



เอกสารประกอบการสอน

วงจรรหัสดิจิทัลและลอจิก
EL01303 Digital and Logic Circuits

กฤษณพงศ์ สมสุข

วท.ม.วิทยาการคอมพิวเตอร์
วศ.ม. วิศวกรรมคอมพิวเตอร์
วศ.บ. วิศวกรรมคอมพิวเตอร์ (เกียรตินิยมอันดับ 2)

คณะเทคโนโลยี
มหาวิทยาลัยราชภัฏอุดรธานี
ปีการศึกษา **2556**

คำนำ

เอกสารประกอบการสอนวิชาวงจรถิทัศน์และลอจิก รหัสวิชา **EL01303** มีจำนวนหน่วยกิต คือ **3(3-0-6)** เป็นวิชาที่ปรับปรุงมาจากวิชาวงจรถิทัศน์และลอจิก รหัสวิชา **5582129** เพื่อใช้สำหรับประกอบการเรียนการสอนหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ มหาวิทยาลัยราชภัฏอุดรธานี ซึ่งผู้เรียบเรียงได้จัดทำตามคำอธิบายรายวิชา เนื้อหาของเอกสารประกอบการสอนเล่มนี้เหมาะสำหรับนักศึกษาในระดับปริญญาตรี โดยแบ่งเนื้อหาออกเป็น **13** บทเรียนซึ่งครอบคลุมตามคำอธิบายรายวิชา โดยเริ่มต้นจากปูพื้นฐานเกี่ยวกับสัญญาณดิจิทัล สัญญาณอนาล็อก ความแตกต่างระหว่างสัญญาณดิจิทัลและสัญญาณอนาล็อก และระบบเลขฐานจนถึงการวิเคราะห์และออกแบบวงจรถิทัศน์ โดยบทที่ **13** เป็นบทเพิ่มเติมเกี่ยวกับการวิเคราะห์และออกแบบวงจรโดยใช้ภาษา **VHDL (VHSIC Hardware Description Language)** เบื้องต้นซึ่งเป็นภาษาที่เหมาะสมสำหรับการออกแบบวงจรมีขนาดใหญ่เพื่อให้ผู้เรียนมีความรู้ความเข้าใจในบทเรียน และสามารถมองเห็นภาพของการออกแบบวงจรถิทัศน์เพื่อนำไปประยุกต์ใช้จริงและเพื่อให้มีผลสัมฤทธิ์ที่ดียิ่งขึ้น เอกสารประกอบการสอนเล่มนี้ผู้เรียบเรียงได้ทำการปรับปรุง แก้ไขเนื้อหาและรูปภาพบางส่วนให้เหมาะสมเพิ่มเติม เนื่องจากใช้ประกอบการสอนตั้งแต่ปี พ.ศ. **2553** จนปัจจุบัน ผู้เรียบเรียงขอขอบพระคุณบิดา มารดา อันเป็นที่รักที่เป็นกำลังใจสนับสนุนให้งานสำเร็จไปด้วยดี นอกเหนือจากนั้น ผู้เรียบเรียงขอขอบพระคุณคณาจารย์สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ที่ให้การสนับสนุนมาโดยตลอด และขอขอบพระคุณผู้เขียนหนังสือและตำราที่ได้ใช้ในการอ้างอิง

สุดท้ายผู้เรียบเรียงหวังเป็นอย่างยิ่งว่าเอกสารประกอบการสอนเล่มนี้จะเป็นประโยชน์ต่อการเรียนของนักศึกษาสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ รวมไปถึงสาขาวิชาอื่นที่เกี่ยวข้องไม่มากนัก

กฤษณพงศ์ สมสุข
พฤษภาคม 2556

สารบัญ

| | หน้า |
|--|------|
| คำนำ | (2) |
| สารบัญ | (3) |
| สารบัญรูป | (10) |
| สารบัญตาราง | (13) |
| แผนบริหารการสอนประจำวิชา | (14) |
| แผนบริหารการสอนประจำบทที่ 1 | 1 |
| บทที่ 1 ระบบดิจิทัลเบื้องต้นและระบบเลขฐาน | 3 |
| 1.1 บทนำ | 3 |
| 1.2 ระบบเลขฐาน | 4 |
| 1.3 การแปลงเลขฐาน | 4 |
| 1.3.1 การแปลงเลขฐานสองเป็นเลขฐานสิบ | 4 |
| 1.3.2 การแปลงเลขฐานแปดเป็นเลขฐานสิบ | 6 |
| 1.3.3 การแปลงเลขฐานสิบหกเป็นเลขฐานสิบ | 6 |
| 1.3.4 การแปลงเลขฐานสิบเป็นเลขฐานสอง | 6 |
| 1.3.5 การแปลงเลขฐานสิบเป็นเลขฐานแปด | 7 |
| 1.3.6 การแปลงเลขฐานสิบเป็นเลขฐานสิบหก | 8 |
| 1.3.7 การแปลงเลขฐานสองเป็นเลขฐานแปดและการแปลงจากเลขฐานแปด เป็นเลขฐานสอง | 8 |
| 1.3.8 การแปลงเลขฐานสองเป็นเลขฐานสิบหกและการแปลงจากเลขฐานสิบหก เป็นเลขฐานสอง | 11 |
| 1.4 การบวกเลขฐาน | 14 |
| 1.5 การลบเลขฐาน | 15 |
| 1.6 การลบเลขฐานสองโดยใช้คอมพลิเมนต์ | 17 |
| 1.6.1 การลบเลขฐานสองด้วยวิธีคอมพลิเมนต์แบบ 1 | 18 |
| 1.6.1 การลบเลขฐานสองด้วยวิธีคอมพลิเมนต์แบบ 2 | 19 |
| 1.7 การคูณเลขฐานสอง | 20 |
| 1.8 การหารเลขฐานสอง | 21 |
| 1.9 บทสรุป | 22 |
| คำถามท้ายบท | 23 |
| เอกสารอ้างอิง | 24 |
| แผนบริหารการสอนประจำบทที่ 2 | 25 |
| บทที่ 2 รหัส | 27 |
| 2.1 บทนำ | 27 |
| 2.2 รหัสบีซีดี | 27 |
| 2.3 รหัสเพิ่ม 3 | 30 |
| 2.4 รหัสตรวจสอบความผิดพลาด | 32 |

| | หน้า |
|--|------|
| 2.5 รหัสเกรย์ | 32 |
| 2.5.1 การแปลงจากรหัสเลขฐานสองเป็นรหัสเกรย์ | 34 |
| 2.5.2 การแปลงจากรหัสเกรย์เป็นรหัสเลขฐานสอง | 36 |
| 2.6 รหัสแอสกี | 37 |
| 2.7 บทสรุป | 39 |
| คำถามท้ายบท | 40 |
| เอกสารอ้างอิง | 42 |
| แผนบริหารการสอนประจำบทที่ 3 | 43 |
| บทที่ 3 เกตและไอซีพื้นฐาน | 45 |
| 3.1 บทนำ | 45 |
| 3.2 ตารางความจริง | 45 |
| 3.3 แผนผังเวลา | 45 |
| 3.4 แอนด์เกต | 46 |
| 3.5 ออร์เกต | 48 |
| 3.6 นีตเกต | 49 |
| 3.7 แนนด์เกต | 50 |
| 3.8 นอร์เกต | 52 |
| 3.9 เอ็กคลูซีฟออร์เกต | 54 |
| 3.10 เอ็กคลูซีฟนอร์เกต | 54 |
| 3.11 คุณสมบัติของเกตพื้นฐาน | 55 |
| 3.11.1 ค่าปลอดภัยสัญญาณรบกวน | 55 |
| 3.11.2 ความเร็วในการทำงาน | 56 |
| 3.11.3 การสูญเสียกำลัง | 56 |
| 3.11.4 ความสามารถในการต่อร่วมกัน | 56 |
| 3.11.5 กระแสซิงค์และกระแสซอร์ส | 56 |
| 3.12 ไอซีพื้นฐาน | 56 |
| 3.12.1 ไอซีเบอร์ 7408 | 59 |
| 3.12.2 ไอซีเบอร์ 7432 | 59 |
| 3.12.3 ไอซีเบอร์ 7404 | 60 |
| 3.12.4 ไอซีเบอร์ 7400 | 61 |
| 3.12.5 ไอซีเบอร์ 7402 | 61 |
| 3.12.6 ไอซีเบอร์ 7486 | 62 |
| 3.12.7 ไอซีเบอร์ 74266 | 63 |
| 3.13 บทสรุป | 64 |
| คำถามท้ายบท | 64 |
| เอกสารอ้างอิง | 65 |
| แผนบริหารการสอนประจำบทที่ 4 | 66 |

| | |
|--|------|
| | หน้า |
| บทที่ 4 พืชชนิดตบูลีน | 68 |
| 4.1 บทนำ | 68 |
| 4.2 การสร้างวงจรรจากสมการพืชชนิดตบูลีน | 68 |
| 4.3 การหาสถานะเอาต์พุตโดยใช้ตารางความจริง | 70 |
| 4.4 กฎพื้นฐานพืชชนิดตบูลีน | 70 |
| 4.4.1 กฎการสลับที่ | 71 |
| 4.4.2 กฎการเปลี่ยนกลุ่ม | 71 |
| 4.4.3 กฎการแจกแจง | 71 |
| 4.4.4 กฎไอดีมโปเทน | 72 |
| 4.4.5 กฎนิเสธสองครั้ง | 72 |
| 4.4.6 กฎการแอนด์ | 73 |
| 4.4.7 กฎการออร์ | 73 |
| 4.4.8 กฎคอมพลิเมนต์ | 73 |
| 4.4.9 กฎความซับซ้อน | 74 |
| 4.4.10 กฎเดอมอร์แกน | 74 |
| 4.5 บทสรุป | 75 |
| คำถามท้ายบท | 76 |
| เอกสารอ้างอิง | 77 |
| แผนบริหารการสอนประจำบทที่ 5 | 78 |
| บทที่ 5 วงจรรวมเชิงจัดหมู่ | 79 |
| 5.1 การแก้ไขวงจรให้อยู่ในรูปที่ง่ายขึ้นโดยใช้กฎพืชชนิดตบูลีน | 79 |
| 5.2 การเขียนสมการพืชชนิดตบูลีนจากผลรวมของผลคูณ | 81 |
| 5.3 การเขียนสมการพืชชนิดตบูลีนจากผลคูณของผลรวม | 81 |
| 5.4 การเขียนสมการพืชชนิดตบูลีนจากตารางความจริง | 82 |
| 5.5 การออกแบบวงจรรวมเชิงจัดหมู่ | 83 |
| 5.6 การใช้ยูนิเวอร์แซลเกตในการออกแบบวงจรรวมเชิงจัดหมู่ | 86 |
| 5.7 บทสรุป | 89 |
| คำถามท้ายบท | 90 |
| เอกสารอ้างอิง | 91 |
| แผนบริหารการสอนประจำบทที่ 6 | 92 |
| บทที่ 6 แผนผังคาร์โนห์ | 94 |
| 6.1 รูปแบบแผนผังคาร์โนห์ | 94 |
| 6.1.1 แผนผังคาร์โนห์ขนาด 2 ตัวแปร | 94 |
| 6.1.2 แผนผังคาร์โนห์ขนาด 3 ตัวแปร | 95 |
| 6.1.3 แผนผังคาร์โนห์ขนาด 4 ตัวแปร | 96 |
| 6.2 การอ่านค่าลงแผนผังคาร์โนห์ | 94 |
| 6.3 การสร้างรูป | 97 |

| | หน้า | |
|----------------|--|-----|
| 6.4 | เงื่อนไขที่ไม่สนใจ | 106 |
| 6.5 | การออกแบบวงจรวกและวงจรถเลขฐานสอง | 108 |
| 6.5.1 | วงจรวกแบบไม่คิดตัวท | 109 |
| 6.5.2 | วงจรวกแบบคิดตัวท | 111 |
| 6.5.3 | วงจรถเลขแบบไม่คิดตัวยืม | 115 |
| 6.5.4 | วงจรถเลขแบบคิดตัวยืม | 118 |
| 6.5.5 | วงจรวกแบบหลายบิต | 122 |
| 6.5.6 | วงจรถเลขแบบหลายบิต | 123 |
| 6.6 | บทสรุป | 124 |
| | คำถามท้ายบท | 125 |
| | เอกสารอ้างอิง | 127 |
| | แผนบริหารการสอนประจำบทที่ 7 | 128 |
| บทที่ 7 | ฟลิปฟล็อป | 130 |
| 7.1 | ฟลิปฟล็อปแบบ RS | 132 |
| 7.1.1 | ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ RS | 137 |
| 7.1.2 | การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ RS | 138 |
| 7.2 | ฟลิปฟล็อปแบบ D | 140 |
| 7.2.1 | ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ D | 142 |
| 7.2.2 | การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D | 143 |
| 7.3 | ฟลิปฟล็อปแบบ T | 144 |
| 7.3.1 | ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ T | 146 |
| 7.3.2 | การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ T | 148 |
| 7.4 | ฟลิปฟล็อปแบบ JK | 149 |
| 7.4.1 | ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ JK | 152 |
| 7.4.2 | การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ JK | 153 |
| 7.5 | บทสรุป | 155 |
| | คำถามท้ายบท | 156 |
| | เอกสารอ้างอิง | 159 |
| | แผนบริหารการสอนประจำบทที่ 8 | 160 |
| บทที่ 8 | การออกแบบวงจรถเลขลำดับ | 162 |
| 8.1 | การออกแบบแผนภาพสถานะและตารางสถานะ | 162 |
| 8.1.1 | รูปแบบของเมลลี | 162 |
| 8.1.2 | รูปแบบของมัวร์ | 164 |

| | หน้า |
|--|------|
| 8.2 การวิเคราะห์วงจรเชิงลำดับ | 165 |
| 8.2.1 วิธีการวิเคราะห์วงจรเชิงลำดับแบบเข้าจังหวะ | 166 |
| 8.2.2 วิธีการวิเคราะห์วงจรเชิงลำดับแบบไม่เข้าจังหวะ | 170 |
| 8.3 การออกแบบวงจรเชิงลำดับ | 177 |
| 8.4 การลดสถานะเอาต์พุตของฟลิปฟล็อป | 185 |
| 8.5 บทสรุป | 192 |
| คำถามท้ายบท | 193 |
| เอกสารอ้างอิง | 196 |
| แผนบริหารการสอนประจำบทที่ 9 | 197 |
| บทที่ 9 วงจรนับและชิพทีริจิสเตอร์ | 198 |
| 9.1 การออกแบบแผนภาพสถานะและตารางสถานะ | 198 |
| 9.1.1 วงจรนับเลขฐานสองแบบไม่เข้าจังหวะ | 198 |
| 9.1.2 วงจรนับเลขฐานสองแบบไม่เข้าจังหวะกรณีที่ไม่ลงตัว | 201 |
| 9.1.3 วงจรนับเลขฐานสองแบบเข้าจังหวะ | 203 |
| 9.2 ชิพทีริจิสเตอร์ | 210 |
| 9.2.1 การถ่ายโอนข้อมูลแบบอนุกรม | 210 |
| 9.2.2 การถ่ายโอนข้อมูลแบบขนาน | 211 |
| 9.3 บทสรุป | 212 |
| คำถามท้ายบท | 213 |
| เอกสารอ้างอิง | 214 |
| แผนบริหารการสอนประจำบทที่ 10 | 215 |
| บทที่ 10 วงจรเข้ารหัสและวงจรถอดรหัส | 216 |
| 10.1 วงจรเข้ารหัส | 216 |
| 10.2 วงจรถอดรหัส | 217 |
| 10.3 แอลอีดี 7 ส่วน | 219 |
| 10.4 บทสรุป | 222 |
| คำถามท้ายบท | 223 |
| เอกสารอ้างอิง | 224 |
| แผนบริหารการสอนประจำบทที่ 11 | 225 |
| บทที่ 11 วงจรมัลติเพล็กซ์เซอร์และวงจรมัลติเพล็กซ์เซอร์ | 226 |
| 11.1 วงจรมัลติเพล็กซ์เซอร์ | 226 |
| 11.2 วงจรมัลติเพล็กซ์เซอร์ | 229 |
| 11.3 บทสรุป | 230 |
| คำถามท้ายบท | 231 |
| เอกสารอ้างอิง | 232 |
| แผนบริหารการสอนประจำบทที่ 12 | 233 |

| | หน้า |
|--|------|
| บทที่ 12 วงจรแปลงสัญญาณ | 234 |
| 12.1 ออปแอมป์ | 234 |
| 12.1.1 วงจรเปรียบเทียบ | 234 |
| 12.1.2 อินเวอร์ตติ้งแอมพลิไฟเออร์ | 235 |
| 12.2 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก | 236 |
| 12.2.1 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก | 237 |
| 12.3 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล | 238 |
| 12.3.1 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล | 239 |
| 12.4 บทสรุป | 240 |
| คำถามท้ายบท | 241 |
| เอกสารอ้างอิง | 242 |
| แผนบริหารการสอนประจำบทที่ 13 | 243 |
| บทที่ 13 การออกแบบวงจรเชิงลำดับ | 245 |
| 13.1 บทนำ | 245 |
| 13.1.1 ไลบราลี | 245 |
| 13.1.2 เอ็นดีตี | 246 |
| 13.1.3 อาซิเทคเซอร์ | 247 |
| 13.1.4 แพ็กเกจ | 248 |
| 13.1.5 คอมพิวกเจอร์ชั้น | 249 |
| 13.2 การตั้งชื่อตัวแปร | 249 |
| 13.3 ชนิดข้อมูล | 249 |
| 13.3.1 ชนิดข้อมูล BIT | 250 |
| 13.3.2 ชนิดข้อมูล BIT_VECTOR | 250 |
| 13.3.3 ชนิดข้อมูล STD_LOGIC | 250 |
| 13.3.4 ชนิดข้อมูล STD_LOGIC_VECTOR | 250 |
| 13.3.5 ชนิดข้อมูล INTEGER | 250 |
| 13.3.6 ชนิดข้อมูลอื่น | 250 |
| 13.4 ตัวดำเนินการทางตรรกะ | 251 |
| 13.5 สัญญาณ | 251 |
| 13.6 ค่าคงที่ | 252 |
| 13.7 คำสั่งเชิงลำดับ | 252 |
| 13.8 คำสั่งเงื่อนไข | 253 |
| 13.8.1 คำสั่ง When - Else | 254 |
| 13.8.2 คำสั่ง With - Select | 255 |
| 13.8.3 คำสั่ง If - Else | 256 |
| 13.8.4 คำสั่ง Case - When | 257 |
| 13.9 การสร้างสัญญาณนาฬิกา | 258 |

13.10 บทสรุป
คำถามท้ายบท
เอกสารอ้างอิง

หน้า
258
259
260

สารบัญรูป

| | หน้า |
|--|------|
| รูปที่ 1.1 ตัวอย่างสัญญาณแบบอนาลอก | 3 |
| รูปที่ 1.2 ตัวอย่างสัญญาณแบบดิจิทัล | 3 |
| รูปที่ 3.1 ตัวอย่างแผนผังเวลา | 46 |
| รูปที่ 3.2 สัญลักษณ์ของแอนด์เกต | 47 |
| รูปที่ 3.3 สัญลักษณ์ของออร์เกต | 48 |
| รูปที่ 3.4 สัญลักษณ์ของน็อตเกต | 50 |
| รูปที่ 3.5 สัญลักษณ์ของแนนด์เกต | 50 |
| รูปที่ 3.6 เปรียบเทียบการใช้งานแนนด์เกต | 51 |
| รูปที่ 3.7 สัญลักษณ์ของนอร์เกต | 53 |
| รูปที่ 3.8 เปรียบเทียบการใช้งานนอร์เกต | 53 |
| รูปที่ 3.9 สัญลักษณ์ของเอ็กคลูซีฟออร์เกต | 54 |
| รูปที่ 3.10 สัญลักษณ์ของเอ็กคลูซีฟนอร์เกต | 55 |
| รูปที่ 3.11 ตัวอย่างแบบจำลองไอซี | 56 |
| รูปที่ 3.12 ระดับลอจิกของไอซีตระกูลทีทีแอล | 57 |
| รูปที่ 3.13 ระดับลอจิกของไอซีตระกูลซีมอส | 58 |
| รูปที่ 3.14 ไอซีเบอร์ 7408 | 59 |
| รูปที่ 3.15 ไอซีเบอร์ 7432 | 60 |
| รูปที่ 3.16 ไอซีเบอร์ 7404 | 60 |
| รูปที่ 3.17 ไอซีเบอร์ 7400 | 61 |
| รูปที่ 3.18 ไอซีเบอร์ 7402 | 62 |
| รูปที่ 3.19 ไอซีเบอร์ 7486 | 62 |
| รูปที่ 3.20 ไอซีเบอร์ 74266 | 63 |
| รูปที่ 5.1 ตัวอย่างวงจรรวมเชิงจัดหมู่ | 79 |
| รูปที่ 5.2 วงจร $Z = ABC + ABC + A$ | 80 |
| รูปที่ 5.3 วงจร $Z = AB$ | 80 |
| รูปที่ 6.1 แผนผังคาร์โนห์แบบ 2 อินพุต | 94 |
| รูปที่ 6.2 แผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวนอน | 95 |
| รูปที่ 6.3 แผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวตั้ง | 95 |
| รูปที่ 6.4 แผนผังคาร์โนห์แบบ 4 อินพุต | 96 |
| รูปที่ 6.5 โครงสร้างวงจรวกแบบไม่คิดตัวทด | 109 |
| รูปที่ 6.6 วงจรวกแบบไม่คิดตัวทด | 111 |
| รูปที่ 6.7 โครงสร้างวงจรวกแบบคิดตัวทด | 111 |
| รูปที่ 6.8 วงจรวกแบบคิดตัวทด | 115 |
| รูปที่ 6.9 โครงสร้างวงจรวกแบบไม่คิดตัวยืม | 115 |
| รูปที่ 6.10 วงจรวกแบบไม่คิดตัวยืม | 117 |
| รูปที่ 6.11 โครงสร้างวงจรวกแบบคิดตัวยืม | 118 |

| | หน้า |
|---|------|
| รูปที่ 6.12 วงจรลบบแบบคิดตัวยืม | 122 |
| รูปที่ 7.1 โครงสร้างฟลิปฟล็อป | 130 |
| รูปที่ 7.2 การเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น | 131 |
| รูปที่ 7.3 การเกิดสัญญาณนาฬิกาที่ขอบขาลง | 131 |
| รูปที่ 7.4 โครงสร้างฟลิปฟล็อปที่มีการเปลี่ยนสถานะเอาต์พุตที่ขอบขาลง | 131 |
| รูปที่ 7.5 สัญลักษณ์ของฟลิปฟล็อปแบบ RS | 132 |
| รูปที่ 7.6 โครงสร้างของฟลิปฟล็อปแบบ RS | 133 |
| รูปที่ 7.7 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ RS | 139 |
| รูปที่ 7.8 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ RS | 140 |
| รูปที่ 7.9 สัญลักษณ์ของฟลิปฟล็อปแบบ D | 140 |
| รูปที่ 7.10 การสร้างฟลิปฟล็อปแบบ D โดยใช้ฟลิปฟล็อปแบบ RS | 141 |
| รูปที่ 7.11 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ D | 143 |
| รูปที่ 7.12 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ D | 144 |
| รูปที่ 7.13 สัญลักษณ์ของฟลิปฟล็อปแบบ T | 144 |
| รูปที่ 7.14 การสร้างฟลิปฟล็อปแบบ T โดยใช้ฟลิปฟล็อปแบบ RS | 145 |
| รูปที่ 7.15 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ T | 148 |
| รูปที่ 7.16 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ T | 148 |
| รูปที่ 7.17 สัญลักษณ์ของฟลิปฟล็อปแบบ JK | 149 |
| รูปที่ 7.18 การสร้างฟลิปฟล็อปแบบ JK โดยใช้ฟลิปฟล็อปแบบ RS | 150 |
| รูปที่ 7.19 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ JK | 154 |
| รูปที่ 7.20 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ JK | 154 |
| รูปที่ 8.1 ตัวอย่างแผนภาพสถานะรูปแบบเมลลี | 162 |
| รูปที่ 8.2 ตัวอย่างแผนภาพสถานะรูปแบบมัวร์ | 164 |
| รูปที่ 9.1 ตัวอย่างวงจรนับ 4 แบบนับขึ้น | 198 |
| รูปที่ 9.2 ตัวอย่างวงจรนับ 4 แบบนับขึ้นโดยใช้ฟลิปฟล็อปแบบ JK | 199 |
| รูปที่ 9.3 ตัวอย่างวงจรนับ 4 แบบนับลง | 200 |
| รูปที่ 9.4 ตัวอย่างวงจรนับ 4 แบบนับลงโดยใช้ฟลิปฟล็อปแบบ JK | 200 |
| รูปที่ 9.5 การถ่ายโอนข้อมูลแบบอนุกรม | 210 |

| | หน้า |
|--|------|
| รูปที่ 9.6 การถ่ายโอนข้อมูลแบบขนาน | 211 |
| รูปที่ 10.1 โครงสร้างแอลอีตี 7 ส่วน | 219 |
| รูปที่ 10.2 การใช้งานไอซีเบอร์ 7447A ร่วมกับแอลอีตี 7 ส่วนแบบคอมมอนแอนด | 221 |
| รูปที่ 10.3 การใช้งานไอซีเบอร์ 7448A ร่วมกับแอลอีตี 7 ส่วนแบบคอมมอนแคโทด | 221 |
| รูปที่ 11.1 โครงสร้างวงจรมัลติเพล็กซ์เซอร์แบบ 4 อินพุต | 226 |
| รูปที่ 11.2 โครงสร้างวงจรมัลติเพล็กซ์เซอร์แบบ 4 เอาต์พุต | 229 |
| รูปที่ 12.1 สัญลักษณ์ของออปแอมป์ | 234 |
| รูปที่ 12.2 วงจรเปรียบเทียบที่สถานะเอาต์พุตมีค่าเป็นบวกเท่านั้น | 235 |
| รูปที่ 12.3 อินเวอร์ติงออปแอมป์ | 235 |
| รูปที่ 12.4 ตัวอย่างวงจร D/A ขนาด 4 บิต | 237 |
| รูปที่ 12.5 ตัวอย่างวงจร D/A แบบ R-2R Ladder ขนาด 4 บิต | 238 |
| รูปที่ 12.6 ตัวอย่างวงจร A/D ขนาด 4 บิต | 239 |
| รูปที่ 13.1 โครงสร้างไลบรารี | 245 |
| รูปที่ 13.2 โครงสร้างเอ็นตีตี้ | 246 |
| รูปที่ 13.3 โครงสร้างอเทกเซอร์ | 247 |
| รูปที่ 13.4 โครงสร้างแพ็กเกจ | 248 |
| รูปที่ 13.5 คำสงวนในภาษา VHDL | 249 |
| รูปที่ 13.6 ตัวอย่างสัญญาณ | 251 |
| รูปที่ 13.7 โครงสร้างโปรเซส | 252 |
| รูปที่ 13.8 โครงสร้างคำสั่ง When – Else | 254 |
| รูปที่ 13.9 โครงสร้างคำสั่ง With – Select | 255 |
| รูปที่ 13.10 โครงสร้างคำสั่ง If – Else | 256 |
| รูปที่ 13.11 โครงสร้างคำสั่ง Case – When | 257 |

สารบัญตาราง

| | หน้า |
|---|------|
| ตารางที่ 1.1 ความสัมพันธ์ระหว่างเลขฐานสองและเลขฐานแปด | 10 |
| ตารางที่ 1.2 ความสัมพันธ์ระหว่างเลขฐานสองและเลขฐานสิบหก | 13 |
| ตารางที่ 2.1 การแทนค่าเลขฐานสิบแต่ละตัวด้วยรหัส BCD – 8421 | 27 |
| ตารางที่ 2.2 การเปรียบเทียบการเปลี่ยนแปลงของบิตข้อมูลระหว่างรหัสเกรย์และรหัสเลขฐานสอง | 33 |
| ตารางที่ 2.3 รหัสแอสกีที่ใช้แทนสัญลักษณ์บางส่วนโดยเปรียบเทียบกับเลขฐานสิบหก | 38 |
| ตารางที่ 3.1 ตัวอย่างของตารางความจริง | 45 |
| ตารางที่ 3.2 ตารางความจริงของแอนด์เกต | 46 |
| ตารางที่ 3.3 ตารางความจริงของออร์เกต | 48 |
| ตารางที่ 3.4 ตารางความจริงของน็อตเกต | 49 |
| ตารางที่ 3.5 ตารางความจริงของแนนด์เกต | 50 |
| ตารางที่ 3.6 ตารางความจริงของนอร์เกต | 53 |
| ตารางที่ 3.7 ตารางความจริงของเอ็กคลูซีฟออร์เกต | 54 |
| ตารางที่ 3.8 ตารางความจริงของเอ็กคลูซีฟนอร์เกต | 55 |
| ตารางที่ 4.1 การหาสถานะเอาต์พุตที่เป็นไปได้ทั้งจากสมการ $Z = A(B+C)$ | 70 |
| ตารางที่ 4.2 คุณสมบัติการสลับที่ภายใต้การ ออร์ | 71 |
| ตารางที่ 4.3 คุณสมบัติการเปลี่ยนกลุ่มภายใต้การ ออร์ | 71 |
| ตารางที่ 4.4 คุณสมบัติการแจกแจง | 72 |
| ตารางที่ 4.5 คุณสมบัติ $A+B = \overline{\overline{A}.B}$ | 75 |
| ตารางที่ 4.6 คุณสมบัติ $AB = \overline{\overline{A} + \overline{B}}$ | 75 |
| ตารางที่ 6.1 ตารางความจริงของวงจรวกแบบไม่คิดตัวทด | 110 |
| ตารางที่ 6.2 ตารางความจริงของวงจรวกแบบคิดตัวทด | 114 |
| ตารางที่ 6.3 ตารางความจริงของวงจรถบแบบไม่คิดตัวยืม | 117 |
| ตารางที่ 6.4 ตารางความจริงของวงจรถบแบบคิดตัวยืม | 121 |
| ตารางที่ 7.1 ตารางความจริงของฟลิปฟลอปแบบ RS | 132 |
| ตารางที่ 7.2 ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ RS | 138 |
| ตารางที่ 7.3 ตารางความจริงของฟลิปฟลอปแบบ D | 140 |
| ตารางที่ 7.4 ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ D | 143 |
| ตารางที่ 7.5 ตารางความจริงของฟลิปฟลอปแบบ T | 145 |
| ตารางที่ 7.6 ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ T | 147 |
| ตารางที่ 7.7 ตารางความจริงของฟลิปฟลอปแบบ JK | 149 |
| ตารางที่ 7.8 ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ JK | 153 |
| ตารางที่ 8.1 ตัวอย่างตารางสถานะรูปแบบเมลลี | 163 |
| ตารางที่ 8.2 ตัวอย่างตารางสถานะรูปแบบมัวร์ | 165 |
| ตารางที่ 10.1 การแสดงผลเลข 0-9 ด้วยแอลอีดี 7 ส่วนแบบคอมมอนแคโทด | 220 |
| ตารางที่ 13.1 ตัวดำเนินการทางตรรกะ | 251 |

แผนบริหารการสอนประจำวิชา

รหัสวิชา **EL01303**

ชื่อวิชา **วงจรรหัสและลอจิก (Digital and Logic Circuits)**

3(3-0-6)

คำอธิบายรายวิชา (Course Description)

ศึกษาระบบเชิงเลขเบื้องต้น พีชคณิตบูลีน ระบบเลขฐานสอง สมบัติวงจรถ่วงเกตเชิงเลข โครงสร้างพื้นฐานของเกตเชิงเลข ทีทีแอล และ ซีเอ็มอส การสังเคราะห์วงจรรหัส วงจรบวก วงจรมัลติเพล็กซ์เซอร์ วงจรไต่รหัส วงจรถอดรหัส วงจรเชิงเลขแบบลำดับ แลตซ์ ฟลิปฟลอป วงจรนับ ตัวแปลงผันสัญญาณ เอพูดี และ ดีทูเอ

วัตถุประสงค์ทั่วไป เพื่อให้ผู้เรียนมีความสามารถดังนี้

1. สามารถอธิบายพื้นฐานเกี่ยวกับระบบเลขฐาน การแปลงเลขฐาน และการดำเนินการระหว่างเลขฐานได้
2. สามารถอธิบายรหัสข้อมูลข่าวสารที่ต้องใช้ในระบบดิจิทัล
3. เข้าใจหลักการทำงานของเกต และไอซีพื้นฐานและตารางความจริง
4. สามารถอธิบายพื้นฐานเกี่ยวกับพีชคณิตบูลีน
5. สามารถอธิบายวงจรรวมเชิงรหัส และการลดขนาดของวงจรถ่วงการพีชคณิตบูลีน
6. สามารถอธิบายการใช้แผนผังคาร์โนห์เพื่อลดรูปสมการพีชคณิตบูลีนได้
7. สามารถอธิบายหลักการทำงานของฟลิปฟลอปชนิดต่างๆได้
8. สามารถอธิบายการนำฟลิปฟลอปชนิดต่างๆมาใช้ในการออกแบบวงจรรหัสลำดับได้
9. สามารถอธิบายการนำฟลิปฟลอปชนิดต่างๆมาใช้ในการออกแบบวงจรรหัสและชิฟต์รีจิสเตอร์ได้
10. สามารถอธิบายการทำงานของวงจรรหัสและวงจรถอดรหัสได้
11. สามารถอธิบายการทำงานของวงจรมัลติเพล็กซ์เซอร์และวงจรมัลติเพล็กซ์เซอร์ได้
12. เข้าใจหลักการแปลงสัญญาณแบบสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก และแบบสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล
13. เข้าใจเกี่ยวกับภาษา VHDL ขั้นพื้นฐาน

เนื้อหา

| | | |
|---|---|-----------------|
| บทที่ 1 ระบบดิจิทัลเบื้องต้นและระบบเลขฐาน | 3 | ชั่วโมง |
| บทที่ 2 รหัส | 1 | ชั่วโมง 30 นาที |
| บทที่ 3 เกตและไอซีพื้นฐาน | 1 | ชั่วโมง 30 นาที |
| บทที่ 4 พีชคณิตบูลีน | 3 | ชั่วโมง |
| บทที่ 5 วงจรรวมเชิงรหัส | 3 | ชั่วโมง |

| | | |
|--|---|-----------------|
| บทที่ 6 แผนผังคาร์โนห์ | 3 | ชั่วโมง |
| บทที่ 7 ฟลิปฟลอป | 3 | ชั่วโมง |
| บทที่ 8 การออกแบบวงจรเชิงลำดับ | 6 | ชั่วโมง |
| บทที่ 9 วงจรนับและซีพรีจิสเตอร์ | 3 | ชั่วโมง |
| บทที่ 10 วงจรเข้ารหัสและวงจรถอดรหัส | 1 | ชั่วโมง 30 นาที |
| บทที่ 11 วงจรมัลติเพล็กซ์เซอร์และวงจรมัลติเพล็กซ์เซอร์ | 1 | ชั่วโมง 30 นาที |
| บทที่ 12 วงจรแปลงสัญญาณ | 3 | ชั่วโมง |
| บทที่ 13 ภาษา VHDL เบื้องต้น | 6 | ชั่วโมง |

วิธีการสอนและกิจกรรม

- อธิบายคำอธิบายรายวิชา เนื้อหา กฎระเบียบต่างๆ รวมทั้งตั้งเกณฑ์การให้คะแนนและการประเมินผล
- นำเข้าสู่บทเรียนโดยการบรรยายเนื้อหาและสื่อบนคอมพิวเตอร์
- นำอุปกรณ์ตัวอย่างหรือภาพเคลื่อนไหวของอุปกรณ์ตัวอย่างเสนอเพื่อกระตุ้นความสนใจเรียน
- ระหว่างนำเสนอตัวอย่างหรือภาพเคลื่อนไหวเปิดโอกาสให้ผู้เรียนได้ซักถาม
- แบ่งผู้เรียนออกเป็นกลุ่มย่อยๆ แล้วให้โจทย์เพื่อเลียนแบบการทำงานโดยใช้โปรแกรม **Xilinx**
- ผู้เรียนฝึกต่อปฏิบัติเพื่อพิสูจน์การทำงานของอุปกรณ์และวงจรจริง
- ผู้เรียนสรุปผลและนำเสนอกับอาจารย์และเพื่อนๆ ในชั้นเรียน

สื่อการเรียนการสอน

- ชุดนำเสนอโครงสร้างและรายละเอียดของรายวิชา
- เอกสารประกอบการสอนรายวิชาวงจรดิจิทัล และลอจิก
- โปรแกรม **Xilinx** หรือโปรแกรมจำลองการทำงานอื่นๆ
- คอมพิวเตอร์และโปรแกรมเพาเวอร์พอยต์
- แบบทดสอบย่อยพร้อมเฉลย
- อุปกรณ์และวงจรดิจิทัลตัวอย่างที่น่าสนใจ

การวัดผล

- | | | |
|---------------------|----|-------|
| 1. คะแนนระหว่างภาค | | |
| 1.1 คะแนนการบ้าน | 20 | คะแนน |
| 1.2 คะแนนสอบกลางภาค | 30 | คะแนน |
| 2. คะแนนสอบปลายภาค | 40 | คะแนน |
| 3. เวลาเรียน | 10 | คะแนน |

- ลักษณะข้อสอบเป็นแบบบรรยาย และคำนวณ
- การตัดเกรดเป็นแบบอิงเกณฑ์

การประเมินผล

| | | |
|-------|----------|------------|
| คะแนน | 0 – 49 | ได้เกรด F |
| คะแนน | 50 – 54 | ได้เกรด D |
| คะแนน | 55 – 59 | ได้เกรด D+ |
| คะแนน | 60 – 64 | ได้เกรด C |
| คะแนน | 65 – 69 | ได้เกรด C+ |
| คะแนน | 70 – 74 | ได้เกรด B |
| คะแนน | 75 – 79 | ได้เกรด B+ |
| คะแนน | 80 – 100 | ได้เกรด A |

หนังสืออ่านประกอบ

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

Mark, Z. (2004). **Digital System Design with VHDL**. United Kingdom: Pearson Education.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิตอลอิเล็กทรอนิกส์**.จ. อุดรธานี: มหาวิทยาลัยราชภัฏอุดรธานี.

รัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิตอลเทคนิคเล่ม 1**.กรุงเทพฯ: มิตรนราการพิมพ์.

รัชชัย เลื่อนฉวี. (2532). **ดิจิตอลเทคนิคเล่ม 2**.กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**.กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**.กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

ชำนาญ ปัญญาใส และ วัชรกร หนูทอง. (2547). **ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัดเอช-เอน การพิมพ์.

ณรงค์ ทองฉิม และ เจริญ วงษ์ชุ่มเย็น. (2552). ออกแบบไอซีดิจิทัลด้วย **FPGA** และ **CPLD** ภาคนิพนธ์โดยใช้ภาษา **VHDL** ซอฟต์แวร์ชุด **ISE WebPack**. กรุงเทพฯ: บริษัท วี.พี.พี. (1991) จำกัด.

แผนการสอนรายสัปดาห์

รหัสวิชา EL01303 ชื่อวิชา วงจรรีจิสเตอร์และลอจิก
 ภาคเรียนที่ 1 ปีการศึกษา 2557

| ครั้งที่ | เรื่อง | กิจกรรม/การบ้าน/อื่นๆ |
|-------------------|---|--|
| 1 | บทนำเกี่ยวกับระบบดิจิทัล ระบบเลขฐาน การแปลงเลขฐาน การบวก ลบ คูณ และหารเลขฐาน และการลบเลขฐานด้วยวิธี Complement | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 2 | รหัสฐานสองรูปแบบต่างๆ เช่นรหัส บีซีดี รหัสเกิน 3 รหัสเกรย์ หรือ รหัสตรวจสอบความผิดพลาด เป็นต้น แนะนำเกต และไอซีพื้นฐาน | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 3 | การสร้างวงจรรวมจากสมการพีชคณิตบูลีน การหาสถานะเอาต์พุตโดยใช้ตารางความจริง และกฎพื้นฐานพีชคณิตบูลีน | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 4 | การแก้ไขวงจรรวมให้อยู่ในรูปที่ง่ายขึ้นโดยใช้กฎพีชคณิตบูลีน การเขียนสมการพีชคณิตบูลีนจากผลรวมของผลคูณ การเขียนสมการพีชคณิตบูลีนจากผลคูณของผลรวม การเขียนสมการพีชคณิตบูลีนจากตารางความจริง และการออกแบบวงจรรวมเชิงจัดหมู่ | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 5 | การลดรูปวงจรรวมโดยใช้แผนผังคาร์โนห์ สถานะที่ไม่สนใจ วงจรบวก ลบ เลขฐานสอง | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 6 | นิยาม และคุณสมบัติของฟลิปฟล็อป 4 ชนิด ประกอบด้วย ฟลิปฟล็อปแบบ RS ฟลิปฟล็อปแบบ D ฟลิปฟล็อปแบบ T และฟลิปฟล็อปแบบ JK | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 7 | การวิเคราะห์วงจรเชิงลำดับ แผนภาพสถานะ และตารางสถานะ | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| สอบกลางภาค | | |
| 8 | การออกแบบวงจรรวมเชิงลำดับ และการลดสถานะเอาต์พุตของฟลิปฟล็อป | ค้นคว้าและทำรายงานหรือการบ้านตามที่ได้รับมอบหมาย |
| 9 | การออกแบบวงจรรวมแบบไม่เข้าจังหวะ และวงจรรวม | ค้นคว้าและทำรายงานหรือ |

| ครั้งที่ | เรื่อง | กิจกรรม/การบ้าน/อื่นๆ |
|-------------------|--|---|
| | แบบเข้าจังหวะ และการออกแบบชิพรีจิสเตอร์ | การบ้านตามที่ได้รับมอบหมาย |
| 10 | วงจรถ่ายรหัส วงจรถอดรหัส วงจรมัลติเพล็กซ์เซอร์ และ วงจรดีมัลติเพล็กซ์เซอร์ | ค้นคว้าและทำรายงานหรือ การบ้านตามที่ได้รับมอบหมาย |
| 11 | วงจรถ่ายสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก และ วงจรถ่ายสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล | ค้นคว้าและทำรายงานหรือ การบ้านตามที่ได้รับมอบหมาย |
| 12 | แนะนำภาษา VHDL โครงสร้างภาษา VHDL ชนิด ข้อมูลตัวดำเนินการสัญญาณ และค่าคงที่ที่ใช้ในภาษา VHDL | ค้นคว้าและทำรายงานหรือ การบ้านตามที่ได้รับมอบหมาย |
| 13 | การเขียนคำสั่งเชิงลำดับและคำสั่งเงื่อนไข ด้วยภาษา VHDL | ค้นคว้าและทำรายงานหรือ การบ้านตามที่ได้รับมอบหมาย |
| 14 | ทบทวนเนื้อหาที่เรียนทั้งหมด | - |
| สอบปลายภาค | | |

หมายเหตุ วิชาบรรยายสอนครบ 14 ครั้ง

แผนบริหารการสอนประจำบทที่ 1 ระบบดิจิตัลเบื้องต้นและระบบเลขฐาน 3 ชั่วโมง

หัวข้อเนื้อหา

1.1 บทนำ

1.2 ระบบเลขฐาน

1.3 การแปลงเลขฐาน

1.3.1 การแปลงเลขฐานสองเป็นเลขฐานสิบ

1.3.2 การแปลงเลขฐานแปดเป็นเลขฐานสิบ

1.3.3 การแปลงเลขฐานสิบหกเป็นเลขฐานสิบ

1.3.4 การแปลงเลขฐานสิบเป็นเลขฐานสอง

1.3.5 การแปลงเลขฐานสิบเป็นเลขฐานแปด

1.3.6 การแปลงเลขฐานสิบเป็นเลขฐานสิบหก

1.3.7 การแปลงเลขฐานสองเป็นเลขฐานแปดและการแปลงเลขฐานแปดเป็นเลขฐานสอง

1.3.8 การแปลงเลขฐานสองเป็นเลขฐานสิบหกและการแปลงเลขฐานสิบหกเป็นเลขฐานสอง

1.4 การบวกเลขฐาน

1.5 การลบเลขฐาน

1.6 การลบเลขฐานสองโดยใช้คอมพลิเมนต์

1.6.1 การลบเลขฐานสองแบบ 1's Complement

1.6.2 การลบเลขฐานสองแบบ 2's Complement

1.7 การคูณเลขฐานสอง

1.8 การหารเลขฐานสอง

1.9 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนเข้าใจพื้นฐานสัญญาณดิจิตัล และความแตกต่างระหว่างสัญญาณดิจิตัลและสัญญาณอนาลอก
2. เพื่อให้ผู้เรียนมีความเข้าใจเกี่ยวกับระบบเลขฐานต่างๆ
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการแปลงเลขฐาน
4. เพื่อให้ผู้เรียนสามารถคำนวณการบวก ลบ คูณ และหารเลขฐานได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย

6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเคลื่อนไหว

การวัดผลและการประเมิน

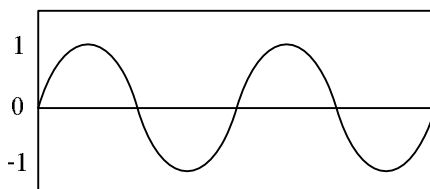
1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 1

ระบบดิจิทัลเบื้องต้นและระบบเลขฐาน

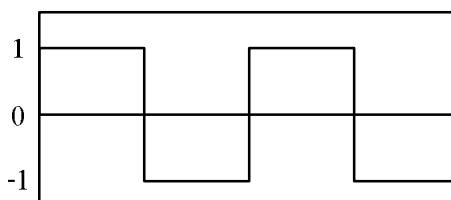
1.1 บทนำ

ปัจจุบันอุปกรณ์อิเล็กทรอนิกส์ที่ใช้สัญญาณแบบดิจิทัลกลายเป็นสิ่งสำคัญที่ขาดไม่ได้สำหรับการดำรงชีวิตของมนุษย์ ตัวอย่างเช่น วงจรที่ใช้สำหรับ เปิด - ปิด สวิตช์ไฟบ้าน นาฬิกาที่เป็นแบบตัวเลขดิจิทัล สัญญาณไฟจราจร หรือ เครื่องคิดเลข เป็นต้น โดยสัญญาณที่เป็นแบบดิจิทัลนั้นจะเป็นสัญญาณแบบไม่ต่อเนื่อง (**Discrete Signal**) คือระดับของสัญญาณในแต่ละระดับจะมีความแตกต่างกันในระดับที่แน่นอน ซึ่งจะแตกต่างกับสัญญาณที่เป็นแบบอนาลอกที่เป็นสัญญาณแบบต่อเนื่อง (**Continuous Signal**) ซึ่งมีขนาดของสัญญาณไม่คงที่ โดยสัญญาณจะมีการเพิ่ม หรือลดลงแบบต่อเนื่อง อุปกรณ์อิเล็กทรอนิกส์บางชนิดถูกพัฒนาออกมาเป็นทั้งแบบระบบอนาลอก และระบบดิจิทัล เช่น วงจรที่ใช้สำหรับควบคุมความสว่างของแสงไฟ กรณีที่เป็นวงจรแบบอนาลอก ผู้ใช้งานสามารถปรับความสว่างของแสงไฟให้เพิ่ม หรือ ลด ได้อย่างต่อเนื่องตามต้องการ แต่หากเป็นวงจรแบบดิจิทัลที่เป็นแบบ 2 ระดับ จะมีเพียงแค่ 2 สถานะเท่านั้น คือสถานะที่ไฟติด และ สถานะที่ไฟดับ



รูปที่ 1.1 ตัวอย่างสัญญาณแบบอนาลอก

รูปที่ 1.1 แสดงตัวอย่างลักษณะของสัญญาณแบบอนาลอก ซึ่งสัญญาณจะมีความต่อเนื่อง โดยค่าของสัญญาณจะอยู่ในช่วง $[-1, 1]$



รูปที่ 1.2 ตัวอย่างสัญญาณแบบดิจิทัล

รูปที่ 1.2 แสดงลักษณะของสัญญาณแบบดิจิทัลซึ่งสัญญาณจะเป็นแบบไม่ต่อเนื่อง โดยจากรูปค่าของสัญญาณจะมีอยู่เพียง 2 ระดับเท่านั้นคือ -1 และ 1

1.2 ระบบเลขฐาน

ระบบดิจิทัลจะมีการประมวลผลคำสั่งทุกคำสั่งในรูปแบบของรหัสต่าง ๆ ที่อยู่ในรูปแบบรหัสเลขฐานสองเท่านั้น แต่เลขฐานที่มนุษย์ใช้กันอยู่ในชีวิตประจำวันคือเลขฐานสิบ ดังนั้นระบบเลขฐานจึงเป็นสิ่งที่สำคัญที่จะช่วยให้การสื่อสารระหว่างมนุษย์ และวงจรนั้นง่ายมากยิ่งขึ้น ในบทนี้จะกล่าวถึงระบบเลขฐาน และการนำเลขฐานมาใช้ในการคำนวณ

กำหนดให้ n เป็นจำนวนเต็มบวก กล่าวได้ว่าระบบเลขฐาน n จะมีสมาชิกทั้งหมดคือ n ตัวประกอบไปด้วย $0, 1, 2, \dots, n-1$ เช่น

ระบบเลขฐาน 2 จะมีสมาชิกทั้งหมด 2 ตัว ประกอบไปด้วย 0 และ 1

ระบบเลขฐาน 3 จะมีสมาชิกทั้งหมด 3 ตัวประกอบไปด้วย 0, 1 และ 2

ระบบเลขฐาน 10 จะมีสมาชิกทั้งหมด 10 ตัวประกอบไปด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8 และ 9

ระบบเลขฐาน 16 จะมีสมาชิกทั้งหมด 16 ตัวประกอบไปด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E และ F

สำหรับระบบเลขฐาน 16 จะสังเกตว่าเลข 2 หลักตั้งแต่ 10 - 15 จะถูกแทนด้วย A - F สาเหตุที่ต้องกำหนดใช้ตัวอักษรมาแทนเนื่องจากป้องกันตัวเลขที่ซ้ำกันในหลักที่ 2 เช่นหากทำการนับเลขฐาน 16 จะพบค่าหลังจาก F จะเป็น 10 ซึ่งหากไม่แทนค่า 10 (ที่นับต่อจากค่า 9) ด้วยตัวอักษร A จะพบว่าตัวเลขซ้ำกันทั้งๆที่มีค่าไม่เท่ากัน

เพื่อให้สามารถบอกได้ว่าค่าจำนวนต่างๆ เป็นเลขฐานอะไร จำเป็นต้องเขียนเลขฐานห้อยไว้ที่ท้ายของจำนวนดังกล่าวด้วย เช่น 108 ของเลขฐาน 10 เขียนได้เป็น 108_{10} , 1011 ของเลขฐานสอง เขียนได้เป็น 1011_2 เป็นต้น แต่โดยปกติแล้วเลขฐานสิบจะไม่นิยมเขียนเลขห้อยไว้ เพราะฉะนั้นหากจำนวนที่กล่าวถึงไม่มีตัวห้อยต่อท้ายให้ทราบได้ทันทีว่าเป็นเลขฐานสิบ

1.3 การแปลงเลขฐาน

เนื่องจากบางครั้งอาจมีความจำเป็นต้องมีการเปลี่ยนแปลงเลขฐาน เช่น การแปลงจากเลขฐานสิบเป็นเลขฐานสอง เพื่อให้เหมาะสมต่อการใช้งานที่ง่ายยิ่งขึ้น ในหัวข้อนี้จะขอกกล่าวถึงกระบวนการแปลงเลขฐานที่สำคัญดังนี้ การแปลงจากเลขฐานสอง ฐานแปด หรือฐานสิบหกเป็นเลขฐานสิบ การแปลงเลขฐานสิบเป็นเลขฐานสอง เลขฐานแปด หรือเลขฐานสิบหก และการแปลงเลขฐานที่ไม่เกี่ยวข้องกับเลขฐานสิบ เช่น การแปลงจากเลขฐานสอง เป็น เลขฐานแปด เป็นต้น

1.3.1 การแปลงเลขฐานสองเป็นเลขฐานสิบ

หลักการคือให้นำค่าเลขฐานสองมาเรียงจากลำดับความสำคัญมากที่สุดไปยังเลขลำดับความสำคัญน้อยที่สุด โดยบิตที่มีลำดับความสำคัญน้อยที่สุดให้คุณด้วย 2^0 บิตที่มีลำดับความสำคัญอันดับต่อมาจะคูณด้วย 2^1 เรียงขึ้นไปจนกระทั่งบิตที่มีลำดับความสำคัญมากที่สุดจะคูณด้วย $2^{\text{bit-length} - 1}$ โดยที่ **bit length** คือจำนวนบิตทั้งหมดของเลขฐานสองที่ต้องการแปลงเป็นเลขฐานสิบ และให้นำผลลัพธ์ที่ได้จากการคูณกันในแต่ละบิตทั้งหมดมาบวกกัน ผลลัพธ์สุดท้ายจะเป็นคำตอบที่อยู่ในรูปแบบของเลขฐานสิบ

ตัวอย่างที่ 1-1 จงเปลี่ยน 101101_2 เป็นเลขฐานสิบ

วิธีทำ

| | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|
| ตัวคูณ | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| เลขประจำตำแหน่ง | 1 | 0 | 1 | 1 | 0 | 1 |

$$\begin{aligned} 101101_2 &= (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 32 + 0 + 8 + 4 + 0 + 1 \\ &= 45_{10} \end{aligned}$$

ตัวอย่างที่ 1-2 จงเปลี่ยน 10110111_2 เป็นเลขฐานสิบ

วิธีทำ

| | | | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|-------|-------|
| ตัวคูณ | 2^7 | 2^6 | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| เลขประจำตำแหน่ง | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

$$\begin{aligned} 10110111_2 &= (1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 128 + 0 + 32 + 16 + 0 + 4 + 2 + 1 \\ &= 183_{10} \end{aligned}$$

สำหรับกรณีที่แปลงตำแหน่งบิตของเลขฐานสองที่เป็นทศนิยม เป็นเลขฐานสิบให้นำบิตทศนิยมที่อยู่ซ้ายสุดคูณด้วย 2^{-1} บิตต่อมาจะคูณด้วย 2^{-2} เรียงไปเช่นนี้จนถึงบิตสุดท้ายที่อยู่ทางขวาสุดของทศนิยม และนำผลลัพธ์ที่ได้ทั้งหมดมาบวกกัน ผลลัพธ์สุดท้ายจะเป็นคำตอบที่อยู่ในรูปแบบของเลขฐานสิบ

ตัวอย่างที่ 1-3 จงเปลี่ยน 0.101_2 เป็นเลขฐานสิบ

วิธีทำ

| | | | |
|-----------------|----------|----------|----------|
| ตัวคูณ | 2^{-1} | 2^{-2} | 2^{-3} |
| เลขประจำตำแหน่ง | 1 | 0 | 1 |

$$\begin{aligned} 0.101_2 &= (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) \\ &= 0.5 + 0 + 0.125 \\ &= 0.625_{10} \end{aligned}$$

1.3.2 การแปลงเลขฐานแปดเป็นเลขฐานสิบ

หลักการจะคล้ายกับการแปลงเลขฐานจากเลขฐานสองเป็นเลขฐานสิบ เพียงแต่ต่างกันตรงที่ตัวคูณให้ใช้เป็นเลข 8

ตัวอย่างที่ 1-4 จงเปลี่ยน 237_8 เป็นเลขฐานสิบ
วิธีทำ

| | | | |
|-----------------|-------|-------|-------|
| ตัวคูณ | 8^2 | 8^1 | 8^0 |
| เลขประจำตำแหน่ง | 2 | 3 | 7 |

$$\begin{aligned} 237_8 &= (2 \times 8^2) + (3 \times 8^1) + (7 \times 8^0) \\ &= 128 + 24 + 7 \\ &= 159_{10} \end{aligned}$$

1.3.3 การแปลงเลขฐานสิบหกเป็นเลขฐานสิบ

หลักการจะคล้ายกับการแปลงเลขฐานจากเลขฐานสองเป็นเลขฐานสิบ เพียงแต่ต่างกันตรงที่ตัวคูณให้ใช้เป็นเลข 16

ตัวอย่างที่ 1-5 จงเปลี่ยน $A4E_{16}$ เป็นเลขฐานสิบ
วิธีทำ

| | | | |
|-----------------|--------|--------|--------|
| ตัวคูณ | 16^2 | 16^1 | 16^0 |
| เลขประจำตำแหน่ง | 10 (A) | 4 | 14 (E) |

$$\begin{aligned} A4E_{16} &= (10 \times 16^2) + (4 \times 16^1) + (14 \times 16^0) \\ &= 2560 + 64 + 14 \\ &= 2638_{10} \end{aligned}$$

1.3.4 การแปลงเลขฐานสิบเป็นเลขฐานสอง

หลักการคือให้นำค่าเลขฐานสิบมาหารด้วยสอง โดยเก็บเศษที่ได้จากการหารไว้ และนำผลลัพธ์ที่ได้จากการหารมาหารด้วยสองอีกครั้ง และเก็บเศษที่ได้จากการหารไว้ ทำเช่นนี้ไปเรื่อย ๆ จนกระทั่งผลลัพธ์ที่ได้จากการหารมีค่าเป็นศูนย์ ให้นำเศษทั้งหมดมาเรียงจากบิตที่มีลำดับความสำคัญสูงสุดไปหาบิตที่มีลำดับความสำคัญต่ำที่สุด โดยเศษที่ได้จากการหารเลขตั้งต้น (เลขฐานสิบที่โจทย์กำหนด) มีลำดับความสำคัญต่ำที่สุด และเศษที่ได้จากการหารครั้งสุดท้าย (ผลลัพธ์ที่ได้จากการหารมีค่าเป็นศูนย์) มีลำดับความสำคัญสูงสุด

ตัวอย่างที่ 1-6 จงเปลี่ยน 43_{10} เป็นเลขฐานสอง

วิธีทำ

$$43/2 = 21 \quad \text{เศษ} \quad 1 \quad \rightarrow \quad (\text{ลำดับความสำคัญต่ำที่สุด})$$

$$21/2 = 10 \quad \text{เศษ} \quad 1$$

$$10/2 = 5 \quad \text{เศษ} \quad 0$$

$$5/2 = 2 \quad \text{เศษ} \quad 1$$

$$2/2 = 1 \quad \text{เศษ} \quad 0$$

$$1/2 = 0 \quad \text{เศษ} \quad 1 \quad \rightarrow \quad (\text{ลำดับความสำคัญสูงที่สุด})$$

เพราะฉะนั้น $43_{10} = 101011_2$

สำหรับกรณีที่เลขฐานสิบเป็นเลขทศนิยมจะมีรูปแบบการแปลงเป็นเลขฐานสองที่แตกต่างกับรูปแบบจำนวนเต็ม โดยการแปลงเลขทศนิยมของเลขฐานสิบเป็นเลขฐานสองสามารถทำได้โดยการนำเลขทศนิยมของเลขฐานสิบมาคูณด้วยสองแล้วตรวจสอบผลลัพธ์ว่ามีค่าทศนิยมเป็นศูนย์หรือไม่ หากยังไม่เป็นศูนย์ให้นำเฉพาะเลขทศนิยมของผลลัพธ์ที่ได้มาคูณด้วยสอง ทำเช่นนี้เรื่อยๆ จนกระทั่งค่าทศนิยมมีค่าเป็นศูนย์ เมื่อทศนิยมมีค่าเป็นศูนย์แล้วให้นำเฉพาะเลขจำนวนเต็มที่เป็นผลลัพธ์มาตอบเรียงจากบิตที่มีลำดับความสำคัญสูงที่สุดไปยังบิตที่มีลำดับความสำคัญต่ำที่สุด บิตที่มีลำดับความสำคัญสูงที่สุดจะเป็นจำนวนเต็มที่ได้จากผลคูณในครั้งแรก และบิตที่มีลำดับความสำคัญต่ำที่สุดจะเป็นจำนวนเต็มที่ได้จากผลคูณในครั้งสุดท้าย ดังนั้นผลลัพธ์ที่เป็นจำนวนเต็มที่ได้จากการคำนวณครั้งแรกจะเป็นเลขทศนิยมตำแหน่งแรก และผลลัพธ์ที่เป็นจำนวนเต็มที่ได้จากการคำนวณครั้งสุดท้ายจะเป็นเลขทศนิยมตำแหน่งสุดท้าย

ตัวอย่างที่ 1-7 จงเปลี่ยน 0.625_{10} เป็นเลขฐานสอง

วิธีทำ

$$0.625 \times \qquad \qquad \qquad 0.250 \times \qquad \qquad \qquad 0.50 \times$$

$$\begin{array}{r} \underline{\quad 2} \\ 0.625 \\ \hline 1.250 \end{array} \qquad \qquad \qquad \begin{array}{r} \underline{\quad 2} \\ 0.250 \\ \hline 0.500 \end{array} \qquad \qquad \qquad \begin{array}{r} \underline{\quad 2} \\ 0.50 \\ \hline 1.00 \end{array}$$

เพราะฉะนั้น $0.75_{10} = 0.101_2$

1.3.5 การแปลงเลขฐานสิบเป็นเลขฐานแปด

หลักการจะคล้ายกับการแปลงเลขฐานจากเลขฐานสิบเป็นเลขฐานสองเพียงแต่ต่างกันตรงที่ตัวหารให้ใช้เป็นเลข 8

ตัวอย่างที่ 1-8 จงเปลี่ยน 137_{10} เป็นเลขฐานแปด

วิธีทำ

$$137/8 = 17 \quad \text{เศษ} \quad 1 \quad \rightarrow \quad (\text{ลำดับความสำคัญต่ำที่สุด})$$

$$17/8 = 2 \quad \text{เศษ} \quad 1$$

$$2/8 = 0 \quad \text{เศษ} \quad 2 \quad \rightarrow \quad (\text{ลำดับความสำคัญสูงที่สุด})$$

เพราะฉะนั้น $137_{10} = 211_8$

1.3.6 การแปลงเลขฐานสิบเป็นเลขฐานสิบหก

หลักการจะคล้ายกับการแปลงเลขฐานจากเลขฐานสิบเป็นเลขฐานสอง เพียงแต่ต่างกันตรงที่ตัวหารให้ใช้เป็นเลข 16

ตัวอย่างที่ 1-9 จงเปลี่ยน 405_{10} เป็นเลขฐานสิบหก

วิธีทำ

$$\begin{aligned} 411/16 &= 25 \quad \text{เศษ} \quad 11 \quad (\text{แทนด้วย B}) \quad \text{--> (ลำดับความสำคัญต่ำที่สุด)} \\ 25/16 &= 1 \quad \text{เศษ} \quad 9 \\ 1/16 &= 0 \quad \text{เศษ} \quad 1 \quad \text{--> (ลำดับความสำคัญสูงที่สุด)} \\ \text{เพราะฉะนั้น} \quad 411_{10} &= 19B_{16} \end{aligned}$$

1.3.7 การแปลงเลขฐานสองเป็นเลขฐานแปดและการแปลงเลขฐานแปดเป็นเลขฐานสอง

การแปลงเลขฐานสองเป็นเลขฐานแปด และการแปลงเลขฐานแปดเป็นเลขฐานสองแบ่งออกเป็น 2 วิธี ดังนี้

วิธีที่ 1

กรณีที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานแปด: ให้แปลงจากเลขฐานสองเป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นเลขฐานแปด

กรณีที่ 2 แปลงจากเลขฐานแปดเป็นเลขฐานสอง: ให้แปลงจากเลขฐานแปดเป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นเลขฐานสอง

ตัวอย่างที่ 1-10 จงเปลี่ยน 11101_2 เป็นเลขฐานแปด

วิธีทำ

ขั้นตอนที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานสิบ

| ตัวคูณ | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
|-----------------|-------|-------|-------|-------|-------|
| เลขประจำตำแหน่ง | 1 | 1 | 1 | 0 | 1 |

$$\begin{aligned} 11101_2 &= (1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 16 + 8 + 4 + 0 + 1 \\ &= 29_{10} \end{aligned}$$

ขั้นตอนที่ 2 แปลงจากเลขฐานสิบเป็นเลขฐานแปด

$$\begin{aligned} 29/8 &= 3 \quad \text{เศษ} \quad 5 \quad \text{--> (ลำดับความสำคัญต่ำที่สุด)} \\ 3/8 &= 0 \quad \text{เศษ} \quad 3 \quad \text{--> (ลำดับความสำคัญสูงที่สุด)} \\ \text{เพราะฉะนั้น} \quad 11101_2 &= 35_8 \end{aligned}$$

ตัวอย่างที่ 1-11 จงเปลี่ยน 741_8 เป็นเลขฐานสอง

วิธีทำ

ขั้นตอนที่ 1 แปลงจากเลขฐานแปดเป็นเลขฐานสิบ

| | | | |
|-----------------|-------|-------|-------|
| ตัวคูณ | 8^2 | 8^1 | 8^0 |
| เลขประจำตำแหน่ง | 7 | 4 | 1 |

$$\begin{aligned} 741_8 &= (7 \times 8^2) + (4 \times 8^1) + (1 \times 8^0) \\ &= 448 + 32 + 1 \\ &= 481_{10} \end{aligned}$$

ขั้นตอนที่ 2 แปลงจากเลขฐานสิบเป็นเลขฐานสอง

$$\begin{array}{rcll} 481/2 = & 240 & \text{เศษ} & 1 & \text{-->} & (\text{ลำดับความสำคัญต่ำที่สุด}) \\ 240/2 = & 120 & \text{เศษ} & 0 & & \\ 120/2 = & 60 & \text{เศษ} & 0 & & \\ 60/2 = & 30 & \text{เศษ} & 0 & & \\ 30/2 = & 15 & \text{เศษ} & 0 & & \\ 15/2 = & 7 & \text{เศษ} & 1 & & \\ 7/2 = & 3 & \text{เศษ} & 1 & & \\ 3/2 = & 1 & \text{เศษ} & 1 & & \\ 1/2 = & 0 & \text{เศษ} & 1 & \text{-->} & (\text{ลำดับความสำคัญสูงที่สุด}) \end{array}$$

เพราะฉะนั้น $741_8 = 111100001_2$

วิธีที่ 2 (วิธีลัด)

หลักการคือให้แทนเลขฐานแปดแต่ละหลักด้วยเลขฐานสองขนาด 3 บิต เช่น 0_8 แทนด้วย 000_2 , 1_8 แทนด้วย 001_2 เป็นต้น และพิจารณาดังนี้

กรณีที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานแปด: ให้แบ่งเลขฐานสองเป็นกลุ่มๆ ละ 3 บิตโดยเริ่มนับจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปหาตำแหน่งที่มีลำดับความสำคัญสูงที่สุด หากกลุ่มสุดท้าย (กลุ่มของบิตที่มีลำดับความสำคัญสูงที่สุด) มีจำนวนไม่ครบ 3 บิตให้เพิ่มบิต 0 ไปข้างหน้าให้ครบจำนวน 3 บิต และให้แทนเลขแต่ละกลุ่มด้วยเลขฐานแปดที่มีค่าตรงกัน

