้นอยู่กับคล็อก และข้อมูลทางอินพุตจะเป็นอย่างไร ค่า CLR จะทำงานเมื่อสัญญาณควบคุมจุดนี้เป็น0

สภาวะปรกติจะต่อเข้ากับ1 ไว้เพื่อความมั่นใจCLR จะไม่ทำงานเมื่อไม่มีการควบคุมจริง

# 641 อาร์เอสแลตซ์ชนิดใช้นอร์แลตซ์ ( R-S Latch )



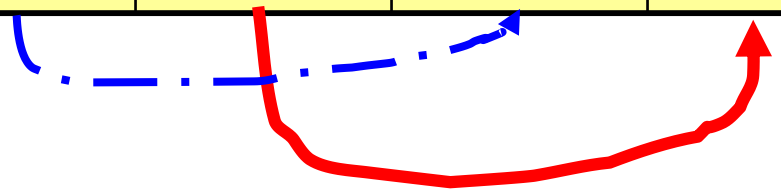



# อธิบาย

- หากนำนอร์เกต 2 อินพุต 2 ตัวมาต่อเข้าด้วยกัน โดยเอาต์พุตของเกตทั้งสองมาต่ออินพุตที่สลับกัน เรียกว่า อาร์เอสแลตซ์ ชนิดนอร์แลตซ์
- อินพุตที่เหลือคือขั้ว S จะเป็นเกตของนอร์เกต 2
- อินพุตที่เหลืออีกขั้ว R จะเป็นเกตของนอร์เกต 1
- สามารถเขียนเป็นสัญลักษณ์ได้ดังรูปบล็อก
- การทำงานของแลตซ์แบบนี้จะเป็นไปตามตารางที่กล่าวมา
- สามารถเขียนเป็นไทม์มิ่งไดอะแกรมดังแสดงด้านขวามือนักศึกษาลองพิจารณาใช้ตารางประกอบ

# ตารางอินพุต - เอาต์พุต

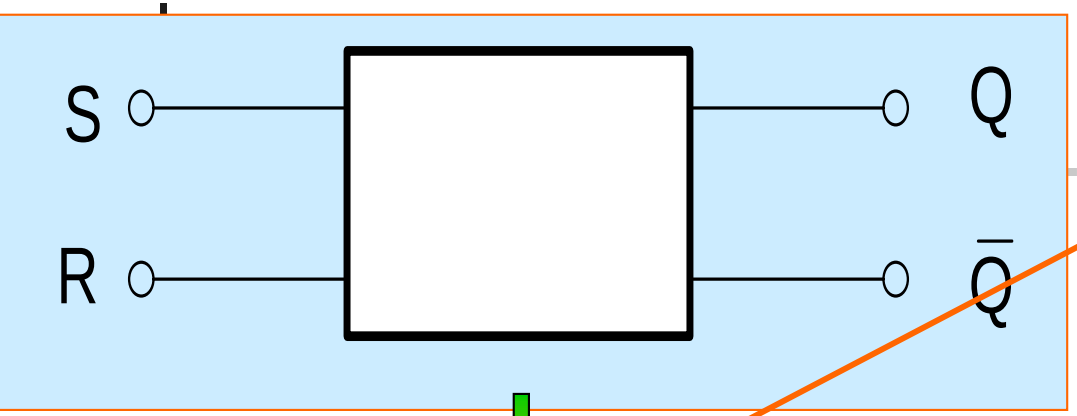
| ลำดับที่ | อินพุต | | เอาต์พุต | |
|---|---|---|---|---|
| | S | R | Q | $\overline{Q}$ |
| 1 | 0 | 0 | X | $\overline{X}$ |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 |
| 4 | 1 | 1 | 0 | 0 |

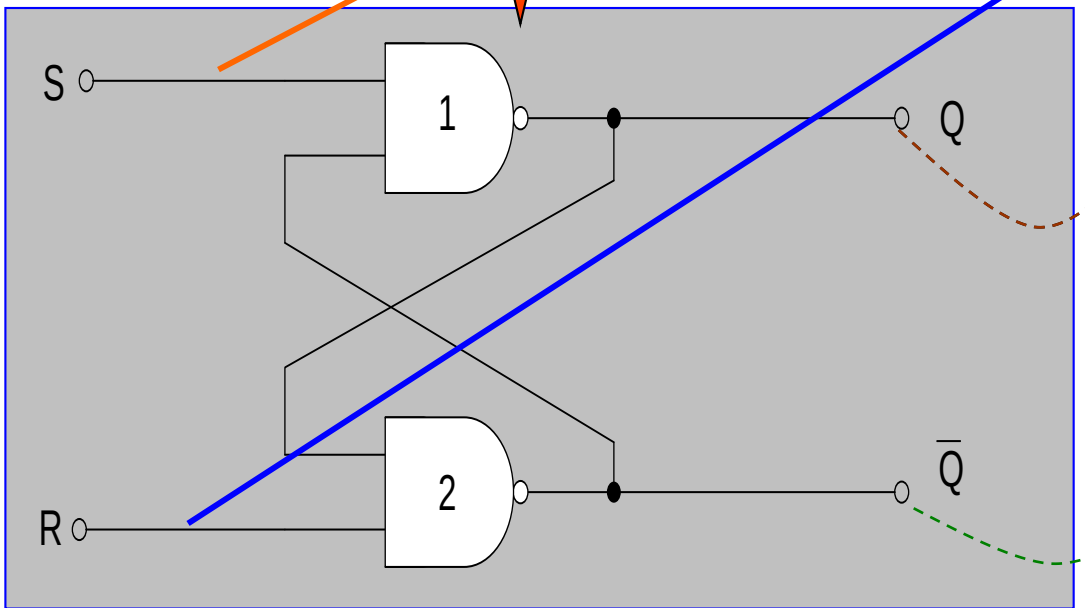


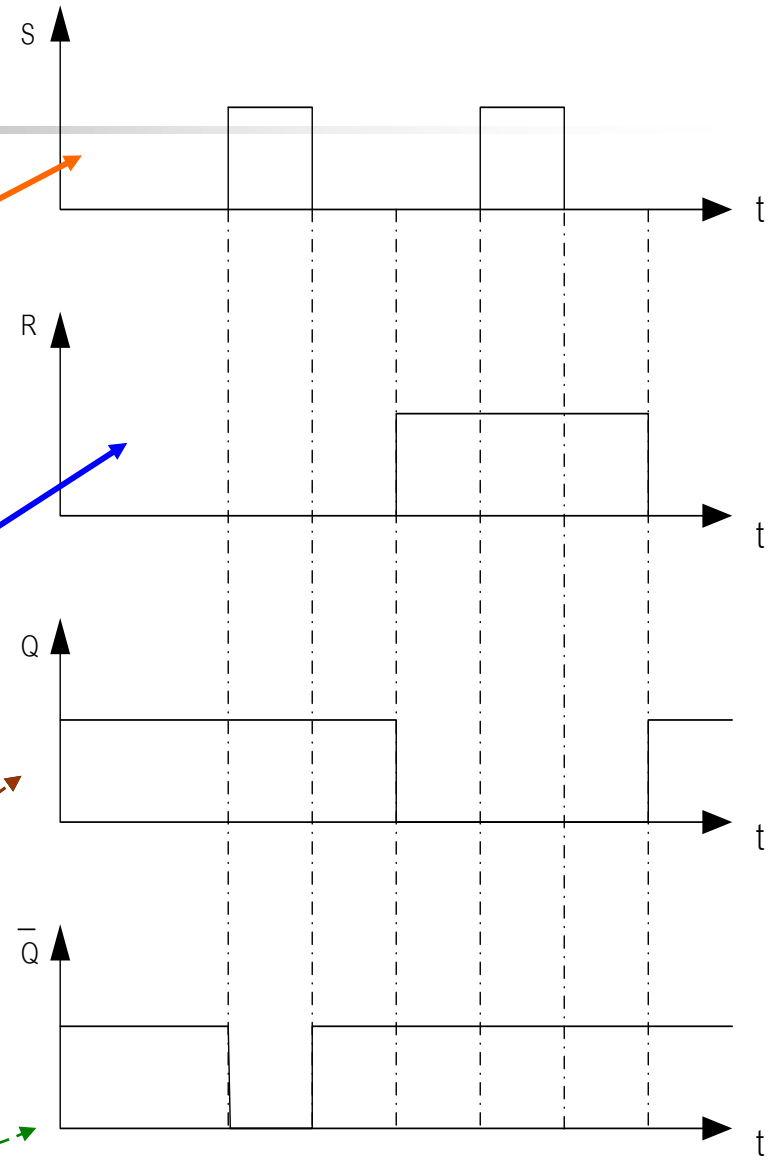
# อธิบาย

- จากตารางการทำงาน เมื่อ S = 0, R = 0, Q = x เอาต์พุตจะคงที่เหมือนเดิมไม่มีการเปลี่ยนแปลง คือเป็น 0 อยู่ก็คงที่ 0 ถ้าเป็น 1 อยู่ก็จะคงที่ 1 อยู่เหมือนเดิมส่วน Q จะตรงข้ามกับ $\bar{Q}$
- เมื่อS=0 ,R=1 สภาวะนี้เอาต์พุตจะเป็น Reset ค่า Q จะเป็นสภาวะอะไรอยู่ก็ตามจะถูกเปลี่ยนเป็น0
- เมื่อ S=1, R=0 สภาวะนี้เอาต์พุตจะเป็น Set ค่า Q จะเป็นสภาวะอะไรอยู่ก็ตามจะถูกเปลี่ยนเป็น1
- เมื่อ S=1, R=1 สภาวะนี้เอาต์พุตจะเป็น 0 ทั้งคู่ ทั้ง Q,$\bar{Q}$ สภาวะแบบนี้จะไม่มีการใช้งาน

# อาร์เอสแลตซ์ชนิดใช้แนนเกต ( R-S Latch )



# อธิบาย

- จากรูปนำR-Sแลตซ์ชนิดที่ใช้แนนด์เกตมาต่อจำนวน 2 ตัว
- $S = 0$ , $R = 0$ ทางด้านเอาต์พุต $Q = 0$ ,$\bar{Q} = 1$
- เป็นสภาวะที่ไม่ใช้งานเพราะวงจรไม่มีเสถียรภาพ ค่า Q , Q มีค่าเหมือนกันทั้งคู่
- $S = 0, R = 1$ ทางด้านเอาต์พุต $Q= 0, \bar{Q}=1$ สภาวะนี้เป็นสภาวะของ Reset
- $S = 1, R = 0$ ทางด้านเอาต์พุต $Q=1$ ,$\bar{Q}=0$ สภาวะนี้เป็นสภาวะของ Set
- $S = 1$ ,$R = 1$ ทางด้านเอาต์พุต $Q=X$ ,$\bar{Q}=\bar{X}$ สภาวะนี้เป็นสภาวะของ Store ค่าในเอาต์พุตทั้งสองไม่มีการเปลี่ยนแปลงใดๆ เมื่อมีการทำงาน

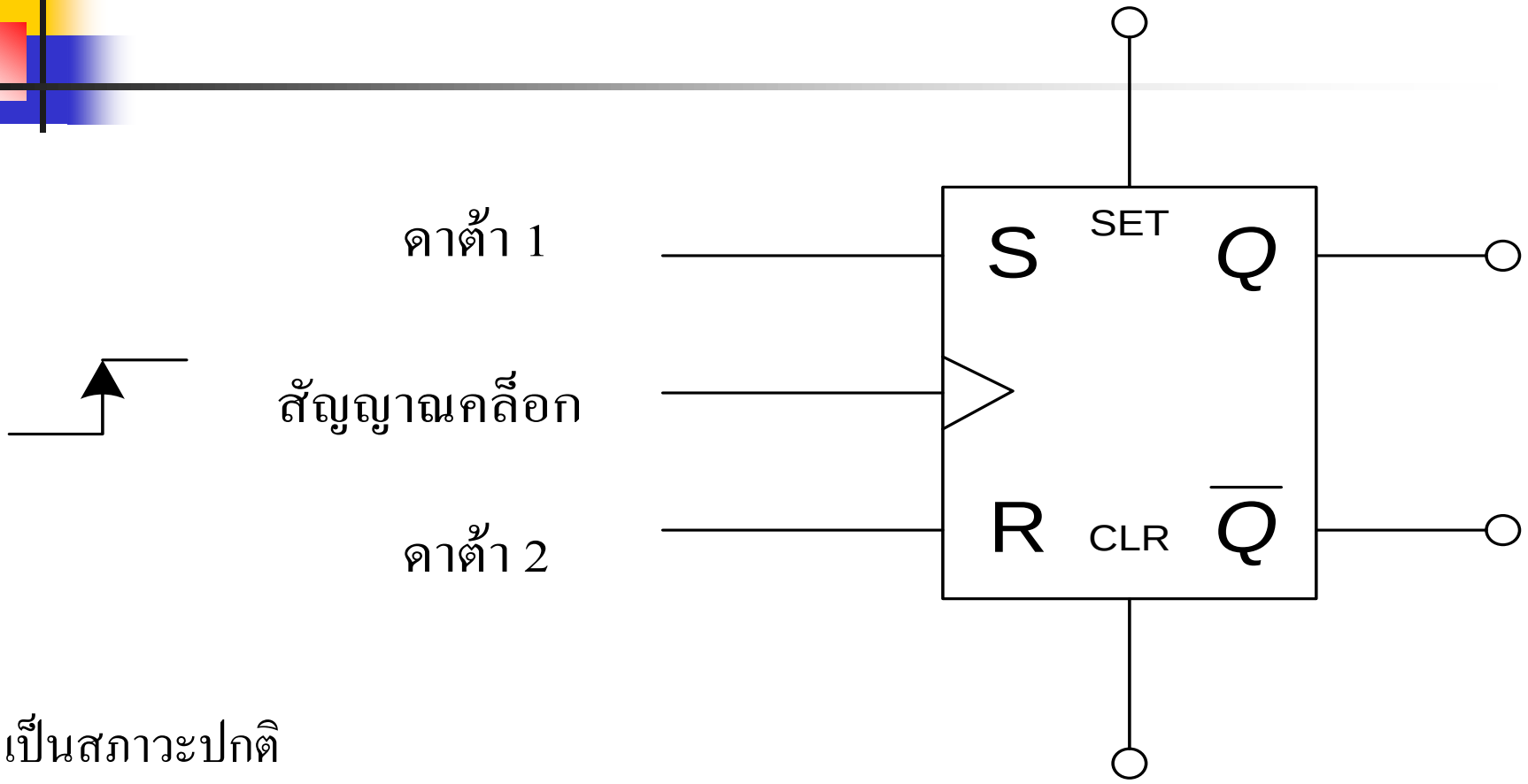
# ตาราง อินพุต - เอาต์พุต

| ลำดับที่ | อินพุต | | เอาต์พุต | |
|---|---|---|---|---|
| | S | R | Q | $\overline{Q}$ |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 |
| 4 | 1 | 1 | X | $\overline{\overline{X}}$ |

# อธิบาย

- ตารางการทำงานของR-S แลตซ์ ชนิดที่ใช้แนนด์เกตเป็นไปดังสภาวะที่กล่าวมาทั้งหมด สามารถนำค่าที่ได้มาเขียนลงในตารางความจริง
- นักศึกษาทดลองลำดับการทำงานตามลำดับในตาราง 1-4 โดยใช้วงจร R-S แนนด์เกตแลตซ์ ประกอบการพิจารณาจะได้เข้าใจอย่างถ่องแท้

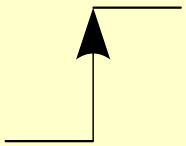
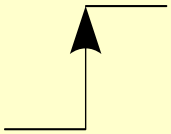
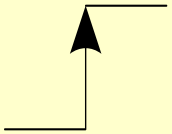
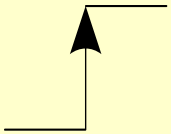
# 642 RS ฟลิปฟลอป



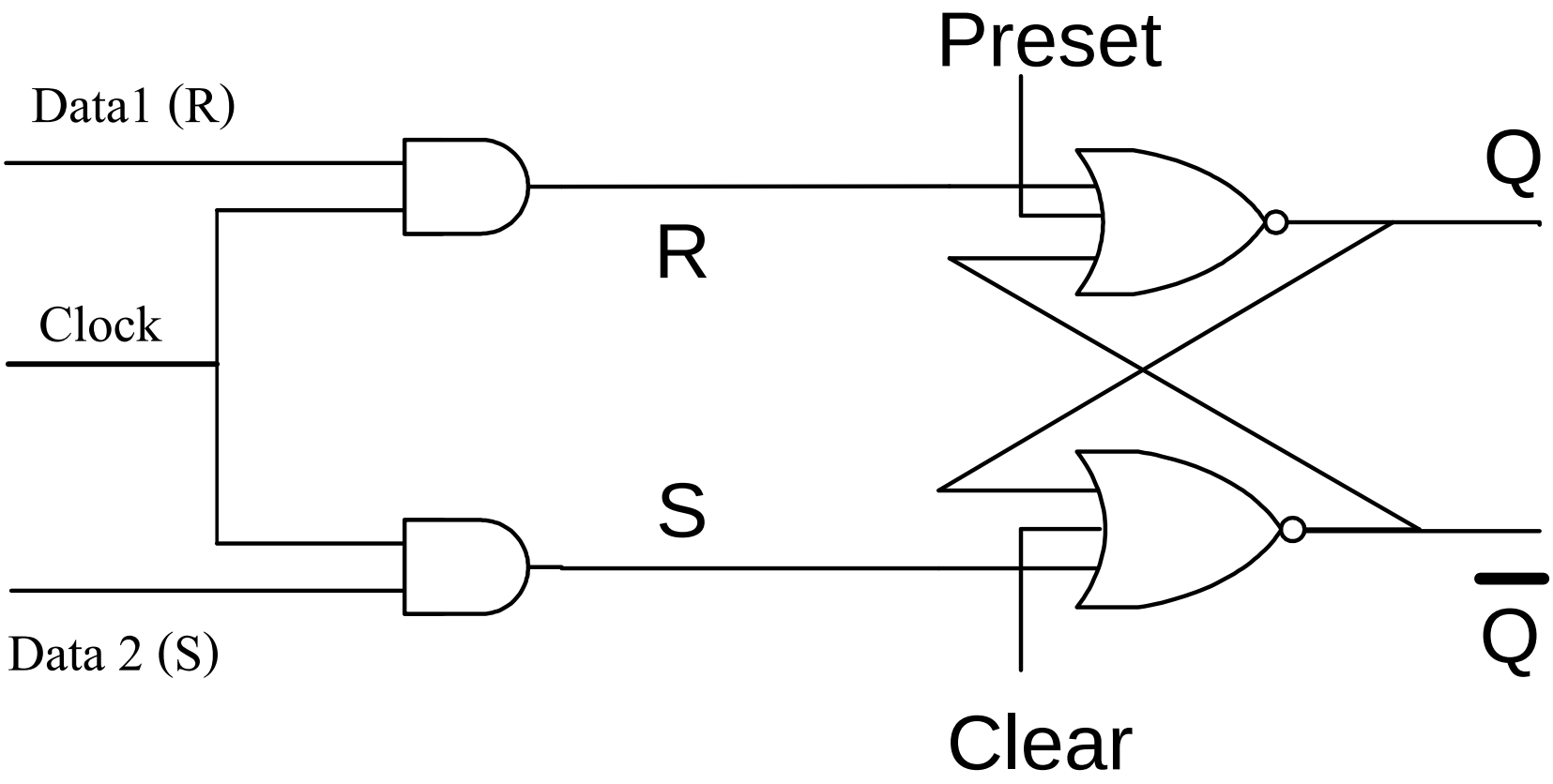

Q เป็นสภาวะปกติ

$Q_n$เป็นสภาวะใหม่ที่เกิดจากการเปลี่ยนแปลงการทำงานโดยมีการควบคุมทางอินพุต

# ตารางการทำงาน R-S ฟลิบฟลอบ

| Ck | R | S | Q | $\bar{Q}$ | |
|---|---|---|---|---|---|
|  | 0 | 0 | $Q_N$ | $\bar{Q}_N$ | คงเดิม(Store) |
|  | 0 | 1 | 1 | 0 | |
|  | 1 | 0 | 0 | 1 | |
|  | 1 | 1 | - | - | ไม่เสถียรภาพ |

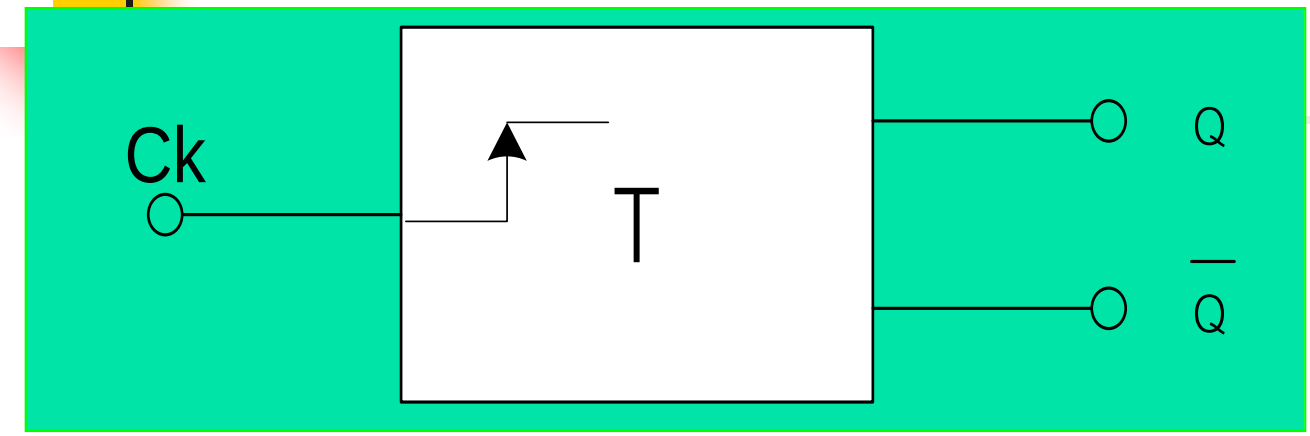
# วงจร RS ฟลิปฟลอบแบบขาขึ้น



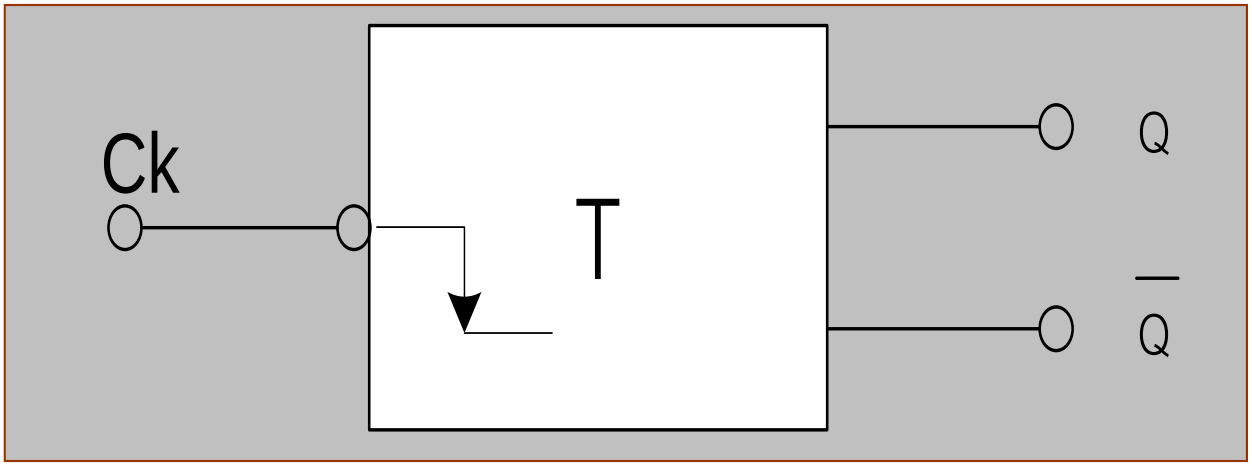
# อธิบาย

- R-S ฟลิบฟลอบ มีพื้นฐานจากการนำวงจรแลตซ์มาใช้ โดยมีสภาวะการควบคุมจากสัญญาณคล็อก ทำงานร่วมกับอินพุต
- สัญญาณคล็อกจะทำงานที่ขอบขาขึ้น ร่วมกับสัญญาณอินพุตที่ประกอบด้วย R-S เป็นไปดังตารางการทำงาน R-S ฟลิปฟลอป
- รูปแบบวงจร R-S ฟลิบฟลอบ โดยใช้นอร์เกต 3 อินพุต พร้อมด้วยขั้วควบคุม Preset (บางครั้งก็เรียก Set ) และขั้วควบคุมClear (CLR)จะทำงานในสภาวะเป็น 1
- สัญญาณR-S จะผ่านแอนด์เกต โดยต่อสัญญาณคล็อกเข้าควบคุมอินพุตของแอนด์เกต การควบคุมR-S ฟลิบฟลอบจึงขึ้นอยู่กับสัญญาณคล็อกดังที่กล่าวมา

# 643 T ฟลิปฟลอป ( Toggle Flip – Flop )




T ฟลิปฟลอบขอบขาขึ้น

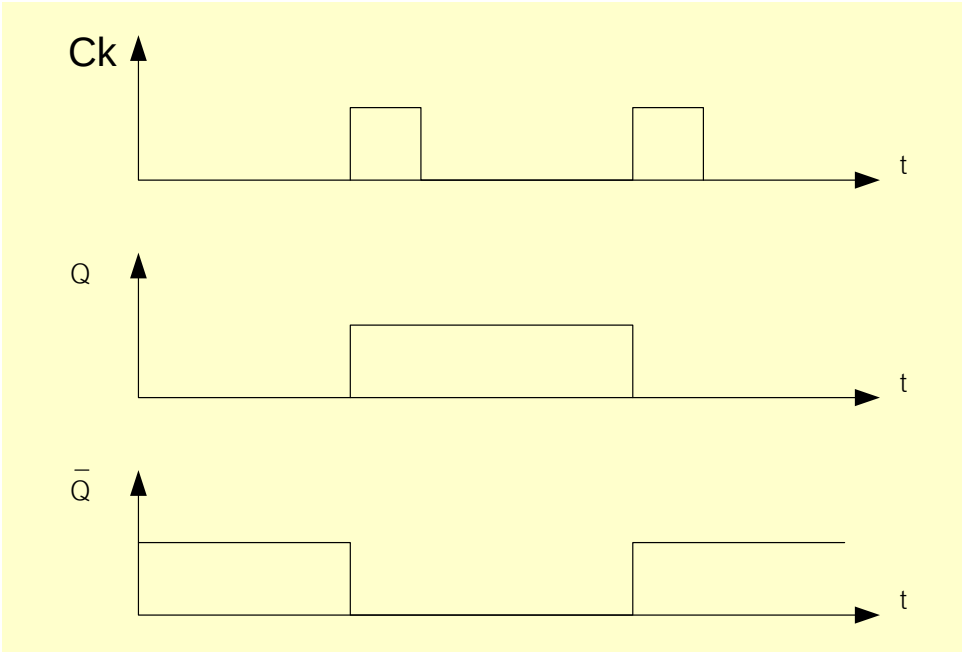



T ฟลิบฟลอบขอบขาลง

# อธิบาย

- T ฟลิบฟลอบ หรือบางครั้งเรียกว่า Toggle ฟลิบฟลอบ
- การทำงานหน้าที่หลักของ T ฟลิบฟลอบนี้ จะใช้เป็นตัวหารคล็อกที่ป้อนเข้ามาทางอินพุต
- ลักษณะการทำงานของ T ฟลิบฟลอบมีการทำงาน 2 รูปแบบดังในรูป
- T ฟลิบฟลอบขอบขาขึ้น จะทำงานเมื่อมีอินพุตป้อนเข้ามามีการเปลี่ยนแปลงจากค่า 0 เป็น 5 โวลต์ของสัญญาณคล็อก
- T ฟลิบฟลอบแบบขอบขาลง จะทำงานเมื่อมีอินพุตป้อนเข้ามามีการเปลี่ยนแปลงจากค่า 5 เป็น 0 โวลต์ของสัญญาณคล็อก T ฟลิบฟลอบแบบนี้จะมีสัญลักษณ์ที่แตกต่างคือมีวงกลม (Bubble) ที่อินพุต

# Timing diagram/Truth Table

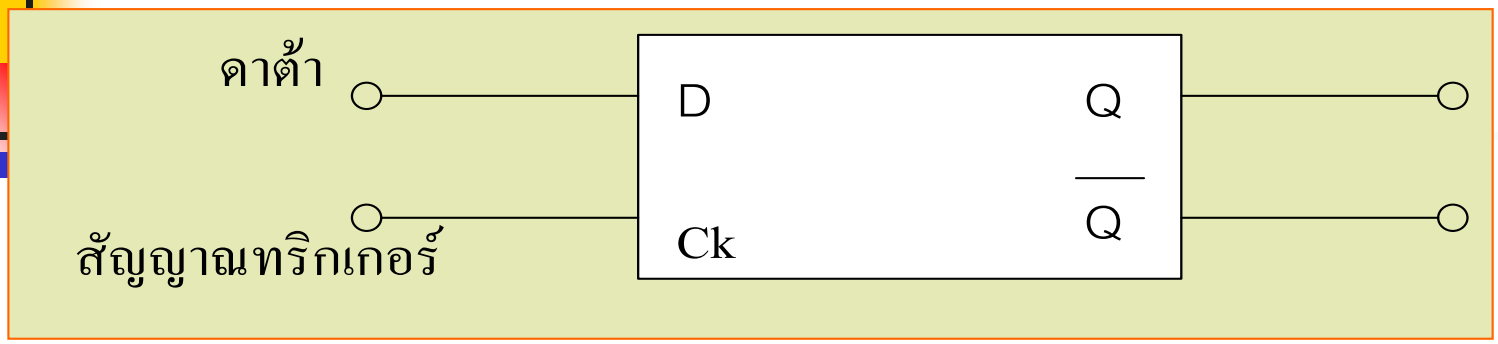



| tn | tn + 1 | |
|---|---|---|
| Ck | Q | $\bar{Q}$ |
| ⎍ | $\bar{Q}_n$ | $Q_n$ |
| ⎍ | $Q_n$ | $\bar{Q}_n$ |

# อธิบาย

- ไทม์มิ่ง ไดอะแกรมของการทำงาน T ฟลิบฟลอบทั้งขอบขาขึ้นและขอบขาลงจะเกิดจากตารางความจริงที่เหมือนกัน
- สัญญาณคล็อกที่ทำงานที่ขอบขาขึ้นเป็นไปดังรูปไทม์มิ่งไดอะแกรมที่แสดงในช่วงที่คล็อกเปลี่ยนจากค่า 0 ไปเป็นค่า 1 จะมีการเปลี่ยนแปลงทางเอาต์พุตจาก 0 ไปเป็น 1 เช่นกัน
- สัญญาณคล็อกที่ทำงานที่ขอบขาลง จะถูกหน่วงทางเอาต์พุต เป็นเวลาค่า T ของช่วงความกว้างของคล็อก คือ ค่าที่เกิดการเปลี่ยนค่าขอบขาลง รูปไม่ได้แสดงไว้ นักศึกษาลองนำกลับไปพิจารณาดูเปรียบกัน

# 644 D ฟลิบฟลอบแบบขาขึ้น

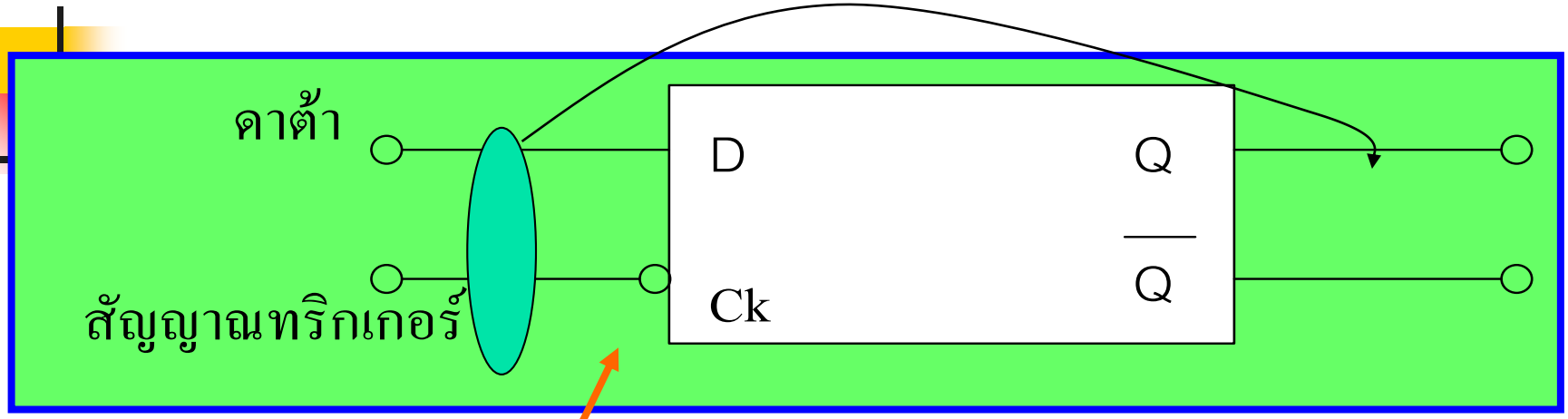



**ตารางการทำงาน**

| สัญญาณคล็อก | ดาต้า | Q | $\overline{Q}$ |
|---|---|---|---|
| – | X | $Q_N$ | $\overline{Q}_N$ |
| ↑ | 0 | 0 | 1 |
| ↑ | 1 | 1 | 0 |

**X= มีค่าเป็นได้ทั้ง 0,1**

# D ฟลิบฟลอบแบบขาลง




| สัญญาณคล็อก | ดาต้า | Q | $\overline{Q}$ |
|---|---|---|---|
| - | X | $Q_N$ | $\overline{Q}_N$ |
| ↓ | 0 | 0 | 1 |
| ↓ | 1 | 1 | 0 |

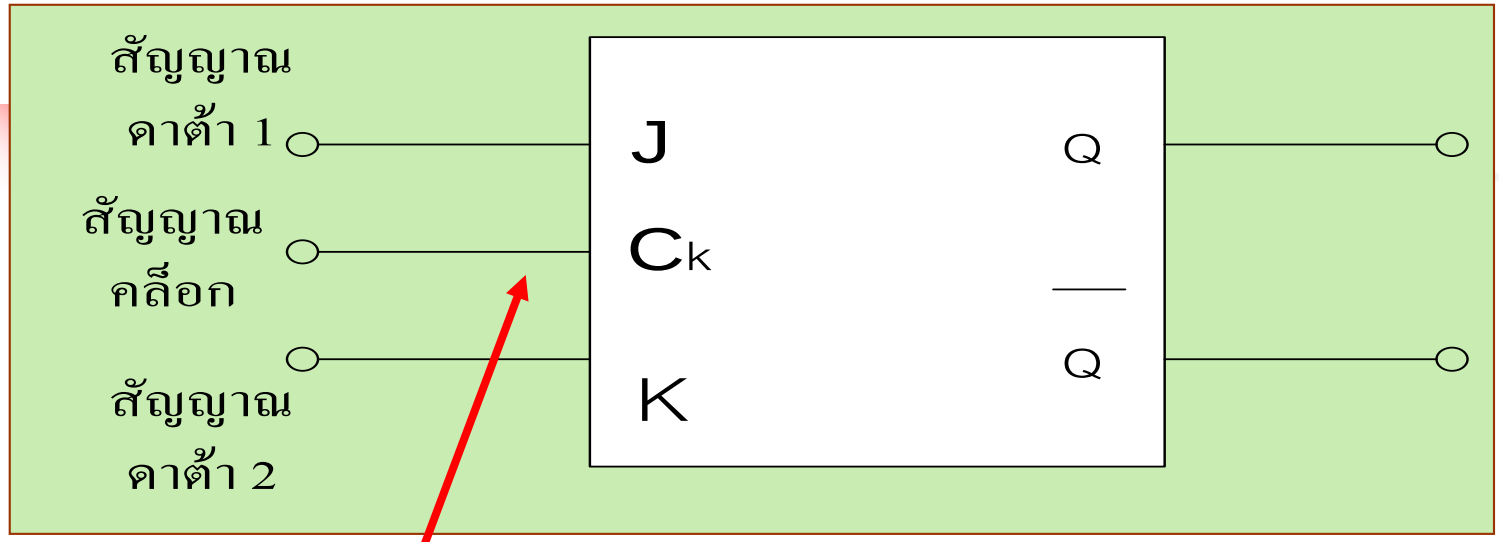
**X= มีค่าเป็นได้ทั้ง 0,1**

# อธิบาย

- D ฟลิบฟลอบ บางตำราใช้คำว่า Data ฟลิบฟลอบเพราะใช้นำไปเก็บข้อมูลส่วนบางตำราจะใช้ Delay ฟลิบฟลอบ
- ในที่นี้ขอใช้คำว่า Delay ฟลิบฟลอบ ทำหน้าที่เป็นตัวส่งข้อมูล หรือเก็บข้อมูล
- การทำงานของ D ฟลิบฟลอบ จะทำงานสัมพันธ์กับคล็อกที่ป้อนเข้ามา หากไม่มีคล็อกป้อนมาทริกจะไม่มีการเปลี่ยนแปลงใดๆ ทางเอาต์พุต
- ถ้ามีสัญญาณคล็อกป้อนเข้ามาด้วย Active High ข้อมูลจะถูกส่งจาก D (อินพุต) ไปสู่เอาต์พุต ดังตารางความจริง
- D ฟลิบฟลอบนี้มีการทำงานทั้ง Active High และ Active Low

# 645 JK ฟลิบฟลอบ

## JK ฟลิบฟลอบแบบขอบขาขึ้น



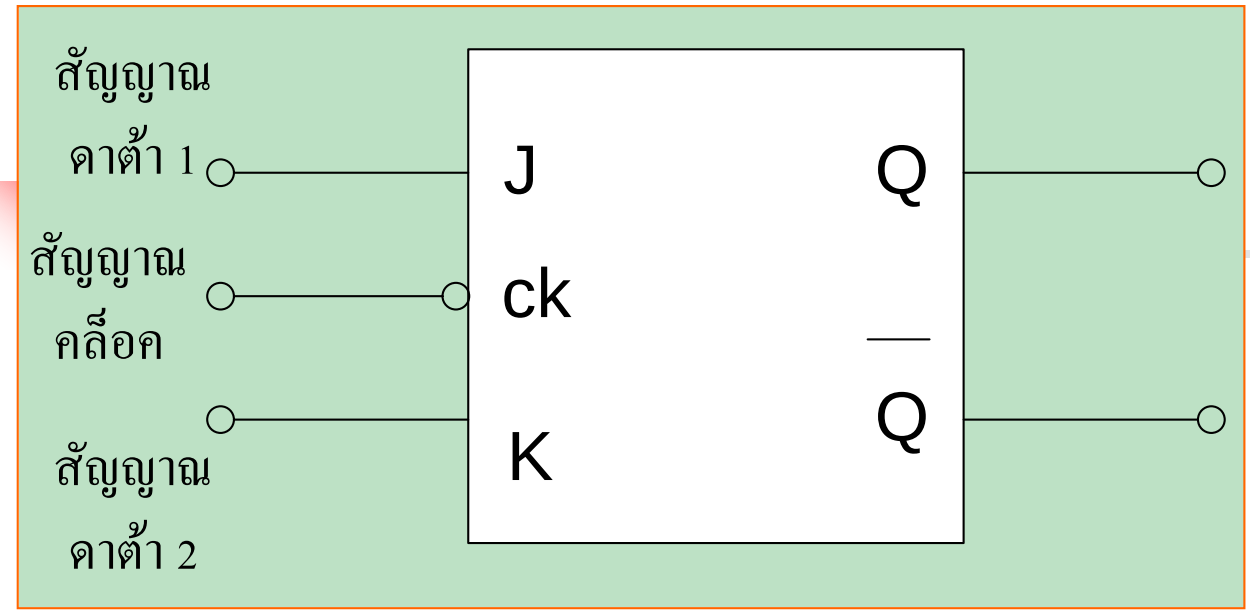

ตารางการทำงาน

| Ck | J | K | Q | $\overline{Q}$ |
|---|---|---|---|---|
| ↑ | 0 | 0 | $Q_N$ | $\overline{Q}_N$ |
| ↑ | 0 | 1 | 0 | 1 |
| ↑ | 1 | 0 | 1 | 0 |
| ↑ | 1 | 1 | $\overline{Q}_N$ | $Q_N$ |

# อธิบาย

- J-Kฟลิบฟลอบมีการทำงาน โดยคุณลักษณะของฟลิบฟลอบมีการแก้ปัญหาของ R-S ในสภาวะที่ไม่เสถียรภาพคือ เป็น1ทั้งคู่ทางอินพุตซึ่งในสภาวะนี้ไม่ยอมให้เกิดใน R-S แต่การทำงานของ J-K นำสภาวะยอมให้เกิดได้ แต่สร้างเงื่อนไขให้ทำงานแบบใหม่ของ J = 1, K = 1 คือแบบToggle การทำงานเทียบกับ R-S
  ฟลิบฟลอบคือ R = K , J = S
- Ck , J=0, K=0 , Q=Qn , $\bar{Q}$=Qn no change
- Ck , J=0, K=1 , Q=0 , $\bar{Q}$= 1 Reset
- Ck, J=1, K=0 , Q=1 , $\bar{Q}$= 0 Set
- Ck, J=1, K=1 , Q=Qn , $\bar{Q}$= $\bar{Q}$n Toggle

# JK ฟลิบฟลอบแบบขาลง



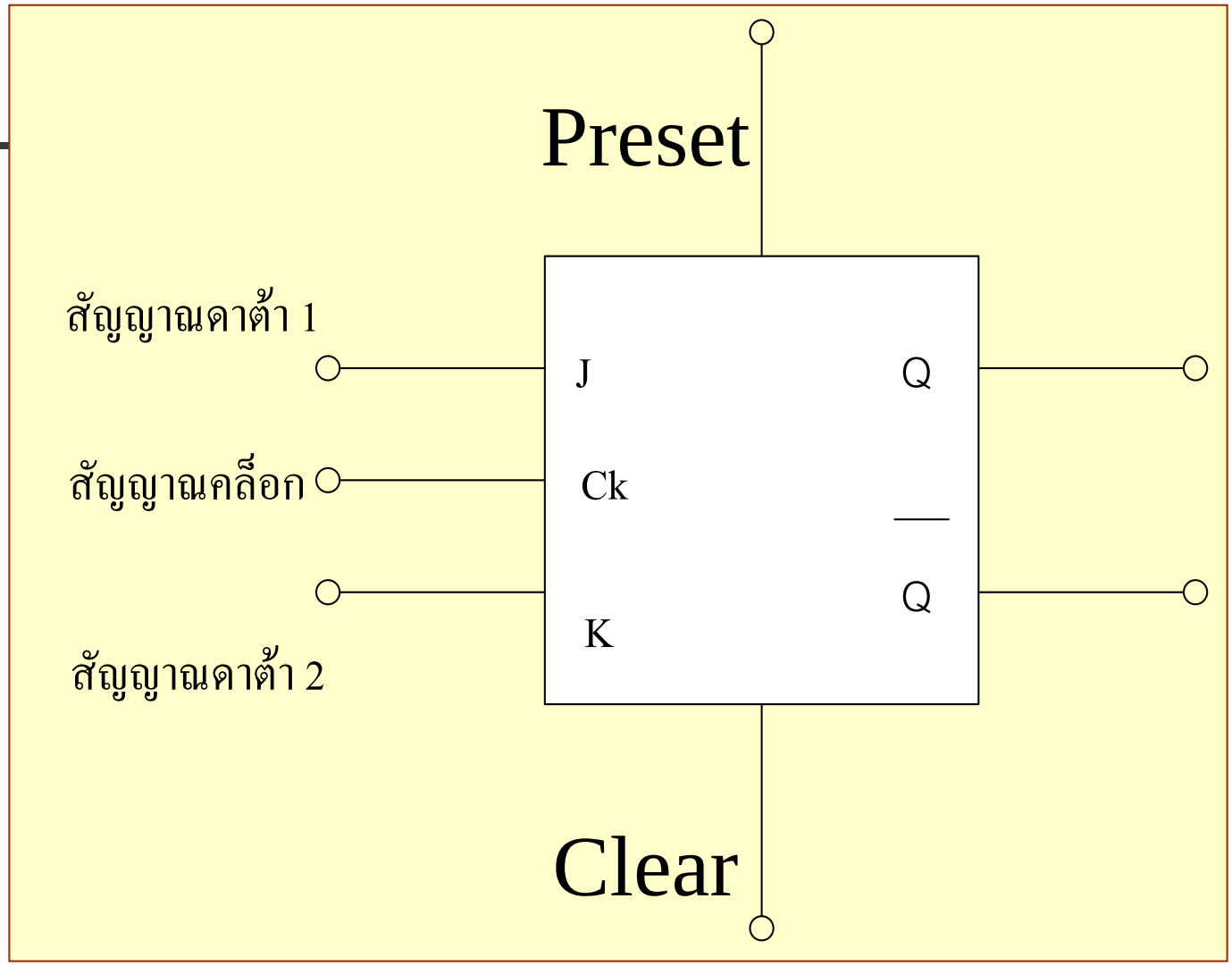

**ตารางการทำงาน**

| Ck | J | K | Q | $\overline{Q}$ |
|---|---|---|---|---|
| ↓ | 0 | 0 | $Q_N$ | $\overline{Q}_N$ |
| ↓ | 0 | 1 | 0 | 1 |
| ↓ | 1 | 0 | 1 | 0 |
| ↓ | 1 | 1 | $\overline{Q}_N$ | $Q_N$ |

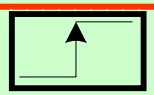
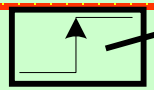
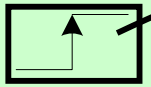
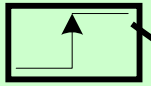
# อธิบาย

- การทำงานของ J-K ฟลิบฟลอบแบบที่ใช้สัญญาณควบคุมขอบขาลงมีสัญลักษณ์ดังรูป
- ตารางการทำงานของ J-K ฟลิบฟลอบ แบบนี้ทำงานเหมือนกันกับแบบขอบขาขึ้น เพียงแต่ต่างกันในช่วงการทำงานของฟลิบ ฟลอบจะเกิดช่วงหลังคล็อก
- ค่าเอาต์พุตที่เกิดตามสภาวะต่างๆ ได้อธิบายมาก่อนหน้าแล้ว

# ไดเรกอินพุตของ J – K ฟลิบฟลอบแอกทีฟไฮ



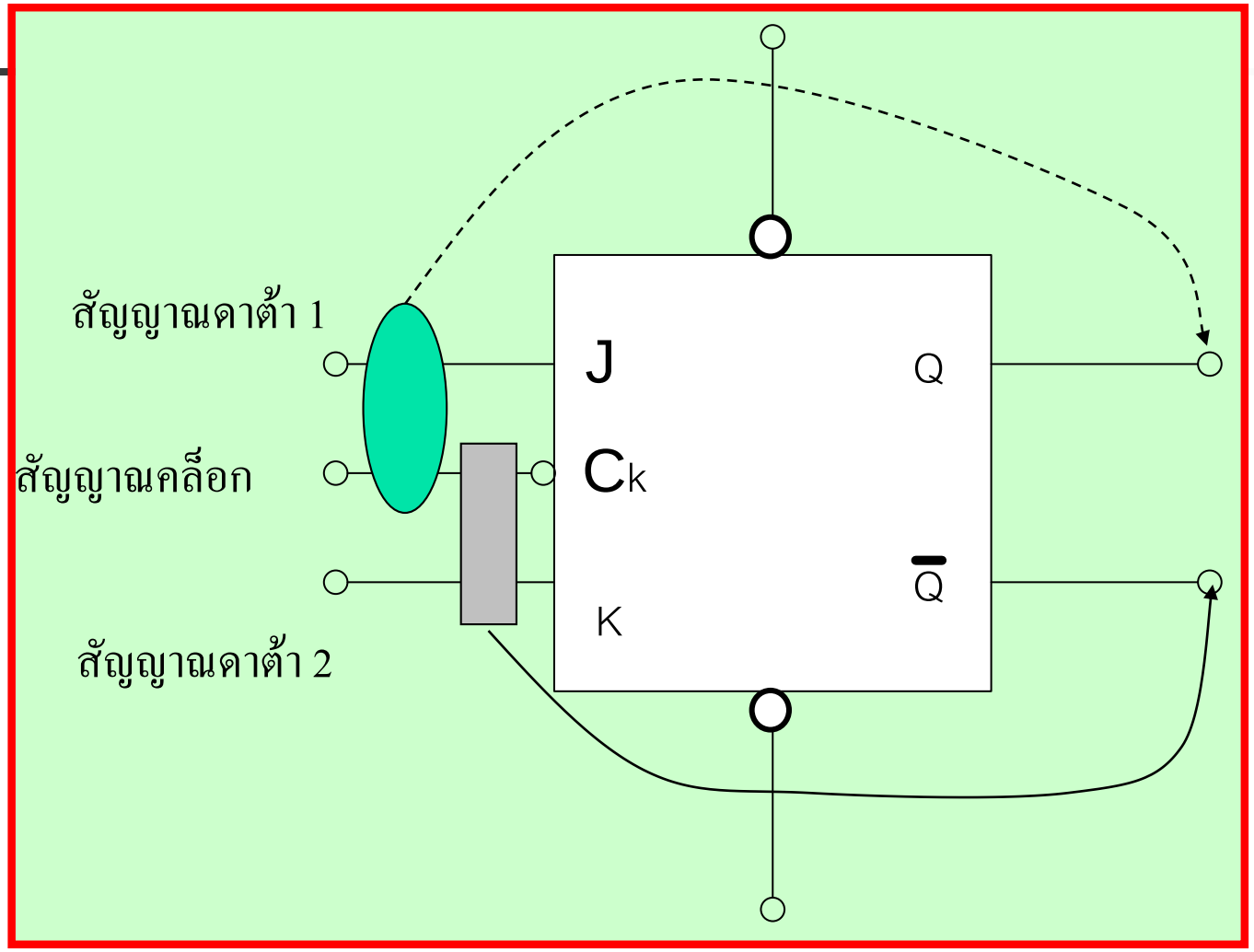
# ตารางการทำงานของแอกทีฟไฮ

| สัญญาณคล็อก | ดาต้า1 | ดาต้า2 | เคลียร์ | พรีเซต | Q | $\overline{Q}$ |
|---|---|---|---|---|---|---|
|  | 0 | 0 | 0 | 0 | $Q_N$ | $\overline{Q}_N$ |
|  | 0 | 1 | 0 | 0 | 0 | 1 |
|  | 1 | 0 | 0 | 0 | 1 | 0 |
|  | 1 | 1 | 0 | 0 | $\overline{Q}_N$ | $Q_N$ |
| X | X | X | 1 | 0 | 0 | 1 |
| X | X | X | 0 | 1 | 1 | 0 |

# อธิบาย

- การทำงานของ J-k ฟลิบฟลอบแบบมีขั้ว Clear และPreset เพิ่มมาในฟลิบฟลอบแบบนี้
- สัญญาณทางอินพุตจะมี 3 ส่วน คือ ดาต้า1, ดาต้า 2 , คล็อก
- สัญญาณการควบคุมเป็นสัญญาณเคลียร์ กับ พรีเซต
- จากรูปสัญญาณคล็อก จะทำงานที่ Active High
- ดูตารางความจริงประกอบ
- Reset Ck , Data1 = 0, Data 2 = 1 Q = 0, Q = 1
- Set Ck , Data1 = 1, Data 2 = 0 Q = 1, Q = 0
- Toggle Ck , Data1 = 1, Data 2 = 1 Q = Qn, Q = Qn
- สัญญาณควบคุมClear จะทำงานที่ค่าระดับHigh โดยไม่สนใจคล็อกและดาต้า1,2 ว่าจะมีเข้ามาหรือไม่ เอาต์พุตจะเป็น0 ทันที
- สัญญาณควบคุม Preset จะทำงานที่ระดับHigh เช่นกันโดยไม่สนใจคล็อกและดาต้า1,2 จะมีเข้ามาหรือไม่ก็ตามเอาต์พุตจะมีค่าเป็น 1 ทันที
- สัญญาณควบคุม Preset, Clear มีหน้าที่กำหนดค่าเริ่มต้น และค่ากระโดดการทำงาน จนถึงกำหนดจุดสุดท้ายของระบบ เมื่อมีการทำงานซ้ำกันอีก

# ไดเรกอินพุตของ J – K ของฟลิบฟลอบแอกทีฟโลว์



# ตารางการทำงานของแอกทีฟโลว์

| สัญญาณคล็อก | ดาต้า1 | ดาต้า2 | เคลียร์ | พรีเซต | Q | $\overline{Q}$ |
|---|---|---|---|---|---|---|
| ↓ | 0 | 0 | 1 | 1 | $Q_N$ | $\overline{Q}_N$ |
| ↓ | 0 | 1 | 1 | 1 | 0 | 1 |
| ↓ | 1 | 0 | 1 | 1 | 1 | 0 |
| ↓ | 1 | 1 | 1 | 1 | $\overline{Q}_N$ | $Q_N$ |
| X | X | X | 1 | 0 | 1 | 0 |
| X | X | X | 0 | 1 | 0 | 1 |

# อธิบาย

- ในรูปแบบหนึ่งของการทำงานJ -K ฟลิบฟลอบ เป็นการทำงานโดย Active Low
- คุณสมบัติการทำงานในสภาวะ Set, Reset , Toggle เป็นเหมือน J-K ฟลิบฟลอบ

  แบบการทำงาน Active High
- สัญญาณคล็อกจะทำงานในช่วง Active Low ที่จะนำไปควบคุมคุณสมบัติ Set, Reset , Toggle
- สัญญาณควบคุม Clear , Preset จะทำงานที่ค่าระดับ Low

# ตาราง J – K Flip – Flop Exciting Table

| สถานะเอาต์พุต | | การทำงาน | J | K | สรุป | |
|---|---|---|---|---|---|---|
| Qเดิม | $Q_n$ใหม่ | | | | J | K |
| 0 | 0 | Store | 0 | 0 | 0 | X |
| | | Reset | 0 | 1 | | |
| 0 | 1 | Toggle | 1 | 1 | 1 | X |
| | | Set | 1 | 0 | | |
| 1 | 0 | Reset | 0 | 1 | X | 1 |
| | | Toggle | 1 | 1 | | |
| 1 | 1 | Store | 0 | 0 | X | 0 |
| | | Set | 1 | 0 | | |

# อธิบาย

- ตารางการทำงานของ J-K ฟลิบฟลอบ ที่พิจารณาถึงสภาวะเดิม(Present state ใช้แทนด้วย Q) ไปสู่สภาวะใหม่ (Next state ใช้แทนด้วย $Q_n$)
- Q = 0, Qn = 0 สภาวะนี้เหมือนเดิมเป็นการทำงานของ Reset กับStore
- ค่า J = 0 , K = 0 ไม่มีการควบคุมใดๆ สภาวะนี้เรียก Store
- ค่า J = 0 , K = 1 มีการควบคุมที่ K สภาวะนี้เป็น Reset
- สรุปการทำงานกรณีที่1 ค่าเอาต์พุตเดิมเป็น0 เปลี่ยนเป็น 0 ค่า J เป็น 0 ค่าK จะเป็นอะไรก็ได้Q=0, Qn=1 กรณีนี้ทางด้านเอาต์พุตจะถูกSet ค่าเป็น1
- เป็นการทำงานของการ Set กับการทำงานของToggle ค่า J=1,K=0และJ=1, K=1

สรุปการทำงานกรณีที่2 สภาวะเอาต์พุตเป็น 0 เปลี่ยนไปเป็น 1 ค่า J เป็น 1 ส่วนค่า Kเป็นอะไรก็ได้

# อธิบาย(ต่อ)

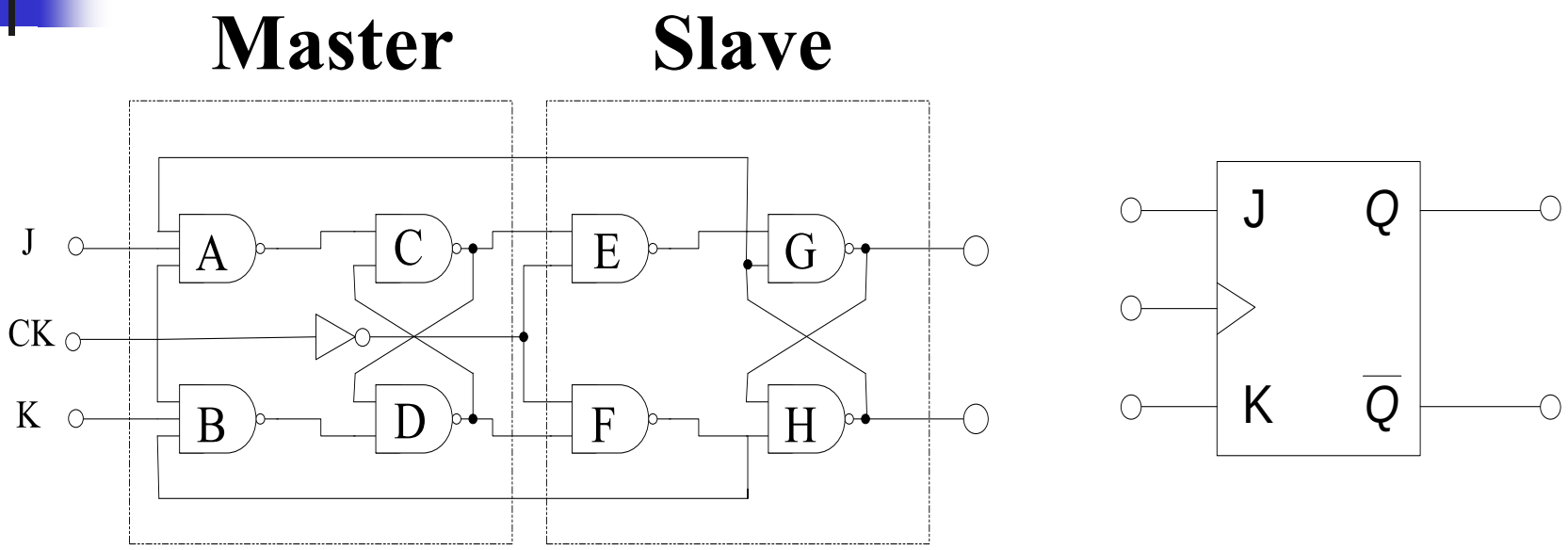
- Q = 1, Qn = 0 เงื่อนไขของกรณีนี้ทางด้านเอาต์พุตจะถูกReset ค่าเป็น0
- เป็นการทำงานของการ Reset กับการทำงานของToggle
- ค่า J=0, K=1 และ J=1 , K=1

**สรุปการทำงานกรณีที่3 สภาวะเอาต์พุตเป็น 1เปลี่ยนไปเป็น0 ค่าJเป็นอะไรก็ได้ ส่วนค่าKจะเป็น1** Q = 1, Qn = 1เงื่อนไขของกรณีนี้ทางด้านเอาต์พุตจะถูกSet ค่าเป็น1

- เป็นการทำเป็นงานของการ Set กับการทำงานของ Store
- ค่า J = 1, K = 0 และ J = 0 , K = 0

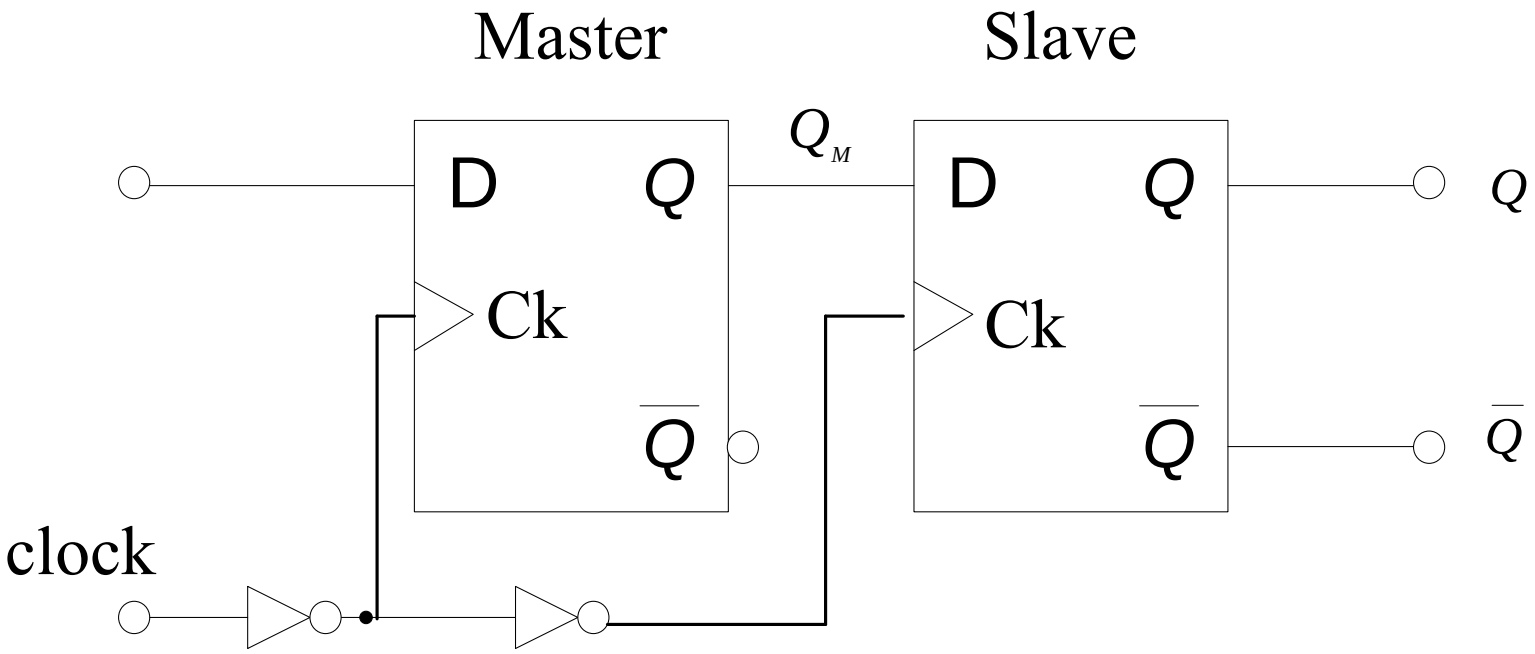
**สรุปการทำงานกรณีที่4 สภาวะเอาต์พุตเปลี่ยน 1 ไปเป็น 0 ค่า J เป็น 1 ส่วนค่า K เป็นอะไรก็ได้**

# 646 Master-Slave Flip-Flop

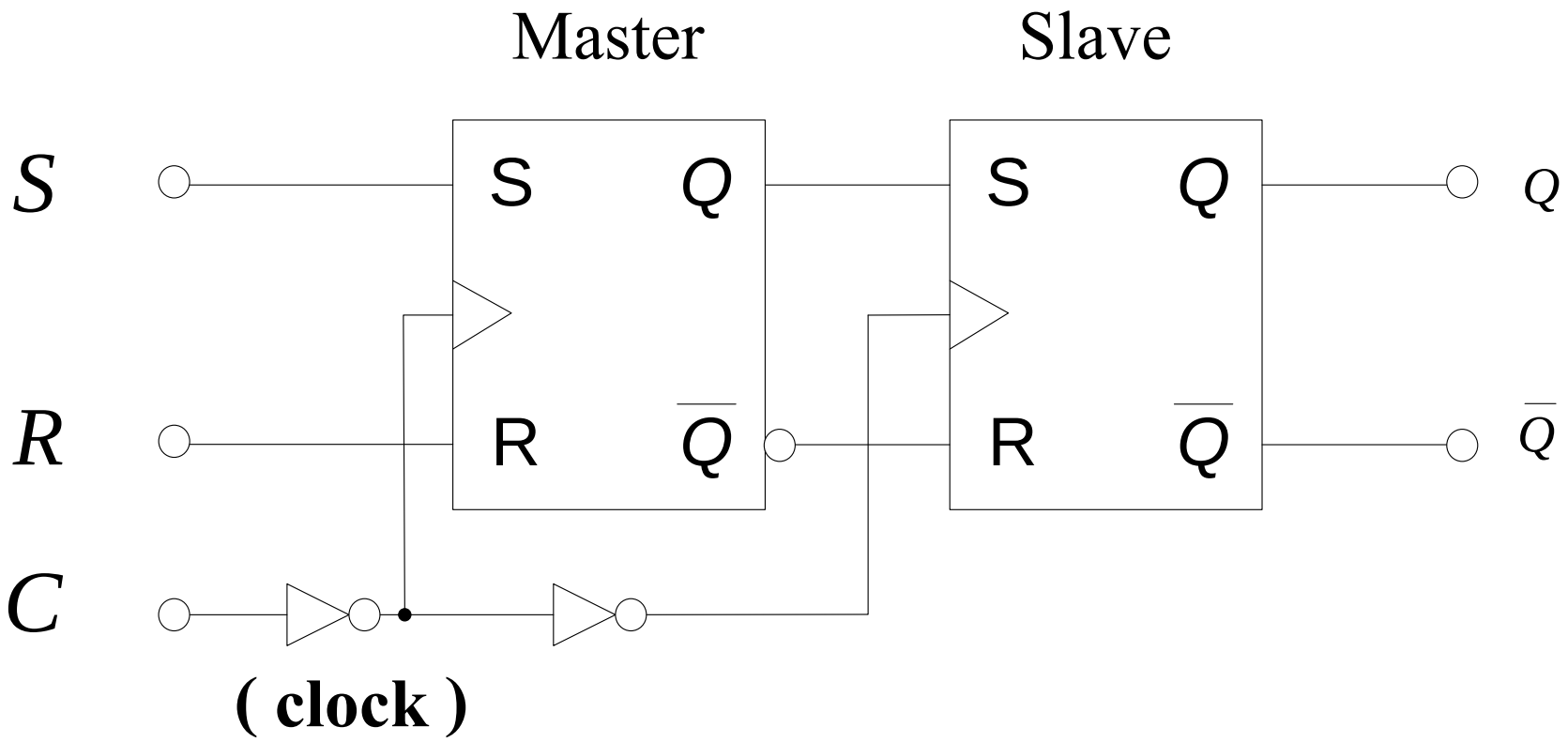



## Master-Slave J-K Flip-Flop

# Master-Slave D Flip-Flop



# Master-Slave R-S Flip-Flop

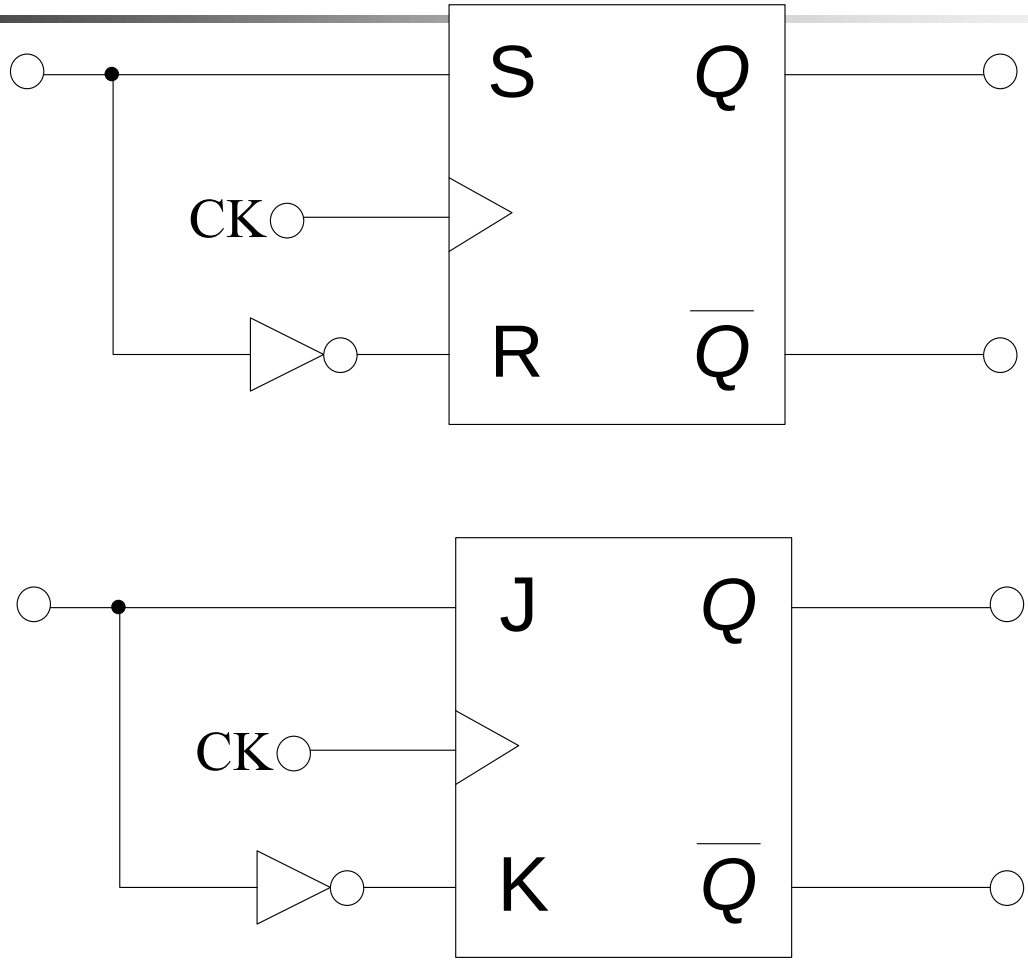


# อธิบาย

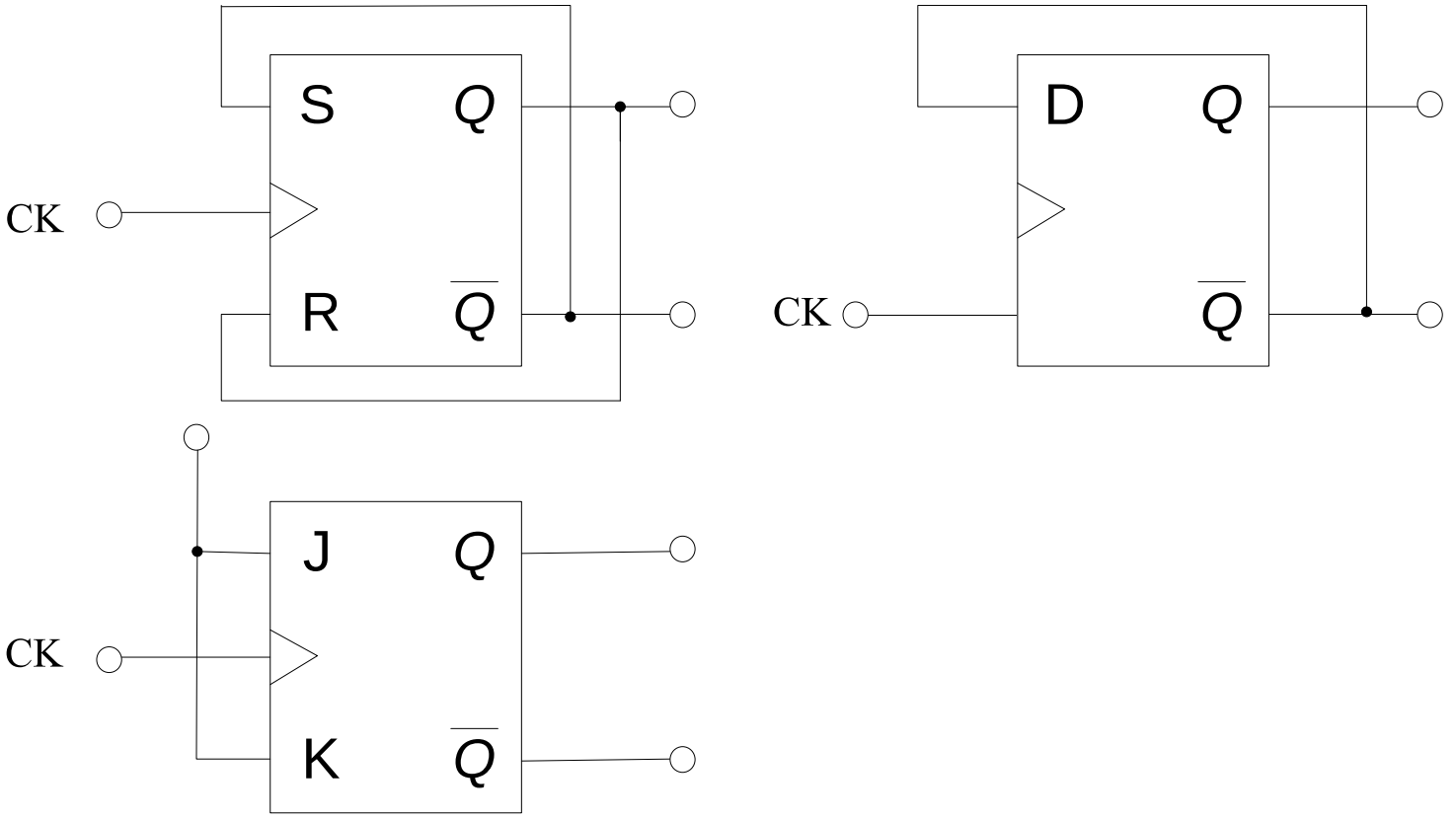
- การทำงานของฟลิบฟลอบแบบมาสเตอร์-สลาฟ
- ในระบบดิจิตอลที่มีการควบคุมการทำงานของวงจรที่ซับซ้อน สัญญาณคล็อกที่เกิดในระบบนี้มักจะเกิดการควบการทำงานไม่พร้อมกันอีกทั้งปัญหาการออส - ซิเลต, สัญญาณรบกวน เป็นเหตุให้ข้อมูลในการส่ง เกิดการผิดพลาด
- สามารถแก้ปัญหาด้วยฟลิบฟลอบแบบมาสเตอร์-สลาฟ
- ฟลิบฟลอบแบบมาสเตอร์-สลาฟจะช่วยการเก็บและส่งข้อมูล โดยจะมีวงจรฟลิบฟลอบ 2 ชุดประกอบภายใน ควบคุมการทำงานในช่วงPositive และ Negative จะส่งข้อมูลไปยังเอาต์พุตสุดท้ายเมื่อครบ 1 Pulse
- จากรูปเป็น Master-Slave D Flip-Flop , Master-Slave R-S Flip-Flop , Master-Slave J-K Flip-Flop

# 647 การแปลงฟลิบฟลอบแบบต่างๆ

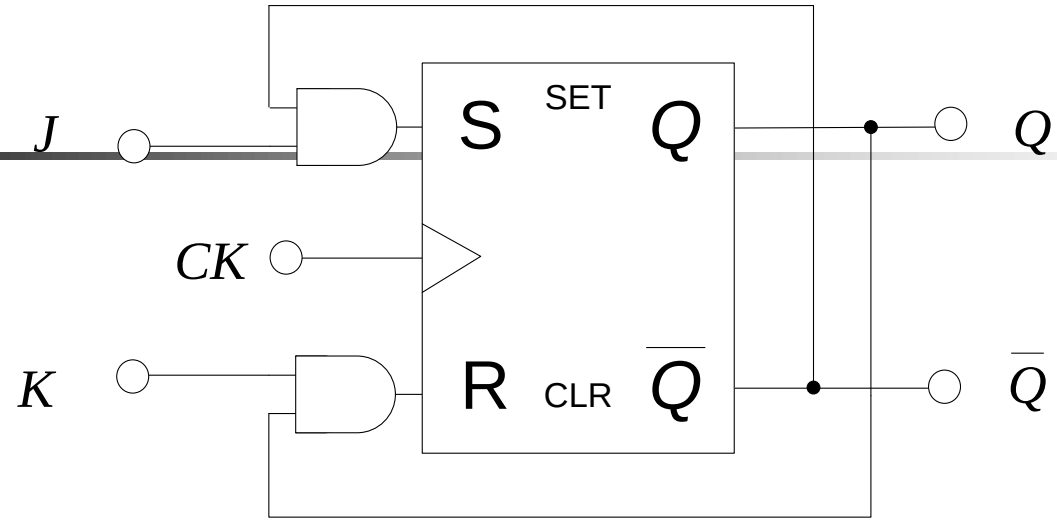
## การนำ R-S ฟลิบฟลอบไปแทน D ฟลิบฟลอบ

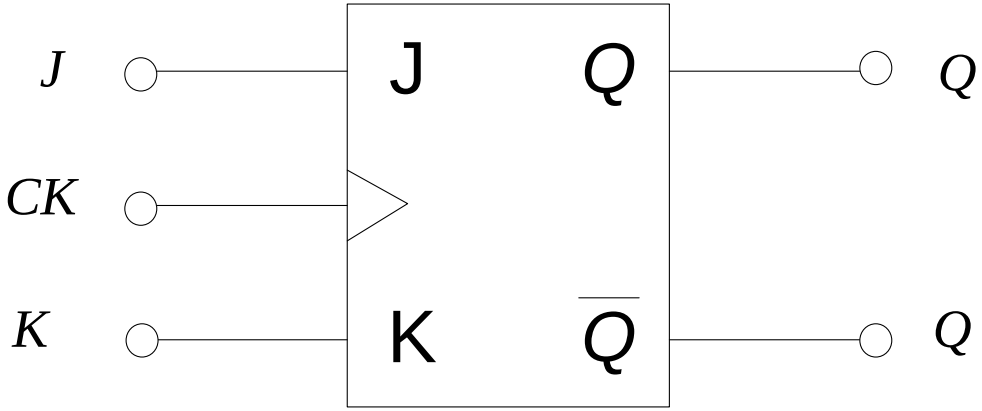


# การแปลงฟลิบฟลอบ J-K ฟลิบฟลอบ เป็น T ฟลิบฟลอบ



# การนำ R-S ฟลิบฟลอบไปแทน J-K ฟลิบฟลอบ



# อธิบาย

- การนำฟลิบฟลอบในรูปแบบต่างๆ มาใช้แทนกันสามารถใช้แทนได้ดังนี้
- R-S ฟลิบฟลอบเป็น T ฟลิบฟลอบ
- J-K ฟลิบฟลอบเป็น T ฟลิบฟลอบ
- R-S ฟลิบฟลอบเป็น J-K ฟลิบฟลอบ
- จากการนำเสนออย่างละเอียดของฟลิปฟลอบชนิดต่างๆ นักศึกษาลองนำไปประยุกต์ออกแบบและ แก้ไขปัญหาที่จะนำไปใช้จริงตามที่กล่าวมา

# ทดสอบความรู้สัปดาห์ที่11

1. R-S ฟลิบฟลอบมีการใช้งานอะไร และมีข้อแตกต่างจาก J-K ฟลิบ-ฟลอบอย่างไร
2. การทำงานของฟลิบฟลอบแบบ slave/Master ต่างจากฟลิบฟลอบทั่วไปอย่างไรบ้าง และมีข้อดีข้อเสียอะไรบ้าง
3. จงออกแบบวงจรสร้างคล็อกขนาด 1KHz มี Duty Cycle 50%
4. จงอธิบายการทำงานของวงจรอะสเตเบิล , โมโนสเตเบิล , ไบสเตเบิล พร้อมยกตัวอย่างการนำไปใช้งาน

# เนื้อหาบรรยายในสัปดาห์ที่ 12 การเลื่อนข้อมูล (Shift Register)

65 ลักษณะทั่วไปของการเลื่อนข้อมูล

651 การเลื่อนข้อมูลแบบ Serial In , Serial Out

652 การเลื่อนข้อมูลไปทางขวา (Shift- Right Register)

653 การเลื่อนข้อมูลไปทางซ้าย (Shift -Left Register)

654 การเลื่อนข้อมูลแบบวนรอบกลับ (Shift Around Register)

655 การเลื่อนข้อมูลแบบขนาน (Parallel In , Parallel Out)

656 การเลื่อนข้อมูลเข้าแบบขนานและเลื่อนข้อมูลออกแบบอนุกรม (Parallel In , Serial Out)

657 การเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน (Serial In , Parallel Out)

658 วงจรเลื่อนข้อมูลแบบ Universal ขนาด 4 บิต

# 65 ลักษณะทั่วไปของการเลื่อนข้อมูล

-ตัวเลื่อนข้อมูลบางตำราใช้คำว่า “ Shift Register” แทน เพื่อให้นักศึกษาได้รู้หลายๆ แง่มุม

ในที่นี้เพื่อกันความสับสนขอใช้คำว่าการเลื่อนข้อมูลแทน

-ตัวเลื่อนข้อมูลโครงสร้างเป็นวงจรซีเควนเชียลลอจิกในกลุ่มของฟลิบฟลอบ

-ทำหน้าที่เลื่อนข้อมูลทาง Binary ไปทางซ้ายหรือทางขวาในรูปแบบอนุกรม

-ในบางกรณี อาจจะเลื่อนข้อมูลเป็นชุดในลักษณะขนาน

-การเลื่อนข้อมูลเพื่อให้เหมาะกับการนำข้อมูลไปกระทำทางด้านคณิตศาสตร์

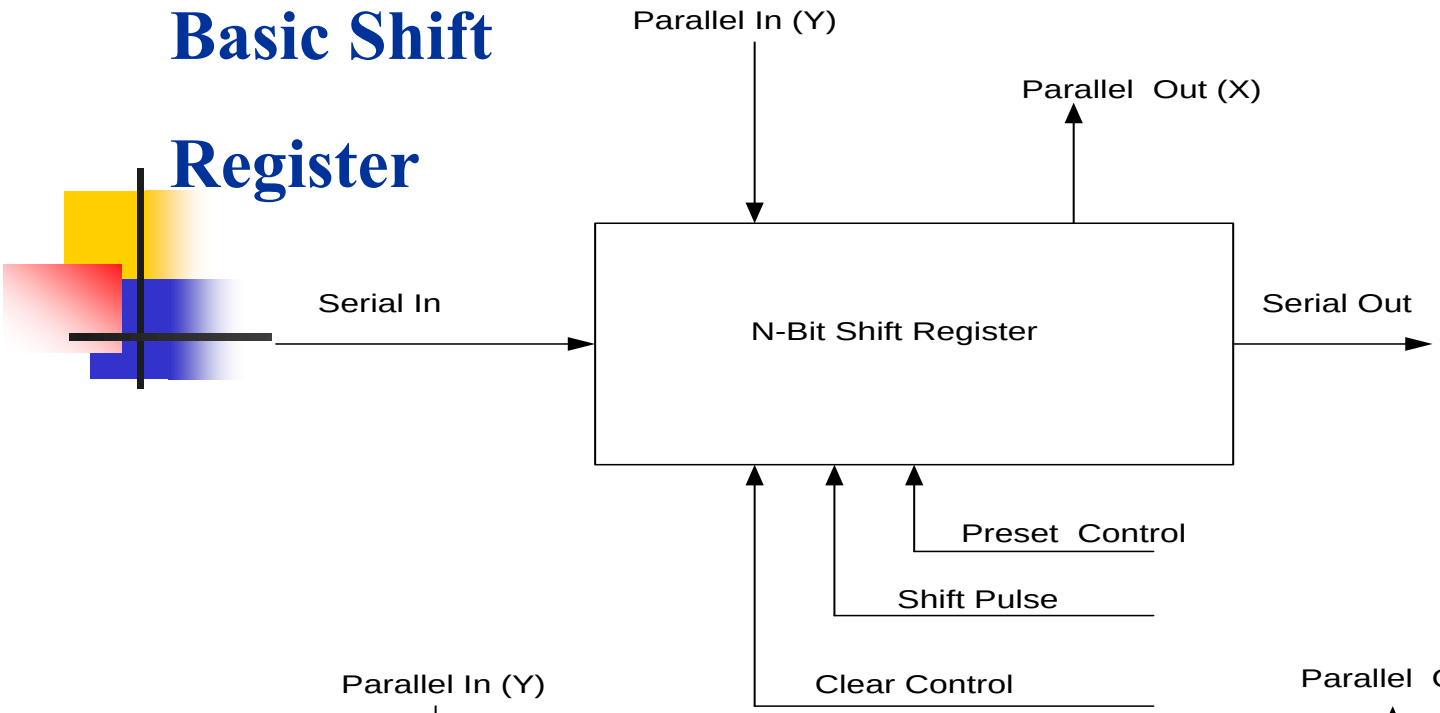
-วงจรเลื่อนข้อมูลนำไปใช้ในส่วนแสดงผล และการส่งข้อมูล

-การเลื่อนข้อมูลคือการย้ายข้อมูลค่า1หรือค่า0ในแต่ละชุดตัวเก็บข้อมูลจากชุดหนึ่งไปอีกชุดหนึ่ง

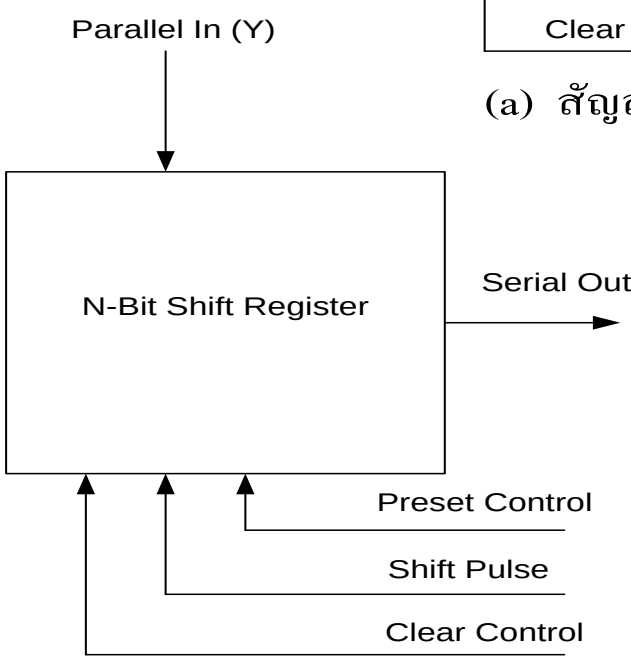
- ตัวเลื่อนข้อมูลสามารถสร้างจากฟลิบฟลอบในรูปแบบ D Flip Flop,

JK Flip Flop, T Flip Flop, RS Flip Flop.

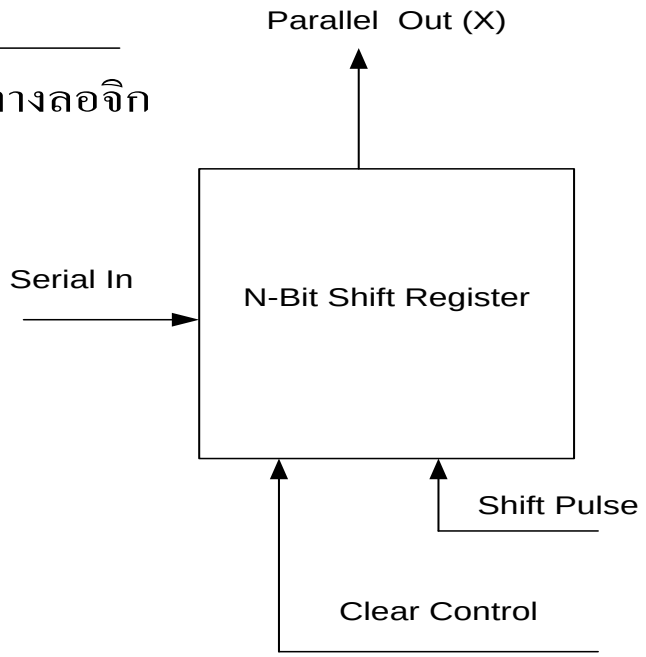
# Basic Shift Register




(a) สัญลักษณ์ทางลอจิก




(b) Parallel -In ,Serial Out

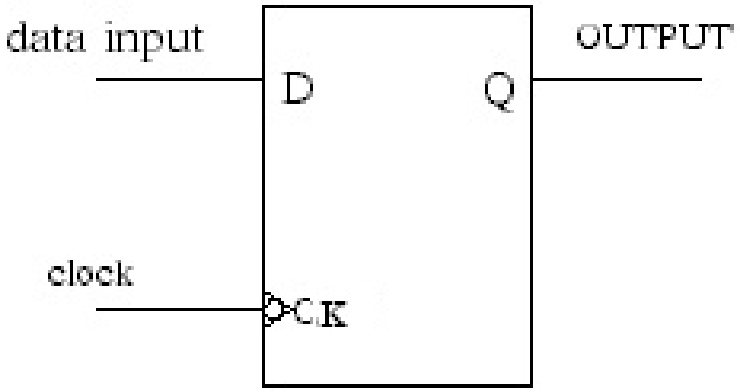


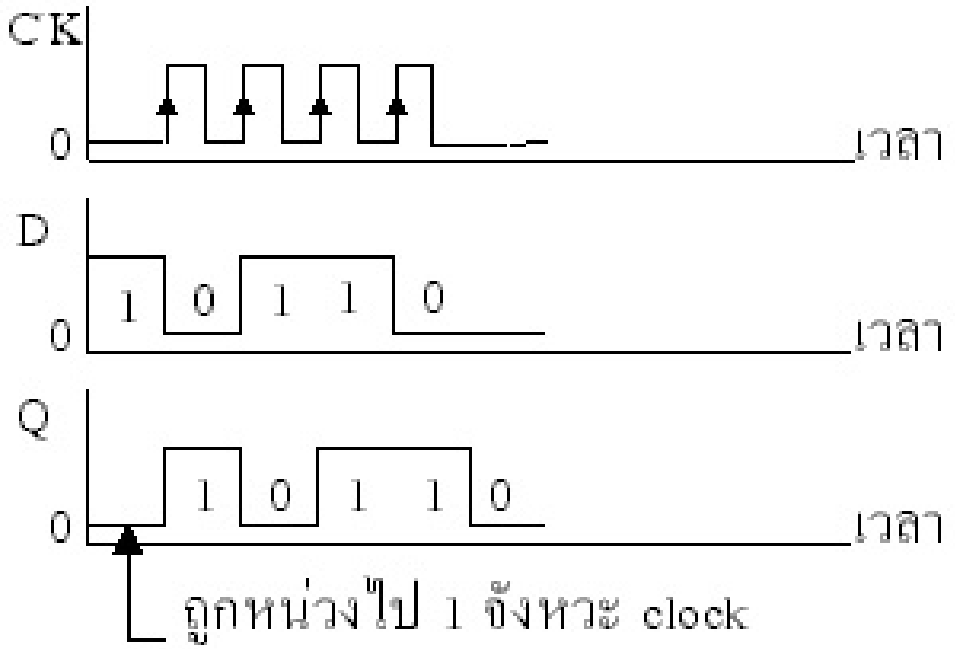

( C ) Serial-In,Parallel-Out

# อธิบาย

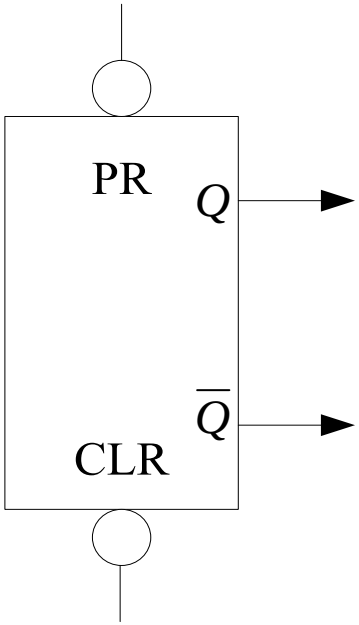
- หลักการพื้นฐานของการเลื่อนข้อมูล
- Parallel In ($Y_i$, = i , n) แทนข้อมูล1 อินพุตใดๆ ที่ป้อนเข้าที่ตัวเลื่อนข้อมูลของแต่ละตัว
- Parallel Out ($X_i$,= i , n) แทนข้อมูลที่ออกจากฟลิบฟลอบแต่ละตัว
- Shift Pulse (Sometimes Labeled คล็อก ) เป็นพลัสในการควบคุมการเคลื่อนของข้อมูลระหว่างฟลิบฟลอบ
- Serial In เป็นข้อมูลเข้าที่ป้อนแก่ฟลิบฟลอบตัวแรก และบิตต่อไปเมื่อมี Shift Pulse เป็นลำดับต่อไป
- Serial Out เป็นข้อมูลที่ออกจากฟลิบฟลอบตัวสุดท้าย ของตัวเลื่อนข้อมูลเมื่อมีShift Pulse เป็นลำดับต่อไป
- Clear Control เป็นพัลซ์ที่กำหนดการทำงานของตัวเลื่อนข้อมูลให้เป็นลอจิก 0
- Preset Control เป็นพัลซ์ที่กำหนดการทำงานของตัวเลื่อนข้อมูลให้เป็นลอจิก 1

# ลักษณะพื้นฐานของการเก็บข้อมูลโดยการใช้ D ฟลิป-ฟลอป



# ลักษณะการนำข้อมูลเข้าโดยใช้ขา PR/CLR



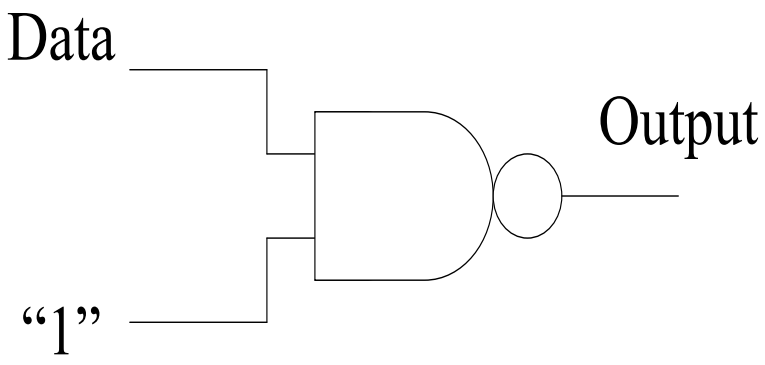

| Data | $\overline{PR}$ | $\overline{CLR}$ | Q |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 |

| Data \ CLR | 0 | 1 |
|---|---|---|
| 0 | | 1 |
| 1 | | 1 |

$$PR = \overline{DATA} + \overline{CLR}$$

$$= \overline{DATA \cdot CLR}$$

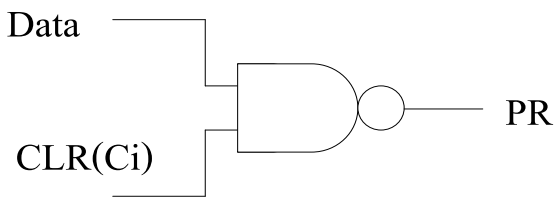
# การควบคุมทางเข้าของอินพุตโดยใช้แนนด์เกต

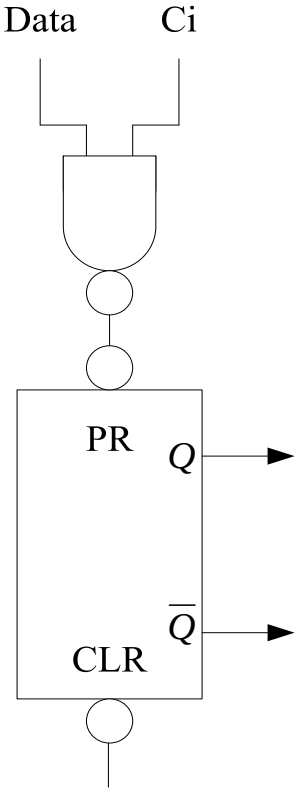


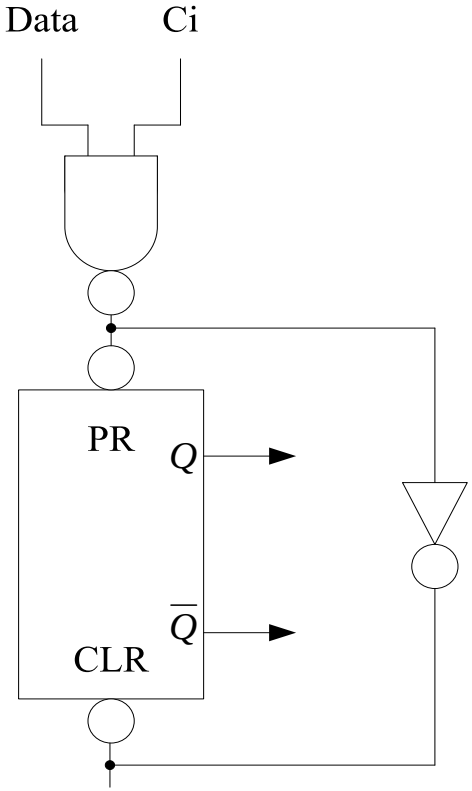

| “1” | Data | Output | CLR | PR | Q |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |

# รูปแบบการนำเข้าของข้อมูลทางอินพุต

# ร่วมกับขาควบคุม ($C_i$)



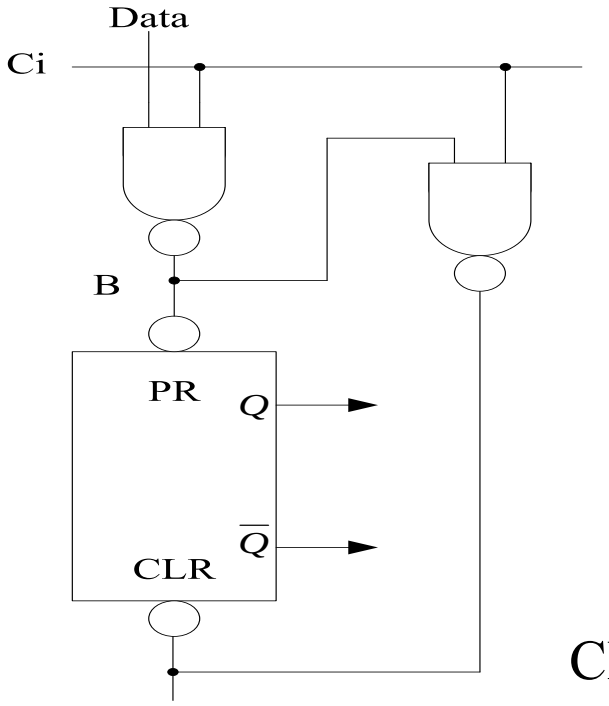



# อธิบาย

- PR เป็นขาควบคุมการทำงานของฟลิบฟลอบ ให้เอาต์พุตเป็น 1 เมื่อการควบคุมมีค่าเท่ากับ 0
- CLR เป็นขาควบคุมการทำงานของฟลิบฟลอบเมื่อสัญญาณควบคุมมีค่าเท่ากับ 0 ให้เอาต์พุตเป็น 0
- Data เป็นส่วนที่จะนำอินพุตป้อนให้กับวงจรสามารถเป็นค่าได้ทั้ง 0/1 Data จะเป็นสัญญาณให้เอาต์พุตเปลี่ยนแปลงตาม โดยมี $C_i$ เป็นตัวควบคุมการเปลี่ยนแปลงที่พร้อมกัน
- การทำงานของวงจรสามารถควบคุมการเก็บข้อมูล โดยป้อนสัญญาณเข้าที่อินพุตของแนนด์เกต พร้อมสัญญาณ $C_i$ (Control input) ซึ่งจะมีค่าเป็น1
- พิจารณาได้จากตาราง K-map ประกอบ

# การควบคุมการนำข้อมูลเข้า และส่งข้อมูล

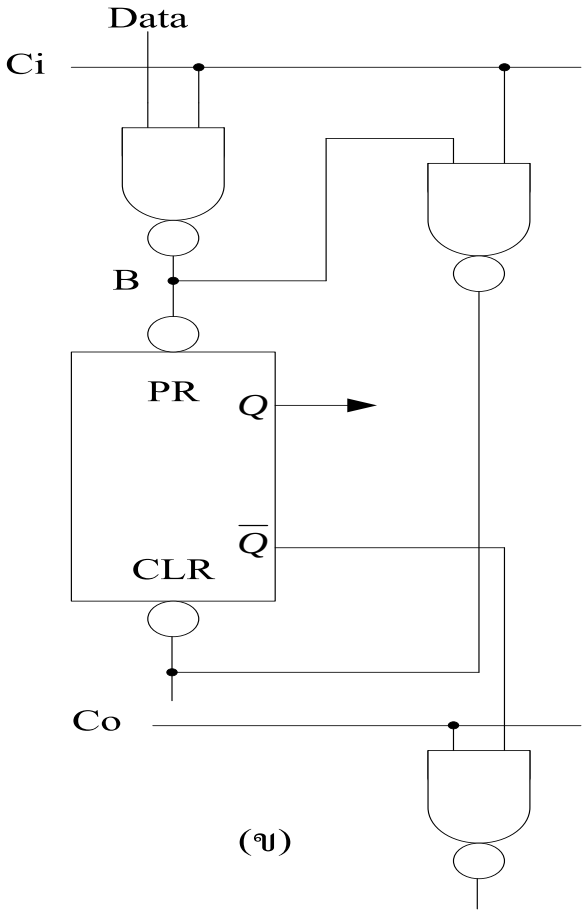
## โดยใช้ แนนด์เกต




(ก)

$PR=Ci.Data$

$CLR=\overline{(Ci.Data)}Ci$



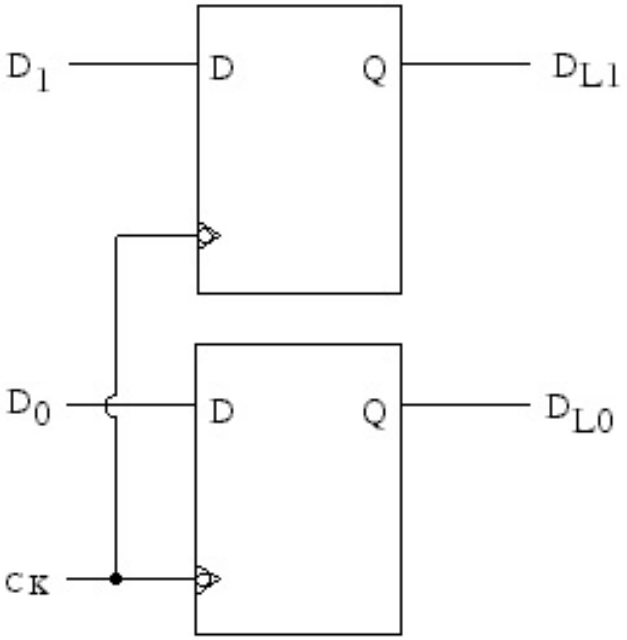

(ข)

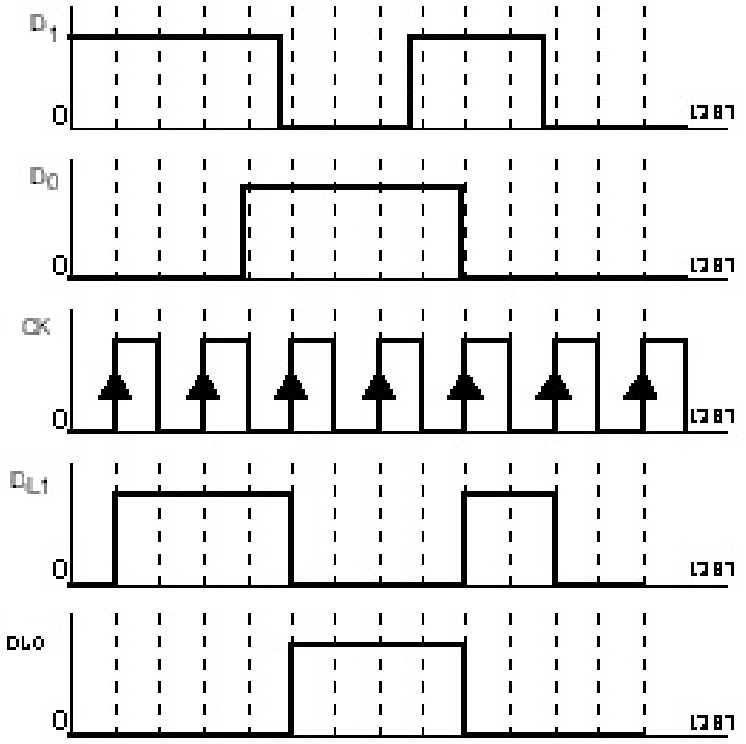
# อธิบาย

- การเก็บข้อมูลเข้ามี $C_i$ เป็นส่วนควบคุมการทำงานทางอินพุต
- ส่วนการนำข้อมูลออกไปใช้งานมี $C_o$ ทำงานร่วมกับ Complement Q
- การนำข้อมูลเข้าโดยการส่ง Data พร้อมกับการส่งสัญญาณ Control Input ($C_i$)
- เมื่อ Data ที่ป้อนเข้ามีค่าเท่ากับ 1 $C_i$ ที่ป้อนจะอยู่ในรูปของสัญญาณ คล็อก Active High เอาต์พุตของแนนด์เกตที่จุด B เป็น 0 สัญญาณควบคุมที่ PR จะเป็น 0 การควบคุมให้ Q มีค่าเท่ากับ 1 เหมือนทาง Data
- เมื่อ Data ที่ป้อนเข้ามีค่าเท่ากับ 0 $C_i$ ควบคุมด้วย 1 สัญญาณที่จุด B เป็น 1 แนนด์เกตขาควบคุม CLR ได้รับเอาต์พุตเป็น 0 ทำให้สภาวะของฟลิปฟลอบถูกควบคุมการทำงานเป็น Reset ให้ Q มีค่าเท่ากับ0 เหมือนทาง Data ที่ป้อนเข้ามา
- การนำข้อมูลออกโดยควบคุม $C_o$ ให้มีค่าเป็น Active High พร้อมกับนำข้อมูลที่ Complement Q นำมาแนนด์กัน ค่าที่ได้จะเป็นไปตามค่า Q

# การเก็บข้อมูลขนาด 2 บิต

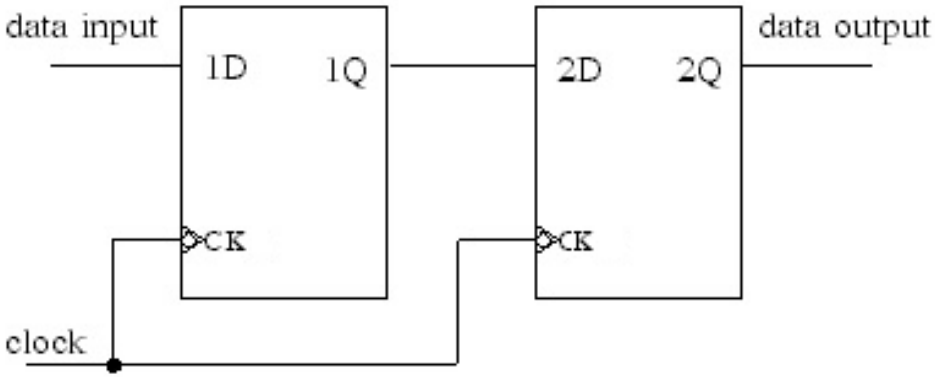
## โดยใช้ D Flip-Flop แบบขนาน

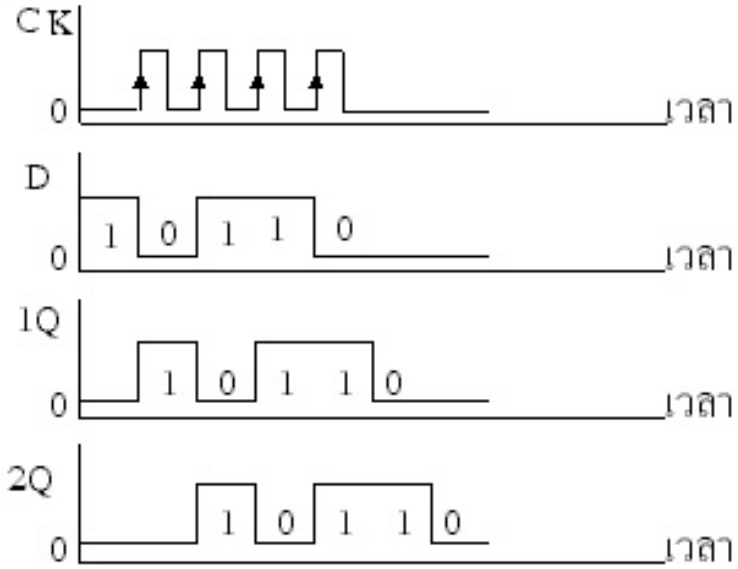


# การเลื่อนข้อมูลขนาด 2 บิต

## แบบอนุกรม



# อธิบาย

.

**Parallel In/Out**

- วงจรส่งข้อมูลโดยการให้ตัวเลื่อนข้อมูลแบบทางเข้าเป็นแบบขนาน
- ใช้สัญญาณควบคุมทางอินพุตเพียง คล็อก เดียว ในการส่งเข้า
- ในการส่งออกของข้อมูลก็ใช้คล็อก เดียว ซึ่งทำให้เกิดความประหยัดในเรื่องเวลา แต่จะใช้สายในการส่งเท่ากับจำนวนบิต

**Serial In/Out**

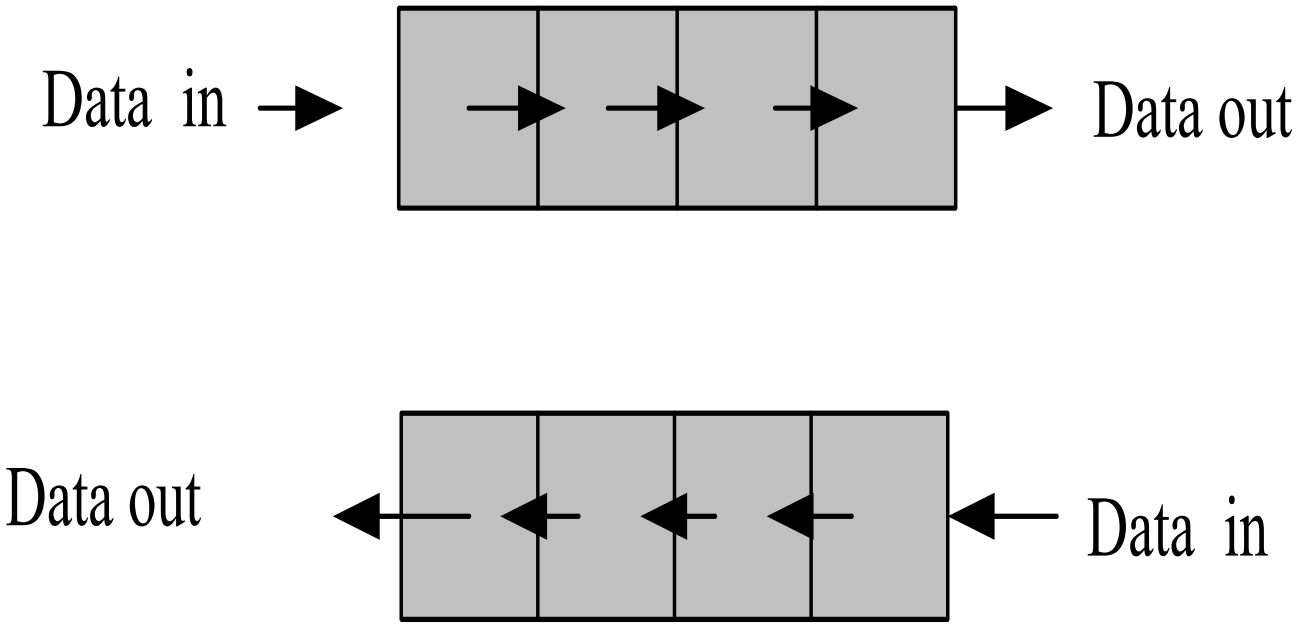
- วงจรส่งข้อมูลโดยการให้ตัวเลื่อนแบบทางเข้าเป็นแบบอนุกรม
- ใช้สัญญาณควบคุมทางอินพุตตามบิตที่ส่งในการส่งเข้า 2 บิต คล็อก ก็จะใช้ 2 พัลส์
- ในการส่งออกของข้อมูล 2 บิต คล็อก ก็จะใช้ 2 พัลส์เช่นกัน
- การส่งแบบนี้ก็มีประโยชน์ในเรื่องของสายที่ส่งสามารถใช้ร่วมกันได้ ทำให้ประหยัดในเรื่องสาย แต่จะเสียประโยชน์ในเรื่องเวลา
- Serial In Parallel Out , Parallel In Serial Out เป็นการนำ Register ทั้งสองแบบมารวมกันเพื่อนำคุณสมบัติบางอย่างมาใช้ให้เหมาะกับงาน

# อธิบาย

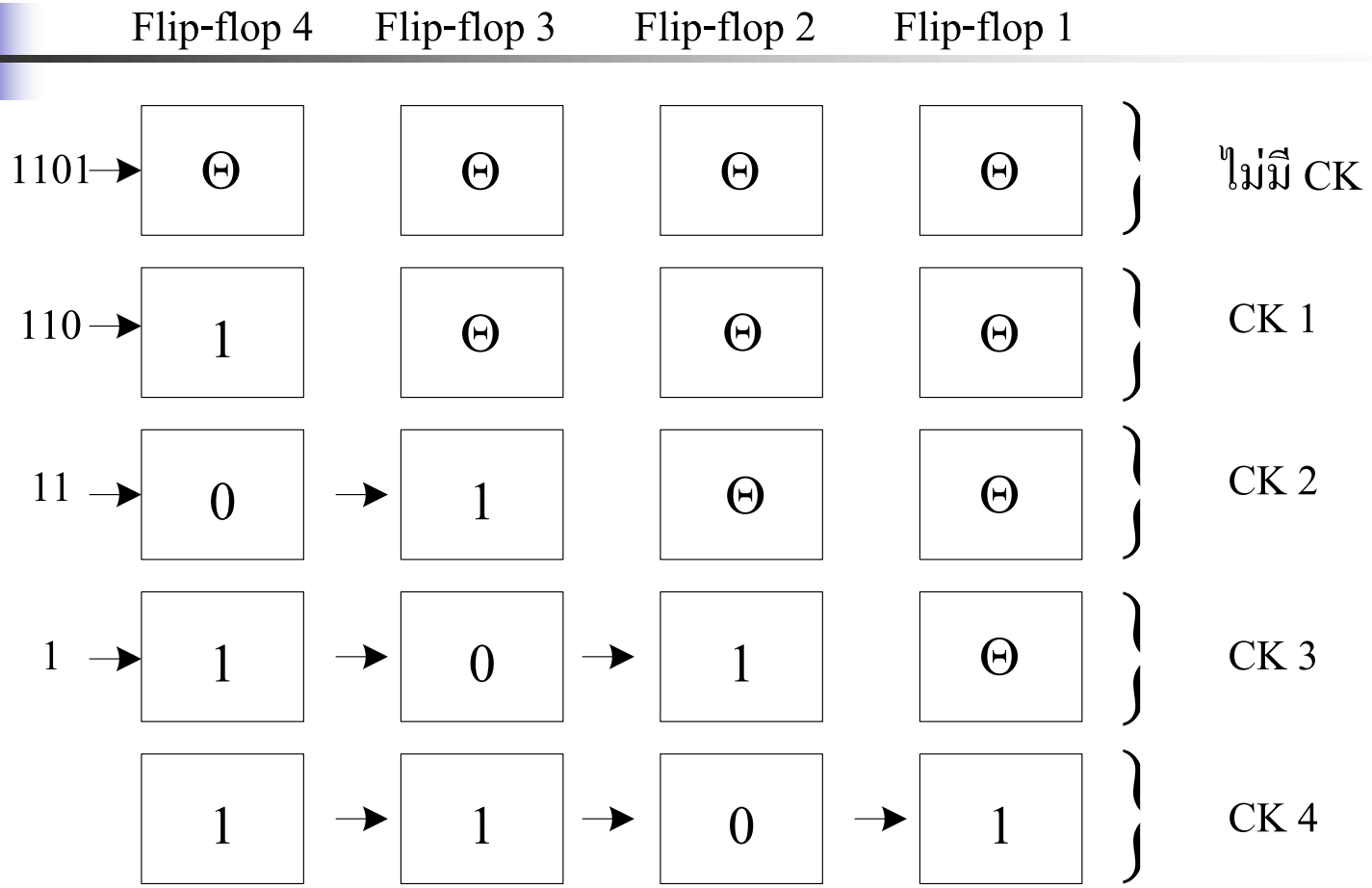
**Shift Register มี 5 แบบที่ครอบคลุมการทำงานทั้งหมดของวงจรดิจิตอล**

- Serial In , Serial Out
- Serial In , Parallel Out
- Parallel In , Parallel Out
- Parallel In , Serial Out
- Shift Around Register

# 651 การเลื่อนข้อมูลแบบ Serial In , Serial Out

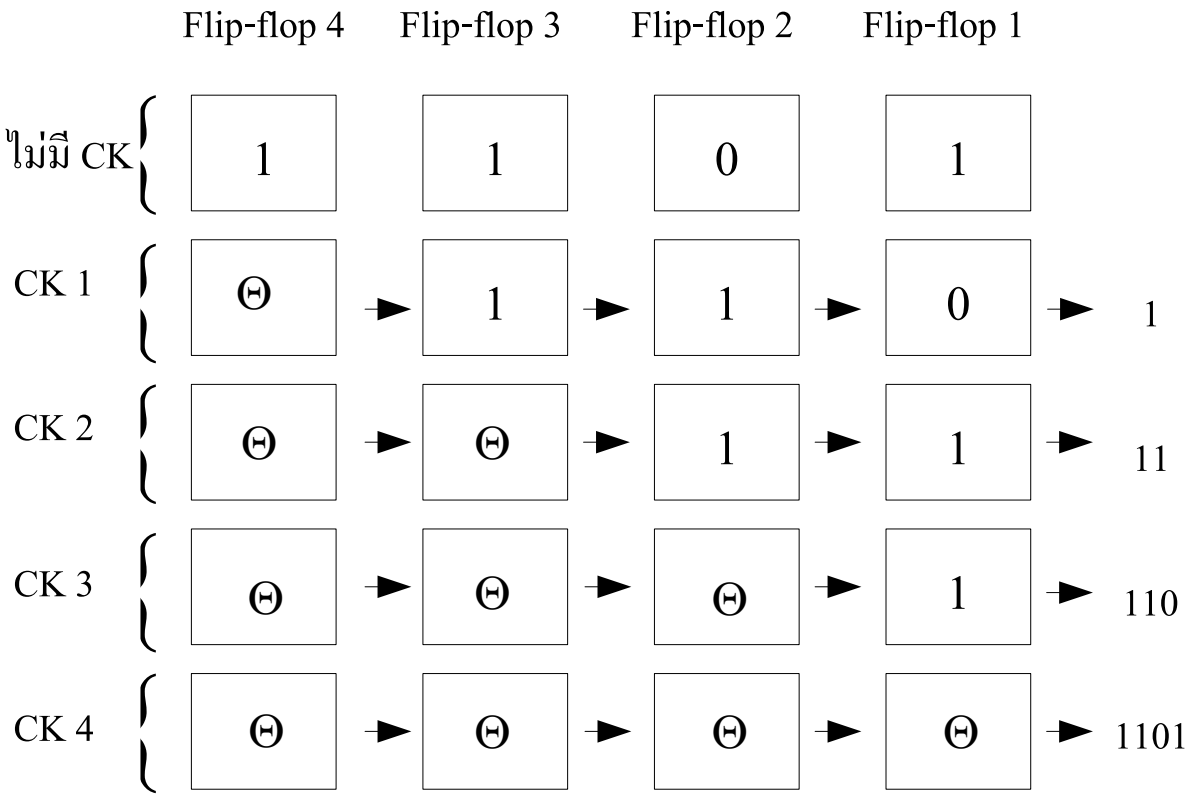


# การนำข้อมูลเข้าในตัวเก็บข้อมูล(Register)แต่ละตัว




การทำงานของ Register แบบนำเข้าของข้อมูลครั้งละบิต

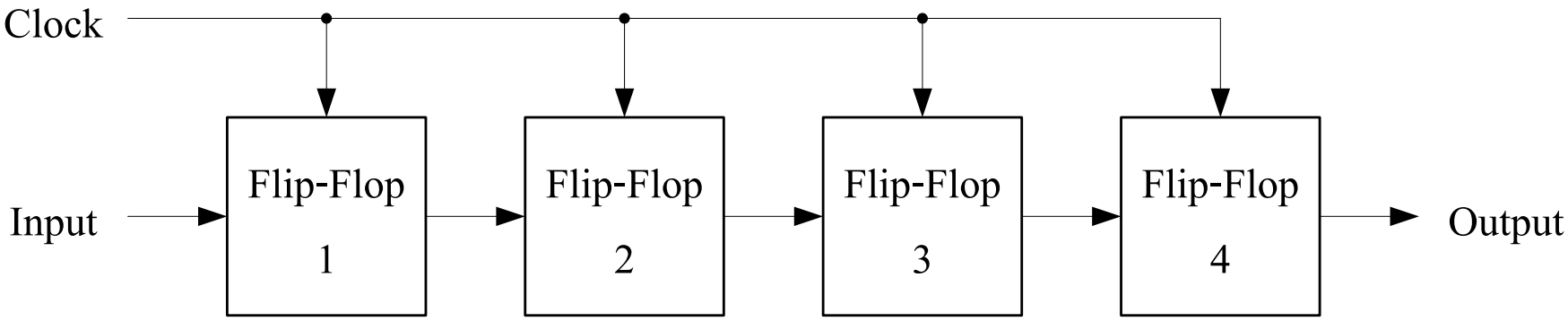
# การนำข้อมูลออกในตัวเก็บข้อมูลแต่ละตัว




แสดงการทำงานในช่วงของการนำสัญญาณข้อมูลออกจากตัว Register

# ลักษณะของตัวเลื่อนข้อมูลแบบนำเข้า – นำออก ครั้งละ 1 บิต

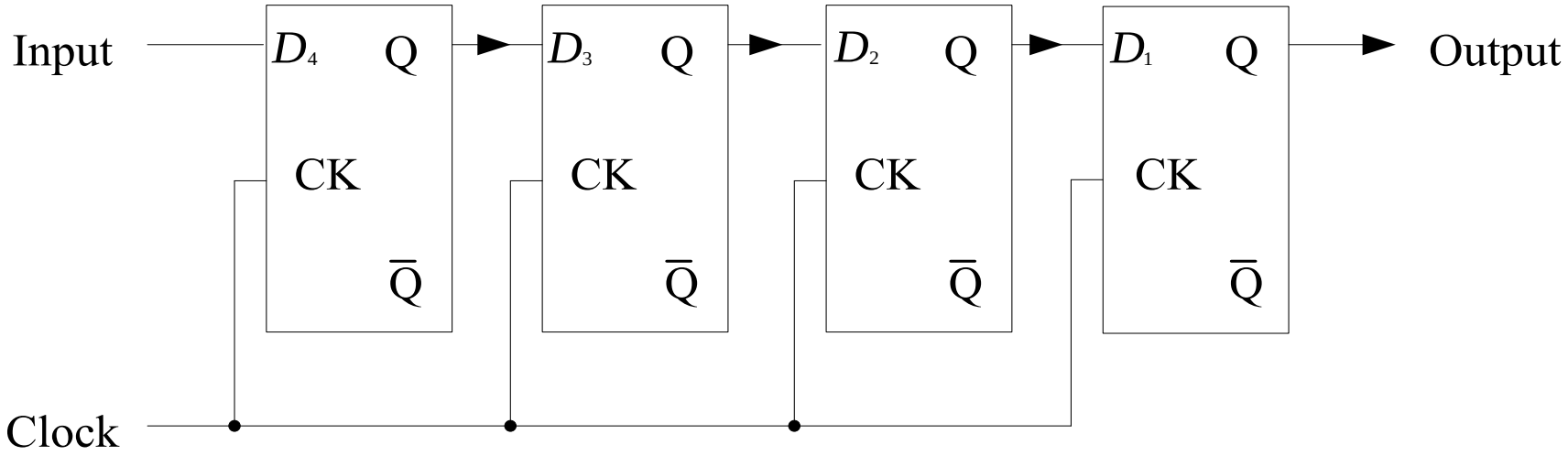
## (แบบอนุกรม)




ลักษณะของตัวเลื่อนข้อมูลแบบนำเข้า-ออกครั้งละ 1 บิตแบบอนุกรม

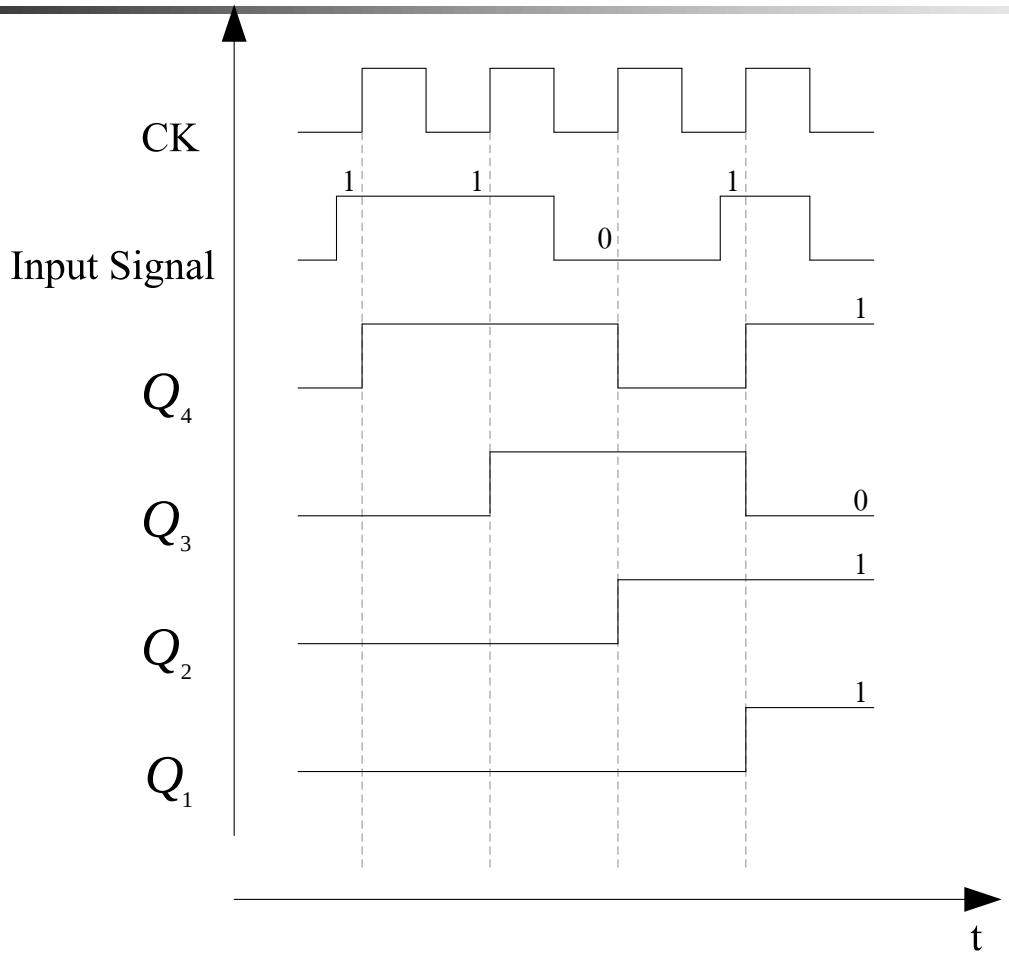
# รูปแบบวงจรเลื่อนข้อมูลชนิด D ฟลิบฟลอบ

## ต่อแบบอนุกรมทางเข้า อนุกรมทางออก




Register แบบนำเข้า-นำออกสัญญาณข้อมูลครั้งละบิต โดยใช้ D Flip-Flop จำนวน 4 บิต

# Timing Diagram การเก็บหรือพักสัญญาณข้อมูล



# 652 การเลื่อนข้อมูลไปทางขวา (Shift- Right Register)

จากวงจรการเลื่อนข้อมูลไปทางขวาการเลื่อนข้อมูลนี้สามารถดูได้จากตารางการทำงานของวงจร

การเลื่อนข้อมูลเข้า (1101) ไปทางขวา 4 ขั้น

| ชิฟต์พัลซ์ | A | B | C | D |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 1 | 0 | 1 | 0 |
| 4 | 1 | 1 | 0 | 1 |

การเลื่อนข้อมูลไปออกทางขวาอีก 4 ขั้น

| ชิฟต์พัลซ์ | A | B | C | D | |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 2 | 0 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 | 1 |

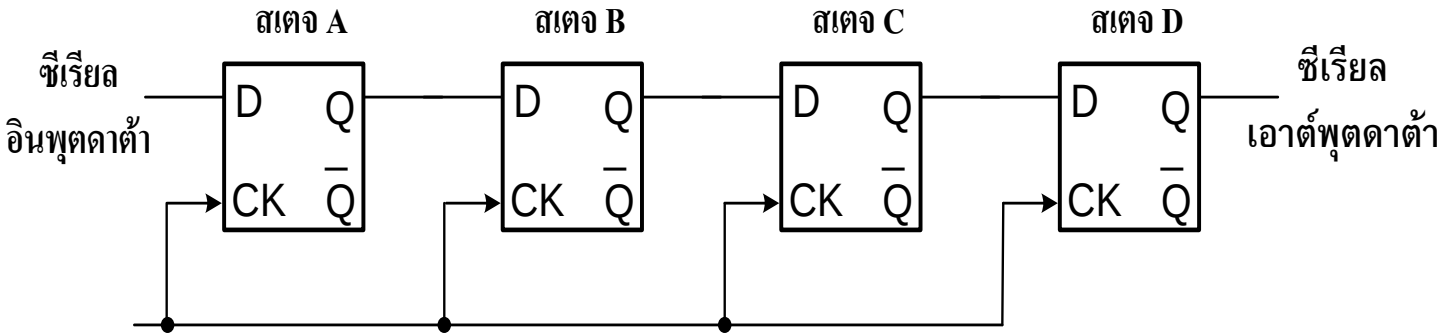
# อธิบาย

- การเลื่อนข้อมูลขาเข้า
- การเลื่อนข้อมูลจากขวาไปซ้ายข้อมูลที่ทำการเลื่อนคือ (1101)
- คล็อกแรกส่งมาข้อมูลบิตแรกทางขวามือ($L_{SB}$)เข้าที่ตัวเก็บข้อมูลAได้$Q_A$
- คล็อกที่2ส่งมาข้อมูลบิตใหม่เข้าที่Aข้อมูลเดิมจากAส่งไปยัง Bได้$Q_A$,$Q_B$
- คล็อกที่ 3 ส่งมาข้อมูลบิตที่สามส่งมาใหม่เข้าที่ A ข้อมูลเดิมจาก A ส่งไปยัง B ข้อมูลเดิมจาก B ส่งไปที่ C ได้ $Q_A$,$Q_B$,$Q_C$
- คล็อกที่ 4 ส่งมาข้อมูลบิตที่สี่ส่งมาใหม่เข้าที่ A ข้อมูลเดิมจากA ส่งไปยัง B ข้อมูลเดิมจาก B ส่งไปที่ C ข้อมูลเดิมจากCส่งไปที่Dได้$Q_A$,$Q_B$,$Q_C$,$Q_D$

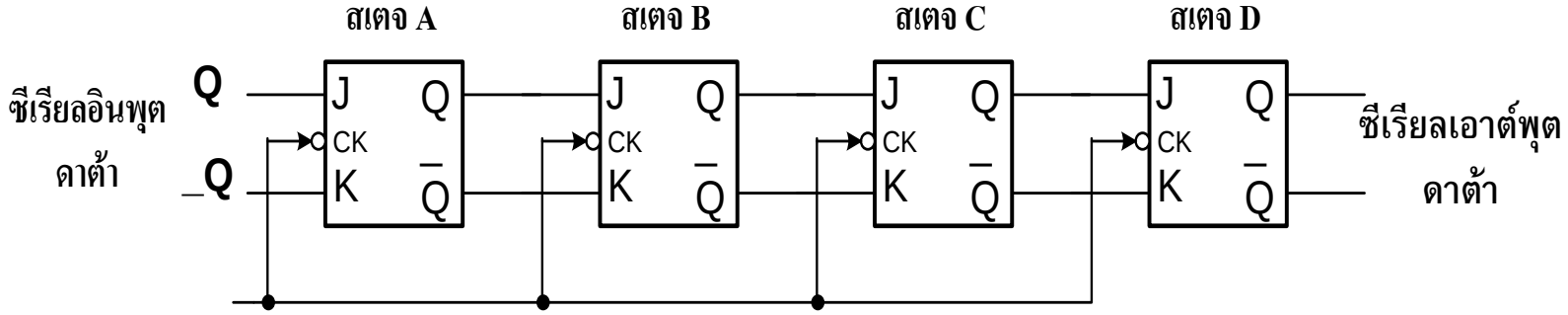
# อธิบาย(ต่อ)

- การเลื่อนข้อมูลขาออกไปทางขวามือ
- การเลื่อนข้อมูลจากขวาไปซ้ายข้อมูลที่ทำการเลื่อนออกคือ (1101)
- คล็อกแรกส่งมาข้อมูลบิตแรกทางขวามือ ($L_{SB}$) ออกจากตัวเก็บข้อมูลD ได้1
- คล็อกที่2ส่งมาข้อมูลบิตส่งมาออกที่ D เป็น 0 ข้อมูลเดิมจาก B ส่งไปยัง C ข้อมูลเดิมจากCส่งไปที่Dเพื่อเตรียมส่งออกในคล็อกที่3 ได้รับเป็น 0
- คล็อกที่ 3 ส่งมาข้อมูลบิตที่สามถูกส่งออกจาก D มีค่าเป็น1ข้อมูลเดิมจาก C ส่งไปยัง Dเป็นข้อมูลของบิตที่ 4 เป็นบิตสุดท้าย
- คล็อกที่4ส่งมาข้อมูลบิตที่4ถูกส่งออกจากDมีค่าเป็น1ดูตารางประกอบการอธิบาย
- ครบรอบการเลื่อนข้อมูลเข้าและออกข้อมูลในตัวเก็บข้อมูลทั้งหมดถูกรีเซ็ตเป็น0

# วงจรการเลื่อนข้อมูลจากซ้ายไปทางขวา (Shift- Right Register)




(ก) การเลื่อนข้อมูลจากซ้ายไปขวาโดยใช้ D ฟลิปฟลอป




(ข) การเลื่อนข้อมูลจากซ้ายไปขวาโดยใช้ J- K ฟลิปฟลอป

# อธิบาย

- จากรูป ก. เป็นการเลื่อนข้อมูลจากซ้ายไปขวาโดยใช้ D ฟลิบฟลอบ
- การทำงานคล็อกแรกจะเลื่อนข้อมูลบิตแรกจากอินพุตเข้าสู่ฟลิบฟลอบ A ได้ $Q_A$
- คล็อกที่สองเลื่อนจาก $Q_A$ ไปยัง $Q_B$ พร้อมกับข้อมูลทางอินพุตป้อนเข้าที่ A จะเป็นข้อมูลบิตที่สอง $Q_A$,$Q_B$
- คล็อกที่สามจะเลื่อนข้อมูลจาก $Q_A$ ไป $Q_B$ และข้อมูล $Q_B$ ไปยัง$Q_C$ พร้อมกับข้อมูลทางอินพุตป้อนเข้าที่ A จะเป็นข้อมูลบิตที่สาม $Q_A$,$Q_B$,$Q_C$
- คล็อกที่สี่จะเลื่อนข้อมูลจาก $Q_A$ ไปยัง $Q_B$ และข้อมูล $Q_B$ ไปยัง $Q_C$ ,ข้อมูล $Q_C$ ไปยัง $Q_D$พร้อมทั้งข้อมูลบิตสุดท้ายได้ $Q_A$,$Q_B$,$Q_C$,$Q_D$
- การเลื่อนข้อมูลจึงใช้คล็อกจำนวน4พัลส์จึงจะเลื่อนข้อมูลทั้ง4 บิตอย่างสมบูรณ์
- จากรูป ข. เป็นการนำ J-K ฟลิบฟลอบมาต่อในรูปแบบ T ฟลิบฟลอบ
- การทำงานของ J-K ฟลิบฟลอบ มีขั้นตอนเหมือนกับ D ฟลิบฟลอบที่กล่าวมา

# 653 การเลื่อนข้อมูลไปทางซ้าย (Shift -Left Register)

| ชิฟต์พัลซ์ | D | C | B | A |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 1 | 0 |
| 4 | 1 | 1 | 0 | 1 |

การเลื่อนข้อมูลเข้าไปทางซ้าย 4 ขั้น

การเลื่อนข้อมูลออกไปทางซ้าย 4 ขั้น

| ชิฟต์พัลซ์ | D | C | B | A |
|---|---|---|---|---|
| 4 | 1 | 1 | 0 | 1 |
| 5 | 1 | 0 | 1 | 0 |
| 6 | 0 | 1 | 0 | 0 |
| 7 | 1 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 |

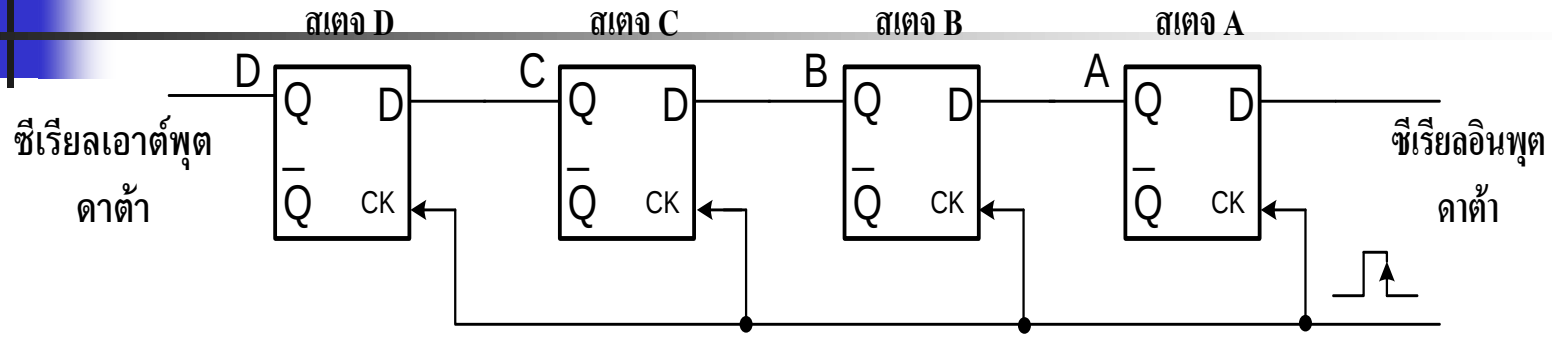
# อธิบาย

- วงจรการเลื่อนข้อมูลเข้าไปทางซ้ายจำนวน 4 บิต ข้อมูลที่ทำการเลื่อนคือ 1101
- เริ่มต้นที่คล็อกที่ 0 คือไม่มีสัญญาณคล็อก ทำการรีเซ็ตฟลิบฟลอบทุกค่าให้เป็น 0
- คล็อกลูกที่1 ป้อนเข้าขณะที่อินพุตมีค่าเท่ากับ 1 ข้อมูล 1 ถูกเก็บที่ฟลิบฟลอบ A
- คล็อกลูกที่2 ป้อนเข้ามาขณะที่อินพุตยังคงเป็น 1 ข้อมูลจาก A จะส่งไปยัง B ข้อมูลใหม่จะถูกแทนที่ใน A ซึ่งยังมีค่าเท่ากับ 1 อยู่
- คล็อกลูกที่3 ส่งมาข้อมูลทางอินพุตเป็น0 ส่งไปยัง A ข้อมูลที่ A ส่งไปยัง B ข้อมูลที่ B ส่งไปยัง C ได้เป็น (CBA = 110)
- คล็อกลูกที่4 ป้อนเข้าอินพุตมีค่าเท่ากับ 1 ส่งไปยัง A เป็น 1 B จะถูกแทนด้วยข้อมูล A เดิมคือ 0 C จะถูกแทนด้วยข้อมูล B และ D จะถูกแทนด้วยข้อมูล C เป็น 1 ,1 ตามลำดับ
- สิ้นคล็อกลูกที่ 4 เป็นการเก็บข้อมูล 1101 ไว้

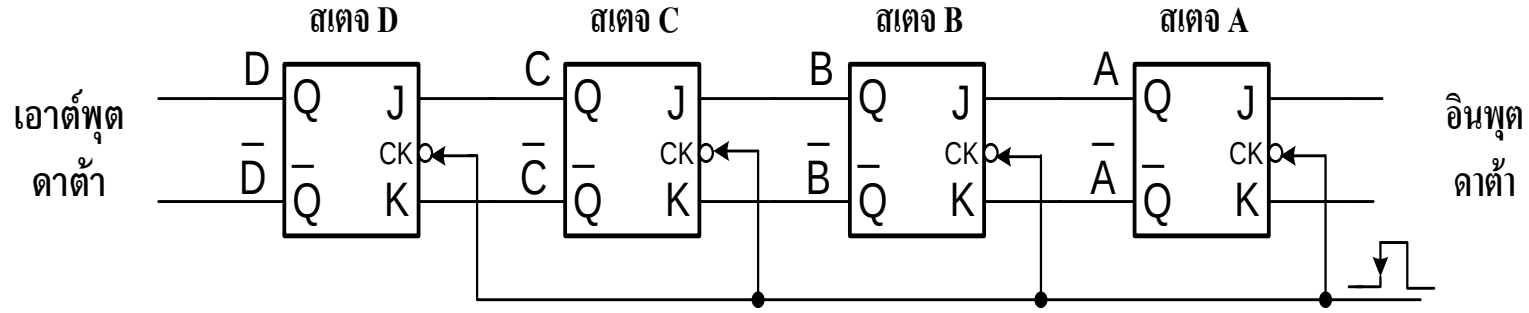
# อธิบาย (ต่อ)

- การเลื่อนข้อมูลออกจากตารางเลื่อนข้อมูลออก
- คล็อกลูกที่5 ป้อนเข้าข้อมูลจาก D ที่เป็น 1 จะถูกส่งออกไป ข้อมูลจาก C ไปแทน D และข้อมูลจาก Bไปแทน Cข้อมูลจากAไปแทนBเลื่อนไปทางซ้ายมือได้ค่าเป็น **(1010)**
- คล็อกลูกที่6 ป้อนเข้าข้อมูลจาก Dที่เป็น1จะถูกส่งออกไป ข้อมูลจาก Cไปแทน D และข้อมูลจากBไปแทนCข้อมูลจากAไปแทนBเลื่อนไปทางซ้ายมือได้ค่าเป็น **(0100)**
- คล็อกลูกที่7 ป้อนเข้าข้อมูลจาก D ที่เป็น 0 จะถูกส่งออกไป ข้อมูลจาก Cไปแทน Dและข้อมูลจากBไปแทนCข้อมูลจากA ไปแทนBเลื่อนไปทางซ้ายมือได้ค่าเป็น **(1000)**
- คล็อกลูกที่8 ป้อนเข้าข้อมูลจากDที่เป็น 1 จะถูกส่งออกไป ข้อมูลจาก Cไปแทน Dและข้อมูลจากBไปแทนCข้อมูลจากAไปแทน Bเลื่อนไปทางซ้ายมือได้ค่าเป็น **(0000)** สิ้นสุดคล็อกลูกที่8 ค่าของตัวเก็บข้อมูลเป็น 0 ทั้งหมด

# วงจรการเลื่อนข้อมูลจากขวาไปทางซ้าย




(ก) วงจรการเลื่อนข้อมูลจากขวาไปซ้ายโดยใช้ D ฟลิปฟลอป



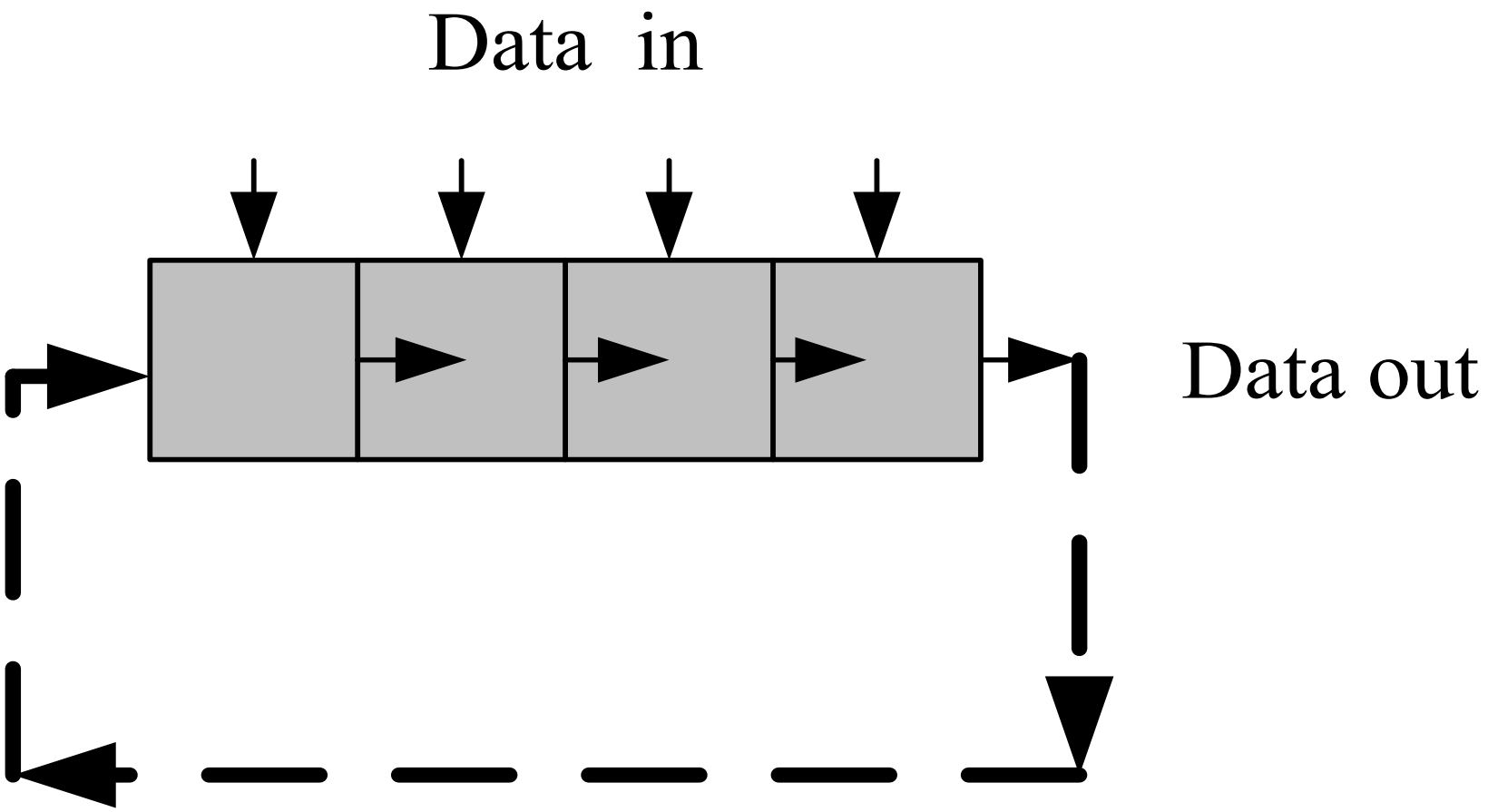

(ข) วงจรการเลื่อนข้อมูลจากขวาไปซ้ายโดยใช้ J- K ฟลิปฟลอป

# อธิบาย

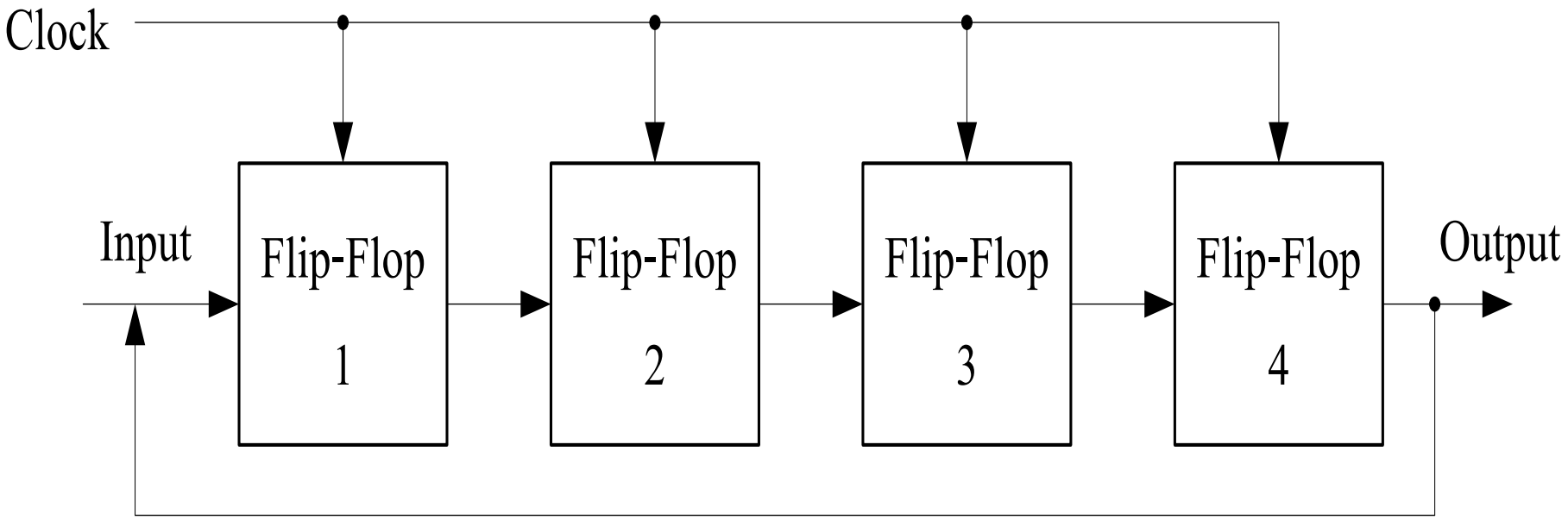
- วงจรจากรูป ก. เป็นการเลื่อนข้อมูลจากขวาไปซ้ายโดยใช้ D ฟลิบฟลอบ
- การทำงานคล็อกแรกจะเลื่อนข้อมูลบิตแรกจากอินพุตเข้าสู่ฟลิบฟลอบ A
- คล็อกที่สองเลื่อนจาก $Q_A$ ไปยัง $Q_B$ พร้อมกับข้อมูลทางอินพุตป้อนเข้าที่ A จะเป็นข้อมูลบิตที่สอง
- คล็อกที่สามจะเลื่อนข้อมูลจาก $Q_A$ ไป $Q_B$ และข้อมูล $Q_B$ ไปยัง $Q_C$ พร้อมกับข้อมูลทางอินพุตป้อนเข้าที่ A จะเป็นข้อมูลบิตที่สาม
- คล็อกที่สี่จะเลื่อนข้อมูลจาก $Q_A$ ไปยัง $Q_B$และข้อมูล $Q_B$ไปยัง $Q_C$ ,ข้อมูล $Q_C$ ไปยัง $Q_D$พร้อมทั้งข้อมูลบิตสุดท้าย
- การเลื่อนข้อมูลจึงใช้คล็อกจำนวน 4พัลส์จึงจะเลื่อนข้อมูลทั้ง4บิตอย่างสมบูรณ์
- วงจรจากรูป ข. เป็นการนำ J-K ฟลิบฟลอบมาต่อในรูปแบบTฟลิบฟลอบ
- การทำงานของ J-K ฟลิบฟลอบ มีขั้นตอนเหมือนกับ D ฟลิบฟลอบที่กล่าวมา

# 654 การเลื่อนข้อมูลแบบวนรอบกลับ (Shift Around Register)

## Shift Around Register ป้อนข้อมูลเข้าแบบขนาน

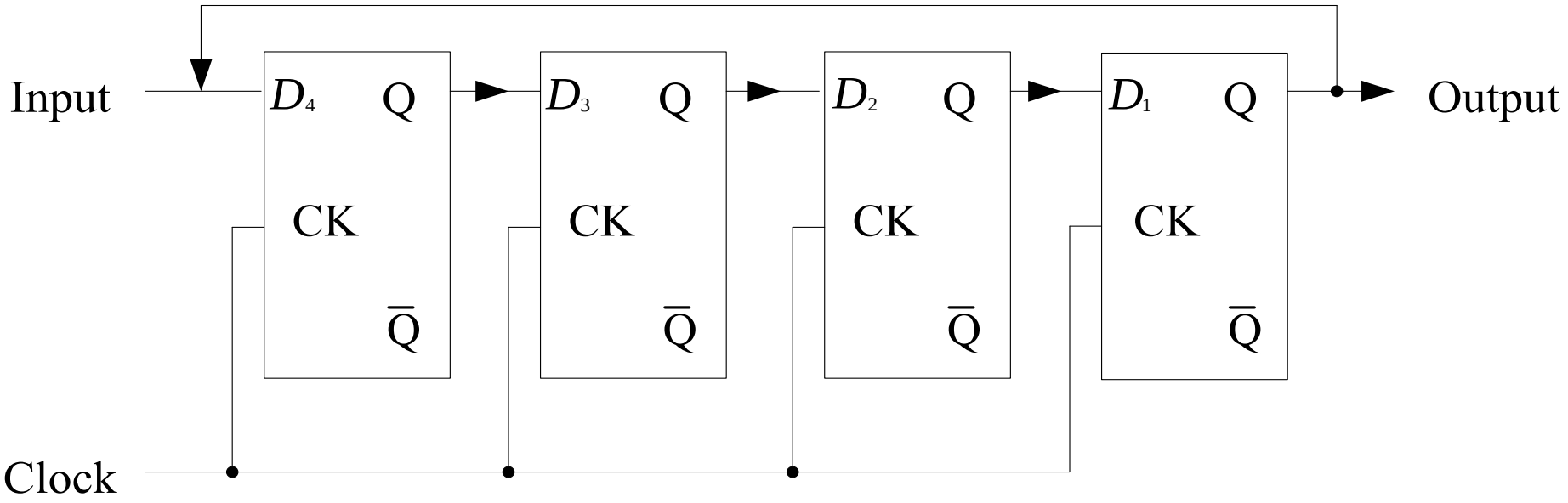


# Shift Around Register ป้อนข้อมูลเข้าแบบอนุกรม




ลักษณะของ Register แบบนำเข้า - นำออกสัญญาณข้อมูลครั้งละบิตแบบวนรอบ

# Register แบบนำเข้า–นำออก ในลักษณะวนรอบโดยใช้ D Flip-Flop

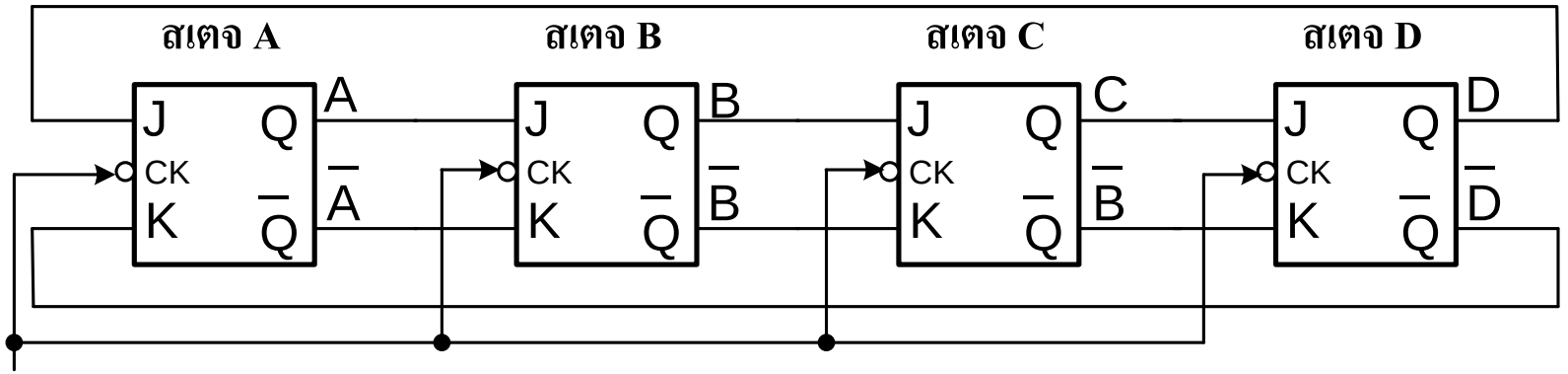



Register แบบนำเข้า-นำออกครั้งละบิตในลักษณะวนรอบโดยใช้ D Flip-Flop

# อธิบาย

- การเลื่อนข้อมูลแบบใช้ D ฟลิบฟลอบในรูปแบบวนรอบ
- การป้อนของข้อมูลเข้าเป็นแบบ Serial ขนาด 4บิตป้อนเข้าจนครบเพียงครั้งเดียว
- ข้อมูลที่ส่งภายในจะส่งซึ่งกันและกันแบบวนรอบ
- ข้อมูลจะถูกส่งภายใต้คล็อก จาก $D_1$ ไปยัง $D_4$ จาก $D_4$ไปยัง $D_3$ จาก $D_3$ ไปยัง $D_2$ จาก $D_2$ไปยัง $D_1$ ข้อมูลจะวนอย่างนี้ตลอด
- การเปลี่ยนข้อมูลภายในโดยการเซ็ต หรือรีเซ็ตที่ฟลิบฟลอบ
- การเลื่อนข้อมูลแบบนี้เป็นการรักษาข้อมูลที่ส่งไปทางเอาต์พุตที่ไม่ให้สูญหายขณะการส่งออกก็จะนำไปเก็บไว้ในรูปแบบวนด้วย