กรณีที่ 2 แปลงจากเลขฐานแปดเป็นเลขฐานสอง: ให้แทนเลขแต่ละหลักของเลขฐานแปดด้วยเลขฐานสองขนาด 3 บิต

ตารางที่ 1.1 ความสัมพันธ์ระหว่างเลขฐานสองและเลขฐานแปด

| เลขฐานแปด | เลขฐานสอง |
|-----------|-----------|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

ตัวอย่างที่ 1-12 จงเปลี่ยน 11101101_2 เป็นเลขฐานแปด

วิธีทำ

ขั้นตอนที่ 1 แบ่งเลขฐานสองเป็นกลุ่ม ๆ ละ 3 บิตโดยแบ่งจากบิตที่มีลำดับความสำคัญต่ำสุดไปหาบิตที่มีลำดับความสำคัญสูงสุดคือ 11101101 ได้ดังนี้

กลุ่มที่ 1 $\rightarrow 101$ กลุ่มที่ 2 $\rightarrow 101$ กลุ่มที่ 3 $\rightarrow 011$

สังเกตว่ากลุ่มที่ 3 จะมีสมาชิกเพียง 2 บิตคือ 11 จึงต้องเพิ่ม 0 ที่ตำแหน่งหน้าสุดอีก 1 บิตเพื่อให้ครบ 3 บิต

ขั้นตอนที่ 2 แทนเลขแต่ละกลุ่มด้วยเลขฐานแปดที่มีค่าตรงกัน

จากตารางที่ 1.1 ได้ผลดังนี้

กลุ่มที่ 1 101 แทนด้วย 5

กลุ่มที่ 2 101 แทนด้วย 5

กลุ่มที่ 3 011 แทนด้วย 3

เพราะฉะนั้น $11101101_2 = 355_8$

ตัวอย่างที่ 1-13 จงเปลี่ยน 7423_8 เป็นเลขฐานสอง

วิธีทำ

จากตารางที่ 1.1 ได้ผลดังนี้

3 แทนด้วย 011

2 แทนด้วย 010

4 แทนด้วย 100

7 แทนด้วย 111

เพราะฉะนั้น $7423_8 = 111100010011_2$

สำหรับกรณีการแปลงเลขทศนิยมจะใช้หลักการเดียวกับเลขจำนวนเต็มแต่จะแตกต่างกันตรงที่ให้เริ่มนับจากบิตที่มีตำแหน่งติดกับทศนิยมมากที่สุดก่อนและนับไปจนถึงตำแหน่งสุดท้ายที่มีค่าเป็น 1

และอยู่ไกลจากจุดทศนิยมมากที่สุดโดยหากกลุ่มสุดท้ายมีจำนวนไม่ครบ 3 บิต ให้เพิ่มบิต 0 ต่อท้ายจนครบ 3 บิต

ตัวอย่างที่ 1-14 จงเปลี่ยน 0.1101101_2 เป็นเลขฐานแปด

วิธีทำ

ขั้นตอนที่ 1 แบ่งเลขฐานสองเป็นกลุ่ม ๆ ละ 3 บิต โดยแบ่งจากบิตที่อยู่ติดกับจุดทศนิยมมากที่สุดไปหาบิตที่มีอยู่ไกลจากทศนิยมมากที่สุดคือ 0.1101101 ได้ดังนี้

กลุ่มที่ 1 -> 110 กลุ่มที่ 2 -> 110 กลุ่มที่ 3 -> 100

สังเกตว่ากลุ่มที่ 3 จะมีสมาชิกเพียง 1 บิตคือ 1 จึงต้องเพิ่ม 0 ที่ตำแหน่งหลังสุดอีก 2 บิตเพื่อให้ครบ 3 บิต

ขั้นตอนที่ 2 แทนเลขแต่ละกลุ่มด้วยเลขฐานแปดที่มีค่าตรงกัน

จากตารางที่ 1.1 ได้ผลดังนี้

กลุ่มที่ 1 110 แทนด้วย 6

กลุ่มที่ 2 110 แทนด้วย 6

กลุ่มที่ 3 100 แทนด้วย 4

เพราะฉะนั้น $0.1101101_2 = 0.664_8$

1.3.8 การแปลงเลขฐานสองเป็นเลขฐานสิบหกและการแปลงเลขฐานสิบหกเป็นเลขฐานสอง

การแปลงเลขฐานสองเป็นเลขฐานสิบหกและการแปลงเลขฐานสิบหกเป็นเลขฐานสองมีหลักการคล้าย ๆ กับการแปลงเลขฐานสองเป็นเลขฐานแปดและการแปลงเลขฐานแปดเป็นเลขฐานสอง ต่างกันเพียงแค่เลขฐานสองจะต้องมีขนาด 4 บิตแบ่งออกเป็น 2 วิธี ดังนี้

วิธีที่ 1

กรณีที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานสิบหก: ให้แปลงจากเลขฐานสองเป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นเลขฐานสิบหก

กรณีที่ 2 แปลงจากเลขฐานสิบหกเป็นเลขฐานสอง: ให้แปลงจากเลขฐานสิบหกเป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นเลขฐานสอง

ตัวอย่างที่ 1-15 จงเปลี่ยน 101011_2 เป็นเลขฐานสิบหก

วิธีทำ

ขั้นตอนที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานสิบ

| | | | | | | |
|-----------------|-------|-------|-------|-------|-------|-------|
| ตัวคูณ | 2^5 | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
| เลขประจำตำแหน่ง | 1 | 0 | 1 | 0 | 1 | 1 |

$$\begin{aligned}
 101011_2 &= (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\
 &= 32 + 0 + 8 + 0 + 2 + 1 \\
 &= 43_{10}
 \end{aligned}$$

ขั้นตอนที่ 2 แปลงจากเลขฐานสิบเป็นเลขฐานสิบหก

$$43/16 = 2 \text{ เศษ } 11 \text{ (แทนด้วย B)} \rightarrow \text{(ลำดับความสำคัญต่ำที่สุด)}$$

$$2/16 = 0 \text{ เศษ } 2 \rightarrow \text{(ลำดับความสำคัญสูงที่สุด)}$$

เพราะฉะนั้น $101011_2 = 2B_{16}$

ตัวอย่างที่ 1-16 จงเปลี่ยน $1AC_{16}$ เป็นเลขฐานสอง

วิธีทำ

ขั้นตอนที่ 1 แปลงจากเลขฐานสิบหกเป็นเลขฐานสิบ

| ตัวคูณ | 16^2 | 16^1 | 16^0 |
|-----------------|--------|-------------------|-------------------|
| เลขประจำตำแหน่ง | 1 | A (แทนด้วย 10) | C (แทนด้วย 12) |

$$\begin{aligned} 1AC_{16} &= (1 \times 16^2) + (10 \times 16^1) + (12 \times 16^0) \\ &= 256 + 160 + 12 \\ &= 428_{10} \end{aligned}$$

ขั้นตอนที่ 2 แปลงจากเลขฐานสิบเป็นเลขฐานสอง

$$428/2 = 214 \text{ เศษ } 0 \rightarrow \text{(ลำดับความสำคัญต่ำที่สุด)}$$

$$214/2 = 107 \text{ เศษ } 0$$

$$107/2 = 53 \text{ เศษ } 1$$

$$53/2 = 26 \text{ เศษ } 1$$

$$26/2 = 13 \text{ เศษ } 0$$

$$13/2 = 6 \text{ เศษ } 1$$

$$6/2 = 3 \text{ เศษ } 0$$

$$3/2 = 1 \text{ เศษ } 1$$

$$1/2 = 0 \text{ เศษ } 1 \rightarrow \text{(ลำดับความสำคัญสูงที่สุด)}$$

เพราะฉะนั้น $1AC_{16} = 110101100_2$

วิธีที่ 2 (วิธีลัด)

หลักการคือให้ แทนเลขฐานสิบหกแต่ละหลักด้วยเลขฐานสองขนาด 4 บิต เช่น 0_{16} แทนด้วย 0000_2 , 1_{16} แทนด้วย 0001_2 เป็นต้น และพิจารณาดังนี้

กรณีที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานสิบหก: ให้แบ่งเลขฐานสองเป็นกลุ่ม ๆ ละ 4 บิตโดยเริ่มนับจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปหาตำแหน่งที่มีลำดับความสำคัญสูงที่สุด หากกลุ่มสุดท้าย (กลุ่มของบิตที่มีลำดับความสำคัญสูงที่สุด) มีจำนวนไม่ครบ 4 บิตให้เพิ่มบิต 0 ไปข้างหน้าให้ครบจำนวน 4 บิต และให้แทนเลขแต่ละกลุ่มด้วยเลขฐานสิบหกที่มีค่าตรงกัน

กรณีที่ 2 แปลงจากเลขฐานสิบหกเป็นเลขฐานสอง: ให้แทนเลขแต่ละหลักของเลขฐานสิบหกด้วยเลขฐานสองขนาด 4 บิต

ตารางที่ **1.2** ความสัมพันธ์ระหว่างเลขฐานสองและเลขฐานสิบหก

| เลขฐานสิบหก | เลขฐานสอง |
|-------------|-----------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

ตัวอย่างที่ **1-17** จงเปลี่ยน 111101101_2 เป็นเลขฐานสิบหก

วิธีทำ

ขั้นตอนที่ 1 แบ่งเลขฐานสองเป็นกลุ่มๆละ 4 บิต โดยแบ่งจากบิตที่มีลำดับความสำคัญต่ำสุดไปหาบิตที่มีลำดับความสำคัญสูงสุดคือ 111101101 ได้ดังนี้

กลุ่มที่ 1 -> 1101 กลุ่มที่ 2 -> 1110 กลุ่มที่ 3 -> 0001

สังเกตว่ากลุ่มที่ 3 จะมีสมาชิกเพียง 1 บิต คือ 1 จึงต้องเพิ่ม 0 ที่ตำแหน่งหน้าสุดอีก 3 บิต เพื่อให้ครบ 4 บิต

ขั้นตอนที่ 2 แทนเลขแต่ละกลุ่มด้วยเลขฐานสิบหกที่มีค่าตรงกัน

จากตารางที่ 1.2 ได้ผลดังนี้

กลุ่มที่ 1 1101 แทนด้วย D

กลุ่มที่ 2 1110 แทนด้วย E

กลุ่มที่ 3 0001 แทนด้วย 1

เพราะฉะนั้น $111101101_2 = 1ED_{16}$

ตัวอย่างที่ **1-18** จงเปลี่ยน $1AC_{16}$ เป็นเลขฐานสอง

วิธีทำ

จากตารางที่ 1.2 ได้ผลดังนี้

C แทนด้วย 1100

A แทนด้วย 1010

1 แทนด้วย 0001

เพราะฉะนั้น $1AC_{16} = 110101100_2$

1.4 การบวกเลขฐาน

หลักการบวกเลขฐานจะคล้ายกับการบวกเลขที่มีการใช้งานกันทั่วไป เพียงแต่การบวกเลขที่ใช้กันทั่วไปนั้นจะเป็นการบวกเลขฐานสิบ ซึ่งการบวกเลขฐานอื่น ๆ จะแตกต่างกับการบวกเลขฐานสิบตรงที่ผลลัพธ์ของการบวกในแต่ละหลักเท่านั้น เพื่อเพิ่มความเข้าใจจึงขออธิบายการบวกเลขฐานสิบดังตัวอย่างต่อไปนี้

$$\begin{array}{r} 184 \\ + \\ \hline 245 \\ \hline 429 \end{array}$$

วิธีคิด

1. การบวกจะเริ่มจากการบวกตำแหน่งที่มีลำดับความสำคัญต่ำสุดก่อน (ตำแหน่งที่ 0) และเรียงไปจนถึงตำแหน่งบิตที่มีลำดับความสำคัญสูงสุด จากตัวอย่างตำแหน่งที่ 0 คือ $4 + 5$ ได้ผลลัพธ์คือ 9

2. บวกบิตที่อยู่ตำแหน่งถัดไป (ตำแหน่งที่ 1) คือ $8 + 4$ ได้ผลลัพธ์คือ 12 กรณีนี้จะเห็นว่าผลลัพธ์ของการบวกมากกว่าหรือเท่ากับ 10 เพราะฉะนั้นให้นำผลลัพธ์ดังกล่าวมาลบด้วย 10 ได้เป็น $12 - 10$ ได้ผลลัพธ์เป็น 2 โดยที่ค่า 2 นี้จะเป็นผลลัพธ์ที่ได้ในการบวกของตำแหน่งนี้ (ตำแหน่งที่ 1) และจะมีตัวทดไปตำแหน่งถัดไป (ตำแหน่งที่ 2) อีก 1

3. บวกบิตที่อยู่ตำแหน่งที่ 2 คือ $1 + 2$ ได้ผลลัพธ์เป็น 3 แต่เนื่องจากมีตัวทดเข้ามาที่ตำแหน่งนี้ด้วยจึงต้องบวกค่าของตัวทอนี้ด้วยได้เป็น $3 + 1$ ได้ผลลัพธ์เป็น 4

$$\text{เพราะฉะนั้น } 184_{10} + 245_{10} = 429_{10}$$

จากตัวอย่างนี้หากเปลี่ยนเป็นการบวกเลขฐานอื่น เช่น เลขฐานสอง กรณีที่ตำแหน่งบิตไหนที่มีผลลัพธ์การบวกมากกว่าหรือเท่ากับ 2 ให้นำผลลัพธ์ที่ได้มาลบด้วย 2 ผลลัพธ์ใหม่ที่ได้จะเป็นผลลัพธ์ของตำแหน่งดังกล่าวและต้องมีตัวทอไปยังตำแหน่งถัดไป การบวกเลขฐานแปด และเลขฐานสิบหกมีหลักการพิจารณาเช่นเดียวกันโดยกรณีของการบวกเกินจะเป็นการบวกเกิน 8 และ บวกเกิน 16 ตามลำดับ

ตัวอย่างที่ 1-19 จงหาผลบวกของเลขฐานต่อไปนี้ $11101_2 + 1001_2$

วิธีทำ

$$\begin{array}{r} 11101 \\ + \\ \hline 1001 \\ \hline 100110 \end{array}$$

วิธีคิด

1. การบวกในบิตที่ 0 (บิตที่มีลำดับความสำคัญต่ำสุด): $1 + 1 = 2$ พบว่าผลลัพธ์ที่ได้มีค่ามากกว่าหรือเท่ากับ 2 จึงนำผลลัพธ์ดังกล่าวมาลบออกด้วย 2 ได้ผลลัพธ์ใหม่เป็น $2 - 2 = 0$ แต่มีตัวทดไปบิตที่ 1

2. การบวกในบิตที่ 1: เนื่องจากมีตัวทดเข้ามาจากบิตที่ 0 จึงต้องนำตัวทอนี้มาบวกด้วย $0 + 0 + 1$ (ตัวทด) = 1

3. การบวกในบิตที่ 2: $1 + 0 = 1$

4. การบวกในบิตที่ 3: $1 + 1 = 2$ พบว่าผลลัพธ์ที่ได้มีค่ามากกว่าหรือเท่ากับ 2 จึงนำผลลัพธ์ดังกล่าวมาลบออกด้วย 2 ได้ผลลัพธ์ใหม่เป็น $2 - 2 = 0$ แต่มีตัวทดไปบิตที่ 4

5. การบวกในบิตที่ 4: เนื่องจาก 1001_2 มีเพียง 4 บิต เพราะฉะนั้นการบวกของบิตที่ 4 นี้จึงเป็นการบวกของตัวตั้ง และตัวทด คือ $1 + 1 = 2$ พบว่าผลลัพธ์ที่ได้มีค่ามากกว่าหรือเท่ากับ 2 จึงนำผลลัพธ์ดังกล่าวมาลบออกด้วย 2 ได้ผลลัพธ์ใหม่เป็น $2 - 2 = 0$ แต่มีตัวทดไปบิตที่ 5

6. การบวกในบิตที่ 5: เนื่องจากค่าที่นำมาบวกกันคือ $11101_2 + 1001_2$ มีเพียง 5 บิต และ 4 บิตตามลำดับ เพราะฉะนั้นบิตที่ 5 นี้จึงมีเพียงค่าของตัวทอดเท่านั้นคือ 1

เพราะฉะนั้น $11101_2 + 1001_2 = 100110_2$

ตัวอย่างที่ 1-20 จงหาผลบวกของเลขฐานต่อไปนี้อย่าง $632_8 + 61_8$

วิธีทำ

$$\begin{array}{r} 632 \\ + \\ \underline{61} \\ 713 \end{array}$$

เพราะฉะนั้น $632_8 + 61_8 = 713_8$

ตัวอย่างที่ 1-21 จงหาผลบวกของเลขฐานต่อไปนี้อย่าง $2CA_{16} + 74_{16}$

วิธีทำ

$$\begin{array}{r} 2CA \\ + \\ \underline{74} \\ 33E \end{array} \quad \begin{array}{l} (A \text{ คือ } 10) \\ \\ (E \text{ คือ } 14) \end{array}$$

เพราะฉะนั้น $2CA_{16} + 74_{16} = 33E_{16}$

1.5 การลบเลขฐาน

หลักการลบเลขฐาน จะคล้ายกับการลบเลขที่มีการใช้งานกันทั่วไป เพียงแต่การลบเลขที่ใช้กันนั้นจะเป็นการลบเลขฐานสิบ ซึ่งการลบเลขฐานอื่นๆ จะแตกต่างกับการลบเลขฐานสิบตรงที่ผลลัพธ์ของการลบในแต่ละหลักเท่านั้น เพื่อเพิ่มความเข้าใจจึงขออธิบายการลบเลขฐานสิบก่อนดังตัวอย่างต่อไปนี้

736

-

374362**วิธีคิด**

1. การลบจะเริ่มจากการลบตำแหน่งที่มีลำดับความสำคัญต่ำสุดก่อน และเรียงไปจนถึงตำแหน่งบิตที่มีลำดับความสำคัญสูงสุด จากตัวอย่างตำแหน่งแรกพบว่าตัวตั้งมีค่ามากกว่าตัวลบจึงลบได้แบบปกติ คือ $6 - 4$ ได้ผลลัพธ์คือ 2

2. ลบบิตที่อยู่ตำแหน่งที่ 1 คือ 3 - 7 กรณีนี้เห็นว่าตัวตั้งมีค่าน้อยกว่าตัวลบซึ่งไม่สามารถลบได้ จึงต้องไปขอยืมจากบิตถัดไป (บิตที่ 2) ซึ่งในกรณีนี้เป็นการลบเลขฐานสิบเพราะฉะนั้นค่าที่ยืมมาจึงมีค่าเท่ากับ 10 ซึ่งเมื่อนำมารวมกับค่าที่มีอยู่แล้วคือ 3 จะได้ค่าใหม่คือ 13 หลังจากนั้นจึงนำค่านี้อลบกับตัวลบคือ $13 - 7$ ได้ผลลัพธ์คือ 6

3. ลบบิตที่อยู่ตำแหน่งที่ 2 แต่เนื่องจากตัวตั้งถูกบิตที่ 1 ยืมค่าไป 1 เพราะฉะนั้นตัวตั้งจึงเหลือค่าเท่ากับ 6 โดยกรณีนี้พบว่าตัวตั้งยังมีค่ามากกว่าตัวลบจึงลบได้แบบปกติ $6 - 3$ ได้ผลลัพธ์เป็น 3

$$\text{เพราะฉะนั้น } 736_{10} - 374_{10} = 362_{10}$$

จากตัวอย่างนี้หากเปลี่ยนเป็นการลบเลขฐานอื่น เช่น เลขฐานสอง กรณีที่ตำแหน่งบิตที่ตัวตั้งมีค่าน้อยกว่าตัวลบ ให้ยืมค่ามาจากบิตถัดไป โดยค่าที่ยืมมาจะมีค่าเท่ากับ 2 เนื่องจากเป็นการลบเลขฐานสอง และนำค่าดังกล่าวนี้มารวมกับค่าของตัวตั้งที่มีอยู่แล้ว และนำมอลบกับตัวลบ การลบเลขฐานแปด และเลขฐานสิบหกมีหลักการพิจารณาเช่นเดียวกันโดยกรณีที่ตัวตั้งมีค่าน้อยกว่าตัวลบจะเป็นยืมค่าจากบิตถัดไปมา 8 และ 16 ตามลำดับ

ตัวอย่างที่ 1-22 จงหาผลลบของเลขฐานต่อไปนี้ $11101_2 - 1011_2$

วิธีทำ

11101

-

101110010**วิธีคิด**

1. การลบในบิตที่ 0 (บิตที่มีลำดับความสำคัญต่ำสุด): $1 - 1 = 0$

2. การลบในบิตที่ 1: เนื่องจากมีตัวตั้งมีค่าน้อยกว่าตัวลบ จึงจำเป็นต้องไปขอยืมค่าจากบิตที่ 2 ซึ่งยืมมา 2 รวมกับค่าที่มีอยู่เดิมคือ 0 ได้ค่าใหม่คือ $2 + 0 = 2$ และนำไปลบด้วยตัวลบได้ผลลัพธ์คือ $2 - 1 = 1$

3. การลบในบิตที่ 2: เนื่องจากตัวตั้งถูกบิตที่ 1 ยืมค่าไป 1 จึงเหลือค่าเป็น 0 ซึ่งมีค่าเท่ากับตัวลบคือ 0 สามารถลบกันได้ตามปกติได้ผลลัพธ์เป็น $0 - 0 = 0$

4. การลบในบิตที่ 3: $1 - 1 = 0$

5. การลบในบิตที่ 4: เนื่องจากตัวลบมีเพียง 4 บิต และตัวตั้งไม่ได้ถูกยืมค่าเพราะฉะนั้นผลลัพธ์ที่ได้คือค่าของบิตของตัวตั้งคือ 1

$$\text{เพราะฉะนั้น } 11101_2 - 1011_2 = 10010_2$$

ตัวอย่างที่ 1-23 จงหาผลลบของเลขฐานต่อไปนี $631_8 - 63_8$

วิธีทำ

$$\begin{array}{r} 631 \\ - 63 \\ \hline 546 \end{array}$$

$$\text{เพราะฉะนั้น } 631_8 - 63_8 = 546_8$$

ตัวอย่างที่ 1-24 จงหาผลลบของเลขฐานต่อไปนี $12B_{16} - AC_{16}$

วิธีทำ

$$\begin{array}{r} 12B \\ - AC \\ \hline 7E \end{array} \quad \begin{array}{l} (B \text{ คือ } 11) \\ (A \text{ คือ } 10, C \text{ คือ } 12) \\ (F \text{ คือ } 15) \end{array}$$

$$\text{เพราะฉะนั้น } 12B_{16} - AC_{16} = 7E_{16}$$

1.6 การลบเลขฐานสองโดยใช้คอมพลิเมนต์

เนื่องจากหากมีการบวก หรือลบเลขฐานจำเป็นต้องมี 2 วงจรคือวงจรวกเลข และวงจรถลเลข โดยจริง ๆ แล้วนั้นสามารถใช้เพียงวงจรวกเลขเพียงวงจรถลเลขได้ โดยหากจะนำมาใช้ในการลบกันก็ใช้คอมพลิเมนต์ (complement) แทน เช่น แทนที่จะหาค่า $8 - 3$ ก็ให้หาเป็น $8 + (-3)$ แทน เป็นต้น โดยคอมพลิเมนต์แบ่งออกเป็น 2 แบบคือ 1's Complement และ 2's Complement ซึ่งทั้ง 2 วิธีจะมีวิธีการหาเป็นดังนี้

หลักการหา 1's Complement

นำเลขฐานสองที่ได้มาทำการกลับบิตสำหรับทุก ๆ บิต โดยบิตที่มีค่าเป็น 0 ให้เปลี่ยนเป็น 1 และบิตที่มีค่าเป็น 1 ให้เปลี่ยนเป็น 0

ตัวอย่างที่ 1-25 จงหา 1's Complement ของเลขฐานสองต่อไปนี 10001_2

วิธีทำ

$$\begin{array}{r} 10001 \\ \text{1's Complement } 01110 \end{array}$$

เพราะฉะนั้น 1's Complement ของ 10001_2 คือ 01110_2

หลักการหา 2's Complement

หา 1's Complement ของเลขฐานสองที่ต้องการ แล้วนำผลลัพธ์ที่ได้มาบวกด้วย 1

ตัวอย่างที่ 1-26 จงหา 2's Complement ของเลขฐานสองต่อไปนี้ 10111_2

วิธีทำ

$$\begin{array}{r}
 10110 \\
 1's\ Complement\ 01001 \\
 + \\
 \hline
 1 \\
 2's\ Complement\ 01010
 \end{array}$$

เพราะฉะนั้น 2's Complement ของ 10110_2 คือ 01010_2 **1.6.1 การลบเลขฐานสองด้วยวิธีคอมพลิเมนต์แบบ 1**

การลบเลขฐานสองด้วยวิธีคอมพลิเมนต์แบบ 1 (1's Complement) มีหลักการดังนี้

1. ตรวจสอบจำนวนบิตตัวตั้ง และจำนวนบิตของตัวลบก่อนโดยหากจำนวนบิตของตัวลบมีน้อยกว่าจำนวนบิตของตัวตั้ง ให้ทำให้จำนวนบิตของตัวลบเท่ากับจำนวนบิตของตัวตั้งก่อนโดยการเพิ่มบิต 0 ที่ข้างหน้าของตัวลบเช่น $1001_2 - 101_2$ พบว่าจำนวนบิตของตัวตั้งมี 4 บิต แต่จำนวนบิตของตัวลบมี 3 บิตเพราะฉะนั้นจึงเพิ่มบิต 0 ที่ข้างหน้าของตัวลบอีก 1 บิตได้เป็น 0101_2

2. หา 1's Complement ของตัวลบ

3. หาผลบวกระหว่างตัวตั้งและ 1's Complement ของตัวลบโดยผลลัพธ์แบ่งออกเป็น 2 กรณี

3.1 กรณีมีตัวทดตัวสุดท้ายให้นำผลลัพธ์ที่ได้จากการบวกที่ได้ในข้อ 3 มาบวกด้วย 1 และผลลัพธ์สุดท้ายมีค่าเป็นบวก

3.2 กรณีไม่มีตัวทดตัวสุดท้าย ให้หา 1's Complement ของผลลัพธ์ที่ได้ในข้อ 3 ซึ่งจะเป็ผลลัพธ์สุดท้ายและมีค่าลบ

ตัวอย่างที่ 1-27 จงหาผลลบของเลขฐานสองต่อไปนี้ $101101_2 - 11010_2$ ด้วยวิธี 1's Complement

วิธีทำ

ขั้นตอนที่ 1 ตรวจสอบจำนวนบิตของตัวตั้งมี 6 บิต แต่ของตัวลบมี 5 บิตซึ่งน้อยกว่าตัวตั้ง 1 บิต จึงเพิ่มบิต 0 ไปข้างหน้าตัวลบอีก 1 บิตได้เป็น 011010_2

ขั้นตอนที่ 2 หา 1's Complement ของตัวลบได้เป็น 100101_2

ขั้นตอนที่ 3 หาผลบวกระหว่างตัวตั้งและ 1's Complement ของตัวลบดังนี้

$$\begin{array}{r}
 101101 \\
 + \\
 100101 \\
 \hline
 (1)010010
 \end{array}$$

เนื่องจากมีตัวทดสุดท้าย จึงนำผลลัพธ์ที่ได้บวกด้วย 1 ดังนี้ $010010 + 1 = 10011$ และผลลัพธ์ที่ได้มีค่าเป็นบวก

$$\text{เพราะฉะนั้น } 101101_2 - 11010_2 = +10011_2$$

ตัวอย่างที่ 1-28 จงหาผลลบของเลขฐานสองต่อไปนี้ $100101_2 - 111010_2$ ด้วยวิธี 1's Complement

วิธีทำ

ขั้นตอนที่ 1 ตรวจสอบจำนวนบิตของตัวตั้งมี 6 บิต แต่ของตัวลบมี 6 บิตซึ่งเท่ากันอยู่แล้วจึงไม่ต้องมีการเติมบิตให้กับตัวลบ

ขั้นตอนที่ 2 หา 1's Complement ของตัวลบได้เป็น 000101_2

ขั้นตอนที่ 3 หาผลบวกระหว่างตัวตั้งและ 1's Complement ของตัวลบดังนี้

$$\begin{array}{r} 100101 \\ + \\ 000101 \\ \hline 101010 \end{array}$$

เนื่องจากไม่มีตัวทดสุดท้าย จึงหา 1's Complement ของผลลัพธ์ที่ได้ (101010) ได้เป็น 010101 และผลลัพธ์ที่ได้มีค่าเป็นลบ

$$\text{เพราะฉะนั้น } 100101_2 - 111010_2 = -10101_2$$

1.6.2 การลบเลขฐานสองด้วยวิธีคอมพลีเมนต์แบบ 2

การลบเลขฐานสองด้วยวิธีคอมพลีเมนต์แบบ 2 (2's Complement) มีหลักการดังนี้

1. ตรวจสอบจำนวนบิตตัวตั้ง และจำนวนบิตของตัวลบก่อนโดยหากจำนวนบิตของตัวลบมีน้อยกว่าจำนวนบิตของตัวตั้ง ให้ทำให้จำนวนบิตของตัวลบเท่ากับจำนวนบิตของตัวตั้งก่อนโดยการเพิ่มบิต 0 ที่ข้างหน้าของตัวลบเช่น $1001_2 - 101_2$ พบว่าจำนวนบิตของตัวตั้งมี 4 บิต แต่จำนวนบิตของตัวลบมี 3 บิตเพราะฉะนั้นจึงเพิ่มบิต 0 ที่ข้างหน้าของตัวลบอีก 1 บิตได้เป็น 0101_2

2. หา 2's Complement ของตัวลบ

3. หาผลบวกระหว่างตัวตั้งและ 2's Complement ของตัวลบโดยผลลัพธ์แบ่งออกเป็น 2 กรณี

3.1 กรณีมีตัวทดตัวสุดท้าย ให้ตัดตัวทอนั้นทิ้งไปเลย และนำผลลัพธ์ที่ได้จากการบวกที่ได้ในข้อ 3 มาเป็นผลลัพธ์สุดท้ายที่มีค่าเป็นบวก

3.2 กรณีไม่มีตัวทดตัวสุดท้าย ให้หา 2's Complement ของผลลัพธ์ที่ได้ในข้อ 3 ซึ่งจะเป็นผลลัพธ์สุดท้ายและมีค่าลบ

ตัวอย่างที่ 1-29 จงหาผลลบของเลขฐานสองต่อไปนี้ $101101_2 - 11010_2$ ด้วยวิธี 2's Complement

วิธีทำ

ขั้นตอนที่ 1 ตรวจสอบจำนวนบิตของตัวตั้งมี 6 บิต แต่ของตัวลบมี 5 บิตซึ่งน้อยกว่าตัวตั้ง 1 บิต จึงเพิ่มบิต 0 ไปข้างหน้าตัวลบอีก 1 บิตได้เป็น 011010_2

ขั้นตอนที่ 2 หา 2's Complement ของตัวลบได้เป็น 100110_2

ขั้นตอนที่ 3 หาผลบวกระหว่างตัวตั้งและ 2's Complement ของตัวลบดังนี้

$$\begin{array}{r} 101101 \\ + \\ \underline{100110} \\ \underline{(1)010011} \end{array}$$

เนื่องจากมีตัวทศสุดท้าย จึงตัดตัวทศนั้นทิ้งไปเลย และผลลัพธ์สุดท้ายที่ได้มีค่าเป็นบวก เพราะฉะนั้น $101101_2 - 11010_2 = +10011_2$

ข้อสังเกตไม่ว่าจะลบกันด้วยวิธี 1's Complement หรือ 2's Complement ผลลัพธ์สุดท้ายที่ได้จะต้องมีค่าเท่ากัน ซึ่งจากตัวอย่างที่ 1-27 และ ตัวอย่างที่ 1-29 เป็นการลบด้วยตัวเลขชุดเดียวกัน แต่ลบคนละวิธีกันซึ่งผลลัพธ์สุดท้ายมีค่าเท่ากันคือ $+10011$

ตัวอย่างที่ 1-30 จงหาผลลบของเลขฐานสองต่อไปนี้ $100101_2 - 111010_2$ ด้วยวิธี 2's Complement

วิธีทำ

ขั้นตอนที่ 1 ตรวจสอบจำนวนบิตของตัวตั้งมี 6 บิต แต่ของตัวลบมี 6 บิตซึ่งเท่ากันอยู่แล้วจึงไม่ต้องมีการเติมบิตให้กับตัวลบ

ขั้นตอนที่ 2 หา 2's Complement ของตัวลบได้เป็น 000110_2

ขั้นตอนที่ 3 หาผลบวกระหว่างตัวตั้งและ 2's Complement ของตัวลบดังนี้

$$\begin{array}{r} 100101 \\ + \\ \underline{000110} \\ \underline{101011} \end{array}$$

เนื่องจากไม่มีตัวทศสุดท้าย จึงหา 2's Complement ของผลลัพธ์ที่ได้ (101011) ได้เป็น 010101 และผลลัพธ์ที่ได้มีค่าเป็นลบ

เพราะฉะนั้น $100101_2 - 111010_2 = -10101_2$

1.7 การคูณเลขฐานสอง

การคูณเลขฐานสองจะมีหลักการคล้ายกับการคูณแบบเลขฐานสิบ แต่จะต่างกันตรงขั้นตอนของการนำผลคูณของแต่ละหลักมาบวกกันนั้นจะบวกกันแบบเลขฐานสอง เพื่อให้ง่ายต่อการอธิบาย ผู้เขียนจะขออธิบายวิธีการหารเลขฐานสองแบบละเอียดในตัวอย่างที่ 1-31