# วงจรการเลื่อนข้อมูลแบบวนรอบกลับ




ชิฟต์พัลส์

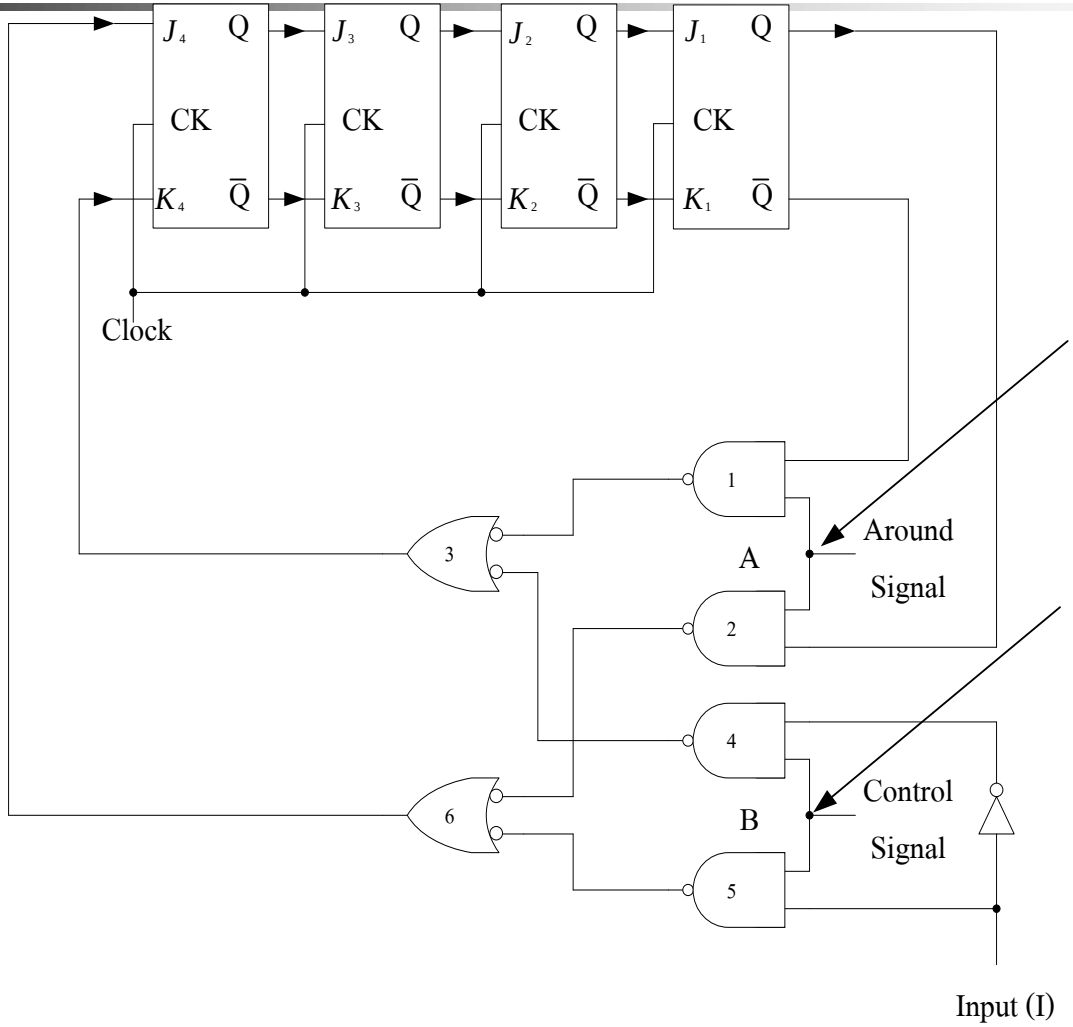
## การเลื่อนข้อมูลแบบวนรอบกลับ 4 ขั้น

| ชิฟต์พัลซ์ | A | B | C | D |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |
| 2 | 0 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 1 |
| 4 | 1 | 1 | 0 | 1 |

# อธิบาย

- การทำงานของวงจรเลื่อนข้อมูลแบบวนรอบแบบใช้ J-K
- ในสภาวะเริ่มต้นเรากำหนดค่าเก็บข้อมูลเป็น 1101 ด้วยการควบคุมการเซ็ต หรือรีเซ็ต ของ PR หรือ CLR ซึ่งกล่าวมาแล้ว
- ขณะไม่มีพลัซ์ควบคุมจะเป็น 1101 เป็นค่าข้อมูลในฟลิบฟลอบ A,B,C,D ตามลำดับ
- เมื่อมีพลัซ์แรกเข้าข้อมูลจาก D จะถ่ายไปยัง A และข้อมูล A จะถูกถ่าย ไปยัง B , C, D ตามลำดับ ในช่วงคล็อกแรก
- ในช่วงคล็อกถัดมาก็จะมีการถ่ายข้อมูลดังเช่นคล็อกแรกจากข้อมูลทางซ้ายมือไปยังขวามือจนครบรอบ คือ 4 ลูก
- ข้อมูลที่เกิดจากการวนครบลูป จะเป็นเหมือนข้อมูลครั้งแรก เรียกว่าข้อมูล 1 Cycle

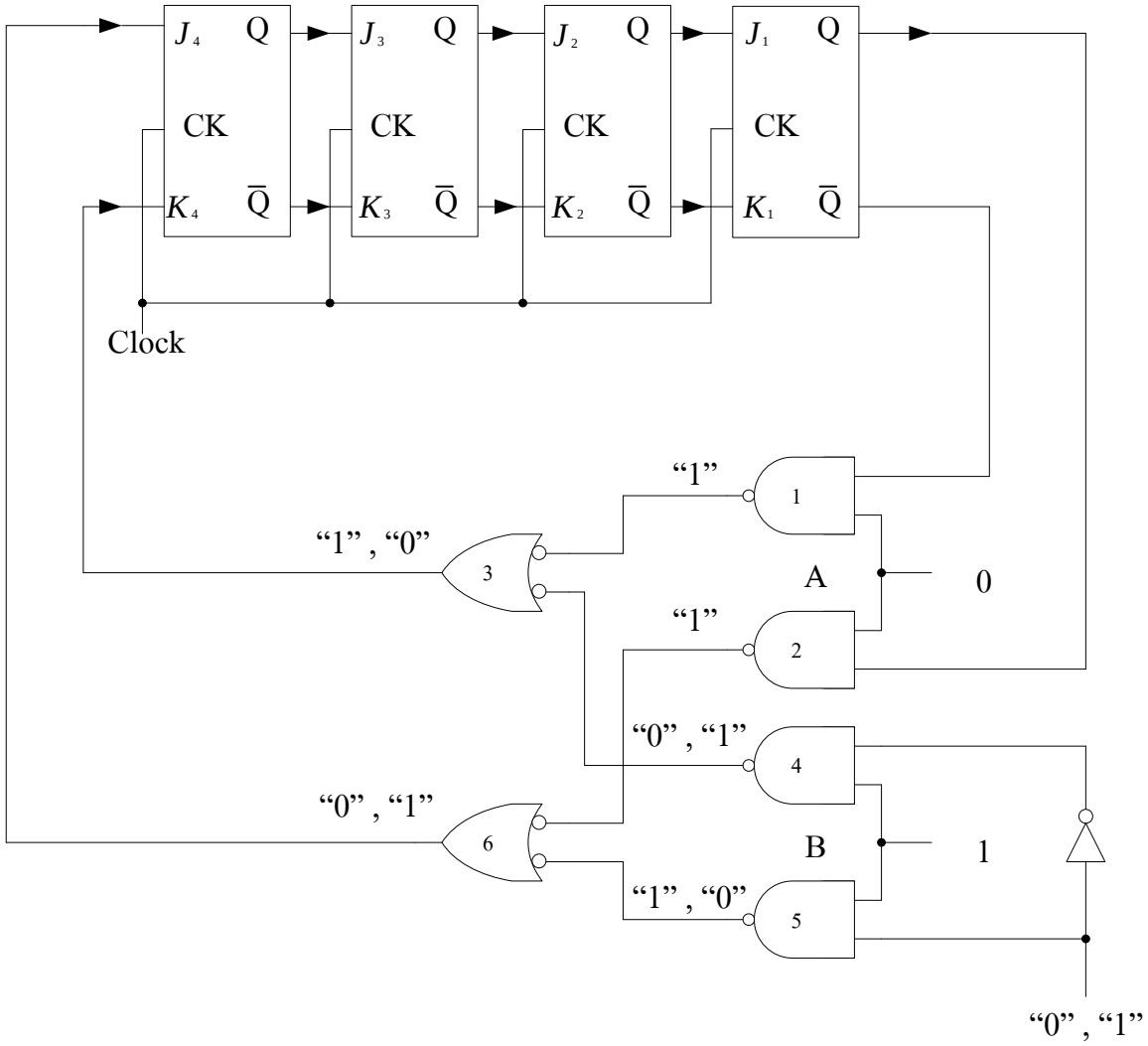
# วงจรการทำงานของวงจรแบบ Shift Around Register



# อธิบาย

- การทำงานของRegister ในรูปเป็นแบบอนุกรมเข้า และอนุกรมออก โดยใช้ J-K ฟลิบฟลอบ
- สามารถสร้างจาก J-K ฟลิบฟลอบเป็นการเก็บข้อมูลชั่วคราวไม่ให้สูญหาย เป็นแบบวนรอบตัว พร้อมกับส่งข้อมูลไปใช้งานด้วย สามารถนำค่าออกจาก $Q_1$
- การทำงานของในรูปแบบการทำงานที่เป็นตัวเก็บข้อมูล
- การรับข้อมูลทางอินพุต (1) Gate 5,6 จะทำงานให้เอาต์พุตเป็น1 ป้อนไปยังอินพุต $J_4$ เป็นการเซตฟลิบฟลอบ เมื่ออินพุตเป็น 1 พร้อมสัญญาณคล็อก
- การรับข้อมูลทางอินพุต (0) Gate 4,3 จะทำงานให้เอาต์พุตเป็น 1 ป้อนไปยังอินพุต $K_4$ จะทำการรีเซต เข้าเมื่ออินพุตเป็น 0 พร้อมสัญญาณคล็อก
- ในทำนองเดียวกันเราสามารถนำเอาต์พุตจากฟลิบฟลอบ 1 มาควบคุมการทำงานในรูปแบบวนรอบตัว โดยผ่าน Gate1,2

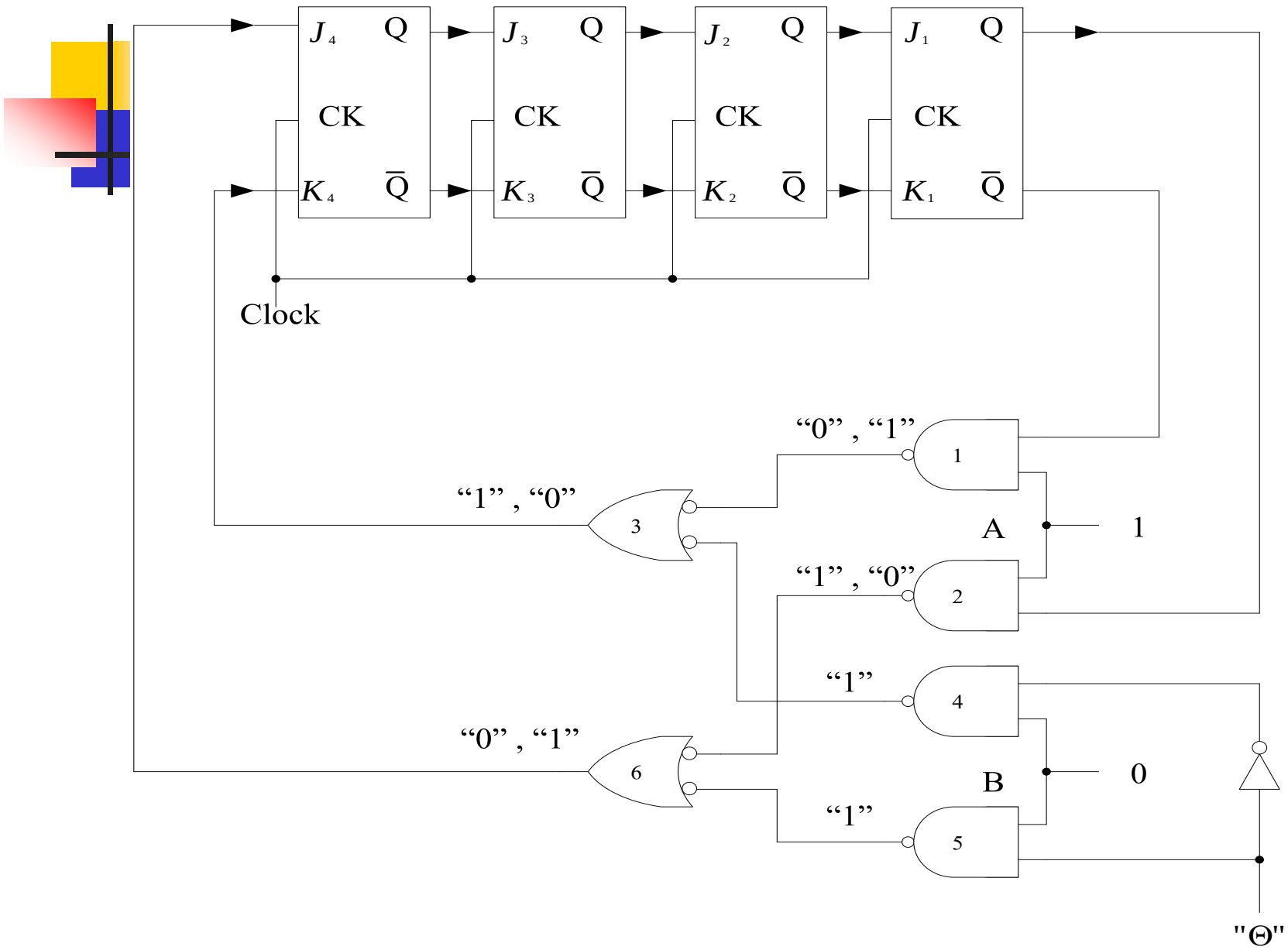
# การทำงานของ Register ในโหมดปกติ



# อธิบาย

- การทำงานของ Register ให้มีการทำงานในโหมดปกติจะเริ่มทำงานเมื่อคล็อกเปลี่ยนจาก 0 ไปเป็น 1 พร้อมขณะที่มีการควบคุมดังนี้
- ป้อนสัญญาณ Control Signal เป็น 1 ที่จุด B จุด A ป้อนค่า 0 เกต 4,5 พร้อมที่จะทำงาน เพื่อป้อนสัญญาณให้กับเกต 3,6 ทำงานต่อ
- การทำงานของเกต 3 ซึ่งจะทำการรีเซ็ต ฟลิบฟลอบ 4 จะทำงานก็ต่อเมื่อได้อินพุตเป็น 0 ซึ่งถูกป้อนมาจากเอาต์พุตของเกต 4 นั่นหมายความว่าอินพุตของข้อมูลจะต้องป้อน 0
- การทำงานของเกต 6 ซึ่งจะทำการเซ็ต ฟลิบฟลอบ 4 จะทำงานก็ต่อเมื่อมีอินพุตเป็น 0 ซึ่งถูกป้อนมาจากเอาต์พุตเกต 5 นั่นหมายความว่าอินพุตข้อมูลจะต้องมีค่าเป็น 1
- การทำงานนี้จะส่งข้อมูลจากฟลิปฟลอบ 4 ไปยังฟลิบฟลอบ 3 …..2…..1 เมื่อครบรอบจำนวน 4 ลูก
- โหมดการทำงานนี้เป็นการนำข้อมูลป้อนเข้าทางอินพุต ควบคุมการทำงานที่ B

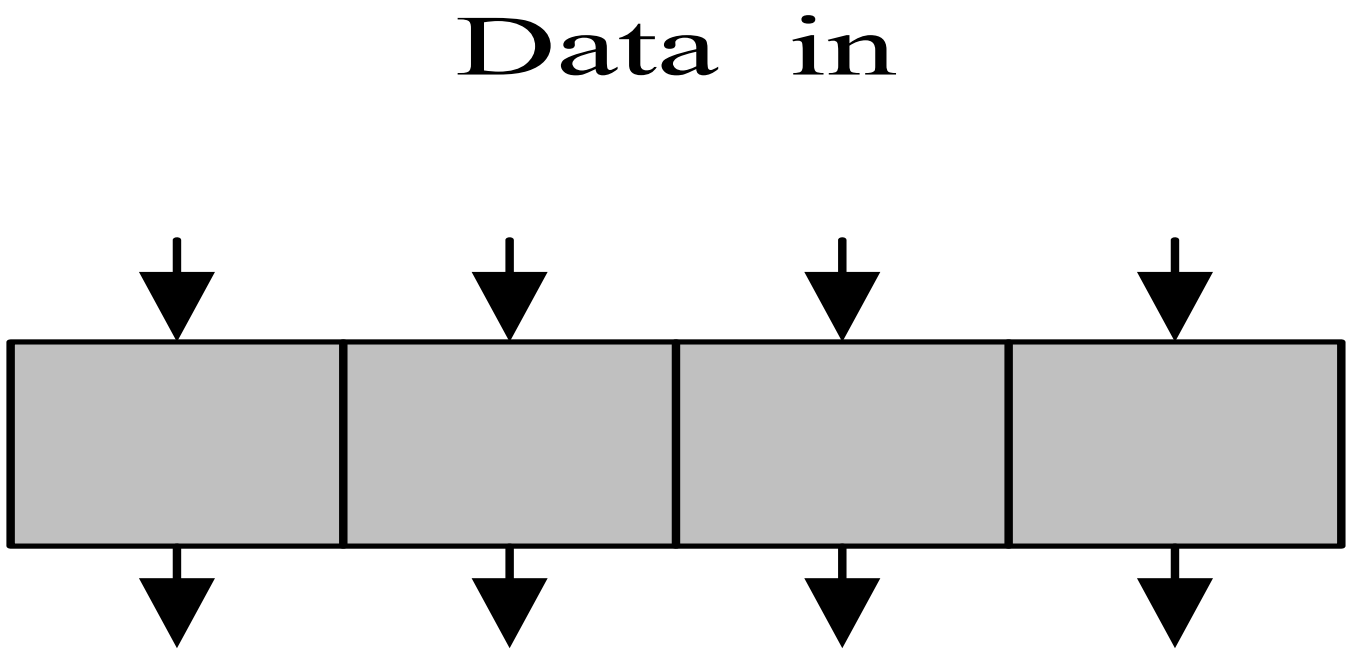
# การทำงานของ Register ในโหมดวงรอบ



# อธิบาย

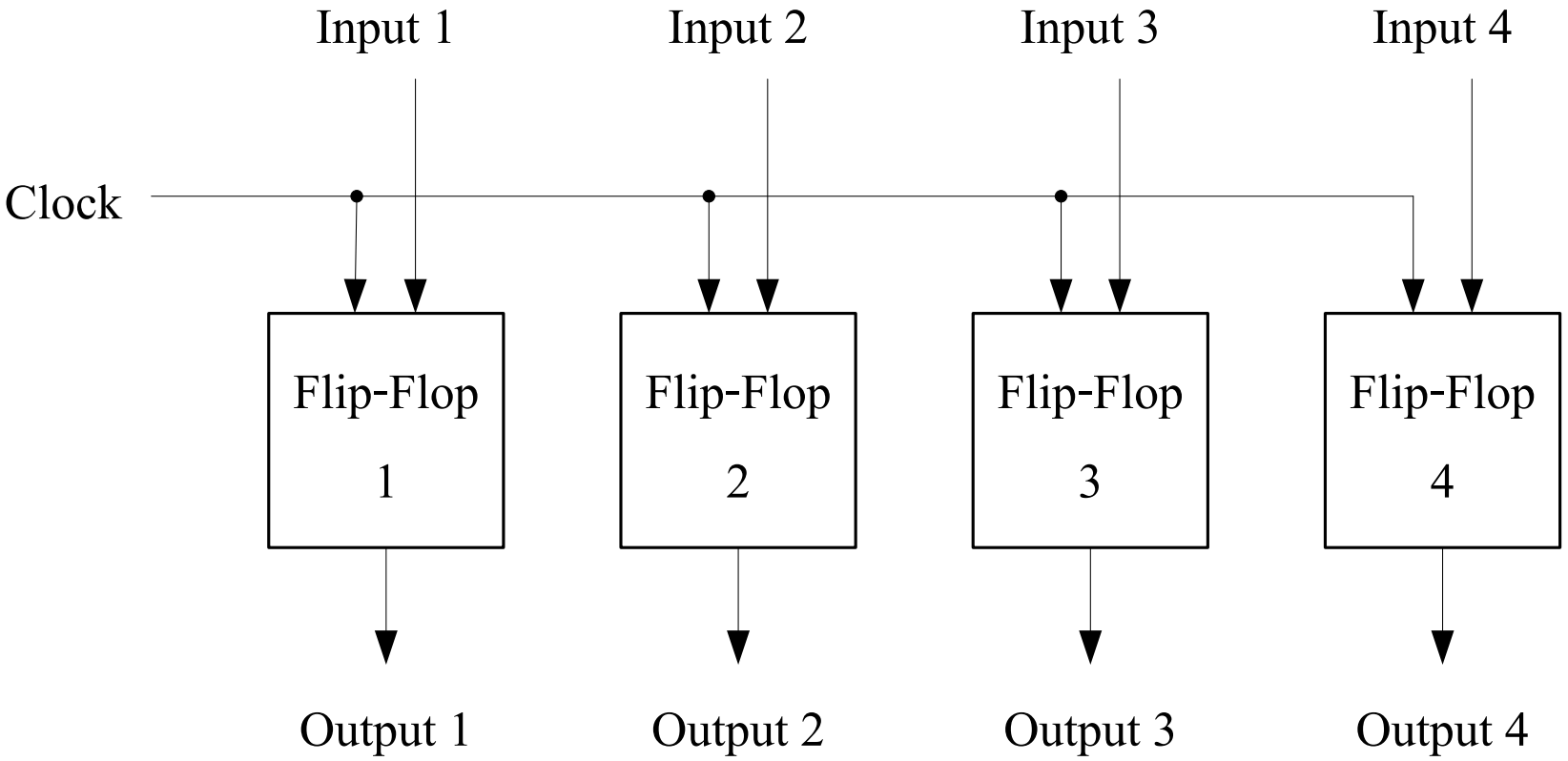
- การทำงานในโหมดวนรอบเพื่อทำการส่งข้อมูล จุด A จะป้อนค่า 1 จุด B จะป้อนค่าเป็น 0
- ข้อมูลที่เกิดที่เอาต์พุตของฟลิบฟลอบ 1 จะถูกป้อนไปยังเกต 1,2 ที่มีค่า Complement ซึ่งกันและกัน
- เกต 1 จะทำงานร่วมกับเกต 3 เพื่อทำการรีเซ็ตฟลิบฟลอบ 4 เมื่อป้อนสัญญาณที่ Q ที่มีค่าเท่ากับ 0 $\overline{Q} = 1$ เอาต์พุตเกต 3 จะมีค่าเป็น 1 พร้อมกับคล็อกทางอินพุตของฟลิบฟลอบ 4 ค่าเอาต์พุตจะเป็น 0
- ในทำนองเดียวกันสัญญาณทางเกต 2 , 6 จะทำหน้าที่เป็นการเซ็ตฟลิบฟลอบ 4
- เมื่อป้อน $Q = 1, \overline{Q} = 0$ เกต จะมีค่าเป็น 1 สัญญาณจะควบคุมที่J ของฟลิบฟลอบ 4 จะทำการเซ็ตค่า $Q_4$

# 655 การเลื่อนข้อมูลแบบขนาน Parallel In , Parallel Out



# การเลื่อนข้อมูลเข้าและออกแบบขนาน

## ( parallel in parallel out shift register )



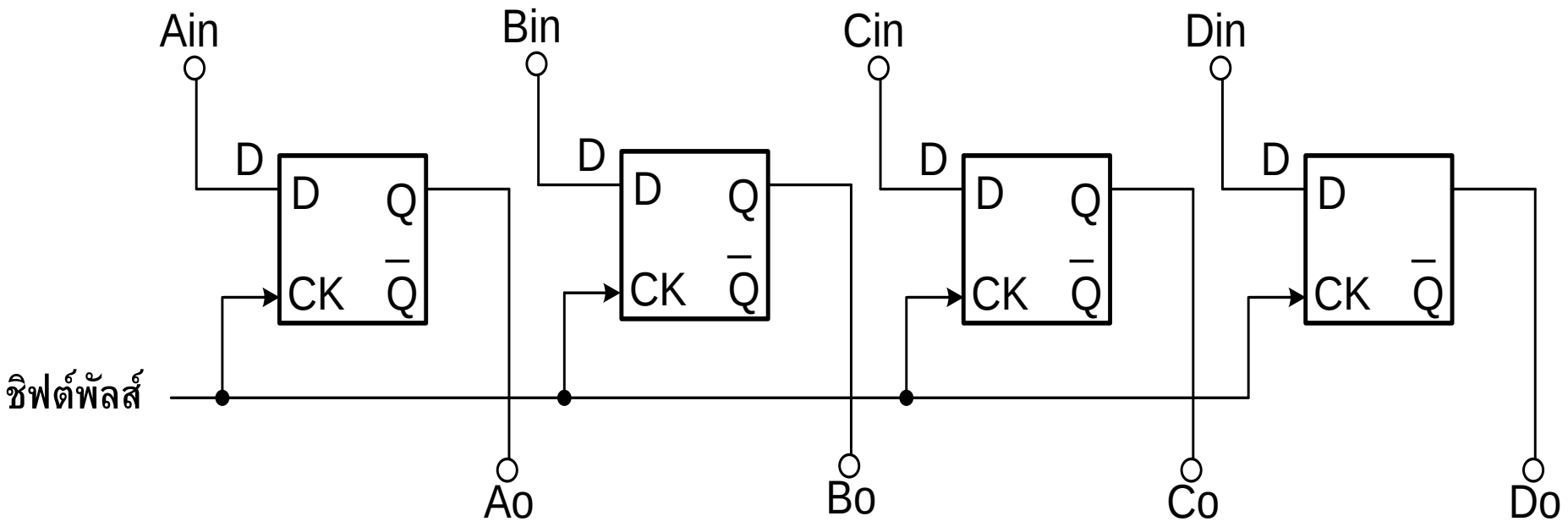

ลักษณะของ Register แบบนำเข้า-นำออก พร้อมกันทุกบิต (แบบขนาน)

# อธิบาย

- หลักการพื้นฐานวงจรเก็บข้อมูลแบบขนานจะต่อฟลิบฟลอบแต่ละบิตอย่างอิสระ
- ข้อมูลทางอินพุตป้อนเข้าเป็น Input 1, 2, 3, 4
- ฟลิบฟลอบจำนวนขนาด 4 บิต ต่อแยกกันทั้งทางอินพุต และเอาต์พุต โดยมี Ck ที่ต่อร่วมกัน
- ฟลิบฟลอบ 1 และ 2 , 3, 4 สามารถนำ ฟลิบฟลอบชนิด D, J-K, R-S,T มาสร้างให้เป็นแบบชนิด D ฟลิบฟลอบ
- คล็อกที่ 1 เข้ามาเป็นการนำข้อมูลจากอินพุตเข้ามาเก็บ (Store)
- คล็อกที่ 2 เข้ามาเป็นการนำข้อมูลจากเอาต์พุตฟลิบฟลอบส่งออกไปใช้งาน
- สัญญาณถูกนำออกไปใช้งาน นำออกจากเอาต์พุต 1,2,3,4

# การเลื่อนข้อมูลแบบขนาน ( parallel data transfer )

## วงจรการเลื่อนข้อมูลแบบขนาน

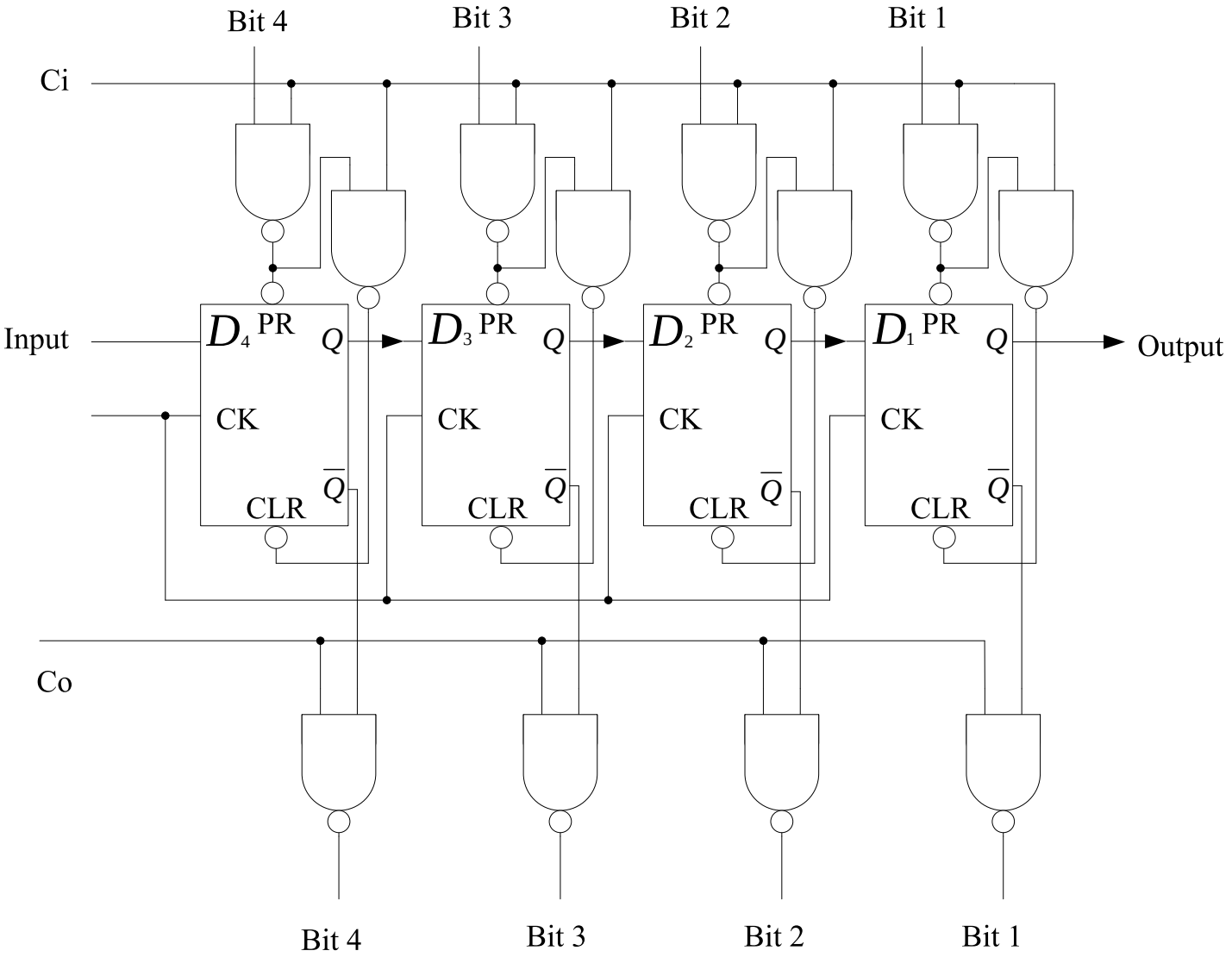


# อธิบาย

- ข้อมูลทางอินพุตที่ป้อนเข้ามา $A_{in}$ , $B_{in}$ , $C_{in}$ , $D_{in}$
- ข้อมูลที่ป้อนเข้าฟลิบฟลอบ เมื่อมีสัญญาณชิฟต์พัลซ์เข้ามา 1 ลูก
- ข้อมูลในฟลิบฟลอบนั้นจะถูกเลื่อนไปที่ขา Q ของ D ฟลิปฟลอปแต่ละตัว ซึ่งเป็นไปตามหลักการทำงานของ D ฟลิปฟลอป
- ข้อมูลที่เอาต์พุตที่ได้คือ $A_o$ , $B_o$ , $C_o$, $D_o$ จะมีข้อมูลเหมือนกับ $A_{in}$ , $B_{in}$ , $C_{in}$ , $D_{in}$ ซึ่งวงจรนี้เรียกว่าการเลื่อนข้อมูลแบบข้อมูลอินพุตขนานและเอาต์พุตขนาน (Parallel In-Parallel Out Shift Register)
- การนำข้อมูลออกไปใช้งานโดยป้อนชิฟต์พลัสที่ 2 เข้ามาก็สามารถนำค่า $A_0$,$B_0$,$C_0$,$D_0$ ออกไปใช้งาน
- การทำงานโดยเลื่อนข้อมูลทั้ง 4 บิต เข้าพร้อมกันและเลื่อนข้อมูลออกขนาด 4 บิต พร้อมกันจะทำให้เกิดความรวดเร็วในการเลื่อนนับเป็นข้อดีของตัวเก็บข้อมูลแบบนี้

# Register แบบนำเข้า-นำออกพร้อมกันทุกบิต

## Parallel In Parallel Out/Shift Right



# อธิบาย

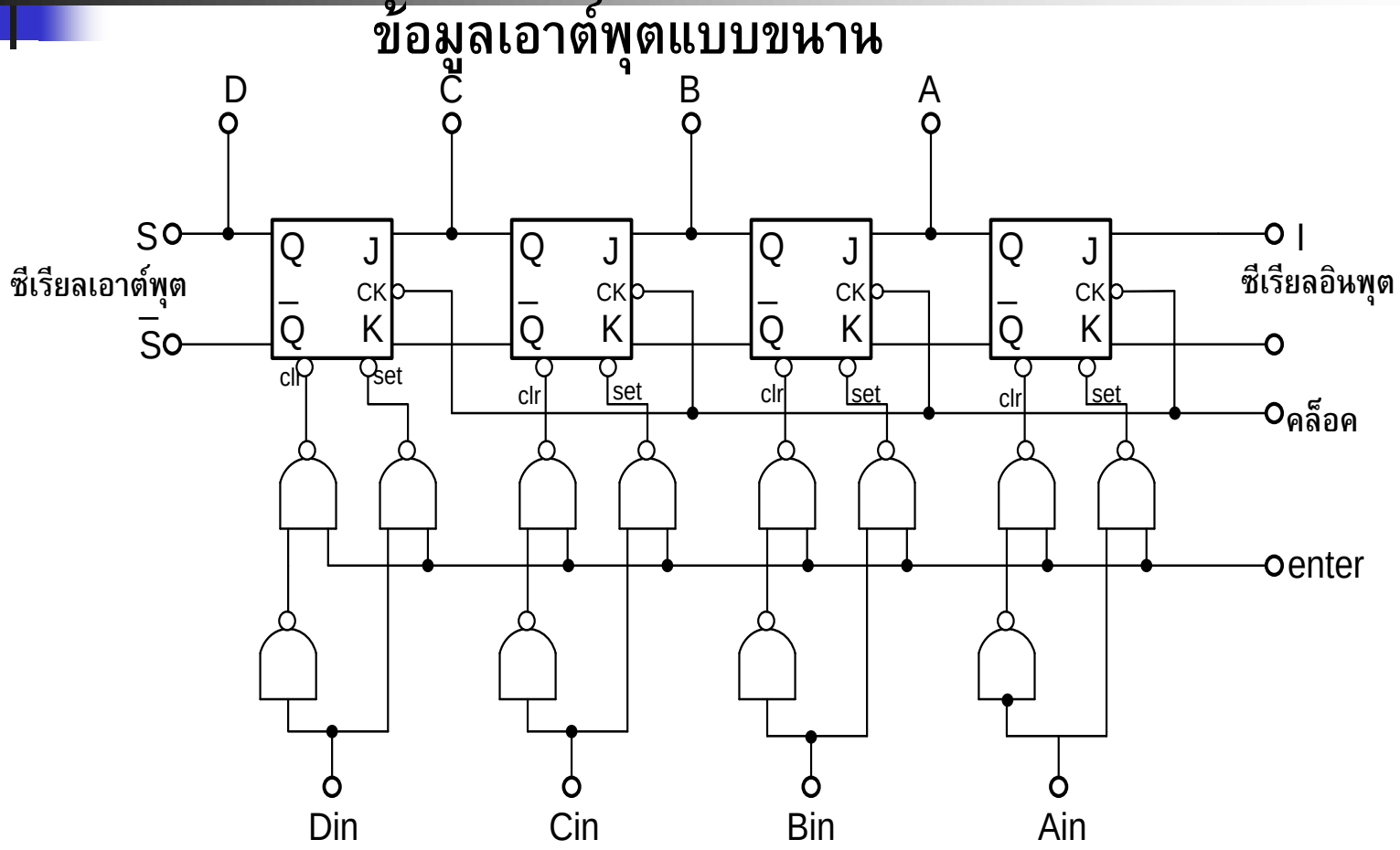
- จากวงจรสามารถส่งข้อมูลแบบขนาน หรือแบบอนุกรมก็ได้ ขึ้นกับการควบคุม
- การส่งในรูปแบบอนุกรมโดยใช้ Ck ทำงานร่วมกับอินพุต D
- ส่วนการนำข้อมูลออกก็ใช้ Ck ทำงานร่วมกับเอาต์พุต $Q_1$ มีหลักการทำงานเหมือนกับทำงานในรูปแบบ Serial In , serial Out
- การส่งข้อมูลเข้าแบบขนานพร้อมกันทุกบิต และออกพร้อมกันทุกบิต
- ข้อดีของวงจรนี้ยังสามารถป้อนข้อมูลเข้าได้ 2 แบบ คือ ขนาน/อนุกรม
- การทำงานในรูปแบบการส่งข้อมูลแบบขนานพร้อมกันทุกบิต
- การส่งข้อมูลออกเป็นแบบขนานเหมือนกับทางอินพุต
- $C_i$ ทำหน้าที่ลำเลียงข้อมูลเข้าโดยค่าระดับสัญญาณเป็น 1 ผ่านการควบคุมที่ PR ค่าระดับสัญญาณเป็น 0 ผ่านการควบคุมที่ CLR

# อธิบาย (ต่อ)

- หลักการนำข้อมูลป้อนเข้าในระบบ ได้กล่าวในเบื้องต้นของการนำข้อมูลเข้ามาแล้ว
- $C_o$ จะนำข้อมูลออกไปใช้งาน ถูกควบคุมด้วยแนนด์เกต โดยนำค่าคอมพลีเมนต์ของ $Q$ มาควบคุม
- ข้อมูลที่ได้ออกมาจะเป็นไปตามข้อมูลจริงมีค่าเท่ากับ $Q$
- การนำข้อมูลเข้าโดยใช้ $C_i$ ควบคุมที่มีค่าระดับ $1$ ส่วนการนำข้อมูลออกโดยใช้ $C_o$ ควบคุม ค่าระดับสัญญาณเป็น $1$ สำหรับการส่งข้อมูลออก

# วงจรการเลื่อนข้อมูลเข้าและออกแบบขนานไปทางซ้ายมือ

## Parallel In Parallel Out/Shift Left Register

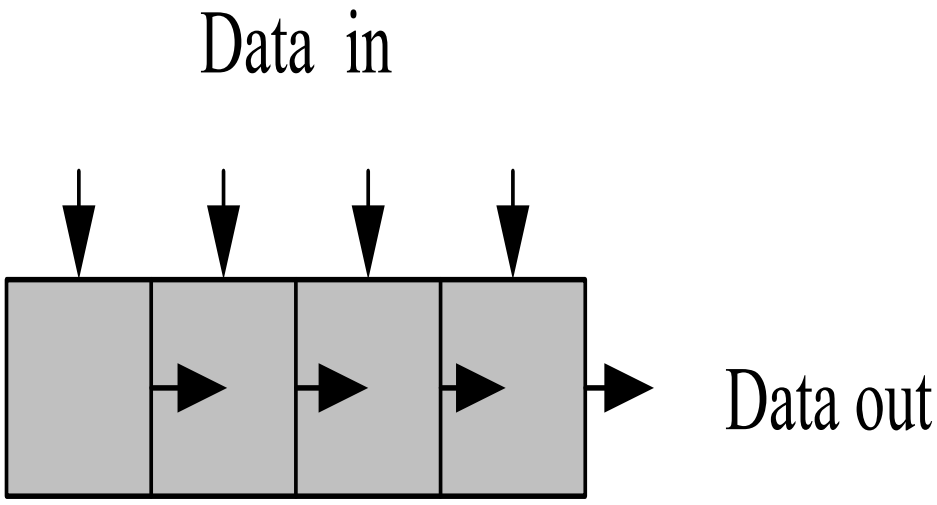


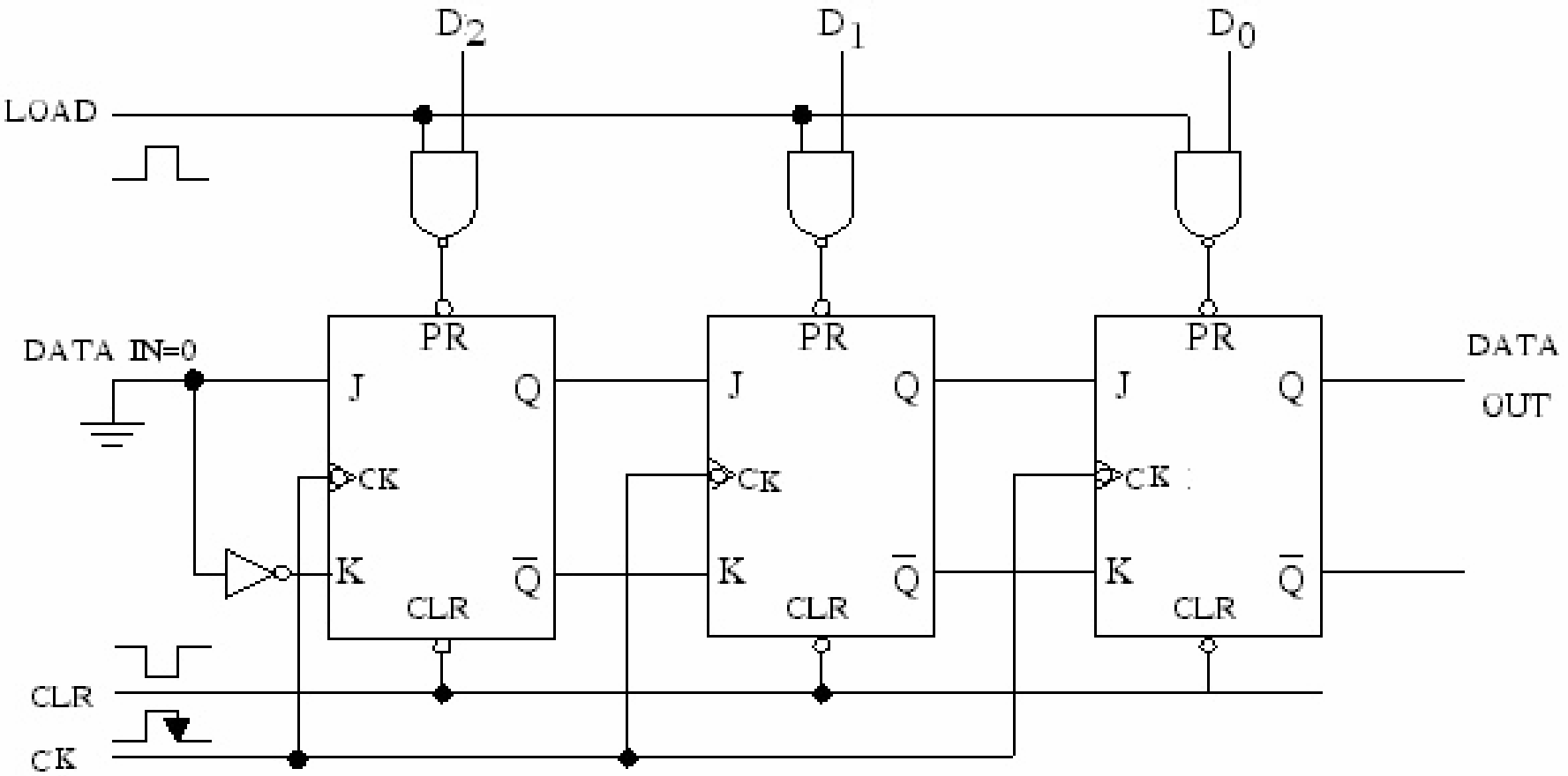
# อธิบาย

- วงจรเลื่อนข้อมูลแบบขนานนี้ สามารถใช้ในการเลื่อนข้อมูลได้ในรูปแบบขนานและอนุกรมที่กล่าวมา
- การเลื่อนเข้าแบบขนานประกอบด้วย $D_{in}$,$C_{in}$,$B_{in}$,$A_{in}$ ทำงานร่วมกับขา Enter
- การเลื่อนออกแบบขนานประกอบด้วย D,C,B,A ทำงานร่วมกับขา คล็อก
- การเลื่อนเข้าแบบอนุกรมใช้ J,K เป็นตัวป้อนข้อมูลทางอินพุต
- ถ้าต้องการเลื่อนข้อมูลเข้าเป็นค่า 1 ป้อนเข้าที่ J ถ้าต้องการ 0 ป้อนเข้าที่ K ด้วยระดับสัญญาณ 1 พร้อมกับคล็อก ตามบิตที่จะเลื่อนเช่น 4 บิตก็ใช้ 4 คล็อก
- การเลื่อนข้อมูลออกแบบอนุกรมสามารถนำออกจาก Q หรือ Complement Q พร้อมกับ คล็อก บิตที่ 5-8 ก็จะได้ข้อมูลครบทุกบิต

# 656 การเลื่อนข้อมูลเข้าแบบขนานและเลื่อนข้อมูลออกแบบอนุกรม

**Parallel In , Serial Out**



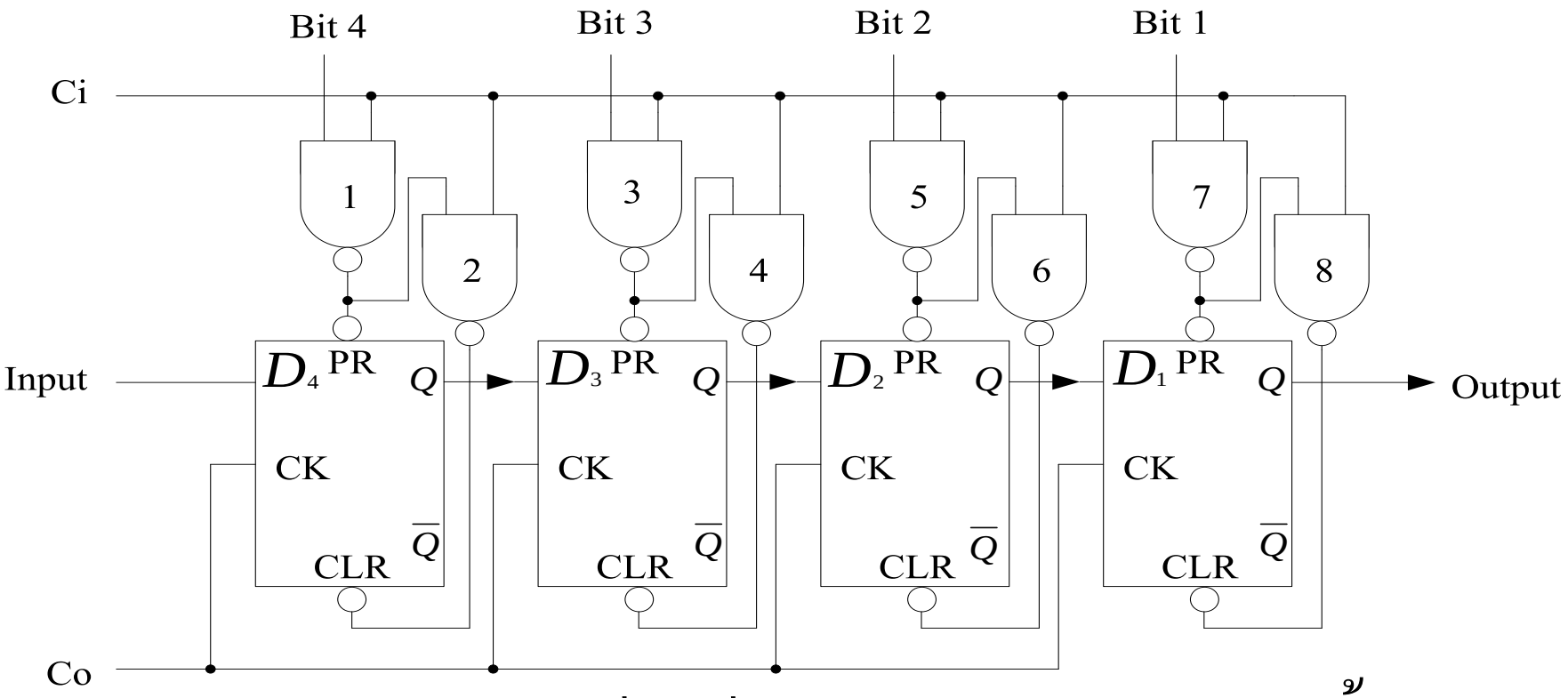

D2
D1
D0
LOAD
DATA IN=0
PR
J
Q
CK
K
Q
CLR
DATA
OUT
CLR
CK

# อธิบาย

- รูปวงจรเลื่อนข้อมูลทางเข้าเป็นแบบขนาน ทางออกเป็นแบบอนุกรม (Parallel In  Serial Out ) หรือเรียกย่อว่า PISO
- การนำเอา JK ฟลิบฟลอบมาใช้ในการเลื่อนข้อมูล ต่อในรูปแบบการเก็บข้อมูล
- จังหวะแรก สัญญาณ CLR จะทำการกำหนดค่าเริ่มต้นของฟลิบฟลอบแต่ละตัวให้มีค่า 0
- ข้อมูลที่ต้องการจะ Load เข้ามาแบบขนานจะต้องป้อนอินพุตเข้ามารอไว้ที่ $D_2$ ,$D_1$, $D_0$
- เมื่อได้รับสัญญาณ Load สัญญาณ คล็อก ที่อินพุตของ Load จะเปลี่ยนจากระดับ 0 ไปเป็นระดับ  "1“  จะเป็นการ Load ได้รับข้อมูลเข้า
- ข้อมูลที่อินพุต จะไปปรากฏที่เอาต์พุตของแนนด์เกต ในลักษณะกลับค่าจากอินพุต
- การควบคุมตัวเก็บข้อมูล JK ฟลิบฟลอบจะใช้ช่วง Negative Edge ในการควบคุมสัญญาณ Preset
- นั่นหมายความว่าสัญญาณอินพุตมีค่าเท่ากับ1 ที่จะบังคับให้ทางเอาต์พุตมีค่าเท่ากับ 1 เป็นการนำข้อมูลทางเอาต์พุตออกไปใช้งาน

# Register แบบนำเข้าพร้อมกันทุกบิต นำออกครั้งละ 1 บิต

## (Parallel In Series Out )




วงจร Register แบบนำเข้าพร้อมกันทุกบิต-นำออกครั้งละบิต

# อธิบาย

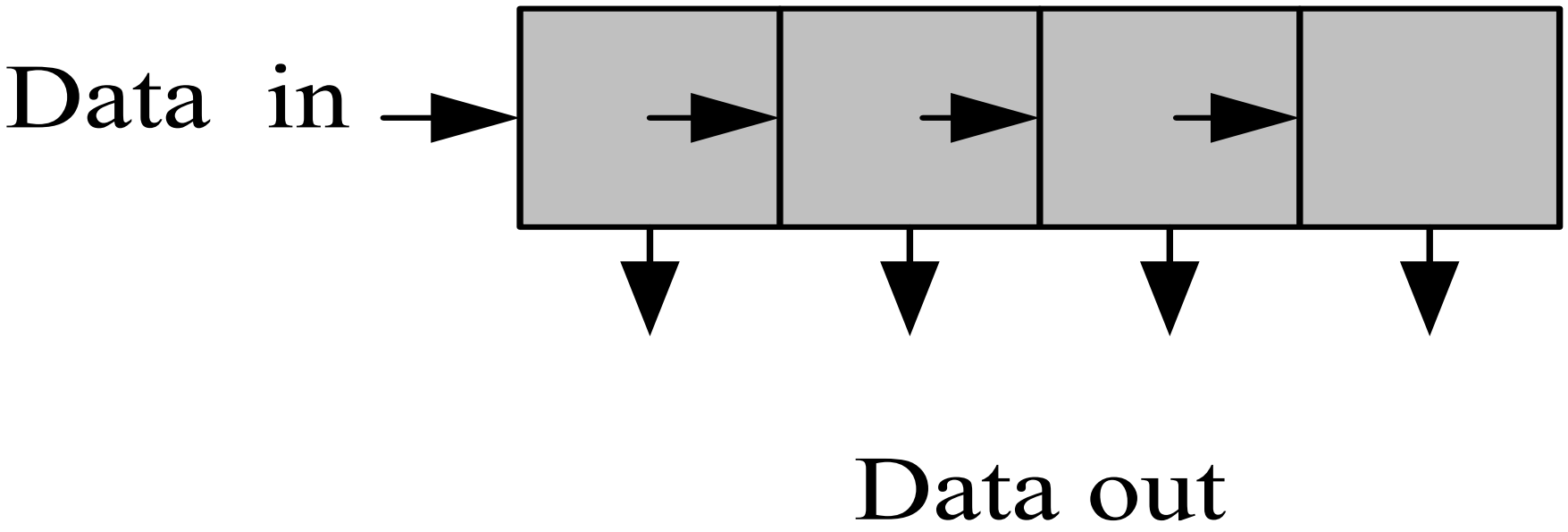
- การทำงานของวงจร Register ขนาด 4 บิต แบบนำข้อมูลเข้าพร้อมกัน
- แนนด์เกตทุกตัว 1 , 2 , 3 , 4 , 5 , 6 , 7 เป็นแนนด์เกตที่ทำหน้าที่ควบคุมอินพุตที่ป้อนเข้า Register ถูกควบคุมโดย $C_i$ ที่ Active High
- แนนด์เกต 1 , 3 , 5 , 7 จะทำงานตรงข้ามกับแนนด์เกต 2 , 4 , 6 , 8 หมายความว่า

  - Data เป็น 1 แนนด์เกต 1,3,5,7 จะทำการเซตฟลิบฟลอบให้ค่าเอาต์พุตเท่ากับ 1

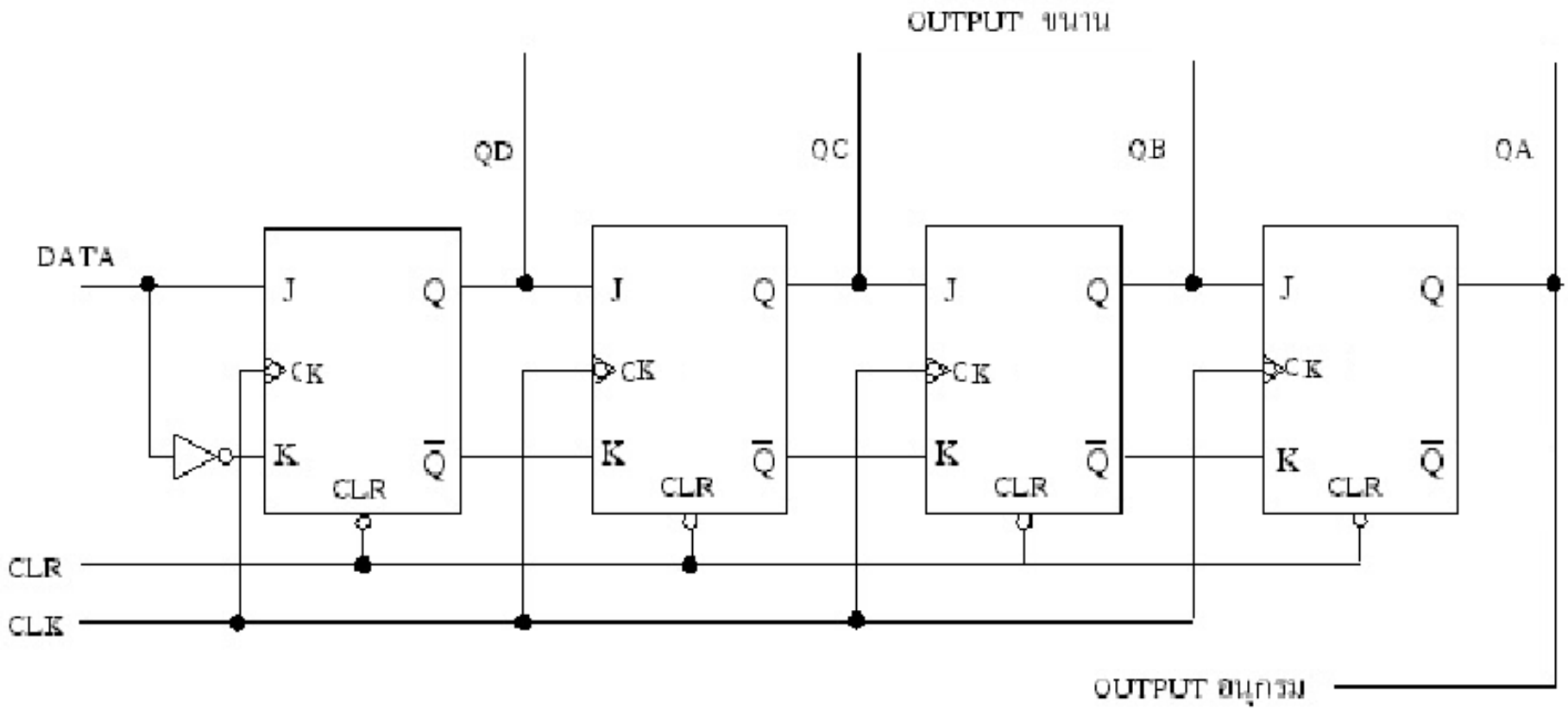
  - Data เป็น 0 แนนด์เกต 2,4,6,8 จะทำการรีเซตฟลิบฟลอบให้ค่าเอาต์พุตเท่ากับ0

# การเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน

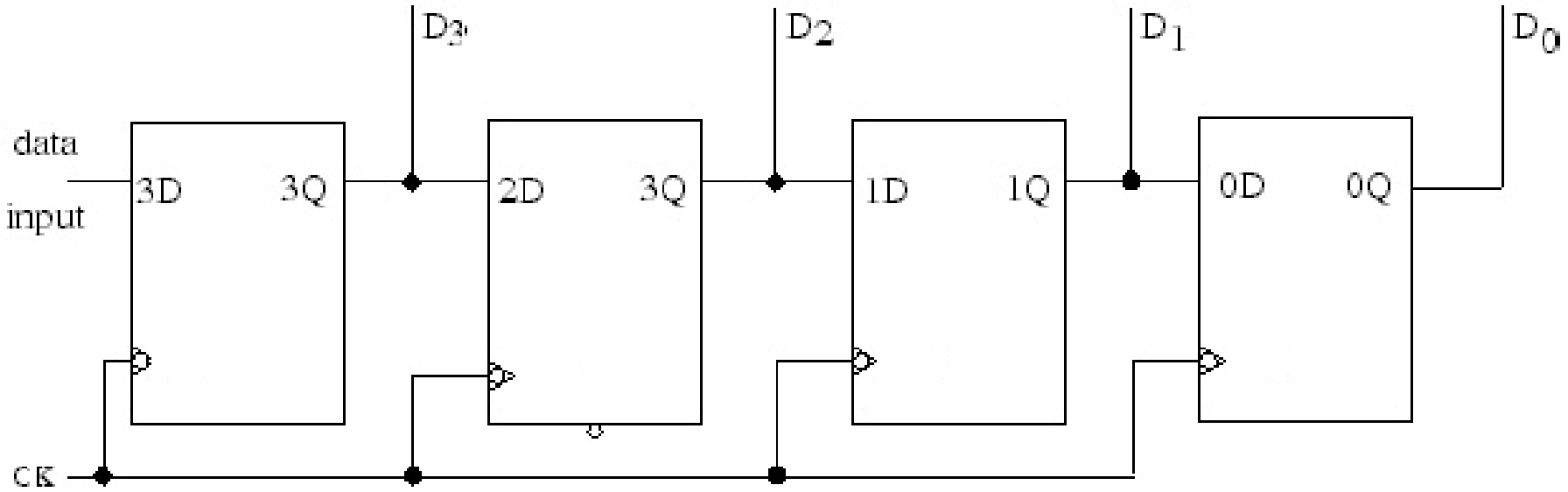
## Serial In , Parallel Out



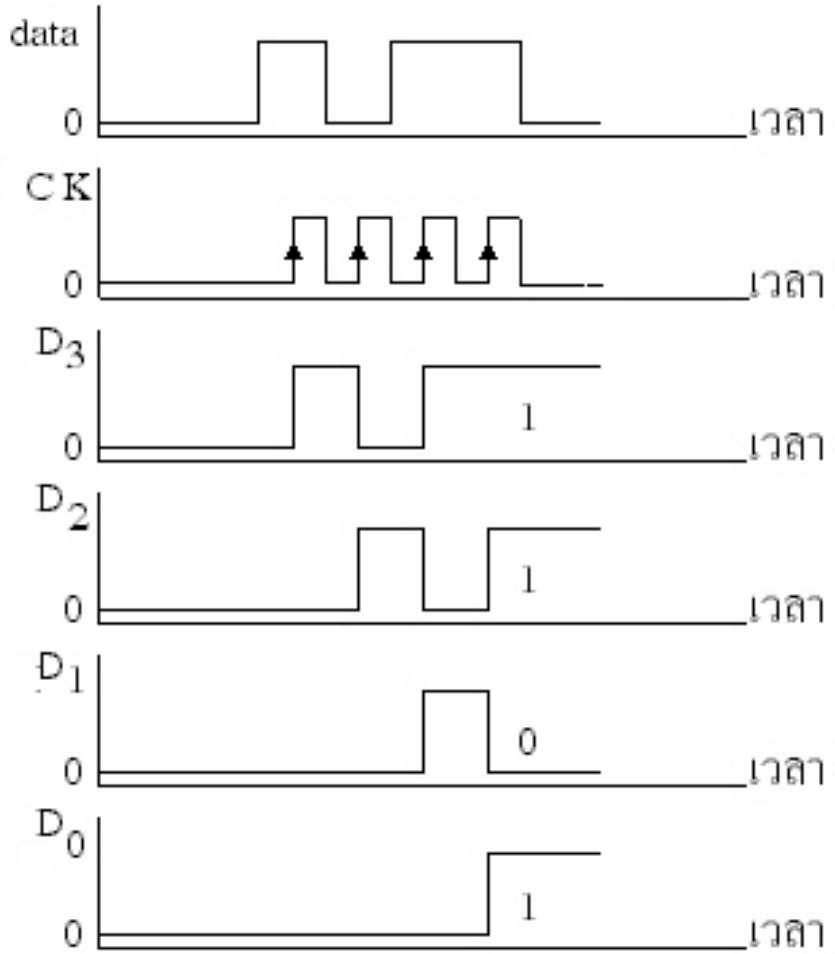
# รูปแสดงวงจรการเลื่อนข้อมูลเข้าอนุกรมออกขนาน โดยใช้ JK ฟลิปฟลอป



# ตัวอย่างการใช้ฟลิปฟลอปชนิด D ในการเลื่อนข้อมูลแบบรับเข้าอนุกรม และส่งออกแบบขนาน



# Timing diagram ของวงจรเลื่อนข้อมูล



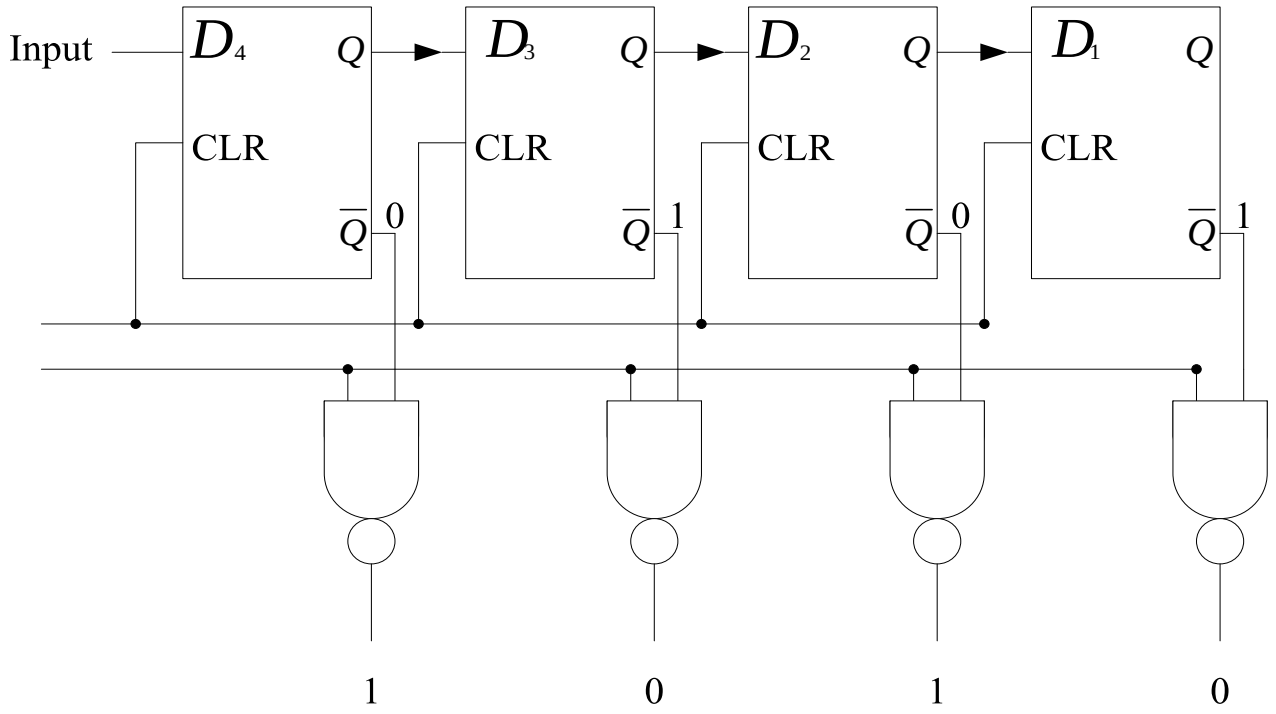
# อธิบาย

- หลังจากมีสัญญาณ คล็อกเข้ามาครบ 4 ลูก ข้อมูลที่ส่งเข้ามาในจังหวะที่สัมพันธ์กับสัญญาณ คล็อก จะปรากฏออกมาที่ขา $D_3$ , $D_2$ , $D_1$และ $D_0$
- การรับข้อมูลแบบอนุกรมในลักษณะนี้สัญญาณ คล็อก ของฝ่ายรับจะต้อง มีคาบเวลาเท่ากับสัญญาณ คล็อก ของฝ่ายหลัง (แต่ไม่จำเป็นต้องเป็น สัญญาณเดียวกัน) ข้อมูลที่รับได้จึงจะเป็นข้อมูลที่ถูกต้อง

# อธิบาย

ในการส่งข้อมูลแบบอนุกรมและตัวรับแบบอนุกรมนี้ไม่มีใช้ในการรับส่งข้อมูลทั่วไป แต่ใช้เพื่อหน่วงเวลาของข้อมูล หากต้องการ หน่วงเวลาของข้อมูลไป 1 ช่วงสัญญาณนาฬิกาก็ได้ฟลิปฟลอปชนิด D จำนวน 1 ภาค และเพิ่มขึ้นทุกจำนวน 1 ภาค ต่อการหน่วง ไป 1 จังหวะของสัญญาณนาฬิกา อย่างไรก็ตามจำนวนบิตของข้อมูลที่ส่งต่อหน่วยเวลาจะต้องเท่ากับความถี่ของสัญญาณนาฬิกาที่ให้กับฟลิปฟลอปชนิด D

# การเปลี่ยนแปลงค่าในRegister เมื่อนำค่าออก

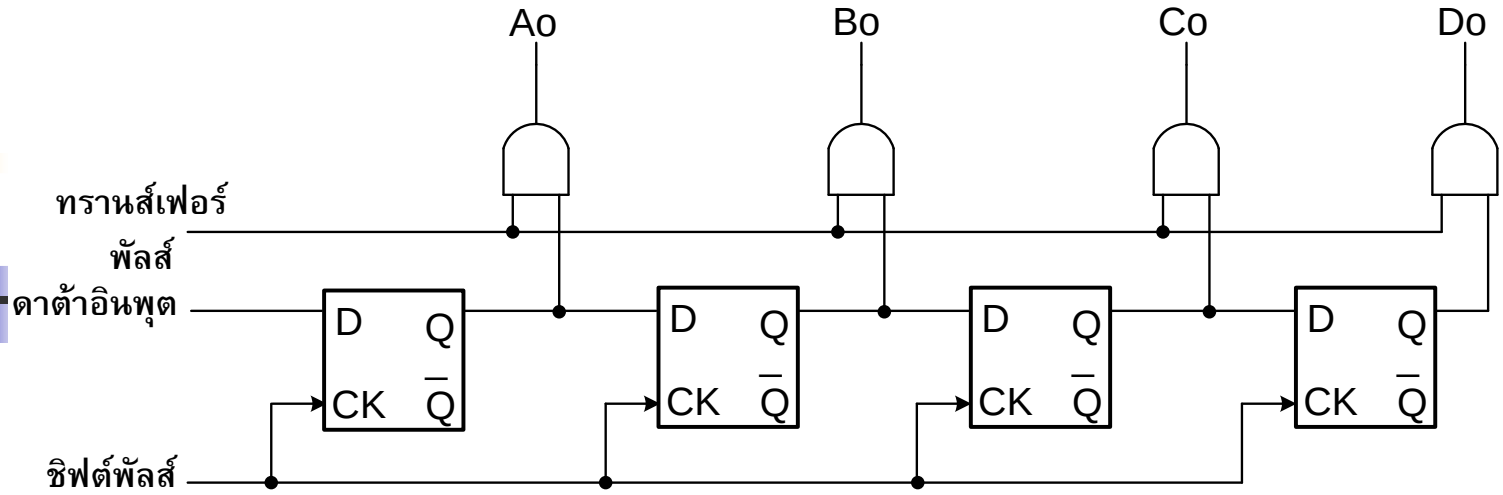



แสดงการเปลี่ยนแปลงค่าระดับสัญญาณของ Register

## 657 การเลื่อนข้อมูลเข้าแบบอนุกรมและข้อมูลออกแบบขนาน

**( serial in – parallel out shift register)**

การเลื่อนข้อมูลในลักษณะนี้จะตรงกันข้ามกับวิธีการเลื่อนข้อมูลเข้าแบบขนาน และการเลื่อนข้อมูลออกแบบอนุกรมดังที่ได้กล่าวมาแล้ว โดยอาศัยทรานส์เฟอร์พัลซ์มาช่วยดักให้มีข้อมูลออกไปแบบขนานการเลื่อนข้อมูลเข้าแบบอนุกรมและการเลื่อนข้อมูลออกแบบขนาน

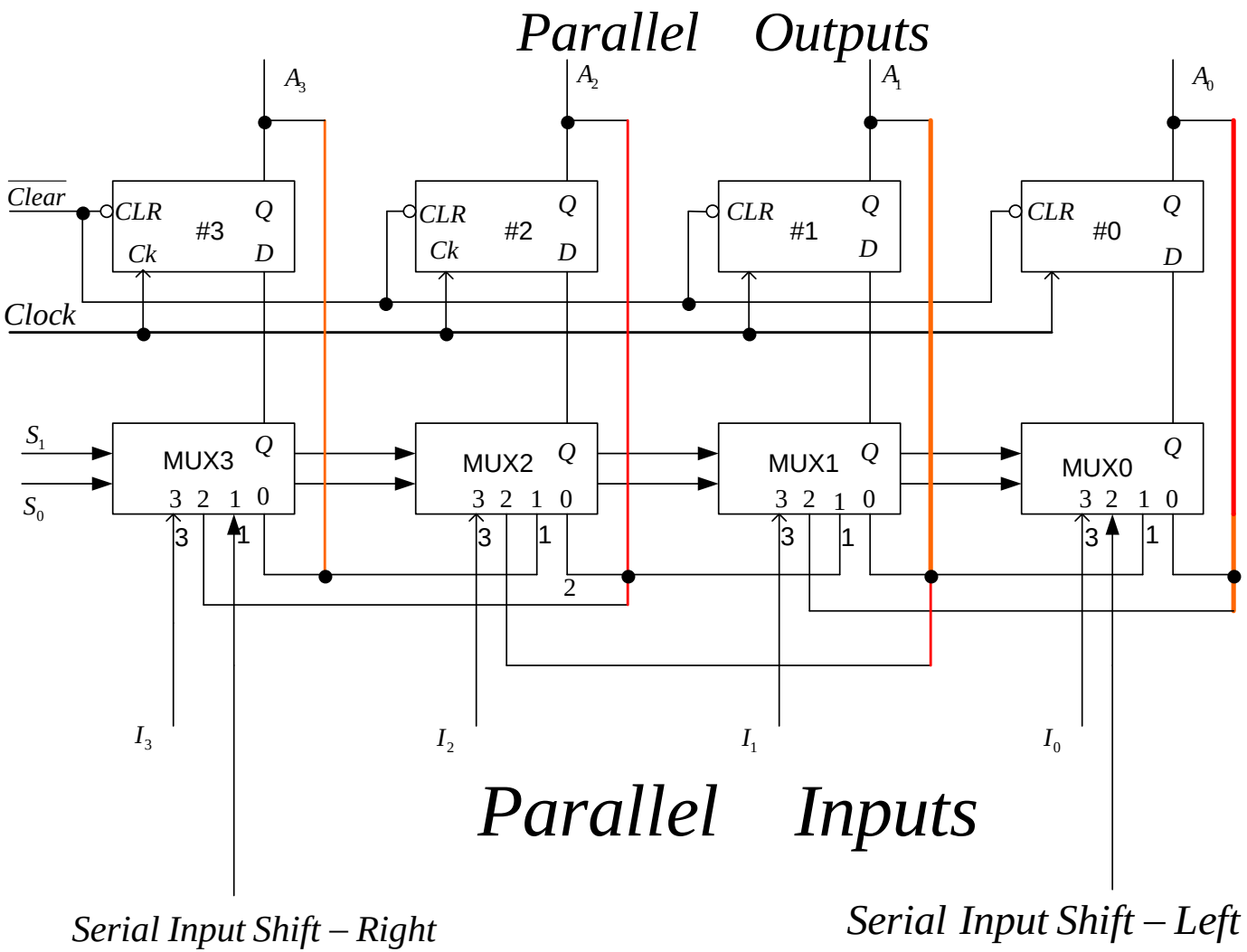



**วงจร serial in – parallel out shift register**

## การทำงาน

ของวงจรทรานส์เฟอร์พัลซ์นี้จะเป็นตัวควบคุมให้ข้อมูลถูกเลื่อนออกไปแบบขนาน โดยการป้อนลอจิก “ 1 ” เข้าที่ทรานส์เฟอร์พัลซ์และมีแอนด์เกตเป็นตัวช่วย แต่ถ้าต้องการเคลียร์ข้อมูลเอาต์พุตก็ต้องป้อนลอจิก “ 0 ” ที่ทรานส์เฟอร์พัลส์ดาต้าอินพุตเป็นตัวป้อนข้อมูลพร้อมกับเลื่อนพัลซ์ที่จะควบคุมข้อมูลให้ถูกเลื่อนออกไปใช้งานได้ตามต้องการ

# 658 วงจรเลื่อนข้อมูลแบบ Universal ขนาด 4 บิต



# อธิบาย

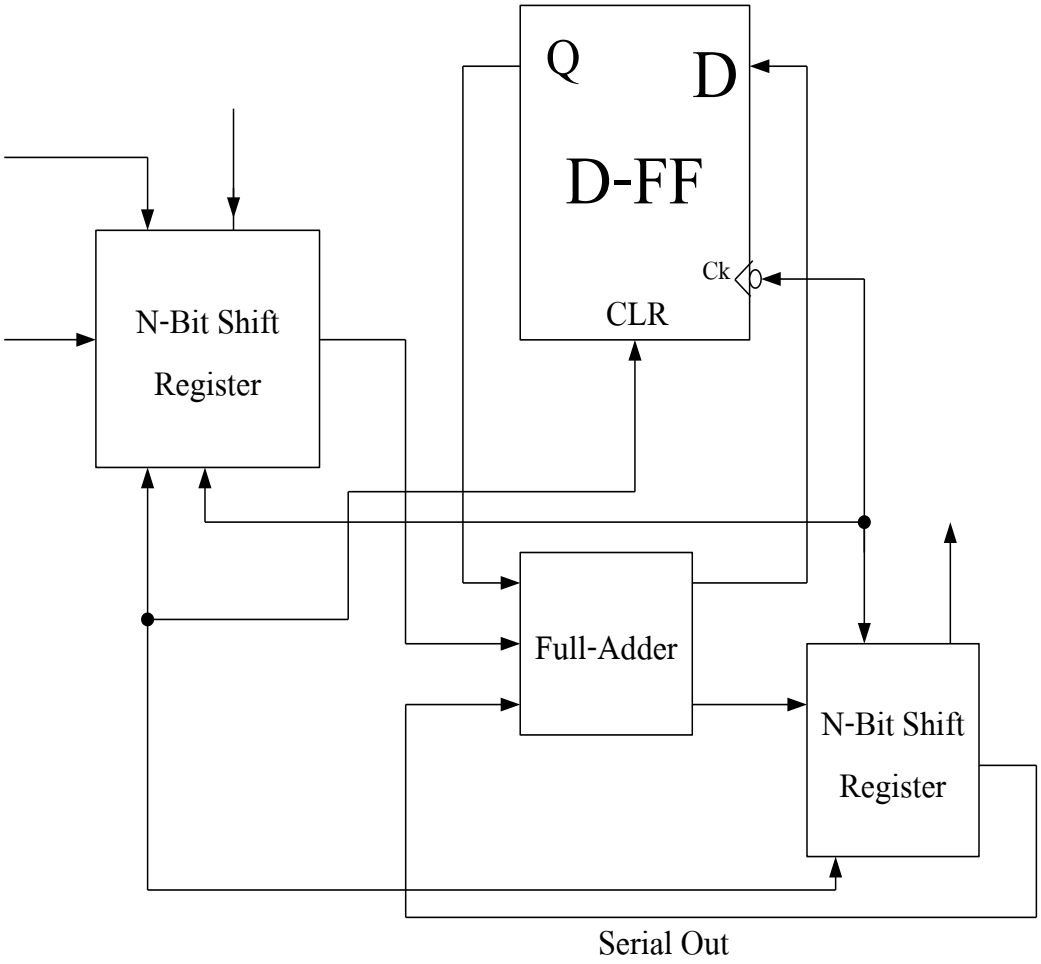
- การสร้าง Universal Shift Register ขนาด 4 บิต
- ประกอบด้วยวงจรเก็บข้อมูล และวงจรมัลติเพล็กซ์ เพื่อเลือกฟังก์ชันการทำงาน
- Universal Shift Register ขนาด 4 บิต ทางด้านอินพุตและเอาต์พุต ประกอบด้วย
  - ข้อมูลอินพุตแบบขนาน 4 บิตประกอบด้วย $I_0$-$I_3$ เอาต์พุตประกอบด้วย $A_0$ -$A_3$
  - ข้อมูลอินพุตที่ใช้ในการเลื่อนไปทางซ้ายป้อนเข้าที่อินพุต 2 ของวงจร MUX0 ข้อมูลทางเอาต์พุตออกที่ขา$A_3$ ของวงจรฟลิบฟลอบ #3
  - ข้อมูลอินพุตที่ใช้ในการเลื่อนไปทางขวามือป้อนเข้าที่อินพุต1 ของวงจร MUX3 ข้อมูลเอาต์พุตขาออกที่ขา $A_0$ ของวงจรฟลิบฟลอบ #0
- การเลือกข้อมูลนั้นสามารถควบคุมการถ่ายเทข้อมูลโดยการควบคุมที่ $S_1$,$S_0$