ตัวอย่างที่ 1-31 จงหาผลคูณของเลขฐานสองต่อไปนี้ $101_2 \times 11_2$

วิธีทำ

$$\begin{array}{r}
 101 \\
 \times 11 \\
 \hline
 101 \\
 101 \\
 \hline
 1111
 \end{array}$$

} บวกกันแบบเลขฐานสอง

เพราะฉะนั้น $101_2 \times 11_2 = 1111_2$

1.8 การหารเลขฐานสอง

การหารเลขฐานสอง ซึ่งในที่นี้ผู้เขียนจะใช้วิธีแบบหารยาว จะมีหลักการหารคล้ายกับวิธีการหารยาวแบบเลขฐานสิบ เพื่อให้ง่ายต่อการอธิบาย ผู้เขียนจะขออธิบายวิธีการหารเลขฐานสองแบบละเอียดในตัวอย่างที่ 1-32

ตัวอย่างที่ 1-32 จงหาผลหารของเลขฐานสองต่อไปนี้ $10010_2 \div 11_2$

วิธีทำ

ขั้นตอนที่ 1 นำตัวหารมาหารตัวตั้งครั้งละ 1 บิต โดยให้เริ่มจากบิตที่มีลำดับความสำคัญสูงสุด ซึ่งจาก 10010_2 บิตที่มีลำดับความสำคัญสูงสุดที่สุดมีค่าเท่ากับ 1 จึงได้ว่า $1 \div 11$ ได้ผลลัพธ์เป็น 0 (เนื่องจากค่าของตัวตั้งน้อยกว่าค่าของตัวหาร) ดังนี้

$$\begin{array}{r}
 0 \\
 11 \overline{) 10010}
 \end{array}$$

ขั้นตอนที่ 2 เนื่องจากค่าของตัวตั้งในขั้นตอนที่ 1 มีค่าน้อยกว่าตัวหารจึงเพิ่มบิตที่อยู่ทางขวามือของตัวตั้งมาอีก 1 บิต และนำมาหารด้วยตัวหารใหม่อีกครั้งได้เป็น $10 \div 11$ ได้ผลลัพธ์เป็น 0 ดังนี้

$$\begin{array}{r}
 00 \\
 11 \overline{) 10010}
 \end{array}$$

ขั้นตอนที่ 3 เนื่องจากค่าของตัวตั้งในขั้นตอนที่ 2 ยังคงมีค่าน้อยกว่าตัวหาร จึงเพิ่มบิตที่อยู่ทางขวามือของตัวตั้งตัวถัดไปมาอีก 1 บิต และนำมาหารด้วยตัวหารใหม่อีกครั้งได้เป็น $100 \div 11$ ได้ผลลัพธ์เป็นดังนี้

$$\begin{array}{r}
 001 \\
 11 \overline{) 10010} \\
 \underline{11} \\
 1
 \end{array}$$

ขั้นตอนที่ 4 เนื่องจากผลลัพธ์ที่ได้จากขั้นตอนที่ 3 คือ 1 ขั้นตอนต่อไปคือให้นำบิตที่อยู่ทางขวามือของตัวตั้งตัวถัดไปมาต่อท้ายผลลัพธ์ที่ได้จากขั้นตอนที่ 3 (ผลลัพธ์ที่ได้จากขั้นตอนที่ 3 คือ 1 และบิตตัวถัดไปของตัวตั้งคือ 1) และนำมาหารต่อด้วยตัวหารได้เป็น $11 \div 11$ ได้ผลลัพธ์เป็นดังนี้

$$\begin{array}{r} 0011 \\ 11 \overline{) 10010} \\ \underline{11} \\ 11 \\ \underline{11} \\ \underline{00} \end{array}$$

ขั้นตอนที่ 5 เนื่องจากผลลัพธ์ที่ได้จากขั้นตอนที่ 4 คือ 0 ขั้นตอนต่อไปคือให้นำบิตที่อยู่ทางขวามือของตัวตั้งตัวถัดไปซึ่งเป็นตัวสุดท้ายมาต่อท้ายผลลัพธ์ที่ได้จากขั้นตอนที่ 4 (ผลลัพธ์ที่ได้จากขั้นตอนที่ 3 คือ 0 และบิตตัวถัดไปของตัวตั้งคือ 0) และนำมาหารต่อด้วยตัวหารได้เป็น $00 \div 11$ ได้ผลลัพธ์ และขั้นตอนของการหารยาวทั้งหมดเป็นดังนี้

$$\begin{array}{r} 00110 \\ 11 \overline{) 10010} \\ \underline{00} \\ 00 \\ \underline{00} \\ \underline{00} \end{array}$$

$$\text{เพราะฉะนั้น } 10010_2 \div 11_2 = 0110_2 \text{ หรือ } 18_{10} \div 3_{10} = 6_{10}$$

1.9 บทสรุป

เนื่องจากระบบดิจิทัลจะมีการประมวลผลตัวเลขที่อยู่ในรูปของรหัสเลขฐานสองเท่านั้น แต่ตัวเลขที่มนุษย์ใช้กันอยู่ในชีวิตประจำวันเป็นเลขฐานสิบ ดังนั้นเพื่อให้การสื่อสารระหว่างวงจรถ และมนุษย์สามารถทำได้ง่าย จึงต้องมีการเรียนรู้เกี่ยวกับระบบเลขฐาน การแปลงเลขฐาน และการบวก การลบ การคูณ และการหารเลขฐาน โดยเลขฐานที่สำคัญและนิยมนำไปใช้งานบ่อยคือเลขฐาน 2, 8, 10 และ 16 นอกเหนือจากนั้นในบทนี้ได้กล่าวถึงการลบเลขฐานสองด้วยวิธีคอมพลิเมนต์ซึ่งมีข้อดีตรงที่การออกแบบวงจรถบวกร และวงจรถลบจะออกแบบเพียงแค่วงจรถเดียวแต่สามารถทำงานได้ทั้งเป็นวงจรถบวกร และวงจรถลบ แต่หากเป็นวงจรถลบแบบปกติจะต้องออกแบบ 2 วงจรถคือวงจรถบวกร และวงจรถลบ

คำถามท้ายบท

1. สัญญาณดิจิทัล และสัญญาณอนาลอกคืออะไร และมีความแตกต่างกันอย่างไร
2. จงยกตัวอย่างอุปกรณ์เครื่องใช้ทางอิเล็กทรอนิกส์ที่เป็นระบบแบบดิจิทัลอย่างน้อย 2 อุปกรณ์
3. จงแปลงเลขฐานสองต่อไปนี้ให้เป็นเลขฐานสิบ
 - 3.1) 101101
 - 3.2) 110110
 - 3.3) 1101101
 - 3.4) 0.1101
 - 3.5) 1101.11011
4. จงแปลงเลขฐานสิบต่อไปนี้เป็นเลขฐานสอง
 - 4.1) 19
 - 4.2) 253
 - 4.3) 967
 - 4.4) 0.75
 - 4.5) 34.5
5. จงแปลงเลขฐานสิบต่อไปนี้เป็นเลขฐานแปด และฐานสิบหก
 - 5.1) 19
 - 5.2) 1523
 - 5.3) 927
 - 5.4) 32
 - 5.5) 346
6. จงหาผลลัพธ์ต่อไปนี้
 - 6.1) $100101_2 + 11011_2 = 0_2$
 - 6.2) $732_8 + 263_8 = 0_8$
 - 6.3) $AH4_{16} + 1101_{16} = 0_{16}$
 - 6.4) $1748F_{16} - CCD_{16} = 0_{16}$
 - 6.5) $1011101_2 - 110110_2 = 0_2$
7. จงหาผลลัพธ์ของการลบเลขฐานสองต่อไปนี้แบบ 1' s complement และ 2' s complement
 - 7.1) $1011101 - 1000111$
 - 7.2) $1011 - 110111$
 - 7.3) $1011101 - 11$

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลลอจิกอิเล็กทรอนิกส์.จ. อุตรธานี: มหาวิทยาลัยราชภัฏอุตรธานี.**
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เกื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1.**กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล.**กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

แผนบริหารการสอนประจำบทที่ 2

รหัส 1 ชั่วโมง 30 นาที

หัวข้อเนื้อหา

- 2.1 บทนำ
- 2.2 รหัสบีซีดี
- 2.3 รหัสเพิ่ม 3
- 2.4 รหัสตรวจสอบความผิดพลาด
- 2.5 รหัสเกรย์
 - 2.5.1 การแปลงจากรหัสเลขฐานสองเป็นรหัสเกรย์
 - 2.5.2 การแปลงจากรหัสเกรย์เป็นรหัสเลขฐานสอง
- 2.6 รหัสแอสกี
- 2.7 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับรหัสต่าง ๆ ที่ใช้ในระบบดิจิทัล
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการแปลงจากรหัสบีซีดีเป็นเลขฐาน และการแปลงจากเลขฐานเป็นรหัสบีซีดี
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการใช้รหัสสำหรับการตรวจสอบความผิดพลาด
4. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการแปลงจากรหัสเกรย์เป็นเลขฐานสอง และการแปลงจากเลขฐานสองเป็นรหัสเกรย์
5. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับรหัสแอสกี และการแปลงค่าระหว่างตัวอักษร และรหัสแอสกีได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 2

รหัส

2.1 บทนำ

ดังที่ได้กล่าวไว้แล้วในบทที่ 1 ระบบดิจิทัลจะใช้เพียงรหัสที่เป็นเลขฐานสองซึ่งจะมีเพียงตัวเลข 0 หรือ 1 เท่านั้นสำหรับการประมวลผล ซึ่งนอกเหนือจากรหัสเลขฐานสองแล้ว ยังมีกลุ่มรหัสอื่น ๆ อีกเป็นจำนวนมากที่มีเพียง 0 และ 1 เป็นสมาชิกและสามารถใช้สำหรับการประมวลผลและสื่อสารในระบบได้

อย่างไรก็ตาม เนื่องจากรหัสที่ใช้สำหรับการประมวลผล และสื่อสารของระบบดิจิทัลมีอยู่เป็นจำนวนมาก ดังนั้นในบทนี้จะกล่าวถึงเฉพาะรหัสที่สำคัญและมีการใช้งานบ่อยเท่านั้น ส่วนรหัสอื่น ๆ ที่ไม่ได้กล่าวถึงจะมีวิธีการใช้งานที่คล้ายๆกัน ต่างกันเพียงรูปแบบของรหัสเท่านั้น

2.2 รหัสบีซีดี (BCD Code)

รหัสบีซีดี หรือ BCD (BCD ย่อมาจาก Binary Coded Decimal) เป็นรหัสที่ใช้แทนเลขฐานสิบจำนวน 10 ตัวด้วยเลขฐานสองที่แตกต่างกันในแต่ละชนิดของรหัส BCD ขนาด 4 บิต ซึ่งรหัส BCD นั้นมีอยู่หลายรหัสด้วยกัน แต่รหัสที่มีการใช้นิยมใช้งานมากที่สุดคือ รหัส BCD - 8421 สำหรับการแทนค่าระหว่างเลขฐานสิบ และรหัส BCD - 8421 พิจารณาได้จากตารางที่ 2.1

ตารางที่ 2.1 การแทนค่าเลขฐานสิบแต่ละตัวด้วยรหัส BCD - 8421

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

จากตารางที่ 2.1 แสดงให้เห็นว่าเลขฐานสิบ 1 ตัวถูกแทนด้วยรหัส BCD - 8421 ขนาด 4 บิต เพราะฉะนั้นการแปลงจากเลขฐานสิบเป็นรหัส BCD - 8421 จะแทนเลขฐานสิบแต่ละตัวด้วยรหัส BCD - 8421 ขนาด 4 บิต

สำหรับการแปลงจากรหัส BCD - 8421 เป็นเลขฐานสิบจะมีหลักการคล้ายกับ วิธีถัดในการแปลงจากเลขฐานสองเป็นเลขฐานสิบหก โดยให้แบ่งรหัส BCD - 8421 ออกเป็นกลุ่มๆละ 4 บิตโดยเริ่มนับจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปหาตำแหน่งที่มีลำดับความสำคัญสูงที่สุด และให้แทนเลขแต่ละกลุ่มด้วยเลขฐานสิบที่มีค่าตรงกันโดยดูจากตารางที่ 2.1

ตัวอย่างที่ 2-1 จงเปลี่ยน 921_{10} เป็นรหัส BCD - 8421

วิธีทำ

จากตารางที่ 2.1 ได้ผลดังนี้

9 แทนด้วย 1001

2 แทนด้วย 0010

1 แทนด้วย 0001

เพราะฉะนั้น $921_{10} = 100100100001$ (BCD - 8421)

ข้อสังเกต จากตารางที่ 2.1 ถึงแม้ว่าการแทนเลขฐานสิบ (0 - 9) ด้วยรหัส BCD - 8421 จะมีค่าตรงกับรหัสเลขฐานสอง แต่การแปลงจากเลขฐานสิบเป็นรหัส BCD - 8421 จะได้ค่าไม่เท่ากับการแปลงจากเลขฐานสิบเป็นรหัสเลขฐานสอง เนื่องจากวิธีการแปลงไม่เหมือนกัน พิจารณาตัวอย่างที่ 2-2

ตัวอย่างที่ 2-2 จงเปลี่ยน 47_{10} เป็นรหัส BCD - 8421 และ รหัสเลขฐานสอง

วิธีทำ

1. แปลง 47_{10} เป็นรหัส BCD - 8421

จากตารางที่ 2.1 ได้ผลดังนี้

4 แทนด้วย 0100

7 แทนด้วย 0111

เพราะฉะนั้น $47_{10} = 01000111$ (BCD - 8421)

2. แปลง 47_{10} เป็นรหัสเลขฐานสอง

$47/2 = 23$ เศษ 1 --> (ลำดับความสำคัญต่ำที่สุด)

$23/2 = 11$ เศษ 1

$11/2 = 5$ เศษ 1

$5/2 = 2$ เศษ 1

$2/2 = 1$ เศษ 0

$1/2 = 0$ เศษ 1 --> (ลำดับความสำคัญสูงที่สุด)

เพราะฉะนั้น $47_{10} = 101111_2$

จากตัวอย่างที่ 2-2 แสดงให้เห็นว่าค่าตัวเลขที่ได้จากการแปลงจากเลขฐานสิบเป็นรหัส BCD - 8421 จะไม่เท่ากับการแปลงจากเลขฐานสิบเป็นรหัสเลขฐานสองโดยที่ รหัส BCD - 8421 จะมีจำนวนบิตที่มากกว่ารหัสเลขฐานสอง

ตัวอย่างที่ 2-3 จงเปลี่ยน 10011_2 เป็นรหัส BCD – 8421

วิธีทำ เนื่องจากไม่มีวิธีการแปลงจากรหัสเลขฐานสองเป็นรหัส BCD – 8421 โดยตรง เพราะฉะนั้นต้องแปลงจากรหัสเลขฐานสองเป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นรหัส BCD – 8421

ขั้นตอนที่ 1 แปลงจากเลขฐานสองเป็นเลขฐานสิบ

| ตัวคูณ | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 |
|-----------------|-------|-------|-------|-------|-------|
| เลขประจำตำแหน่ง | 1 | 0 | 0 | 1 | 1 |

$$\begin{aligned} 10011_2 &= (1 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0) \\ &= 16 + 0 + 0 + 2 + 1 \\ &= 19_{10} \end{aligned}$$

ขั้นตอนที่ 2 แปลงจากเลขฐานสิบเป็นรหัส BCD - 8421

จากตารางที่ 2.1 ได้ผลดังนี้

1 แทนด้วย 0001

9 แทนด้วย 1001

เพราะฉะนั้น $10011_2 = 00011001$ (BCD - 8421)

ตัวอย่างที่ 2-4 จงเปลี่ยนรหัส BCD – 8421 ต่อไปนี้ 1001100000110110 เป็นเลขฐานสิบ

วิธีทำ

ขั้นตอนที่ 1 แบ่งรหัส BCD – 8421 เป็นกลุ่มๆละ 4 บิตโดยแบ่งจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปหาบิตที่มีลำดับความสำคัญสูงที่สุดคือ 1001100000110110 ได้ดังนี้

กลุ่มที่ 1 -> 0110 กลุ่มที่ 2 -> 0011 กลุ่มที่ 3 -> 1000 กลุ่มที่ 4 -> 1001

ขั้นตอนที่ 2 แทนเลขแต่ละกลุ่มด้วยเลขฐานสิบที่มีค่าตรงกัน

จากตารางที่ 2.1 ได้ผลดังนี้

กลุ่มที่ 1 0110 แทนด้วย 6

กลุ่มที่ 2 0011 แทนด้วย 3

กลุ่มที่ 3 1000 แทนด้วย 8

กลุ่มที่ 4 1001 แทนด้วย 9

เพราะฉะนั้น 1001100000110110 (BCD - 8421) = 9836_{10}

ตัวอย่างที่ 2-5 จงเปลี่ยน BCD – 8421 ต่อไปนี้ 000100110101 เป็นรหัสเลขฐานสอง

วิธีทำ เนื่องจากไม่มีวิธีการแปลงจากรหัส BCD – 8421 เป็นรหัสเลขฐานสองโดยตรง เพราะฉะนั้นต้องแปลงจากรหัส BCD – 8421 เป็นเลขฐานสิบก่อน แล้วจึงแปลงจากเลขฐานสิบเป็นรหัสเลขฐานสอง

ขั้นตอนที่ 1 แปลงจากรหัส BCD - 8421 เป็นเลขฐานสิบ โดยแบ่งรหัส BCD - 8421 เป็นกลุ่มๆละ 4 บิตโดยแบ่งจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปหาบิตที่มีลำดับความสำคัญสูงที่สุดคือ 000100110101 ได้ดังนี้

กลุ่มที่ 1 ->0101 กลุ่มที่ 2 -> 0011 กลุ่มที่ 3 ->0001

ขั้นตอนที่ 2 แทนเลขแต่ละกลุ่มด้วยเลขฐานสิบที่มีค่าตรงกัน

จากตารางที่ 2.1 ได้ผลดังนี้

กลุ่มที่ 1 0101 แทนด้วย 5

กลุ่มที่ 2 0011 แทนด้วย 3

กลุ่มที่ 3 0001 แทนด้วย 1

ได้เป็น 000100110101 (BCD - 8421) = 135₁₀

ขั้นตอนที่ 3 แปลงจากเลขฐานสิบเป็นรหัสเลขฐานสอง

135/2 = 67 เศษ 1 --> (ลำดับความสำคัญต่ำที่สุด)

67/2 = 33 เศษ 1

33/2 = 16 เศษ 1

16/2 = 8 เศษ 0

8/2 = 4 เศษ 0

4/2 = 2 เศษ 0

2/2 = 1 เศษ 0

1/2 = 0 เศษ 1 --> (ลำดับความสำคัญสูงที่สุด)

ได้เป็น 135₁₀ = 10000111₂

เพราะฉะนั้น 000100110101 (BCD - 8421) = 10000111₂

นอกจากรหัส BCD - 8421 แล้วยังมีรหัส BCD แบบอื่นอีกเป็นจำนวนมากโดยแต่ละชนิดจะมีสถานะเอาต์พุตที่มีขนาด 4 บิตที่แตกต่างกันออกไป ตัวอย่างของรหัส BCD ชนิดอื่นเช่น BCD - 7421 หรือ BCD - 6421 เป็นต้น

2.3 รหัสเพิ่ม 3 (Excess - 3 Code)

รหัสเพิ่ม 3 นั้นเป็นรหัสที่ไม่มีน้ำหนัก (Nonweighted Code) ความหมายคือ ไม่มีตำแหน่งบิตในกลุ่มรหัสตัวใดที่สามารถบอกถึงน้ำหนักเป็นพิเศษ จึงไม่เหมาะสมสำหรับใช้คำนวณ โดยรหัส เพิ่ม 3 นั้นมีการปรับปรุงมาจากรหัส BCD - 8421 ซึ่งค่าแต่ละหลักของรหัสเพิ่ม 3 มีค่าเท่ากับค่าแต่ละหลักของรหัส BCD - 8421 บวกเพิ่มด้วย 3 จึงเป็นที่มาของชื่อรหัสเพิ่ม 3 ส่วนวิธีการแปลงจากเลขฐานสิบเป็นรหัส เพิ่ม 3 นั้นจะใช้วิธีการแปลงจากเลขฐานสิบเป็นรหัส BCD - 8421 ก่อนแล้วจึงนำรหัส BCD - 8421 แต่ละชุด (ชุดละ 4 บิตที่แปลงมาจากเลขฐานสิบ) มาบวกเพิ่มด้วย 0011₂ (0011₂ = 3₁₀)

ตัวอย่างที่ 2-6 จงเปลี่ยน 24₁₀ เป็นรหัสเพิ่ม 3

วิธีทำ

ขั้นตอนที่ 1 แปลงจากเลขฐานสิบเป็นรหัส BCD - 8421

จากตารางที่ 2.1 ได้ผลดังนี้

4 แทนด้วย 0100
 2 แทนด้วย 0010
 ได้เป็น $24_{10} = 00100100$ (BCD - 8421)
 ขั้นตอนที่ 2 บวกรหัส BCD - 8421 แต่ละชุดด้วย 0011

| | | |
|----------------|--|----------------|
| 0 0 1 0 | | 0 1 0 0 |
| + | | + |
| <u>0 0 1 1</u> | | <u>0 0 1 1</u> |
| <u>0 1 0 1</u> | | <u>0 1 1 1</u> |

เพราะฉะนั้น $24_{10} = 01010111$ (รหัสเพิ่ม 3)

ประโยชน์ของรหัสเพิ่ม 3 คือนำมาใช้ในการหาผลลัพธ์ของการบวกกันของรหัส BCD - 8421 ในกรณีที่ผลลัพธ์ของการบวกกันของรหัส BCD - 8421 ในแต่ละชุดข้อมูลมีค่าเกิน 9 พิจารณาตัวอย่างที่ 2-7 และตัวอย่างที่ 2-8 เพื่อประกอบการอธิบาย

ตัวอย่างที่ 2-7 จงหาผลบวกของรหัส BCD - 8421 ต่อไปนี้ 0001 + 0011
 วิธีทำ

| | | |
|----------------|--|----------|
| 0 0 0 1 | | 1 |
| + | | |
| <u>0 0 1 1</u> | | 3 |
| <u>0 1 0 0</u> | | <u>4</u> |

เพราะฉะนั้น $0001 + 0011 = 0100$ ซึ่งมีค่าตรงกับเลขฐานสิบคือ $1 + 3 = 4$
 จากตัวอย่างที่ 2-7 กรณีที่นำรหัส BCD - 8421 มาบวกกันและค่าที่ได้ไม่เกิน 9 แล้วจะสามารถบวกกันได้ตามปกติ ซึ่งจากตัวอย่างหากมองเป็นเลขฐานสิบคือ $1 + 3 = 4$ ซึ่งไม่เกิน 9

ตัวอย่างที่ 2-8 จงหาผลบวกของรหัส BCD - 8421 ต่อไปนี้ 0111 + 0110
 วิธีทำ

| | | |
|----------------|--|----------|
| 0 1 1 1 | | 7 |
| + | | |
| <u>0 1 1 0</u> | | <u>6</u> |
| <u>1 1 0 1</u> | | <u>?</u> |

เนื่องจาก $0111 + 0110 = 1101$ ซึ่งไม่มีอยู่ในรหัส BCD-8421 จึงใช้รหัสเพิ่ม 3 ในการบวกแทนได้ ดังนี้

| | | |
|--------------------|--|-----------|
| 1 0 1 0 | | 7 |
| + | | |
| <u>1 0 0 1</u> | | <u>6</u> |
| <u>(1) 0 0 1 1</u> | | <u>13</u> |

จากตัวอย่างที่ 2-8 กรณีที่นำรหัส BCD – 8421 มาบวกกันและค่าที่ได้เกิน 9 จะต้องแปลงรหัส BCD-8421 เป็นรหัสเพิ่ม 3 แล้วจึงนำมาบวกกันโดยผลลัพธ์ที่ได้จากการบวกกันเป็นรหัส BCD - 8421

2.4 รหัสตรวจสอบความผิดพลาด (Error - Detecting Code)

เนื่องจากข้อมูลที่จะต้องมีารรับ หรือส่งด้วยระบบดิจิทัล อาจมีข้อผิดพลาดเกิดขึ้นไม่ว่าจะเป็นข้อผิดพลาดที่เกิดขึ้นก่อนที่จะมีการรับส่งข้อมูล หรือข้อผิดพลาดที่เกิดขึ้นหลังจากรับส่งข้อมูลเรียบร้อยแล้ว ดังนั้นจึงจำเป็นที่จะต้องมีการตรวจสอบหาข้อผิดพลาดของข้อมูลก่อนที่จะนำไปใช้งานจริง เพื่อไม่ให้เกิดความเสียหาย หรือเกิดความเสียหายน้อยที่สุด การใช้บิตพริตี้ (Parity bit) เป็นวิธีหนึ่งที่สามารถนำมาตรวจสอบหาความผิดพลาดของข้อมูลได้ โดยการพิจารณาเฉพาะจำนวนของบิตที่มีค่าเป็น 1 จากจำนวนบิตทั้งหมดซึ่งแบ่งออกเป็น 2 วิธี ดังนี้

วิธีที่ 1 พริตี้คู่ (Even parity) คือ การทำให้จำนวนบิตทั้งหมดมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคู่ โดยการเติมบิตเพิ่มเข้าไปในข้อมูลอีก 1 บิต โดยหากข้อมูลเดิมมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคี่ให้เติมบิต "1" เพิ่มเข้าไปเพื่อเปลี่ยนให้จำนวนบิตที่มีค่าเป็น 1 กลายเป็นจำนวนคู่ แต่หากข้อมูลเดิมมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคู่อยู่แล้วให้เติมบิต "0" เพิ่มเข้าไปเพื่อให้จำนวนบิตที่มีค่าเป็น 1 มีเป็นจำนวนคู่คงเดิม

วิธีที่ 2 พริตี้คี่ (Odd parity) คือ การทำให้จำนวนบิตทั้งหมดมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคี่ โดยการเติมบิตเพิ่มเข้าไปในข้อมูลอีก 1 บิต โดยหากข้อมูลเดิมมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคู่ให้เติมบิต "1" เพิ่มเข้าไปเพื่อเปลี่ยนให้จำนวนบิตที่มีค่าเป็น 1 มีเป็นจำนวนคี่ แต่หากข้อมูลเดิมมีจำนวนบิตที่มีค่าเป็น 1 เป็นจำนวนคี่อยู่แล้วให้เติมบิต "0" เพิ่มเข้าไปเพื่อให้จำนวนบิตที่มีค่าเป็น 1 มีเป็นจำนวนคี่คงเดิม

ตัวอย่างที่ 2-9 จงเติมพริตี้คู่ และ พริตี้คี่จากให้รหัส 1001

วิธีทำ เนื่องจากรหัส 1001 มีจำนวนบิตที่มีค่าเป็น 1 อยู่ 2 ตัว ดังนั้นการเติมพริตี้คู่ และ พริตี้คี่ เป็นดังนี้

การเติมพริตี้คู่: เติม "0" เพิ่มเข้าไป ได้เป็น "10010"

การเติมพริตี้คี่: เติม "1" เพิ่มเข้าไป ได้เป็น "10011"

2.5 รหัสเกรย์ (Gray Code)

รหัสเกรย์ เป็นรหัสที่อยู่ในประเภทที่ไม่มีน้ำหนัก เช่นเดียวกับรหัส Excess – 3 เพราะฉะนั้นจึงเป็นรหัสที่ไม่เหมาะที่จะนำมาใช้ในการคำนวณทางคณิตศาสตร์ โดยลักษณะของรหัสเกรย์ คือจะมีการเปลี่ยนแปลงข้อมูลในแต่ละลำดับเพียงแค่ 1 บิตเท่านั้นจึงทำให้โอกาสเกิดความผิดพลาดมีน้อย ซึ่งหากใช้รหัสเกรย์เป็นอินพุตหรือเอาต์พุตจะมีความผิดพลาดน้อยกว่าหากเทียบกับรหัสเลขฐานสอง

พิจารณาตารางที่ 2.2 ประกอบการอธิบาย

ตารางที่ 2.2 การเปรียบเทียบการเปลี่ยนแปลงของบิตข้อมูลระหว่างรหัสเกรย์และรหัสเลขฐานสอง

| เลขฐานสิบ | รหัสเลขฐานสอง | รหัสเกรย์ |
|-----------|---------------|-----------|
| 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 |

จากตารางที่ 2.2 แสดงให้เห็นว่าการเปลี่ยนแปลงค่าในแต่ละลำดับขั้นนั้น รหัสเกรย์จะมีการเปลี่ยนแปลงเพียงครั้งละ 1 บิตเท่านั้น แต่รหัสเลขฐานสองจะมีการเปลี่ยนแปลงอย่างน้อย 1 บิต (ซึ่งอาจจะมากกว่า 1 บิต) เช่น

การเปลี่ยนจาก 1 เป็น 2 ในเลขฐานสิบ

| เลขฐานสิบ | รหัสเลขฐานสอง | รหัสเกรย์ |
|-----------|---------------|-----------|
| 1 | 0001 | 0001 |
| 2 | 0010 | 0011 |

การเปลี่ยนจาก 7 เป็น 8 ในเลขฐานสิบ

| เลขฐานสิบ | รหัสเลขฐานสอง | รหัสเกรย์ |
|-----------|---------------|-----------|
| 7 | 0111 | 0100 |
| 8 | 1000 | 1100 |

จากตัวอย่างข้างต้นสังเกตเห็นว่าการเปลี่ยนจาก 1 เป็น 2 และจาก 7 เป็น 8 ในเลขฐานสิบ จะทำให้รหัสเลขฐานสองมีการเปลี่ยนแปลง 2 และ 4 บิตตามลำดับ แต่ในรหัสเกรย์นั้นจะมีการเปลี่ยนแปลงเพียง 1 บิตเสมอ

ประโยชน์ของรหัสเกรย์ คือนำไปใช้กับ Encoder ซึ่งเป็นอุปกรณ์ที่ใช้สำหรับการวัดตำแหน่งและนับความเร็วรอบที่มีความแม่นยำสูง

สำหรับวิธีการคำนวณหารหัสเกรย์นั้นทำได้โดยการแปลงมาจากรหัสเลขฐานสอง โดยการแปลงจากรหัสเลขฐานสองเป็นรหัสเกรย์ และการแปลงจากรหัสเกรย์เป็นรหัสเลขฐานสองจะมีวิธีการที่คล้ายกัน ดังนี้

2.5.1 การแปลงจากรหัสเลขฐานสองเป็นรหัสเกรย์

การแปลงจากรหัสเลขฐานสองเป็นรหัสเกรย์มีขั้นตอนดังนี้

1. กำหนดให้บิตที่มีลำดับความสำคัญสูงสุดที่สุดคือ n (จำนวนบิตทั้งหมดของรหัสเลขฐานสองลบด้วย 1) และบิตที่มีลำดับความสำคัญต่ำที่สุดคือ 0
2. บิตที่ n ของรหัสเกรย์จะมีค่าเท่ากับบิตที่ n ของรหัสเลขฐานสอง
3. นำบิตที่ n และบิตที่ $n-1$ ของรหัสเลขฐานสองมาบวกกันผลลัพธ์ที่ได้จะเป็นบิตที่ $n-1$ ของรหัสเกรย์โดยการบวกนั้นจะเป็นการบวกแบบเลขฐานสองโดยไม่คิดตัวทด เพราะฉะนั้น $1 + 1 = 0$

4. กำหนดให้บิตที่ $n-1$ ของรหัสเลขฐานสองเป็นบิตที่ n ซึ่งพิจารณาได้ดังนี้

4.1 กรณีที่บิตที่ n ยังไม่เท่ากับ 0 ให้กลับไปขั้นตอนที่ 3

4.2 กรณีที่บิตที่ n มีค่าเท่ากับ 0 จบการทำงานโดยคำตอบของรหัสเกรย์คือบิตที่ n ของรหัสเกรย์จะเป็นบิตที่มีลำดับความสำคัญสูงสุดเรียงลงไปจนถึงบิตที่ 0 ของรหัสเกรย์ซึ่งเป็นบิตที่มีลำดับความสำคัญต่ำที่สุด

ตัวอย่างที่ 2-10 จงแปลงจากรหัสเลขฐานสองต่อไปนี้ 10011_2 เป็นรหัสเกรย์

วิธีทำ

จากตัวอย่างพบว่า 10011_2 มีทั้งหมด 5 บิตเพราะฉะนั้น n มีค่าเท่ากับ $5-1 = 4$ การคำนวณหา รหัสเกรย์สามารถทำได้ดังนี้

ขั้นตอนที่ 1 บิตที่ n ของรหัสเกรย์จะมีค่าเท่ากับบิตที่ n ของรหัสเลขฐานสอง

| | | | | | |
|---------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเลขฐานสอง | 1 | 0 | 0 | 1 | 1 |
| รหัสเกรย์ | ① | | | | |

ขั้นตอนที่ 2 นำบิตที่ 4 และบิตที่ 3 ของรหัสเลขฐานสองมาบวกกันผลลัพธ์ที่ได้จะเป็นบิตที่ 3 ของรหัสเกรย์ได้เป็น

1 (บิตที่ 4 ของรหัสเลขฐานสอง) + 0 (บิตที่ 3 ของรหัสเลขฐานสอง) = 1 (บิตที่ 3 ของรหัสเกรย์)

| | | | | | |
|---------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเลขฐานสอง | 1 | 0 | 0 | 1 | 1 |
| รหัสเกรย์ | 1 | ① | | | |