$S_1, S_0 = 00$ ไม่มีการถ่ายข้อมูล / $S_1, S_0 = 01$ การเลื่อนข้อมูลไปทางขวา

$S_1, S_0 = 10$ การเลื่อนข้อมูลไปทางซ้าย / $S_1, S_0 = 11$ การเลื่อนข้อมูลแบบขนาน

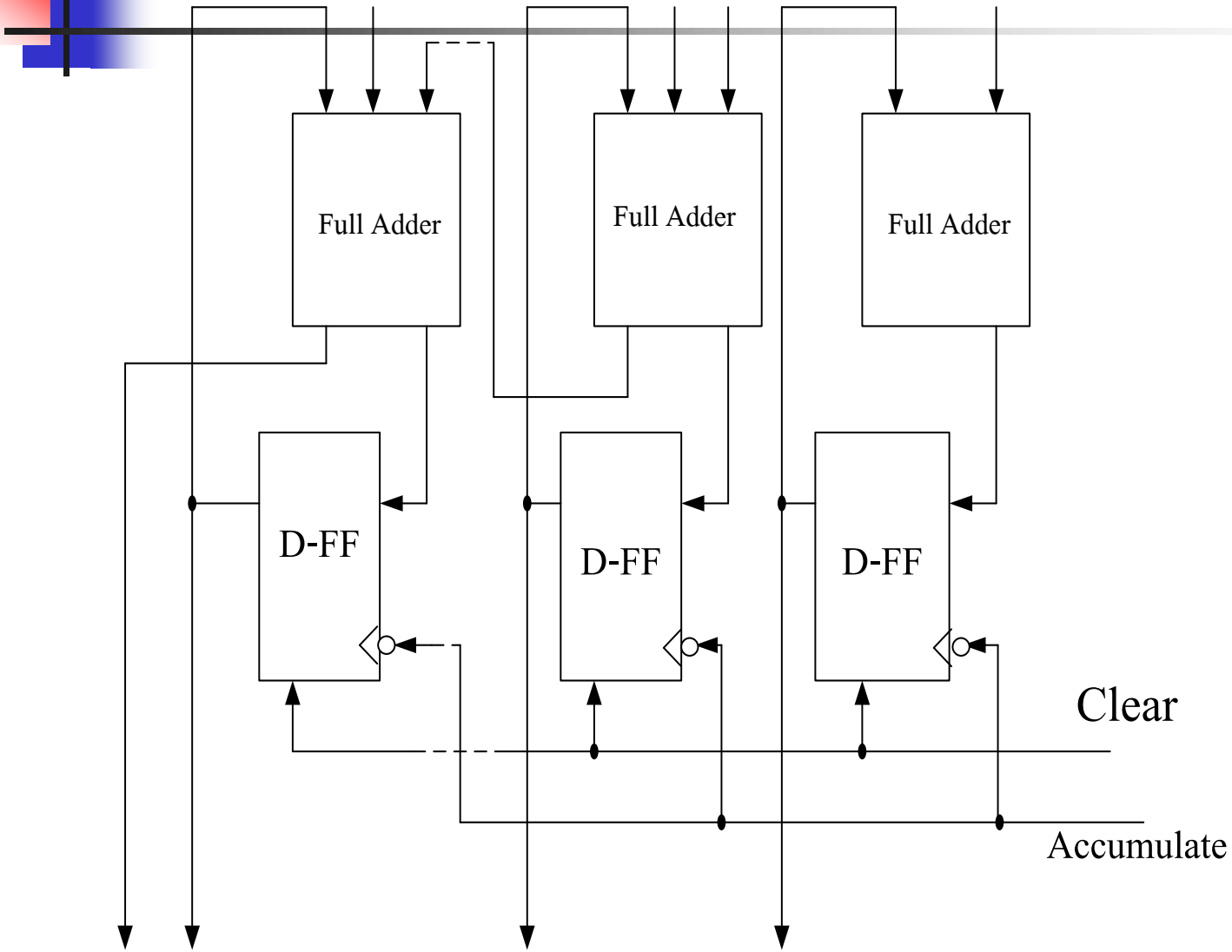
# ทดสอบความรู้สัปดาห์ที่12

1. จงอธิบาย Serial Accumulators ถึงการทำงาน และ ทิศทางการส่งถ่ายข้อมูล



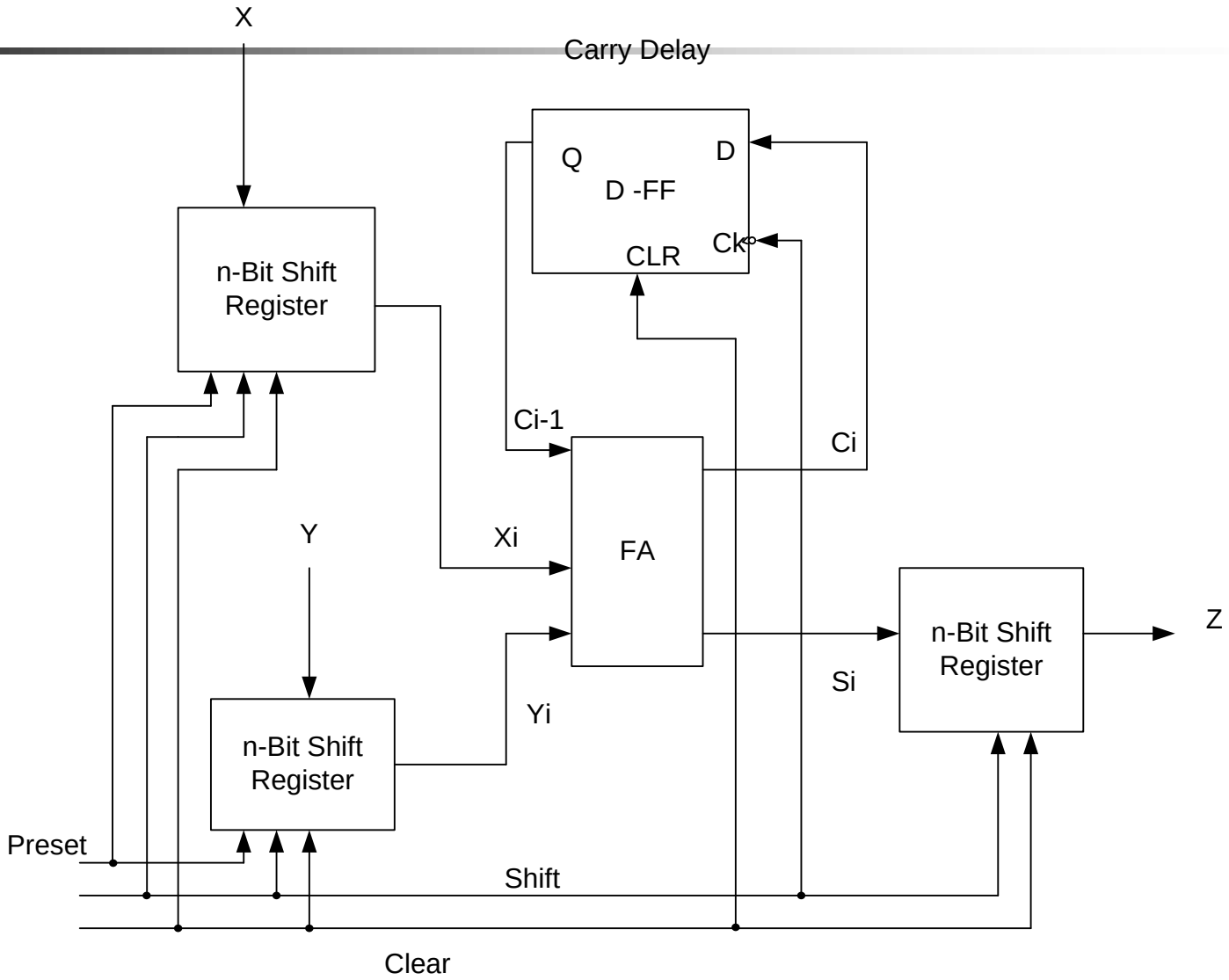
# ทดสอบความรู้(ต่อ)

2. จงอธิบาย Parallel Accumulators ถึงการทำงาน และ ทิศทางการส่งถ่ายข้อมูล

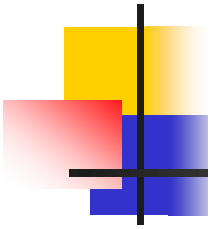


# ทดสอบความรู้(ต่อ)

3. จงอธิบาย การทำงานวงจรบวกแบบอนุกรมที่ใช้ตัวเก็บข้อมูล



# เนื้อหาคำบรรยายในสัปดาห์ที่ 13 วงจรนับเลข



**66 หลักการทั่วไปของการนับ**

**67 การนับแบบริปเปิล (Asynchronous Counter )**

671 วงจรนับขึ้นแบบริปเปิล

672 วงจรนับลงแบบริ ปเปิล

673 วงจรนับขึ้น/ลงแบบริปเปิล

**68การนับตามโหมด**

681 วงจรนับขึ้นโหมด 6

682 วงจรนับขึ้นโหมด 11

683 การนับลงโหมด 5

684 การนับลงโหมด 8

# เนื้อหาคำบรรยายในสัปดาห์ที่ 13 วงจรนับเลข (ต่อ)

**69 การนับแบบซิงโครนัส (Synchronous Counter )**

691 การออกแบบวงจรนับแบบซิงโครนัส

692 การออกแบบวงจรนับแบบรหัสเกร์ย

693 การออกแบบวงจรนับแบบซิงโครนัสโหมด 5

694 การออกแบบวงจรนับขึ้นสิบ

695 การออกแบบวงจรนับลงแบบซิงโครนัสขนาด 4 บิต

696 การออกแบบวงจรนับกระโดดแบบซิงโครนัสขนาด 4 บิต

697 การนำวงจรนับไปใช้ในการหารความถี่

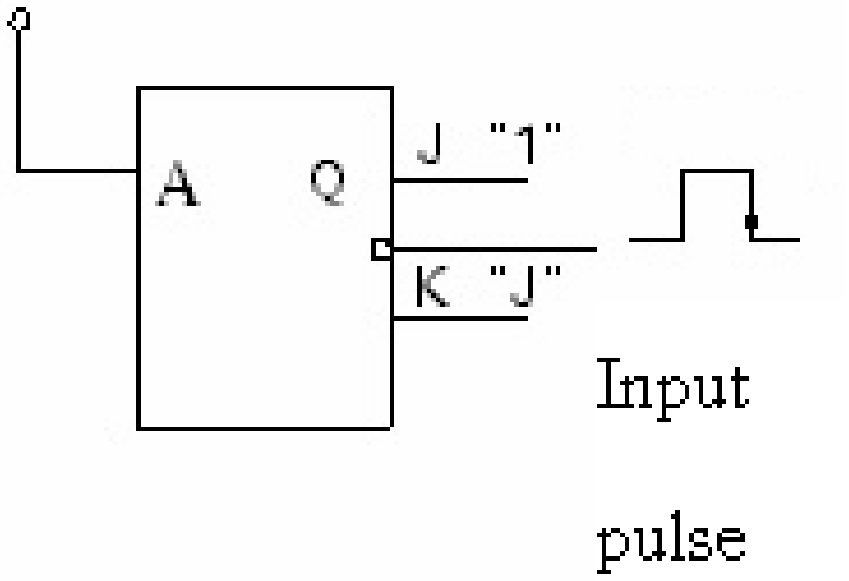
698 การนำวงจรนับไปประยุกต์ใช้งาน

# 66 หลักการทั่วไปของการนับ

วงจรนับแบบไบนารี่ (Binary Counter)

โครงสร้างของวงจรนับแบบไบนารี่จะใช้T-Flip Flop เป็นตัวนับ

เอาต์พุตของหลักต่ำจะถูกส่งไปเป็นอินพุตของภาคที่สูงกว่าดังรูปถัดไป




**วงจรนับขนาด 1 บิตใช้ T-Flip Flop ที่สร้างจาก JK- Flip Flop**

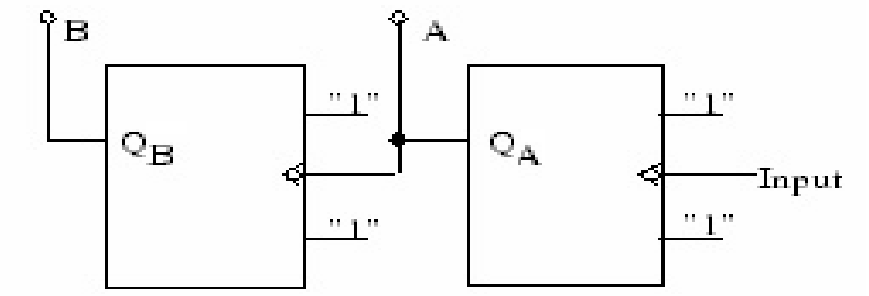
# อธิบาย

- วงจรการนับคล้ายกับวงจรรีจีสเตอร์เป็นการประยุกต์เอาฟลิบฟลอปมาใช้งาน
- วงจรนับเป็นวงจรที่เกิดจากการนำฟลิบฟลอปมาต่อรวมกันหลายตัว
- หน้าที่วงจรนับเพื่อทำหน้าที่นับพัลส์คล็อก (Clock) หรือพัลส์(Pulse)
- วงจรหารความถี่นั้นก็ได้จากเอาต์พุตของวงจรนับในโหมดต่างๆ
- วงจรนับจะประกอบด้วย ฟลิบฟลอปเป็นพื้นฐาน สามารถนับค่าได้ทุกจำนวนที่ต้องการ
- ในสัปดาห์นี้จะได้กล่าวถึงรายละเอียดของวงจรการนับในรูปแบบที่มีการนำไปใช้งานในระบบดิจิตอลต่อไป วงจรนับแบ่งได้เป็น 2 ชนิด ใหญ่ๆ ที่ครอบคลุมการนับขึ้น, นับลง , นับแบบกระโดดได้แก่

1. วงจรนับแบบไม่เข้าจังหวะ (Asynchronous Counter)
2. วงจรนับแบบเข้าจังหวะ (Synchronous Counter)

# 67 การนับแบบริปเปิล (Asynchronous Counter)

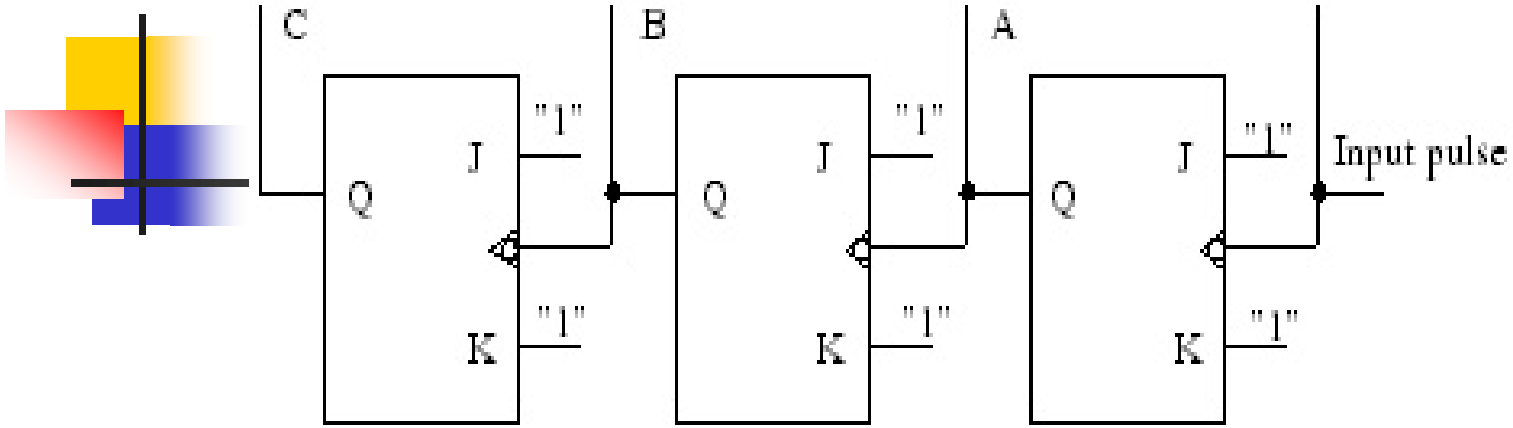
## การนับขึ้นขนาด 2บิต




**วงจรนับ 4 (โหมด 4) ค่า 0- 3**

- การนำฟลิบฟลอบมาต่อเป็นวงจรนับขนาด 2บิต
- สามารถนับค่าได้ไม่ซ้ำกัน 4 ค่า
- ค่าเอาต์พุตบิตต่ำนำไปต่อที่อินพุตบิตที่สูงกว่าดังรูป

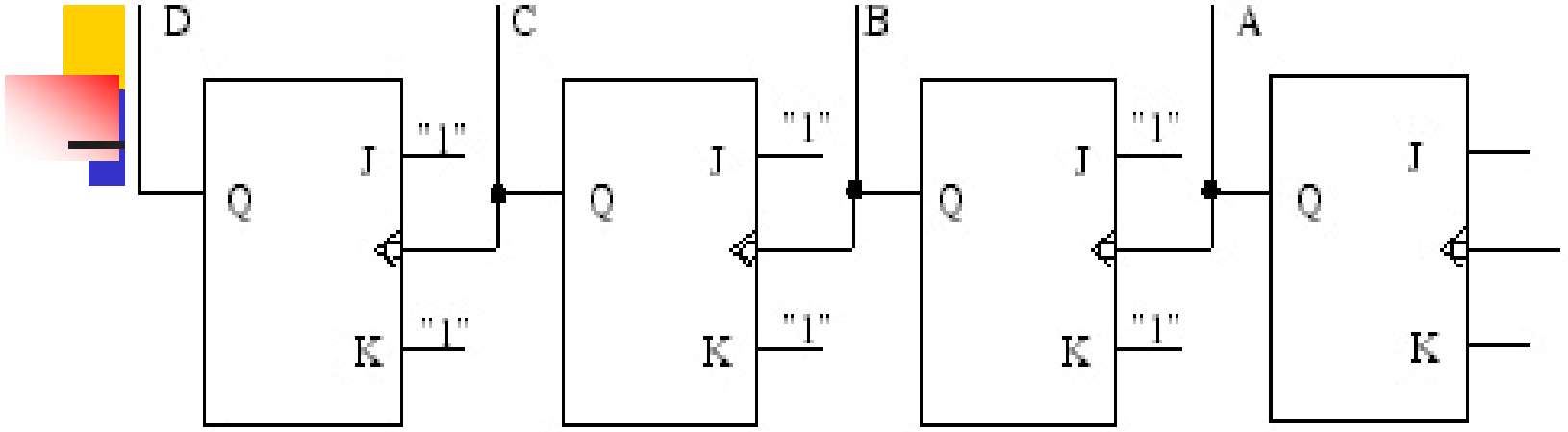
# การนับขึ้นขนาด 3 บิต




**วงจรนับ 8 (โหมด 8)**

- วิธีการนำเอาต์พุตหลัก A ส่งไปเป็นอินพุตของหลัก B
- จากนั้นนำเอาต์พุตหลัก B ส่งไปเป็นอินพุตของหลัก C
- จากหลักการดังกล่าวนี้สามารถนำไปประยุกต์ เป็นวงจรนับ 8 และ 16
- ด้วยวิธีเพิ่มค่าบิตเป็น D การเพิ่มที่บิตนี้ เป็นการเพิ่มขึ้นของการนับในวิธีการของ BCD
- ค่าจะมากขึ้นบิตละ 2 เท่า (8 *2 =16 ค่า)

# การนับขึ้นขนาด 4 บิต

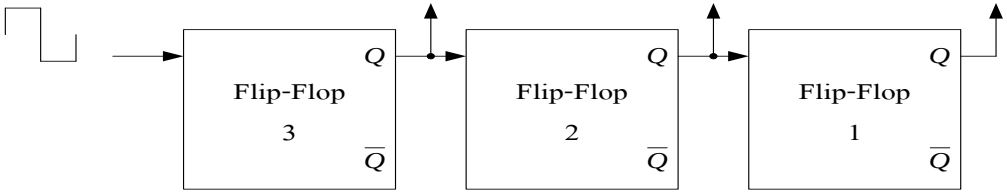



วงจรนับ 16 (โหมด 16)

- เอาต์พุตของวงจรนับขนาด 1,2,3,4 บิต จะนับค่าได้เป็น 2, 4, 8, 16
- จากรูปวงจรนี้มีขนาด 4 บิต A,B,C,D เป็นตัวนับจะทำงานในสภาวะ Active Low
- ค่าเอาต์พุตจะเปลี่ยนแปลงตามเลขไบนารี่ตามค่าประจำหลัก
- ค่าทั้งหมดจะเปลี่ยนไปตาม A,B,C และ D ตั้งแต่ 0000-1111

# ตารางการนับขึ้นขนาด 3บิตแบบริบเปิล

# (Asynchronous Counter)

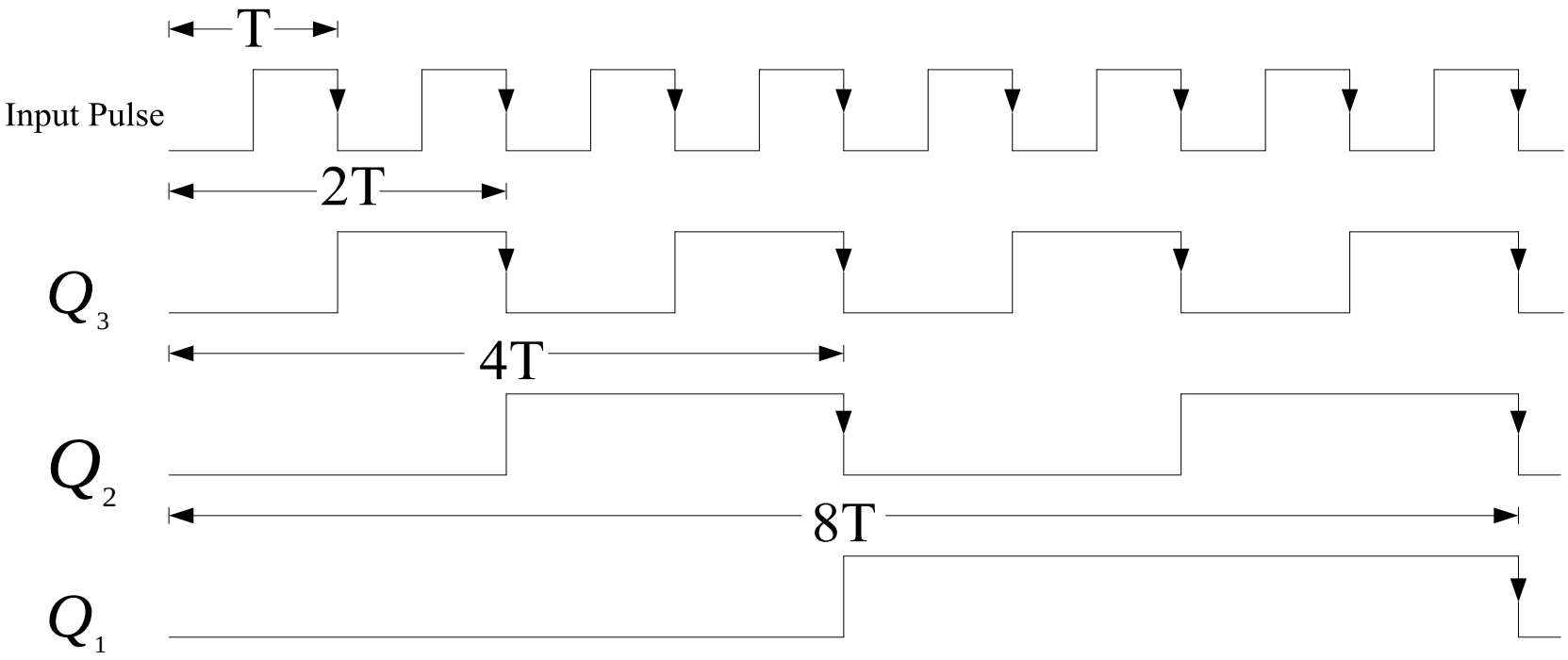



(a)

| ลำดับ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 |
| 8 | 0 | 0 | 0 |

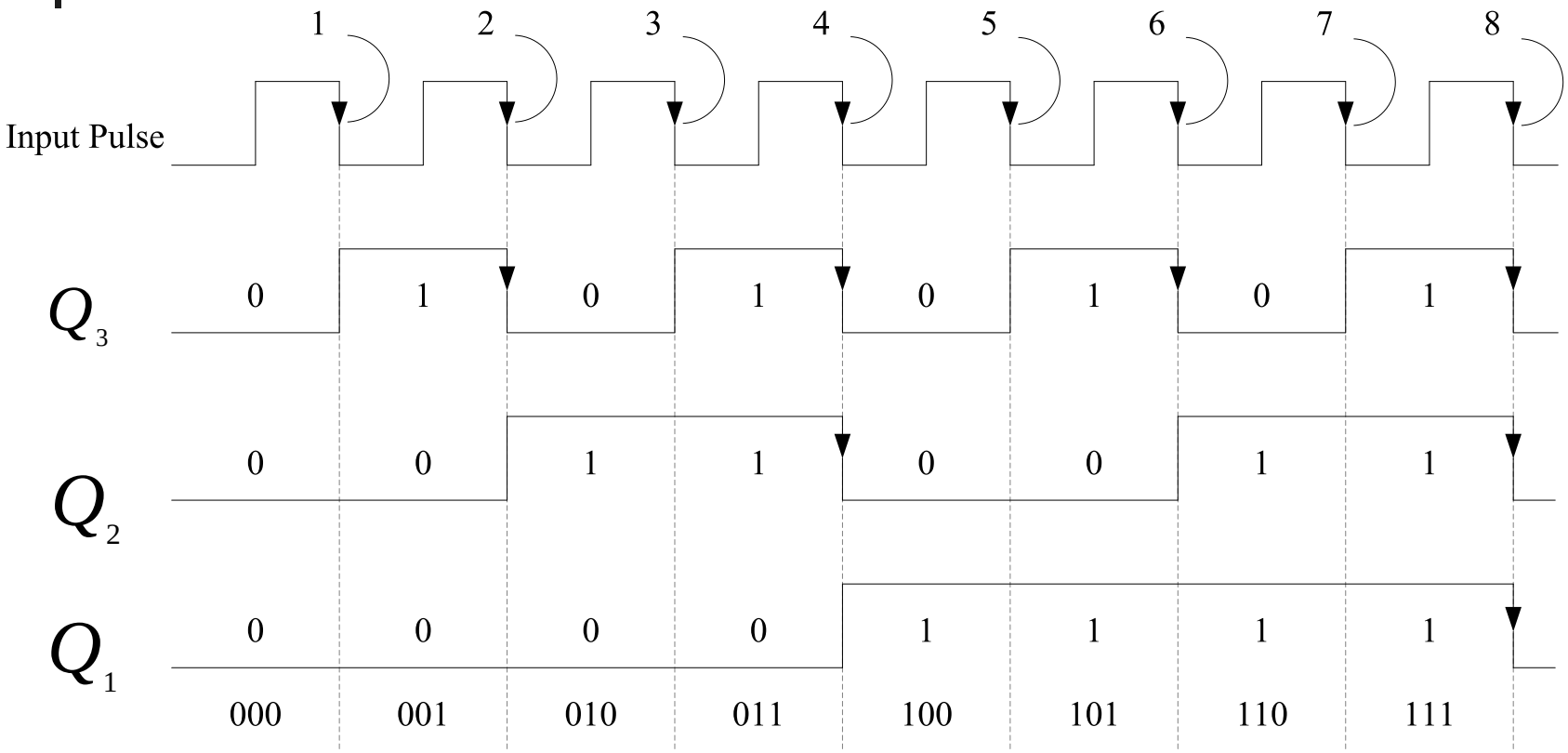
(b)

# Timing Diagram ของการนับขึ้นขนาด 3 บิต




Timing Diagram การทำงานของวงจรนับไปข้างหน้าแบบเลขฐานสอง

# Timing Diagram แสดงรหัสฐาน 2 ขนาด 3 บิต



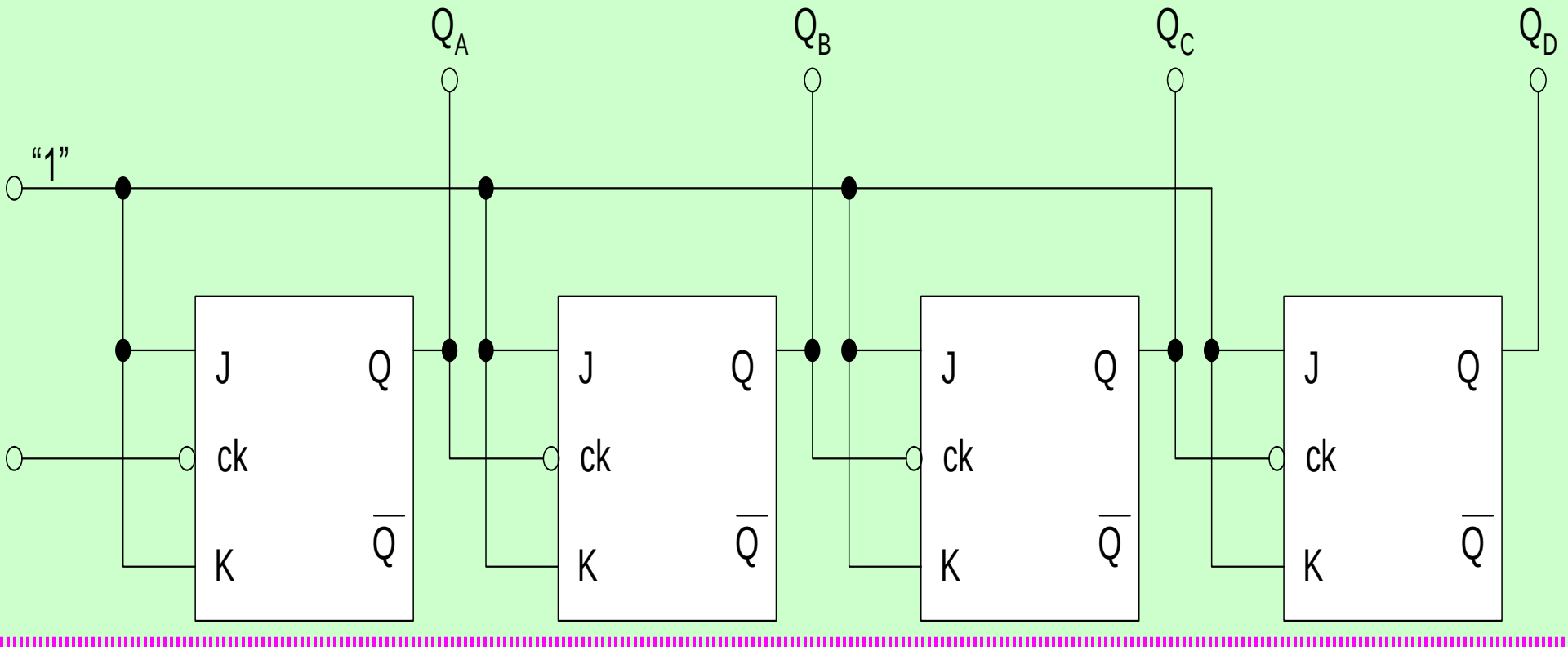

Timing Diagram การทำงานของวงจรนับไปข้างหน้าแบบเลขฐานสอง

# อธิบาย

- วงจรนับขึ้นขนาด 3 บิตแบบริบเปิล เอาต์พุต $Q_3$ ของบิต $L_{SB}$ นำไปต่อเป็นอินพุตของบิต $Q_2$ และ $Q_1$ ซึ่งเป็น $M_{SB}$ พิจารณาดังรูป (a)
- ในรูป (b) เป็นตารางการทำงาน D ฟลิบฟลอบทั้งสามบิต ตามหลักการที่กล่าวมาในสัปดาห์ที่ 11
- อินพุตที่ป้อนเป็นสัญญาณคล็อกที่มีความถี่เท่ากับ 1/T ดังที่ศึกษาผ่านมาแล้ว
- จาก Timing Diagram ของ $Q_3$ มีค่าเท่ากับ 2T หรือค่าความถี่ต่ำกว่าอินพุต 2 เท่า ในความหมายคือ ความถี่ถูกหารสอง
- เอาต์พุต$Q_2$ มีค่าเท่ากับ 4T หรือค่าความต่ำกว่าอินพุต 4 เท่า ความถี่ถูกหารสี่
- เอาต์พุต$Q_1$ มีค่าเท่ากับ 8T หรือค่าความต่ำกว่าอินพุต 8 เท่า ความถี่ถูกหารแปด
- พิจารณาการนับของTiming Diagramค่าที่เอาต์พุตจะถูกเปลี่ยนค่าจาก 000 โดยค่าจะเพิ่มครั้งละ1 เป็น 001 ,010 จนกระทั่งถึง 111 เป็นค่าสูงสุดที่วงจรจะนับได้เป็นคล็อกที่แปดของอินพุต –
- -ข้อสังเกตุ ถ้านำคอมพลีเมนต์ไปใช้งานจะเกิดอะไรขึ้น

# 671 วงจรนับขึ้นแบบริบเปิล (Asynchronous Up Counters)

## วงจรนับขึ้นแบบไบนารี่ – ริปเปิล 4 บิต โดยใช้ J - K ฟลิบฟลอป



# อธิบาย

- การนับแบบริปเปิล(Asynchronous Counter) แบบนับขึ้นขนาด4 บิตใช้ J-K ฟลิบฟลอบ
- อินพุต Ck ของ $FF_A$ เท่านั้น ที่ได้รับสัญญาณคล็อกจะเปลี่ยนแปลงในทุกครั้งที่มีสัญญาณคล็อกเปลี่ยนแปลง
- เอาต์พุตของ $FF_A$ จะท็อกเกิล (เปลี่ยนสภาวะเป็นตรงกันข้าม) ทุกครั้งที่สัญญาณคล็อกเปลี่ยน จาก High เป็น Low
- เอาต์พุตปกติของ $FF_A$ เป็นอินพุต Ck ของ$FF_B$และคลื่นเอาต์พุต$FF_B$ จะท็อกเกิลทุกครั้งที่เอาต์พุตของ $FF_A$ เปลี่ยนจาก 1 เป็น 0
- ในทำนองเดียวกันรูปคลื่นเอาต์พุตของ $FF_C$ จะท็อกเกิลเมื่อเอาต์พุตของ $FF_B$ เปลี่ยน จาก 1 เป็น 0

-เอาต์พุต $FF_D$ จะท็อกเกิล เมื่อเอาต์พุต $FF_C$ เปลี่ยนจาก 1 เป็น 0

- จำนวนนับรหัสเลขฐานสองสูงสุดที่เป็นไปได้คือ 2n จะนับได้สุงสูด $2^4$ คือ 16 ค่า ตั้งแต่ค่า (0000-1111)

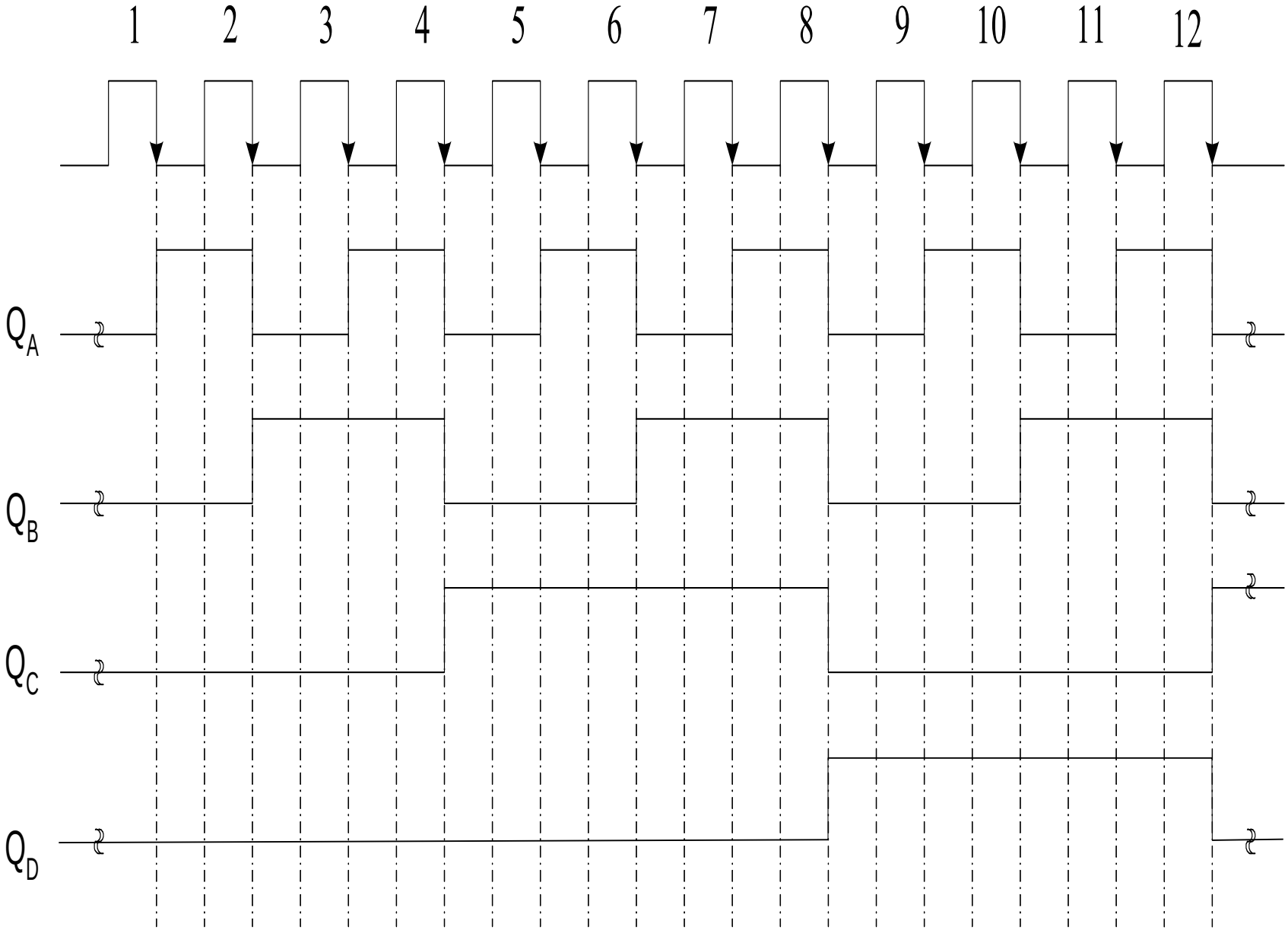
# ตารางสถานะเอาต์พุตของวงจรนับขึ้นแบบไบนารี่ 4 บิต

| จำนวนพัลส์ | $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ | เลขนับฐานสิบ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 2 |
| 3 | 0 | 0 | 1 | 1 | 3 |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| 14 | 1 | 1 | 1 | 0 | 14 |
| 15 | 1 | 1 | 1 | 1 | 15 |
| 16 | 0 | 0 | 0 | 0 | 16 |

# อธิบาย

- ตารางแสดงลำดับฟลิบฟลอบ D,C,B และ A แสดงจำนวนเลขฐานสอง โดย D เป็น $M_{SB}$ และ A เป็น $L_{SB}$ จะได้ลำดับการนับเลขฐานสองจาก 0000 ถึง 1111
- หลังจากสัญญาณคล็อกที่ 15 ปรากฏขึ้นแล้วฟลิบฟลอบของวงจรนับจะแสดงสภาวะ 1111 นั่นคือ วงจรนับทำการนับครบรอบหรือครบไซเคิล (Cycle)
- เมื่อสัญญาณคล็อกที่ 16 ปรากฏวงจรนับจะนับย้อนกลับ (Recycles) เป็น 0000 จากนั้นจึงเริ่มต้นการนับใหม่
- การทำงานของวงจรนับชนิดนี้ มีลักษณะเป็นระลอกหรือริปเปิล (Ripple) กล่าวคือรูปคลื่นเอาต์พุตของฟลิบฟลอบแต่ละตัวต่อเข้ากับอินพุตCkของฟลิบฟลอบตัวถัดไป วงจรนับประเภทนี้มักเรียกว่าวงจรนับแบบริปเปิล หรือเรียกว่าวงจรนับแบบอะซิงโครนัส

# แผนภาพไทม์มิ่ง



# 672 วงจรนับลงแบบริปเปิล

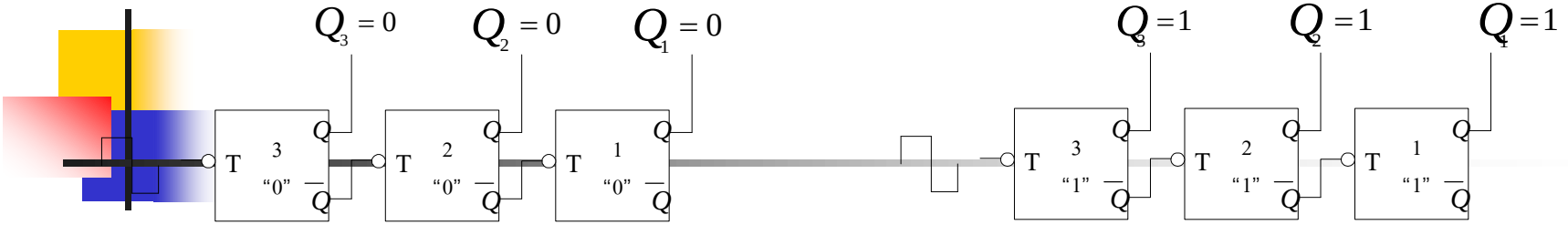
# (Asynchronous Down Counter)

วงจรนับลงนี้สามารถสร้างได้ 2 วิธีคือ

-วิธีที่1นำค่าที่ได้จากคอมพลีเมนต์ของเอาต์พุตฟลิบฟลอปมาใช้งานโดยตรง ซึ่งจะมีค่าตรงข้ามการนับขึ้น

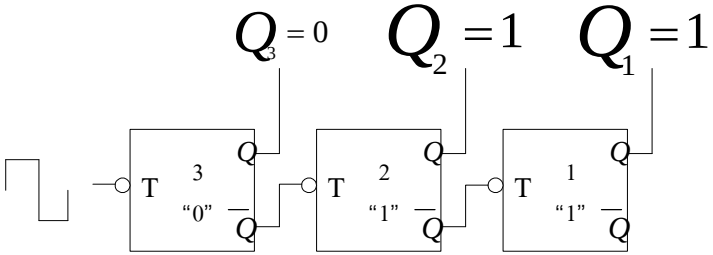
-วิธีที่2 นำค่าเอาต์พุตไปใช้งานแต่นำค่าคอมพลีเมนต์ไปต่อเป็นอินพุตของฟลิบฟลอบตัวถัดไป

# วงจรการนับลงแบบริปเปิล

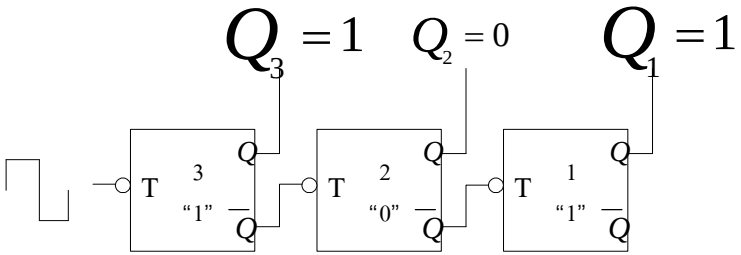



(A)

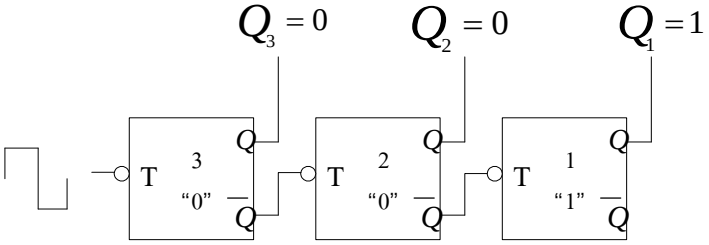
(B) Pulse Set 1




(C) Pulse Set 2

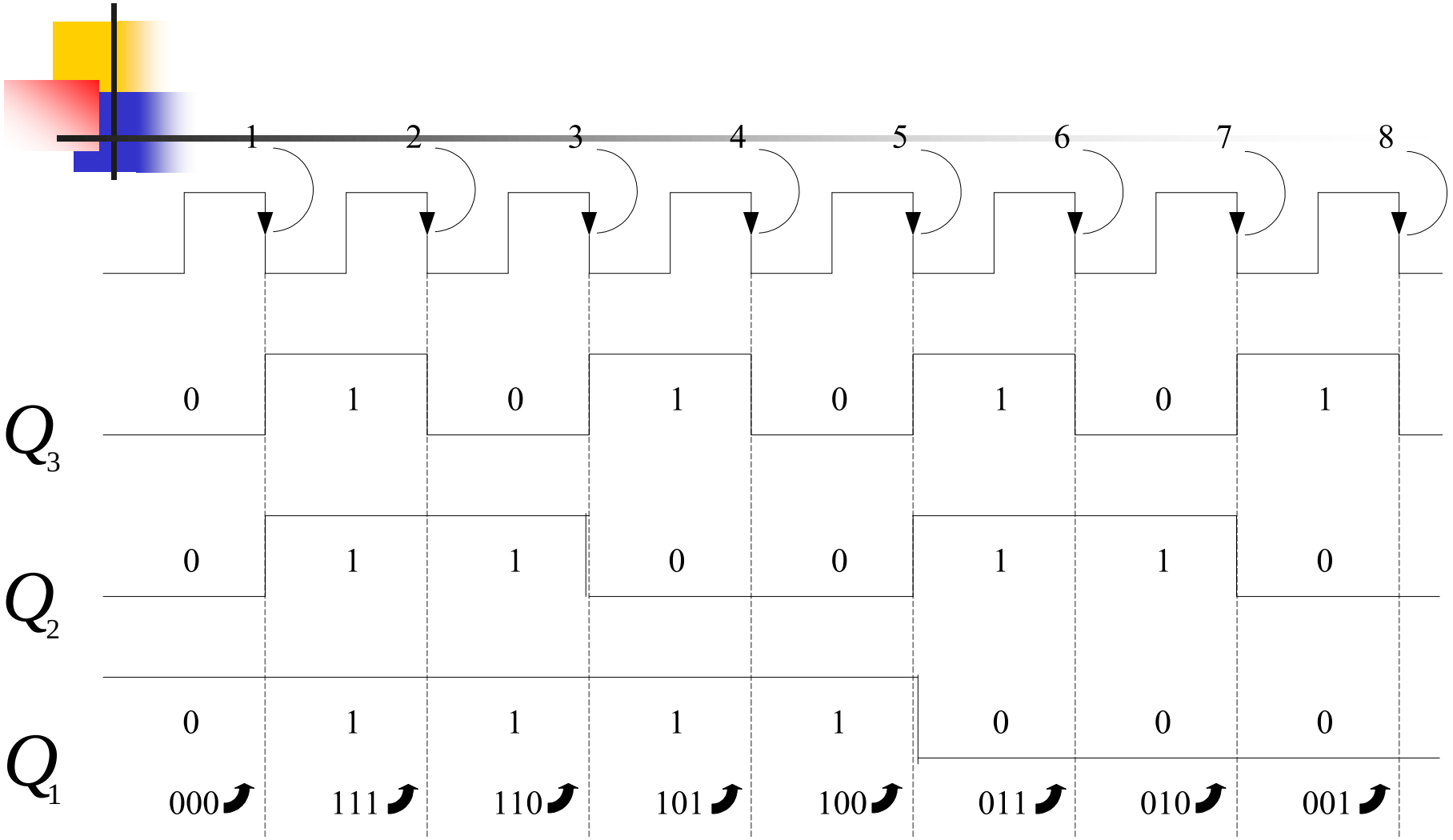



(D) Pulse Set 3




(E) Pulse Set 4

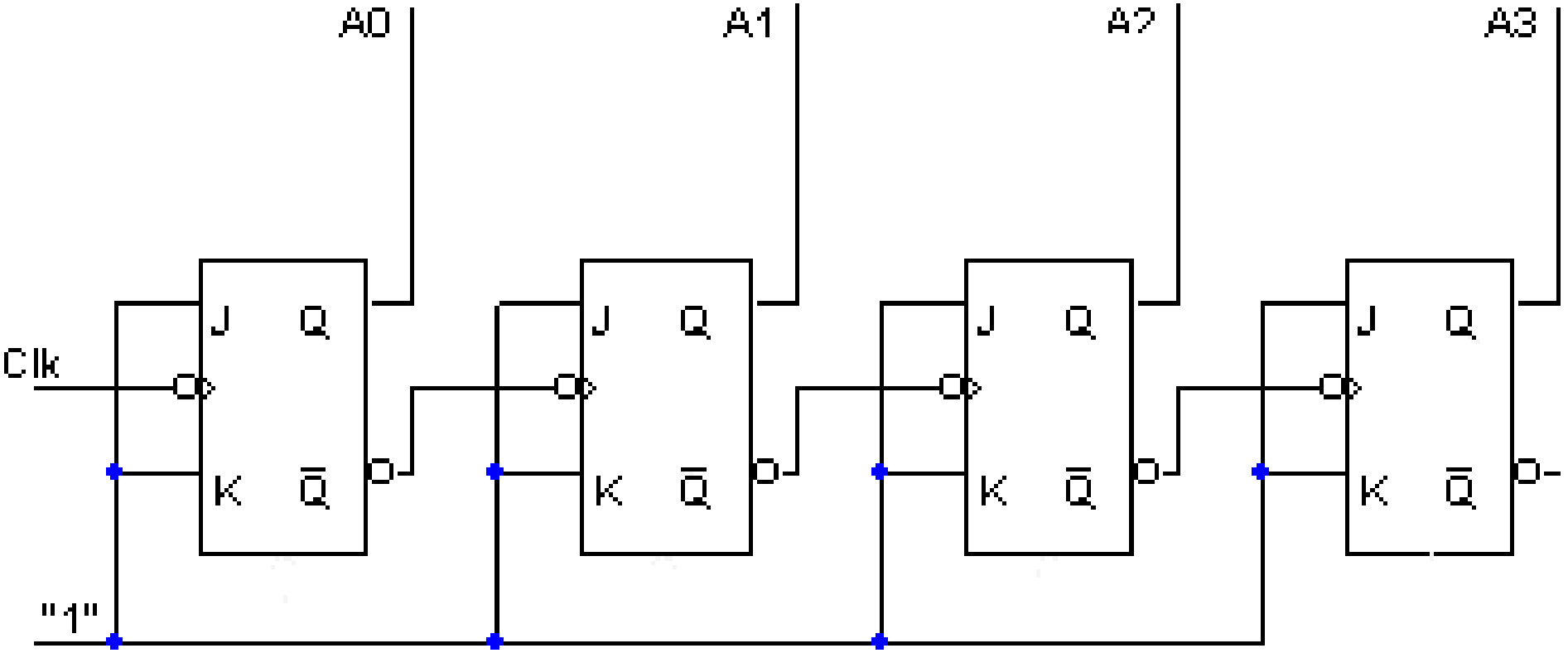
# Timing ของการนับลงขนาด 3บิต



# อธิบาย

- วงจรนับลงแบบริปเปิลขนาด3 บิต ประกอบด้วย T-FlipFlop 1,2,3
- การนำค่าเอาต์พุตไปใช้งาน $Q_1$,$Q_2$,$Q_3$
- ส่วนการต่อระหว่างบิตใช้ที่เป็นส่วนคอมพลีเมนต์ของ $Q_3$ ต่อไปยังอินพุต $Q_2$ คอมพลีเมนต์ของ $Q_2$ ต่อไปยังอินพุต $Q_1$ดังรูป (A) ยังไม่มีการเปลี่ยนแปลงของคล็อก
- การทำงานของฟลิบฟลอบจะทำงานที่ Active Low
- คล็อกแรกเข้ามาจะเป็น $Q_1$,$Q_2$,$Q_3$ เปลี่ยนสภาวะเป็น 111 ดังรูป (B)
- คล็อกสองเข้ามาจะเป็น $Q_1$,$Q_2$,$Q_3$ เปลี่ยนสภาวะเป็น 110 ดังรูป (C)
- คล็อกสามเข้ามาจะเป็น $Q_1$,$Q_2$,$Q_3$ เปลี่ยนสภาวะเป็น 101 ดังรูป (D)
- คล็อกสี่เข้ามาจะเป็น $Q_1$,$Q_2$,$Q_3$ เปลี่ยนสภาวะเป็น 100 ดังรูป (E)
- การเปลี่ยนแปลงของคล็อก 5,6,7,8 จะทำให้เอาต์พุตมีการเปลี่ยนแปลงในการลดค่า1 ไปจะครบรอบ พิจารณาจาก Timing Diagram
- วงจรนี้จึงเป็นการนับค่าลง จาก 111 จนกระทั่งเป็น 000

# วงจรนับลงแบบริปเปิลขนาด 4 บิต ใช้ J-K ฟลิบฟลอบ

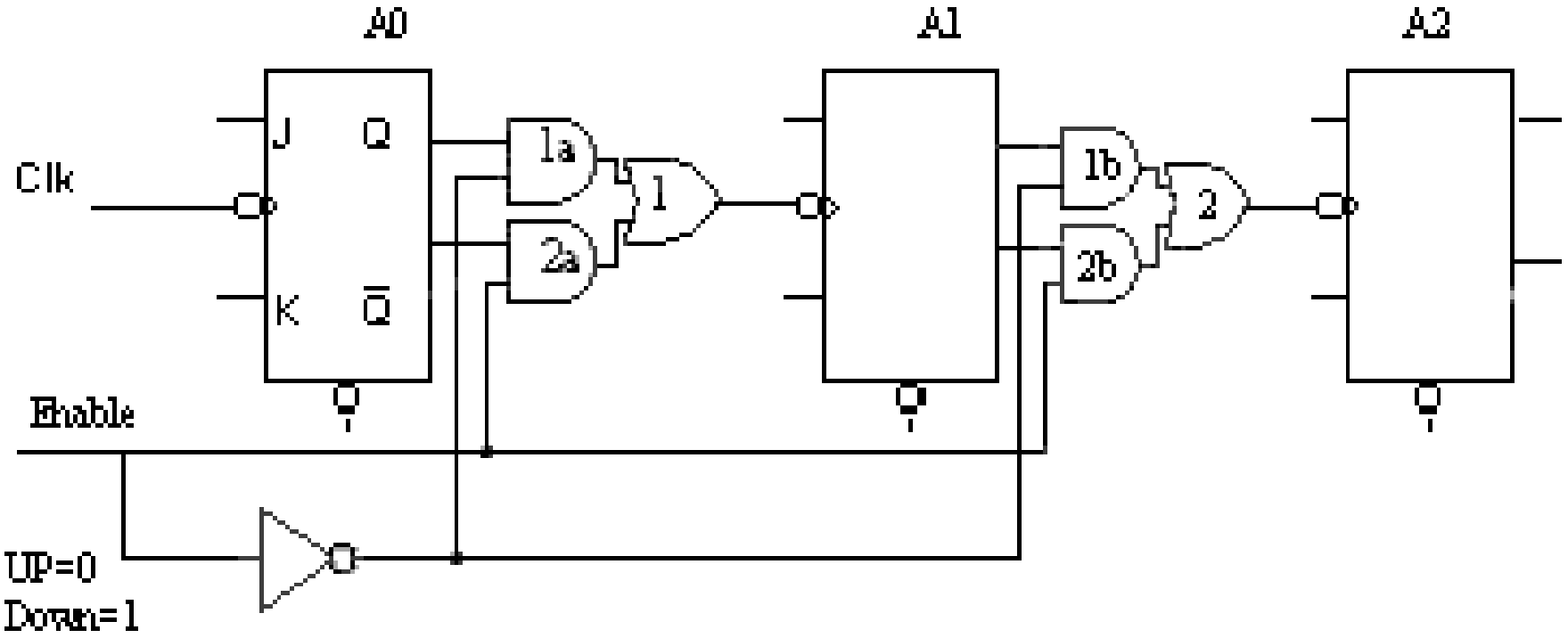


# อธิบาย

- การทำงานของวงจรนับลงขนาด 4 บิต โดยใช้ J-K ฟลิบฟลอบ
- หลักการนำ J-K มาต่อเข้ากับระดับสัญญาณ 1 เพื่อให้ทำงานในรูปแบบ Toggle เมื่อมีสัญญาณป้อนเข้าทางอินพุต
- สัญญาณที่นำไปป้อนให้กับฟลิบฟลอบตัวถัดไป นำค่าคอมพลีเมนต์ของ Q เป็นอินพุตแบบริปเปิล นับต่อกันที่ละบิต
- ค่าที่นำไปป้อนทางอินพุตของบิตถัดไปจะเป็นค่าตรงข้ามกับการนับขึ้น ดังนั้นในรูปแบบนี้จึงเป็นการนับลงค่าที่ทำการนับจะเป็นไปได้ตั้งแต่ (1111-0000)
- การนำฟลิบฟลอบมาใช้ในการนับนั้น สามารถใช้ในรูปแบบ Toggle, Delay ก็ได้ แต่ส่วนใหญ่ที่นำ J-K ฟลิบฟลอบมาใช้เพื่อนำขา J และ K มาควบคุมการนับในโหมดที่ต้องการ

# 673 วงจรนับขึ้น/นับลงแบบริปเปิล

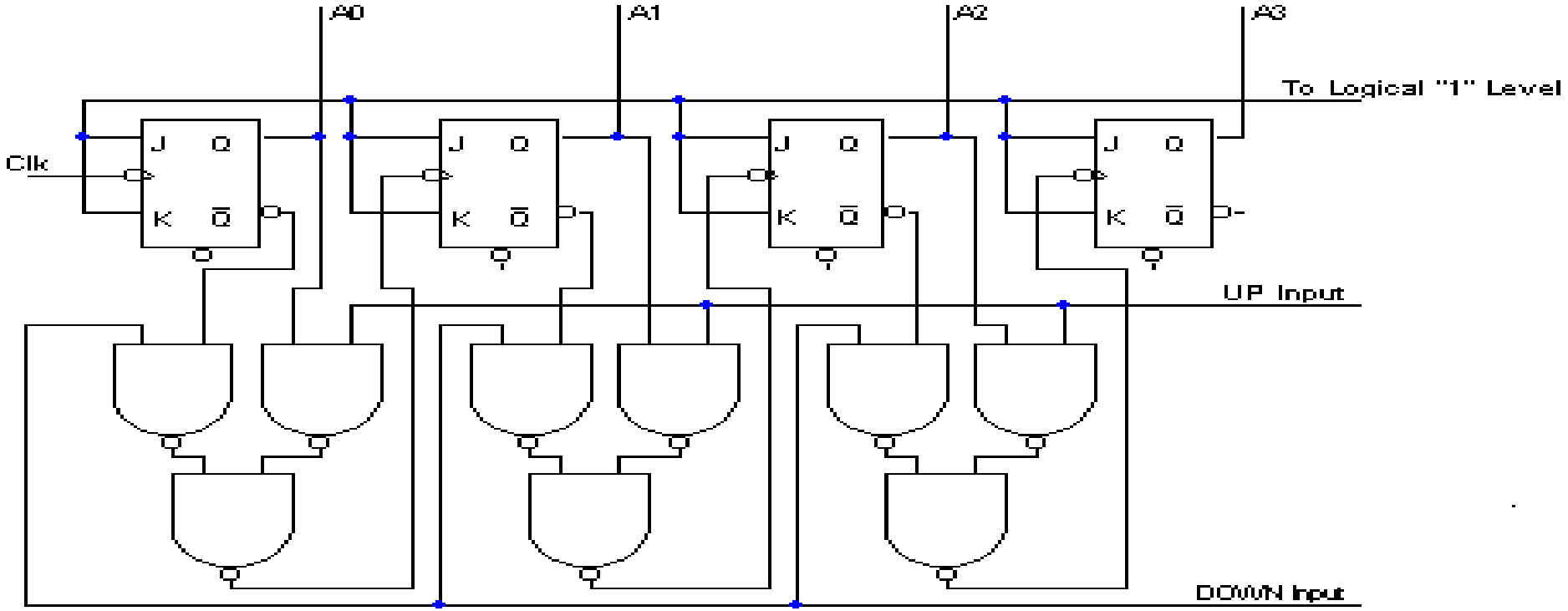
## (Asynchronous Count Up/Down)



# อธิบาย

- การออกแบบให้วงจรสามารถนับขึ้นและนับลงโดยใช้การควบคุมเอาต์พุต
- การจะเลือกให้วงจรนับขึ้นหรือนับลง จะทำได้โดยการสับเปลี่ยนสวิทช์ หรือการDisible และ Enable วงจรไปมาระหว่าง Q และ ให้ส่งสัญญาณ Clock ไปกระตุ้นฟลิบฟลอปตัวถัดไป
- ถ้าต้องการนับขึ้น จะต้องนำสาย Enable ต่อ ลงกราวด์ หรือให้มีค่าลอจิกเป็น "0" จะทำให้ขาอินพุตของ AND เกต1a และ1bที่ต่อมาจาก NOT เกต มีค่าลอจิกเป็น "1" ส่วนขาอินพุตของAND เกต2a และ 2b ที่ต่อมาจากสาย Enable โดยตรงจะมีค่าลอจิกเป็น "0" จะเห็นว่า AND เกต 1a และ 1b อยู่ในสถานะ Enable พร้อมที่จะให้สัญญาณ (Clock) จาก Q ผ่านได้ทุกขณะ
- ถ้าต้องการนับลงก็ทำกลับกันโดยสาย Enable ต่อเข้า +5 Vหรือให้ค่าลอจิกเป็น "1“
- ค่าเอาต์พุตที่ได้ออกจากQ จำนวนการนับขึ้นอยู่กับฟลิปฟลอป

# การออกแบบวงจรนับขึ้น/นับลงแบบริปเปิล ขนาด 4 บิต



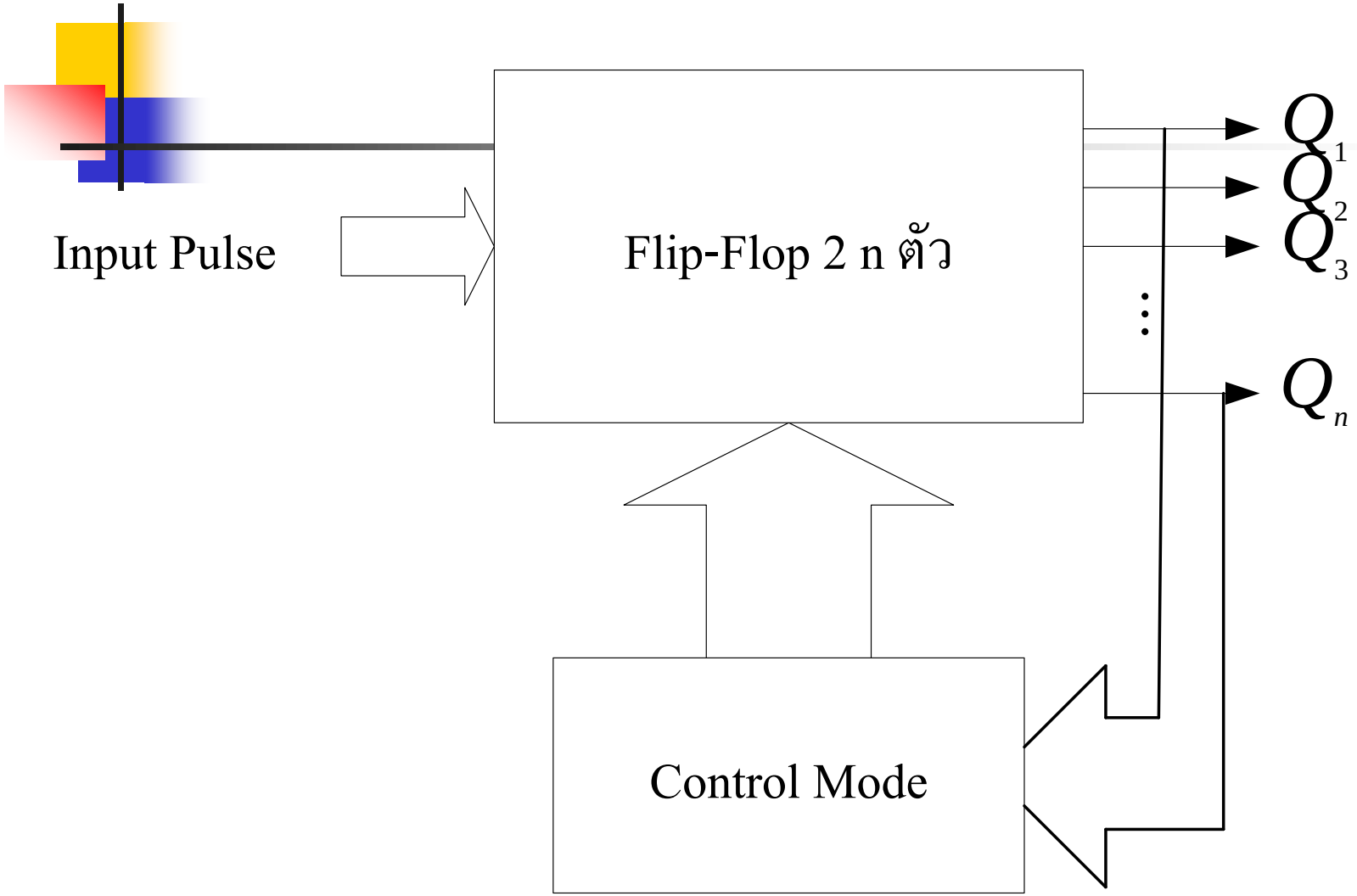

UP/DOWN Counter (Dr.V.K Jain : 1995 : 244)

**ลอจิกเกตที่ใช้ควบคุมการนับขึ้น-นับลง ออกแบบเป็น NAND เกต ทั้งหมด**

# อธิบาย

- วงจรนับขึ้นและลงขนาด 4 บิตแบบริปเปิล โดยใช้ J-K ฟลิบฟลอบ เลือกฟังก์ชัน
- การควบคุมหน้าที่การทำงานการนับขึ้น กับนับลง สามารถส่งถ่ายข้อมูลของ Q และ Complement Q ที่ควบคุมด้วย แนนด์เกต ดังที่กล่าวมาในเรื่องรีจิสเตอร์ สัปดาห์ที่ 12 นักศึกษาลองทบทวน
- การควบคุมการนับขึ้นนำค่าเอาต์พุตจาก Q แต่ละบิตส่งข้อมูลผ่านแนนด์เกต ควบคุมหน้าที่ โดยมีสัญญาณควบคุมเป็น 1 ที่ขานับขึ้น ข้อมูลทางเอาต์พุต Q แต่ละตัวส่งผ่านไปยังอินพุตขา Clock
- ในการนับลงก็นำค่าเอาต์พุตของคอมพลีเมนต์ Q ส่งผ่านไปยังอินพุต Clock ของแต่ละตัว ดังเช่นการนับขึ้น โดยมีสัญญาณควบคุมเป็น1ที่ขานับลง

# 68 การนับตามโหมด



# อธิบาย

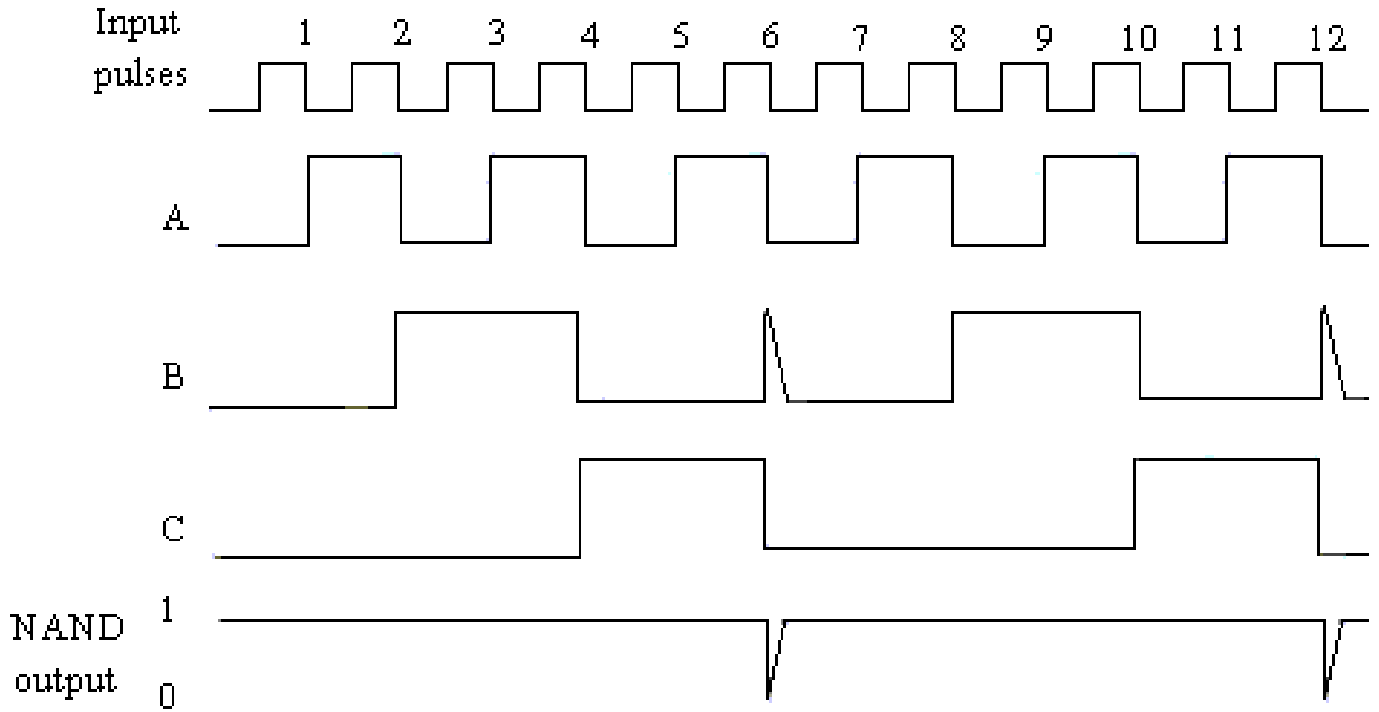
- การนับตามโหมดประกอบด้วยฟลิบฟลอบต่อแบบริปเปิล จำนวนฟลิบฟลอบเป็นไปตามจำนวนการนับของเลขฐานสอง
- จำนวน MOD (MOD Number )

  - MOD-16 คือสภาวะการนับที่แตกต่างกัน 16 สภาวะ ( 0000 ถึง 1111 )
  - ในวงจรนับค่าให้มากขึ้นจำนวน MOD จะเพิ่มขึ้นตามสมการต่อไปนี้

  MOD number = 2N ……( 10-1 ) เมื่อ N เป็นจำนวนฟลิบฟลอบของวงจรนับ
- วงจรคอลโทรลโหมดทำหน้าที่ตรวจจับเอาต์พุตเมื่อถึงค่าเอาต์พุตที่ต้องการจะนับ วงจรจะทำการรีเซ็ต ฟลิบฟลอบทั้งหมดให้เริ่มที่ 0 ใหม่ เป็นการนับไซเคิลที่ 2
- จากการควบคุมการทำงานสามารถกำหนดค่าโหมดได้ทุกค่า
- รูปแบบการนับเอาต์พุตที่บิต $M_{SB}$ จะมีค่าเท่ากับการหาร เช่น MOD-5 เท่ากับหาร 5 ของความถี่ที่ป้อนทางอินพุต

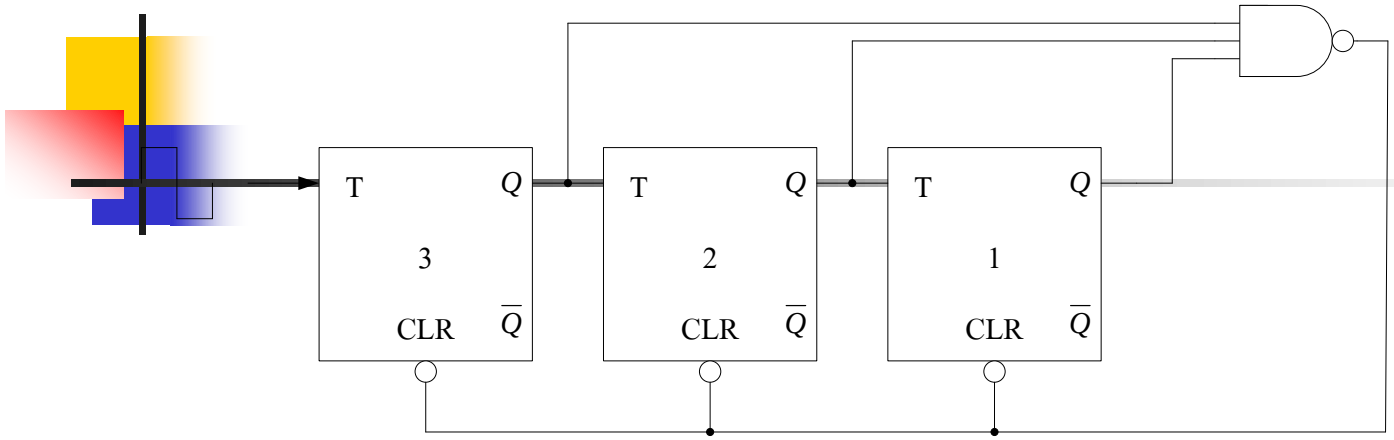
## วงจรนับที่มีจำนวน MOD <$2^N$ (Counters with MOD Number <2N)

- วงจรนับขึ้นพื้นฐาน มีจำนวน MOD เท่ากับ $2^N$ โดย N เป็นจำนวนของ ฟลิบฟลอบ
- สามารถปรับปรุงวงจรนับพื้นฐานนี้ให้มีจำนวน MOD น้อยกว่า $2^N$ ได้ทุกค่า
- โดยการทำให้วงจรนับมีสภาวะการนับกระโดดข้ามลำดับให้มีค่าเป็นเอาต์พุต 0




**แสดงการสร้างสัญญาณ MOD**

# 681 วงจรนับขึ้นโหมด 6 (MOD-6)




| ลำดับ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| 6 | 1 | 1 | 0 |
| | 0 | 0 | 0 |

**สัญญาณเอาต์พุตอยู่ที่ Q**

**สัญญาณReset=$\overline{Q_1.Q_2.Q_3}$=0**

ตัวอย่างวงจรนับตามโหมดจาก 0-5

# อธิบาย

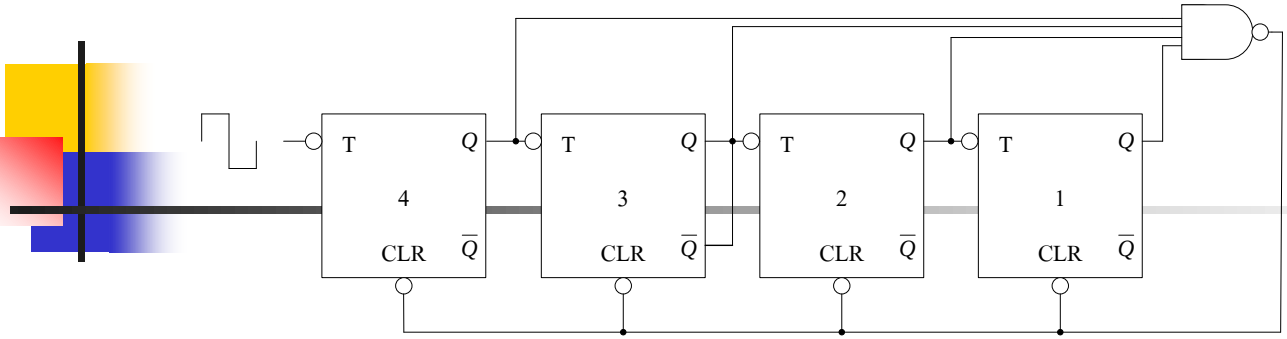
- การสร้างวงจรนับ MOD -6 ที่นับจาก 000 (เลขฐานสิบเป็น 0) ถึง101 (เลขฐานสิบเป็นนับถึง) 0- 5 นับค่าได้ 6 ค่า

**วิธีทำ**

หาค่าฟลิบฟลอบ $2^3 = 8$ และ $2^4 = 16$ ดังนั้น วงจรนับนี้จึงประกอบด้วย ฟลิบฟลอบ จำนวน 3 ตัว

- การที่วงจรนับจะแสดงสภาวะการนับ101ได้นั้น ต้องมีสภาวะการนับมากกว่า 101 อยู่ 1 สภาวะ นั่นคือ วงจรนับต้องเคลียร์เป็น 0 เมื่อการนับเป็น 110 ดังนั้น
- จึงต้องต่อเอาต์พุตของฟลิบฟลอบ1และฟลิบฟลอบ2และคอมพลีเมนต์ฟลิบ-ฟลอบ3 เป็นอินพุตของแนนด์เกตโดยมี (ฟลิบฟลอบ3เป็น $L_{SB}$)
- เมื่อนับมาถึงค่า110 ค่าเอาต์พุตของแนนด์เกตเป็น 0 นำค่านี้ไปต่อกับ CLRของทุก ฟลิบฟลอบ เป็นการเริ่มการนับใหม่

# 682 วงจรนับขึ้นโหมด11 (MOD-11)

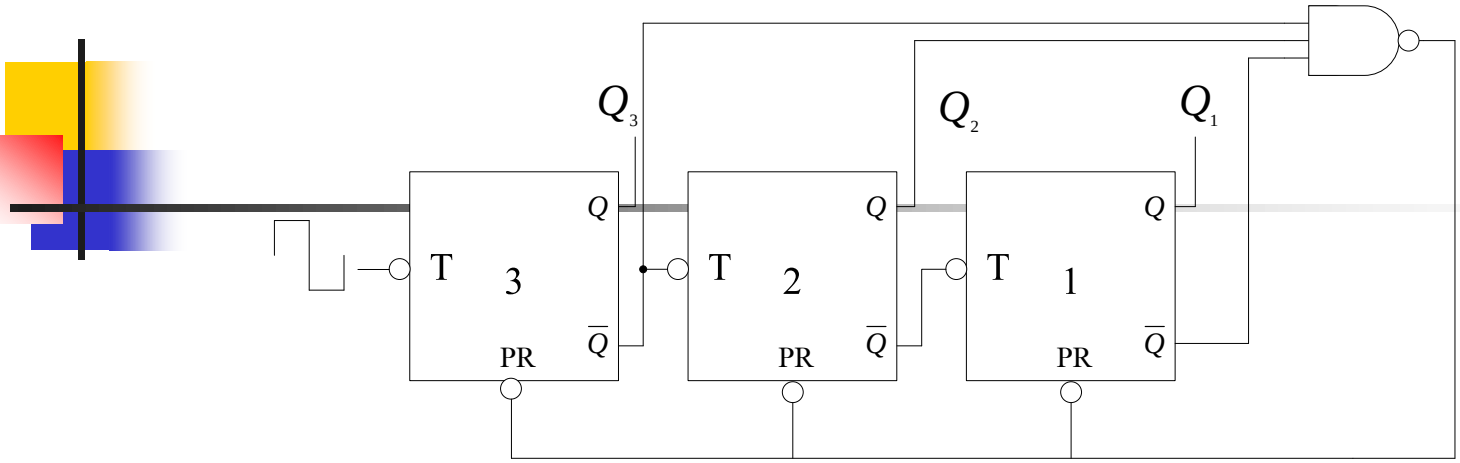



| ลำดับ | $Q_1$ | $Q_2$ | $Q_3$ | $Q_4$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| | 0 | 0 | 0 | 0 |

# อธิบาย

- การสร้างวงจรที่มีการนับจาก 0- 10
- จำนวนฟลิบฟลอบจะต้องนับค่า 0000 – 1010
- จำนวนบิตที่ใช้มีค่าเท่ากับ 4 บิต แต่นับค่าสูงสุดเพียง 1010 แล้วทำการรีเซ็ต เป็น0000 ใหม่
- ดังนั้นการออกแบบเมื่อถึงค่า1011 ให้กำหนดฟังก์ชันการรีเซ็ต โดยนำค่าที่เกิดขึ้นไปผ่านวงจรแนนด์เกตสร้างค่าเอาต์พุตเป็น0 ดังแสดงในตารางความจริง
- ค่าที่เกิดขึ้นจากแนนด์เกตนำไปทำการรีเซ็ตฟลิบฟลอบทุกตัว
- การนับจะเกิดการเวียนในรอบต่อไปอย่างนี้ไม่มีที่สิ้นสุด

# 683 วงจรนับลงโหมด5( MOD-5)

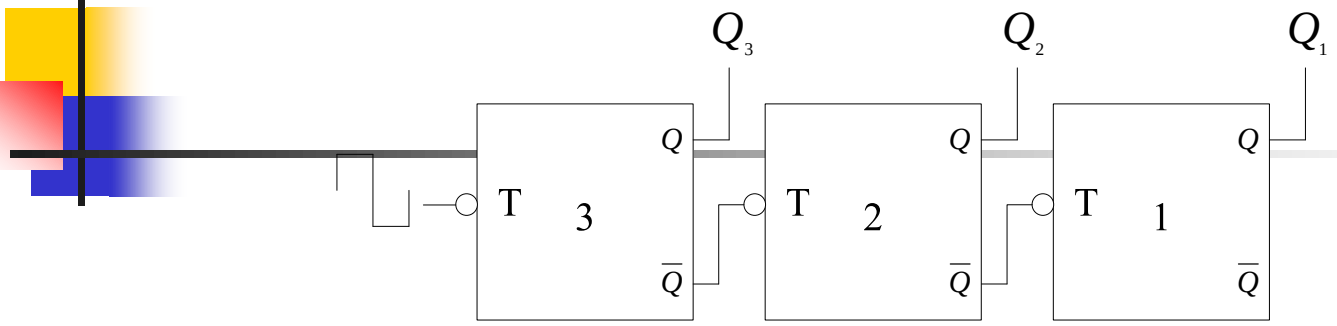



| ลำดับ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|
| สภาวะปกติ | 0 | 0 | 0 |
| Pulse Set (0) | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 |
| | 0 | 0 | 0 |

# อธิบาย

- การทำงานของวงจรนับลงขนาด 3 บิต โหมด 5
- หลักการเช่นเดียวกับการลงที่กล่าวมาแล้ว
- การทำงานจะมีการนับจาก 100 , 011 , 010 , 001- 000 จำนวน 5 ค่า
- สามารถพิจารณาจากตารางความจริงประกอบคำอธิบายที่ผ่านมาในเรื่องการนับค่าลง
- เ มื่อถึงค่า 111 ก็จะนำค่านี้ไปกำหนดโหมดการรีเซ็ตโดยผ่านแนนด์เกตไปบังคับการทำงานของฟลิบฟลอบทุกตัวให้เป็น 0 ดังรูปวงจร
- การนับของวงจรจะถูกกำหนดค่าที่กล่าวมาแล้วตลอดไปจนเริ่มรอบใหม่จนถึง 000 แล้วก็จะเริ่ม 100 ไปในรอบต่อไปจนไม่มีที่สิ้นสุด

# 684วงจรนับลงโหมด8( MOD-8)




| ลำดับ | $Q_1$ | $Q_2$ | $Q_3$ |
|---|---|---|---|
| สภาวะปกติ | 0 | 0 | 0 |
| Pulse Set (0) | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 2 | 1 | 0 | 1 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 0 | 1 | 0 |
| 6 | 0 | 0 | 1 |
| 7 | 0 | 0 | 0 |

# อธิบาย

- วงจรนับ 8 (Mod-8)
- การนับจะเป็นไปตามรหัสเลขฐานสอง
- แปรไปตามสัญญาณคล็อกทางด้านอินพุตโดยนับ 7,6,5,4,3,2,1,0
- เมื่อถึงพัลส์ที่ 8 แล้วทำการรีเซตมีลำดับการทำงานดังนี้

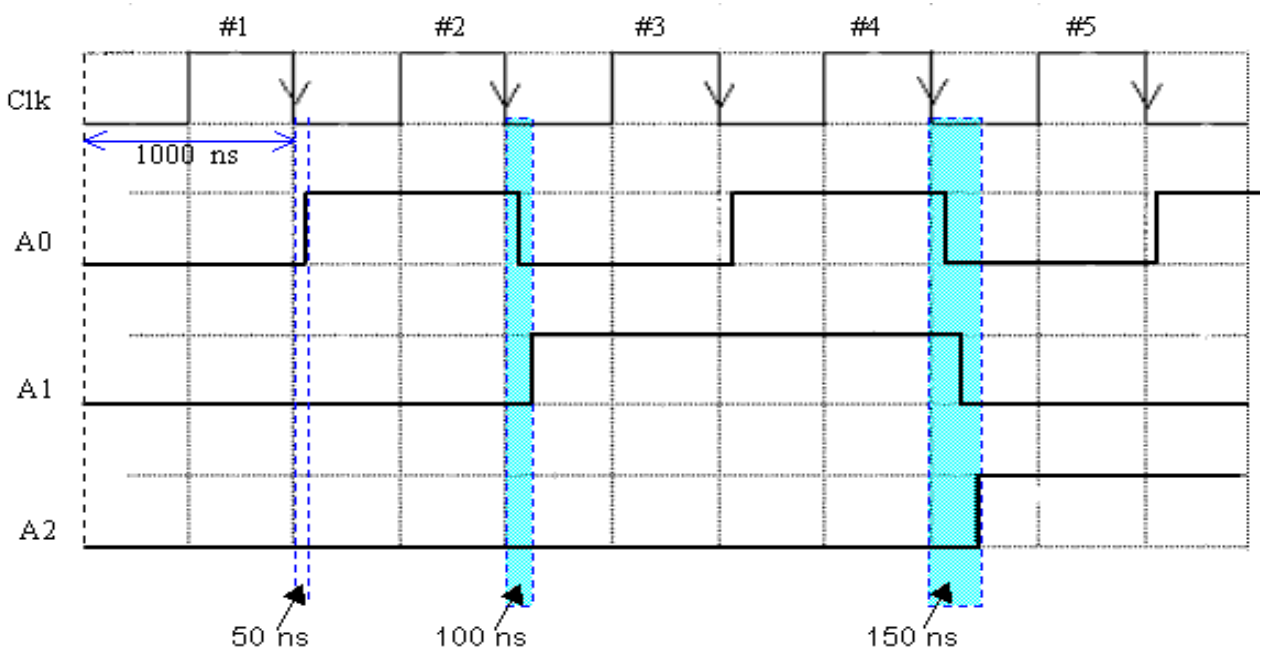
1. ขณะวงจรรีเซต (ไม่มีสัญญาณคล็อก) รหัสเลขฐานสองที่ได้ คือ 111
2. ขณะมีสัญญาณคล็อกของวงจรจะนับตามรหัสเลขฐานสองไปเรื่อย ๆ จาก 111 (เลข7) จนถึง000 (เลข0)

# 69 การนับแบบซิงโครนัส (Synchronous Counters)

## Synchronous Counter (Parallel Counter)

วงจรนับแบบริปเปิล (Asynchronous Counter) ที่ได้ศึกษามาแล้วนั้นใช้ได้ดีกับความถี่ต่ำๆ เนื่องจากปัญหาความล่าช้าของสัญญาณเอาต์พุตหรือเกิดเวลาหน่วงในการส่งข้อมูลจากอินพุตไปยังเอาต์พุต ของฟลิบฟลอปแต่ละตัว(Propagation deley) ถ้าผลรวมของเวลาที่หน่วงของฟลิบฟลอป ทุกตัวค่าน้อยกว่าเวลาที่เกิดสัญญาณคล็อกในหนึ่งลูกคลื่น ก็จะไม่เกิดปัญหาในการนับ จากตัวอย่างเวลาการเกิดหนึ่งลูกคลื่นของสัญญาณคล็อกคือ 1000 ns ฟลิบฟลอปแต่ละตัวมี เวลาหน่วง (Propagation deley) 50 ns ใช้ฟลิบฟลอป 3 ตัว เวลาหน่วงรวม 150 ns ตรงสัญญาณคล็อกลูกที่ 4 จะเห็นว่าเอาต์พุตของฟลิปฟลอปจะเปลี่ยนสถานะได้พอดีและถูกต้อง

แต่ถ้าความถี่คล็อกสูงขึ้น เวลาที่เกิดหนึ่งลูกคลื่นจะสั้นลง เช่น จากตัวอย่าง 1000 ns เปลี่ยนเป็น 120 ns จะทำสถานะเอาต์พุตของฟลิบฟลอป จะไปเปลี่ยนตรงสัญญาณคล็อกลูกที่ 5 นั่นแสดงว่าวงจรทำงานผิดพลาด สำหรับการทำงานของวงจรนับที่ความถี่สูง ๆ จำเป็นที่จะต้องให้ฟลิบฟลอปได้รับสัญญาณกระตุ้นไปพร้อมๆ กันเพื่อหลีกเลี่ยงปัญหาที่เกิดจากการหน่วงเวลาของฟลิบฟลอป วงจรนับชนิดนี้เรียกว่า วงจรนับแบบเข้าจังหวะ (Synchronous Counter)




Waveforms of a 3 bit ripple counter (Tocci, Ronald J. : 1998 : 338)

# วงจรนับชนิด Synchronous

Synchronous หรือ Parallel เป็น Counter ที่ฟลิบฟลอบทุกตัวเปลี่ยนสภาวะพร้อม ๆ กัน เนื่องจากฟลิบฟลอบ แต่ละตัวได้รับการ Trigger จากสัญญาณคล็อกเหมือนกันหมด Counter แบบนี้ จะดีกว่านับแบบริปเปิลเนื่องจากสามารถทำงานในความถี่สูง ๆ ได้ ในขณะที่การนับแบบระลอก ทำงานที่ความถี่สูงมาก แล้วจะเกิดการผิดพลาดขึ้นได้ เนื่องจากฟลิบฟลอบ แต่ละตัวต่อแบบอันดับ การเปลี่ยนสภาวะของ ฟลิบฟลอบ แต่ละตัวของการรอการเปลี่ยนสภาวะจาก ฟลิบฟลอบตัวหน้าก่อนจึงทำงานได้ในความถี่ที่ไม่สูงมากนัก ก่อนที่จะกล่าวถึง Synchronus Counter ก็จะกล่าวถึง Action table ของ JK Flip Flop เสียก่อน

# หลักการสร้างวงจรนับแบบซิงโครนัส

เขียนตารางการนับเมื่อคล็อก พัลส์ป้อนเข้ามา แต่ละลูกจะมีการเปลี่ยนแปลงเป็นอย่างไร ในสภาวะแรกเอาต์พุต Q ของฟลิบฟลอบ ทุกตัวเป็น 1 หมด เมื่อมีสัญญาณคล็อกลูกที่หนึ่งถูกป้อนเข้ามา จะทำให้สภาวะต่อของ Q แต่ละตัวเปลี่ยนแปลงไป คือ

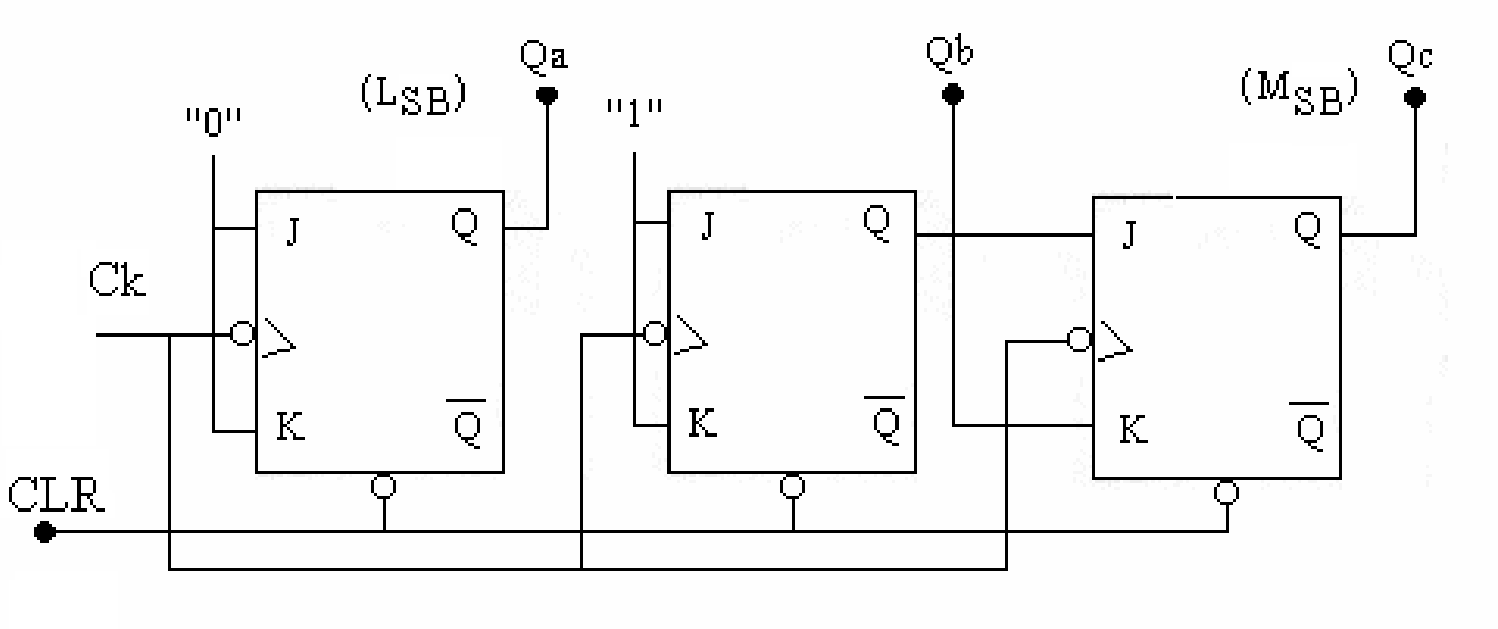
| สัญญาณคล็อก | $Q_c$ | $Q_b$ | $Q_a$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 2 | 1 | 0 | 1 |
| 4 | 0 | 0 | 1 |

**ตารางการนับ MOD-4**

1.เมื่อเขียน Action table ได้แล้ว

2.เราก็หาค่าของ Transition ของเอาต์พุตแต่ละตัวเพื่อกำหนด JK อินพุตแต่ละตัวโดยใช้ K- map

3. เมื่อได้ค่า JK อินพุต แต่ละตัวแล้วนำไปวาดรูปวงจร




**วงจรนับแบบซิงโครนัสขนาด 3บิต**

# ตาราง J – K Flip – Flop Exciting Table

| สถานะเอาต์พุต | | การทำงาน | J | K | สรุปการควบคุม | |
|---|---|---|---|---|---|---|
| Qเดิม | Qnใหม่ | | | | J | K |
| 0 | 0 | Store | 0 | 0 | 0 | X |
| | | Reset | 0 | 1 | | |
| 0 | 1 | Toggle | 1 | 1 | 1 | X |
| | | Set | 1 | 0 | | |
| 1 | 0 | Reset | 0 | 1 | X | 1 |
| | | Toggle | 1 | 1 | | |
| 1 | 1 | Store | 0 | 0 | X | 0 |
| | | Set | 1 | 0 | | |

# อธิบาย

- การทำงานของตาราง J-K ฟลิบฟลอบ Exciting สามารถทำงานได้ 4 รูปแบบ ที่ไม่ซ้ำกันดังจะอธิบายต่อจากนี้
- สถานะเอาต์พุตจะแสดงด้วย Q เป็นสถานะปัจจุบัน ส่วน Qn แสดงสภาวะเมื่อฟลิบฟลอบถูกควบคุมการทำงานที่อินพุต
- สภาวะเดิมเป็น 0 สภาวะใหม่เป็น0 การทำงานของ J-K ฟลิบฟลอบ มี 2 กรณีที่เป็นไปได้ในการควบคุมคือรักษาสภาพเดิมไว้ กับการทำการรีเซ็ตฟลิบฟลอบ
- การรักษาสภาพเดิมต้องควบคุมการทำงาน คือ $J = 0$ , $K = 0$
- การทำการรีเซ็ตต้องควบคุมการทำงาน คือ $J = 0$ , $K = 1$
- การเปลี่ยนสภาวะจากเอาต์พุต0 เป็น0 สรุปกรณีนี้ J ต้องเป็น0 ส่วนK สามารถเป็น1 ก็ได้0 ก็ได้ จึงใช้เครื่องหมาย Don,t Care (X)

# อธิบาย(ต่อ)

- สภาวะเดิมเป็น0 สภาวะใหม่เป็น 1 การทำงานของ J-K ฟลิบฟลอบ มี 2 กรณีที่เป็นไปได้ในการควบคุม คือ การ Toggle กับการ
  - การเซ็ต J =1, K= 0
  - การToggle J=1 , K=1
- การเปลี่ยนสภาวะจากเอาต์พุต 0 เป็น 1 สรุปกรณีนี้ J ต้องเป็น 1 ส่วน K สามารถเป็น1 ก็ได้ 0 ก็ได้ จึงใช้เครื่องหมาย Don't Care (X)
- สภาวะเดิมเป็น1 สภาวะใหม่เป็น 0 การทำงานของ J-K ฟลิบฟลอบ มี 2 กรณีที่เป็นไปได้ในการควบคุมการทำการรีเซ็ตฟลิบฟลอบกับการToggle

  การรีเซ็ตต้องควบคุมการทำงาน คือ J=0 , K=1

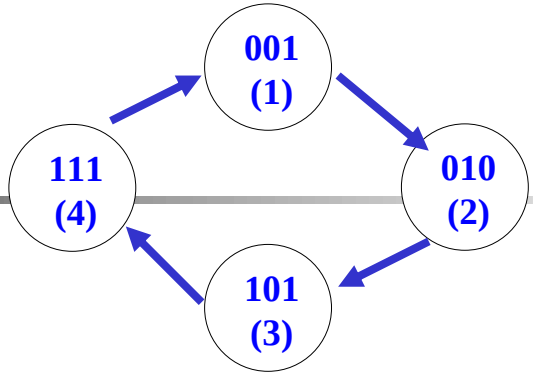
  การ Toggle ต้องควบคุมการทำงาน คือ J=1 , K=1

# อธิบาย(ต่อ)

- การเปลี่ยนสภาวะจากเอาต์พุต1 เป็น 0 สรุปกรณีนี้ J สามารถเป็น1 ก็ได้ 0ก็ได้ จึงใช้เครื่องหมาย Don't Care (X) ส่วน K ต้องเป็น 1
- สภาวะเดิมเป็น 1 สภาวะใหม่เป็น 1 การทำงานของ J-K ฟลิบฟลอบ มี 2 กรณีที่เป็นไปได้ในการควบคุมคือ การรักษาสภาพเดิม กับการเซ็ต
- การรักษาสภาพเดิม ต้องควบคุมการทำงาน คือ J=0 , K=0
- การทำการเซ็ตต้องควบคุมการทำงาน คือ J=1 , K=0
- การเปลี่ยนสภาวะจากเอาต์พุต 1 เป็น 1 สรุปกรณีนี้ J สามารถเป็น 1 ก็ได้ 0 ก็ได้ จึงใช้เครื่องหมาย Don't Care (X) ส่วน K ต้องเป็น 0
- นักศึกษาพึงระลึกถึงการทำงานของการทำงานของ J-K นี้ไว้ให้ดี เพราะจะนำไปกำหนดการทำงานของวงจรนับแบบต่อเนื่อง (Synchronous Counters)

# 691 การออกแบบวงจรนับซิงโครนัส

## Synchonous Counter




| สถานะปัจจุบัน | | | สถานะถัดไป | | |
|---|---|---|---|---|---|
| $Q_2$ | $Q_1$ | $Q_0$ | $Q_2$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

(ก)

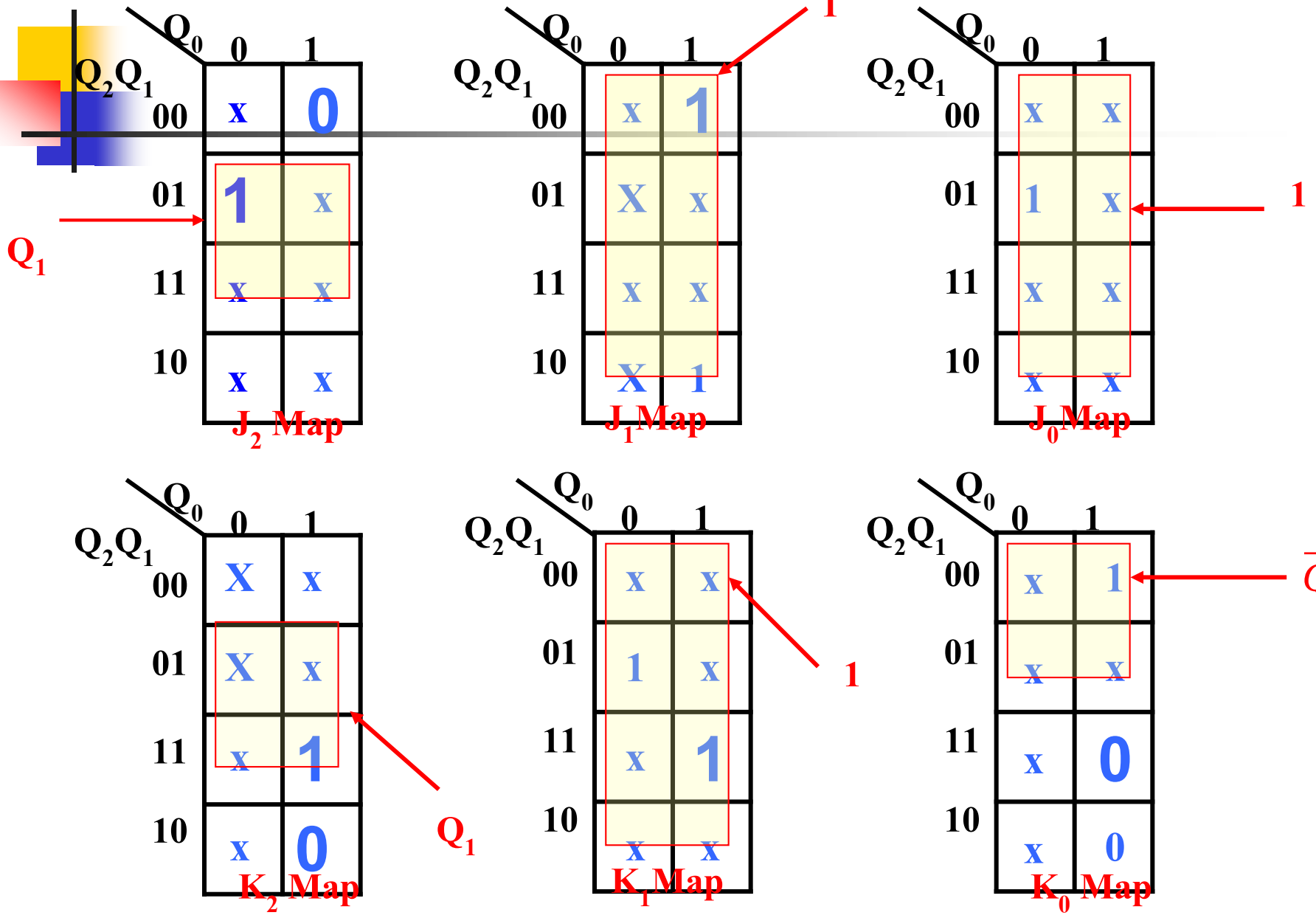
| การเปลี่ยนเอาต์พุต | | | อินพุตฟลิบฟลอป | | |
|---|---|---|---|---|---|
| $Q_{N+1}$ | | $Q_N$ | $J_2$ | | $K_2$ |
| 0 | → | 0 | 0 | | x |
| 0 | → | 1 | 1 | | x |
| 1 | → | 0 | x | | 1 |
| 1 | → | 1 | x | | 0 |

(ข)

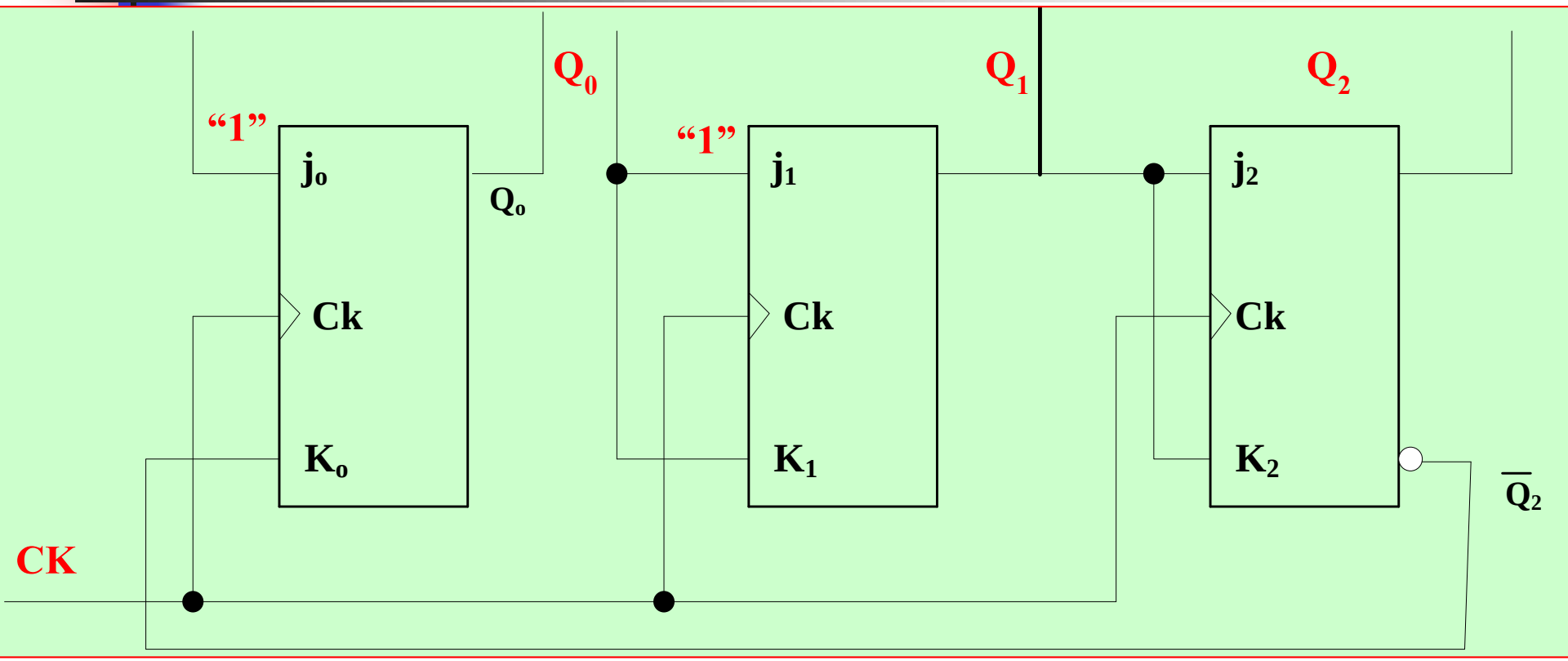
$J_2 = K_2 = Q_1$

$J_1 = K_1 = 1$

$J_0 = 1, K_0 = \overline{Q}_2$

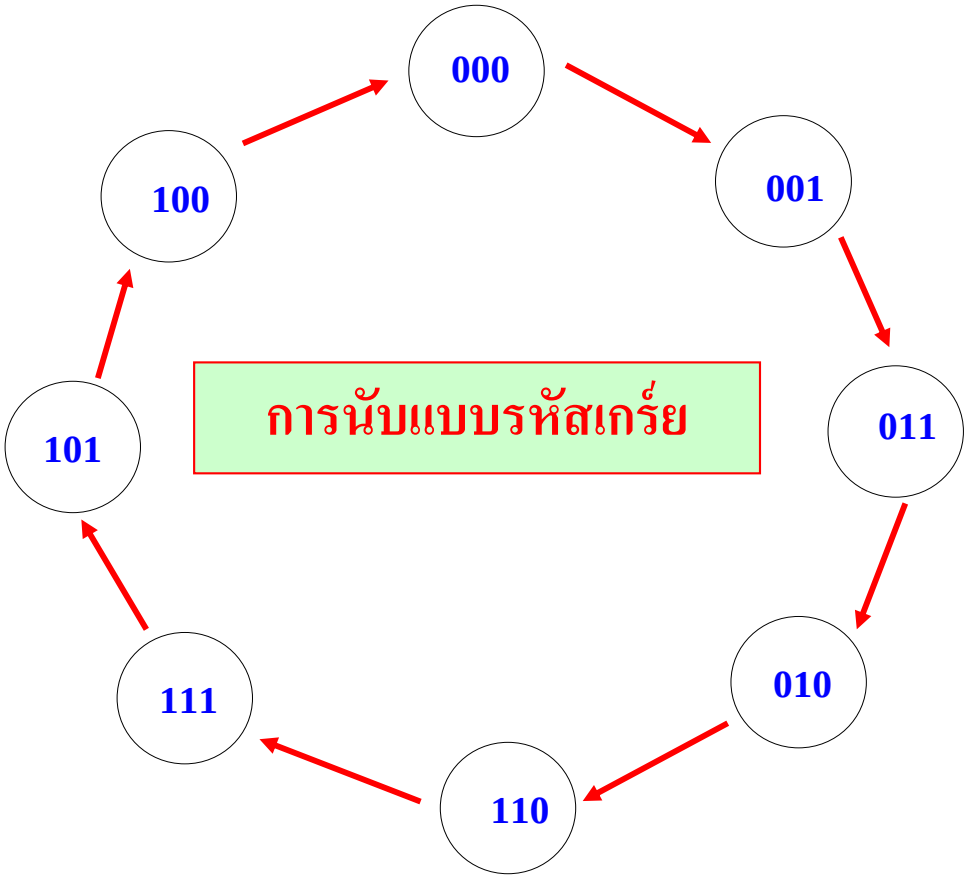


# วงจรนับ 4 ค่าดังตารางการนับ



# 692 การออกแบบวงจรนับซิงโครนัสรหัสเกร์ย

## ขั้นตอนที่1 เขียนแผนภาพสถานะ (State Diagram)



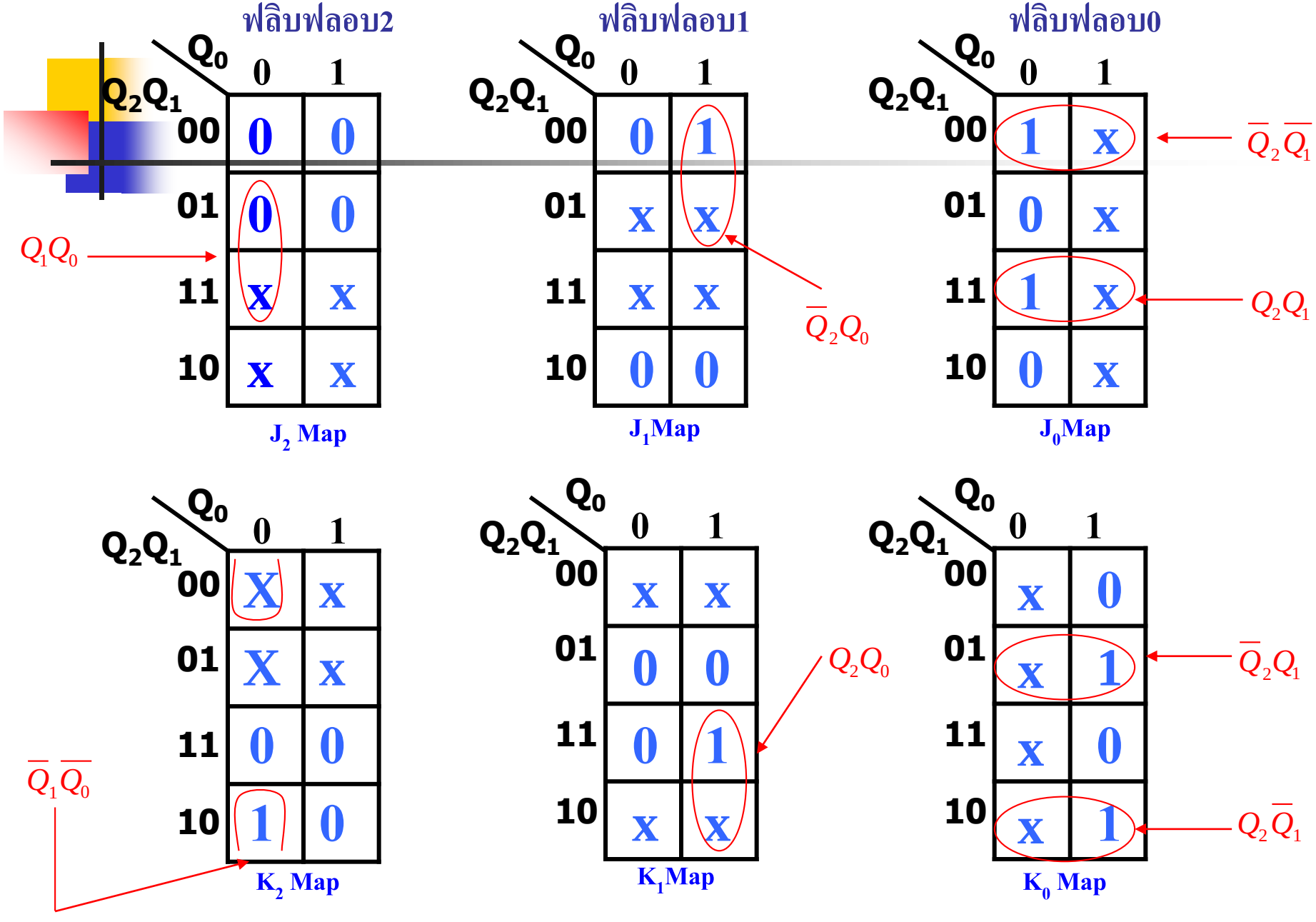
# ขั้นตอนที่ 2 เขียนตารางสถานะ (Next-State Table)

| สถานะปัจจุบัน | | | สถานะต่อไป | | |
|---|---|---|---|---|---|
| $Q_2$ | $Q_1$ | $Q_0$ | $Q_2$ | $Q_1$ | $Q_0$ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |

# ขั้นตอนที่ 3 เลือกฟลิบฟลอปและเขียนตารางการทำงาน

| การเปลี่ยนสถานะเอาต์พุต | | อินพุตของฟลิบฟลอบ | |
|---|---|---|---|
| $Q_N$ | $Q_{N+1}$ | J | K |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

# ขั้นตอนที่ 4 เขียนแผนผังของคาร์นอห์



# ขั้นตอนที่ 5 เขียนสมการลอจิกทางอินพุตของฟลิบฟลอป

$$J_O = Q_2 Q_1 + \overline{Q}_2 \overline{Q}_1 = \overline{Q_2 \oplus Q_1}$$

$$K_0 = Q_2 \overline{Q}_1 + \overline{Q}_2 Q_1 = Q_2 \oplus Q_1$$

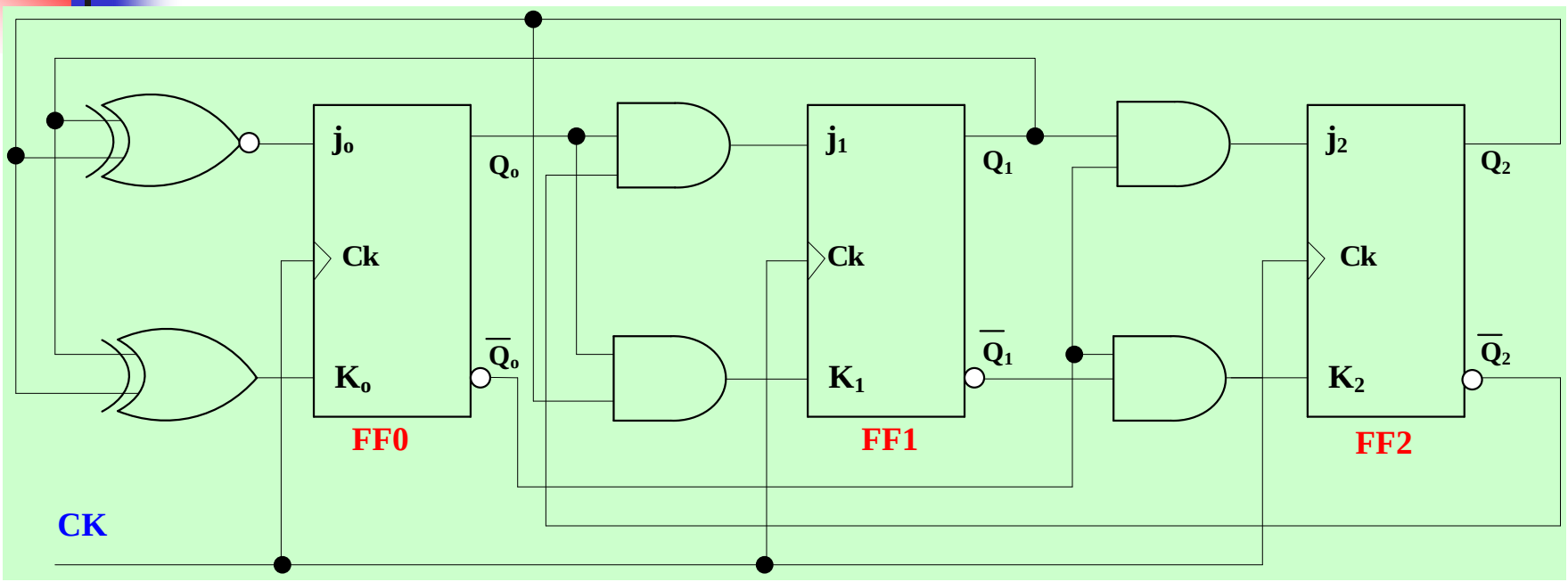
$$J_1 = \overline{Q}_2 Q_0$$

$$K_1 = Q_2 Q_0$$

$$J_2 = Q_1 \overline{Q}_0$$

$$K_2 = \overline{Q}_1 \overline{Q}_0$$

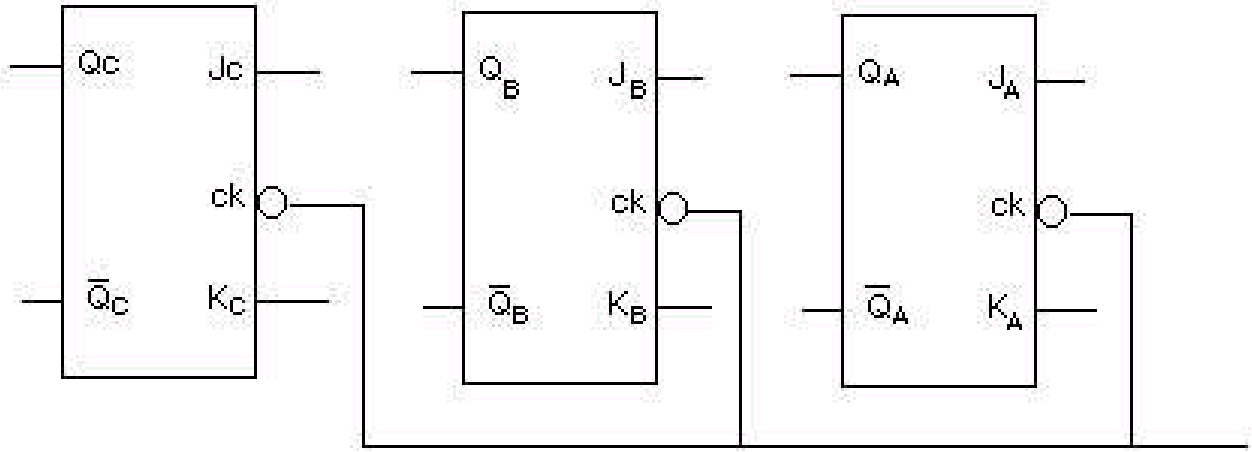
# ขั้นตอนที่ 6 เขียนวงจรนับ




## วงจรนับแบบเกร์ยแบบ 3 บิต

# 693 การออกแบบวงจรนับแบบซิงโครนัสโหมด 5

- ในการออกแบบวงจร Mod – 5 Synchronous Counter แบบต่อเนื่อง การกำหนดขนาดบิตเป็นจุดเริ่มต้นเพื่อนำไปกำหนดค่าในตาราง
- จากนั้นก็จะใช้ Action table เพื่อกำหนดสภาวะเมื่อ Clock pulse ผ่านไปแต่ละลูก
- วิธีการออกแบบ แบ่งออกเป็นขั้นตอน ดังจะอธิบายได้ดังนี้
- ขั้นตอนที่หนึ่งเขียนวงจร JK Flip Flop ตามจำนวนที่ต้องการใช้ ในที่นี้เป็นวงจรนับ 5 จึงใช้ JK Flip Flop 3 ตัว
- กำหนดค่าฟังก์ชันของแต่ละอินพุตว่าสามารถสร้างค่าดังตารางการทำงาน
- ส่วนขา Ck ต่อเข้าด้วยกันทั้ง 3 ฟลิบฟลอบปล่อยลอยไว้ก่อน




**วงจร JK Flip Flop**

**ขั้นตอนที่สอง** เขียนตารางการนับ(แสดงไว้แล้วตามตารางที่ 15.2) ตารางนี้จะทำให้เราทราบว่าเมื่อมี Clock pulse ป้อนเข้ามา แต่ละตัวจะมีการเปลี่ยนแปลงเป็นอย่างไร ในที่นี้สภาวะเริ่มแรกเอาต์พุตQ ของฟลิบฟลอบทุกตัวเป็น0 หมด เมื่อมี Clock pulse ลูกที่หนึ่งถูกป้อนเข้ามา จะทำให้สภาวะต่อไปคือ $Q_c = 0$, $Q_B = 0$ และ $Q_A = 1$ เป็นต้น

| สัญญาณคล็อก | $Q_C$ | $Q_B$ | $Q_A$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 |

**ตารางการนับ**

**ขั้นตอนที่สาม** จากตารางที่ 2 และ Action table ของ JK ฟลิบฟลอบ เราสามารถเขียน Transition table ได้ดังนี้

| Present state | | | Next state | | | Input | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $Q_C$ | $Q_B$ | $Q_A$ | $Q_C$ | $Q_B$ | $Q_A$ | $J_C$ | $K_C$ | $J_B$ | $K_B$ | $J_A$ | $K_A$ |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | D | 0 | D | 1 | D |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | D | 1 | D | D | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | D | D | 0 | 1 | D |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | D | D | 1 | D | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | D | 1 | 0 | D | 0 | D |

**Transition table ของ Mod-5 Synchronous counter**

ตารางTransition จะอธิบายได้ดังนี้คือ ช่อง Next state ได้มาจากตารางการนับค่า กล่าวคือ สภาวะเดิม (Present state) $Q_c = 0$ , $Q_B = 0$, $Q_A = 0$ เมื่อมี Clock pulse ถูกป้อนเข้ามา หนึ่งลูก ทำให้สภาวะต่อไป (Next state) เป็น $Q_c = 0$ , $Q_A = 1$ เป็นต้น ต่อไปช่องของ JK Input แต่ละตัวก็ได้มากจากการตรวจดูว่า เมื่อ คล็อกพัลส์ลูกที่หนึ่งผ่านไปทำให้ $Q_C$ เปลี่ยนสภาวะจาก 0 เป็น 0 (หรือไม่เปลี่ยนสภาวะ) $J_C = 0$, $K_C = d$ (ตามค่าของ Action table) ส่วน $Q_B$ก็เช่นเดียวกัน และ $Q_A$ เปลี่ยนสภาวะจาก 0 เป็น 1 เราจะได้ $J_A = 1$, $K_A = d$ สำหรับ คล็อกพัลส์ลูกที่สองที่ถูกป้อนเข้าไป เราก็พิจารณาแบบเช่นเดียวกัน

**ขั้นตอนที่สี่** เมื่อเขียน Transition table ได้แล้ว เราก็หาค่าของอินพุต JK แต่ละตัวโดยใช้ Karnaugh map ดังนี้

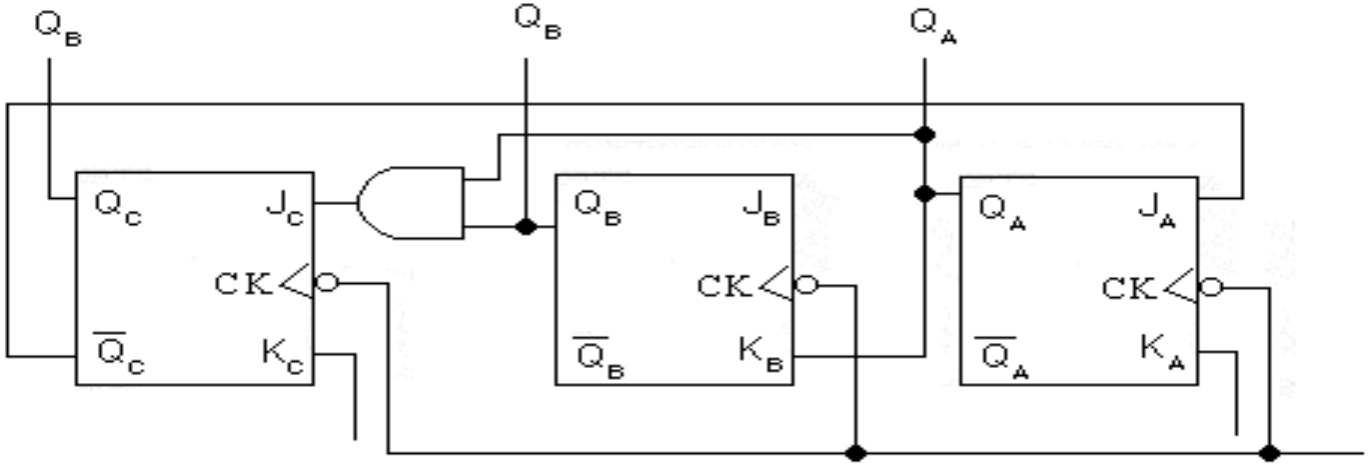
**ขั้นตอนที่ห้า** เมื่อได้ค่าอินพุต JK แต่ละตัวแล้วก็นำไปต่อให้กับวงจร

| $Q_A$ \ $Q_C Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 1 | d | 0 |
| 1 | d | d | d | d |

$J_A = Q_B$

| $Q_A$ \ $Q_C Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | d | d | d | d |
| 1 | 1 | 1 | d | d |

$K_A = 1$



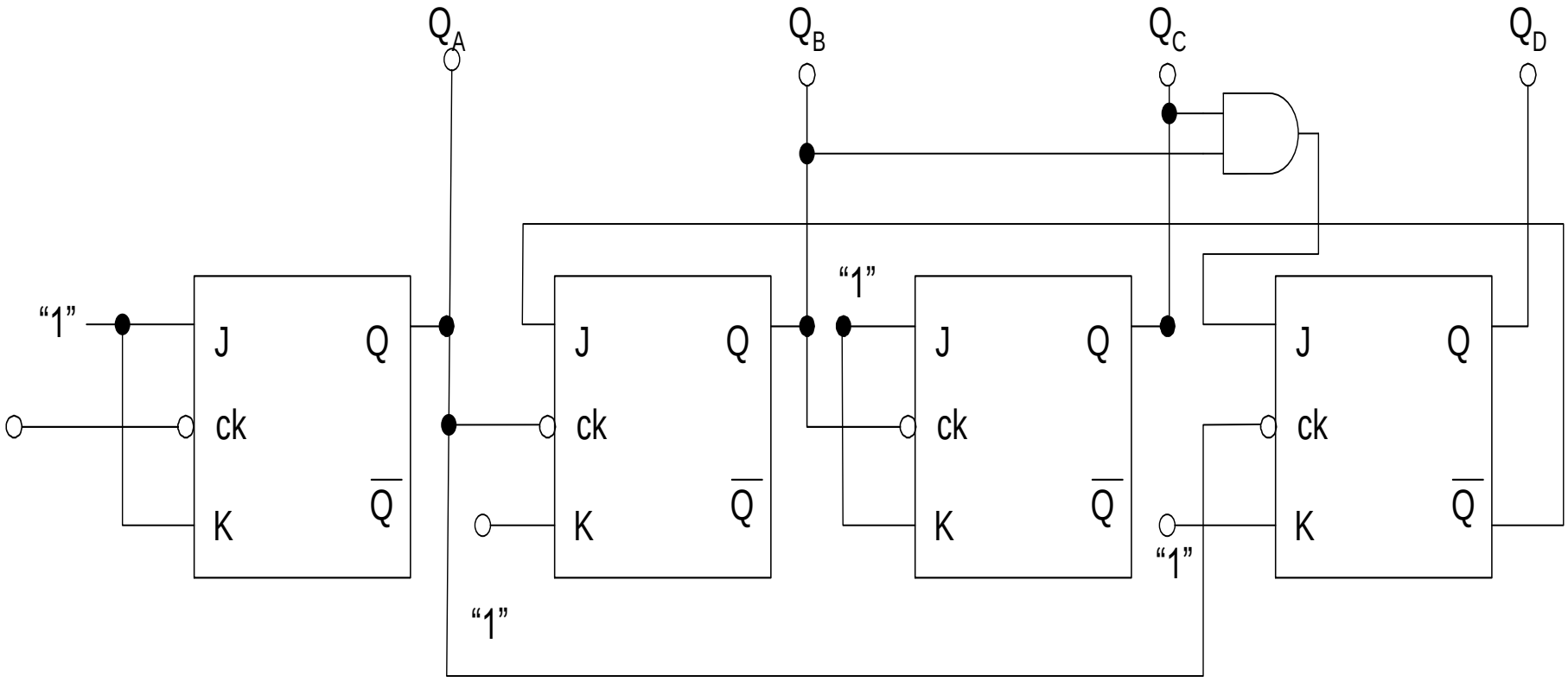

**วงจร Mod – 5 Synchronous Counter.**

# 694 การออกแบบวงจรนับขึ้นสิบ

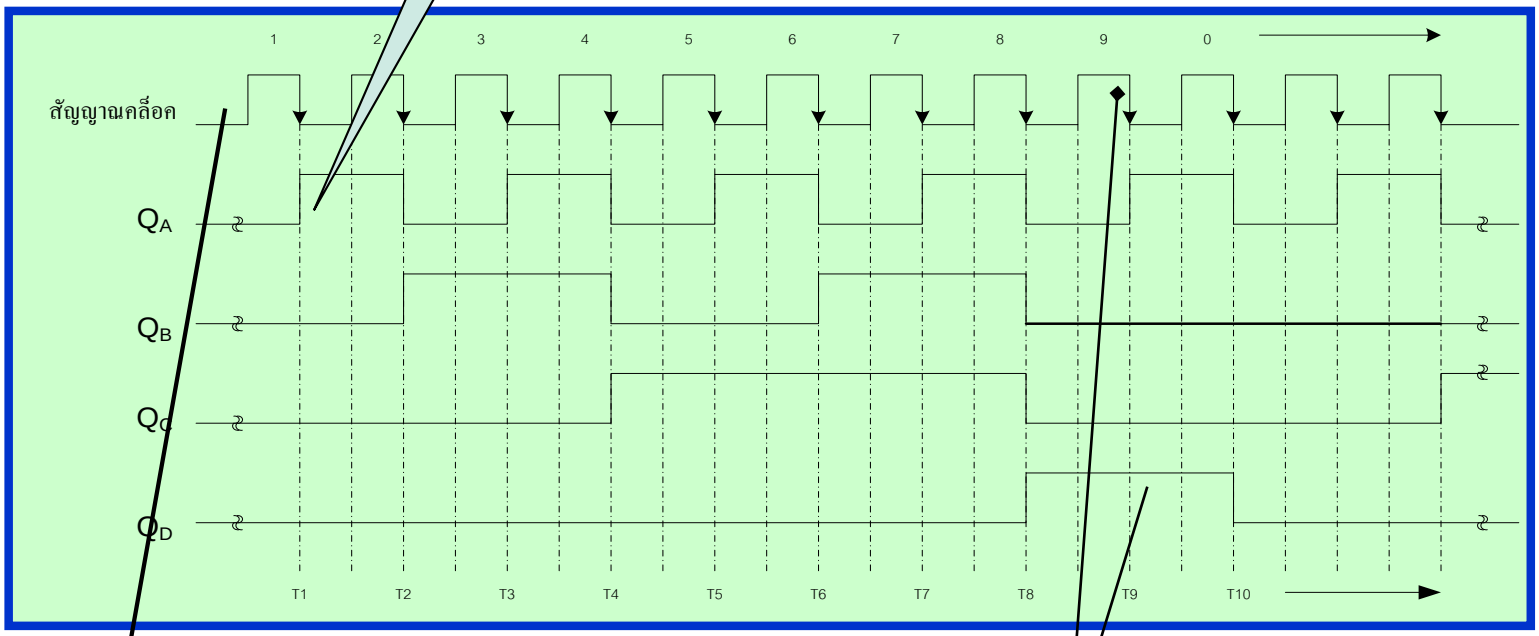
| $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ | เลขฐานสอง |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | 1 | **1** |
| 0 | 0 | 1 | 0 | **2** |
| 0 | 0 | 1 | 1 | **3** |
| 0 | 1 | 0 | 0 | **4** |
| 0 | 1 | 0 | 1 | **5** |
| 0 | 1 | 1 | 0 | **6** |
| 0 | 1 | 1 | 1 | **7** |
| 1 | 0 | 0 | 0 | **8** |
| 1 | 0 | 0 | 1 | **9** |
| 0 | 0 | 0 | 0 | **0** |

**ตารางการทำงานฟลิบฟลอป**

# วงจรนับขึ้นแบบดีเคดเดดชนิดไดเรกรีเซต (นับสิบ)



# แผนภาพไทม์มิ่ง




T2 นับ 2

นับ 1

ที่ T9 นับ 9

# 695 การออกแบบวงจรนับลงแบบซิงโครนัสขนาด 4 บิต

| จำนวนคล็อก | $Q_D$ | $Q_C$ | $Q_B$ | $Q_A$ | การนับฐานสิบ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 15 |
| 2 | 1 | 1 | 1 | 0 | 14 |
| 3 | 1 | 1 | 0 | 1 | 13 |
| ↓ | ตารางเอาต์พุตวงจรนับลง 4 บิต | | | | ↓ |
| 14 | 0 | 0 | 1 | 0 | 2 |
| 15 | 0 | 0 | 0 | 1 | 1 |
| 16 | 0 | 0 | 0 | 0 | 0 |

# Flip Flop A พิจารณา $Q_A$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 → 1 | 1 → 0 | 1 → 0 | 0 → 1 |
| 01 | 0 → 1 | 1 → 0 | 1 → 0 | 0 → 1 |
| 11 | 0 → 1 | 1 → 0 | 1 → 0 | 0 → 1 |
| 10 | 0 → 1 | 1 → 0 | 1 → 0 | 0 → 1 |

$J_A$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | X | X | 1 |
| 01 | 1 | X | X | 1 |
| 11 | 1 | X | X | 1 |
| 10 | 1 | X | X | 1 |

$J_A = 1$

$K_A$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | 1 | X | 1 |
| 01 | X | 1 | X | 1 |
| 11 | X | 1 | X | 1 |
| 10 | X | 1 | X | 1 |

$K_A = 1$

# Flip Flop B พิจารณา $Q_B$

## พิจารณา $Q_B$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0<br>0→0 | 1<br>0→1 | 3<br>1→0 | 2<br>1→1 |
| 01 | 4<br>0→0 | 5<br>0→1 | 7<br>1→0 | 6<br>1→1 |
| 11 | 12<br>0→0 | 13<br>0→1 | 15<br>1→0 | 14<br>1→1 |
| 10 | 8<br>0→0 | 9<br>0→1 | 11<br>1→0 | 10<br>1→1 |

$K_B$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | **1** | **X** | X |
| 01 | 0 | **1** | **X** | X |
| 11 | 0 | **1** | **X** | X |
| 10 | 0 | **1** | **X** | X |

$J_B = Q_A$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | **X** | **1** | 0 |
| 01 | X | **X** | **1** | 0 |
| 11 | X | **X** | **1** | 0 |
| 10 | X | **X** | **1** | 0 |

$K_B = Q_A$

$J_B = Q_A$

$K_B = Q_A$

# พิจารณา $Q_C$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 → 0 | 0 → 0 | 0 → 1 | 0 → 0 |
| 01 | 1 → 1 | 1 → 1 | 1 → 0 | 1 → 1 |
| 11 | 1 → 1 | 1 → 1 | 1 → 1 | 1 → 1 |
| 10 | 0 → 0 | 0 → 0 | 0 → 0 | 0 → 0 |

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | **1** | 0 |
| 01 | X | X | **X** | X |
| 11 | X | X | **X** | X |
| 10 | 0 | 0 | **1** | 0 |

$J_C = Q_AQ_B$

$K_C$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | **X** | X |
| 01 | 0 | 0 | **1** | 0 |
| 11 | 0 | 0 | **1** | 0 |
| 10 | X | X | **X** | X |

$K_C = Q_AQ_B$

$J_C = Q_AQ_B$

$K_C = Q_AQ_B$

# พิจารณา $Q_D$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0: 0→0 | 1: 0→0 | 3: 0→0 | 2: 0→0 |
| 01 | 4: 0→0 | 5: 0→0 | 7: 0→1 | 6: 0→0 |
| 11 | 12: 1→1 | 13: 1→1 | 15: 1→0 | 14: 1→1 |
| 10 | 8: 1→1 | 9: 1→1 | 11: 1→1 | 10: 1→1 |

$J_D$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | X | X | X | X |
| 10 | X | X | X | X |

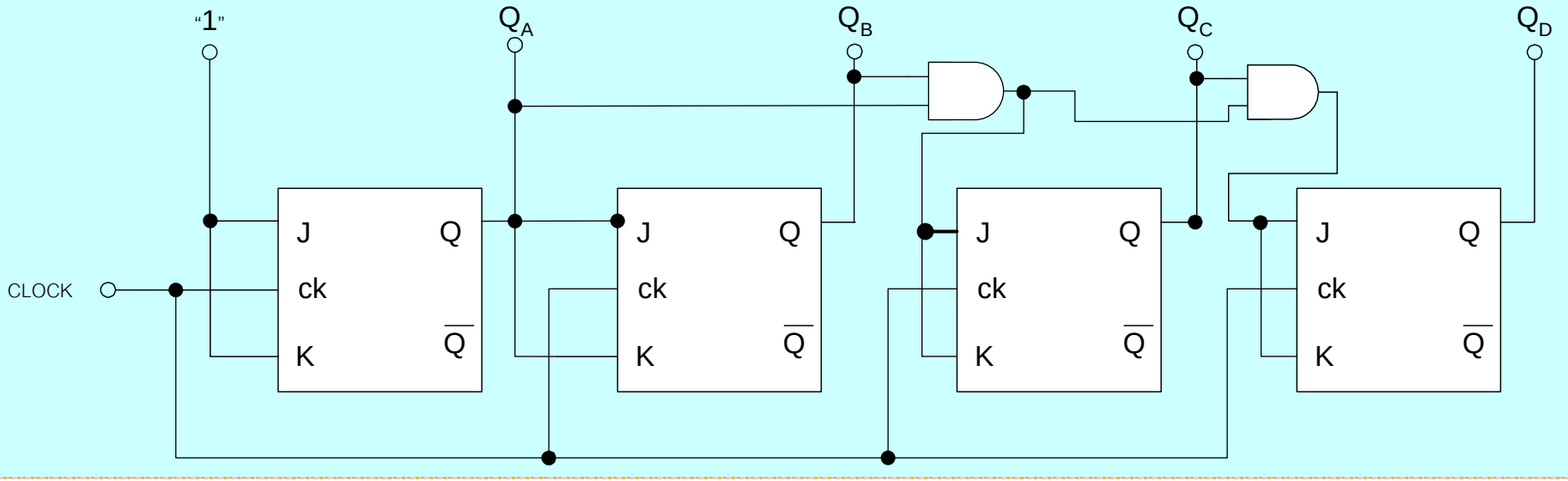
$K_D$

| $Q_DQ_C$ \ $Q_BQ_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | X | X | X | X |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 |

$J_D = Q_A Q_B Q_C$

$K_D = Q_A Q_B Q_C$

# การนับลงขนาด 4 บิต



# 696 การออกแบบวงจรนับแบบซิงโครนัสชนิดกระโดดขนาด 4 บิต

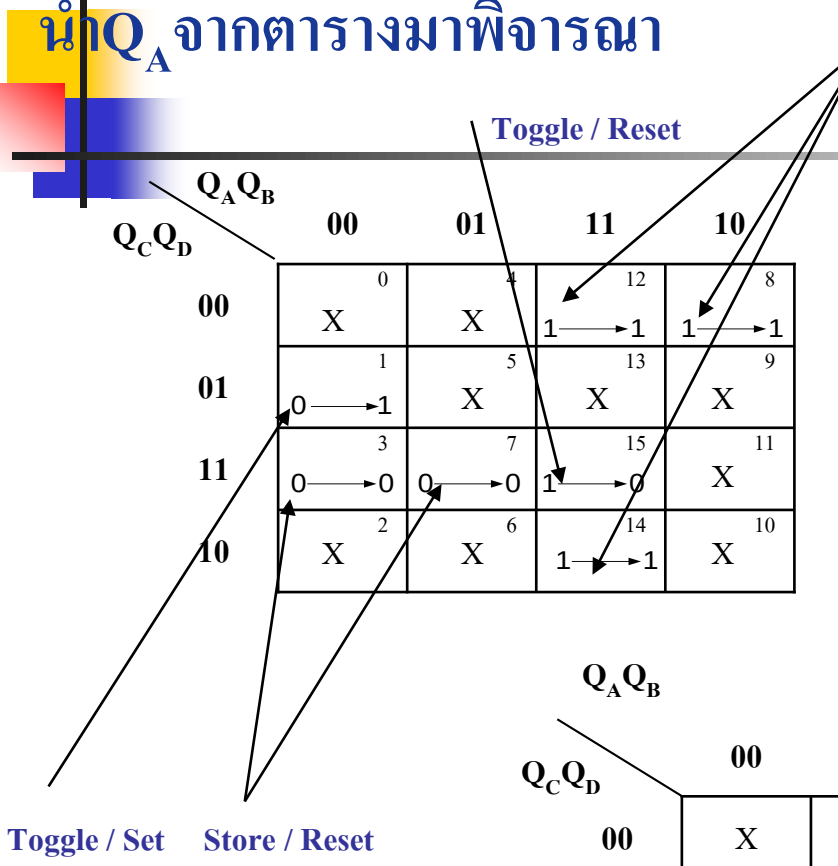
| เลขฐานสิบ | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|
| 8 | 1 | 0 | 0 | 0 |
| 12 | 1 | 1 | 0 | 0 |
| 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 |

# อธิบาย

- การสร้างวงจรนับแบบกระโดดสามารถนำรูปแบบการนับแบบต่อเนื่องมาใช้งาน
- รูปแบบการนับเป็นเลข 8 ,12 ,14 , 15 , 7 , 3 , 1 , 8 วนกลับไปมา
- จากตารางการนับค่ากำหนดค่าในตารางการนับเป็นเลขฐานสอง
- พิจารณาที่$Q_A$ จากการนับเลข 8ไปเป็นเลข 12 $Q_A$มีการเปลี่ยนค่าจาก 1เป็น 1
- นับค่าเลข12 เป็น 14 ค่าที่เปลี่ยนจาก1เป็น1
- นับค่าเลข14 เป็น 15 ค่าที่เปลี่ยนจาก1เป็น1
- นับค่าเลข15 เป็น 7 ค่าที่เปลี่ยนจาก1เป็น0
- นับค่าเลข7 เป็น 3 ค่าที่เปลี่ยนจาก0เป็น0
- นับค่าเลข3 เป็น 1 ค่าที่เปลี่ยนจาก0เป็น0
- นับค่าเลข1 เป็น 8 ค่าที่เปลี่ยนจาก0เป็น1
- นำค่าที่เปลี่ยนแปลงไปลงใน K-map ที่มี $Q_A$,$Q_B$,$Q_C$,$Q_D$ เป็นตัวแปรค่า
- พิจารณา $Q_B$,$Q_C$,$Q_D$ ในทำนองเดียวกันนี้ ก็สามารถนำอินพุตของแต่ละฟลิบฟลอบไปต่อกับวงจรโดยตรงดังอธิบายใน K-map ต่อไปนี้

# วิธีการออกแบบวงจรนับซิงโครนัสแบบกระโดดขนาด 4 บิต

## นำ $Q_A$ จากตารางมาพิจารณา




| $Q_AQ_B$ / $J_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | 1 | X | X | X |
| 11 | 0 | 0 | X | X |
| 10 | X | X | X | X |

$J_A = \overline{Q_C}$

| $Q_CQ_D$ / $K_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | 0 | 0 |
| 01 | X | X | X | X |
| 11 | X | X | 1 | X |
| 10 | X | X | 0 | X |

$K_A = Q_D$

ฟลิปฟลอบ A ต่อดังนี้

$$J_A = \overline{Q_C}$$

$$K_A = Q_D$$

# นำ $Q_B$ จากตารางมาพิจารณา

| $Q_C Q_D$ \ $Q_A Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0: X | 4: X | 12: 1→1 | 8: 0→1 |
| 01 | 1: 0→1 | 5: X | 13: X | 9: X |
| 11 | 3: 0→0 | 7: 1→0 | 15: 1→1 | 11: X |
| 10 | 2: X | 6: X | 14: 1→1 | 10: X |

$J_B$

| $Q_D Q_C$ \ $Q_A Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | 1 | X | X | X |
| 11 | 0 | 0 | X | X |
| 10 | X | X | X | X |

$J_B = Q_A$

$K_B$

| $Q_D Q_C$ \ $Q_A Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | 0 | 0 |
| 01 | X | X | X | X |
| 11 | X | 1 | 0 | X |
| 10 | X | X | 0 | X |

$K_B = \overline{Q}_A$

ฟลิบฟลอบ B ต่อดังนี้

$J_B = Q_A$

$K_B = \overline{Q}_A$

# นำ $Q_C$ จากตารางมาพิจารณา

| $Q_D Q_C$ \ $Q_B Q_A$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0: X | 4: X | 12: 0→1 | 8: 0→0 |
| 01 | 1: 0→0 | 5: X | 13: X | 9: X |
| 11 | 3: 1→0 | 7: 1→1 | 15: 1→1 | 11: X |
| 10 | 2: X | 6: X | 14: 1→1 | 10: X |

| $Q_D Q_C$ \ $Q_A Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | 1 | 0 |
| 01 | 0 | X | X | X |
| 11 | X | X | X | X |
| 10 | X | X | X | X |

$J_C$

$J_C = Q_B$

| $Q_D Q_C$ \ $Q_A Q_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | X | X | X | X |
| 11 | 1 | 0 | 0 | X |
| 10 | X | X | 0 | X |

$K_C$

$K_C = \overline{Q_B}$

ฟลิบฟลอบ C ต่อดังนี้

$J_C = Q_B$

$K_C = \overline{Q_B}$

# นำ $Q_D$ จากตารางมาพิจารณา

| $Q_CQ_D$ \ $Q_AQ_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 X | 4 X | 12 0→0 | 8 0→0 |
| 01 | 1 1→0 | 5 X | 13 X | 9 X |
| 11 | 3 1→1 | 7 1→1 | 15 1→1 | 11 X |
| 10 | 2 X | 6 X | 14 0→1 | 10 X |

| $Q_DQ_C$ \ $Q_AQ_B$ ($J_D$) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | 0 | 0 |
| 01 | X | X | X | X |
| 11 | X | X | X | X |
| 10 | X | X | 1 | X |

$J_D = Q_C$

| $Q_DQ_C$ \ $Q_AQ_B$ ($K_D$) | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | X | X | X | X |
| 01 | 1 | X | X | X |
| 11 | 0 | 0 | 0 | X |
| 10 | X | X | X | X |

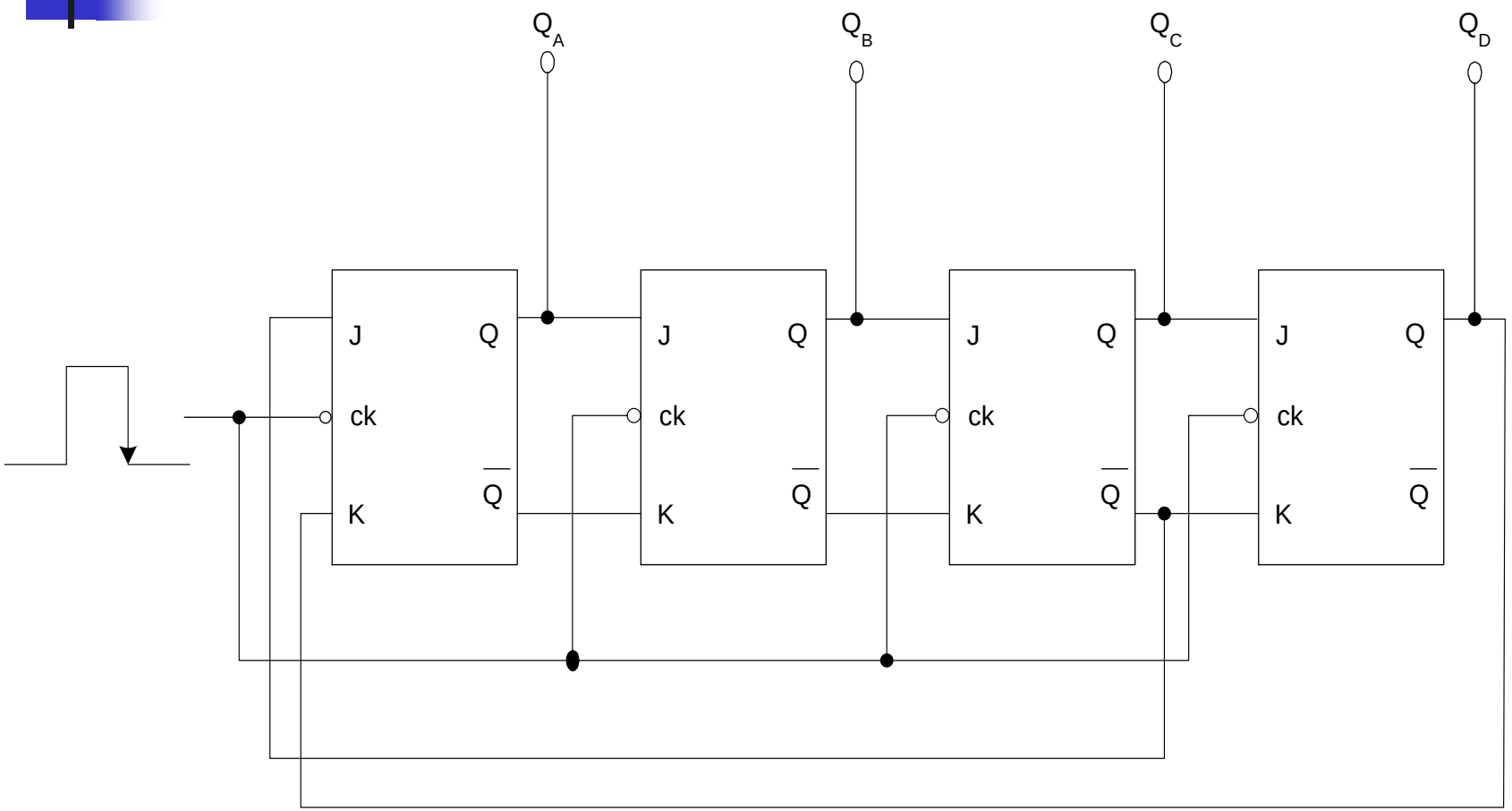
$K_D = \overline{Q_C}$

ฟลิบฟลอบ D นำไปต่อ

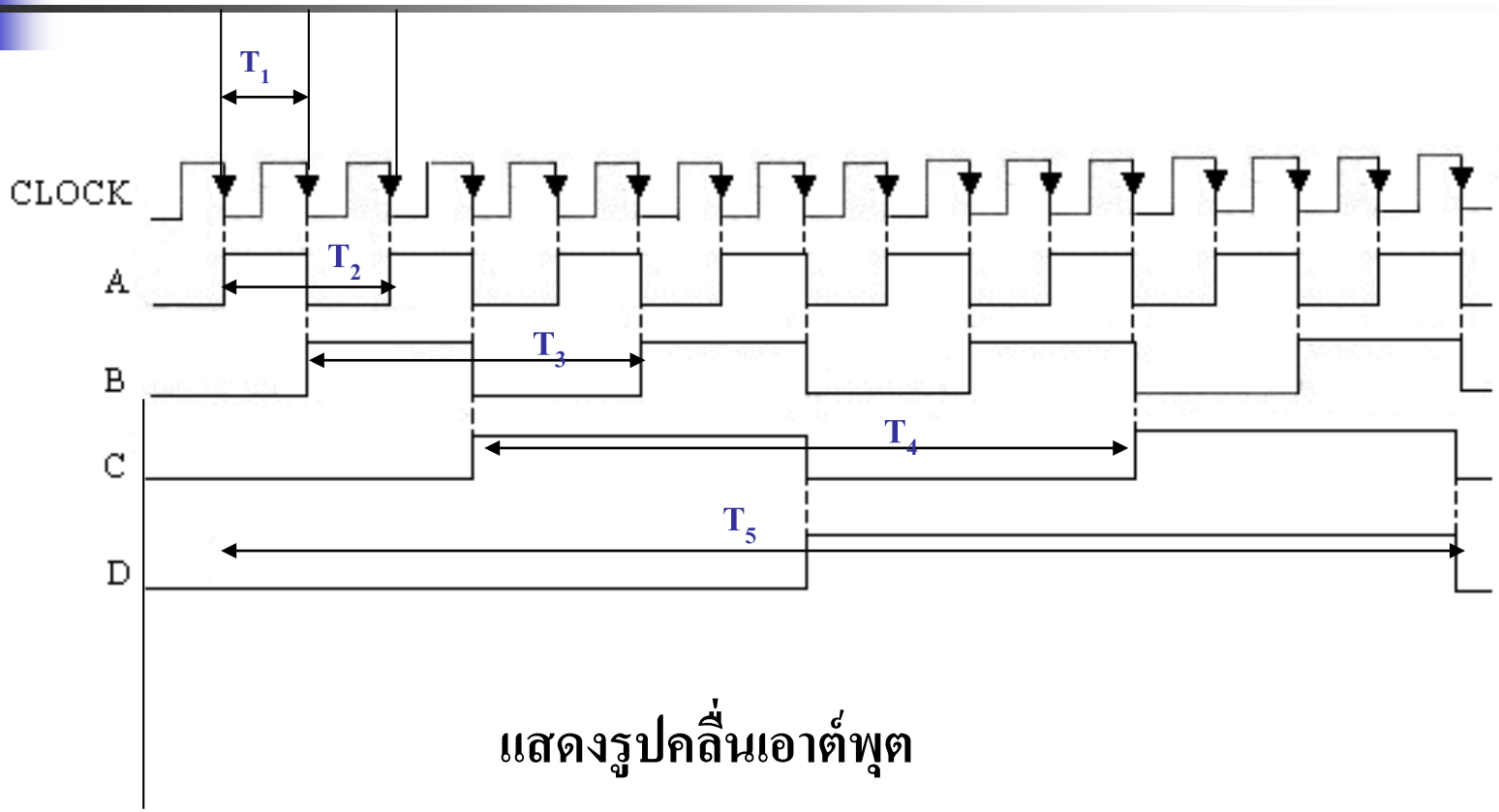
$J_D = Q_C$

$K_D = \overline{Q_C}$

# วงจรนับแบบกระโดดข้าม ที่ออกแบบเป็นซิงโครนัสขนาด 4 บิต



# 697 การนำวงจรนับไปใช้ในการหารความถี่ (Frequency Division)

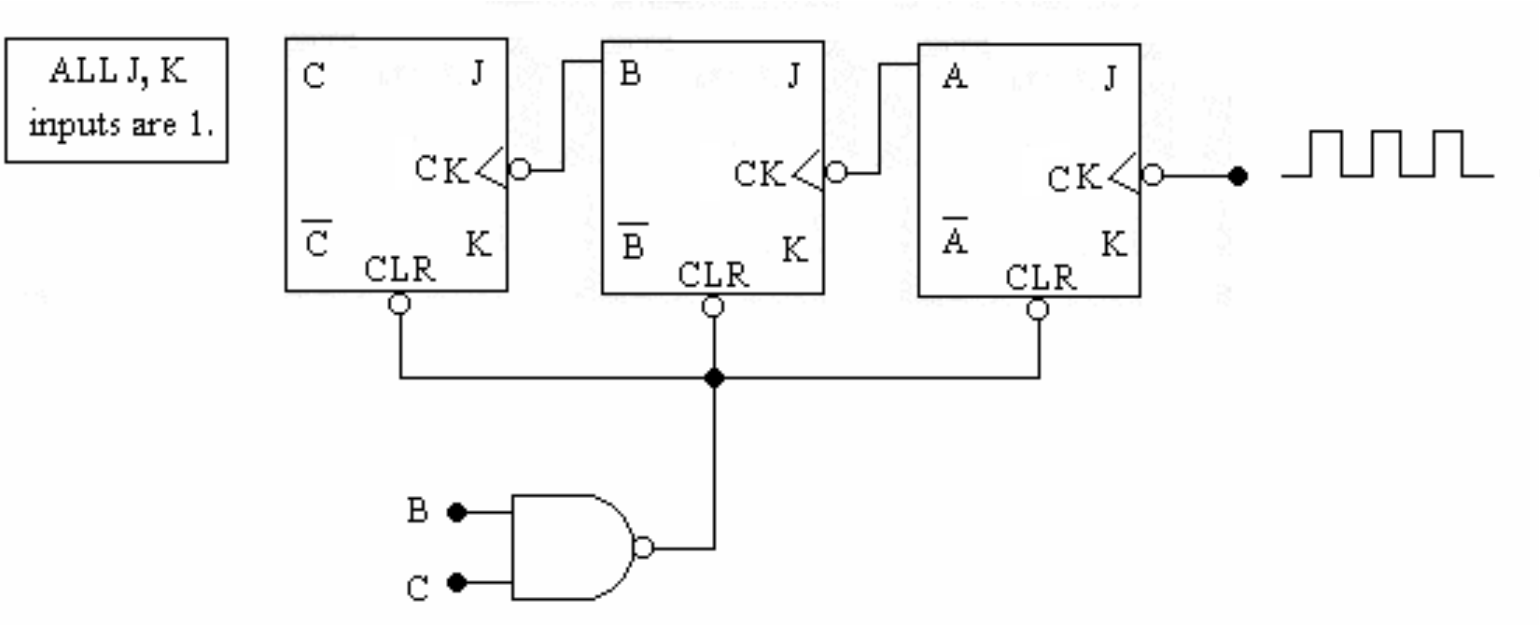



แสดงรูปคลื่นเอาต์พุต

# อธิบาย

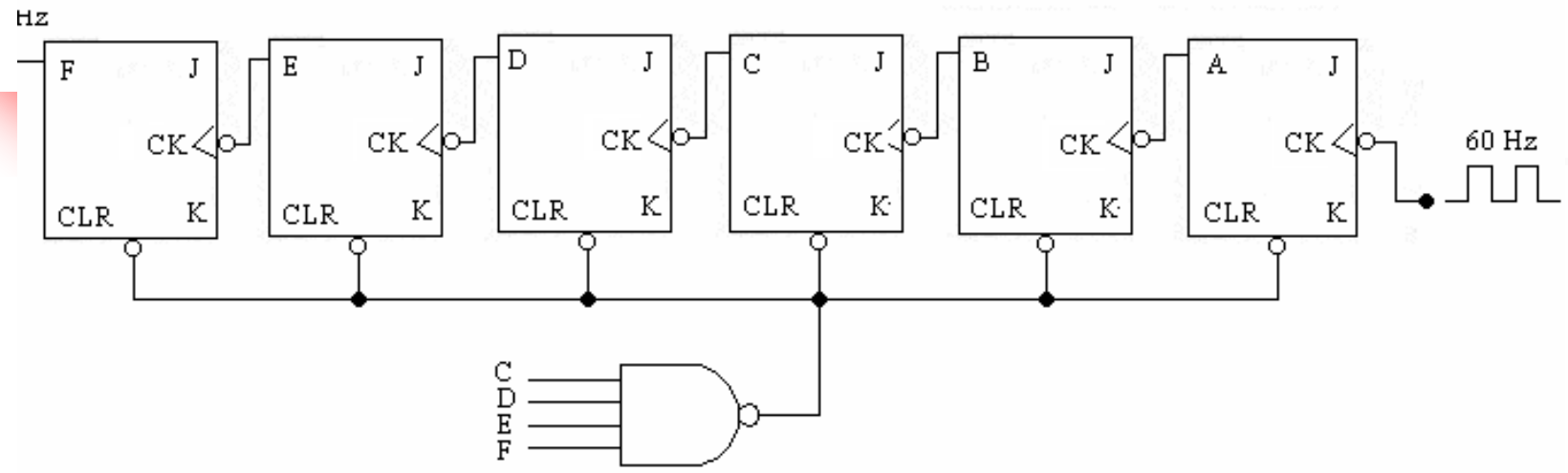
- พัลส์ทางเอาต์พุตของฟลิบฟลอบ แต่ละตัวจะเป็นสัดส่วนกับความถี่สัญญาณคล็อกของอินพุต
- ความถี่สัญญาณคล็อกของอินพุตในตารางเป็น 16 KHz
- หาความถี่รูปคลื่นเอาต์พุตของฟลิบฟลอบ แต่ละตัวจากรูป ได้ดังนี้
- รูปคลื่นเอาต์พุตฟลิบฟลอบA เป็นคลื่นสี่เหลี่ยมที่มีความถี่ 8 KHz,
- รูปคลื่นเอาต์พุตB มีความถี่ 4 KHz, รูปคลื่นเอาต์พุต C มีความถี่ 2 KHz และรูปคลื่นเอาต์พุตD มีความถี่ 1 KHz
- สังเกตว่ารูปคลื่นเอาต์พุตของหารด้วย 2 จนถึงฟลิบฟลอบ D มีความถี่เท่ากับความถี่สัญญาณคล็อกของอินพุตหารด้วย 16
- ทำให้ทราบว่าความถี่เอาต์พุตของ ฟลิบฟลอบ ตัวสุดท้ายในวงจรนับ คือ ความถี่สัญญาณคล็อกของอินพุตหารด้วยจำนวน MOD ของวงจรนับนั้น