ขั้นตอนที่ 3 กำหนดให้บิตที่ $n-1$ ของรหัสเลขฐานสองเป็นบิตที่ n เพราะฉะนั้นปัจจุบันบิตที่ n คือบิตที่ 3 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 4 นำบิตที่ 3 และบิตที่ 2 ของรหัสเลขฐานสองมาบวกกันผลลัพธ์ที่ได้จะเป็นบิตที่ 2 ของรหัสเกรย์ได้เป็น

$$0 \text{ (บิตที่ 3 ของรหัสเลขฐานสอง)} + 0 \text{ (บิตที่ 2 ของรหัสเลขฐานสอง)} = 0 \text{ (บิตที่ 2 ของรหัสเกรย์)}$$

| | | | | | |
|-------------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัส เลขฐานสอง | 1 | 0 | 0 | 1 | 1 |
| รหัสเกรย์ | 1 | 1 | 0 | | |

ขั้นตอนที่ 5 กำหนดให้บิตที่ $n-1$ ของรหัสเลขฐานสองเป็นบิตที่ n เพราะฉะนั้นปัจจุบันบิตที่ n คือบิตที่ 2 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 6 นำบิตที่ 2 และบิตที่ 1 ของรหัสเลขฐานสองมาบวกกันผลลัพธ์ที่ได้จะเป็นบิตที่ 1 ของรหัสเกรย์ได้เป็น

$$0 \text{ (บิตที่ 2 ของรหัสเลขฐานสอง)} + 1 \text{ (บิตที่ 1 ของรหัสเลขฐานสอง)} = 1 \text{ (บิตที่ 1 ของรหัสเกรย์)}$$

| | | | | | |
|-------------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัส เลขฐานสอง | 1 | 0 | 0 | 1 | 1 |
| รหัสเกรย์ | 1 | 1 | 0 | 1 | |

ขั้นตอนที่ 7 กำหนดให้บิตที่ $n-1$ ของรหัสเลขฐานสองเป็นบิตที่ n เพราะฉะนั้นปัจจุบันบิตที่ n คือบิตที่ 1 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 8 นำบิตที่ 1 และบิตที่ 0 ของรหัสเลขฐานสองมาบวกกันผลลัพธ์ที่ได้จะเป็นบิตที่ 0 ของรหัสเกรย์ได้เป็น

$$1 \text{ (บิตที่ 1 ของรหัสเลขฐานสอง)} + 1 \text{ (บิตที่ 0 ของรหัสเลขฐานสอง)} = 0 \text{ (บิตที่ 0 ของรหัสเกรย์)}$$

| | | | | | |
|-------------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัส เลขฐานสอง | 1 | 0 | 0 | 1 | 1 |
| รหัสเกรย์ | 1 | 1 | 0 | 1 | 0 |

ขั้นตอนที่ 9 กำหนดให้บิตที่ $n-1$ ของรหัสเลขฐานสองเป็นบิตที่ n เพราะฉะนั้นปัจจุบันบิตที่ n คือบิตที่ 0 ซึ่งมีค่าเท่ากับ 0 แล้ว

$$\text{เพราะฉะนั้น } 10011_2 = 11010 \text{ (Gray Code)}$$

2.5.2 การแปลงจากรหัสเกรย์เป็นรหัสเลขฐานสอง

การแปลงจากรหัสเกรย์เป็นรหัสเลขฐานสองมีขั้นตอนดังนี้

1. กำหนดให้บิตที่มีลำดับความสำคัญสูงที่สุดคือ n (จำนวนบิตทั้งหมดของรหัสเกรย์ลบด้วย 1) และบิตที่มีลำดับความสำคัญต่ำที่สุดคือ 0

2. บิตที่ n ของรหัสเลขฐานสองจะมีค่าเท่ากับบิตที่ n ของรหัสเกรย์

3. นำบิตที่ n ของรหัสเลขฐานสอง บวกกับบิตที่ $n-1$ ของรหัสเกรย์ ผลลัพธ์ที่ได้จะเป็นบิตที่ $n - 1$ ของรหัสเลขฐานสองโดยการบวกนั้นจะเป็นการบวกแบบเลขฐานสองโดยไม่คิดตัวทด เพราะฉะนั้น $1 + 1 = 0$

4. กำหนดให้บิตที่ $n-1$ ของรหัสเกรย์ และรหัสเลขฐานสองมาเป็นบิตที่ n ซึ่งพิจารณา ดังนี้

4.1 กรณีบิตที่ n ยังไม่เท่ากับ 0 ให้กลับไปขั้นตอนที่ 3

4.2 กรณีที่บิตที่ n มีค่าเท่ากับ 0 จบการทำงานโดยคำตอบของรหัสเลขฐานสองคือบิตที่ n ของรหัสเลขฐานสองจะเป็นบิตที่มีลำดับความสำคัญสูงสุดเรียงลงไปจนถึงบิตที่ 0 ของรหัสเลขฐานสองซึ่งเป็นบิตที่มีลำดับความสำคัญต่ำที่สุด

ตัวอย่างที่ 2-11 จงแปลงจากรหัสเกรย์ต่อไปนี้ 11100 เป็นรหัสเลขฐานสอง

วิธีทำ

จากตัวอย่างพบว่า 11100 มีทั้งหมด 5 บิตเพราะฉะนั้น n มีค่าเท่ากับ $5 - 1 = 4$ การคำนวณหารหัสเลขฐานสองสามารถทำได้ดังนี้

ขั้นตอนที่ 1 บิตที่ n ของรหัสเลขฐานสองจะมีค่าเท่ากับบิตที่ n ของรหัสเกรย์

| | | | | | |
|---------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเกรย์ | 1 | 1 | 1 | 0 | 0 |
| รหัสเลขฐานสอง | ① | | | | |

ขั้นตอนที่ 2 นำบิตที่ 4 ของรหัสเลขฐานสองมาบวกกับบิตที่ 3 ของรหัสเกรย์ ผลลัพธ์ที่ได้จะเป็นบิตที่ 3 ของรหัสเลขฐานสองได้เป็น

1 (บิตที่ 4 ของรหัสเลขฐานสอง) + 1 (บิตที่ 3 ของรหัสเกรย์) = 0 (บิตที่ 3 ของรหัสเลขฐานสอง)

| | | | | | |
|---------------|---|---|---|---|---|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเกรย์ | 1 | 1 | 1 | 0 | 0 |
| รหัสเลขฐานสอง | 1 | ① | | | |

ขั้นตอนที่ 3 กำหนดให้บิตที่ $n-1$ ของรหัสเกรย์ และรหัสเลขฐานสอง เป็นบิตที่ n เพราะฉะนั้นปัจจุบันบิตที่ n คือบิตที่ 3 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 4 นำบิตที่ 3 ของรหัสเลขฐานสองมาบวกกับบิตที่ 2 ของรหัสเกรย์ ผลลัพธ์ที่ได้จะเป็นบิตที่ 2 ของรหัสเลขฐานสองได้เป็น

0 (บิตที่ 3 ของรหัสเลขฐานสอง) + 1 (บิตที่ 2 ของรหัสเกรย์) = 1 (บิตที่ 2 ของรหัสเลขฐานสอง)

| | | | | | |
|---------------|----------|----------|----------|----------|----------|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเกรย์ | 1 | 1 | 1 | 0 | 0 |
| รหัสเลขฐานสอง | 1 | 0 | Ⓛ | | |

ขั้นตอนที่ 5 กำหนดให้บิตที่ $n-1$ ของรหัสเกรย์ และรหัสเลขฐานสอง เป็นบิตที่ n เพราะฉะนั้น ปัจจุบันบิตที่ n คือบิตที่ 2 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 6 นำบิตที่ 2 ของรหัสเลขฐานสองมาบวกกับบิตที่ 1 ของรหัสเกรย์ ผลลัพธ์ที่ได้จะเป็น บิตที่ 1 ของรหัสเลขฐานสองได้เป็น

$$1 (\text{บิตที่ 2 ของรหัสเลขฐานสอง}) + 0 (\text{บิตที่ 1 ของรหัสเกรย์}) = 1 (\text{บิตที่ 1 ของรหัสเลขฐานสอง})$$

| | | | | | |
|---------------|----------|----------|----------|----------|----------|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเกรย์ | 1 | 1 | 1 | 0 | 0 |
| รหัสเลขฐานสอง | 1 | 0 | 1 | Ⓛ | |

ขั้นตอนที่ 7 กำหนดให้บิตที่ $n-1$ ของรหัสเกรย์ และรหัสเลขฐานสอง เป็นบิตที่ n เพราะฉะนั้น ปัจจุบันบิตที่ n คือบิตที่ 1 ซึ่งยังมีค่าไม่เท่ากับ 0

ขั้นตอนที่ 8 นำบิตที่ 1 ของรหัสเลขฐานสองมาบวกกับบิตที่ 0 ของรหัสเกรย์ ผลลัพธ์ที่ได้จะเป็น บิตที่ 0 ของรหัสเลขฐานสองได้เป็น

$$1 (\text{บิตที่ 1 ของรหัสเลขฐานสอง}) + 0 (\text{บิตที่ 0 ของรหัสเกรย์}) = 1 (\text{บิตที่ 0 ของรหัสเลขฐานสอง})$$

| | | | | | |
|---------------|----------|----------|----------|----------|----------|
| บิตที่ | 4 | 3 | 2 | 1 | 0 |
| รหัสเกรย์ | 1 | 1 | 1 | 0 | 0 |
| รหัสเลขฐานสอง | 1 | 0 | 1 | 1 | Ⓛ |

ขั้นตอนที่ 9 กำหนดให้บิตที่ $n-1$ ของรหัสเกรย์ และรหัสเลขฐานสอง เป็นบิตที่ n เพราะฉะนั้น ปัจจุบันบิตที่ n คือบิตที่ 0 ซึ่งมีค่าเท่ากับ 0 แล้ว

$$\text{เพราะฉะนั้น } 11100 (\text{Gray Code}) = 10111_2$$

2.6 รหัสแอสกี (ASCII Code)

รหัสแอสกี หรือ ASCII Code มีชื่อเต็มคือ American Standard Code for Information Interchange คือรหัสที่ประกอบไปด้วยกลุ่มของเลขฐานสองขนาด 7 บิต ซึ่งแต่ละกลุ่มของรหัสดังกล่าว จะถูกแทนด้วยตัวอักษรภาษาอังกฤษพิมพ์ใหญ่ (A - Z), ตัวอักษรภาษาอังกฤษพิมพ์เล็ก (a - z), ตัวเลขอารบิก (0 - 9) และอักขระอื่นๆ โดยที่ รหัส 1 ตัวจะถูกแทนด้วยอักขระ 1 ตัว เช่น 41_{16} จะหมายถึงตัวอักษร "A", 61_{16} จะหมายถึงตัวอักษร "a" เป็นต้น

ตารางที่ 2.3 รหัสแอสกีที่ใช้แทนสัญลักษณ์บางส่วนโดยเปรียบเทียบกับเลขฐานสิบหก

| เลขฐานสิบหก | สัญลักษณ์ | เลขฐานสิบหก | สัญลักษณ์ | เลขฐานสิบหก | สัญลักษณ์ |
|-------------|-----------|-------------|-----------|-------------|-----------|
| 41 | A | 61 | a | 30 | 0 |
| 42 | B | 62 | b | 31 | 1 |
| 43 | C | 63 | c | 32 | 2 |
| 44 | D | 64 | d | 33 | 3 |
| 45 | E | 65 | e | 34 | 4 |
| 46 | F | 66 | f | 35 | 5 |
| 47 | G | 67 | g | 36 | 6 |
| 48 | H | 68 | h | 37 | 7 |
| 49 | I | 69 | i | 38 | 8 |
| 4A | J | 6A | j | 39 | 9 |
| 4B | K | 6B | k | | |
| 4C | L | 6C | l | | |
| 4D | M | 6D | m | | |
| 4E | N | 6E | n | | |
| 4F | O | 6F | o | | |
| 50 | P | 70 | p | | |
| 51 | Q | 71 | q | | |
| 52 | R | 72 | r | | |
| 53 | S | 73 | s | | |
| 54 | T | 74 | t | | |
| 55 | U | 75 | u | | |
| 56 | V | 76 | v | | |
| 57 | W | 77 | w | | |
| 58 | X | 78 | x | | |
| 59 | Y | 79 | y | | |
| 5A | Z | 7A | z | | |

ตารางที่ 2.3 แสดงการเปรียบเทียบค่าระหว่างเลขฐานสิบหก กับสัญลักษณ์ต่างๆ บางส่วน โดยที่เลขฐานสิบหกข้างต้นนี้เกิดจากการแปลงมาจากเลขฐานสองขนาด 7 บิต เช่น หากต้องการสัญลักษณ์ "A" ก็ต้องแทนค่าด้วย 41_{16} ซึ่ง 41_{16} หากมองในรูปแบบของเลขฐานสองคือ 1000001 (หากต้องการแปลงเป็นเลขฐานสิบหกโดยใช้วิธีใส่บิต 0 เพิ่มที่ข้างหน้าอีก 1 บิตสำหรับแต่ละกลุ่มของรหัสแอสกีซึ่งจากตัวอย่างได้เป็น 01000001 ซึ่งจะมีค่าเท่ากับ 41_{16} เช่นเดิม)

ตัวอย่างที่ 2-12 จงแสดงรหัสแอสกีต่อไปนี้ออกมาเป็นข้อความ

44_{16} 49_{16} 47_{16} 49_{16} 54_{16} 41_{16} $4C_{16}$

วิธีทำ จากตารางที่ 2.3 ได้ผลดังนี้

44_{16} แทนด้วย D

49_{16} แทนด้วย I

47_{16} แทนด้วย G

49_{16} แทนด้วย I

54_{16} แทนด้วย T

41_{16} แทนด้วย A

$4C_{16}$ แทนด้วย L

เพราะฉะนั้น ข้อความที่ได้คือ **DIGITAL**

ตัวอย่างที่ 2-13 จงแสดงรหัสแอสกีต่อไปนี้ออกมาเป็นข้อความ

100001110000011010100_2

วิธีทำ เนื่องจากโจทย์กำหนดเป็นรหัสเลขฐานสองเพื่อให้ง่ายในการคำนวณ ควรแปลงจากรหัสเลขฐานสองเป็นเลขฐานสิบหกก่อนดังนี้

ขั้นตอนที่ 1 แแบงรหัสเลขฐานสองออกเป็นกลุ่มๆละ 7 บิต โดยแบ่งจากบิตที่มีลำดับความสำคัญต่ำสุดไปหาบิตที่มีลำดับความสำคัญสูงสุดคือ 100001110000011010100 ได้ดังนี้

กลุ่มที่ 1 คือ 1010100

กลุ่มที่ 2 คือ 1000001

กลุ่มที่ 3 คือ 1000011

ขั้นตอนที่ 2 แปลงจากรหัสเลขฐานสองของแต่ละกลุ่มเป็นเลขฐานสิบหก ได้ดังนี้

กลุ่มที่ 1 คือ $1010100 = 54_{16}$

กลุ่มที่ 2 คือ $1000001 = 41_{16}$

กลุ่มที่ 3 คือ $1000011 = 43_{16}$

ขั้นตอนที่ 3 แทนเลขฐานสิบหกแต่ละกลุ่มด้วยตัวอักษรที่ตรงตามตารางที่ 2.3 ได้ดังนี้

54_{16} แทนด้วย T

41_{16} แทนด้วย A

43_{16} แทนด้วย C

เพราะฉะนั้น ข้อความที่ได้คือ **CAT**

2.7 บทสรุป

วงจรถิจิตัลจะประมวลผลเพียงรหัสที่อยู่ในรูปแบบรหัสเลขฐานสองเท่านั้น ซึ่งรหัสมีอยู่หลายชนิด เช่น รหัส BCD - 8421 รหัสเพิ่ม 3 รหัสเกรย์ หรือ รหัสแอสกี เป็นต้น โดยรหัสเพิ่ม 3 เกิดจากการบวกค่ารหัส BCD - 8421 ที่มีขนาด 4 บิต ด้วย 3 โดยรหัสเพิ่ม 3 เป็นรหัสที่ไม่มีการถ่วงน้ำหนัก เช่นเดียวกันกับ รหัสเกรย์ จึงไม่เหมาะในการนำมาคำนวณทางคณิตศาสตร์ แต่ถูกนิยมนำมาใช้กับอุปกรณ์อินพุต หรือเอาต์พุต รหัสแอสกีเป็นรหัสมาตรฐานที่ใช้แทนตัวอักษรภาษาอังกฤษ รวมถึงอักขระที่สำคัญและเป็นมาตรฐานตัวอื่นๆ นอกจากนั้นรหัสยังสามารถถูกนำมาใช้สำหรับการ

ตรวจสอบข้อผิดพลาดของข้อมูล ซึ่งในเอกสารเล่มนี้กล่าวถึงการตรวจสอบความผิดพลาดด้วยวิธี พาริตีคู่ และ พาริตีคี่

คำถามท้ายบท

1. รหัสคืออะไร
2. รหัสเลขฐานสอง และรหัส BCD – 8421 มีความแตกต่างกันอย่างไร
3. จงแปลงจากรหัสเลขฐานสิบต่อไปนี้เป็นรหัส BCD – 8421
 - 3.1) 79
 - 3.2) 94
 - 3.3) 186
 - 3.4) 399
 - 3.5) 4101
4. จงแปลงจากรหัส BCD – 8421 ต่อไปนี้เป็นรหัสเลขฐานสิบ
 - 4.1) 0110011110010101
 - 4.2) 1000010000100001
 - 4.3) 0010001101010110
 - 4.4) 0111100100100111
 - 4.5) 1001011010011001
5. จงแปลงจากรหัสเลขฐานสองต่อไปนี้เป็นรหัส BCD – 8421
 - 5.1) 1011101101
 - 5.2) 11011101111
 - 5.3) 10111011011
 - 5.4) 101101011101
 - 5.5) 10111010101110
6. จงแปลงจากรหัสเลขฐานสิบหกต่อไปนี้เป็นรหัส BCD – 8421
 - 6.1) AHF
 - 6.2) 147
 - 6.3) 8A4
 - 6.4) 9F
 - 6.5) 1011
7. จงเติมพาริตีคู่และพาริตีคี่ ให้กับรหัสต่อไปนี้เป็น
 - 7.1) 101110
 - 7.2) 101100
 - 7.3) 101101
 - 7.4) 100110
 - 7.5) 111111

8. จงแปลงจากรหัสเลขฐานสองต่อไปนี้เป็นรหัสเกรย์
 - 8.1) 10111011
 - 8.2) 111101110111
 - 8.3) 101101011
 - 8.4) 101111011
 - 8.5) 1011011111
9. จงแปลงจากรหัสเกรย์ต่อไปนี้เป็นรหัสเลขฐานสอง
 - 9.1) 101101111011
 - 9.2) 101101101011
 - 9.3) 1011101101111
 - 9.4) 1011110111011
 - 9.5) 1111011110111
10. จงแปลงจากรหัสเลขฐานสิบต่อไปนี้เป็นรหัสเกรย์
 - 10.1) 75
 - 10.2) 97
 - 10.3) 621
 - 10.4) 854
 - 10.5) 812
11. จงแปลงจากรหัส BCD – 8421 ต่อไปนี้เป็นรหัสเกรย์
 - 11.1) 100110000011
 - 11.2) 01110110
 - 11.3) 10000111
 - 11.4) 000100100011
 - 11.5) 01110110
12. จงแปลงจากข้อความต่อไปนี้เป็นรหัสแอสกี
 - 12.1) ELETRONIC
 - 12.2) electronic
 - 12.3) ENGINEERING
 - 12.4) engineering
 - 12.5) UDRU

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

แผนบริหารการสอนประจำบทที่ 3 เกตและไอซีพื้นฐาน 1 ชั่วโมง 30 นาที

หัวข้อเนื้อหา

- 3.1 บทนำ
- 3.2 ตารางความจริง
- 3.3 แผนผังเวลา
- 3.4 แอนด์เกต
- 3.5 ออร์เกต
- 3.6 นีตเกต
- 3.7 แนนด์เกต
- 3.8 นอร์เกต
- 3.9 เอ็กคลูซีพอร์เกต
- 3.10 เอ็กคลูซีพอร์เกต
- 3.11 คุณสมบัติของเกตพื้นฐาน
 - 3.11.1 ค่าปลอดสัญญาณรบกวน
 - 3.11.2 ความเร็วในการทำงาน
 - 3.11.3 การสูญเสียกำลัง
 - 3.11.4 ความสามารถในการต่อร่วมกัน
 - 3.11.5 กระแสซิงค์และกระแสซอร์ส
- 3.12 ไอซีพื้นฐาน
 - 3.12.1 ไอซีเบอร์ 7408
 - 3.12.2 ไอซีเบอร์ 7432
 - 3.12.3 ไอซีเบอร์ 7404
 - 3.12.4 ไอซีเบอร์ 7400
 - 3.12.5 ไอซีเบอร์ 7402
 - 3.12.6 ไอซีเบอร์ 7486
 - 3.12.7 ไอซีเบอร์ 74266
- 3.13 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการใช้งานเกตพื้นฐาน
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการใช้งานไอซีพื้นฐาน
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับตารางความจริง

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ

2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 3

เกตและไอซีพื้นฐาน

3.1 บทนำ

ในบทนี้จะกล่าวถึงเกตและไอซีพื้นฐานที่มีการใช้งานในระบบดิจิทัล โดยที่ลักษณะของเกตแต่ละชนิดนั้นจะมีอินพุตอย่างน้อย 1 ขา ส่วนเอาต์พุตจะมีเพียง 1 ขาเท่านั้น ซึ่งผลลัพธ์ทางเอาต์พุตที่ได้ ออกมาจะมีความแตกต่างกันขึ้นอยู่กับคุณสมบัติของเกตแต่ละชนิดโดยใช้ตารางความจริง (Truth Table) หรือ แผนผังเวลา (Timing Diagram) เป็นตัวบอกคุณสมบัติของเกต

กำหนดให้สถานะของอินพุตและเอาต์พุตของเกตพื้นฐานมีทั้งหมด 2 สถานะ ดังต่อไปนี้ สถานะ 0 (สายดิน หรือ กราวด์) และ สถานะ 1 (แรงดัน 5 โวลต์)

3.2 ตารางความจริง

ตารางความจริง คือ ตารางที่ใช้ในการแสดงคุณสมบัติของเอาต์พุตของเกตหรือวงจรแต่ละชนิด จากความสัมพันธ์ของอินพุต ซึ่งเอาต์พุตที่เป็นไปได้ทั้งหมดจะขึ้นอยู่กับจำนวนของอินพุต โดยจำนวนของเอาต์พุตที่เป็นไปได้ทั้งหมด คือ $2^{\text{จำนวนอินพุต}}$ เช่น มีอินพุต 3 ตัว จำนวนของผลลัพธ์ของเอาต์พุตจะมีทั้งหมด $2^3 = 8$ กรณี เป็นต้น

ตารางที่ 3.1 ตัวอย่างของตารางความจริง

| อินพุต | | เอาต์พุต |
|--------|---|----------|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

ตารางที่ 3.1 แสดงตัวอย่างของตารางความจริงที่มีอินพุต 2 ตัว เพราะฉะนั้นของผลลัพธ์ของเอาต์พุตจะมีทั้งหมด $2^2 = 4$ กรณีดังนี้

อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Y = 1

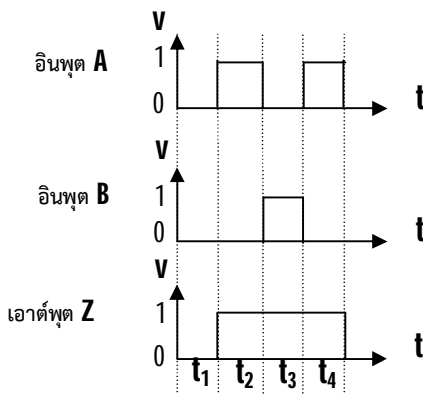
อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Y = 1

อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Y = 0

อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Y = 1

3.3 แผนผังเวลา

แผนผังเวลา คือ แผนผังที่ถูกลำนำมาใช้สำหรับพิจารณาความสัมพันธ์ระหว่างสถานะของสัญญาณอินพุตทั้งหมดและสถานะของสัญญาณเอาต์พุตทั้งหมดที่เกิดขึ้นในแต่ละช่วงเวลา



รูปที่ 3.1 ตัวอย่างแผนผังเวลา

จากรูปที่ 3.1 แสดงตัวอย่างของแผนผังเวลาจากความสัมพันธ์ของอินพุตและเอาต์พุตทั้งหมดที่เกิดขึ้นในแต่ละช่วงเวลา ดังนี้

ช่วงเวลา t_1 : อินพุต $A = 0$, อินพุต $B = 0$ จะได้เอาต์พุต $Z = 0$

ช่วงเวลา t_2 : อินพุต $A = 1$, อินพุต $B = 0$ จะได้เอาต์พุต $Z = 1$

ช่วงเวลา t_3 : อินพุต $A = 0$, อินพุต $B = 1$ จะได้เอาต์พุต $Z = 1$

ช่วงเวลา t_4 : อินพุต $A = 1$, อินพุต $B = 0$ จะได้เอาต์พุต $Z = 1$

3.4 แอนด์เกต (AND GATE)

แอนด์เกต คือเกตที่มีอินพุตอย่างน้อย 2 ขา และมีเอาต์พุตเพียง 1 ขา โดยการทำงานของแอนด์เกตคือ เอาต์พุตมีค่าเป็น 1 ได้ก็ต่อเมื่ออินพุตทุกขาจะต้องมีค่าเป็น 1 แต่หากมีอินพุตอย่างน้อย 1 ขามีค่าเป็น 0 แล้วจะทำให้เอาต์พุตมีค่าออกมาเป็น 0 ในทันที

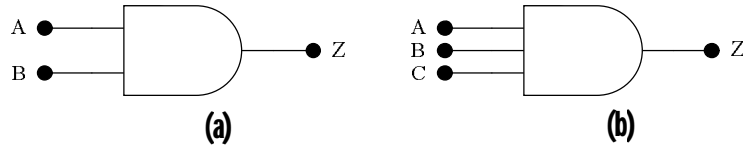
แอนด์เกตจะใช้สัญลักษณ์ "." แทนความหมายของการ แอนด์ เช่น หากสมมติว่าเป็นแอนด์เกตแบบอินพุต 2 ขา โดยมีตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = A.B$ หรือ หากเป็นแอนด์เกตแบบอินพุต 3 ขา โดยมีตัวแปร A ตัวแปร B และ ตัวแปร C เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = A.B.C$ สำหรับการเขียนสมการของการแอนด์อาจจะตัดสัญลักษณ์ "." ทิ้งได้เช่น การแอนด์แบบอินพุตที่มี 2 ขา เขียนสมการได้เป็น $Z = AB$ เป็นต้น

ตารางที่ 3.2 ตารางความจริงของแอนด์เกต

| อินพุต | | เอาต์พุต |
|--------|---|----------|
| A | B | $Z = AB$ |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

ตารางที่ 3.2 แสดงตารางความจริงของแอนด์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

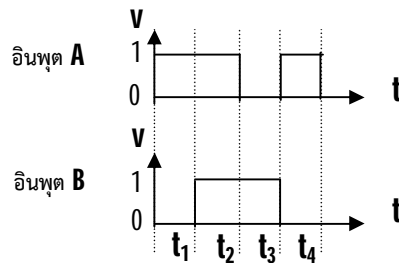
อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Z = 0
 อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Z = 0
 อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Z = 0
 อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Z = 1



รูปที่ 3.2 สัญลักษณ์ของแอนด์เกต

รูปที่ 3.2 แสดงสัญลักษณ์ของแอนด์เกต โดยที่ (a) แสดงสัญลักษณ์ของแอนด์เกตที่มีอินพุตแบบ 2 ขา และ (b) แสดงแอนด์เกตที่มีอินพุตแบบ 3 ขา

ตัวอย่างที่ 3-1 จากไดอะแกรมแสดงเวลา (Timing Diagram) ของแอนด์เกตที่มีอินพุตแบบ 2 ขา ต่อไปนี้จึงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

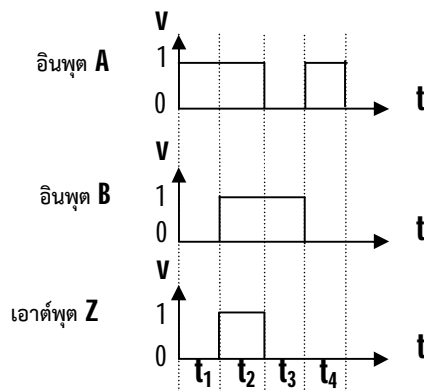
ช่วงเวลา t_1 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตเป็น 0

ช่วงเวลา t_2 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตเป็น 1

ช่วงเวลา t_3 อินพุต A มีค่าเป็น 0 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตเป็น 0

ช่วงเวลา t_4 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตเป็น 0

จากเอาต์พุตที่ได้มาในแต่ละช่วงเวลา จึงสามารถเขียนออกมาเป็นไดอะแกรมแสดงเวลาได้ดังนี้



3.5 ออร์เกต (OR GATE)

ออร์เกต คือเกตที่มีอินพุตอย่างน้อย 2 ขา และมีเอาต์พุตเพียง 1 ขาเช่นเดียวกับแอนด์เกต โดยการทำงานของออร์เกตจะทำให้เอาต์พุตมีค่าเป็น 1 ได้ก็ต่อเมื่อมีอินพุตอย่างน้อย 1 ขาที่มีค่าเป็น 1 และจะทำให้เอาต์พุตมีค่าเป็น 0 ได้ก็ต่อเมื่ออินพุตทุกขามีค่าเป็น 0

ออร์เกตจะใช้สัญลักษณ์ "+" แทนความหมายของการ ออร์ เช่น หากสมมติว่าเป็นออร์เกตแบบอินพุต 2 ขา โดยมีตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = A + B$ หรือ หากเป็นออร์เกตแบบอินพุต 3 ขา โดยมีตัวแปร A ตัวแปร B และ ตัวแปร C เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = A + B + C$

ตารางที่ 3.3 ตารางความจริงของออร์เกต

| อินพุต | | เอาต์พุต |
|--------|---|-------------|
| A | B | $Z = A + B$ |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

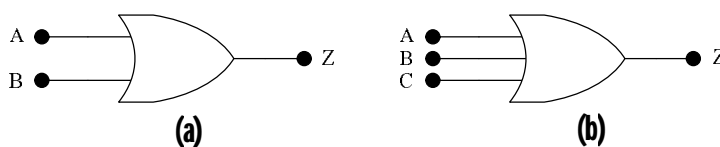
ตารางที่ 3.3 แสดงตารางความจริงของออร์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Z = 0

อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Z = 1

อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Z = 1

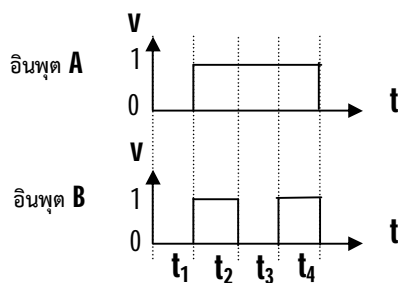
อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Z = 1



รูปที่ 3.3 สัญลักษณ์ของออร์เกต

รูปที่ 3.3 แสดงสัญลักษณ์ของออร์เกต โดยที่ (a) แสดงสัญลักษณ์ของออร์เกตที่มีอินพุตแบบ 2 ขา และ (b) แสดงออร์เกตที่มีอินพุตแบบ 3 ขา

ตัวอย่างที่ 3-2 จากไดอะแกรมแสดงเวลาของออร์เกตที่มีอินพุตแบบ 2 ขาต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

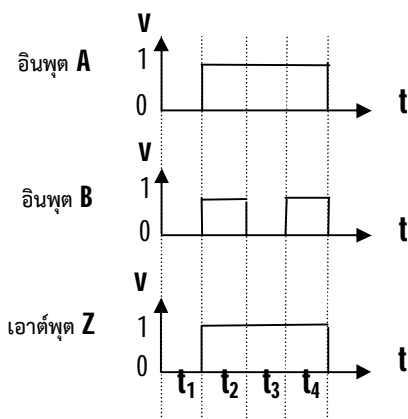
ช่วงเวลา t_1 อินพุต A มีค่าเป็น 0 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตเป็น 0

ช่วงเวลา t_2 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตเป็น 1

ช่วงเวลา t_3 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตเป็น 1

ช่วงเวลา t_4 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตเป็น 1

จากเอาต์พุตที่ได้มาในแต่ละช่วงเวลา จึงสามารถเขียนออกมาเป็นไดอะแกรมแสดงเวลาได้ดังนี้



3.6 นี้อตเกต (NOT GATE)

นี้อตเกต คือเกตที่มีอินพุต 1 ขา และมีเอาต์พุต 1 ขา โดยการทำงานของนี้อตเกตคือ เอาต์พุตจะมีค่าเป็นตรงกันข้ามกับอินพุตเสมอ เช่น หากอินพุตมีค่าเป็น 0 เอาต์พุตจะมีค่าเป็น 1 หรือ หากอินพุตมีค่าเป็น 1 เอาต์พุตจะมีค่าเป็น 0

หากสมมติว่าตัวแปร A เป็นอินพุตของนี้อตเกต เอาต์พุตของนี้อตเกตเขียนได้ดังนี้ \bar{A}

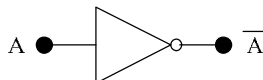
ตารางที่ 3.4 ตารางความจริงของนี้อตเกต

| อินพุต | เอาต์พุต |
|--------|-----------|
| A | \bar{A} |
| 0 | 1 |
| 1 | 0 |

ตารางที่ 3.4 แสดงตารางความจริงของนอตเกต อธิบายได้ดังนี้

อินพุต $A = 0$ จะได้เอาต์พุต $\bar{A} = 1$

อินพุต $A = 1$ จะได้เอาต์พุต $\bar{A} = 0$



รูปที่ 3.4 สัญลักษณ์ของนอตเกต

3.7 แนนด์เกต (NAND GATE)