# การสร้างสัญญาณคล็อก



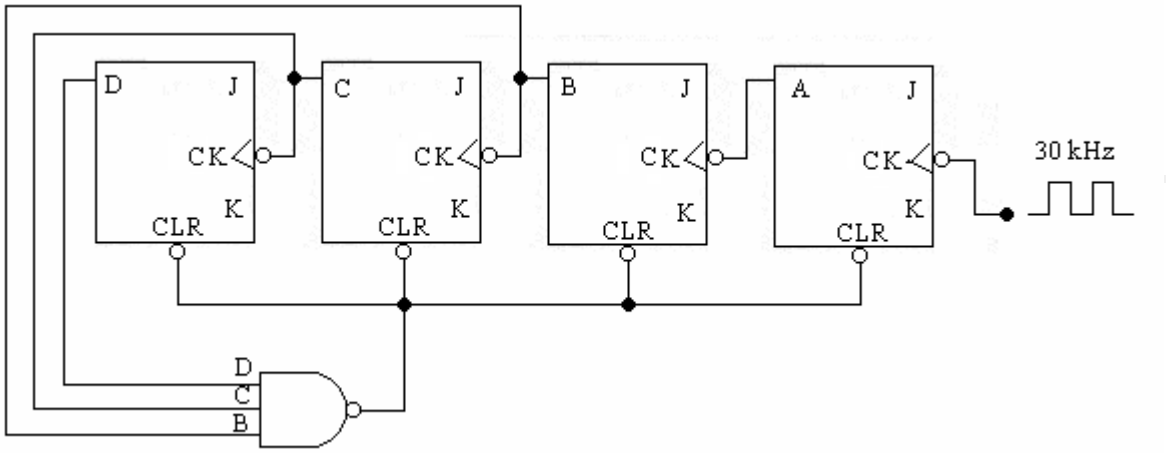

- การสร้างสัญญาณคล็อกจากรูปคลื่นไซน์ความถี่ 60 Hz ในระบบดิจิตอล คือ ป้อนรูปคลื่นไซน์ความถี่ 60 Hz
- วงจรปรับรูปคลื่น (Shaper) เปลี่ยนรูปคลื่นไซน์ดังกล่าว เป็นรูปคลื่นสี่เหลี่ยม ความถี่ 60 Hz
- แล้วป้อนรูปคลื่นสี่เหลี่ยม 60 Hz นี้ไป

# การหารความถี่สัญญาณคล็อกจาก 60 Hz เป็น 1Hz




- วงจรนับ MOD -60 ใช้ในการหารความถี่สัญญาณคล็อกจาก 60 Hz เป็น 1Hz
- วิธีการออกแบบวงจรนับ 60
- การหาจำนวนฟลิบฟลอบ $2^5$=32 ถึง $2^6$= 64 วงจรนับนี้จึงประกอบด้วยฟลิบฟลอบจำนวน 6 ตัว -อินพุต J-K ทำการต่อแบบ T Flip-Flop
- ต้องเคลียร์วงจรนับเมื่อค่าของการนับเป็น 60 (111100)
- ต้องต่อเอาต์พุตฟลิบฟลอบ C,D,E และ F เข้ากับอินพุตของเกต NAND สร้างสัญญาณรีเซ็ตฟลิบฟลอบเมื่อนับถึง 60 จึงทำให้รูปคลื่นเอาต์พุตของ ฟลิบฟลอบ F มีค่าความถี่เท่ากับ1 Hz

# 698 การนำวงจรนับไปใช้ประยุกต์ใช้งาน

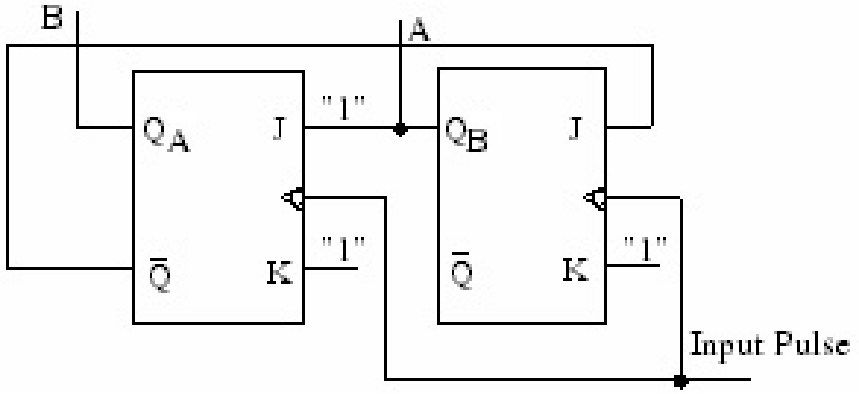



- J , K เป็น HIGH จงหาจำนวน MOD และความถี่เอาต์พุตของฟลิบฟลอบ D
- วงจรนับขนาด 4 บิต ซึ่งโดยปกติ จะนับ 0000 ถึง 1111
- กรณีนี้อินพุตของเกต NAND รับ D,C และ B
- แสดงว่าวงจรนับ จะนับย้อนกลับเป็น 0000 ทันทีเมื่อการนับถึงสภาวะ 1110 (เลขฐานสิบคือค่าเป็น 14)
- วงจรนับนี้มีสภาวะการนับ 14 สภาวะ (จาก 0000 ถึง 1101) จึงเป็นวงจรนับ MOD - 14
- ความถี่สัญญาณคล็อกของอินพุต 30 KHz ความถี่รูปคลื่นเอาต์พุตจะเป็น 30 kHz / 14= 2 .14 kHz ที่เอาต์พุต D
- การประยุกต์ใช้งานในการหารค่า 26

# วงจรนับที่ไม่ใช่ BINARY

**วงจรนับ** 3 (MODULUS 3 COUNTER)

เช่น วงจรนับที่มี 2 บิต ค่าการนับ จะเริ่มจาก 0, 1, 2 และวนกลับมา 0 ใหม่

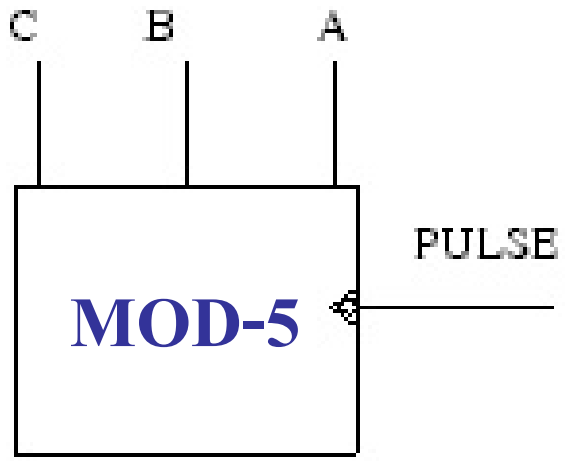



| ผลการนับ | B | A |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 0 | 0 | 0 |

**วงจรนับ 3 และ ผลการนับ**

# วงจรนับ 5 (MODULUS 5 COUNTER )

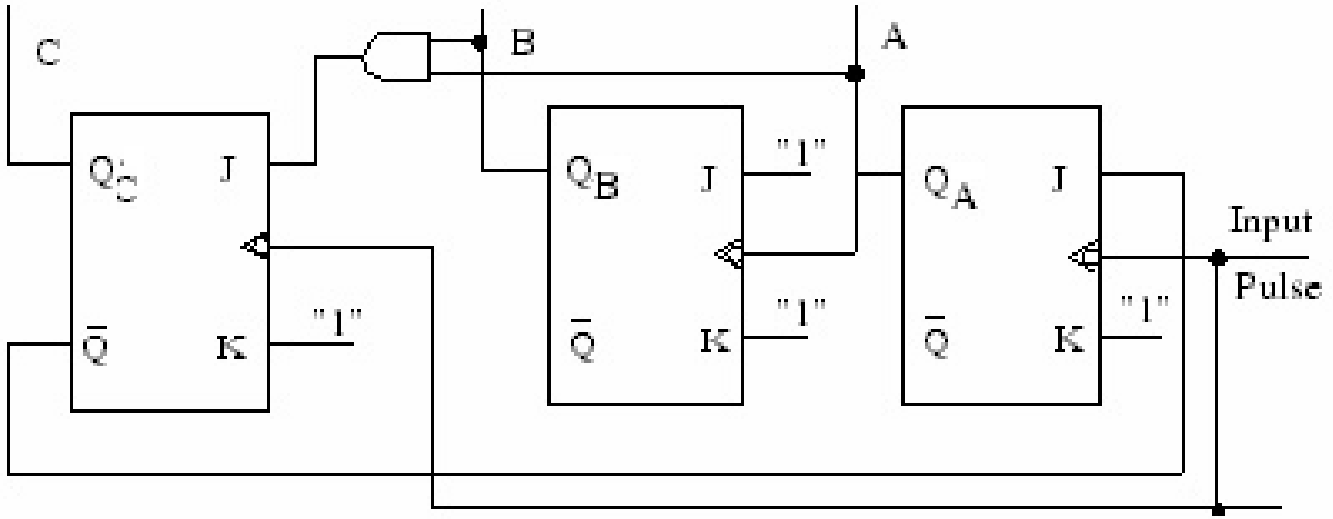
เป็นวงจรนับ 3 บิต ผลการนับเป็นดังนี้คือ 0, 1, 2, 3, 4 ดังแสดงในตารางเอาต์พุต C B A




| ผลการนับ | C | B | A |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 |
| 3 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 |

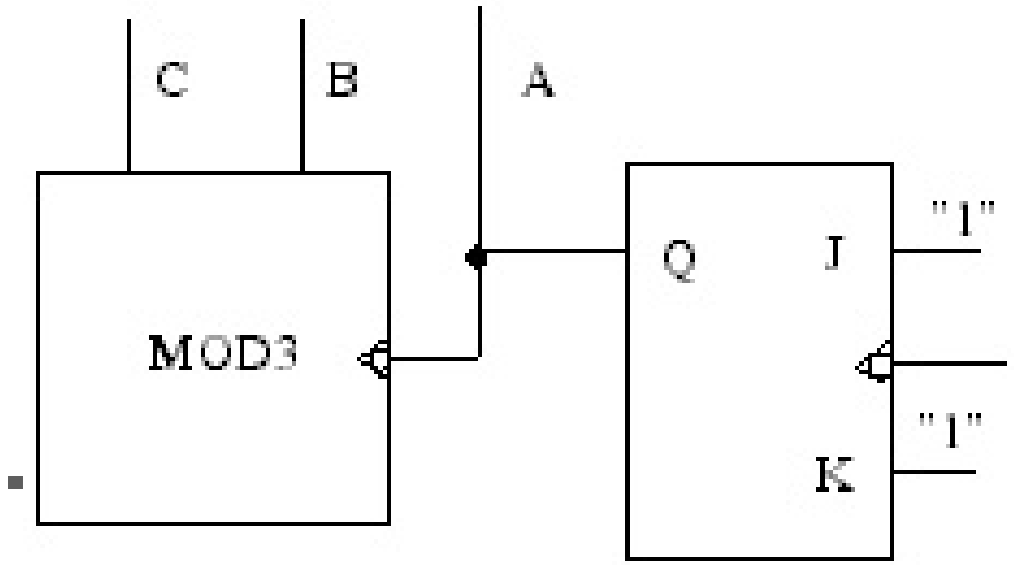
**ผังวงจรนับ 5 และ ผลการนับ**

# วงจรนับค่า 0-4 (MOD-5)




วงจรนับ 5

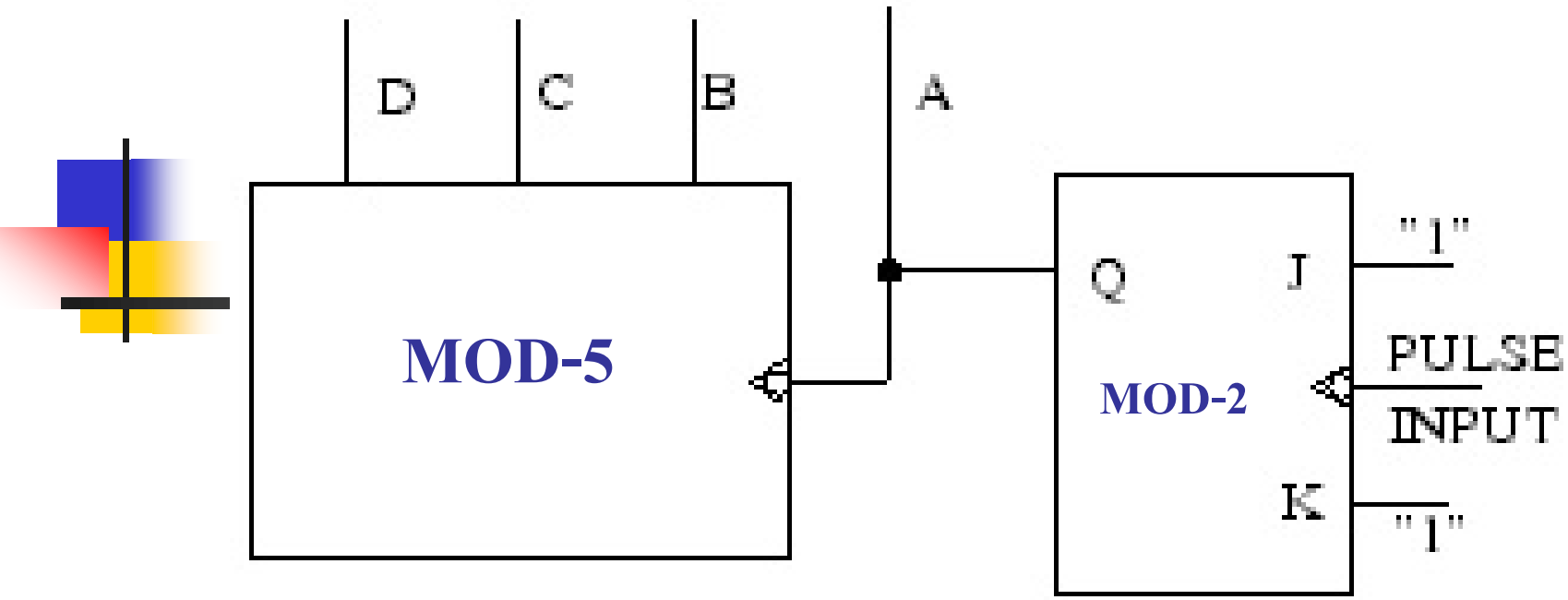
# วงจรนับ 6 และนับ 10




วงจรนับ 6

**วงจรนับ 6 และนับ 10** เป็นวงจรผสมระหว่างวงจรนับไบนารี่ และวงจรนับที่ไม่ใช่ไบนารี่วงจรนับ 6 ได้ผลลัพธ์มาจากการนับ 2 แล้วส่งผลให้ วงจรนับ 3 ส่วนวงจรนับ 10 ได้ผลลัพธ์มาจากวงจรนับ 2 แล้วส่งผลให้วงจรนับ 5 ซึ่งได้แสดงรูปในเฟรมต่อไป

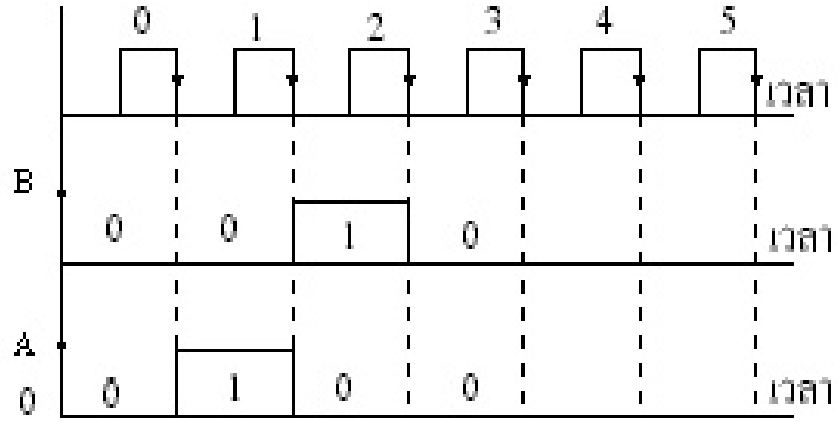
# การนำวงจรนับไปใช้งานมีค่าเท่ากับวงจรหาร 10




วงจรนับ 10

# ผังเวลา (TIMING DIAGRAM วงจรนับ)

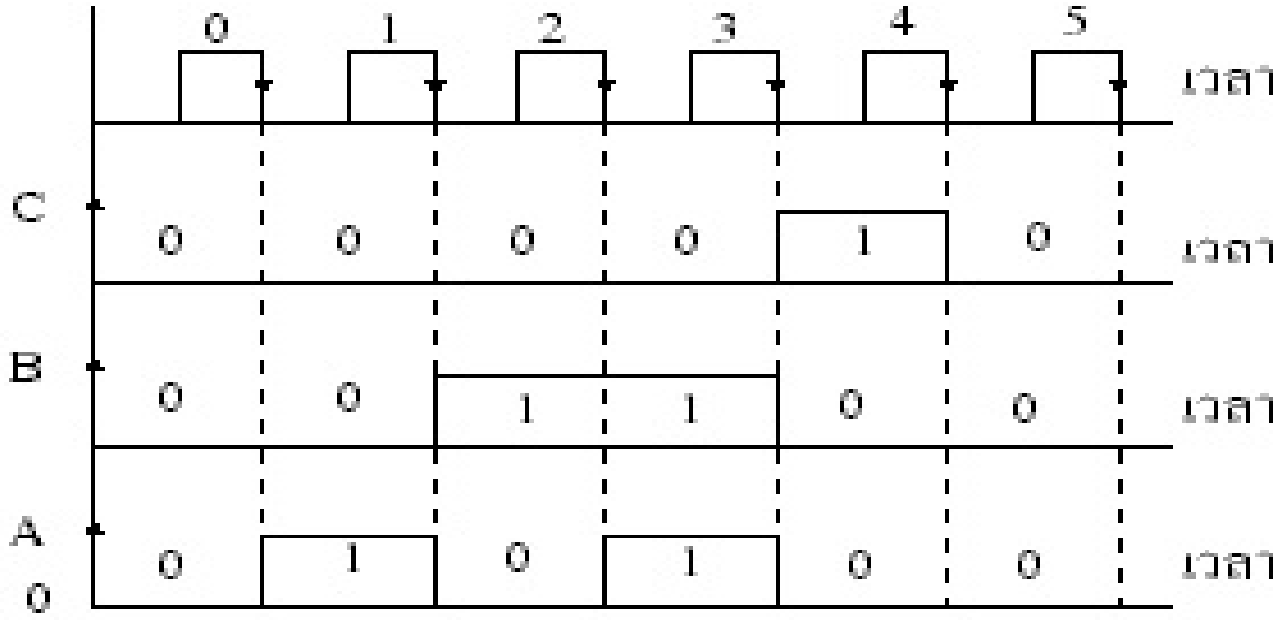
ผังเวลา (TIMING DIAGRAM) เป็นเครื่องมือช่วยอธิบายการทำงานของวงจรดิจิตอลที่เกี่ยวเนื่องกับ เวลาดังตัวอย่างของวงจรนับ 3




| ผลการนับ | B | A |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 0 | 0 | 0 |

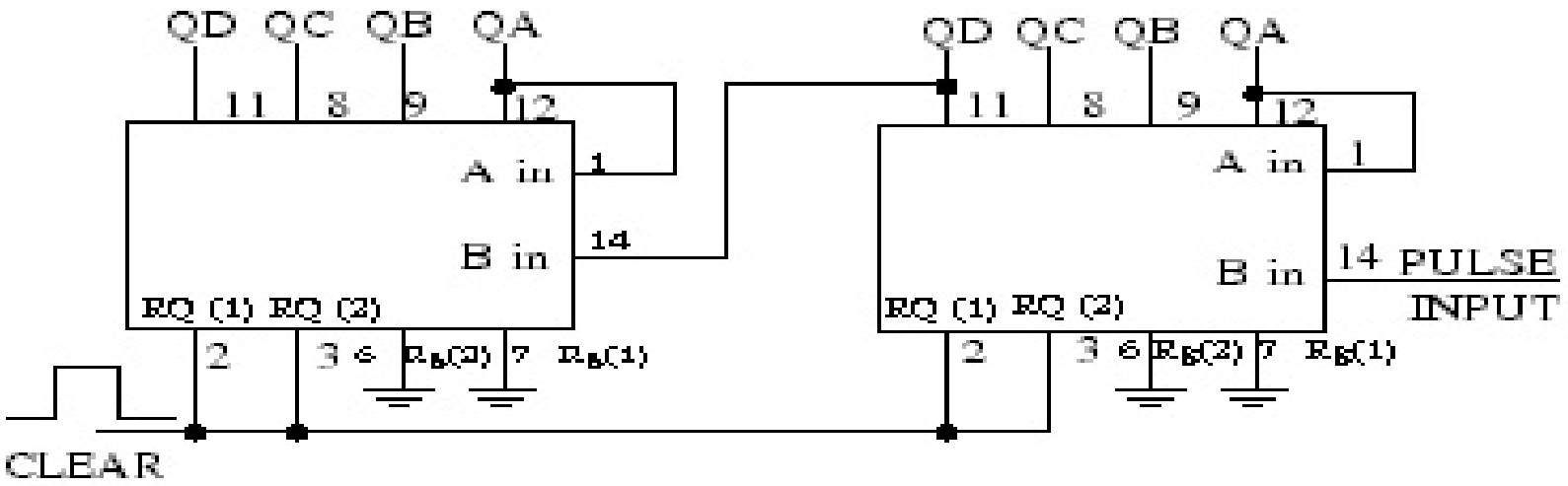
แสดงผังเวลาของวงจรนับ3

# ผังเวลา (TIMING DIAGRAM วงจรนับ)




ผลการนับและผังเวลาวงจรนับ 5

# ไอซีวงจรนับแบบ ASYNCHRONOUS



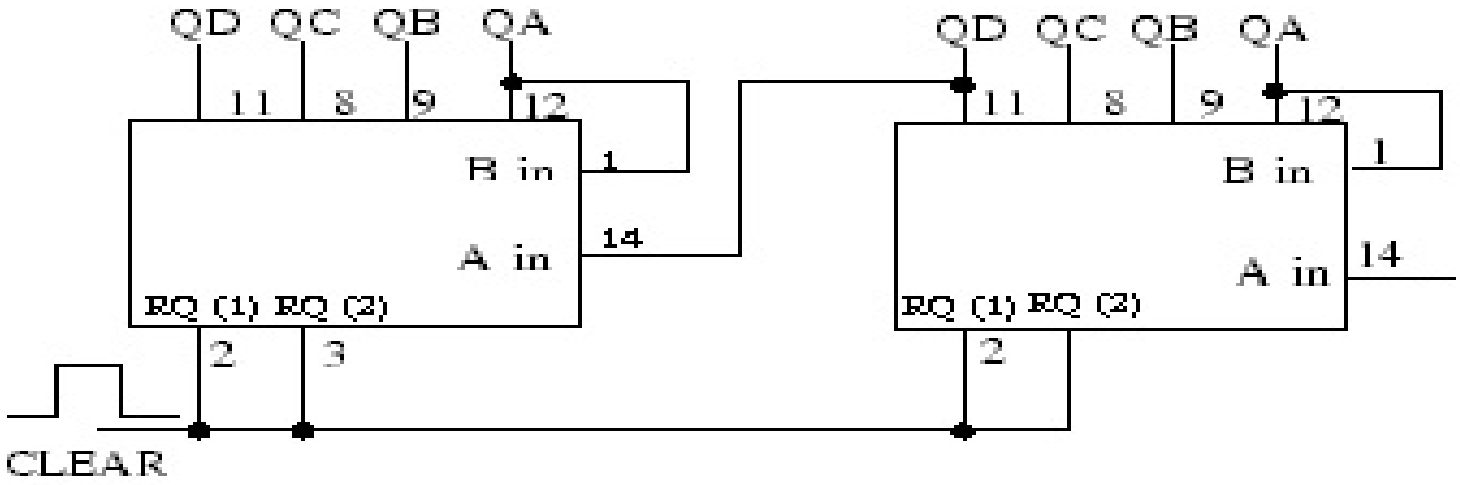

**ขา 5 = + Vcc**

**ขา 10 = Gnd (สำหรับ 7490)**

**แสดงวงจรนับส่งจำนวน 2 หลักใช้ ไอซี TTL เบอร์ 7490**

# การใช้งานวงจรนับ 7493 , 74LS93

# เป็นวงจรนับขนาด Binary 4 บิต

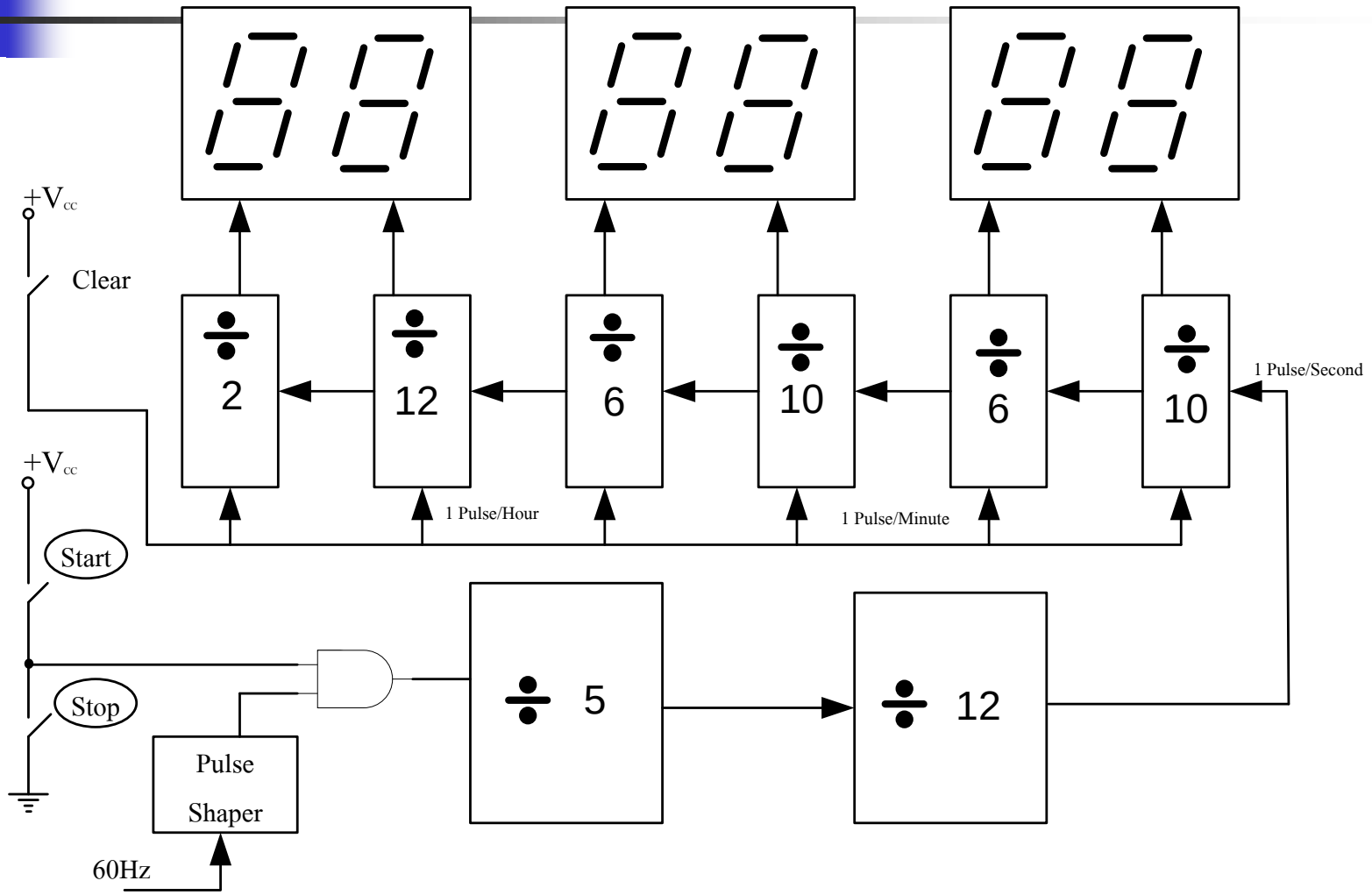



ขา 5 = +Vcc

ขา10 = Gnd (สำหรับ 7493)

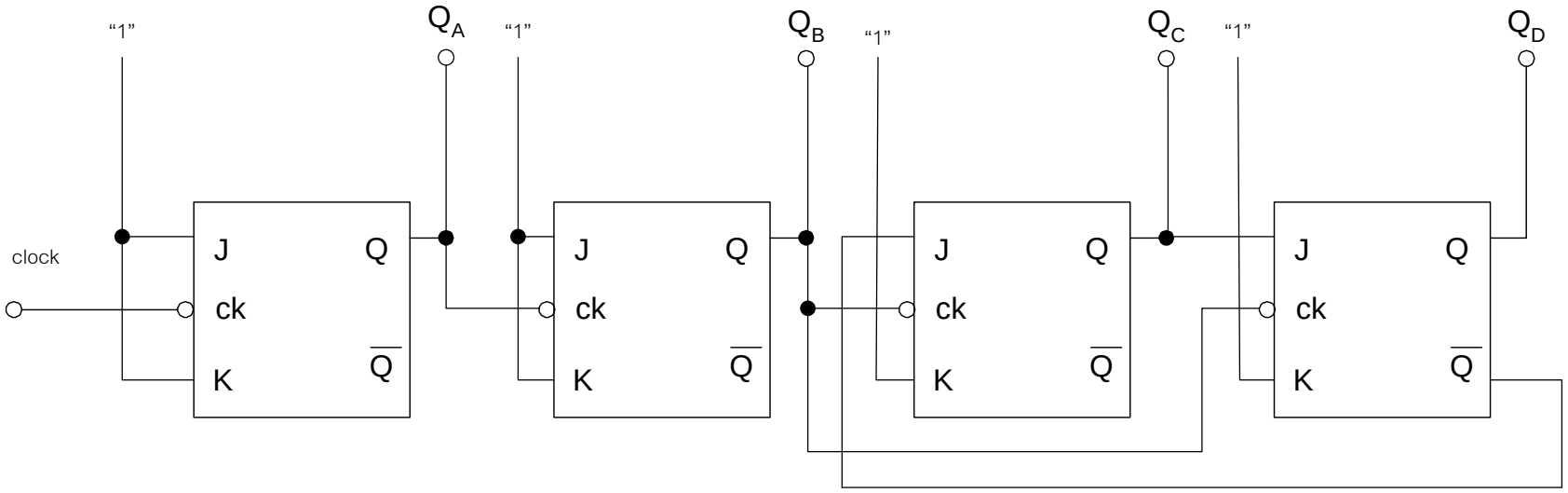
แสดงวงจรนับไบนารี่ 8 บิตใช้ไอซี TTL เบอร์ 7493

# การนำวงจรนับไปประยุกต์ใช้งาน ในรูปแบบนาฬิกาดิจิตอล

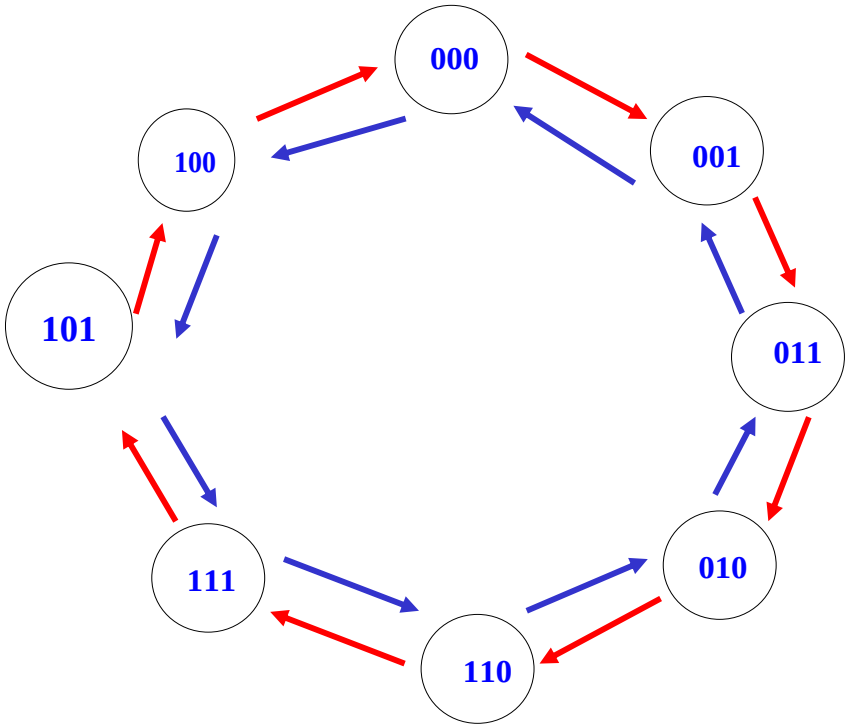


# ทดสอบความรู้สัปดาห์ที่13

1. นักศึกษามีแนวทางประยุกต์วงจรนับที่ได้กล่าวมาทั้งหมดนี้อย่างไร ถึงการนับขึ้น/ลง กระโดด จงแสดงความคิดพร้อมทั้งนำความรู้ที่ได้เรียนมาทั้งหมดทุกสัปดาห์ นำมาออกแบบให้เป็นโมดุล ที่สามารถควบคุมฟังก์ชันการทำงานของวงจรนับ
2. จากรูปวงจรนับจงพิจารณาพร้อมทั้งอธิบายการทำงาน รวมถึงเขียน Timing Diagram และตารางความจริง



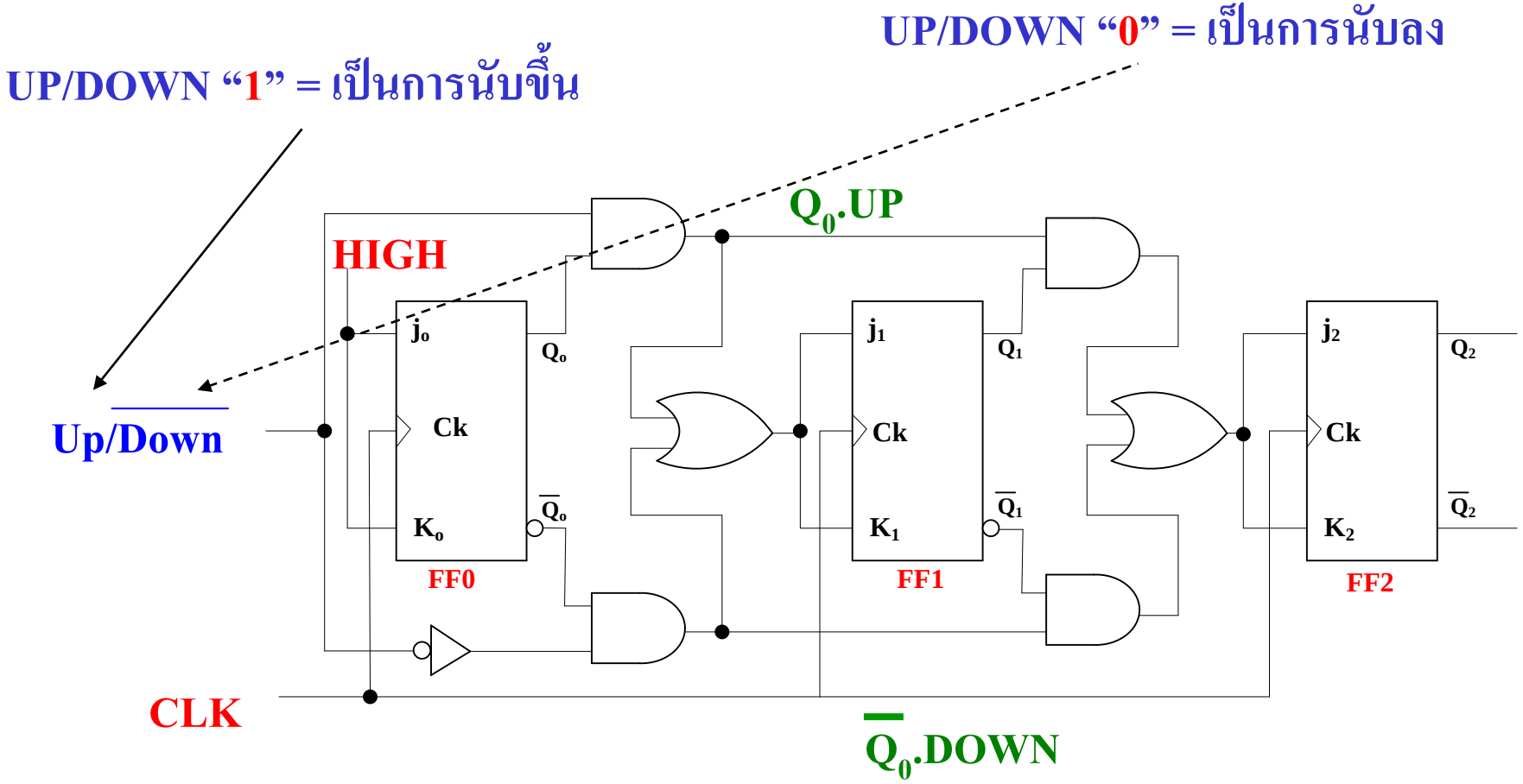
3. จงออกแบบวงจรนับได้จังหวะแบบ 3 บิต โดยสามารถนับขึ้นลงเป็นรหัสเกร์ย เมื่อขา UP/DOWN ลอจิกเป็น “1” นับขึ้นลอจิกเป็น “0” นับลง




**แผนภาพสถานการณ์นับขึ้น นับลง รหัสเกรย์แบบ 3 บิต**

4 . จงวิเคราะห์วงจรนับขึ้น/ลงได้จังหวะแบบ 3 บิต พร้อมเขียน Timing Diagram

$$J_2 = K_2 = (Q_0.Q_1.UP) + (\overline{Q}_0.\overline{Q}_1.DOWN)$$



# เนื้อหาคำบรรยายในสัปดาห์ที่14

## การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล

**71 การแปลงสัญญาณ แอนะลอกเป็นดิจิตอล และ ดิจิตอลเป็นแอนะลอก**

711 พื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอล

712 การชักตัวอย่าง (Sampling)

713 การควอนไตซ์

**72 หลักการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก**

721 Resolution

722 Percentage Resolution

723 การหาค่าแรงดันเอาต์พุต

724 การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกโดยใช้วงจรแบบ R-2R Ladder

# เนื้อหาคำบรรยายในสัปดาห์ที่14

## การแปลงสัญญาณแอนะลอกกับสัญญาณดิจิตอล (ต่อ)

**73 หลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล**

731 Analog-to-Digital Conversion

732 Digital –Ramp ADC

733 Successive- Approximation ADC

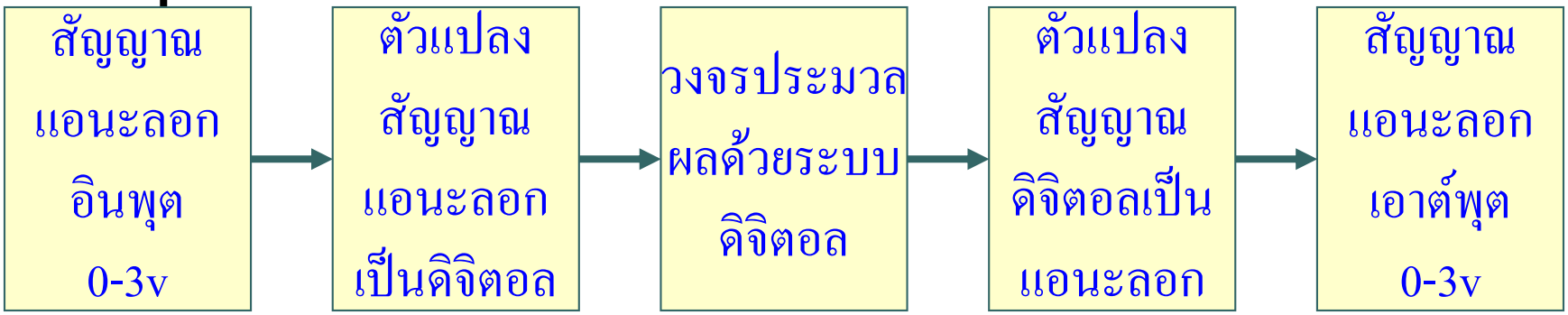
734 การออกแบบวงจรแบบ Flash ADCs

**74 การประยุกต์ใช้งานจากวงจรADC กับ DAC**

741 การนำไอซีเบอร์ DAC 0808 ไปใช้งานแปลงค่าดิจิตอลเป็นแอนะลอก

742 การนำระบบดิจิตอลไปใช้งานในเครื่องมือวัดแบบตัวเลข

# 71 การแปลงสัญญาณแอนะลอกเป็นดิจิตอลและดิจิตอลเป็นแอนะลอก




**บล็อกแสดงการใช้งานแอนะลอกและดิจิตอล**

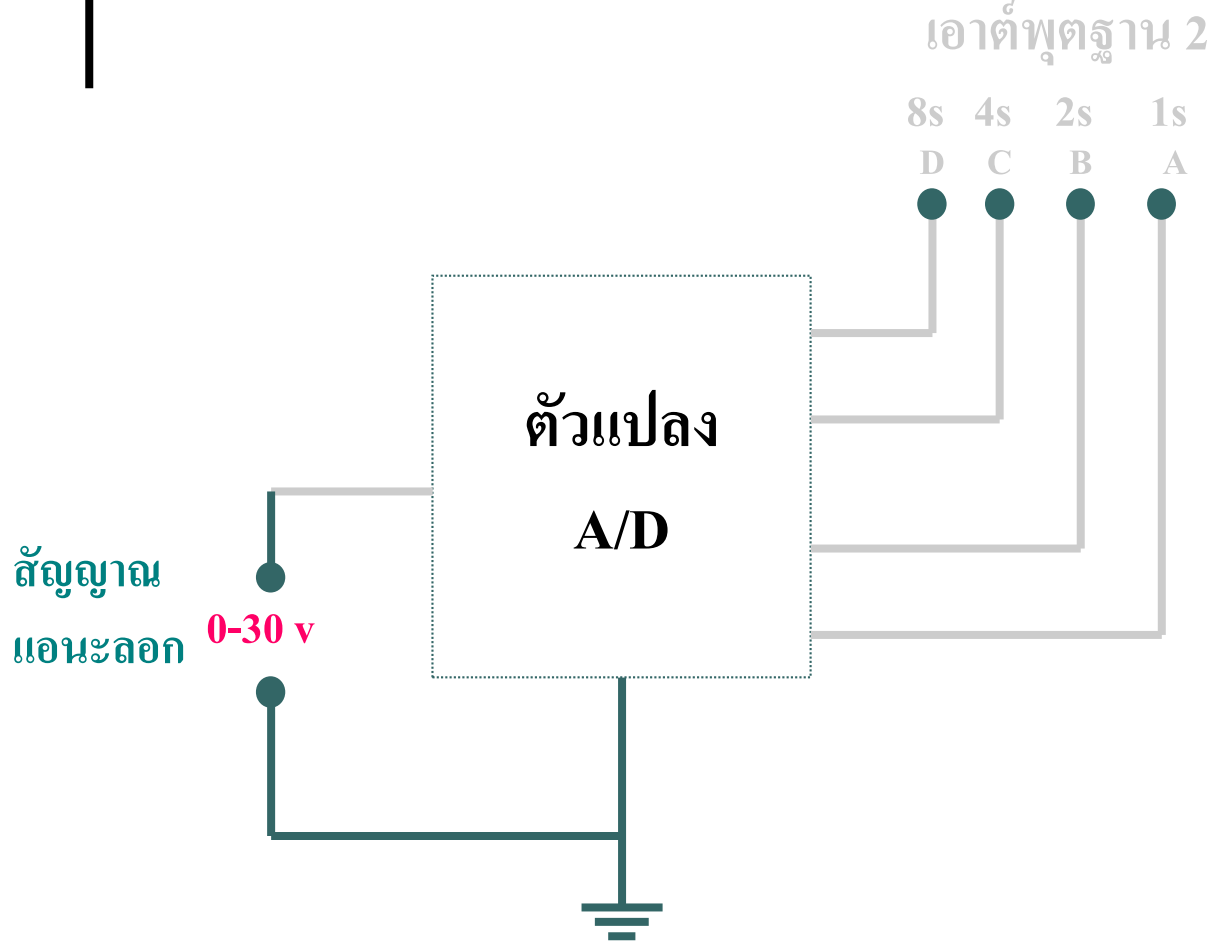
- สามารถนำอุปกรณ์อิเล็กทรอนิกส์ที่สร้างจากสัญญาณแอนะลอกทำงานร่วมกับระบบประมวลผลทางดิจิตอลให้ทำงานร่วมกันได้
- สามารถนำข้อมูลที่ได้ไปประมวลผลด้วยคอมพิวเตอร์และ เก็บข้อมูลที่ได้ในรูปแบบของความจำ
- การควบคุมในเชิงดิจิตอลมีข้อได้เปรียบกว่าทางแอนะลอก

# อธิบาย

การแปลงดิจิตอลเป็นแอนะลอก และแปลงแอนะลอกเป็นดิจิตอลนั้นจะใช้ สำหรับการเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์ที่เป็นดิจิตอลและแอนะลอกเพื่อให้ใช้สื่อสาร

การเชื่อมโยง และเพื่อใช้งานร่วมกันได้ซึ่งอุปกรณ์ที่ใช้ในการแปลงสัญญาณได้แก่ตัวต้านทานออปแอมป์ต่างๆ และมีรูปแบบ การแปลงอยู่หลายวิธีด้วยกันซึ่งขึ้นอยู่กับผู้ใช้ว่าจะเลือกใช้งานแบบไหนให้สะดวกถูกต้องและมีความละเอียดในการแปลงสัญญาณมากที่สุด ดังจะได้อธิบายตามลำดับต่อไป

# 711 พื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอล




วงจรการแปลงสัญญาณแอนะลอกเป็นดิจิตอล

# หลักการพื้นฐานการแปลงสัญญาณแอนะลอกเป็นดิจิตอล

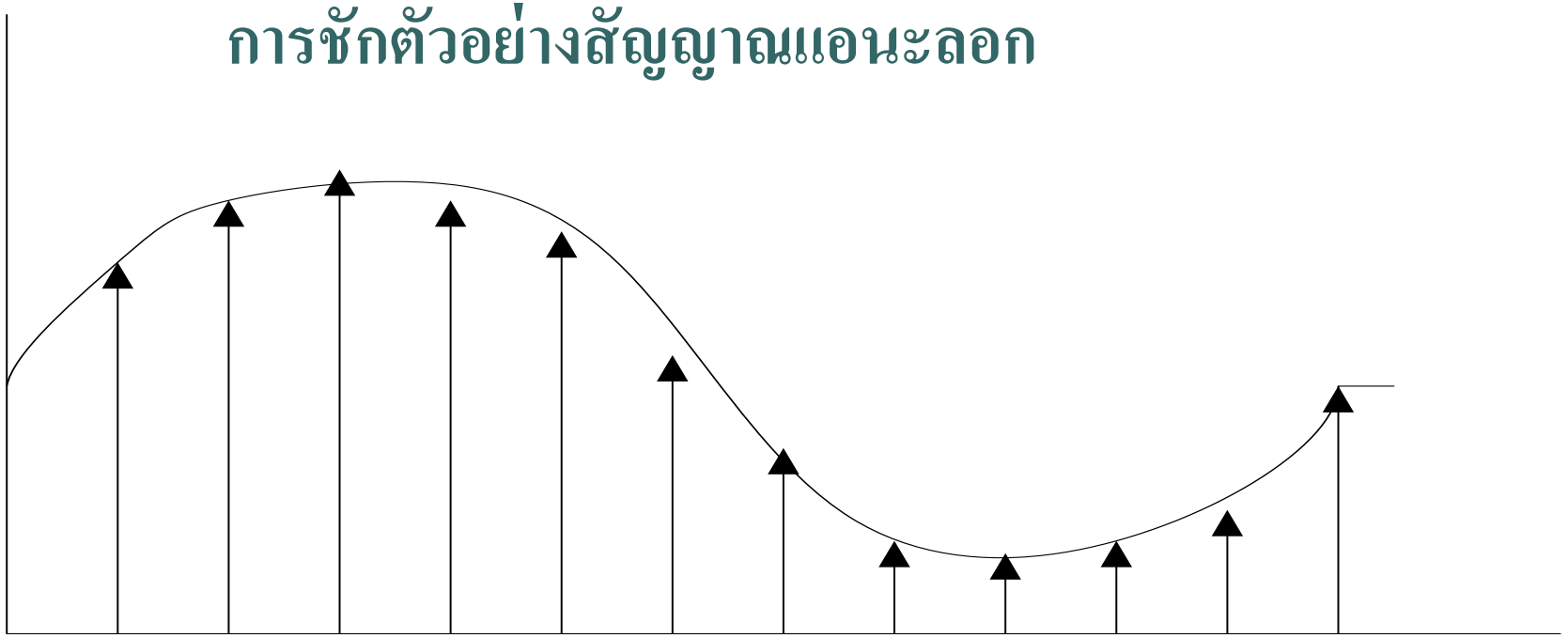
การแปลงสัญญาณแอนะลอก เป็นสัญญาณดิจิตอลประกอบด้วยวงจรเทียบแรงดัน ที่เป็นตัวสำคัญสำหรับการแปลงสัญญาณแอนะลอกเป็นดิจิตอล

จากรูปถ้าอินพุต A >B เอาต์พุตจะเป็น1 และถ้าอินพุต A<B เอาต์พุตจะเป็น0

**บล็อกแสดงการเปรียบเทียบแรงดัน**

# 712 การชักตัวอย่าง( Sampling)

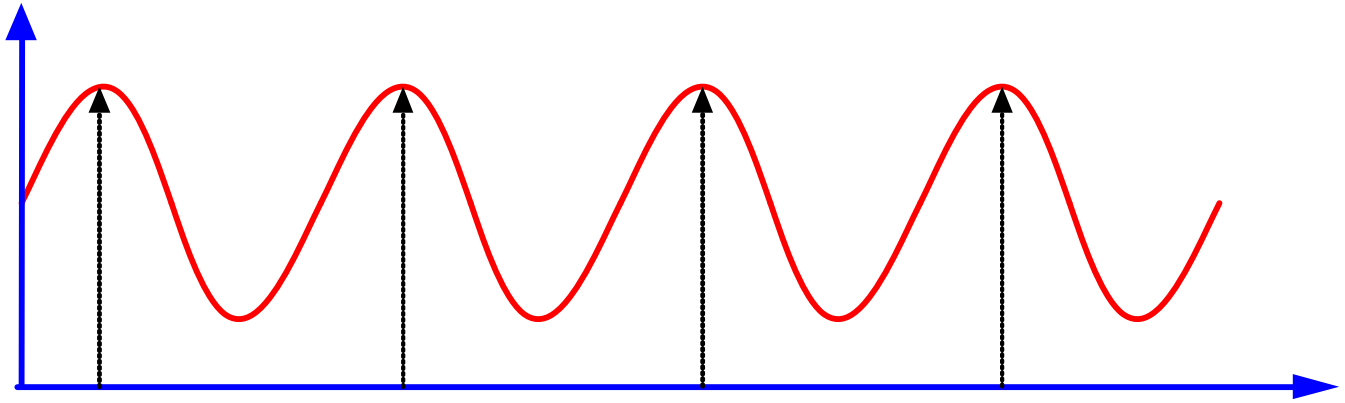
## การชักตัวอย่างสัญญาณแอนะลอก



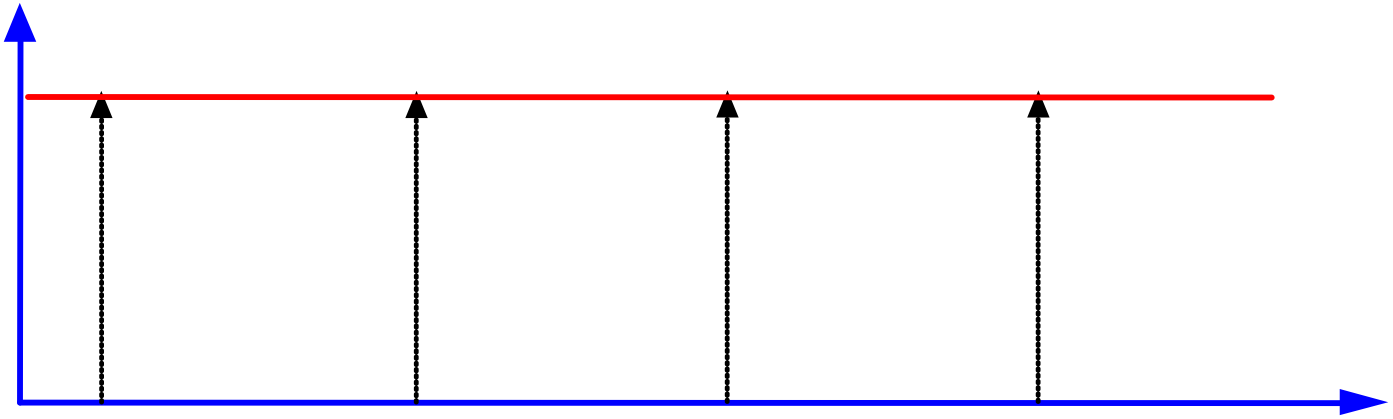
การชักตัวอย่างสัญญาณแอนะลอก

การชักตัวอย่างสัญญาณเป็นการเลือกสัญญาณแอนะลอกให้ไม่มีความต่อเนื่องในทางเวลาและเมื่อได้สัญญาณที่ชักตัวอย่างมาแล้วจากนั้นก็มาทำการควอนไตซ์

## ความถี่ของการชักตัวอย่างมีค่าเท่ากับความถี่ของสัญญาณอินพุต

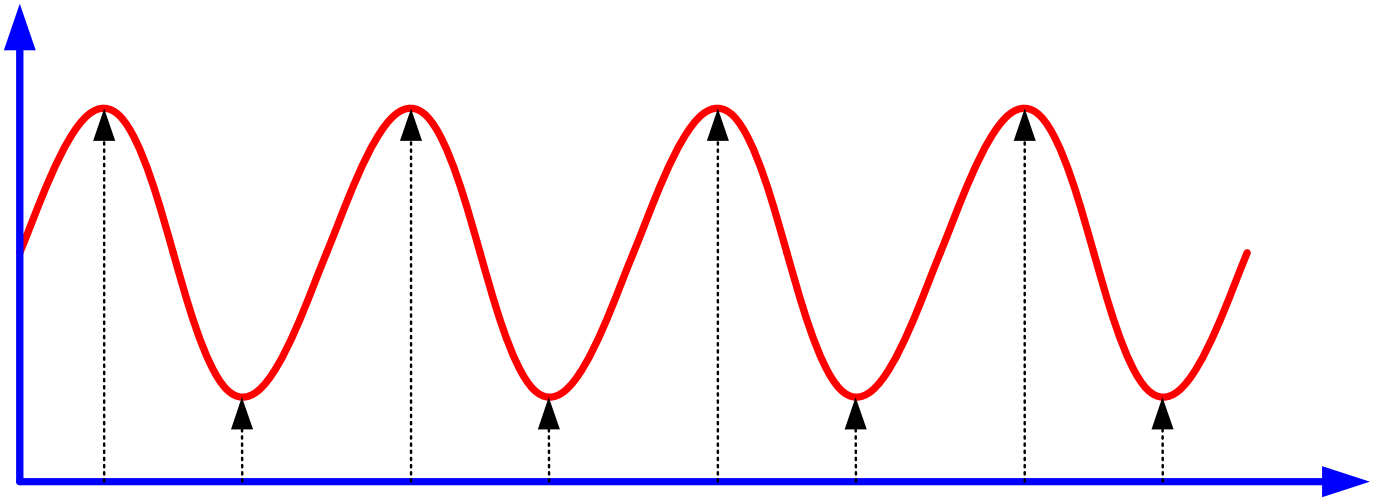


การชักตัวอย่างความถี่เท่ากับสัญญาณไซน์

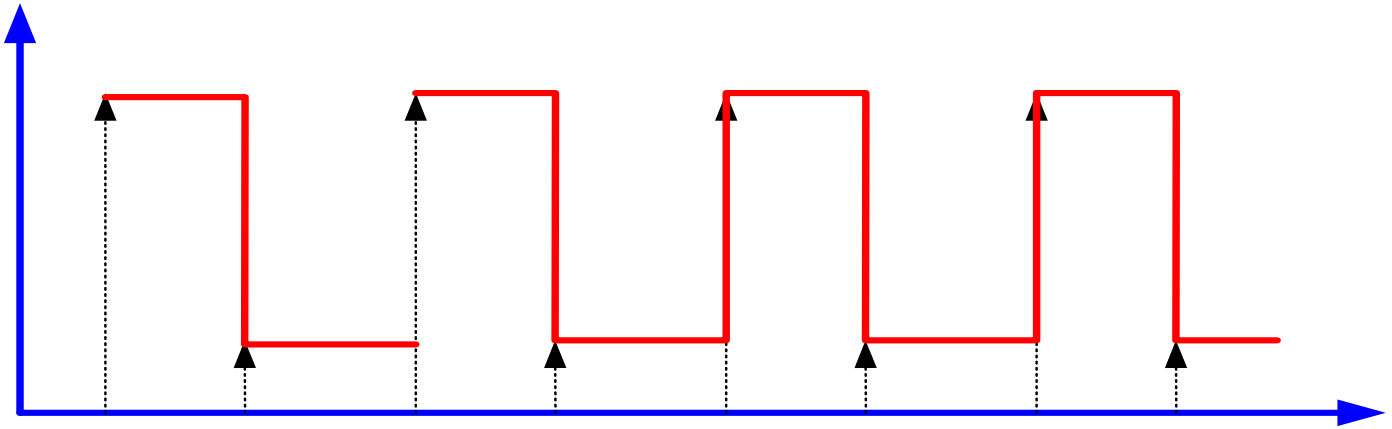


เมื่อแปลงข้อมูลกลับจะได้เป็นไฟDC

# ความถี่ของการ ชักตัวอย่าง มีค่า2เท่าของความถี่สัญญาณอินพุต

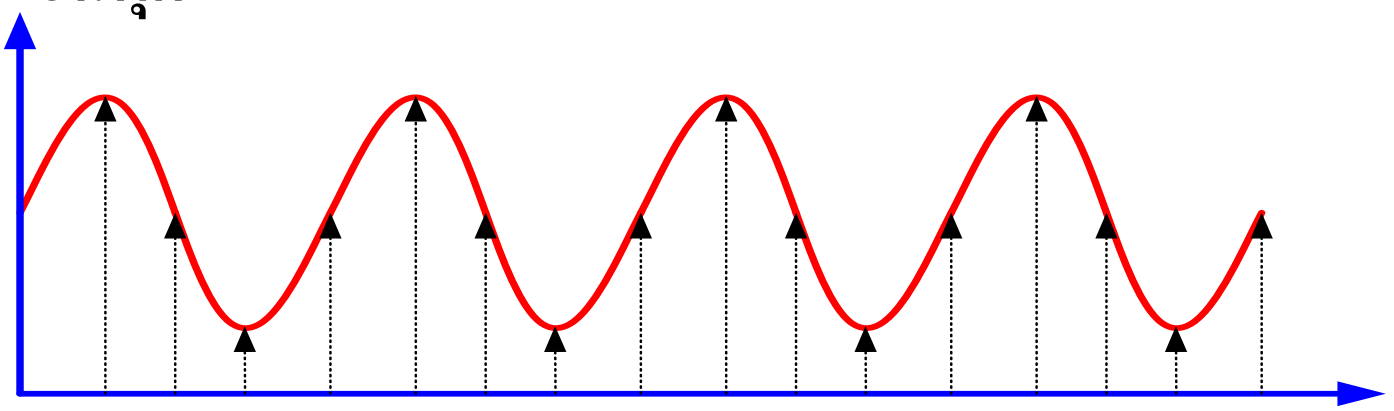


ความถี่การชักตัวอย่างสองเท่าของสัญญาณไซน์

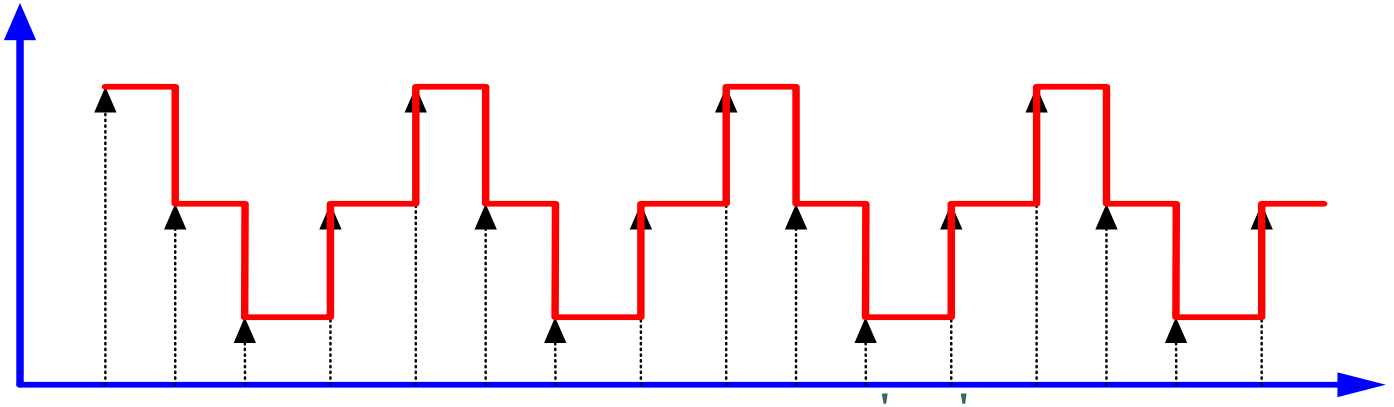


ผลจากการแปลงกลับเป็นรูปสี่เหลี่ยม

ความถี่ของการ ชักตัวอย่าง มีค่า3เท่าของความถี่สัญญาณอินพุต

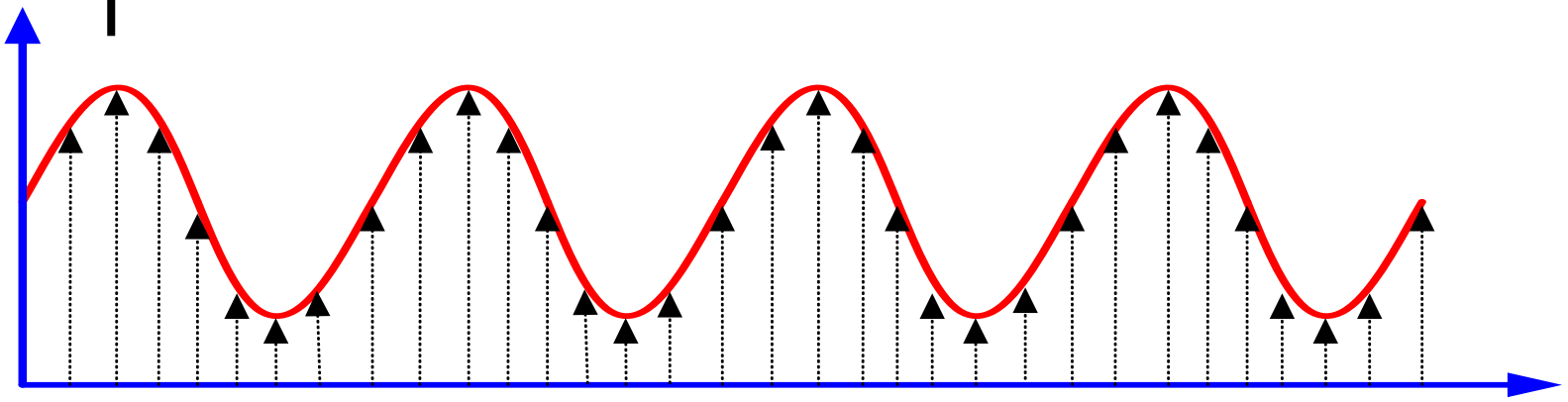


ความถี่การชักตัวอย่าง สี่เท่าของสัญญาณไซน์

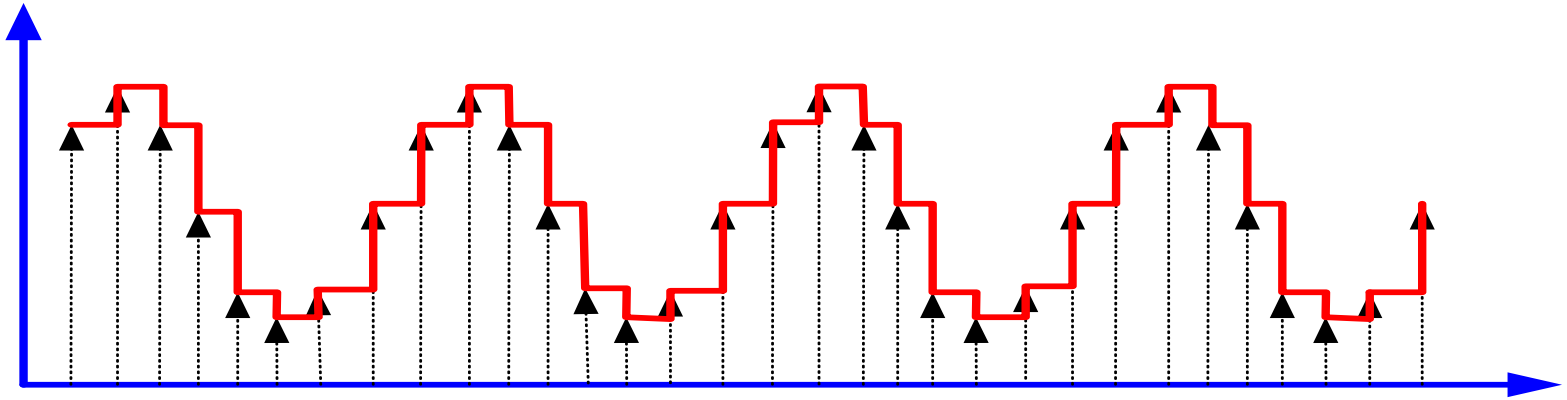


ผลจากการแปลงกลับเป็นรูปสี่เหลี่ยม

ความถี่ของการ ชักตัวอย่าง มีค่า6เท่าของความถี่สัญญาณอินพุต



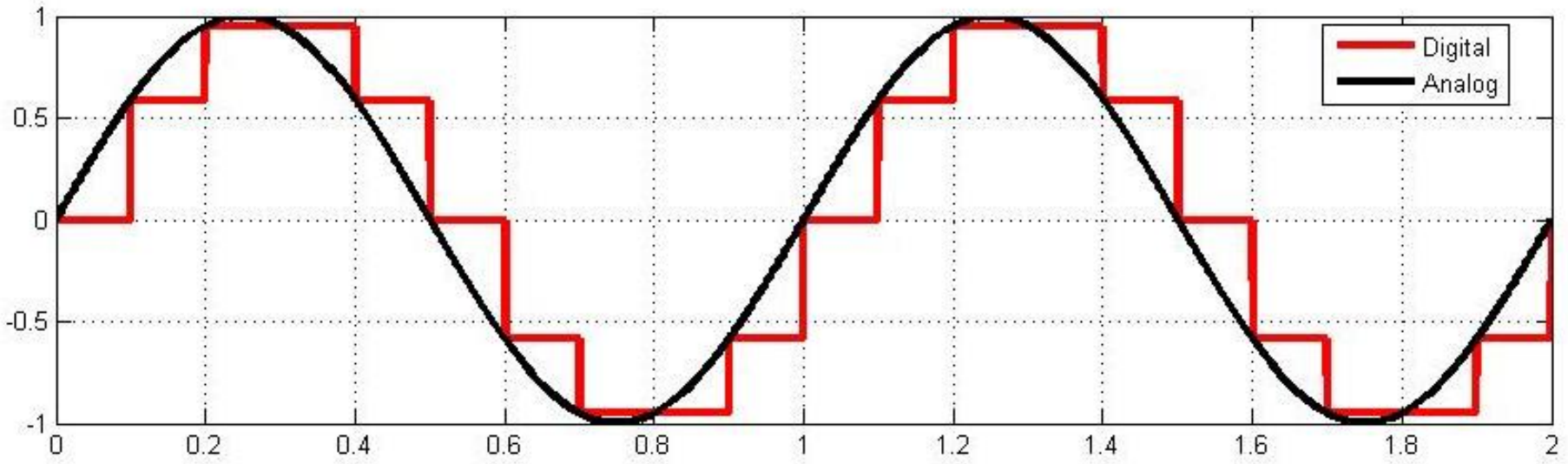
ความถี่การชักตัวอย่าง แปดเท่าของสัญญาณไซน์



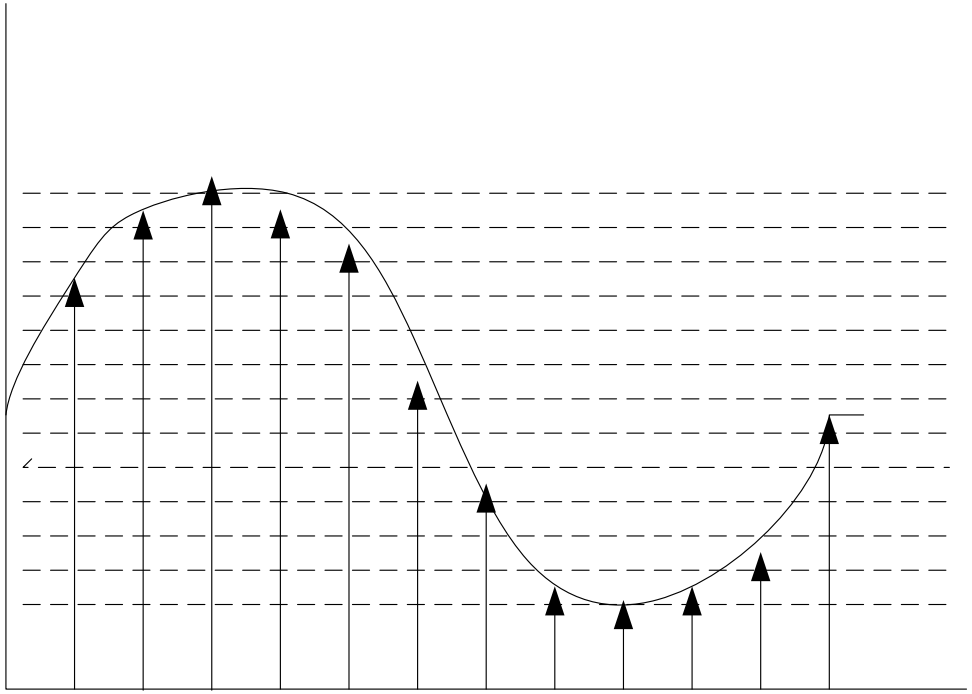
ผลจากการแปลงกลับเป็นรูปสี่เหลี่ยม

ความถี่ของการชักตัวอย่าง โดย 1คาบใช้เวลา 1s “T”