แนนนด์เกต คือเกตที่มีอินพุตอย่างน้อย 2 ขา และมีเอาต์พุตเพียง 1 ขา โดยการทำงานของแนนนด์เกต คือเอาต์พุตมีค่าเป็น 0 ก็ต่อเมื่ออินพุตทุกขาจะต้องมีค่าเป็น 1 แต่หากมีอินพุตอย่างน้อย 1 ขามีค่าเป็น 0 แล้วจะทำให้เอาต์พุตมีค่าออกมาเป็น 1 ในทันที

หากสมมติว่าเป็นแนนนด์เกตแบบอินพุต 2 ขา โดยมีตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = \overline{A \cdot B}$ หรือ หากเป็นแนนนด์เกตแบบอินพุต 3 ขา โดยมีตัวแปร A ตัวแปร B และ ตัวแปร C เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = \overline{A \cdot B \cdot C}$

ตารางที่ 3.5 ตารางความจริงของแนนนด์เกต

| อินพุต | | เอาต์พุต |
|--------|---|----------------------------|
| A | B | $Z = \overline{A \cdot B}$ |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

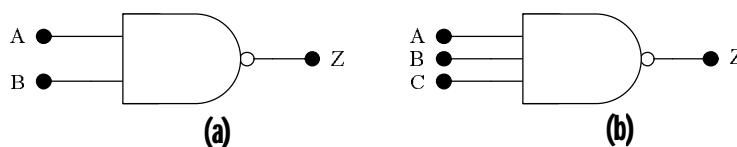
ตารางที่ 3.5 แสดงตารางความจริงของแนนนด์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

อินพุต $A = 0$, อินพุต $B = 0$ จะได้เอาต์พุต $Z = 1$

อินพุต $A = 0$, อินพุต $B = 1$ จะได้เอาต์พุต $Z = 1$

อินพุต $A = 1$, อินพุต $B = 0$ จะได้เอาต์พุต $Z = 1$

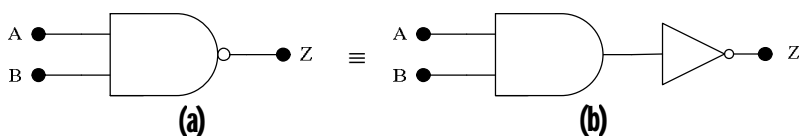
อินพุต $A = 1$, อินพุต $B = 1$ จะได้เอาต์พุต $Z = 0$



รูปที่ 3.5 สัญลักษณ์ของแนนนด์เกต

รูปที่ 3.5 แสดงสัญลักษณ์ของแอนด์เกต โดยที่ (a) แสดงสัญลักษณ์ของแอนด์เกตที่มีอินพุตแบบ 2 ขา และ (b) แสดงแอนด์เกตที่มีอินพุตแบบ 3 ขา

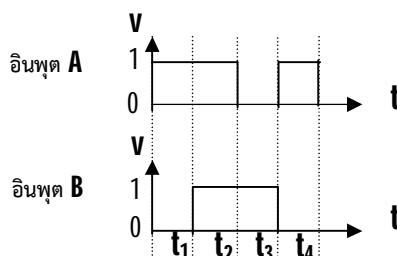
สัญลักษณ์ของแอนด์เกตจะมีความคล้ายคลึงกับสัญลักษณ์ของแอนด์เกต ต่างกันเพียงที่แอนด์เกตจะมีวงกลมวงเล็กๆที่ตำแหน่งปลายของเอาต์พุต โดยที่จริงแล้วนั้นเอาต์พุตของแอนด์เกตเป็นส่วนกลับของเอาต์พุตของแอนด์เกตนั่นเอง โดยที่ส่วนกลับของแอนด์เกตสามารถทำได้อีกวิธีโดยนำเอาต์พุตของแอนด์เกตมาเป็นอินพุตให้กับน็อตเกต



รูปที่ 3.6 เปรียบเทียบการใช้งานแอนด์เกต

รูปที่ 3.6 แสดงให้เห็นว่าเอาต์พุต Z ของแอนด์เกต (a) และ เอาต์พุต Z ที่เกิดจากการนำเอาต์พุตของแอนด์เกตมาเป็นอินพุตให้กับน็อตเกต (b) นั้นมีค่าเท่ากัน เพราะฉะนั้นหากผู้ใช้งานต้องการต่อวงจรดังรูปที่ 3.6(b) แต่ไม่มีแอนด์เกต (หรือไม่มีน็อตเกต) ผู้ใช้งานสามารถใช้แอนด์เกตแทนได้ หรือในทางกลับกัน หากผู้ใช้งานต้องการใช้งานแอนด์เกต แต่ไม่มีแอนด์เกตผู้ใช้งานสามารถต่อวงจรดังรูปที่ 3.6(b) แทนได้

ตัวอย่างที่ 3-3 จากไดอะแกรมแสดงเวลาของต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา ของแอนด์เกต และแอนด์เกต



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

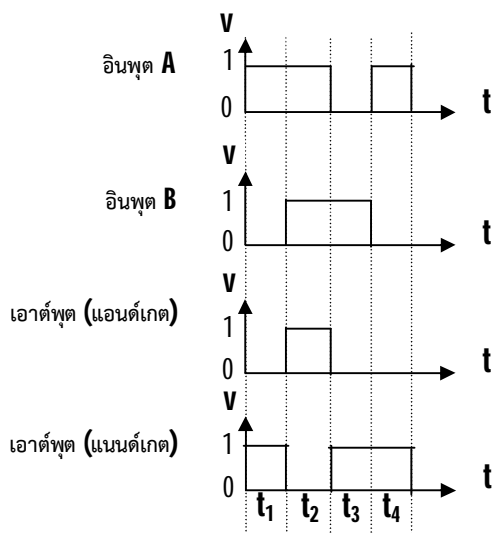
ช่วงเวลา t_1 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตแอนด์เกตเป็น 0, ได้เอาต์พุตแอนด์เกตเป็น 1

ช่วงเวลา t_2 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตแอนด์เกตเป็น 1, ได้เอาต์พุตแอนด์เกตเป็น 0

ช่วงเวลา t_3 อินพุต A มีค่าเป็น 0 อินพุต B มีค่าเป็น 1 ได้เอาต์พุตแอนด์เกตเป็น 0, ได้เอาต์พุตแอนด์เกตเป็น 1

ช่วงเวลา t_4 อินพุต A มีค่าเป็น 1 อินพุต B มีค่าเป็น 0 ได้เอาต์พุตแอนด์เกตเป็น 0, ได้เอาต์พุตแอนด์เกตเป็น 1

จากเอาต์พุตที่ได้มาในแต่ละช่วงเวลา จึงสามารถเขียนออกมาเป็นไดอะแกรมแสดงเวลาได้ ดังนี้



จากไดอะแกรมแสดงเวลาของเอาต์พุตทั้งสองแสดงให้เห็นว่าเอาต์พุตของแอนด์เกตและเอาต์พุตของแนนด์เกตจะมีค่าตรงกันข้ามกันเสมอ

ข้อดีของแนนด์เกตคือ สามารถนำแนนด์เกตไปดัดแปลงเป็นเกตชนิดอื่นๆได้ทั้งหมดไม่ว่าจะเป็น แอนด์เกต ออร์เกต นีตเกต รวมถึงเกตชนิดอื่นๆที่อย่างไม่ได้กล่าวถึง จากความสามารถดังกล่าวนี้จึงสามารถเรียกแนนด์เกตอีกชื่อหนึ่งคือ ยูนิเวอร์แซลเกต ซึ่งวิธีการนำแนนด์เกตไปต่อเป็นเกตชนิดอื่นๆ นั้นจะกล่าวถึงในบทถัดไป

3.8 นอร์เกต (NOR GATE)

นอร์เกต คือเกตที่มีอินพุตอย่างน้อย 2 ขา และมีเอาต์พุตเพียง 1 ขา โดยการทำงานของนอร์เกตคือเอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่ออินพุตทุกขาจะต้องมีค่าเป็น 0 แต่หากมีอินพุตอย่างน้อย 1 ขามีค่าเป็น 1 แล้วจะทำให้เอาต์พุตมีค่าออกมาเป็น 0 ในทันทีโดยสามารถนำนอร์เกตมาต่อเพื่อใช้งานแทนเกตชนิดอื่นๆได้เช่นเดียวกันกับ แแนนด์เกต

หากสมมติว่าเป็นนอร์เกตแบบอินพุต 2 ขา โดยมีตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = \overline{A+B}$ หรือ หากเป็นนอร์เกตแบบอินพุต 3 ขา โดยมีตัวแปร A ตัวแปร B และ ตัวแปร C เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = \overline{A+B+C}$

ตารางที่ 3.6 ตารางความจริงของนอร์เกต

| อินพุต | | เอาต์พุต |
|--------|---|-----------|
| A | B | $Z = A+B$ |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

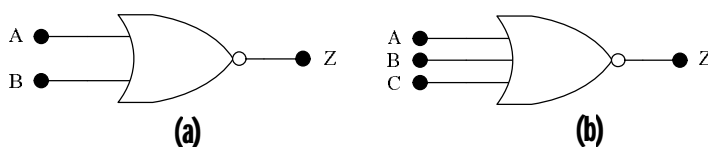
ตารางที่ 3.6 แสดงตารางความจริงของนอร์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Z = 1

อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Z = 0

อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Z = 0

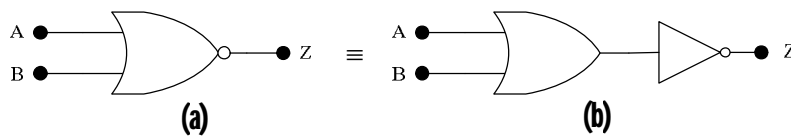
อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Z = 0



รูปที่ 3.7 สัญลักษณ์ของนอร์เกต

รูปที่ 3.7 แสดงสัญลักษณ์ของนอร์เกต โดยที่ (a) แสดงสัญลักษณ์ของนอร์เกตที่มีอินพุตแบบ 2 ขา และ (b) แสดงนอร์เกตที่มีอินพุตแบบ 3 ขา

สัญลักษณ์ของนอร์เกตจะมีความคล้ายคลึงกับสัญลักษณ์ของออร์เกต ต่างกันเพียงที่นอร์เกตจะมีวงกลมวงเล็กๆที่ตำแหน่งปลายของเอาต์พุต โดยที่จริงแล้วนั้นเอาต์พุตของนอร์เกตเป็นส่วนกลับของเอาต์พุตของออร์เกตนั่นเอง โดยที่ส่วนกลับของออร์เกตสามารถทำได้อีกวิธีโดยนำเอาต์พุตของออร์เกตมาเป็นอินพุตให้กับน็อตเกต



รูปที่ 3.8 เปรียบเทียบการใช้งานนอร์เกต

รูปที่ 3.8 แสดงให้เห็นว่าเอาต์พุต Z ของนอร์เกต (a) และ เอาต์พุต Z ที่เกิดจากการนำเอาต์พุตของออร์เกตมาเป็นอินพุตให้กับน็อตเกต (b) นั้นมีค่าเท่ากัน เพราะฉะนั้นหากผู้ใช้งานต้องการต่อวงจรดังรูปที่ 3.8(b) แต่ไม่มีออร์เกต (หรือไม่มีน็อตเกต) ผู้ใช้งานสามารถใช้นอร์เกตแทนได้ หรือในทาง

กลับกัน หากผู้ใช้งานต้องการใช้งานนอร์เกต แต่ไม่มีนอร์เกตผู้ใช้งานสามารถต่อวงจรดังรูปที่ 3.8(b) แทนได้

3.9 เอ็กคลูซีฟออร์เกต (Exclusive OR GATE)

เอ็กคลูซีฟออร์เกต หรือนิยมเรียกอีกชื่อว่า เอ็กออร์ (XOR) คือเกตที่มีอินพุต 2 ขา และมีเอาต์พุตเพียง 1 ขา โดยการทำงานของเอ็กออร์เกต คือเอาต์พุตมีค่าเป็น 0 ก็ต่อเมื่ออินพุตทุกขาจะต้องมีค่าเท่ากัน แต่หากมีอินพุตมีค่าแตกต่างกันแล้วจะทำให้เอาต์พุตมีค่าออกมาเป็น 1 ในทันที

สมมติตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = A \oplus B$

ตารางที่ 3.7 ตารางความจริงของเอ็กคลูซีฟออร์เกต

| อินพุต | | เอาต์พุต |
|--------|---|------------------|
| A | B | $Z = A \oplus B$ |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

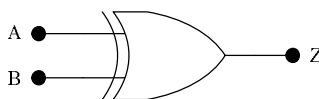
ตารางที่ 3.7 แสดงตารางความจริงของเอ็กออร์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Z = 0

อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Z = 1

อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Z = 1

อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Z = 0



รูปที่ 3.9 สัญลักษณ์ของเอ็กคลูซีฟออร์เกต

3.10 เอ็กคลูซีฟนอร์เกต (Exclusive NOR GATE)

เอ็กคลูซีฟนอร์เกต หรือนิยมเรียกอีกชื่อว่า เอ็กนอร์ (XNOR) คือเกตที่มีอินพุต 2 ขา และมีเอาต์พุตเพียง 1 ขา โดยการทำงานของเอ็กนอร์เกต คือเอาต์พุตมีค่าเป็น 0 ก็ต่อเมื่ออินพุตทุกขาจะต้องมีค่าแตกต่างกัน แต่หากอินพุตทุกตัวมีค่าเท่ากันแล้วจะทำให้เอาต์พุตมีค่าออกมาเป็น 1 ในทันที ซึ่งเอ็กนอร์เกตเปรียบเสมือนเป็นนิเสธของ เอ็กออร์เกต นั่นเอง

สมมติตัวแปร A และ ตัวแปร B เป็นอินพุต และมีตัวแปร Z เป็นเอาต์พุต จะเขียนเป็นสมการได้ดังนี้ $Z = \overline{A \oplus B}$

ตารางที่ 3.8 ตารางความจริงของเอ็กคลูซีฟนอร์เกต

| อินพุต | | เอาต์พุต |
|--------|---|------------------|
| A | B | $Z = A \oplus B$ |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

ตารางที่ 3.8 แสดงตารางความจริงของเอ็กนอร์เกตแบบอินพุต 2 ขา อธิบายได้ดังนี้

อินพุต A = 0, อินพุต B = 0 จะได้เอาต์พุต Z = 1

อินพุต A = 0, อินพุต B = 1 จะได้เอาต์พุต Z = 0

อินพุต A = 1, อินพุต B = 0 จะได้เอาต์พุต Z = 0

อินพุต A = 1, อินพุต B = 1 จะได้เอาต์พุต Z = 1



รูปที่ 3.10 สัญลักษณ์ของเอ็กคลูซีฟนอร์เกต

3.11 คุณสมบัติของเกตพื้นฐาน

คุณสมบัติที่สำคัญของเกตพื้นฐานแต่ละชนิดที่ควรทราบแบ่งออกเป็น 5 คุณสมบัติ ดังนี้

3.11.1 ค่าปลอดภัยสัญญาณรบกวน (noise immunity)

ค่าปลอดภัยสัญญาณรบกวน คือ ช่วงของแรงดันที่ยังคงยอมรับได้ว่าเป็นระดับลอจิกต่ำหรือระดับลอจิกสูงแบ่งออกเป็น 2 ประเภท ดังนี้

$$\text{ช่วงปลอดภัยสัญญาณรบกวนระดับสูง} = V_{OH,min} - V_{IH,min}$$

โดย $V_{IH,min}$ คือ ค่าต่ำสุดของแรงดันอินพุตที่ยอมรับได้ว่าเป็นระดับลอจิกสูง
 $V_{OH,min}$ คือ ค่าต่ำสุดของแรงดันเอาต์พุตที่ยอมรับได้ว่าเป็นระดับลอจิกสูง

$$\text{ช่วงปลอดภัยสัญญาณรบกวนระดับต่ำ} = V_{OL,max} - V_{IL,max}$$

โดย $V_{IL,max}$ คือ ค่าสูงสุดของแรงดันอินพุตที่ยอมรับได้ว่าเป็นระดับลอจิกต่ำ
 $V_{OL,max}$ คือ ค่าสูงสุดของแรงดันเอาต์พุตที่ยอมรับได้ว่าเป็นระดับลอจิกต่ำ

ตัวอย่างที่ 3-4 เกิดพื้นฐานชนิดหนึ่งมีค่าต่ำสุดของแรงดันอินพุตที่ยอมรับได้ว่าเป็นระดับลอจิกสูงคือ $2V$ และ ค่าต่ำสุดของแรงดันเอาต์พุตที่ยอมรับได้ว่าเป็นระดับลอจิกสูง $2.4V$ จงหาช่วงปลอดภัยสัญญาณรบกวนระดับสูง

วิธีทำ จากตัวอย่างกำหนดให้ $V_{IH, \min} = 2V$ และ $V_{OH, \min} = 2.4V$

$$\begin{aligned} \text{ได้ว่า} \quad \text{ช่วงปลอดภัยสัญญาณรบกวนระดับสูง} &= V_{OH, \min} - V_{IH, \min} \\ &= 2.4V - 2V \\ &= 0.4V \end{aligned}$$

3.11.2 ความเร็วในการทำงาน (Speed of Operation)

ความเร็วในการทำงาน คือ ช่วงเวลาที่ช่วงที่เอาต์พุตจะต้องเปลี่ยนสถานะตามอินพุตหรือที่นิยมเรียกว่าเวลาหน่วง (Delay Time)

3.11.3 การสูญเสียกำลัง (Power Dissipation)

การสูญเสียกำลัง คือ ค่าของกำลังไฟฟ้าที่เกิดพื้นฐานแต่ละชนิดต้องการเพื่อให้สามารถทำงานได้

3.11.4 ความสามารถในการต่อร่วมกัน

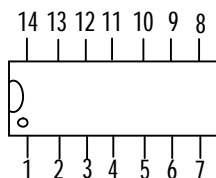
ความสามารถในการต่อร่วมกัน คือ ความสามารถสูงสุดในการรับอินพุตหรือตัวแปรที่แตกต่างกัน (Fan-In) และความสามารถสูงสุดในการต่อโหลดที่เอาต์พุตของเกต (Fan-Out)

3.11.5 กระแสซิงค์และกระแสซอร์ส (Current Sink and Current Source)

กระแสซิงค์และกระแสซอร์ส คือ การเกิดกระแสที่ไหลจากอินพุตของเกตตัวหนึ่งไปยังเอาต์พุตของเกตตัวอื่นๆ เรียกว่า กระแสซิงค์ หรือ การเกิดกระแสที่ไหลจากเอาต์พุตของเกตตัวหนึ่งไปยังอินพุตของเกตตัวอื่นๆ เรียกว่า กระแสซอร์ส

3.12 ไอซีพื้นฐาน

เนื่องจากวงจรถิทัศน์แต่ละวงจรมักจะมีการใช้งานเกตอยู่เป็นจำนวนมาก ทำให้การนำเกตแต่ละชนิดมาต่อใช้งานจะมีความยุ่งยากมากยิ่งขึ้น ดังนั้นจึงนำเกตแต่ละชนิดบรรจุลงในไอซี ซึ่งทำให้ไอซี 1 ตัวประกอบด้วยเกตอย่างน้อย 1 ชนิดที่มีจำนวนมากกว่า 1 ตัว เช่น ไอซีตระกูลทีทีแอล (TTL) เบอร์ 7400 จำนวน 1 ตัวจะประกอบไปด้วยแอนด์เกตทั้งหมด 4 ตัว เป็นต้น



รูปที่ 3.11 ตัวอย่างแบบจำลองไอซี

จากรูปที่ 3.11 แสดงตัวอย่างแบบจำลองของไอซีที่นิยมนำมาใช้เรียกว่าตัวถังแบบเส้นคู่ (Dual in Line Package: DIP) โดยตำแหน่งขาแต่ละตำแหน่งพิจารณาได้จากจุดวงกลมสีขาว ซึ่งตำแหน่งที่อยู่ติดกับจุดวงกลมสีขาวจะเป็นตำแหน่งขาหมายเลขหนึ่งและเรียงไปในทิศทางเข็มนาฬิกาจนครบทุกขา โดยตระกูลของไอซีที่ใช้งานทางด้านดิจิทัลถูกแบ่งออกเป็น 2 ชนิดหลัก คือ ไอซีตระกูลทีทีแอล (TTL) และ ไอซีตระกูลซีมอส (CMOS)

1. ไอซีตระกูลทีทีแอล: TTL ย่อมาจาก Transistor – transistor Logic เป็นไอซีที่ได้รับความนิยมมากในปัจจุบัน เนื่องจากทำงานได้รวดเร็ว และมีราคาไม่แพง โดยไอซีตระกูลนี้จะมีหมายเลขขึ้นต้นด้วย “74” หรือ “54” เสมอ และตัวเลขที่ตามหลังจะเป็นตัวบอกว่าเป็นไอซีทีทีแอลชนิดใด เช่น 7400 เป็นไอซีชนิดแอนด์เกต หรือ 7408 เป็นไอซีชนิดแอนด์เกต เป็นต้น โดยไอซีตระกูลทีทีแอลจะถูกแบ่งออกเป็นทั้งหมด 6 กลุ่ม ดังนี้

1.1 ทีทีแอลมาตรฐาน: ขึ้นต้นด้วย 54 หรือ 74 โดยใช้ความถี่สูงสุด 20MHz

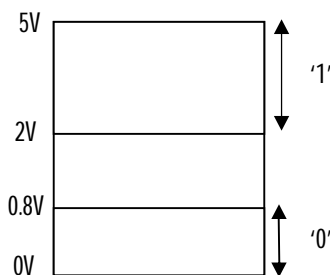
1.2 ทีทีแอลกำลังสูญเสียต่ำ: ขึ้นต้นด้วย 54L หรือ 74L โดยไอซีชนิดนี้จะมีอัตราการสูญเสียกำลังไฟฟ้าที่ต่ำกว่าทีทีแอลมาตรฐานถึง 10 เท่า

1.3 ทีทีแอลความเร็วสูง: ขึ้นต้นด้วย 54H หรือ 74H โดยไอซีชนิดนี้จะมีความเร็วสูงกว่าไอซีทีทีแอลแบบมาตรฐานซึ่งใช้เวลาในการเปลี่ยนสถานะเพียง 6ns แต่มีอัตราการสูญเสียกำลังไฟฟ้าที่ต่ำกว่าไอซีทีทีแอลมาตรฐาน

1.4 ขอตต์กีกำลังต่ำ: ขึ้นต้นด้วย 74LS เป็นไอซีที่ใช้กำลังไฟฟ้าน้อยและมีความเร็วที่สูงเมื่อเปรียบเทียบกับไอซีทีทีแอลความเร็วสูง

1.5 แอควานซ์ขอตต์กีกำลังต่ำทีทีแอล: ขึ้นต้นด้วย 74ALS กินกำลังไฟฟ้าน้อยกว่าขอตต์กีกำลังต่ำโดยไอซีชนิดนี้จะกินกำลังไฟฟ้าเพียง 1mW และมีเวลาหน่วงเหลือเพียง 4ns ต่อเกต 1 ตัว โดยไอซีชนิดนี้เป็นไอซีที่พัฒนาต่อมาจากขอตต์กีกำลังต่ำ

1.6 ฟาสต์ทีทีแอล: ไอซีที่มีความเร็วสูงกว่า แอควานซ์ขอตต์กีกำลังต่ำทีทีแอล โดยจะขึ้นต้นด้วย 74F



รูปที่ 3.12 ระดับลอจิกของไอซีตระกูลทีทีแอล

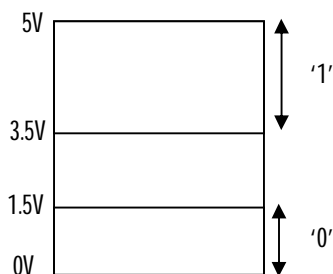
จากรูปที่ 3.12 แสดงระดับลอจิกของไอซีตระกูลทีทีแอลโดยที่ค่าแรงดันที่ถูกนำไปเป็นไฟเลี้ยงให้ ไอซีทีทีแอลจะต้องไม่เกิน 5V หากแรงดันที่ป้อนเข้าสู่ไอซีมีค่าอยู่ระหว่าง 0 – 1.5V จะถูกกำหนดให้เป็นระดับลอจิกต่ำ หรือ “0” หากแรงดันที่ป้อนเข้าสู่ไอซีมีค่าอยู่ระหว่าง 2 – 5V จะถูกกำหนดให้เป็นระดับลอจิกสูง หรือ “1” แต่หากเป็นแรงดันที่อยู่ในช่วง 0.8 – 2V จะถูกกำหนดให้เป็นระดับลอจิกที่ไม่แน่นอน

2. ไอซีตระกูลซีมอส: CMOS ย่อมาจาก **Complementary Metal Oxide Silicon** เป็นไอซีที่ใช้ทรานซิสเตอร์แบบเฟตเป็นตัวออกแบบวงจรภายใน จุดเด่นคือวงจรมีขนาดเล็ก และใช้พลังงานไฟฟ้าต่ำ แต่อย่างไรก็ตามซีมอสจะมีความเร็วที่ต่ำและมีชนิดของวงจรให้เลือกใช้งานน้อย เมื่อเปรียบเทียบกับไอซีตระกูลทีทีแอล โดยไอซีตระกูลซีมอสจะถูกแบ่งเป็นกลุ่ม ดังนี้

2.1 ซีมอสมาตรฐาน: ไอซีอนุกรมแบบ 4000 โดยจะขึ้นต้นด้วย 40XX และ 45XX เสมอ

2.2 ซีมอสความเร็วสูง: ขึ้นต้นด้วย 74HC หรือ 74HCT ซึ่งสังเกตได้ว่าจะเป็นเลขที่ขึ้นต้นเหมือนไอซีตระกูลทีทีแอล ที่เป็นเช่นนี้เนื่องจากว่าไอซีประเภทนี้มีฟังก์ชันการทำงานคล้ายกับไอซีทีทีแอลที่ขึ้นต้นด้วย 74 เพียงแต่มีโครงสร้างแบบซีมอสที่มีความถี่สูงถึง 35MHz โดยที่หากเป็นไอซีเบอร์ 74HC สัญญาณทางอินพุตจะเป็นระดับสัญญาณที่อยู่ในระดับซีมอส (รูปที่ 3.13) แต่สัญญาณเอาต์พุตสามารถให้ได้ทั้งระดับสัญญาณของซีมอสและทีทีแอล ไอซีชนิดนี้ใช้กำลังไฟฟ้าประมาณ 1 mW ต่อเกต 1 ตัว แต่หากเป็นไอซีเบอร์ 74HCT สัญญาณทางอินพุตจะเป็นระดับสัญญาณที่อยู่ในระดับทีทีแอล แต่สัญญาณเอาต์พุตสามารถให้ได้ทั้งระดับสัญญาณของซีมอสและทีทีแอล เนื่องจากช่วงของแรงดันสำหรับการแบ่งระดับสัญญาณของไอซีทีทีแอลและซีมอสมีความแตกต่างกัน ดังนั้นสำหรับไอซีชนิดนี้ควรกำหนดให้มีแรงดัน 0V สำหรับกรณีระดับสัญญาณต่ำ และกำหนดให้มีแรงดัน 5V ในกรณีระดับลอจิกสูง

2.3 แอดวานซ์ซีมอสความเร็วสูง: ขึ้นต้นด้วย 74AC และ 74ACT ซึ่งไอซีชนิดนี้จะมีความเร็วสูงกว่าไอซีกลุ่มซีมอสความเร็วสูงเป็นอย่างมาก



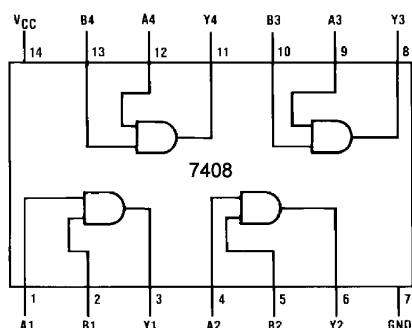
รูปที่ 3.13 ระดับลอจิกของไอซีตระกูลซีมอส

จากรูปที่ 3.13 แสดงระดับลอจิกของไอซีตระกูลซีมอสโดยที่ค่าแรงดันที่ถูกนำไปเป็นไฟเลี้ยงให้ไอซีซีมอสจะต้องไม่เกิน 5V หากแรงดันที่ป้อนเข้าสู่ไอซีมีค่าอยู่ระหว่าง 0 – 1.5V จะถูกกำหนดให้เป็นระดับลอจิกต่ำ หรือ “0” หากแรงดันที่ป้อนเข้าสู่ไอซีมีค่าอยู่ระหว่าง 3.5 – 5V จะถูกกำหนดให้เป็นระดับลอจิกสูง หรือ “1” แต่หากเป็นแรงดันที่อยู่ในช่วง 1.5 – 3.5V จะถูกกำหนดให้เป็นระดับลอจิกที่ไม่แน่นอน

จากไอซีทั้ง 2 ประเภทสรุปได้ว่าหมายเลขขึ้นต้น 2 ตัวแรกจะแสดงตระกูลของไอซี เช่น 74 คือไอซีตระกูลทีทีแอล (หรือซีมอสบางประเภท) หรือ 40 คือไอซีตระกูลซีมอส อย่างไรก็ตามตัวเลขกลุ่มหลังจะแสดงถึงประเภทของไอซี เช่น ไอซีเบอร์ 7408 ความหมายเป็นดังนี้ 74 แสดงให้เห็นว่าไอซีตัวนี้คือไอซีตระกูลทีทีแอล และ 08 คือไอซีที่บรรจุแอนเกตแบบ 2 อินพุตจำนวน 4 ตัว เป็นต้น โดยตัวอย่างไอซีที่บรรจุเกตพื้นฐานมีดังต่อไปนี้

3.12.1 ไอซีเบอร์ 7408

ไอซีเบอร์ 7408 เป็นไอซีที่บรรจุแอนด์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 7408 จำนวน 1 ตัวจะประกอบด้วยแอนด์เกตทั้งหมด 4 ตัว



รูปที่ 3.14 ไอซีเบอร์ 7408

ที่มาของภาพ: <https://faculty-web.msoe.edu/tritt/hs/hsdig01.html>

รูปที่ 3.14 แสดงโครงสร้างของไอซีเบอร์ 7408 ซึ่งมีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ VCC (ขา 14) และขากาวต่ออยู่ที่ขา GND (ขา 7) การต่อใช้งานแอนด์เกตทำได้ดังนี้

การใช้งานแอนด์เกตตัวที่ 1: ต่ออินพุตที่ขา A1 (ขา 1) และขา B1 (ขา 2) และต่อเอาต์พุตที่ขา Y1 (ขา 3)

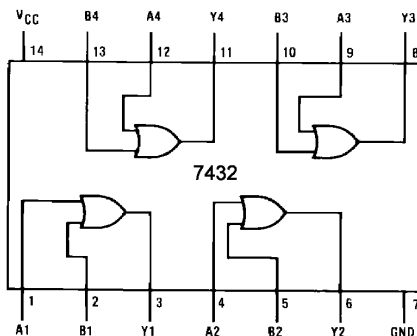
การใช้งานแอนด์เกตตัวที่ 2: ต่ออินพุตที่ขา A2 (ขา 4) และขา B2 (ขา 5) และต่อเอาต์พุตที่ขา Y2 (ขา 6)

การใช้งานแอนด์เกตตัวที่ 3: ต่ออินพุตที่ขา A3 (ขา 9) และขา B3 (ขา 10) และต่อเอาต์พุตที่ขา Y3 (ขา 8)

การใช้งานแอนด์เกตตัวที่ 4: ต่ออินพุตที่ขา A4 (ขา 12) และขา B4 (ขา 13) และต่อเอาต์พุตที่ขา Y4 (ขา 11)

3.12.2 ไอซีเบอร์ 7432

ไอซีเบอร์ 7432 เป็นไอซีที่บรรจุออร์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 7432 จำนวน 1 ตัวจะประกอบด้วยออร์เกตทั้งหมด 4 ตัว



รูปที่ 3.15 ไอซีเบอร์ 7432

ที่มาของภาพ: <https://faculty-web.msoe.edu/tritt/hs/hsdig01.html>

รูปที่ 3.15 แสดงโครงสร้างของไอซีเบอร์ 7432 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ VCC (ขา 14) และขากราวด์อยู่ที่ขา GND (ขา 7) การต่อใช้งานออร์เกตทำได้ดังนี้

การใช้งานออร์เกตตัวที่ 1: ต่ออินพุตที่ขา A1 (ขา 1) และขา B1 (ขา 2) และต่อเอาต์พุตที่ขา Y1 (ขา 3)

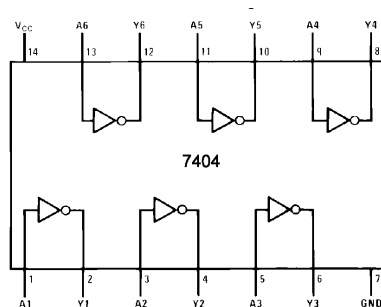
การใช้งานออร์เกตตัวที่ 2: ต่ออินพุตที่ขา A2 (ขา 4) และขา B2 (ขา 5) และต่อเอาต์พุตที่ขา Y2 (ขา 6)

การใช้งานออร์เกตตัวที่ 3: ต่ออินพุตที่ขา A3 (ขา 9) และขา B3 (ขา 10) และต่อเอาต์พุตที่ขา Y3 (ขา 8)

การใช้งานออร์เกตตัวที่ 4: ต่ออินพุตที่ขา A4 (ขา 12) และขา B4 (ขา 13) และต่อเอาต์พุตที่ขา Y4 (ขา 11)

3.12.3 ไอซีเบอร์ 7404

ไอซีเบอร์ 7404 เป็นไอซีที่บรรจุรีตเกตอยู่ภายในโดยที่ไอซีเบอร์ 7404 จำนวน 1 ตัว จะประกอบด้วยรีตเกตทั้งหมด 6 ตัว



รูปที่ 3.16 ไอซีเบอร์ 7404

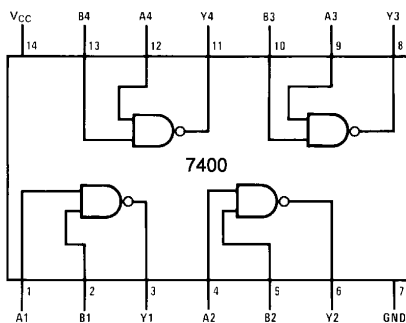
ที่มาของภาพ: <https://faculty-web.msoe.edu/tritt/hs/hsdig01.html>

รูปที่ 3.16 แสดงโครงสร้างของไอซีเบอร์ 7404 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ VCC (ขา 14) และขากราวด์อยู่ที่ขา GND (ขา 7) การต่อใช้งานรีตเกตทำได้ดังนี้

- การใช้งานนอตเกตตัวที่ **1**: ต่ออินพุตที่ขา A1 (ขา 1) และต่อเอาต์พุตที่ขา Y1 (ขา 2)
 การใช้งานนอตเกตตัวที่ **2**: ต่ออินพุตที่ขา A2 (ขา 3) และต่อเอาต์พุตที่ขา Y2 (ขา 4)
 การใช้งานนอตเกตตัวที่ **3**: ต่ออินพุตที่ขา A3 (ขา 5) และต่อเอาต์พุตที่ขา Y3 (ขา 6)
 การใช้งานนอตเกตตัวที่ **4**: ต่ออินพุตที่ขา A4 (ขา 9) และต่อเอาต์พุตที่ขา Y4 (ขา 8)
 การใช้งานนอตเกตตัวที่ **5**: ต่ออินพุตที่ขา A5 (ขา 11) และต่อเอาต์พุตที่ขา Y5 (ขา 10)
 การใช้งานนอตเกตตัวที่ **6**: ต่ออินพุตที่ขา A6 (ขา 13) และต่อเอาต์พุตที่ขา Y6 (ขา 12)

3.12.4 ไอซีเบอร์ 7400

ไอซีเบอร์ 7400 เป็นไอซีที่บรรจุแชนด์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 7400 จำนวน 1 ตัวจะประกอบด้วยแชนด์เกตทั้งหมด 4 ตัว



รูปที่ 3.17 ไอซีเบอร์ 7400

ที่มาของภาพ: <https://faculty-web.msoe.edu/tritt/hs/hsdig01.html>

รูปที่ 3.17 แสดงโครงสร้างของไอซีเบอร์ 7400 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ขา VCC (ขา 14) และขากาวด์อยู่ที่ขา GND (ขา 7) การต่อใช้งานแชนด์เกตทำได้ดังนี้

การใช้งานแชนด์เกตตัวที่ **1**: ต่ออินพุตที่ขา A1 (ขา 1) และขา B1 (ขา 2) และต่อเอาต์พุตที่ขา Y1 (ขา 3)

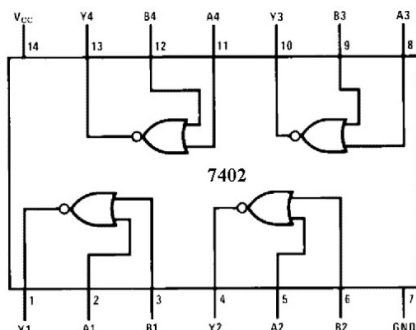
การใช้งานแชนด์เกตตัวที่ **2**: ต่ออินพุตที่ขา A2 (ขา 4) และขา B2 (ขา 5) และต่อเอาต์พุตที่ขา Y2 (ขา 6)

การใช้งานแชนด์เกตตัวที่ **3**: ต่ออินพุตที่ขา A3 (ขา 9) และขา B3 (ขา 10) และต่อเอาต์พุตที่ขา Y3 (ขา 8)

การใช้งานแชนด์เกตตัวที่ **4**: ต่ออินพุตที่ขา A4 (ขา 12) และขา B4 (ขา 13) และต่อเอาต์พุตที่ขา Y4 (ขา 11)

3.12.5 ไอซีเบอร์ 7402

ไอซีเบอร์ 7402 เป็นไอซีที่บรรจุอินเวอร์เตอร์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 7402 จำนวน 1 ตัวจะประกอบด้วยอินเวอร์เตอร์เกตทั้งหมด 4 ตัว



รูปที่ 3.18 ไอซีเบอร์ 7402

ที่มาของภาพ: <http://www.hardwaresecrets.com/article/Introduction-to-Logic-Gates/237/6>

รูปที่ 3.18 แสดงโครงสร้างของไอซีเบอร์ 7402 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ขา VCC (ขา 14) และขากาวรัดอยู่ที่ขา GND (ขา 7) การต่อใช้งานนอร์เกตทำได้ดังนี้

การใช้งานนอร์เกตตัวที่ 1: ต่ออินพุตที่ขา A1 (ขา 2) และขา B1 (ขา 3) และต่อเอาต์พุตที่ขา Y1 (ขา 1)

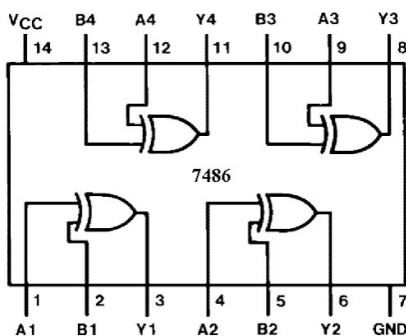
การใช้งานนอร์เกตตัวที่ 2: ต่ออินพุตที่ขา A2 (ขา 5) และขา B2 (ขา 6) และต่อเอาต์พุตที่ขา Y2 (ขา 4)

การใช้งานนอร์เกตตัวที่ 3: ต่ออินพุตที่ขา A3 (ขา 8) และขา B3 (ขา 9) และต่อเอาต์พุตที่ขา Y3 (ขา 10)

การใช้งานนอร์เกตตัวที่ 4: ต่ออินพุตที่ขา A4 (ขา 11) และขา B4 (ขา 12) และต่อเอาต์พุตที่ขา Y4 (ขา 13)

3.12.6 ไอซีเบอร์ 7486

ไอซีเบอร์ 7486 เป็นไอซีที่บรรจุเอ็กออร์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 7486 จำนวน 1 ตัวจะประกอบด้วยเอ็กออร์เกตทั้งหมด 4 ตัว



รูปที่ 3.19 ไอซีเบอร์ 7486

ที่มาของภาพ: <http://www.hardwaresecrets.com/article/Introduction-to-Logic-Gates/237/7>

รูปที่ 3.19 แสดงโครงสร้างของไอซีเบอร์ 7486 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ขา VCC (ขา 14) และขากราวด์อยู่ที่ขา GND (ขา 7) การต่อใช้งานเอ็กออร์เกตทำได้ดังนี้

การใช้งานเอ็กออร์เกตตัวที่ 1: ต่ออินพุตที่ขา A1 (ขา 1) และขา B1 (ขา 2) และต่อเอาต์พุตที่ขา Y1 (ขา 3)

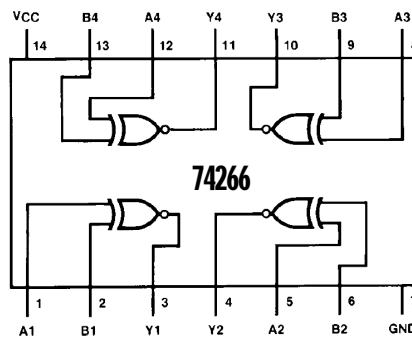
การใช้งานเอ็กออร์เกตตัวที่ 2: ต่ออินพุตที่ขา A2 (ขา 4) และขา B2 (ขา 5) และต่อเอาต์พุตที่ขา Y2 (ขา 6)

การใช้งานเอ็กออร์เกตตัวที่ 3: ต่ออินพุตที่ขา A3 (ขา 9) และขา B3 (ขา 10) และต่อเอาต์พุตที่ขา Y3 (ขา 8)

การใช้งานเอ็กออร์เกตตัวที่ 4: ต่ออินพุตที่ขา A4 (ขา 12) และขา B4 (ขา 13) และต่อเอาต์พุตที่ขา Y4 (ขา 11)

3.12.7 ไอซีเบอร์ 74266

ไอซีเบอร์ 74266 เป็นไอซีที่บรรจุเอ็กออร์เกตแบบ 2 อินพุตอยู่ภายในโดยที่ไอซีเบอร์ 74266 จำนวน 1 ตัวจะประกอบด้วยเอ็กออร์เกตทั้งหมด 4 ตัว



รูปที่ 3.20 ไอซีเบอร์ 74266

ที่มาของภาพ: <http://www.hardwaresecrets.com/article/Introduction-to-Logic-Gates/237/8>

รูปที่ 3.20 แสดงโครงสร้างของไอซีเบอร์ 74266 มีทั้งหมด 14 ขาโดยขาไฟเลี้ยงอยู่ที่ขา VCC (ขา 14) และขากราวด์อยู่ที่ขา GND (ขา 7) การต่อใช้งานเอ็กออร์เกตทำได้ดังนี้

การใช้งานเอ็กออร์เกตตัวที่ 1: ต่ออินพุตที่ขา A1 (ขา 1) และขา B1 (ขา 2) และต่อเอาต์พุตที่ขา Y1 (ขา 3)

การใช้งานเอ็กออร์เกตตัวที่ 2: ต่ออินพุตที่ขา A2 (ขา 5) และขา B2 (ขา 6) และต่อเอาต์พุตที่ขา Y2 (ขา 4)

การใช้งานเอ็กออร์เกตตัวที่ 3: ต่ออินพุตที่ขา A3 (ขา 8) และขา B3 (ขา 9) และต่อเอาต์พุตที่ขา Y3 (ขา 10)

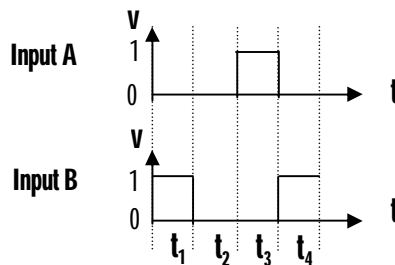
การใช้งานเอ็กออร์เกตตัวที่ 4: ต่ออินพุตที่ขา A4 (ขา 12) และขา B4 (ขา 13) และต่อเอาต์พุตที่ขา Y4 (ขา 11)

3.13 บทสรุป

เกตดิจิทัล คืออุปกรณ์ที่มีอินพุตอย่างน้อย 1 ขา และเอาต์พุต 1 ขา โดยสถานะของเอาต์พุตจะขึ้นอยู่กับคุณสมบัติของเกตแต่ละชนิด โดยในบทนี้กล่าวถึงเกตพื้นฐานทั้งหมด 7 ประเภท ประกอบด้วย แอนด์เกต ออร์เกต นีตเกต นอร์เกต แนนด์เกต เอ็กออร์เกต และ เอ็กนอร์เกต ซึ่งเกตแต่ละชนิดจะถูกบรรจุไว้ในไอซี โดยไอซี 1 ตัวจะมีเกตพื้นฐานชนิดหนึ่งบรรจุไว้ภายในมากกว่า 1 ตัว เช่น ไอซีเบอร์ 7408 จำนวน 1 ตัวจะประกอบไปด้วยแอนด์เกตทั้งหมด 4 ตัว เป็นต้น

คำถามท้ายบท

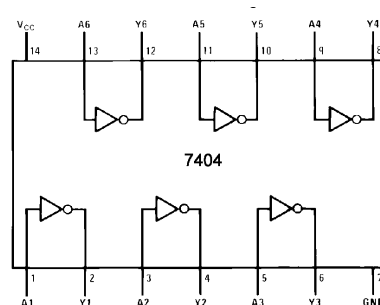
1. จากไดอะแกรมแสดงเวลา (Timing Diagram) ที่มีอินพุตแบบ 2 ขาต่อไปนี้



จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลาเมื่อเกตที่ใช้งานคือ

- 1.1) แอนด์เกต
- 1.2) ออร์เกต
- 1.3) แนนด์เกต
- 1.4) นอร์เกต
- 1.5) เอ็กออร์เกต
- 1.6) เอ็กนอร์เกต

2. จากรูปที่กำหนดให้ต่อไปนี้

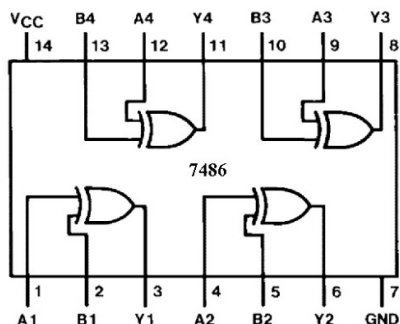


จงหาสถานะ Y_3 (ขา 6) เมื่ออินพุตคือ A_1 (ขา 1) โดยกำหนดให้

- $A_1 = 1$ (5V)
- ต่อ Y_1 (ขา 2) ร่วมกับ A_2 (ขา 3)

- ต่อ Y_2 (ขา 4) ร่วมกับ A_3 (ขา 5)
- ต่อขา 7 ลงสายดิน
- ต่อขา 14 กับไฟเลี้ยง 5V

3. จากรูปที่กำหนดให้ต่อไปนี้



จงหาสถานะ Y_3 (ขา 6) โดยกำหนดให้

- A_3 (ขา 9) = 1 (5V)
- B_3 (ขา 10) = 1 (5V)
- ต่อขา 7 ลงสายดิน
- ต่อขา 14 กับไฟเลี้ยง 5V

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- รัชชัย เลื่อนฉวี และ อนุรักษ์ เกื้อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.
- <https://faculty-web.msoe.edu/tritt/hs/hsdig01.html>
- <http://www.hardwaresecrets.com>

แผนบริหารการสอนประจำบทที่ 4 พีชคณิตบูลีน 3 ชั่วโมง

หัวข้อเนื้อหา

- 4.1 บทนำ
- 4.2 การสร้างวงจรถอกจากสมการพีชคณิตบูลีน
- 4.3 การหาสถานะเอาต์พุตโดยใช้ตารางความจริง
- 4.4 กฎพื้นฐานพีชคณิตบูลีน
 - 4.4.1 กฎการสลับที่
 - 4.4.2 กฎการเปลี่ยนกลุ่ม
 - 4.4.3 กฎการแจกแจง
 - 4.4.4 กฎไอดีมโปเทน
 - 4.4.5 กฎนิเสธสองครั้ง
 - 4.4.6 กฎการแอนด์
 - 4.4.7 กฎการออร์
 - 4.4.8 กฎคอมพลิเมนต์
 - 4.4.9 กฎความซับซ้อน
 - 4.4.10 กฎเดออร์มอร์แกน
- 4.5 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนสามารถสร้างวงจรถอกจากสมการพีชคณิตบูลีนได้
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการหาสถานะเอาต์พุตของวงจรถอโดยใช้ตารางความจริง
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับกฎพื้นฐานพีชคณิตบูลีน

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรถิทัศน์และลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 4 พีชคณิตบูลีน

4.1 บทนำ

พีชคณิตบูลีนเป็นการดำเนินการทางคณิตศาสตร์ในรูปแบบตรรกะที่สามารถทำให้ทราบถึงสถานะต่าง ๆ ในกรณีที่นำลอจิกเกตมากกว่าหนึ่งตัวมาต่อร่วมกัน สำหรับเนื้อหาในบทนี้จะกล่าวถึงหัวข้อที่สำคัญดังนี้ การสร้างวงจรจากสมการพีชคณิตบูลีน การหาสถานะเอาต์พุตของวงจรโดยใช้ตารางความจริงและกฎพื้นฐานพีชคณิตบูลีน

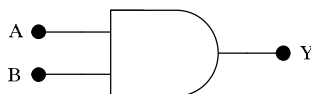
4.2 การสร้างวงจรจากสมการพีชคณิตบูลีน

การสร้างวงจรจากสมการพีชคณิตบูลีน คือการนำสมการพีชคณิตบูลีนมาสร้างเป็นวงจรโดยจะต้องสร้างตามลำดับความสำคัญทางลอจิกดังต่อไปนี้

1. อินพุตใดๆ ที่มีน๊อต และมีตัวดำเนินการกับอินพุตตัวอื่นจะต้องใช้ตัวดำเนินการน๊อตก่อนเสมอ เช่น \overline{AB} จะต้องนำ A ผ่านน๊อตเกตก่อนแล้วจึงนำผลลัพธ์ที่ได้ไปแอนด์กับ B แต่หากเป็นกรณีที่ใช้ตัวดำเนินการน๊อตกับอินพุต 2 ตัวที่ดำเนินการกันอยู่จะต้องดำเนินการระหว่างอินพุต 2 ตัวก่อนแล้วจึงนำมาผ่านน๊อตเกต เช่น \overline{AB} จะต้องนำ A และ B มาแอนด์กันก่อนแล้วจึงนำมาผ่านน๊อตเกต
2. แอนด์มีลำดับความสำคัญสูงกว่า ออร์
3. กรณีสี่วงเล็บต้องกระทำภายในวงเล็บก่อนเสมอ

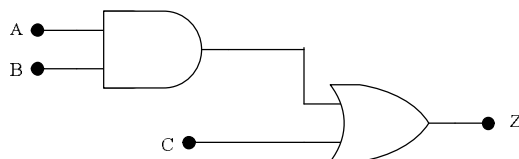
ตัวอย่างที่ 4-1 จงสร้างวงจรจากสมการพีชคณิตบูลีนต่อไปนี้ $Z = AB + C$

วิธีทำ จากสมการพีชคณิตบูลีนพบว่ามีการใช้ตัวดำเนินการแอนด์ และ ออร์ เนื่องจาก แอนด์ มีลำดับความสำคัญมากกว่า ออร์ ดังนั้นจึงนำ A และ B มา แอนด์ กันก่อนโดยใช้แอนด์เกตได้ดังนี้



สถานะของ $Y = AB$

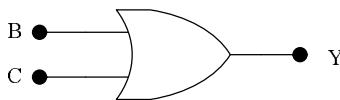
ขั้นตอนสุดท้ายเป็นการนำเอาต์พุตที่ได้จาก แอนด์ (Y) มา ออร์ กับ C ดังนี้



สถานะของ $Z = AB + C$

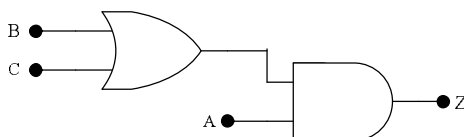
ตัวอย่างที่ 4-2 จงสร้างวงจรจากสมการพีชคณิตบูลีนต่อไปนี้ $Z = A(B + C)$

วิธีทำ จากสมการพีชคณิตบูลีนพบว่ามีการใช้ตัวดำเนินการแบบ แอนด์ และ แบบ ออร์ แต่เนื่องจากตัวอย่งนี้มีวงเล็บ ดังนั้นจึงดำเนินการภายในวงเล็บก่อน นั่นคือ $B + C$



สถานะของ $Y = B + C$

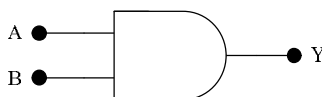
ขั้นตอนสุดท้ายเป็นการนำเอาต์พุตที่ได้จาก ออร์ (Y) มา แอนด์ กับ A ดังนี้



สถานะของ $Z = A(B+C)$

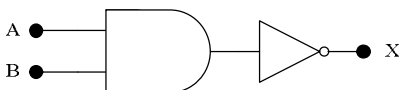
ตัวอย่างที่ 4-3 จงสร้างวงจรจากสมการพีชคณิตบูลีนต่อไปนี้ $Z = ABC$ โดยใช้ แอนด์เกต ออร์เกต หรือ นีตเกตเท่านั้น

วิธีทำ จากสมการพีชคณิตบูลีนพบว่ามีนีตเกตอยู่ที่ผลลัพธ์ของการนำ A มาผ่านกระบวนการแอนด์กับ B ดังนั้นจึงต้องดำเนินการของอินพุต A และ B ก่อนได้ดังนี้



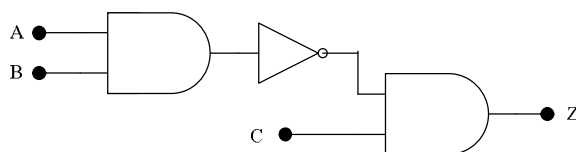
สถานะของ $Y = AB$

ขั้นตอนต่อไปเป็นการนำผลลัพธ์ที่ได้จากการแอนด์มาผ่านนีตเกต ดังนี้



สถานะของ $X = \overline{AB}$

ขั้นตอนสุดท้ายเป็นการนำเอาต์พุต X มาเป็นอินพุตเพื่อแอนด์กับ C ดังนี้



สถานะของ $Z = ABC$

4.3 การหาสถานะเอาต์พุตโดยใช้ตารางความจริง

ตารางความจริงสามารถถูกนำมาใช้สำหรับ การหาสถานะที่เป็นไปได้ทั้งหมดของเอาต์พุตจากสมการพีชคณิตบูลีน โดยจากสมการพีชคณิตบูลีนที่กำหนดให้ จะต้องคำนวณหาการดำเนินการระหว่างอินพุตที่มีลำดับความสำคัญสูงสุดก่อนเสมอ (แต่หากมีการดำเนินการระหว่างอินพุตที่มีระดับความสำคัญเท่ากัน สามารถเลือกคำนวณคู่ใดก่อนหลังได้) และคำนวณระดับความสำคัญที่ต่ำลงมาจนได้ผลลัพธ์ที่เป็นสถานะทั้งหมดของเอาต์พุตของสมการพีชคณิตบูลีนดังกล่าว

ตัวอย่างที่ 4-4 จงคำนวณหาผลลัพธ์ที่เป็นไปได้ทั้งหมดจากสมการ $Z = A(B+C)$ โดยใช้ตารางความจริง

วิธีทำ เนื่องจากมีอินพุตทั้งหมด 3 ตัว สถานะเอาต์พุตที่เป็นไปได้จึงมีทั้งหมด 8 กรณี จากสมการพีชคณิตบูลีนมีการดำเนินการภายในวงเล็บนั้นคือ $B+C$ เพราะฉะนั้นคำนวณหาสถานะเอาต์พุตของสมการดังกล่าวก่อน ขั้นตอนสุดท้ายนำ A มาแอนด์ กับ $B+C$

ตารางที่ 4.1 การหาสถานะเอาต์พุตที่เป็นไปได้ทั้งหมดจากสมการ $Z = A(B+C)$

| A | B | C | B+C | Z = A(B+C) |
|---|---|---|-----|------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

4.4 กฎพื้นฐานพีชคณิตบูลีน

กฎพื้นฐานพีชคณิตบูลีน เป็นสิ่งสำคัญที่จะช่วยให้สมการพีชคณิตบูลีนที่ต้องการนำไปสร้างเป็นวงจรมีอยู่ในรูปที่สั้นลง และง่ายมากยิ่งขึ้น นั่นคือเมื่อสมการมีขนาดที่สั้นลง ส่งผลให้จำนวนเกตที่ต้องนำมาใช้งานมีน้อยลงด้วย โดยที่สถานะของเอาต์พุตที่เป็นไปได้ทั้งหมดยังคงเดิม โดยกฎพื้นฐานที่สำคัญประกอบด้วยดังนี้

4.4.1 กฎการสลับที่ (Commutative Laws)

1. $A + B = B + A$
2. $A.B = B.A$

กฎการสลับที่ คือ การเปลี่ยนแปลงตำแหน่งของอินพุต จะไม่ส่งผลต่อสถานะของเอาต์พุต ซึ่งกฎข้อนี้สามารถพิสูจน์ได้ โดยใช้ตารางความจริง

ตารางที่ 4.2 คุณสมบัติการสลับที่ภายใต้การ ออร์

| A | B | A + B | B + A |
|---|---|-------|-------|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 |

4.4.2 กฎการเปลี่ยนกลุ่ม (Associative Laws)

1. $(A + B) + C = A + (B + C)$
2. $(A.B).C = A.(B.C)$

กฎการเปลี่ยนกลุ่ม คือ การที่สามารถเลือกดำเนินการกับอินพุตคู่ใดก่อนได้ โดยไม่ส่งผลกระทบต่อสถานะของเอาต์พุต แต่ตัวดำเนินการจะต้องเหมือนกันทั้งหมด แต่หากเป็นกรณีที่ตัวดำเนินการไม่เหมือนกัน หากเปลี่ยนกลุ่มการดำเนินการจะทำให้สถานะของเอาต์พุตมีความแตกต่างกัน เช่น $(A.B)+C$ จะไม่เท่ากับ $A.(B.C)$

ตารางที่ 4.3 คุณสมบัติการเปลี่ยนกลุ่มภายใต้การ ออร์

| A | B | C | A+B | * (A+B)+C | B+C | * A+(B+C) |
|---|---|---|-----|--------------|-----|--------------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

4.4.3 กฎการแจกแจง (Distributive Laws)

1. $A.(B + C) = A.B + A.C$
2. $A + (B.C) = (A + B).(A+C)$

กฎการแจกแจง มีลักษณะคล้ายกับกฎทางคณิตศาสตร์ที่เป็นการกระจายการคูณเข้าไปในวงเล็บ สามารถพิสูจน์ได้ง่าย ด้วยตารางความจริงดังนี้

ตารางที่ 4.4 คุณสมบัติการแจกแจง

| A | B | C | B+C | * | AB | AC | * |
|---|---|---|-----|---------|----|----|-------------|
| | | | | A.(B+C) | | | (A.B)+(A.C) |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

4.4.4 กฎไอดีมโปเทนต์ (Idempotent Laws)

1. $A + A = A$
2. $A.A = A$

กฎไอดีมโปเทนต์คือการดำเนินการ กับตัวเองด้วยตัวดำเนินการ ออร์ หรือ ตัวดำเนินการ แอนด์ โดยสถานะเอาต์พุตที่ได้มีค่าเหมือนกับสถานะเดิมของอินพุตตัวดังกล่าวทุกประการ สามารถพิสูจน์ได้ดังนี้

ตัวดำเนินการ ออร์

$$\text{กรณีที่ } A = 0, \quad 0 + 0 = 0 = A$$

$$\text{กรณีที่ } A = 1, \quad 1 + 1 = 1 = A$$

ตัวดำเนินการ แอนด์

$$\text{กรณีที่ } A = 0, \quad 0.0 = 0 = A$$

$$\text{กรณีที่ } A = 1, \quad 1.1 = 1 = A$$

4.4.5 กฎนิเสธสองครั้ง (Double Negation Law)

$$\bar{\bar{A}} = A$$

เมื่อนำอินพุตมาใส่ในนิเสธ 2 ครั้ง สถานะของเอาต์พุตที่ได้จะเหมือนกับอินพุตเดิมที่ยังไม่ผ่านนิเสธ สามารถพิสูจน์ได้ดังนี้

พิสูจน์

$$\text{กรณีที่ } A = 0, \quad \bar{\bar{A}} = 1, \bar{\bar{A}} = 0 = A$$

$$\text{กรณีที่ } A = 1, \quad \bar{\bar{A}} = 0, \bar{\bar{A}} = 1 = A$$

4.4.6 กฎการแอนด์ (AND Laws)

1. $A \cdot 0 = 0$

2. $A \cdot 1 = A$

พิสูจน์ กฎข้อที่ 1 ($A \cdot 0 = 0$)

กรณีที่ $A = 0$, $0 \cdot 0 = 0$

กรณีที่ $A = 1$ $1 \cdot 0 = 0$

ดังนั้นจากกฎการแอนด์ข้อที่ 1 สรุปได้ว่า อินพุตใดๆ แอนด์กับ 0 จะได้เอาต์พุตเป็น 0 เสมอ

พิสูจน์ กฎข้อที่ 2 ($A \cdot 1 = A$)

กรณีที่ $A = 0$, $0 \cdot 1 = 0 = A$

กรณีที่ $A = 1$ $1 \cdot 1 = 1 = A$

ดังนั้นจากกฎการแอนด์ข้อที่ 2 สรุปได้ว่า อินพุตใดๆ แอนด์กับ 1 เอาต์พุตที่ได้จะมีสถานะเดียวกันกับอินพุตเสมอ

4.4.7 กฎการออร์ (OR Laws)

1. $A + 0 = A$

2. $A + 1 = 1$

พิสูจน์ กฎข้อที่ 1 ($A + 0 = A$)

กรณีที่ $A = 0$, $0 + 0 = 0 = A$

กรณีที่ $A = 1$ $1 + 0 = 1 = A$

ดังนั้นจากกฎการออร์ข้อที่ 1 สรุปได้ว่า อินพุตใดๆ ออรรกับ 0 จะได้เอาต์พุตที่มีสถานะเดียวกับอินพุตเสมอ

พิสูจน์ กฎข้อที่ 2 ($A + 1 = 1$)

กรณีที่ $A = 0$, $0 + 1 = 1$

กรณีที่ $A = 1$ $1 + 1 = 1$

ดังนั้นจากกฎการออร์ข้อที่ 2 สรุปได้ว่า อินพุตใดๆ ออรรกับ 1 เอาต์พุตที่ได้จะมีค่าเป็น 1 เสมอ

4.4.8 กฎคอมพลิเมนต์ (Complement Laws)

1. $A \cdot \bar{A} = 0$

2. $A + \bar{A} = 1$

กฎการคอมพลิเมนต์ คือ กรณีนำอินพุต และ นิเสธของอินพุตดังกล่าวมาผ่านตัวดำเนินการแอนด์ เอาต์พุตที่ได้มีค่าเป็น 0 เสมอ แต่หากนำอินพุต และ นิเสธของอินพุตดังกล่าวมาผ่านตัวดำเนินการ ออร์ เอาต์พุตที่ได้มีค่าเป็น 1 เสมอ ดังนี้

พิสูจน์ กฎข้อที่ 1 ($A \cdot \bar{A} = 0$)

กรณีที่ $A = 0$, $0 \cdot 0 = 0 = 0$

กรณีที่ $A = 1$ $1 \cdot 0 = 0 = 0$

ดังนั้นจากกฎคอมพลิเมนต์ข้อที่ 1 สรุปได้ว่า หากนำอินพุตใดๆ และ นิเสธของอินพุตดังกล่าวมาผ่านตัวดำเนินการ แอนด์ ได้เอาต์พุตที่มีค่าเป็น 0 เสมอ

พิสูจน์ กฎข้อที่ 2 ($A + \bar{A} = 1$)

$$\text{กรณีที่ } A = 0, \quad 0 + 0 = 0 + 1 = 1$$

$$\text{กรณีที่ } A = 1 \quad 1 + 1 = 1 + 0 = 1$$

ดังนั้นจากกฎคอมพิเมนต์ข้อที่ 2 สรุปได้ว่า หากนำอินพุตใดๆ และ นิเสธของอินพุตดังกล่าวมาผ่านตัวดำเนินการ ออร์ ได้เอาต์พุตที่มีค่าเป็น 1 เสมอ

4.4.9 กฎความซ้ำซ้อน (Redundance Laws)

$$1. A + A.B = A$$

$$2. A.(A+B) = A$$

$$3. A + \bar{A}.B = A + B$$

$$4. A(\bar{A} + B) = A.B$$

กฎความซ้ำซ้อน สามารถถูกพิสูจน์ได้ง่ายโดยใช้กฎพีชคณิตบูลีนที่ได้รับการพิสูจน์แล้วว่าจริง ดังนี้

พิสูจน์ กฎข้อที่ 1 ($A + A.B = A$)

$$A + A.B = A.1 + A.B \quad \text{กฎ 4.4.6 ข้อที่ 2}$$

$$= A.(1 + B) \quad \text{กฎ 4.4.3 ข้อที่ 1}$$

$$= A.1 \quad \text{กฎ 4.4.7 ข้อที่ 2}$$

$$= A \quad \text{กฎ 4.4.6 ข้อที่ 2}$$

พิสูจน์ กฎข้อที่ 2 ($A.(A+B) = A$)

$$A.(A+B) = A.A + A.B \quad \text{กฎ 4.4.3 ข้อที่ 1}$$

$$= A + A.B \quad \text{กฎ 4.4.4 ข้อที่ 2}$$

$$= A \quad \text{กฎ 4.4.9 ข้อที่ 1}$$

พิสูจน์ กฎข้อที่ 3 ($A + \bar{A}.B = A + B$)

$$A + \bar{A}.B = (A + \bar{A}).(A + B) \quad \text{กฎ 4.4.3 ข้อที่ 2}$$

$$= 1.(A + B) \quad \text{กฎ 4.4.8 ข้อที่ 2}$$

$$= A + B \quad \text{กฎ 4.4.6 ข้อที่ 2}$$

พิสูจน์ กฎข้อที่ 4 ($A(\bar{A} + B) = A.B$)

$$A(\bar{A} + B) = A.\bar{A} + A.B \quad \text{กฎ 4.4.3 ข้อที่ 1}$$

$$= 0 + A.B \quad \text{กฎ 4.4.8 ข้อที่ 1}$$

$$= A.B \quad \text{กฎ 4.4.7 ข้อที่ 1}$$

4.4.10 กฎเดอมอร์แกน (De Morgan Laws)

$$1. \bar{A+B} = \bar{A}.B$$

$$2. \overline{A.B} = \bar{A} + \bar{B}$$

กฎเดอมอร์แกนสามารถพิสูจน์ได้ โดยใช้ตารางความจริง ดังนี้

ตารางที่ 4.5 คุณสมบัติ $\overline{A+B} = \overline{A} \cdot \overline{B}$

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

ตารางที่ 4.6 คุณสมบัติ $\overline{AB} = \overline{A} + \overline{B}$

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

4.5 บทสรุป

การออกแบบวงจรถิทัศน์ นั้นจำเป็นต้องคำนึงถึงการใช้งานอุปกรณ์ให้น้อยที่สุดด้วย เพื่อให้ประหยัดค่าใช้จ่ายให้มากที่สุด ซึ่งการลดขนาดของวงจรรวมเชิงจัดหมู่สามารถทำได้โดยการใช้กฎพื้นฐานของบูลีน ดังนั้นผู้ออกแบบจึงต้องมีความรู้เกี่ยวกับกฎพื้นฐานต่างๆ ของบูลีนเป็นอย่างดี การวาดวงจรจากสมการบูลีนจะต้องให้ลำดับความสำคัญกับตัวดำเนินการนอตก่อนเสมอ แล้วจึงตามด้วยตัวดำเนินการแอนด์ และ ตัวดำเนินการออร์ ตามลำดับ การหาสถานะที่เป็นไปได้ทั้งหมดของวงจรมีสามารถทำได้โดยใช้ตารางความจริง นอกเหนือจากนั้นยังสามารถใช้ตารางความจริงสำหรับการพิสูจน์สมการพีชคณิตบูลีนได้ด้วยเช่นกัน