และSampling Time “Ts” = 0.1s

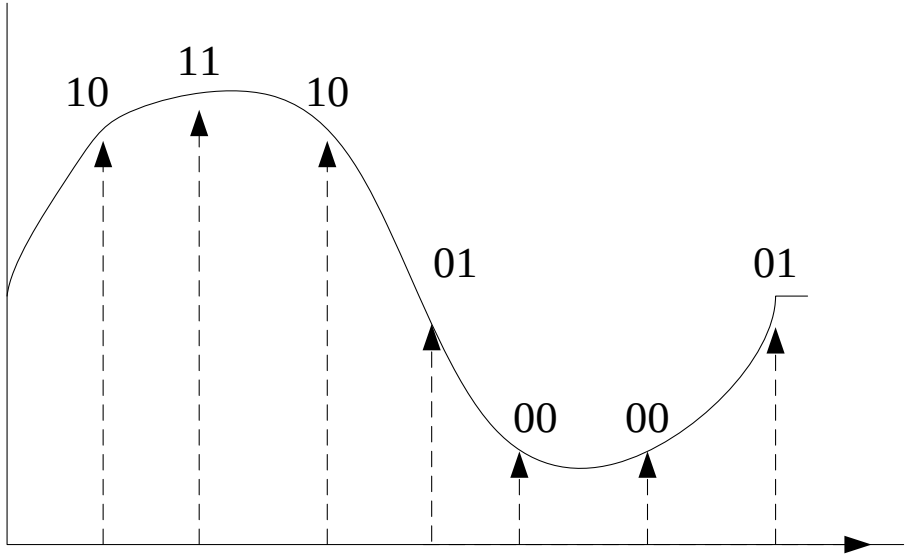


# 713 การควอนไท



การควอนไท

- การควอนไตซ์เป็นการนำสัญญาณที่ผ่านการชักตัวอย่างแต่ละช่วงเวลามาแทนด้วยข้อมูลเลขฐานที่ต่างกัน จากนั้นนำข้อมูลที่ได้ไปเก็บในหน่วยความจำ




การควอนไท

นำข้อมูลที่ได้จาก
การควอนไตซ์ไปเก็บใน
หน่วยความจำ

หน่วยความจำ

| |
|---|
| 10 |
| 11 |
| 10 |
| 01 |
| 00 |
| 00 |
| 01 |

# การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

## สิ่งที่ต้องคำนึงในการแปลงสัญญาณA/D

- อัตราการชักตัวอย่างหรือการเปรียบเทียบข้อมูลถ้าใช้ความถี่ในการชักตัวอย่างสูงจะได้สัญญาณที่แปลงกลับมาใกล้เคียงสัญญาณเดิมมากความละเอียดในการควอนไท

- การควอนไทยิ่งมากเท่าไรจะมีความถูกต้องสูงแต่ทำให้ต้องใช้บิตมากเพื่อเก็บรายละเอียดในการแทนค่ามากขึ้นหน่วยความจำก็ต้องใช้มากขึ้น

# 72 หลักการแปลงสัญญาณดิจิตอลไปเป็นสัญญาณแอนะลอก

บล็อกการแปลงดิจิตอลเป็นแอนะลอก

จากรูปเป็นการแปลงสัญญาณดิจิตอล 4 บิตให้เป็นแรงดันแอนะลอก

- ถ้ารหัส 0000 ก็จะได้เอาต์พุต 0 V
- ถ้ารหัส 1000 ก็จะได้เอาต์พุต 8 V

| D | C | B | A | Vout (Volts) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | 10 |
| 1 | 0 | 1 | 1 | 11 |
| 1 | 1 | 0 | 0 | 12 |
| 1 | 1 | 0 | 1 | 13 |
| 1 | 1 | 1 | 0 | 14 |
| 1 | 1 | 1 | 1 | 15 |

ตารางการแปลงรหัส 4 บิตเป็นแรงดันแอนะลอกซึ่งการแยกแรงดันห่างกัน 1V

# อธิบาย

Digital –To- Analog Conversion (DAC)

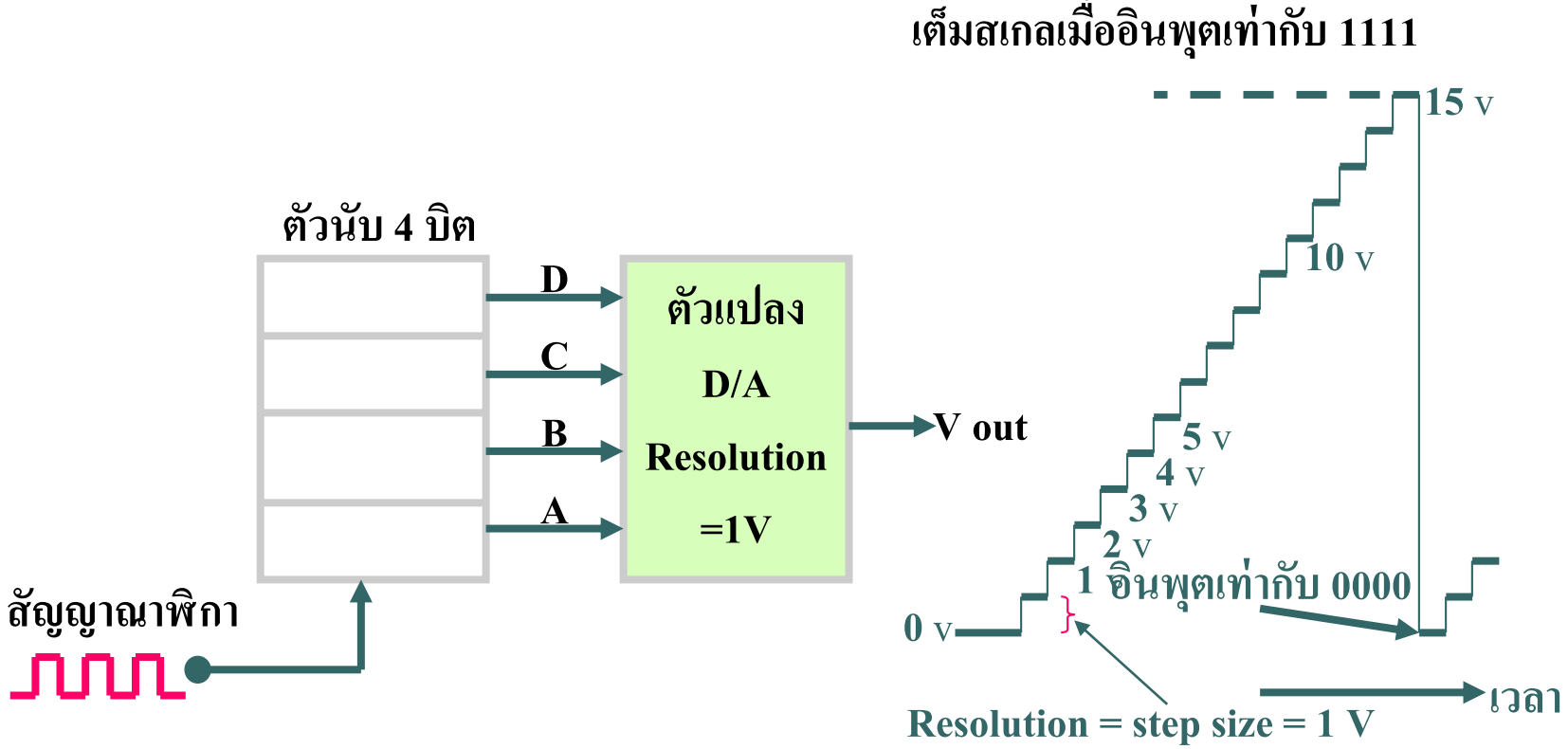
หลักการเบื้องต้นการเปลี่ยนค่าสัญญาณ แอนะลอกเป็น Digital คือ

- การแทนค่าระดับสัญญาณทาง แอนะลอกเป็นค่ารหัสทางดิจิตอล(BCD)
- ค่าที่เกิดขึ้นจะมีความสัมพันธ์กันดังตารางแปลงรหัส 4 บิต

การหาค่าแรงดันเอาต์พุต ของ DAC มีค่าเท่ากับ

- Analog Output = K * Digital Input
- K = เป็นค่าคงที่ที่เป็นสัดส่วนแน่นอนของแรงดันสูงสุดทาง Analog หารด้วย ค่ารหัสสูงสุดของรหัสตัวเลขที่คิดในเลขฐาน10
- Analog Output = ค่าใดๆ เมื่อถูกกำหนดให้มีค่าตามค่า Step Size ของวงจรสามารถอยู่ในรูปกระแสไฟฟ้า หรือ แรงดันไฟฟ้า
- Digital Input = ค่ารหัสทางดิจิตอลอยู่ในรูปของ ( BCD )

# 721 Resolution (Step Size)




บล็อกแสดงการแปลงดิจิตอลเป็นแอนะลอก

# อธิบาย

- ค่า Resolution (Step Size) คือ การเปลี่ยนแปลงในขั้นที่เล็กที่สุดที่เกิดขึ้นกับสัญญาแอนะลอกเอาต์พุต เมื่อสัญญาณทางอินพุตที่เป็นดิจิตอลเปลี่ยนแปลง1
- แรงดันทางเอาต์พุตจะเปลี่ยนแต่ละขั้น (Step Size) 1Volt เมื่ออินพุตเปลี่ยนค่าเพิ่มขึ้นทีละขั้น
- สมมุติว่า วงจรนับค่าขนาด 4 บิต DAC จะนับค่าสูงสุดเท่ากับ 16 สภาวะ

  -Resolution 1Volt/Step

  - Counter นับค่าได้เท่ากับ 1011 มีค่าเท่ากับ$(11)_{10}$ = 11 Volt
- ตัวอย่าง การหาค่าแรงดันสูงสุดของค่าทางแอนะลอก ให้นำค่าที่เกิดจากสัญญาณดิจิตอลสูงสุด คูณด้วยค่าการเปลี่ยนแปลงของแต่ละ Step Size

$$\text{Full Scale (Voltage)} = \text{Step Size} * \text{Number (Max)}$$

$$= 1 *(15)_{10}$$

$$= 15 \text{ Volts}$$

# 722 Percentage Resolution

การหาค่าเปอร์เซ็นต์ของ Resolution ของค่าแรงดัน

- เมื่อเอาต์พุตมีค่าสูงสุดเต็มสเกลซึ่งพิจารณาจากตารางเป็น 15 โวลต์
- เมื่ออินพุตของสัญญาณดิจิตอลเป็น 1111 (Step Size = 1 โวลต์)

$$\%Resolution = \frac{Step\ Size}{Full\ Scale} X100$$

$$= \frac{1}{15} X100\% = 6.67\%$$

**ตัวอย่างที่14.1** วงจร DAC มีเอาต์พุตขนาด 10 บิต Step Size = 10 mv จงหาค่า Full Scale และเปอร์เซ็นต์ของ Resolution

- 10 บิตมีค่าเท่ากับเลขฐานสอง $2^{10}-1 = 1023$ ขั้นบันได แต่ละขั้นมีค่าเท่ากับ 10 mV

- Full scale Output =10 mV x 1023 =10.23 Volts

(วิธีที่1) % Resolution = 10/10.23 *100 % = 0.1 %

(วิธีที่2) $$\%Resolution = \frac{1}{Total\ number\ of\ Step} X100\%$$

$$\%Resolution = \frac{1}{2^{10}-1} X100\%$$

- เปอร์เซ็นต์Resolutionมีค่าเท่ากับ 0.1%

# 723 การหาค่าแรงดันทางเอาต์พุต

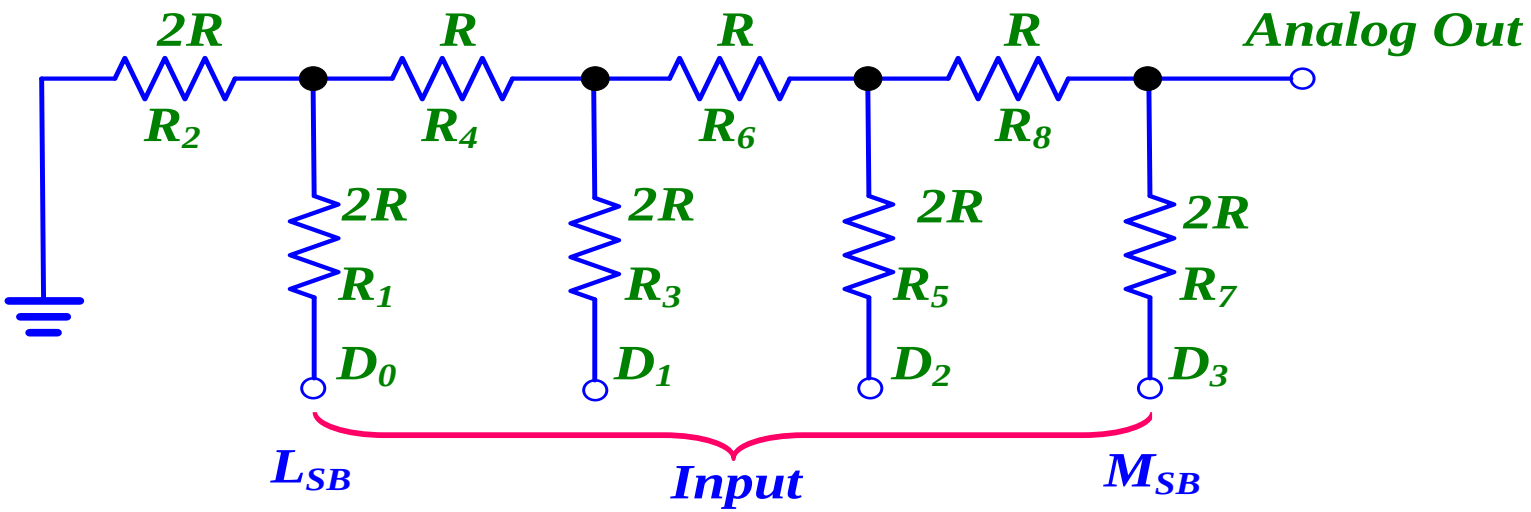
**ตัวอย่างที่14.2** จงหาค่ารงดันที่เอาต์พุตของวงจรDAC ที่สร้างสัญญาณ Step Size มีค่าเท่ากับ 0.2 โวลต์ สัญญาณทางอินพุตเป็นดิจิตอลขนาด5 บิต คือ 10001 $=(17)_{10}$

$$\text{Vout} = K * \text{Digital Input}$$

$$= 0.2 * 17$$

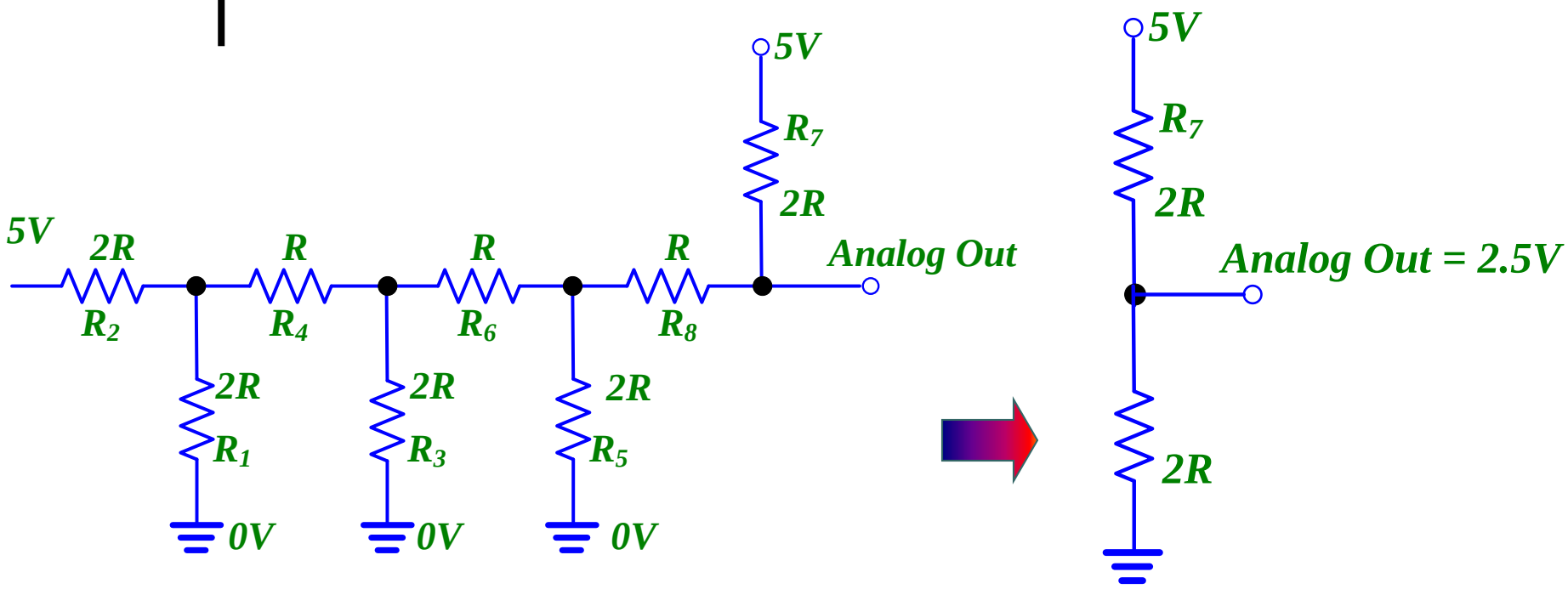
$$= 3.4 \text{ Volts}$$

# 724 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกโดยใช้วงจรแบบ R-2R Ladder



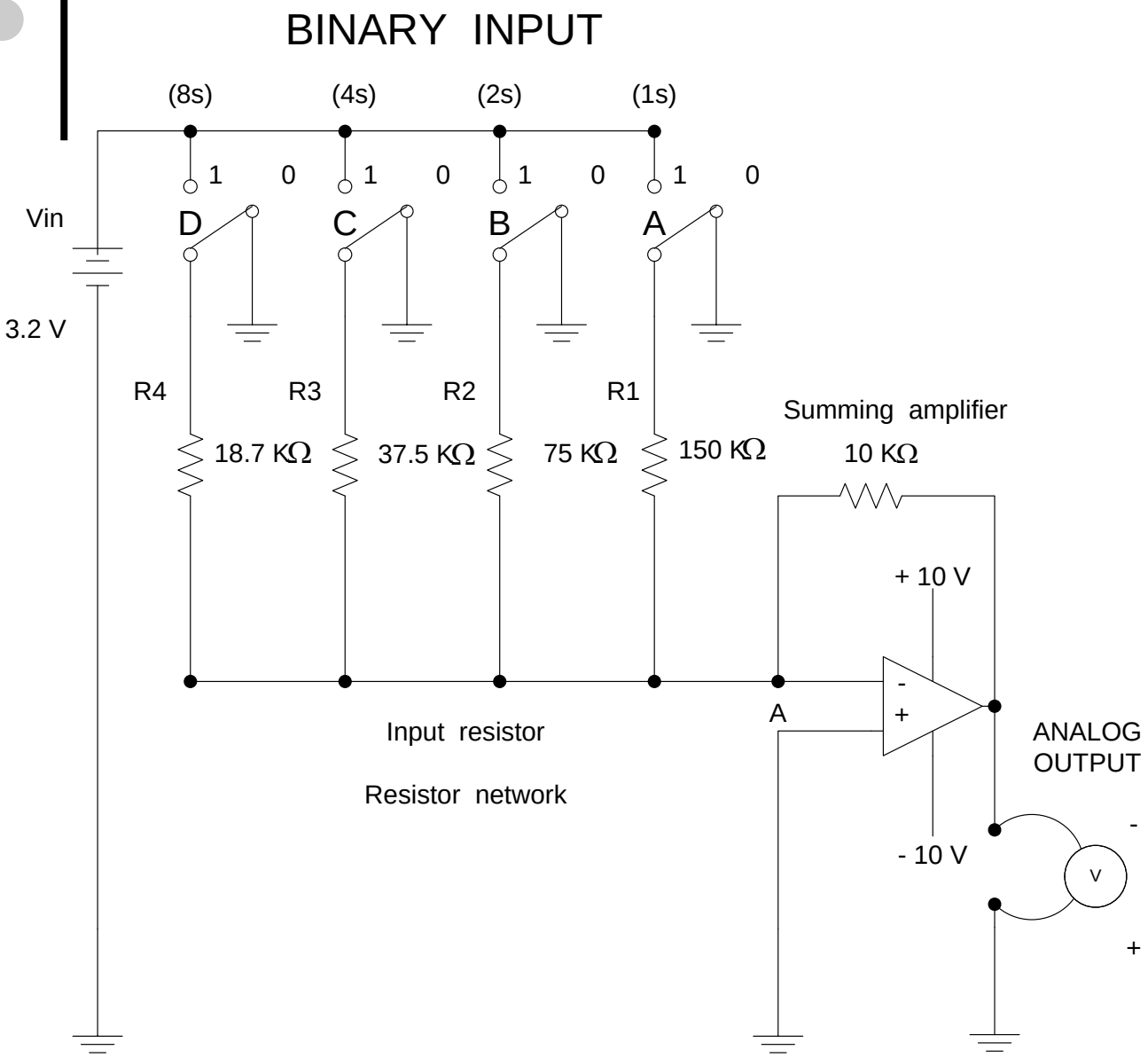

การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกโดยใช้วงจรแบบ R-2R Ladder

# การสร้างสัญญาณแอนะลอกจากรหัสดิจิตอล




**ตัวอย่าง** การแปลงดิจิตอลเป็นแอนะลอกโดยใช้วงจรแบบ R-2R Ladder เมื่ออินพุตเข้ามาเป็น1000 ได้เอาต์พุตเป็น 2.5 V

# วงจร D/A แบบวัดค่าความต้านทานน้ำหนักของรหัสไบนารี่



# ตารางการเปลี่ยนข้อมูลไบนารี่เป็นค่าแรงดันแอนะลอก

| | Digital Input | | | | Analog Output |
|---|---|---|---|---|---|
| | D | C | B | A | Volts |
| ROW 1 | 0 | 0 | 0 | 0 | 0 |
| ROW 2 | 0 | 0 | 0 | 1 | 0.2 |
| ROW 3 | 0 | 0 | 1 | 0 | 0.4 |
| ROW 4 | 0 | 0 | 1 | 1 | 0.6 |
| ROW 5 | 0 | 1 | 0 | 0 | 0.8 |
| ROW 6 | 0 | 1 | 0 | 1 | 1 |
| ROW 7 | 0 | 1 | 1 | 0 | 1.2 |
| ROW 8 | 0 | 1 | 1 | 1 | 1.1 |
| ROW 9 | 1 | 0 | 0 | 0 | 1.6 |
| ROW 10 | 1 | 0 | 0 | 1 | 1.8 |
| ROW 11 | 1 | 0 | 1 | 0 | 2 |
| ROW 12 | 1 | 0 | 1 | 1 | 2.2 |
| ROW 13 | 1 | 1 | 0 | 0 | 2.4 |
| ROW 14 | 1 | 1 | 0 | 1 | 2.6 |
| ROW 15 | 1 | 1 | 1 | 0 | 2.8 |
| ROW 16 | 1 | 1 | 1 | 1 | 3 |

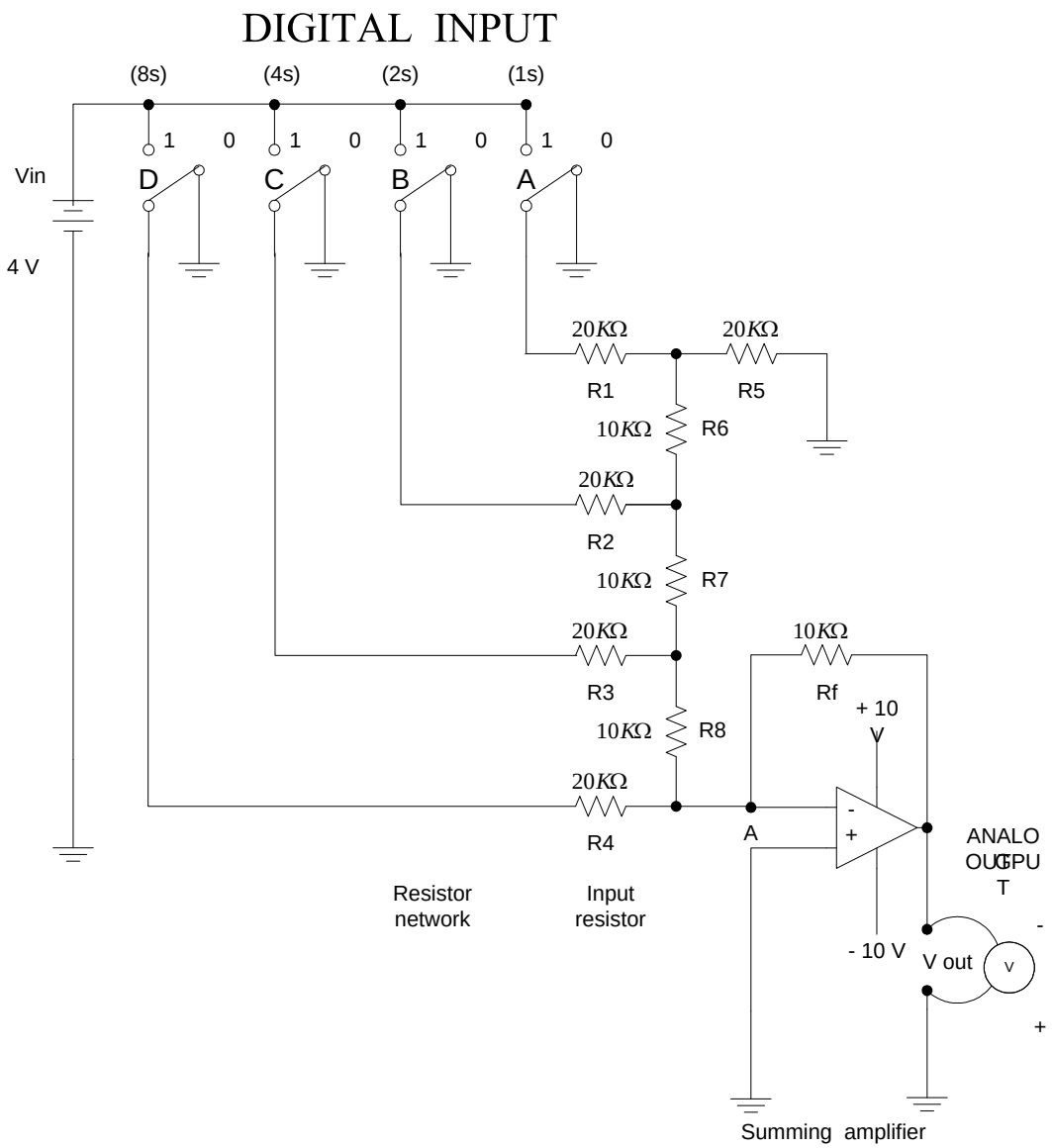
# อธิบาย

- จากรูปวงจรเป็นวงจรแปลงสัญญาณดิจิตอลเป็น แอนะลอก
- ด้วยการใช้วงจรรวมสัญญาณแบบไม่กลับขั้วสัญญาณ
- เกนการขยายสัญญาณจะมีค่าเท่ากับ

$$(10/R_1+10/R_2+10/R_3+10/R_4)$$

- ค่าสัญญาณทางเอาต์พุต จะมีค่าเท่ากับเกนคูณกับแรงดันอินพุต ซึ่งมีค่าคงที่อยู่ที่ 3.2 โวลต์
- พิจารณาจากรูปวงจรแปลงสัญญาณ มีสวิทซ์ควบคุมแต่ละบิต ซึ่งทำหน้าที่กำหนดแรงดันทางทางเอาต์พุต เป็นไปตาราง

# การออกแบบวงจรแลดเดอร์ ( R – 2R ) D/A



# ตัวอย่างหาค่าแรงดันทางเอาต์พุต

(4 Stage Ladder Network ) ของไบนารี่ที่ใช้ระดับแรงดันลอจิก “1” เท่ากับ +16 โวลต์ และแรงดันลอจิก “0” เท่ากับ 0 โวลต์ กำหนดเลขไบนารี่ต่างๆเหล่านี้

(ก) 1010 (ข) 0001 (ค) 1110 (ง) 1000 (จ) 1011

**วิธีทำ** ค่าเอาต์พุตจะได้รับแรงดันที่แตกต่างตามรหัสไบนารี่ $2^4 = 16$

ก) 1010

$$\text{Vout} = \frac{2^3 + 0 + 2^1 + 0}{16} \times 16 = 10\,Volts$$

ข) 0001

$$\text{Vout} = \frac{0 + 0 + 0 + 2^0}{16} \times 16 = 1\,Volts$$

# การคำณวนค่าระดับแรงดันทางเอาต์พุต

ค) 1110

$$\text{Vout} = \frac{(2^3 + 2^2 + 2^1)}{2^4} \times 16 = 14\,Volts$$

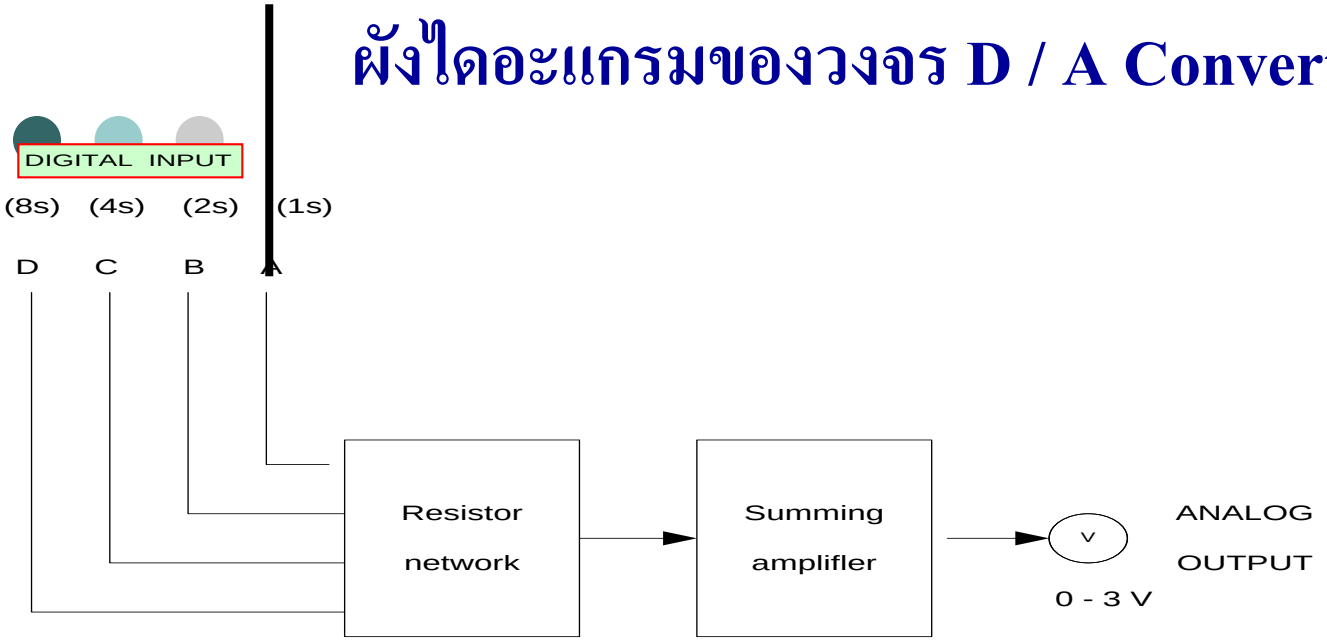
ง) 1000

$$\text{Vout} = \frac{(2^3 + 0 + 0 + 0)}{2^4} \times 16 = 8\,Volts$$

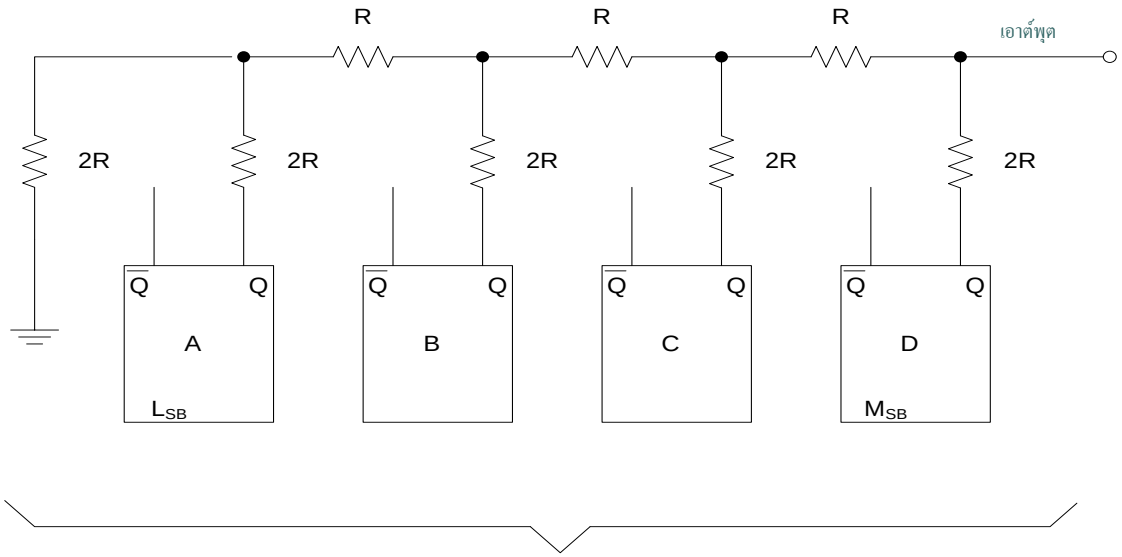
จ) 1011

$$\text{Vout} = \frac{(2^3 + 0 + 2^1 + 2^0)}{2^4} \times 16 = 11\,Volts$$

# ผังไดอะแกรมของวงจร D / A Convertor




## วงจร R – 2R แลดเดอร์



| Binary Input | | | | Analog Output |
|---|---|---|---|---|
| 8s(D) | 4s(C) | 2s(B) | 1s(A) | Volts |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0.25 |
| 0 | 0 | 1 | 0 | 0.5 |
| 0 | 0 | 1 | 1 | 0.75 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1.25 |
| 0 | 1 | 1 | 0 | 1.5 |
| 0 | 1 | 1 | 1 | 1.75 |
| 1 | 0 | 0 | 0 | 2 |
| 1 | 0 | 0 | 1 | 2.25 |
| 1 | 0 | 1 | 0 | 2.5 |
| 1 | 0 | 1 | 1 | 2.75 |
| 1 | 1 | 0 | 0 | 3 |
| 1 | 1 | 0 | 1 | 3.25 |
| 1 | 1 | 1 | 0 | 3.5 |
| 1 | 1 | 1 | 1 | 3.75 |

→ $L_{SB}$

**ตารางค่าแรงดันเอาต์พุตของวงจร R – 2R แลดเดอร์**

→ **Full Scale**

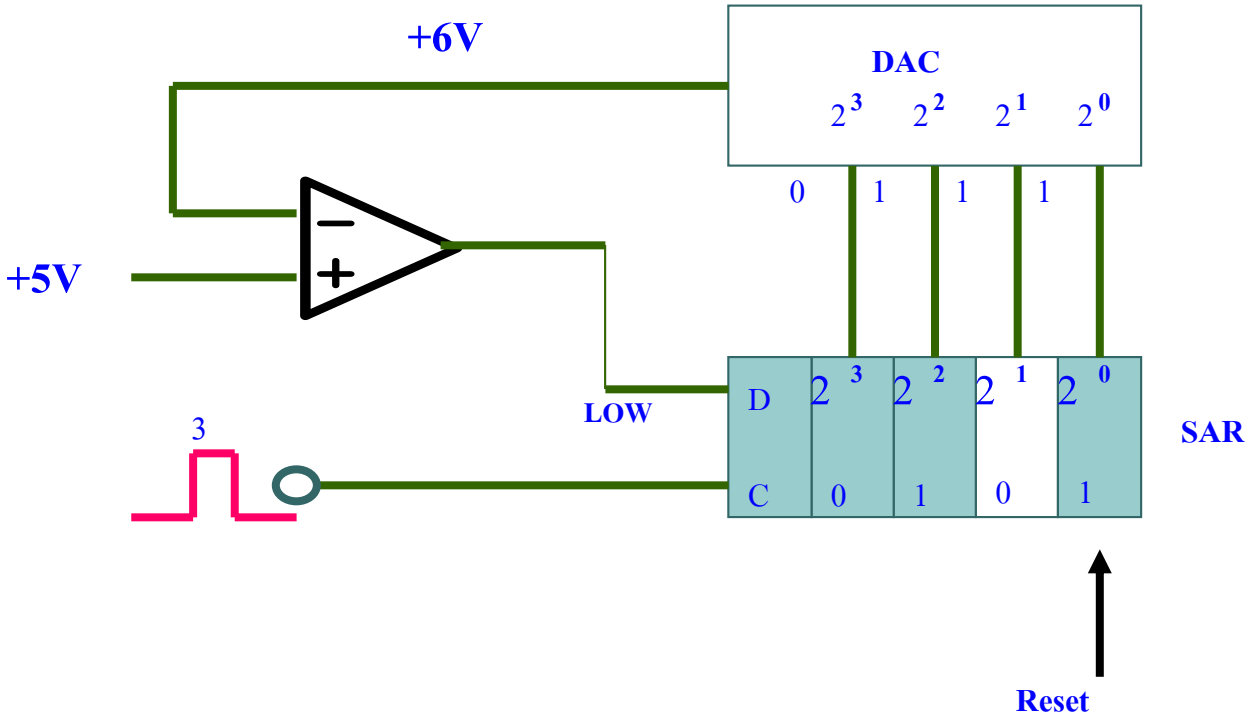
## 73 หลักการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

จากบล็อกไดอะแกรมการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลแบบSuccessive-Approximation (SA)จะมีหลักการทำงานคือ

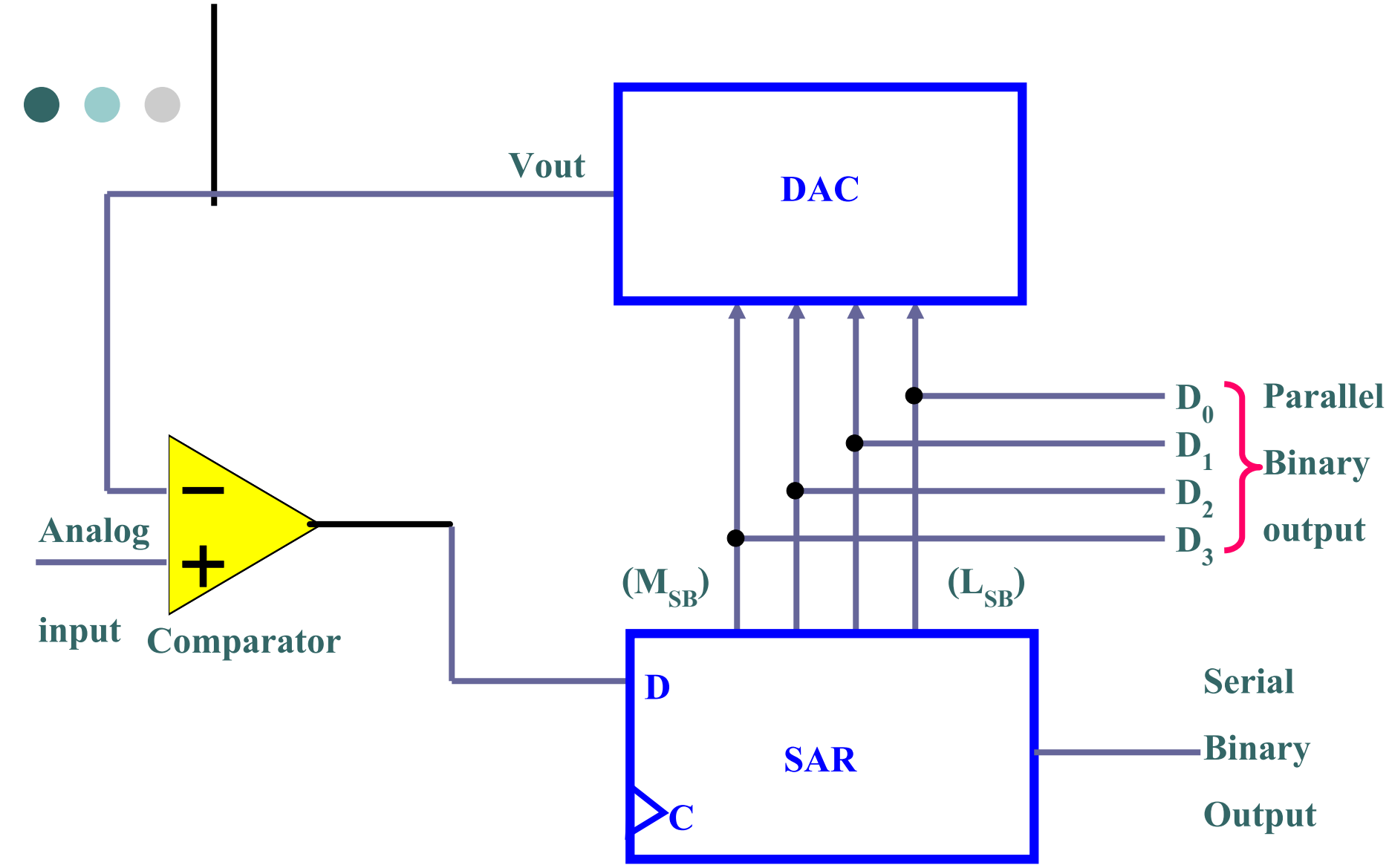
- SAR จะเป็นตัวเปรียบเทียบอินพุตกับเอาต์พุต
- ถ้าอินพุตมากกว่าเอาต์พุตSARจะทำการรีเซทค่าบิตใน $M_{SB}$ ให้เป็น 0 เอาต์พุตจะเป็น Low
- ถ้าอินพุตน้อยกว่าเอาต์พุตSARจะกำหนดให้ค่าใน $M_{SB}$ เป็น1เอาต์พุตจะเป็น High
- การเปรียบเทียบจะเป็นเช่นนี้จนครบทุกบิต

# 731 การแปลงแอนะลอกเป็นดิจิตอล

## (Analog-to-Digital Conversion)




การแปลงแอนะลอกเป็นดิจิตอลแบบ Successive-Approximation (SA)



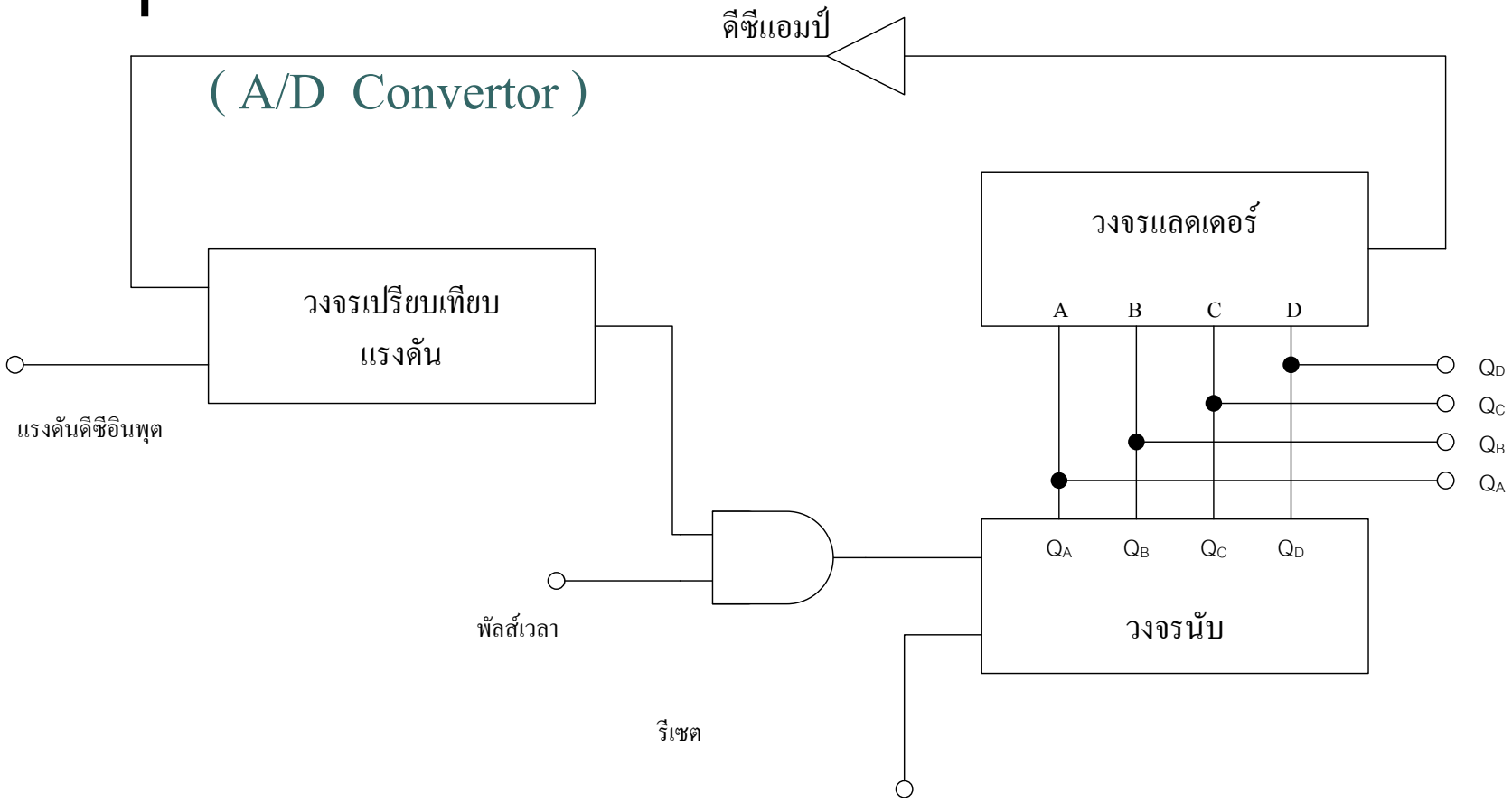

# วงจรแปลงแอนะลอกเป็นดิจิตอล

# อธิบาย

**วงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล จะประกอบไปด้วย 4 ส่วน**

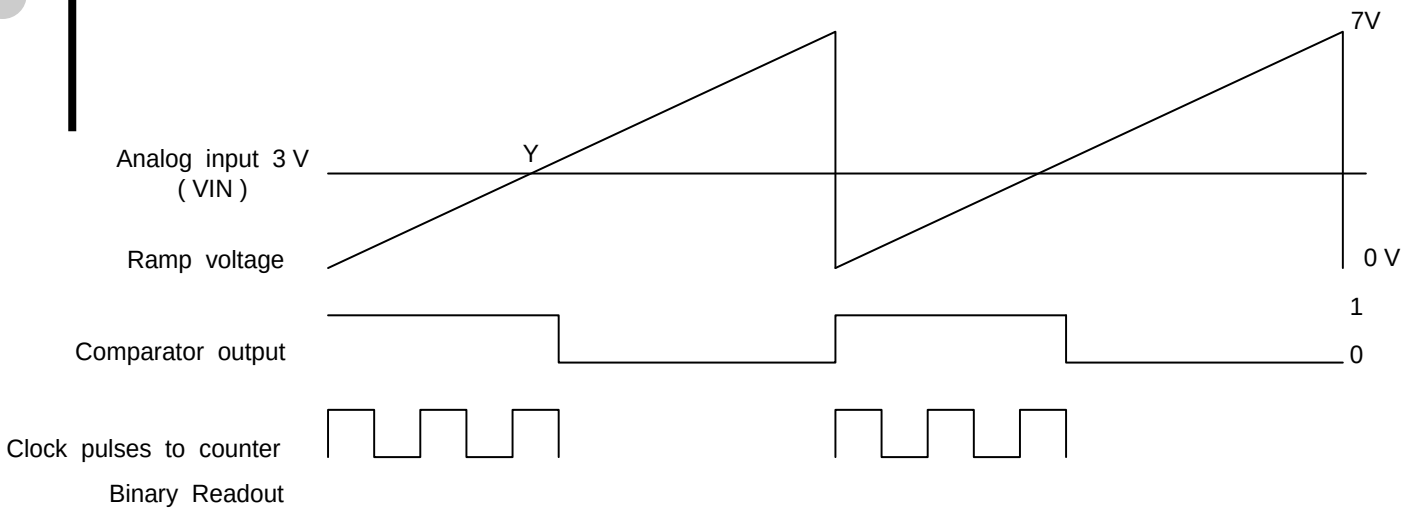
- เปรียบเทียบแรงดันอินพุตกับสัญญาณอ้างอิง
- เกตจะเปิด-ปิดสัญญาณนาฬิกาเพื่อป้อนเข้าวงจรนับ
- ตัวนับบีซีดีแสดงผลในการนับพัลซ์และส่งสัญญาณดิจิตอลไปยังวงจร
- ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกจะป้อนกลับไปยังอินพุตแบบลาดเอียง

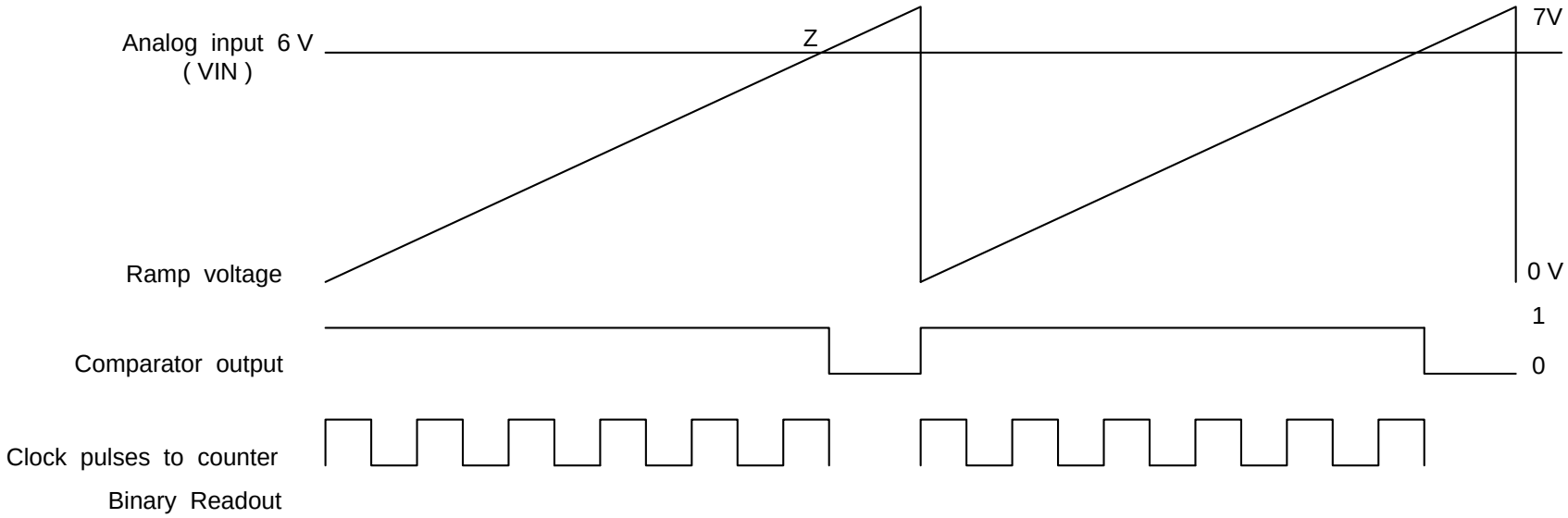
# 732การทำงานของA/Dแบบเคาว์เตอร์แร็มพ (Counter Ramp)



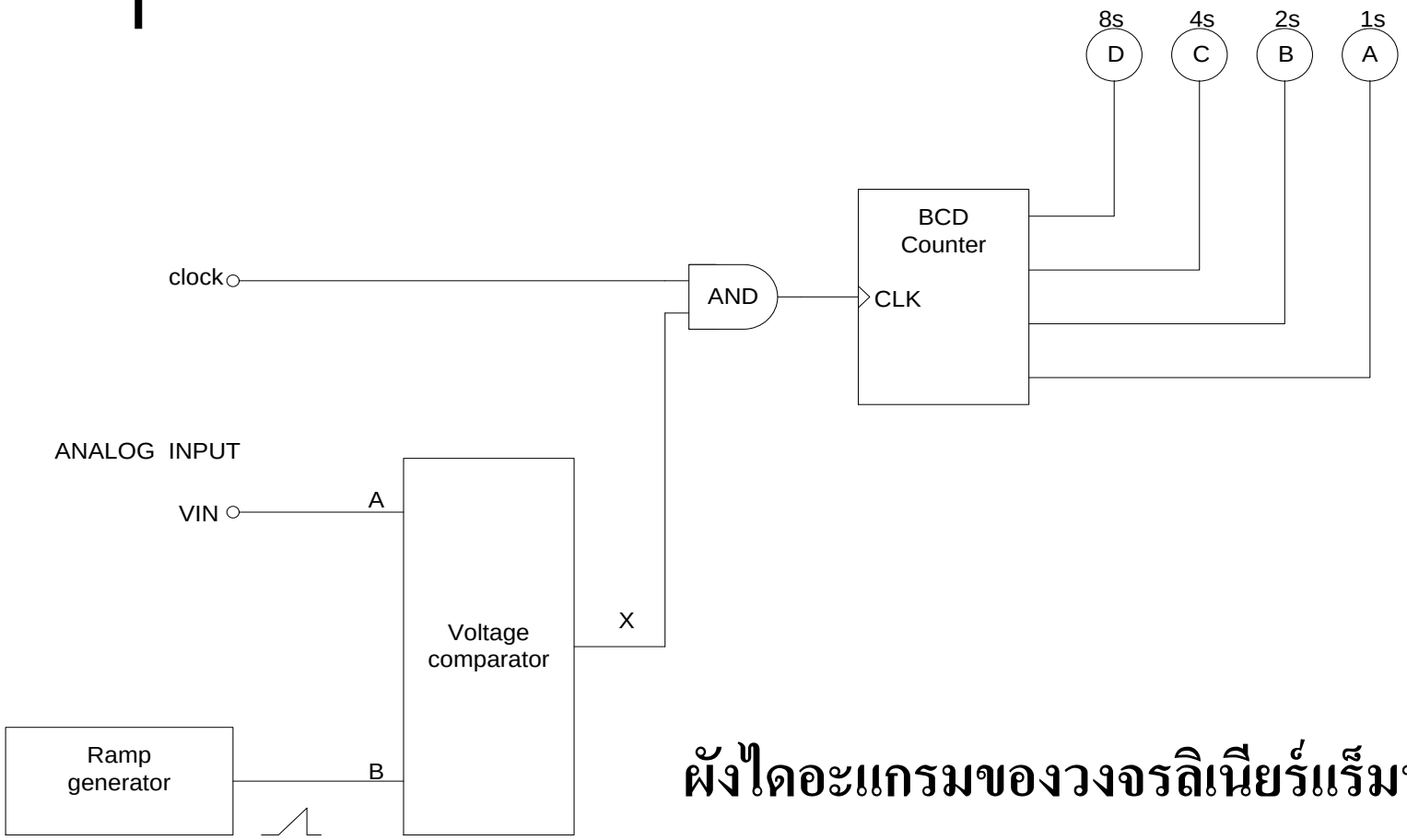

**วงจร A/D แบบเคาว์เตอร์แร็มพ (Counter Ramp)**

# รูปสัญญาณการทำงาน



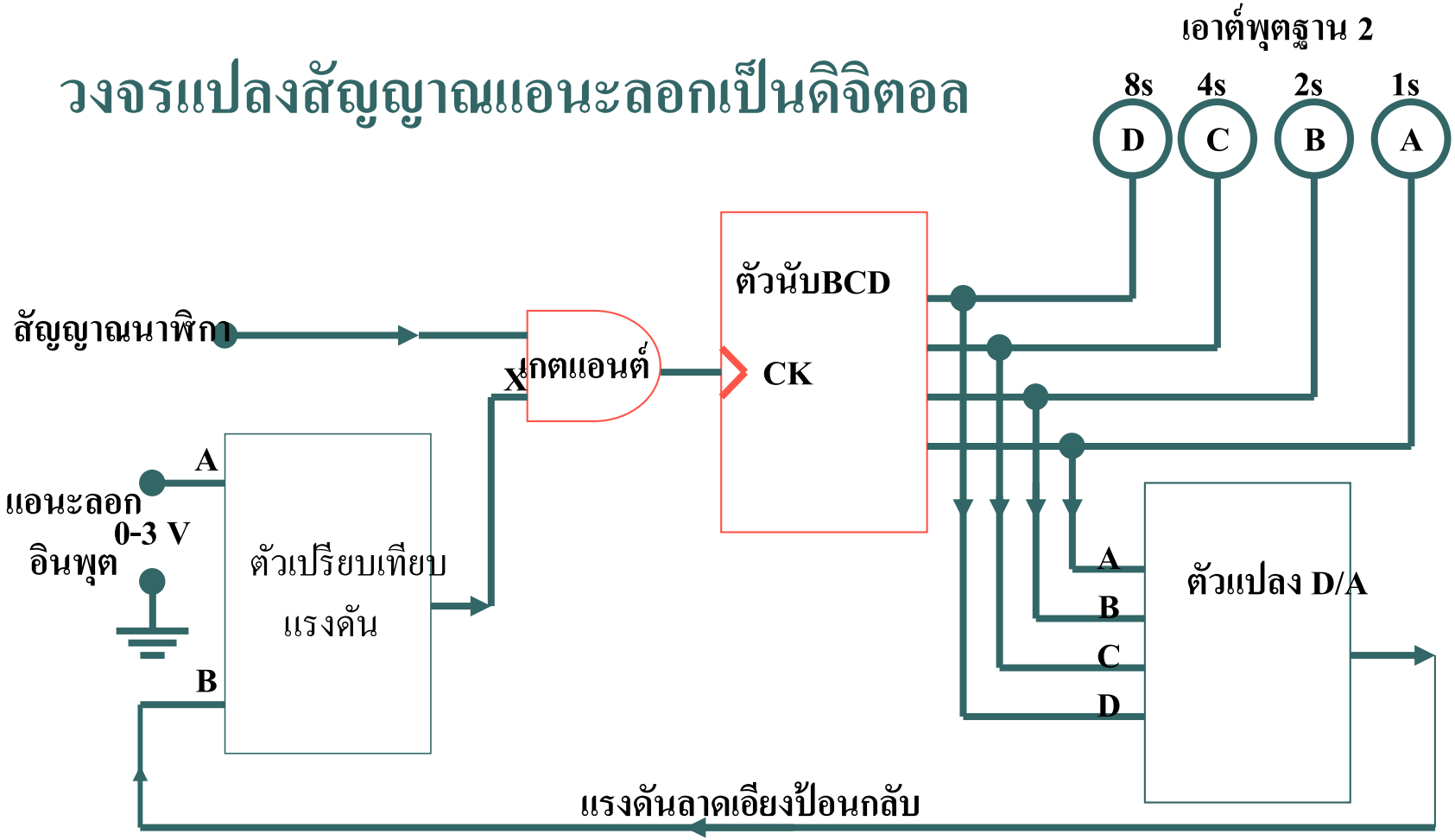
# วงจรลิเนียร์แร็มพ A/D ( Linear Ramp A/D )



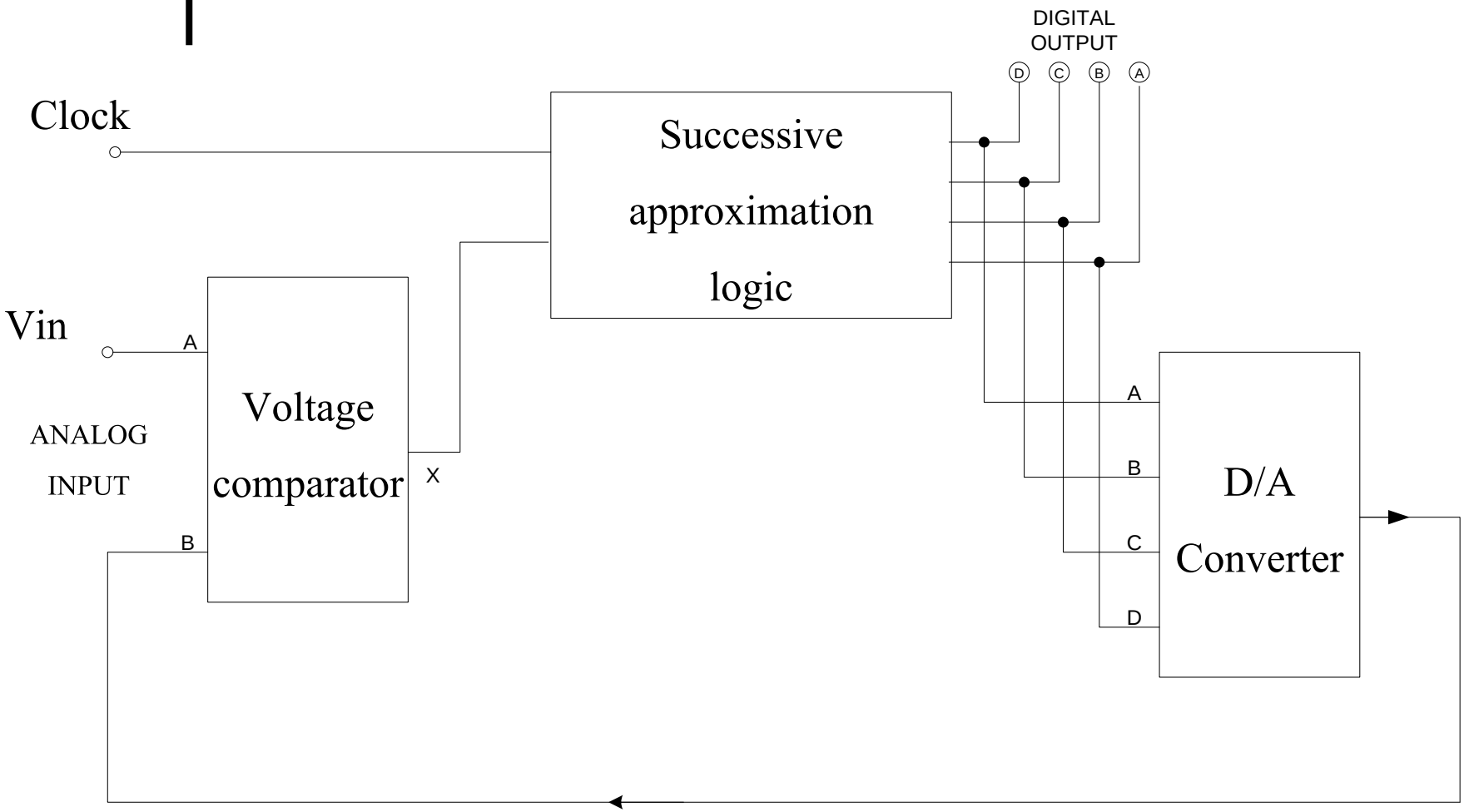

ผังไดอะแกรมของวงจรลิเนียร์แร็มพ์ A/D

# การแปลงแบบลาดเอียง

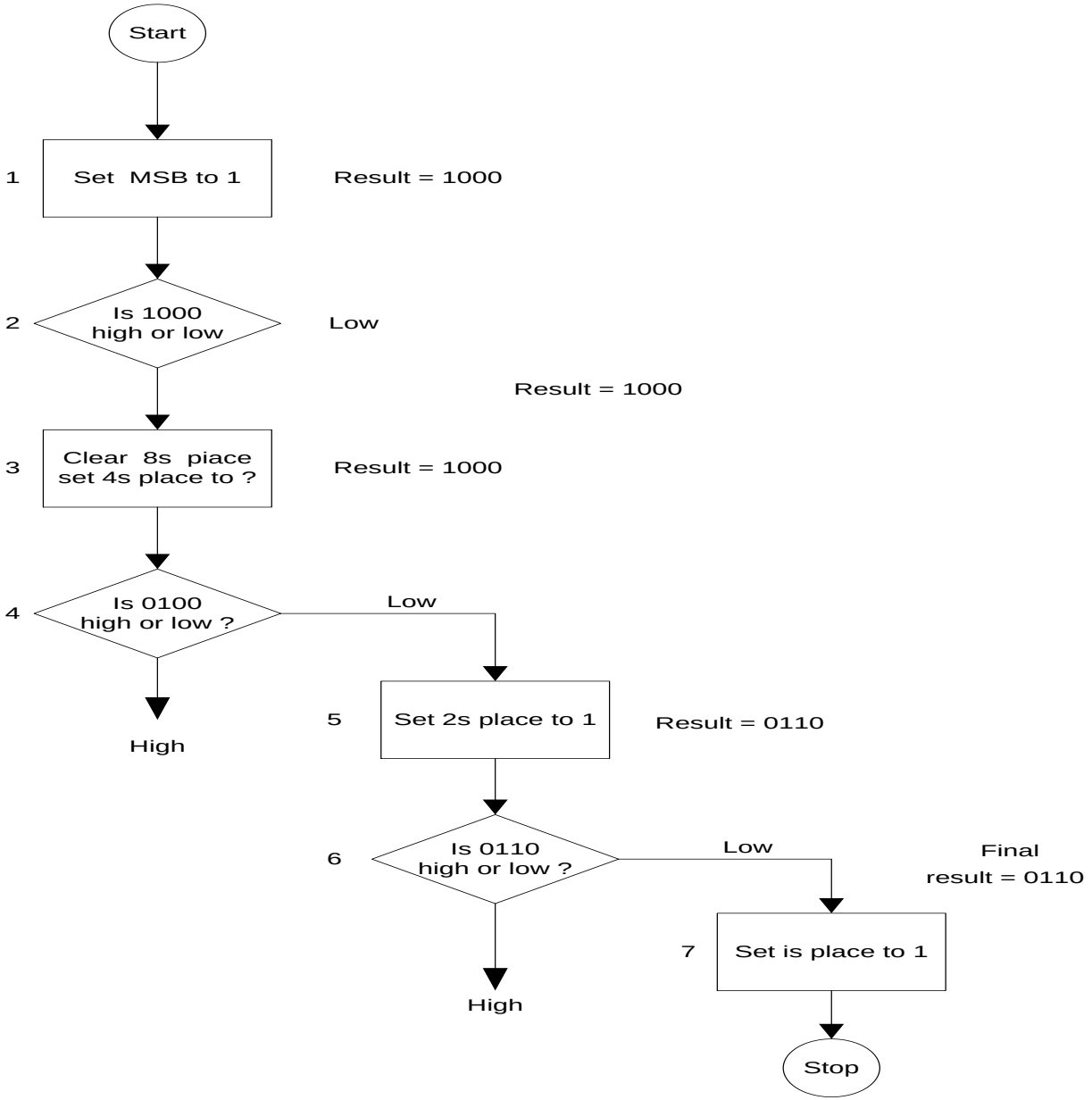
## วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล



# 733 วงจร Succesive – Approximation ADC

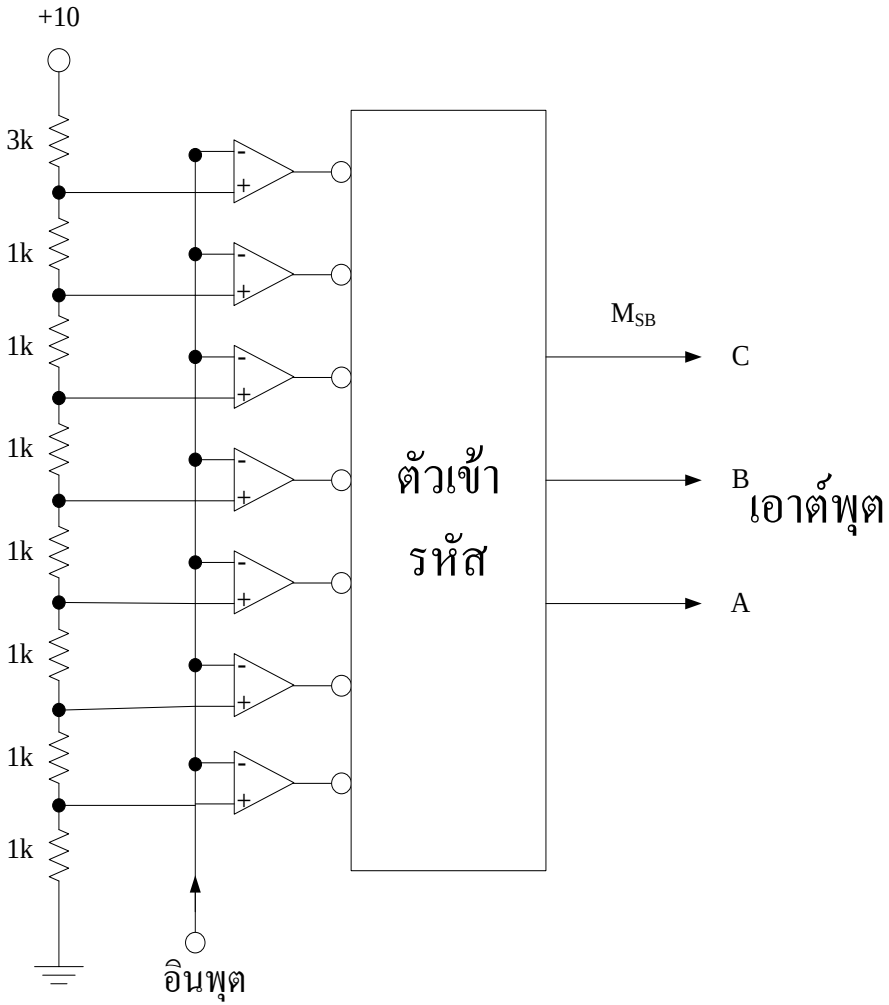


# ผังไดอะแกรม Successive Approximation A/D



# 734 การออกแบบวงจรแบบ Flash ADCs

การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลแบบแฟลช



## ตารางแสดงการเปรียบเทียบแรงดันและเอาต์พุตของวงจรแปลง A/D แบบแฟลช

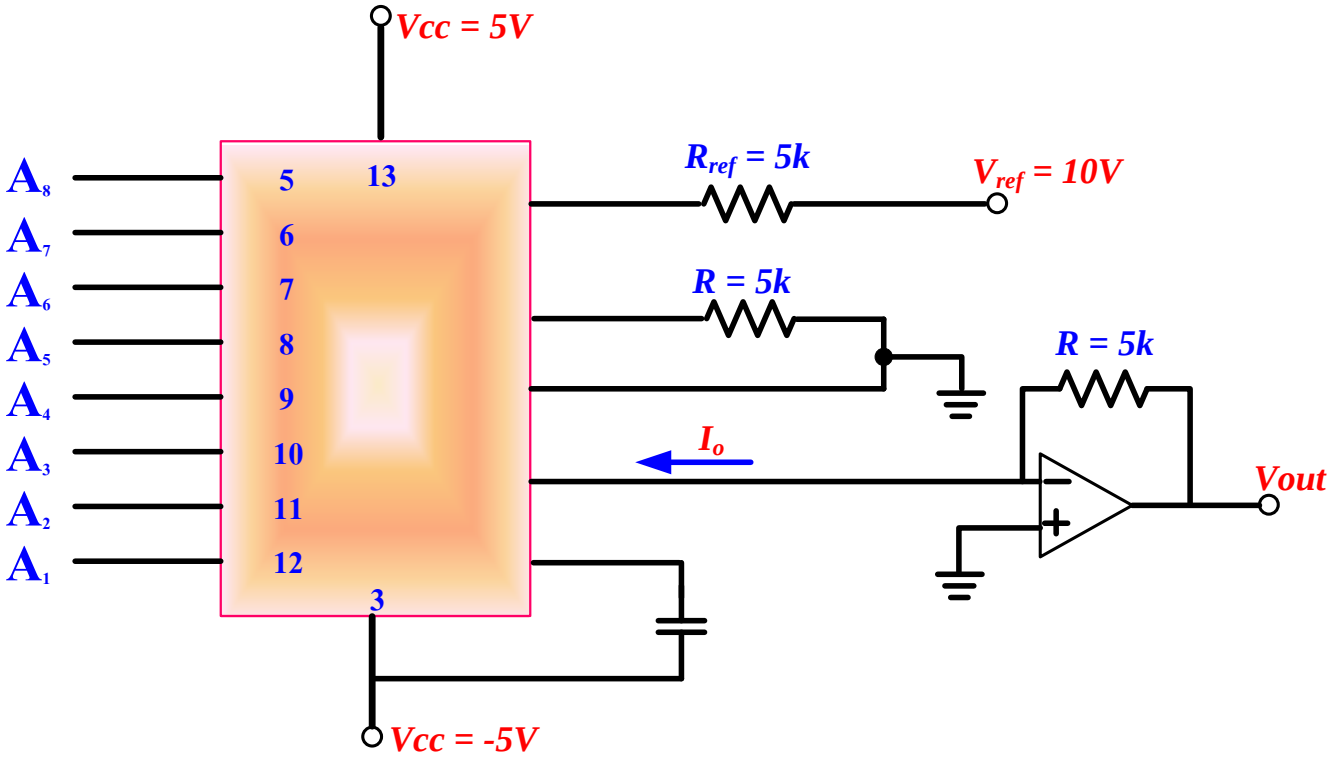
| อินพุต | เอาต์พุตของออปแอมป์ | | | | | | | ดิจิตอลเอาต์พุต | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0V | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1-2V | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 2-3V | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 3-4V | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 4-5V | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 5-6V | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 6-7V | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| >7V | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

# อธิบาย

- วงจรแปลงแบบแฟลชนั้นมีความเร็วในการทำงานเมื่อเทียบกับแบบอื่นๆ
- ในการแปลงสัญญาณที่สูงมากเนื่องจากออปแอมป์มีความไวสูงแต่วงจรก็จะมีขนาดใหญ่กว่าแบบอื่นๆ
- แต่ก็มีข้อเสียของขนาดในการใช้ออปแอมป์ในการเปรียบเทียบแรงดันหลายตัวด้วยกัน
- ค่าแรงดันการเปรียบเทียบจากตารางสามารถกำหนดการเปรียบเทียบโดยใช้วงจรแบ่งแรงดันดูวงจรประกอบ
- สัญญาณอินพุตทางแอนะลอกจะเปลี่ยนแปลงขึ้นและลงที่เอาต์พุตออปแอมป์ จะเกิดค่าระดับทางดิจิตอลเป็น0 และ1 ค่าระดับที่ได้นี้จะควบคุมวงจรเข้ารหัส
- การสร้างรหัสทางเอาต์พุต ดูจากตารางประกอบเปรียบเทียบค่าอินพุตกับเอาต์พุต
- รหัสทางBCDทางเอาต์พุต จะถูกสร้างจากวงจรเข้ารหัส โดยมีเอาต์พุต ของวงจรเปรียบเทียบควบคุมการสร้างรหัส

# 74 การประยุกต์ใช้งานจากวงจรADC กับ DAC

## 741 การนำไอซี เบอร์ DAC 0808 ไปใช้งานแปลงค่า D/A

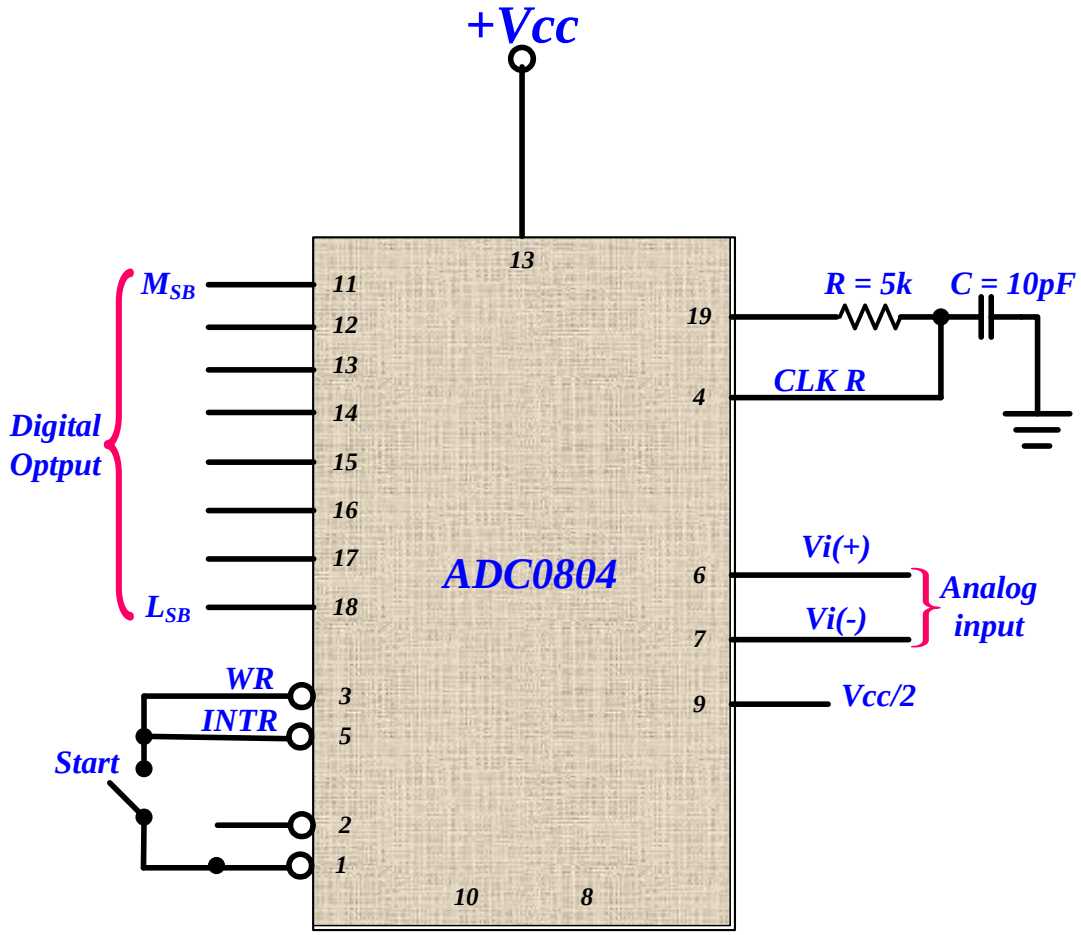



วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกที่ใช้ตัว DAC0808 ขนาด 8 บิต

# การแปลงสัญญาณดิจิตอลไปเป็นสัญญาณแอนะลอก

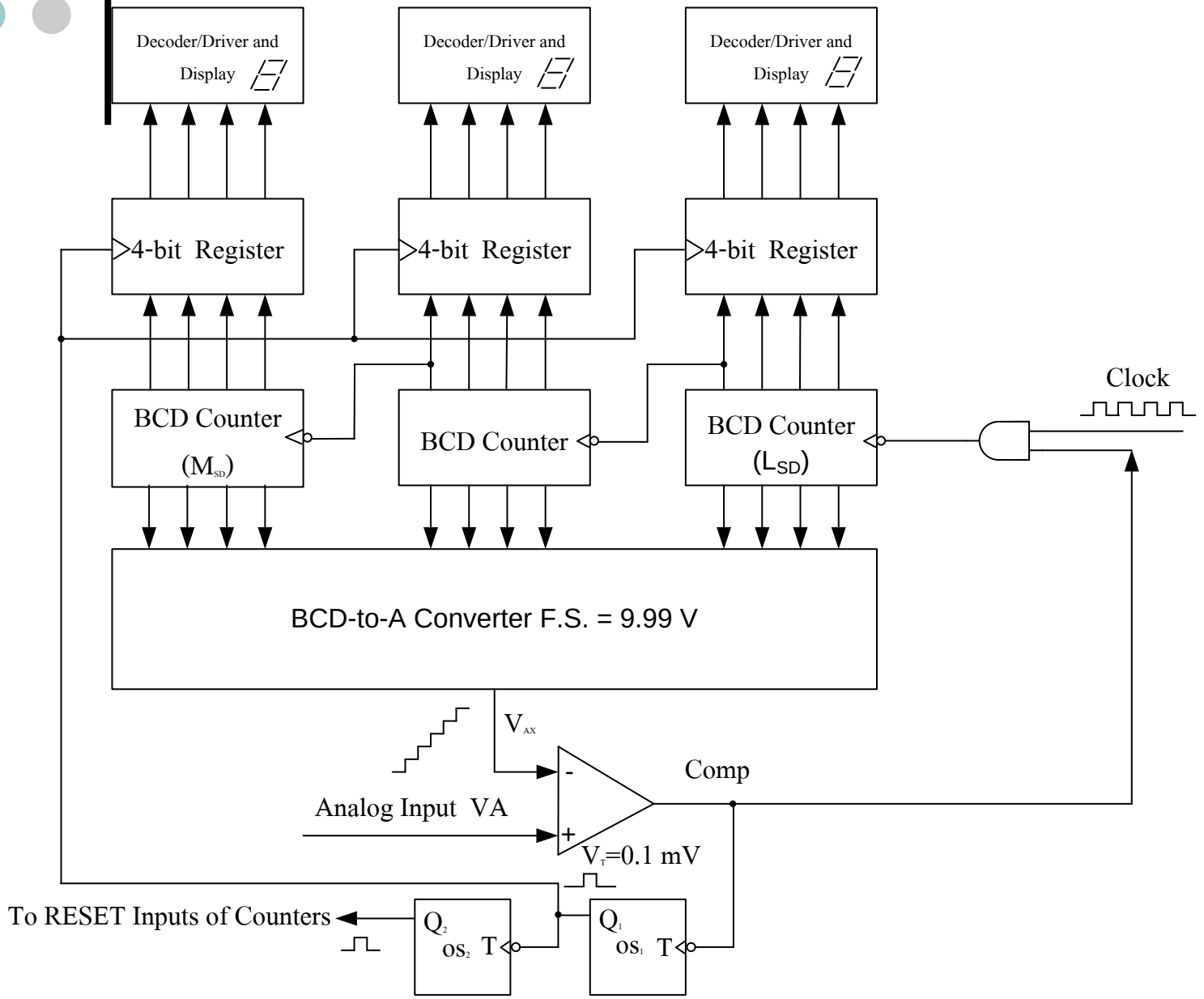
DAC0808 ให้สัญญาณเอาต์พุตเปลี่ยนแปลงในรูปของกระแส ( $I_o$) โดยกระแสจะมีทิศทางเป็นบวกถ้ากระแสไหลเข้าตัว DAC ดังนั้น หากใช้งานเอาต์พุตที่อยู่ในรูปของแรงดัน ต้องต่อวงจรแปลงกระแส เป็นแรงดันก่อนดังรูปข้างล่างโดยใช้ออปแอมป์เป็นตัวเปลี่ยนกระแสที่ไหลผ่านความต้านทาน 5Kให้เป็นแรงดัน โดยแรงดันที่เอาต์พุตของออปแอมป์จะมีค่าเท่ากับ +10 โวลท์ ถ้าข้อมูลอินพุตทั้งหมดเป็น 1(high) และแรงดันที่เอาต์พุตจะมีค่าเท่ากับ 0 โวลท์ ถ้าข้อมูลอินพุตทั้งหมดเป็น 0 (Low)

# การแปลงสัญญาณดิจิตอลไปเป็นสัญญาณแอนะลอก

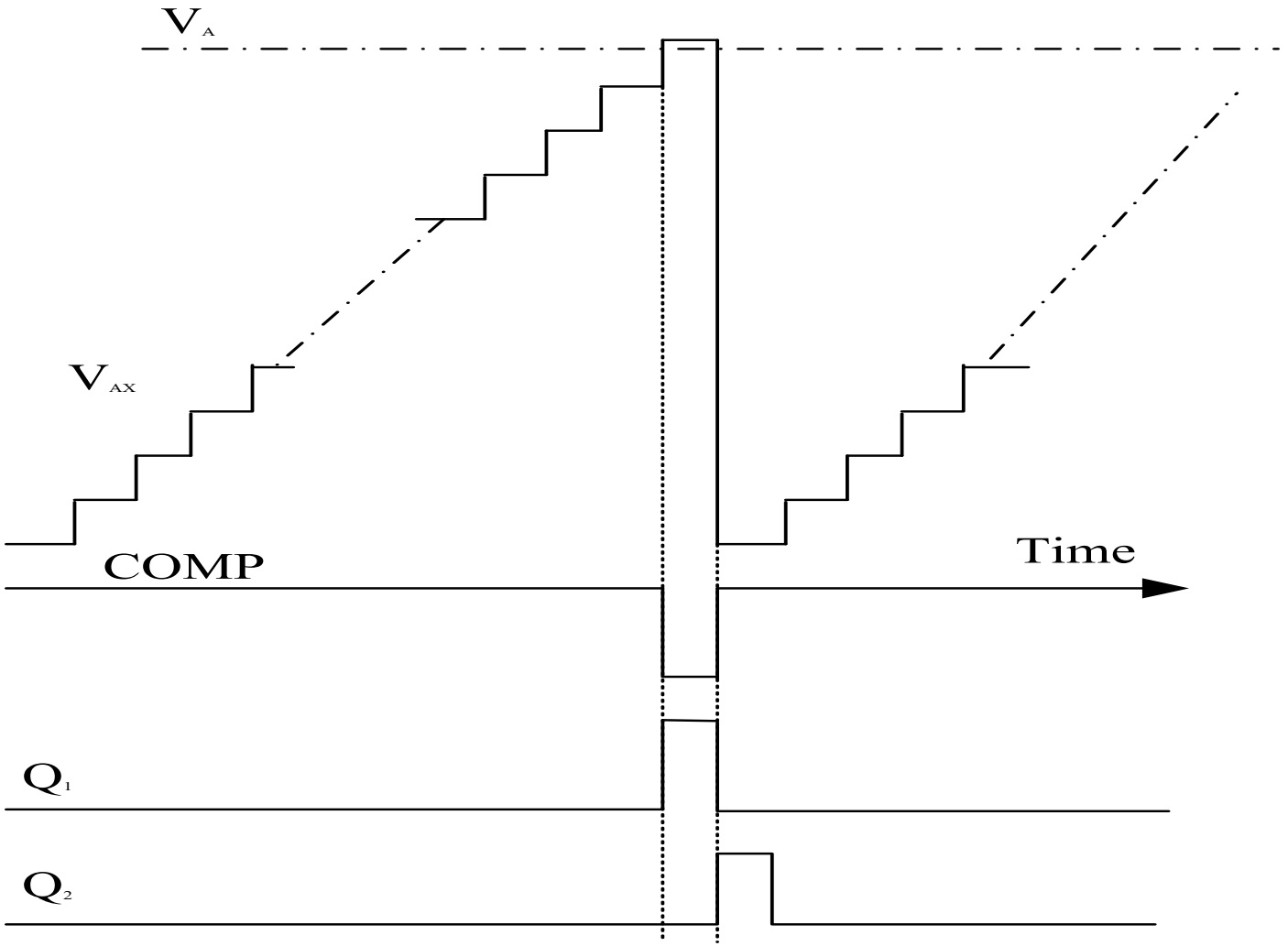



ตัวอย่างการใช้งาน IC D/A

# 742 การนำระบบดิจิตอลไปใช้งานในเครื่องมือวัดแบบตัวเลข



# รูปแสดง Waveforms ของสัญญาณควบคุมการทำงานวงจร DVM



# อธิบาย

- การนำวงจรA/D ไปใช้ในการเปลี่ยนค่าปริมาณของระดับแรงดันทางไฟฟ้าในรูปแบบ แอนะลอกให้แสดงผลเป็นตัวเลขทางดิจิตอล
- การทำงานของวงจรจะเปลี่ยนแรงดันทางแอนะลอกเป็นรหัสทางBCD มีค่าเต็มสเกล 9.99 โวลต์ Resolution= 10mV
- BCD Counter ขนาด 3 บิต ทำหน้าที่นับเลขแบบ BCD ตามวงจร A/D ที่ส่งค่ามาควบคุมคล็อก
- จากการนับที่เกิดขึ้นที่วงจรนี้จะส่งเอาต์พุตไปยัง Register ทำการเก็บข้อมูล เพื่อนำไปแสดงผล
- วงจรดีโค้ดเดอร์และวงจรแสดงผลทำหน้าที่ถอดรหัสทางดิจิตอล BCD ให้เป็นตัวเลขฐานสิบเพื่อนำไปแสดงผลด้วย Seven Segment ต่อไป
- นักศึกษาลองทำความเข้าใจ ตามที่ได้ศึกษากันมาครบทุกวงจร

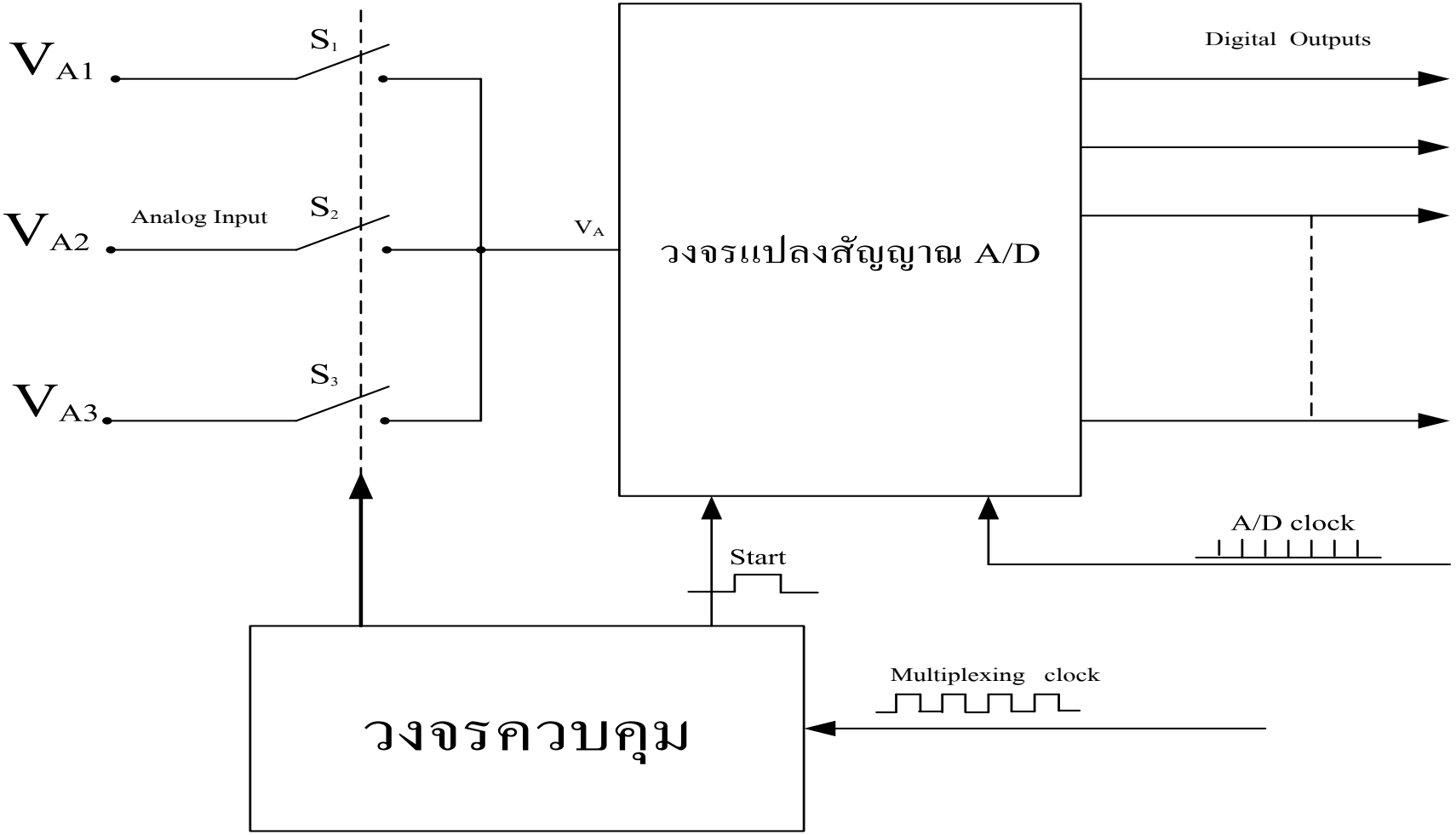
# ทดสอบความรู้สัปดาห์ที่ 14

1. จงอธิบายถึงความหมายของคำเหล่านี้พร้อม ยกตัวอย่างการนำไปใช้งาน

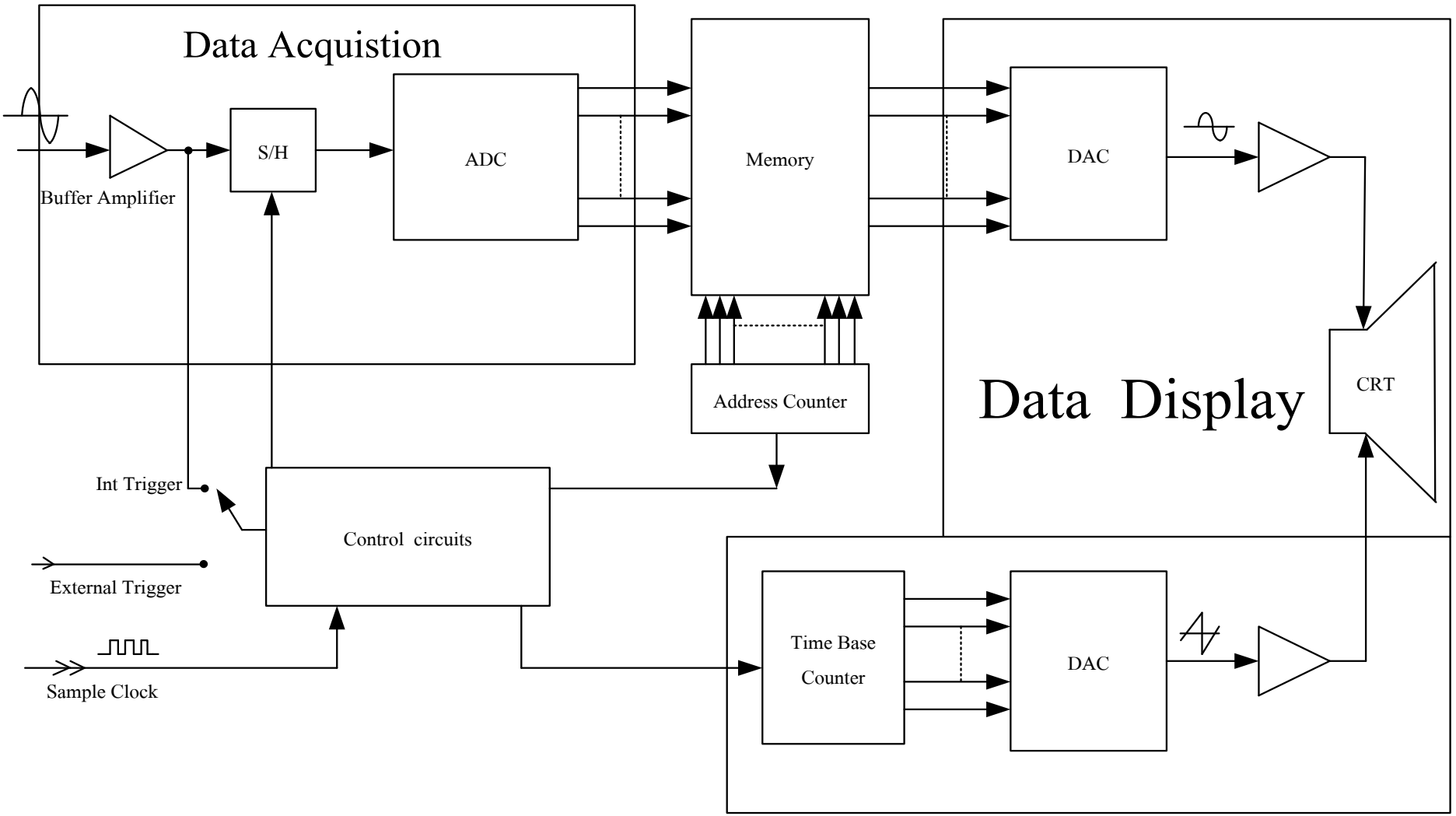
- ADC
- DAC
- Resolution
- Sampling-And-Hold Circuit
- Conversion Time, tc
- Digital –Ramp ADC
- Data Acquisition

# 2. ให้นักศึกษาอธิบายการทำงานของวงจรถึงลำดับการทำงาน

- พร้อมหน้าที่ของแต่ละบล็อก



3. จากบล็อกไดอะแกรมของ Digital Storage Oscilloscope ที่ให้มา นักศึกษาลองวิเคราะห์การทำงาน โดยนำหลักการที่ศึกษามาทั้งหมด



# เนื้อหาคำบรรยาย สัปดาห์ที่15 การประยุกต์ใช้วงจรดิจิตอลในงานอุตสาหกรรม

# 81 โปรแกรมเมเบิลคอนโทรลเลอร์

PLC หรือ Programmable Logic Controller เริ่มได้รับการพัฒนาขึ้นครั้งแรกในปี ค.ศ 1968 โดย บริษัท General

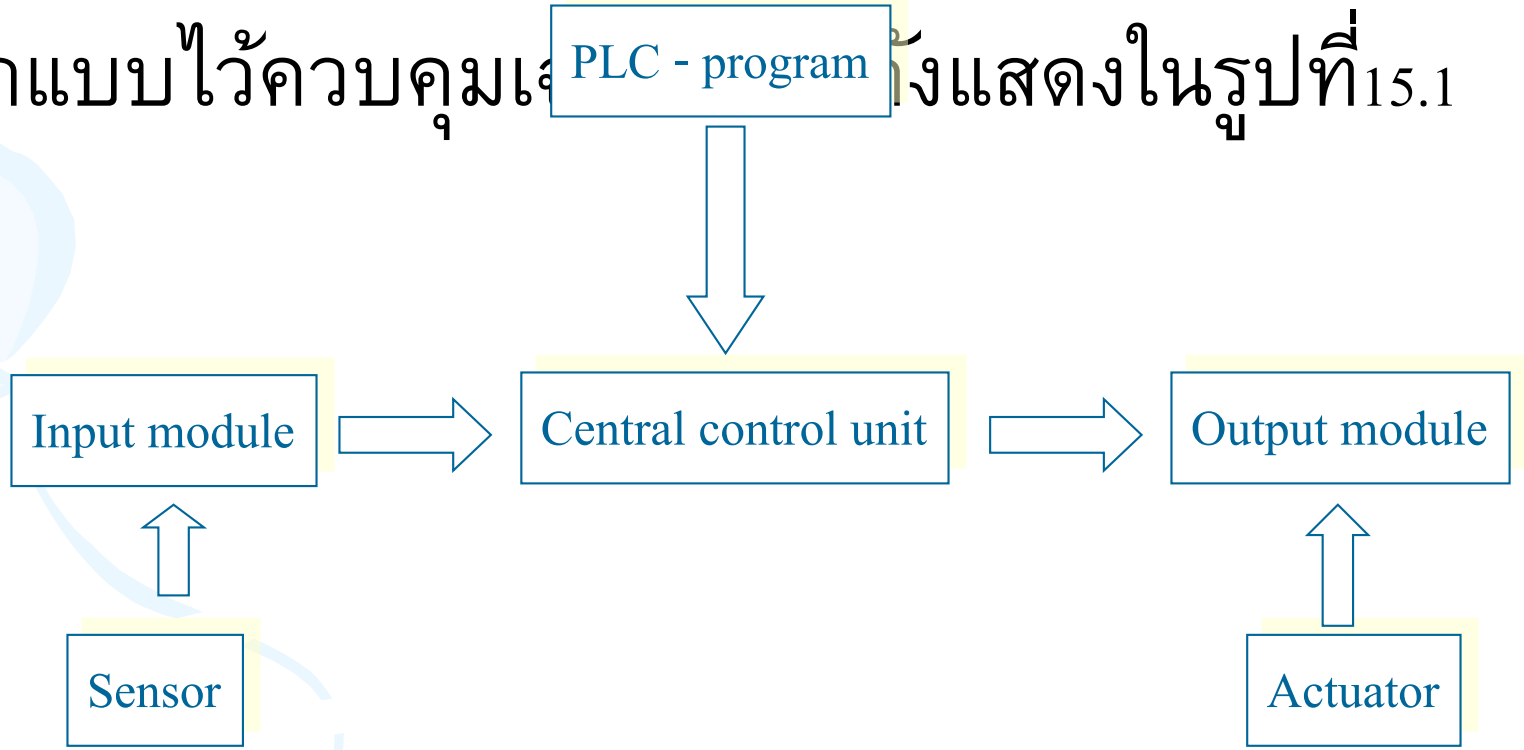
รูปที่ 15.1 โปรแกรมเมเบิลลอจิกคอนโทรล

# 811ลักษณะเด่นของ PLC

- ใช้การเขียนโปรแกรมในการสร้างฟังก์ชันการทำงานแทนการใช้สายไฟ เพื่อควบคุมการทำงานของระบบเหมาะกับงานที่มีการเปลี่ยนแปลงและแก้ไขการควบคุมอยู่ตลอดเวลา
- มีประสิทธิภาพ และมีขนาดเล็กกว่า เมื่อเทียบกับ

# 812 ส่วนประกอบของระบบใน PLC

PLC มีลักษณะไม่แตกต่างจากคอมพิวเตอร์ แต่จะ
ออกแบบไว้ควบคุมเ ังแสดงในรูปที่15.1




รูปที่ 15.2 ผังแสดงส่วนประกอบของระบบใน PLC

# ขนาด PLC

| ขนาดของ PLC | จำนวน I/O สูงสุด | หน่วยความจำโปรแกรม |
|---|---|---|
| ขนาดเล็ก (Small size) | ไม่เกิน 128 / 128 | 4 Kbyte |
| ขนาดกลาง (Medium size) | ไม่เกิน 1024 / 1024 | 16 Kbyte |
| ขนาดใหญ่ (Large size) | ไม่เกิน 2048 / 2048 | 64 Kbyte |
| ขนาดใหญ่มาก (Very large size) | ประมาณ 8192 / 8192 | 256 Kbyte |
| แสดงการจำแนกขนาดของโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ | | |

นอกจากนั้นเรายังสามารถที่จะแบ่ง โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ตามโครงสร้างออกเป็น 2 ลักษณะ คือ

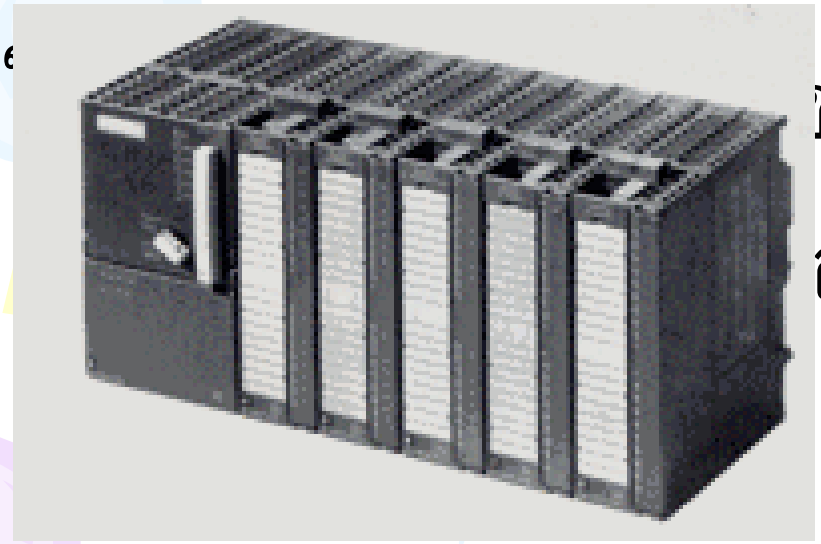
# 814 PLC แบบ Compact

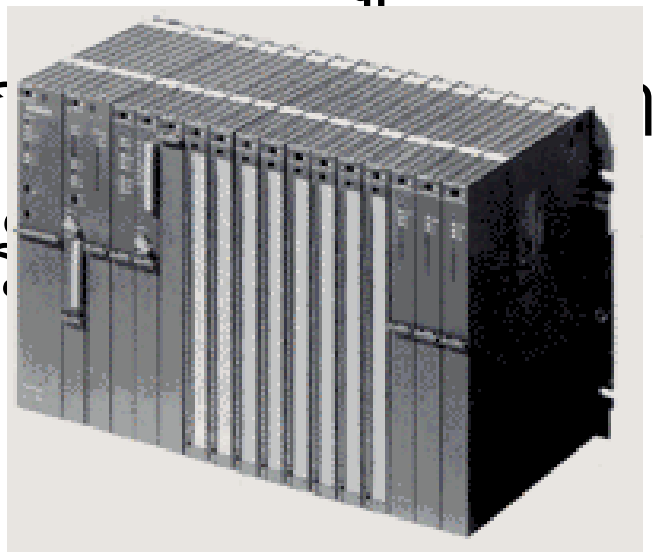
ที่มีขนาดเล็กกะทัดรัดมีหน่วยอินพุต/เอาต์พุตและหน่วยสำหรับติดต่อสื่อสารข้อมูล ประกอบรวมกันอยู่ภายในโครงสร้างเดียวกัน เหมาะสำหรับงานที่มีการกำหน อินพุต/เอาต์พุตที่เ จำนวนไม่มาก เ การควบคุมเครื่ ต้น

รูปที่15.3 PLC โครงสร้างแบบ Compact

# 815 PLC แบบ Modular

มีลักษณะเป็นโมดูล เชื่อมต่อกันอยู่บน Rack สามารถจะทำการถอดและเสียบโมดูลที่ต้องการ





รูปที่15.4 PLC โครงสร้างแบบ Modular

# 82 มาตรฐานสำหรับ PLC

ใน PLC รุ่นเก่า แต่ละบริษัทไม่สามารถเทียบเคียงภาษาที่ใช้ในควบคุมโปรแกรมตลอดจนอุปกรณ์ที่ใช้ ทำให้ยุ่งยากในการเขียนโปรแกรมของแต่ละบริษัท

ในปี 1992 IEC ได้จัดทำเป็นมาตรฐานสากล (International standard) สำหรับ PLC โดย

# 83 มาตรฐานใหม่ IEC 1131 สำหรับ PLC

1) ข่าวสารทั่วไป (General Information)

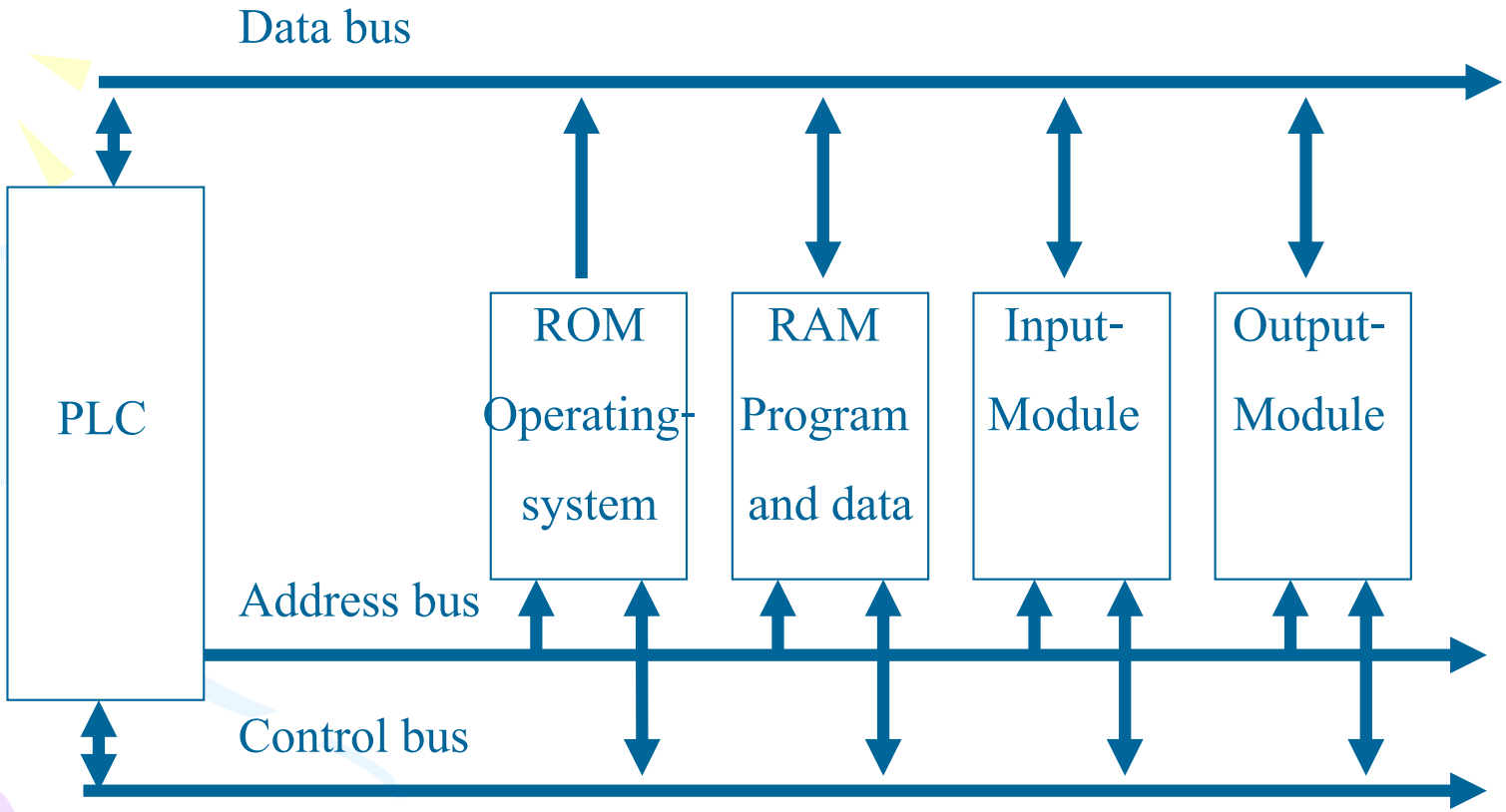
2) ความต้องการอุปกรณ์และทดสอบ (Equipment Requirements and test)

3) ภาษาการทำโปรแกรม (Programming Language)

4) ไกด์ลายสำหรับผู้ใช้ (User Guidelines)

5) ข้อกำหนดรายละเอียดการบริการส่งข่าวสารโดยตรง

# 84 พื้นฐานการออกแบบระบบ PLC




รูปที่ 15.5 พื้นฐานการออกแบบระบบ PLC

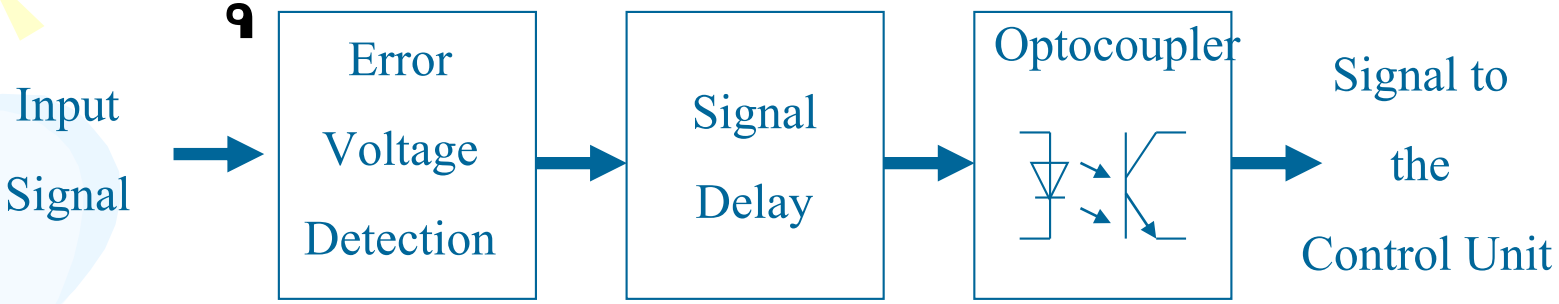
# 841 หน่วยอินพุต

หน้าที่สำคัญของหน่วยอินพุตจะมีดังนี้

1) ทำการตรวจสอบสัญญาณที่รับเข้ามาว่ามีความถูกต้องน่าเชื่อถือเพียงใด

2) จะทำหน้าที่ปรับขนาดของสัญญาณที่รับเข้ามาให้มีขนาดเหมาะสมกับหน่วยควบคุมกลาง

# 842 บล็อกไดอะแกรมของหน่วยอินพุต




รูปที่ 15.6 บล็อกไดอะแกรมของหน่วยอินพุต

# 843 หน่วยเอาต์พุต

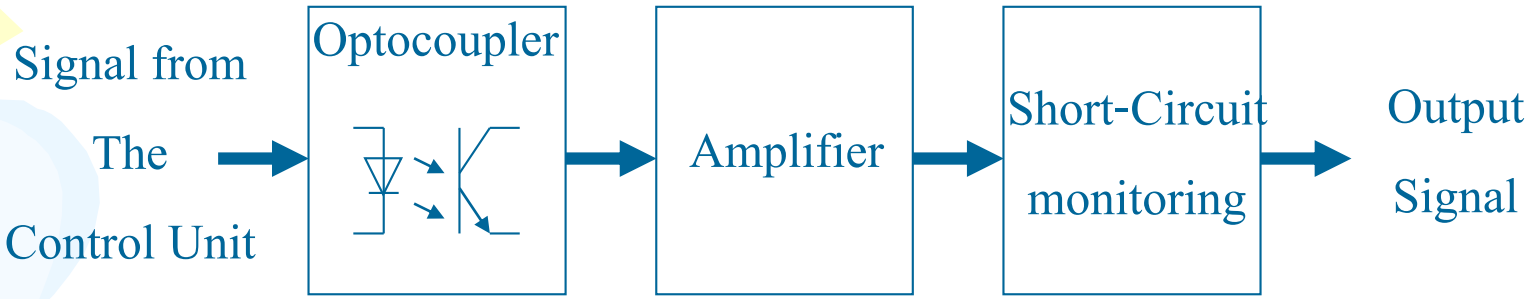
หน้าที่สำคัญของหน่วยเอาต์พุตจะมีดังนี้

-ทำหน้าที่ปรับแรงดันลอจิกจากการประมวลผลให้เป็นแรงดันควบคุม

- ป้องกันการรบกวนจากสัญญาณภายนอก ไม่ให้เข้ามารบกวนการ

ทำงานของระบบ PLC

# 844 บล็อกไดอะแกรมของหน่วยเอาต์พุต




รูปที่ 15.7 บล็อกไดอะแกรมของหน่วยเอาต์พุต

# 85 การเขียนโปรกแกรมของ PLC

มี 5 ขั้นตอนในการสร้างโปรแรกมควบคุมของ PLC

1 Specification

- Description of control task
- Position sketch
- macrostructure of control program

2 Design

- Function chart to IEC 848
- Function diagrams such as displacement step diagram
- Function table
- Definition of software modules
- Circuit diagram
- Input/Output list

3 **Realization**

- Programming in LD, FBD, IL, ST, and SFC
- Simulation of subprograms

4 **Commissioning**

- Design of system
- Testing of subprograms
- Testing of overall program

5 **Documentation**

- Description of control task
- Position sketch
- Circuit diagram
- Terminal diagram
- printouts of control programs in programming languages
- Allocation list of inputs and outputs

# 851ภาษาโปรแกรม

ภาษาที่ใช้ในการเขียนโปรแกรมตามมาตรฐาน IEC 1131-3 กำหนดไว้ 5 ภาษา คือ

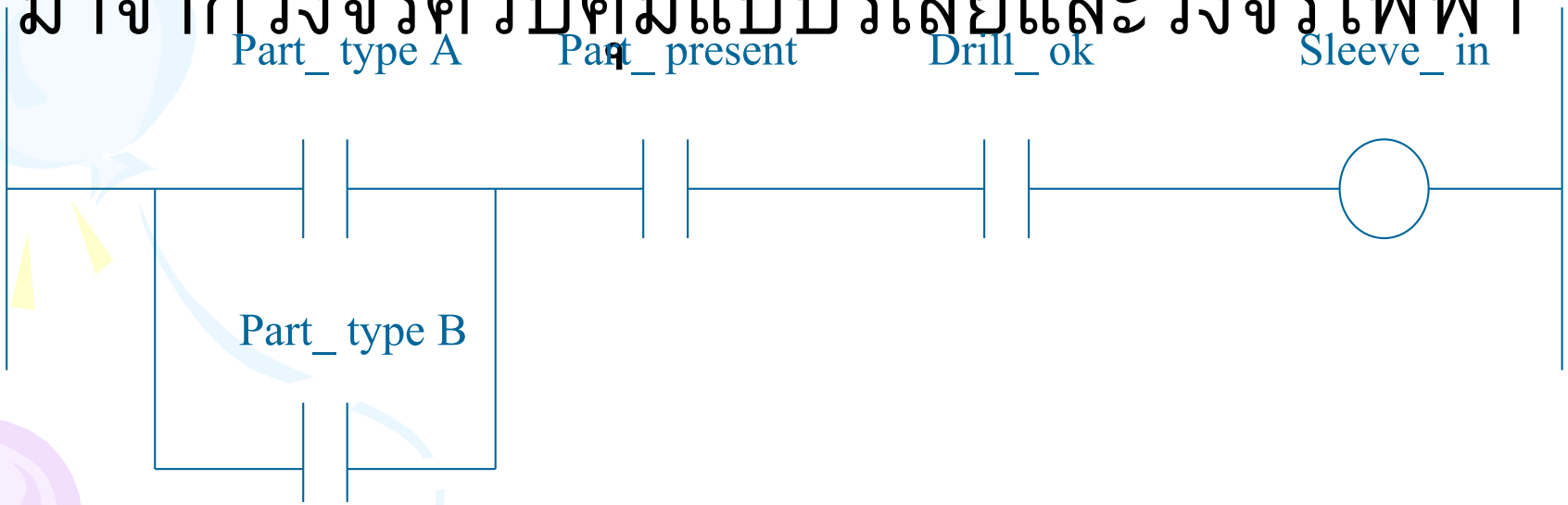
1) LD (Ladder Diagram)

2) FBD (Function Block Diagram)

3) IL (Instruction List)

# 852 แลดเดอร์ไดอะแกรม (Ladder diagram : LD)

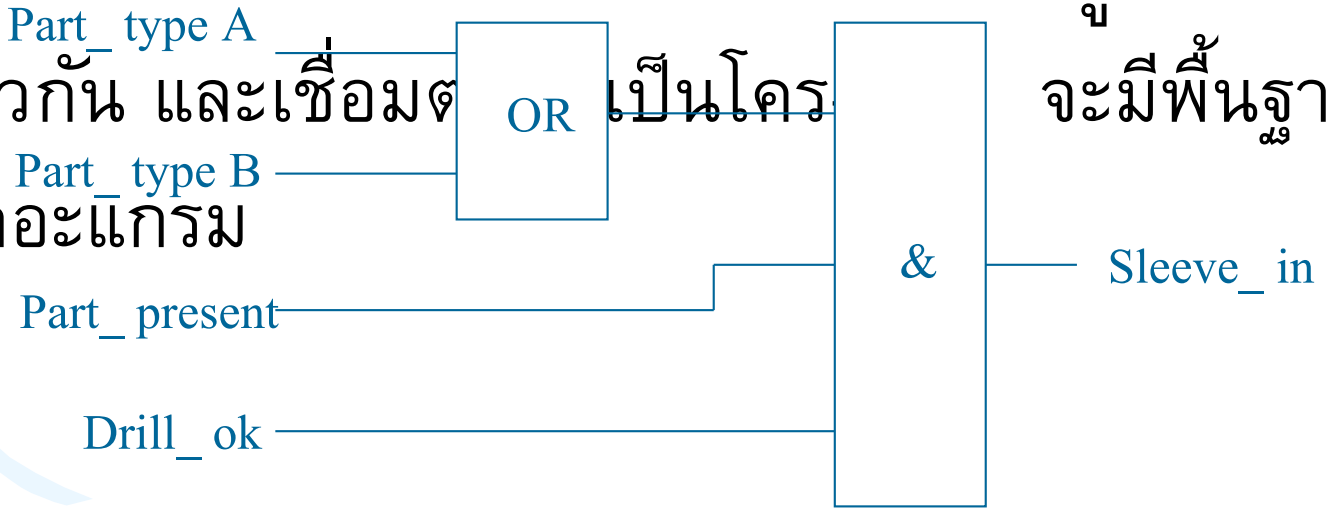
-เป็นภาษาที่เขียนอยู่ในรูปของกราฟฟิกมีพื้นฐานมาจากวงจรควบคุมแบบรีเลย์และวงจรไฟฟ้า




รูปที่ 15.8 ตัวอย่างโปรแกรมในรูปแบบของภาษา LD

# 853 ฟังก์ชันบล็อกไดอะแกรม (Function block diagram : FBD)

เป็นภาษาที่แสดงฟังก์ชันการทำงานในรูปของการาฟิก
เช่นเดียวกัน และเชื่อมต OR เป็นโคร จะมีพื้นฐานมาจาก
ลอจิกไดอะแกรม




รูปที่ 15.9 ตัวอย่างโปรแกรมในรูปแบบของภาษา FBD

# 854 อินสตรัคชันลิสท์ (Instruction list : IL)

เป็นภาษาที่ที่อยู่ในรูปของข้อความ คล้ายกับภาษาแอสเซมบลี่(Assembly) ซึ่งภายในหนึ่งคำสั่งควบคุมประกอบด้วย ส่วนปฏิบัติการ(Operator) และส่วนที่ถูกดำเนินการ(Operand)

ตัวอย่าง

```
LD      Part_typeA
OR      Part_typeB
AND     Part_present
```

# 855 สตรัคเจอร์เทซท์ (Structure text : ST)

จัดเป็นภาษาระดับสูง มีพื้นฐานมาจากภาษา Pascal ซึ่งจะประกอบไปด้วยนิพจน์ (Expression) และคำสั่ง(Instruction)

ตัวอย่าง

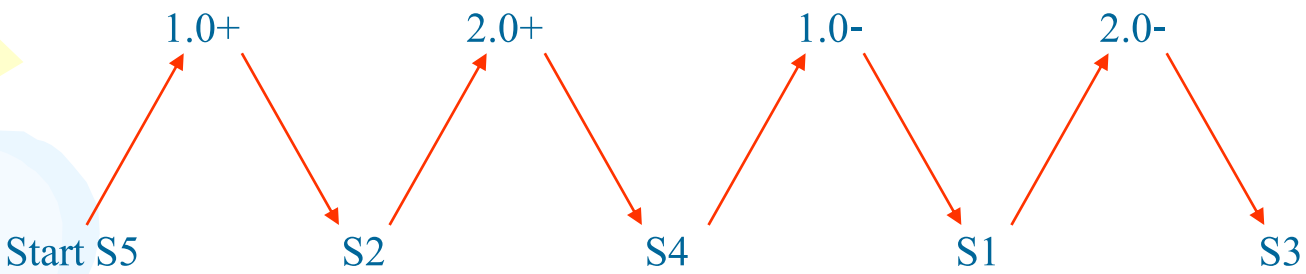
```
ACTION Action_2:

        %QX1.2:=%IX0.5 OR (%MX1 AND %MX3);
```

## 856ซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)

เป็นภาษาที่รองรับการเขียนโปรแกรมที่มีโครงสร้างการทำงานแบบซีเควนซ์ ซึ่งส่วนประกอบของ SFC จะประกอบด้วย Step (คำสั่งในการปฏิบัติการในแต่ละขั้นตอน) และ Transition (เงื่อนไขที่กำหนดให้กระทำคำสั่งในแต่ละ Step)

ในแต่ละ Step จะแสดงสภาวะของ

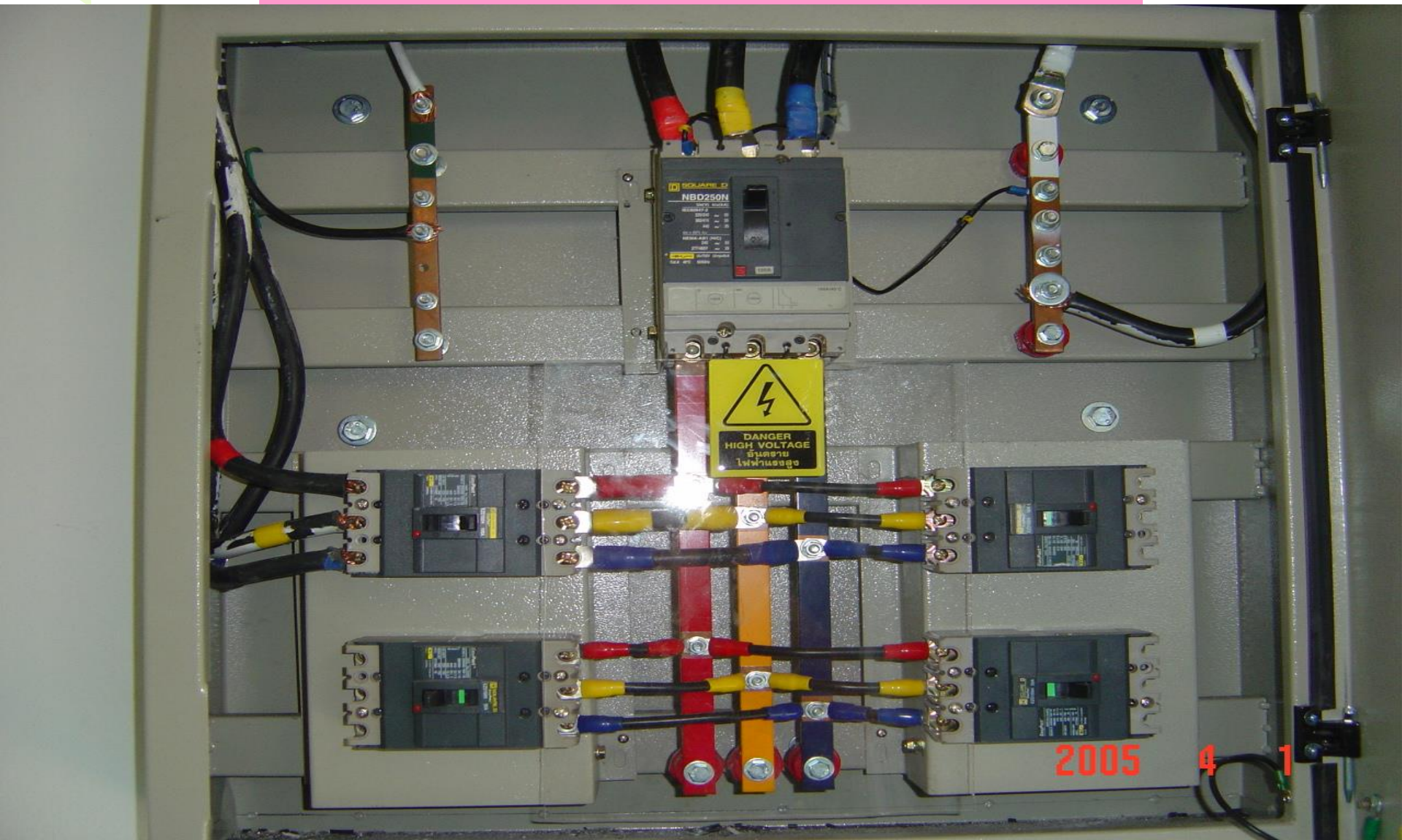
# 857 ซีเควนเชียลฟังก์ชันชาร์ท (Sequential function chart : SFC)



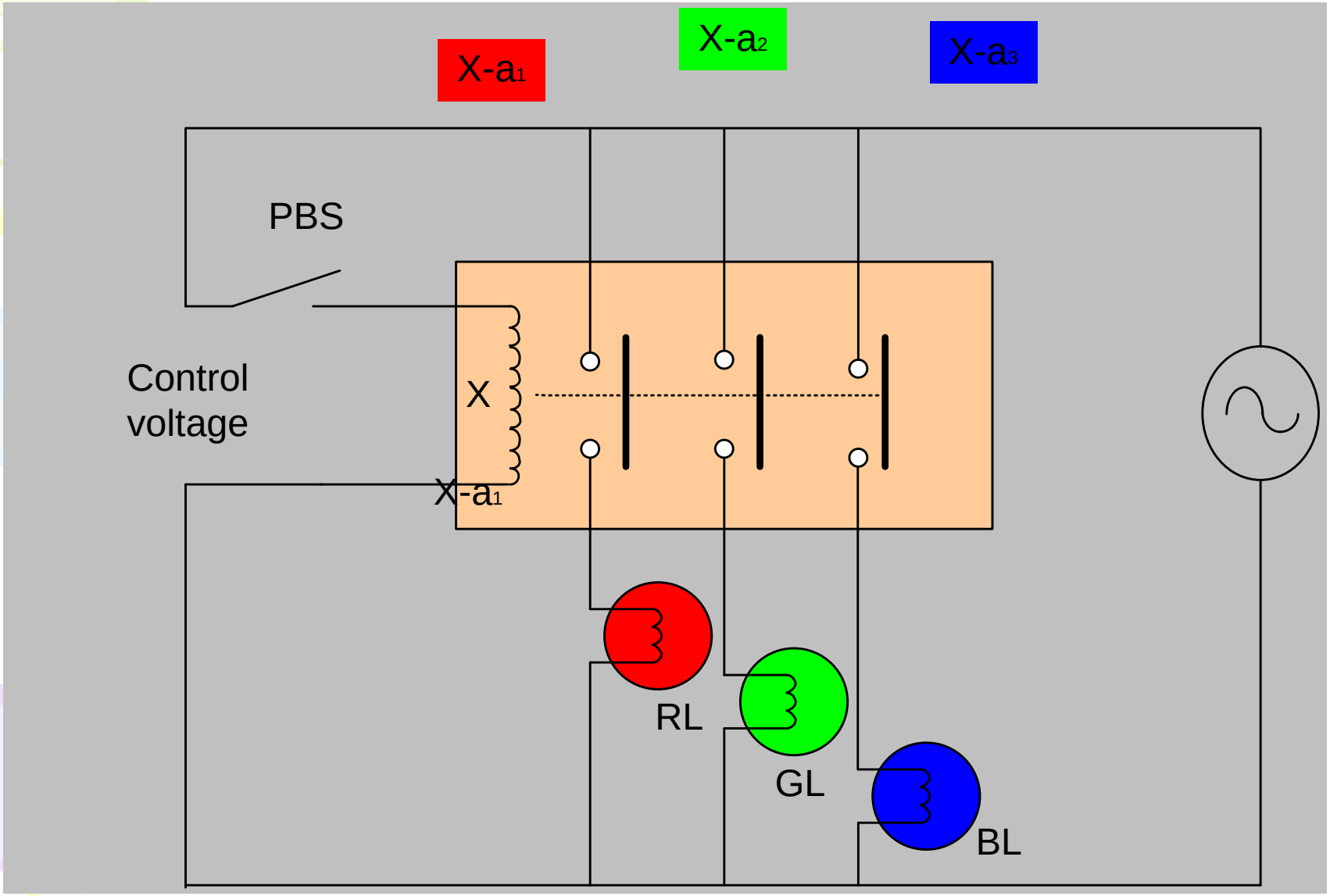

รูปที่ 15.10 ไดอะแกรมตัวอักษร(Alphabetic diagram)

# งาน

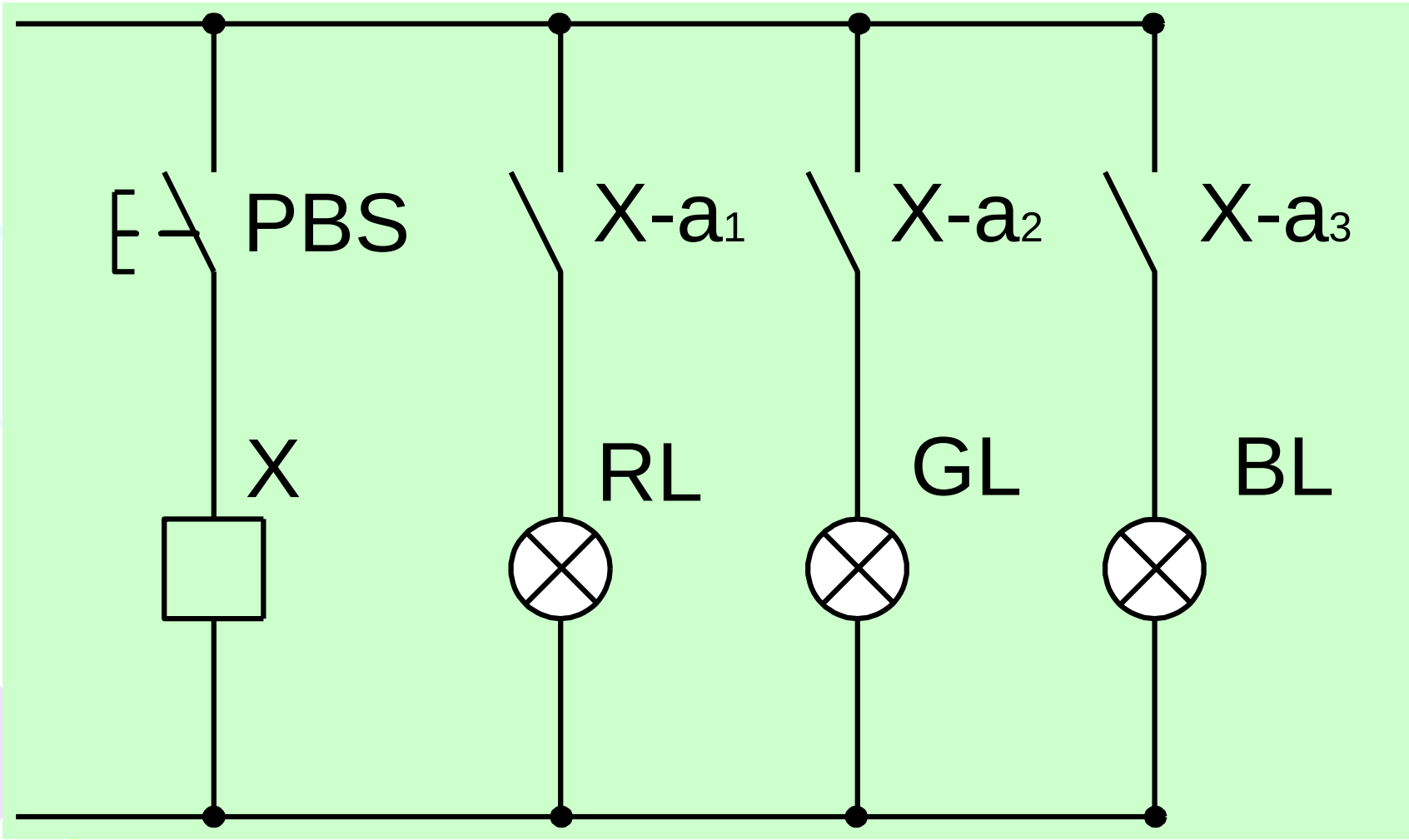
## วงจรเซอร์กิตเบรกเกอร์ที่ใช้กับแรงดันสูง



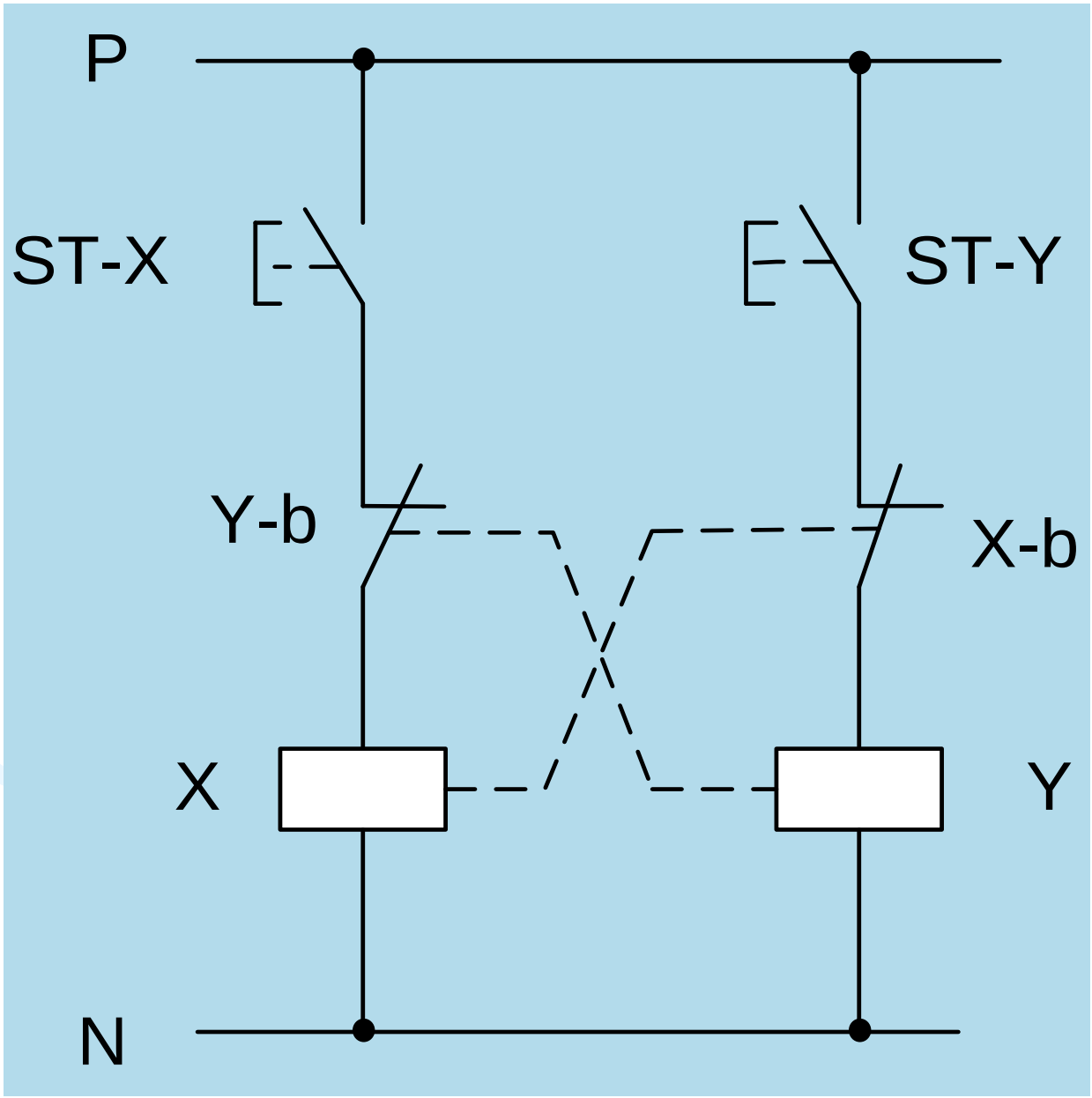
# การทำงานของวงจร Magnetic Relay



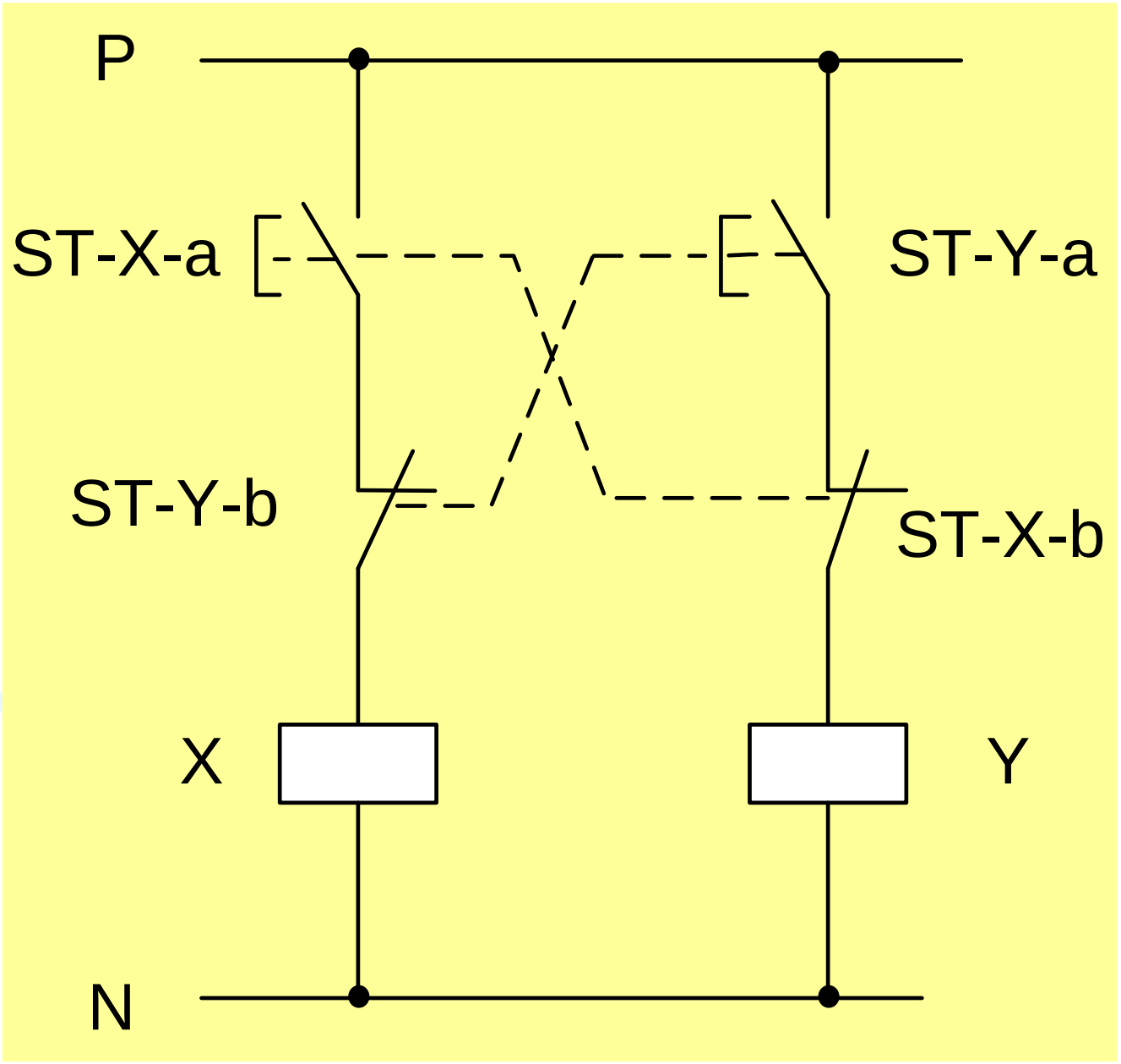
# Circuit Diagram



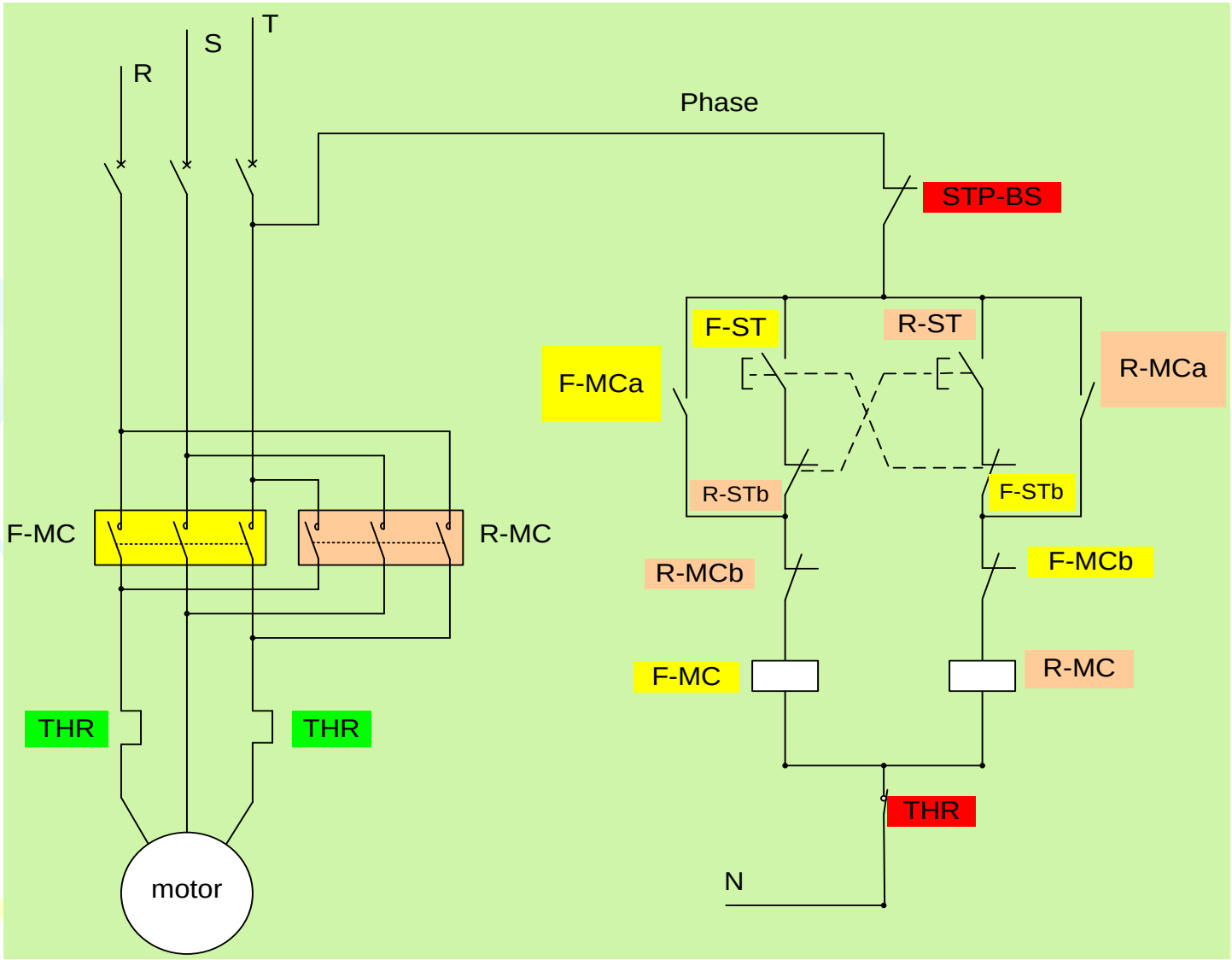
# อินเตอร์ล็อกแบบใช้รีเลย์ควบคุม



# อินเตอร์ล็อกโดยใช้สวิทซ์ควบคุมแบบใช้รีเลย์



# การควบคุมทิศทางของมอเตอร์สามเฟส



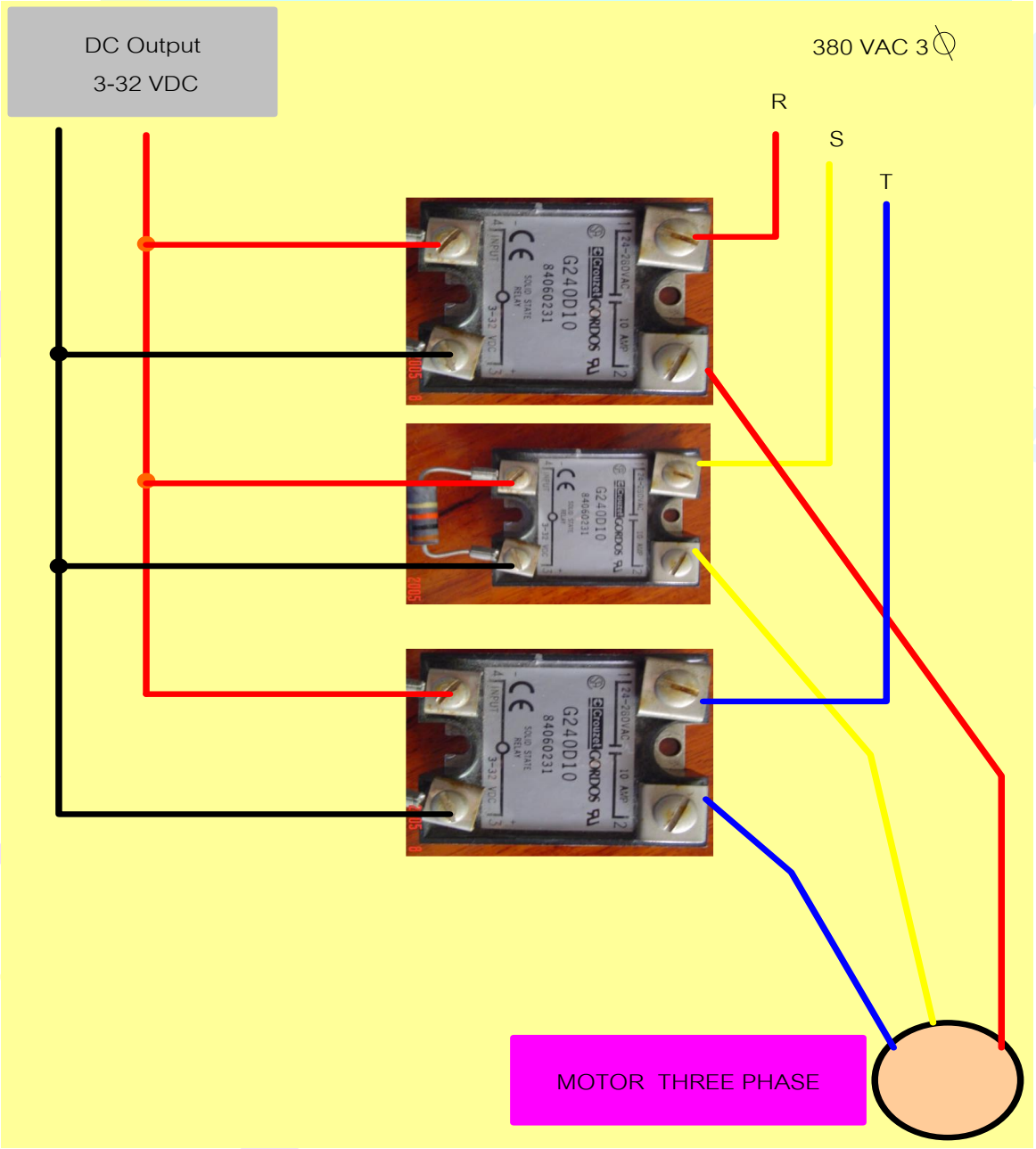
# อุปกรณ์ถ่ายทอดกำลังงานในรูปแบบใช้สารกึ่งตัวนำ

เอาต์พุตทำงานที่แรงดัน 24 ถึง 280 Volts ทาง
กระแสสลับทนกระแสสูงสุดในวงจรถึง 10 Amperes.

Solid State Relay

อินพุตใช้ไฟตรงประมาณ 3-32 Volts ซึ่งนำมาจากเอาต์พุตของวงจรดิจิตอลได้โดยตรง

# PLC ในการควบคุมงานอุตสาหกรรม



# ทดสอบความรู้สัปดาห์ที่15

1. จงอธิบายความแตกต่างระหว่าง PLC แบบ Compact กับแบบ Modular

2. จงอธิบายหน้าที่สำคัญของหน่วยอินพุต และหน่วยเอาต์พุต พร้อมทั้งเขียน

บล็อกไดอะแกรมของหน่วยทั้งสอง

3. หลักในการเขียนโปรแกรมควบคุม PLC มีกี่ขั้นตอน

## ตอนที่3

**ภาคผนวก**

**วิธีสอนและกิจกรรม**

1. สอนโดยบรรยายด้วยสไลด์ ในโปรแกรม Power Point
2. ตั้งข้อสังเกตของเนื้อหาที่จะสอนก่อนคำบรรยาย ซักถามตามคำบรรยายที่ผ่านในหัวข้อที่สำคัญ และสอดแทรกข้อคิดให้กลับไปทบทวนในสิ่งที่ผู้สอนบรรยายมา
3. เนื้อหาที่แบ่งแต่ละสัปดาห์กำหนดให้จบลงเป็นเรื่องๆ เพื่อให้นักศึกษาลำดับความรู้ได้อย่างเป็นระเบียบ และนำไปเป็นพื้นฐานไปใช้ในสัปดาห์ต่อไป
4. การเชื่อมโยงของระบบดิจิตอลที่สอนแต่ละสัปดาห์ เมื่อทำการสอนจนครบทุกสัปดาห์ นักศึกษาจะได้ความรู้และพัฒนาการอย่างเป็นระบบ โดยผู้สอนจะมอบหมายงานให้ทำคำถามท้ายบท
5. ข้อมูลของเนื้อหาที่ใช้ จะนำมาจากหนังสืออ้างอิงที่เขียนกันมากมายแต่จะมีทฤษฎีที่เหมือนกัน ส่วนที่แตกต่างกันอยู่ที่การถ่ายทอดของหนังสือแต่ละเล่ม
6. ให้นักศึกษาค้นคว้าจากสื่ออิเล็กทรอนิกส์ โดยกำหนดให้หาข้อมูลจาก E-Book ของ บริษัท บุ๊ค โปรโมชั่น แอนด์ เซอร์วิส จำกัด ที่ สกอ เป็นผู้จัดหาให้สำหรับมหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ศาลายา

**สื่อการสอน**

**-เอกสารอ้างอิง**


1. Douglas V.Hall Digital circuit and System, MC Graw-Hill Publishing Company, 1989
2. Milos D.Ercegovac, Tomas Lang and Jame H.Morino, Introduction to Digital System, John wiley & sons, Inc 1999.
3. Victor P.Nelson, H.Troy Nagle, Bill D.carrall J.David Trwin, Digital logic circuit analysis and design, Prentice Mall, International edition,1995.
4. M.Morris Mano, Digital design, Third edition, Prentice Mall, International edition 2002.
5. Ronald J.Tocci , Digital System Principles and Applications, Prentice Hall , Fifth Edition, 1991
6. วัฒนา แก้วดุก การออกแบบดิจิตอลวิศวกรรม, ทฤษฎีและการประยุกต์ใช้งาน, สำนักพิมพ์ สสท.(ไทย-ญี่ปุ่น), 2546.
7. ธนะชัย ชัยยุทธ, กณพ แก้วพิชัย ดิจิตอลพื้นฐาน, บริษัทซีเอ็ดยูเคชัน จำกัด 2521
8. ธีรวัฒน์ ประกอบผล ดิจิตอลเทนิค, บริษัทซีเอ็ดยูเคชัน 2546

9. ถวัลย์วงศ์ ไกรโรจนานันท์, อิเล็กทรอนิกส์ระบบดิจิตอล ฉบับปรับปรุง, สวทช. 2542
10. ธวัชชัย เลื่อนฉวี, อนุรักษ์ เถื่อนศิริ, ดิจิตอลเทคนิค เล่ม 1-2 2543, 2532
11. มงคล อาวัชนาการ , โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ ,มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน วิทยาเขตขอนแก่น
12. ศัพท์เทคนิควิศวกรรมไฟฟ้าสื่อสาร สมาคมวิศวกรรมสถานแห่งประเทศไทยในพระบรมราชูปถัมภ์, กันยายน, 2541

**-เอกสารประกอบการสอน**

1. วิชา วงจรดิจิตอลและการออกแบบลอจิก เรียบเรียงโดย นายชัยวัฒน์ จงกุลสถิตชัย อาจารย์ประจำคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ศาลายา

**-โสตทัศนูปกรณ์**

1.แผ่นข้อมูลเนื้อหาแต่ละบทเรียน ทำในรูปแบบPower Point

2.เครื่องคอมพิวเตอร์

3.เครื่องฉายโปรเจคเตอร์

**งานที่มอบหมาย**

1.ให้นักศึกษาทำคำถามท้ายบทเรียนในแต่ละสัปดาห์ที่เรียนผ่านมา และข้อคิดที่ผู้สอนได้สอดแทรก ในการบรรยาย นักศึกษาจะต้องนำข้อคิดกลับไปบูรณาการของเนื้อหา

2. การทำคำถามท้ายบทเรียนกำหนดส่งภายใน 1 สัปดาห์ ในเนื้อหาที่สอนแต่ละสัปดาห์ โดยค้นจากหนังอ้างอิงเพิ่มเติม หรือ E-Book

3.การทำรายงานกลุ่ม กำหนดเนื้อหาตามหน่วยเรียน8 หน่วยเรียน โดยแบ่งเป็น 8 กลุ่มย่อย กำหนดส่งดูความก้าวหน้ากลางภาค และปลายภาค ให้เวลาในการค้นคว้าแบ่งกลุ่มทำงานที่รับมอบหมาย ค้นคว้าหาข้อมูลจากสำนักวิทยบริการ

**กิจกรรม** การทำรายงานกลุ่มให้ทำเป็นแบบรูปเล่ม หรือทำในรูปสื่อ CD และนักศึกษาจะต้องนำมาแสดงในชั้นเรียนเมื่อ เรียนครบเนื้อหา 14 สัปดาห์ โดยนำข้อมูลที่นักศึกษาทำงานเป็นกลุ่มแสดงโดยใช้สื่อแผ่นใส สื่อด้วยโปรแกรมทางคอมพิวเตอร์ และบรรยายเนื้อหาที่ได้ทำรายงานให้เพื่อนร่วมชั้นฟัง

**การวัดผล**

1.สังเกตความสนใจขณะเรียนโดยการถาม-ตอบ

2.จากการทำคำถามท้ายบททุกสัปดาห์

3.จากรายงานที่ทำส่งในรายบุคคล และรายงานกลุ่ม

4. คะแนนจากการสอบกลางภาค

5. คะแนนจากการสอบปลายภาค

# บันทึกการสอน

| บันทึกการสอน | สิ่งที่ควรสังเกตและปรับปรุง |
| --- | --- |
| | |