คำถามท้ายบท

1. จงสร้างวงจรจากสมการพีชคณิตบูลีนต่อไปนี้
 - 1.1 $Z = A.(B+C)$
 - 1.2 $Z = A.B.(C+D)$
 - 1.3 $Z = \overline{A}B+C$
 - 1.4 $Z = \overline{A}+BC$

2. จงคำนวณหาผลลัพธ์ที่เป็นไปได้ทั้งหมดจากสมการพีชคณิตบูลีนต่อไปนี้ โดยใช้ตารางความจริง
 - 2.1 $Z = A.(B+C)$
 - 2.2 $Z = A.B.(C+D)$
 - 2.3 $Z = \overline{A}.B+C$
 - 2.4 $Z = \overline{A}+B.C$

3. จงพิสูจน์สมการพีชคณิตบูลีนต่อไปนี้โดยใช้ตารางความจริง
 - 3.1 $A.\overline{A} = 0$
 - 3.2 $A + \overline{A} = 1$
 - 3.3 $A + A.B = A$
 - 3.4 $A + \overline{A}B = A+B$

4. จงพิสูจน์สมการพีชคณิตบูลีนต่อไปนี้โดยใช้กฎพื้นฐานบูลีน
 - 4.1 $A + A.\overline{A} = A$
 - 4.2 $A + \overline{A}B + AB = A + B$
 - 4.3 $\overline{A}B + A + B = 1$
 - 3.4 $A + \overline{A}B = A+B$

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

แผนบริหารการสอนประจำบทที่ 5

วงจรรวมเชิงจัดหมู่ 3 ชั่วโมง

หัวข้อเนื้อหา

- 5.1 การแก้ไขวงจรให้อยู่ในรูปที่ง่ายขึ้นโดยใช้กฎพีชคณิตบูลีน
- 5.2 การเขียนสมการพีชคณิตบูลีนจากผลรวมของผลคูณ
- 5.3 การเขียนสมการพีชคณิตบูลีนจากผลคูณของผลรวม
- 5.4 การเขียนสมการพีชคณิตบูลีนจากตารางความจริง
- 5.5 การออกแบบวงจรรวมเชิงจัดหมู่
- 5.6 การใช้ยูนิเวอร์แซลเกตในการออกแบบวงจรรวมเชิงจัดหมู่
- 5.7 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนสามารถลดรูปสมการพีชคณิตบูลีนโดยใช้กฎพื้นฐานพีชคณิตบูลีน
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการเขียนสมการพีชคณิตบูลีนจากผลรวมของผลคูณ
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการเขียนสมการพีชคณิตบูลีนจากผลคูณของผลรวม
4. เพื่อให้ผู้เรียนสามารถเขียนสมการบูลีนของค่าเอาต์พุตจากตารางความจริง
5. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการออกแบบวงจรรวมเชิงจัดหมู่
6. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับยูนิเวอร์แซลเกต และสามารถนำยูนิเวอร์แซลเกตมาใช้แทนเกตพื้นฐานชนิดอื่นได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

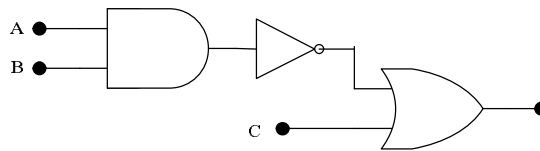
การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 5

วงจรรวมเชิงจัดหมู่

วงจรรวมเชิงจัดหมู่ (Combinatorial Logic Circuits) คือ วงจรที่สถานะของเอาต์พุตขึ้นอยู่กับสถานะของการจัดหมู่ของอินพุตทั้งหมดที่มีอยู่ในวงจร โดยผลการทำงานของเอาต์พุตจะขึ้นอยู่กับคุณสมบัติของเกตพื้นฐานต่าง ๆ และสัญญาณของอินพุตที่ป้อนเข้าสู่วงจร โดยที่วงจรดังกล่าวจะไม่สามารถเก็บสถานะไว้ใช้งานในช่วงเวลาต่อไปได้ ดังนั้นจึงสรุปได้ว่าวงจรรวมเชิงจัดหมู่ คือวงจรที่เกิดจากการต่อใช้งานร่วมกันของเกตพื้นฐานเท่านั้น ดังตัวอย่างต่อไปนี้



รูปที่ 5.1 ตัวอย่างวงจรรวมเชิงจัดหมู่

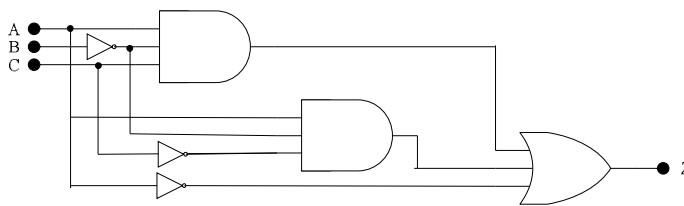
รูปที่ 5.1 แสดงตัวอย่างของวงจรรวมเชิงจัดหมู่ ซึ่งเกิดจากการต่อใช้งานร่วมกันเฉพาะเกตพื้นฐานเท่านั้น โดยจากรูป 5.1 เกตพื้นฐานประกอบด้วย แอนด์เกต 1 ตัว นีตเกต 1 ตัว และ ออร์เกต 1 ตัว

5.1 การแก้ไขวงจรให้อยู่ในรูปที่ง่ายขึ้นโดยใช้กฎพีชคณิตบูลีน

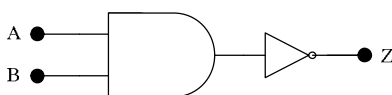
สำหรับวงจรลอจิกบางวงจร พบว่าสมการพีชคณิตบูลีนยังมีความกำกวมอยู่ ซึ่งสามารถแก้ความกำกวมได้โดยใช้กฎพีชคณิตบูลีน ข้อดีคือจะทำให้สมการพีชคณิตบูลีนดูง่ายขึ้น ทำให้วงจรมีขนาดเล็กลง ส่งผลให้ใช้งานจำนวนเกตลดน้อยลง และประหยัดค่าใช้จ่าย ในขณะที่สถานะเอาต์พุตยังคงเดิม

ตัวอย่างที่ 5-1 จงลดรูปสมการพีชคณิตบูลีนต่อไปนี้ $Z = ABC + ABC + \bar{A}$ โดยใช้กฎพีชคณิตบูลีนให้อยู่ในรูปอย่างง่ายพร้อมวาดวงจรลอจิก

| | | |
|--------|--|--|
| วิธีทำ | $Z = ABC + ABC + \bar{A}$ $= AB(C + C) + \bar{A}$ $= AB1 + \bar{A}$ $= AB + \bar{A}$ $= \bar{A} + AB$ $= \bar{A} + B$ $= \bar{A}B$ | <p>กฎ 4.4.3 ข้อที่ 1</p> <p>กฎ 4.4.8 ข้อที่ 2</p> <p>กฎ 4.4.6 ข้อที่ 2</p> <p>กฎ 4.4.1 ข้อที่ 1</p> <p>กฎ 4.4.9 ข้อที่ 3</p> <p>กฎ 4.4.10 ข้อที่ 2</p> |
|--------|--|--|



รูปที่ 5.2 วงจร $Z = ABC + ABC + A$



รูปที่ 5.3 วงจร $Z = \overline{AB}$

รูปที่ 5.2 แสดงวงจรลอจิกของสมการ $Z = ABC + ABC + A$ และรูปที่ 5.3 แสดงวงจรลอจิกของสมการ $Z = \overline{AB}$ ซึ่งเป็นวงจรลอจิกที่เกิดจากการนำสมการพีชคณิตบูลีนของวงจรในรูปที่ 5.2 มาจัดรูปใหม่โดยใช้กฎพีชคณิตบูลีน ซึ่งวงจรมีขนาดเล็กกลง แต่สถานะของเอาต์พุตยังคงเดิม

ตัวอย่างที่ 5-2 จงลดรูปสมการพีชคณิตบูลีนต่อไปนี้ $Z = ABC + AB(\overline{BC})$ โดยใช้กฎบูลีนให้อยู่ในรูปอย่างง่าย

วิธีทำ

$$Z = ABC + AB(\overline{BC})$$

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

กฏ 4.4.10 ข้อที่ 2

$$= ABC + AB(B + C)$$

กฏ 4.4.5

$$= ABC + ABB + ABC$$

กฏ 4.4.3 ข้อที่ 1

$$= ABC + AB + ABC$$

กฏ 4.4.4 ข้อที่ 2

$$= ABC + AB1 + ABC$$

กฏ 4.4.6 ข้อที่ 2

$$= ABC + AB(1 + C)$$

กฏ 4.4.3 ข้อที่ 1

$$= ABC + AB1$$

กฏ 4.4.7 ข้อที่ 2

$$= ABC + AB$$

กฏ 4.4.6 ข้อที่ 2

$$= (AC + A)B$$

กฏ 4.4.3 ข้อที่ 1

$$= (C + A)B$$

กฏ 4.4.9 ข้อที่ 3

$$= BC + AB$$

กฏ 4.4.3 ข้อที่ 1

$$= \overline{AB} + BC$$

กฏ 4.4.1 ข้อที่ 1

ตัวอย่างที่ 5-3 จงลดรูปสมการพีชคณิตบูลีนต่อไปนี้ $Z = \overline{AB} + A$ โดยใช้กฎบูลีนให้อยู่ในรูปอย่างง่าย

วิธีทำ

$$Z = \overline{AB} + A$$

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

กฏ 4.4.10 ข้อที่ 2

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

กฏ 4.4.1 ข้อที่ 1

$$= 1 + B$$

กฏ 4.4.8 ข้อที่ 2

$$=1$$

กฎ 4.4.7 ข้อที่ 2

$$=0$$

5.2 การเขียนสมการพีชคณิตบูลีนจากผลรวมของผลคูณ

ผลรวมของผลคูณ (Sum of Product) สามารถนำมาเขียนให้อยู่ในรูปสมการพีชคณิตบูลีนได้ โดยมีหลักการดังต่อไปนี้

1. นำตัวเลขฐานสิบแต่ละพจน์ย่อยมาแปลงเป็นเลขฐานสอง และใส่จำนวนบิตให้เท่ากับจำนวนอินพุต เช่นกรณี 3 อินพุตของ 2_{10} เมื่อแปลงเป็นฐานสองจะได้ 10_2 แต่เนื่องจากเป็น 3 อินพุต จึงต้องเพิ่ม 0 ไว้ตำแหน่งหน้าสุดอีก 1 ตัวได้เป็น 010_2

2. แทนค่าเลขฐานสองที่ได้จากข้อ 1 ด้วยอินพุตแต่ละตัวและนำมาเชื่อมกันโดยใช้ตัวดำเนินการแอนด์ โดยกรณีที่เป็นการจับบิตที่มีค่าเป็น 0 ให้ใส่คอมพลิเมนต์ด้วย เช่น กรณีอินพุต (A, B, C) มีค่าเป็น 2 เมื่อแปลงจาก 010 จะได้ $\bar{A}BC$

3. นำพจน์ย่อยแต่ละพจน์มาเชื่อมกันด้วยตัวดำเนินการออร์

ตัวอย่างที่ 5-4 จงแปลง $f(A,B,C) = \sum m(0,1,3,5)$ ให้อยู่ในรูปสมการพีชคณิตบูลีน

วิธีทำ เนื่องจากมีอินพุตทั้งหมด 3 ตัว จึงสามารถแปลงพจน์ย่อยแต่ละพจน์ได้ดังนี้

| | | | | | | |
|----------|----|---------|----|-------|----|-------------------------|
| 0_{10} | -> | 000_2 | -> | 000 | -> | $\bar{A}\bar{B}\bar{C}$ |
| 1_{10} | -> | 001_2 | -> | 001 | -> | $\bar{A}\bar{B}C$ |
| 3_{10} | -> | 011_2 | -> | 011 | -> | $\bar{A}BC$ |
| 5_{10} | -> | 101_2 | -> | 101 | -> | $A\bar{B}C$ |

เพราะฉะนั้น $f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$

ตัวอย่างที่ 5-5 จงแปลง $f(A,B,C,D) = \sum m(2,3,5,9,11)$ ให้อยู่ในรูปสมการพีชคณิตบูลีน

วิธีทำ เนื่องจากมีอินพุตทั้งหมด 4 ตัว จึงสามารถแปลงพจน์ย่อยแต่ละพจน์ได้ดังนี้

| | | | | | | |
|-----------|----|----------|----|--------|----|--------------------------|
| 2_{10} | -> | 0010_2 | -> | 0010 | -> | $\bar{A}\bar{B}C\bar{D}$ |
| 3_{10} | -> | 0011_2 | -> | 0011 | -> | $\bar{A}\bar{B}CD$ |
| 5_{10} | -> | 0101_2 | -> | 0101 | -> | $\bar{A}B\bar{C}D$ |
| 9_{10} | -> | 1001_2 | -> | 1001 | -> | $A\bar{B}\bar{C}D$ |
| 11_{10} | -> | 1011_2 | -> | 1011 | -> | $A\bar{B}CD$ |

เพราะฉะนั้น $f(A,B,C,D) = \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + A\bar{B}\bar{C}D + A\bar{B}CD$

สำหรับพจน์ย่อยแต่ละพจน์ของสมการพีชคณิตบูลีนที่ซึ่งแอนด์กันอยู่ เรียกว่า มินเทอม (minterm)

5.3 การเขียนสมการพีชคณิตบูลีนจากผลคูณของผลรวม

นอกจากผลรวมของผลคูณแล้ว ผลคูณของผลรวม (Product of Sum) สามารถนำมาเขียนให้อยู่ในรูปสมการพีชคณิตบูลีนได้เช่นเดียวกัน โดยมีหลักการดังต่อไปนี้

- นำตัวเลขฐานสิบแต่ละพจน์ย่อยมาแปลงเป็นเลขฐานสอง และใส่จำนวนบิตให้เท่ากับจำนวนอินพุต เช่นกรณี 3 อินพุตของ 2_{10} เมื่อแปลงเป็นฐานสองจะได้ 10_2 แต่เนื่องจากเป็น 3 อินพุต จึงต้องเพิ่ม 0 ไว้ตำแหน่งหน้าสุดอีก 1 ตัวได้เป็น 010_2
- แทนค่าเลขฐานสองที่ได้จากข้อ 1 ด้วยอินพุตแต่ละตัวและนำมาเชื่อมกันโดยใช้ตัวดำเนินการออร์โดยกรณีที่มีค่าประจำบิตเป็น 1 ให้ใส่คอมพลิเมนต์ด้วย เช่น กรณีอินพุต (A, B, C) มีค่าเป็น 2 เมื่อแปลงจาก $0+1+0$ จะได้ $A+B+C$
- นำพจน์ย่อยแต่ละพจน์มาเชื่อมกันด้วยตัวดำเนินการแอนด์

ตัวอย่างที่ 5-6 จงแปลง $f(A,B) = \prod M(0,1)$ ให้อยู่ในรูปสมการพีชคณิตบูลีน

วิธีทำ เนื่องจากมีอินพุตทั้งหมด 2 ตัว จึงสามารถแปลงพจน์ย่อยแต่ละพจน์ได้ดังนี้

$$\begin{array}{l} 0_{10} \rightarrow 00_2 \rightarrow 0 + 0 \rightarrow (A+B) \\ 1_{10} \rightarrow 01_2 \rightarrow 0 + 1 \rightarrow (A+B) \end{array}$$

เพราะฉะนั้น $f(A,B) = (A+B)(A+B)$

ตัวอย่างที่ 5-7 จงแปลง $f(A,B,C) = \prod M(0,2,5,7)$ ให้อยู่ในรูปสมการพีชคณิตบูลีน

วิธีทำ เนื่องจากมีอินพุตทั้งหมด 3 ตัว จึงสามารถแปลงพจน์ย่อยแต่ละพจน์ได้ดังนี้

$$\begin{array}{l} 0_{10} \rightarrow 000_2 \rightarrow 0 + 0 + 0 \rightarrow (A+B+C) \\ 2_{10} \rightarrow 010_2 \rightarrow 0 + 1 + 0 \rightarrow (A+B+C) \\ 5_{10} \rightarrow 101_2 \rightarrow 1 + 0 + 1 \rightarrow (A+B+C) \\ 7_{10} \rightarrow 111_2 \rightarrow 1 + 1 + 1 \rightarrow (A+B+C) \end{array}$$

เพราะฉะนั้น $f(A,B,C) = (A+B+C)(A+B+C)(A+B+C)(A+B+C)$

สำหรับพจน์ย่อยแต่ละพจน์ของสมการพีชคณิตบูลีนที่ซึ่งออร์กันอยู่ เรียกว่า แมกเทอม (maxterm)

5.4 การเขียนสมการพีชคณิตบูลีนจากตารางความจริง

การนำสถานะของเอาต์พุตจากตารางความจริงมาเขียนเป็นสมการพีชคณิตบูลีน สามารถพิจารณาได้ 2 กรณี ดังนี้

กรณีที่ 1 เลือกสถานะของเอาต์พุตเฉพาะกรณีที่มีค่าเป็น 1 ทั้งหมด และเขียนสมการพีชคณิตบูลีนในรูปของผลรวมของผลคูณ (สถานะอินพุตที่มีค่าเป็น 0 ให้ใส่คอมพลิเมนต์)

กรณีที่ 2 เลือกสถานะของเอาต์พุตเฉพาะที่มีค่าเป็น 0 ทั้งหมด และเขียนสมการพีชคณิตบูลีนในรูปของผลคูณของผลรวม (สถานะอินพุตที่มีค่าเป็น 1 ให้ใส่คอมพลิเมนต์)

ตัวอย่างที่ 5-8 จงเขียนสมการพีชคณิตบูลีนในรูปผลรวมของผลคูณ และ ผลคูณของผลรวมจากตารางความจริงต่อไปนี้

| อินพุต | | | เอาต์พุต |
|--------|---|---|----------|
| A | B | C | Z |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

กรณีที่ 1 เลือกสถานะของเอาต์พุตที่มีค่าเป็น 1

วิธีทำ เนื่องจากสถานะของอินพุตที่ทำให้สถานะของเอาต์พุตมีค่าเป็น 1 มีดังนี้

1. $A = 0, B = 1$ และ $C = 0$
2. $A = 0, B = 1$ และ $C = 1$
3. $A = 1, B = 0$ และ $C = 1$

ดังนั้น ดั่งเขียนเป็นสมการพีชคณิตบูลีนได้ดังนี้ $Z = ABC + ABC + ABC$

กรณีที่ 2 เลือกสถานะของเอาต์พุตที่มีค่าเป็น 0

วิธีทำ เนื่องจากสถานะของอินพุตที่ทำให้สถานะของเอาต์พุตมีค่าเป็น 0 มีดังนี้

1. $A = 0, B = 0$ และ $C = 0$
2. $A = 0, B = 0$ และ $C = 1$
3. $A = 1, B = 0$ และ $C = 0$
4. $A = 1, B = 1$ และ $C = 0$
5. $A = 1, B = 1$ และ $C = 1$

ดังนั้น ดั่งเขียนเป็นสมการพีชคณิตบูลีนได้ดังนี้

$$Z = (A+B+C)(A+B+C)(A+B+C)(A+B+C)(A+B+C)$$

5.5 การออกแบบวงจรรวมเชิงจัดหมู่

การออกแบบวงจรรวมเชิงจัดหมู่ คือการที่ผู้ออกแบบวงจร ทำการออกแบบวงจรตามคุณลักษณะของวงจรที่ต้องการโดยใช้ตารางความจริงสำหรับออกแบบสถานะของเอาต์พุต เพื่อที่จะแปลงมาเป็นสมการพีชคณิตบูลีนต่อไป โดยมีขั้นตอนดังนี้

1. จากคุณลักษณะของวงจรที่ต้องการ นำมาออกแบบโดยใช้ตารางความจริง
2. จากตารางความจริงที่ได้นำมาแปลงเป็นสมการพีชคณิตบูลีน
3. ทำการลดรูปสมการพีชคณิตบูลีนให้อยู่ในรูปอย่างง่าย
4. สร้างวงจรตามสมการพีชคณิตบูลีนที่ได้ในขั้นตอนที่ 3

ตัวอย่างที่ 5-9 จงออกแบบวงจรลอจิกที่สถานะของเอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่ออินพุตอย่างน้อย 2 ตัวจากทั้งหมด 3 ตัวมีสถานะเป็น 1

วิธีทำ จากโจทย์กำหนดว่าอินพุตอย่างน้อย 2 ตัวจากทั้งหมด 3 ตัว แสดงว่ามีจำนวนอินพุตทั้งหมด 3 ตัว โดยที่สถานะของเอาต์พุตจะมีค่าเป็น 1 ได้นั้นจะต้องมีอินพุตอย่างน้อย 2 ตัวมีค่าเป็น 1 พร้อมกันเท่านั้น ส่วนกรณีอื่นๆ สถานะของเอาต์พุตจะมีค่าเป็น 0 ทั้งหมด สามารถสร้างเป็นตารางความจริงได้ดังนี้

| แถวที่ | อินพุต | | | เอาต์พุต Z |
|--------|--------|---|---|---------------|
| | A | B | C | |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 1 | 0 | 1 | 1 |
| 7 | 1 | 1 | 0 | 1 |
| 8 | 1 | 1 | 1 | 1 |

จากตาราง เห็นได้ว่า มีเพียงแถวสีเทาเท่านั้นที่ทำให้สถานะเอาต์พุตมีค่าเป็น 1 เนื่องจากเป็นแถวที่มีอินพุตอย่างน้อย 2 ตัวที่มีค่าเป็น 1 พร้อมกัน โดยมีทั้งหมด 4 แถวดังนี้

แถวที่ 4 $A = 0, B = 1$ และ $C = 1$

แถวที่ 6 $A = 1, B = 0$ และ $C = 1$

แถวที่ 7 $A = 1, B = 1$ และ $C = 0$

แถวที่ 8 $A = 1, B = 1$ และ $C = 1$

ดังนั้น ดังเขียนเป็นสมการพีชคณิตบูลีนได้ดังนี้ $Z = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

จากสมการพีชคณิตบูลีนที่ได้ สามารถใช้กฎพีชคณิตบูลีนในการลดรูปได้ดังนี้

$$Z = \overline{A}BC + \overline{A}BC + AB\overline{C} + ABC$$

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

$$= \overline{A}BC + \overline{A}BC + AB(1)$$

$$= \overline{A}BC + \overline{A}BC + AB$$

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

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

$$= \overline{A}BC + AC + AB$$

กฎ 4.4.3 ข้อที่ 1

กฎ 4.4.8 ข้อที่ 2

กฎ 4.4.6 ข้อที่ 2

กฎ 4.4.3 ข้อที่ 1

กฎ 4.4.9 ข้อที่ 3

กฎ 4.4.3 ข้อที่ 1

$$\begin{aligned}
 &= ABC + AB + AC \\
 &= (AC + A)B + AC \\
 &= (C + A)B + AC \\
 &= BC + AB + AC
 \end{aligned}$$

กฎ 4.4.1 ข้อที่ 1

กฎ 4.4.3 ข้อที่ 1

กฎ 4.4.9 ข้อที่ 3

กฎ 4.4.3 ข้อที่ 1

เพราะฉะนั้น ได้สมการพีชคณิตบูลีนที่อยู่ในรูปที่ง่ายที่สุดเป็นดังนี้ $Z = AB + AC + BC$

ตัวอย่างที่ 5-10 จงออกแบบวงจรลอจิกที่สถานะของเอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่ออินพุต 2 ตัวจากทั้งหมด 4 ตัวมีสถานะเป็น 1

วิธีทำ จากโจทย์กำหนดว่าอินพุต 2 ตัวจากทั้งหมด 4 ตัว แสดงว่ามีจำนวนอินพุตทั้งหมด 4 ตัว โดยที่สถานะของเอาต์พุตจะมีค่าเป็น 1 ได้นั้นจะต้องมีอินพุต 2 ตัวมีค่าเป็น 1 พร้อมกันเท่านั้น ส่วนกรณีอื่น ๆ สถานะของเอาต์พุตจะมีค่าเป็น 0 ทั้งหมด สามารถสร้างเป็นตารางความจริงได้ดังนี้

| แถวที่ | อินพุต | | | | เอาต์พุต Z |
|--------|--------|---|---|---|---------------|
| | A | B | C | D | |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 1 | 1 |
| 7 | 0 | 1 | 1 | 0 | 1 |
| 8 | 0 | 1 | 1 | 1 | 0 |
| 9 | 1 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 1 | 1 |
| 11 | 1 | 0 | 1 | 0 | 1 |
| 12 | 1 | 0 | 1 | 1 | 0 |
| 13 | 1 | 1 | 0 | 0 | 1 |
| 14 | 1 | 1 | 0 | 1 | 0 |
| 15 | 1 | 1 | 1 | 0 | 0 |
| 16 | 1 | 1 | 1 | 1 | 0 |

จากตาราง เห็นได้ว่า มีเพียงแถวสีเทาเท่านั้นที่ทำให้สถานะเอาต์พุตมีค่าเป็น 1 เนื่องจากเป็นแถวที่มีอินพุต 2 ตัวเท่านั้นที่มีค่าเป็น 1 พร้อมกัน โดยมีทั้งหมด 6 แถวดังนี้

แถวที่ 4 $A = 0, B = 0, C = 1$ และ $D = 1$

แถวที่ 6 $A = 0, B = 1, C = 0$ และ $D = 1$

แถวที่ 7 $A = 0, B = 1, C = 1$ และ $D = 0$

แถวที่ 10 $A = 1, B = 0, C = 0$ และ $D = 1$

แถวที่ 11 $A = 1, B = 0, C = 1$ และ $D = 0$

แถวที่ 13 $A = 1, B = 1, C = 0$ และ $D = 0$

ดังนั้น ดังเขียนเป็นสมการพีชคณิตบูลีนได้ดังนี้

$$Z = \overline{A}BCD + A\overline{B}CD + ABC\overline{D} + ABCD + ABCD + ABCD$$

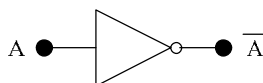
จากสมการพีชคณิตบูลีนที่ได้ พบว่าไม่สามารถลดรูปได้แล้ว เพราะฉะนั้นสรุปได้ว่าสมการดังกล่าวเป็นสมการที่อยู่ในรูปที่ง่ายที่สุดแล้ว

5.6 การใช้ยูนิเวอร์แซลเกตในการออกแบบวงจรรวมเชิงจัดหมู่

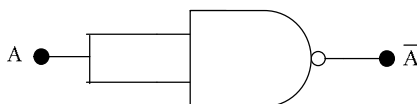
ยูนิเวอร์แซลเกต (Universal Gate) คือ เกตพื้นฐานที่สามารถนำไปใช้งานแทนเกตพื้นฐานชนิดอื่นๆ ได้ทั้งหมด ซึ่งเป็นกลุ่มของเกตพื้นฐานที่ช่วยอำนวยความสะดวกให้ผู้ออกแบบได้เป็นอย่างดี เนื่องจากไม่จำเป็นต้องมองหาเกตพื้นฐานชนิดอื่นอีกเลย เพราะสามารถใช้ยูนิเวอร์แซลเกตแทนได้ทั้งหมด โดยเกตพื้นฐานที่เป็นยูนิเวอร์แซลเกตมี 2 ชนิดคือ แนนด์เกต และ นอร์เกต

ตัวอย่างที่ 5-11 จงใช้แนนด์เกตมาสร้างวงจรที่มีการทำงานเป็น นี้อตเกต แอนด์เกต และ ออร์เกต

วิธีทำ 1. นี้อตเกต

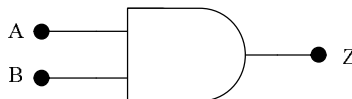


จากรูป อินพุต คือ A และ เอาต์พุตคือ \overline{A} ดังนั้น หากใช้แนนด์เกตในการสร้างวงจรนี้อตเกตดังกล่าว จะได้ดังต่อไปนี้

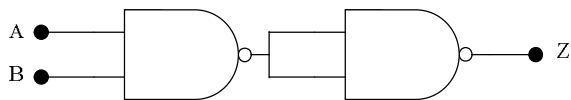


จากรูปอินพุต A เชื่อมต่อกับ ขาอินพุตทั้งสองขาของแนนด์เกต ดังนั้นจากกฎ 4.4.4 ข้อ 2 ได้ $AA = A$

2. แอนด์เกต

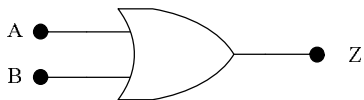


จากรูป อินพุต คือ A และ B เอาต์พุตคือ Z ดังนั้น หากใช้แนนด์เกตในการสร้างวงจรแอนด์เกตดังกล่าวจะได้ดังต่อไปนี้

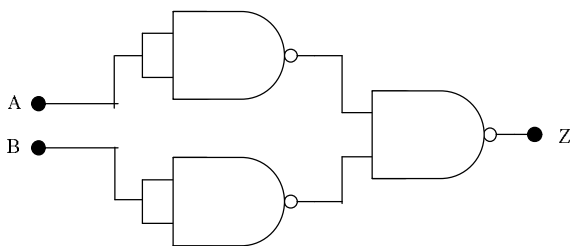


จากรูป อินพุต **A** และ **B** เชื่อมต่อกับ ขาอินพุตทั้งสองขาของแอนด์เกตได้เป็น AB และนำผลลัพธ์ดังกล่าวมาผ่านแอนด์เกตตัวที่สองซึ่งทำหน้าที่เป็นน็อตเกต ดังนั้นจากกฎ 4.4.5 ได้ $\overline{AB}=AB$

3. ออร์เกต

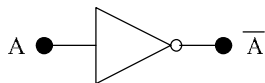


จากรูป อินพุต คือ **A** และ **B** เอาต์พุตคือ **Z** ดังนั้น หากใช้แอนด์เกตในการสร้างวงจรออร์เกต ดังกล่าวจะได้ดังต่อไปนี้

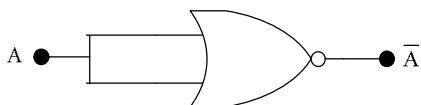


จากรูป อินพุต **A** และ **B** แต่ละตัวถูกเชื่อมต่อกับแอนด์เกตที่ทำหน้าที่เป็นน็อตเกตได้เป็น \overline{A} และ \overline{B} และนำผลลัพธ์ทั้งสองมาใช้งานเป็นอินพุตให้แอนด์เกตตัวสุดท้ายได้เป็น \overline{AB} และจากกฎเดอมอร์แกน (กฎ 4.4.10 ข้อ 2) ได้ว่า $\overline{AB}=\overline{A}+\overline{B}=A+B$

ตัวอย่างที่ 5-12 จงใช้อินเวอร์ตมาสร้างวงจรที่มึการทำงานเป็น น็อตเกต แอนด์เกต และ ออร์เกต
วิธีทำ **1. น็อตเกต**

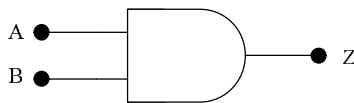


จากรูป อินพุต คือ **A** และ เอาต์พุตคือ \overline{A} ดังนั้น หากใช้อินเวอร์ตในการสร้างวงจรรน็อตเกต ดังกล่าวจะได้ดังต่อไปนี้

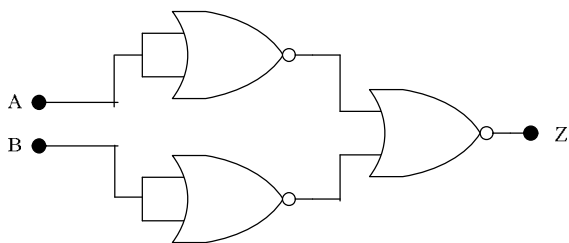


จากรูป อินพุต **A** เชื่อมต่อกับ ขาอินพุตทั้งสองขาของนอร์เกต ดังนั้น $\overline{A+A}=A$

2. แอนด์เกต

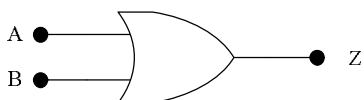


จากรูป อินพุต คือ **A** และ **B** เอาต์พุตคือ **Z** ดังนั้น หากใช้นอร์เกตในการสร้างวงจรแอนด์เกต ดังกล่าวจะได้ดังต่อไปนี้

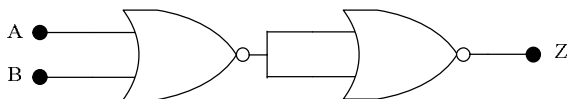


จากรูป อินพุต **A** และ **B** แต่ละตัวถูกเชื่อมต่อกับนอร์เกตที่ทำหน้าที่เป็นน็อตเกตได้เป็น \bar{A} และ \bar{B} และนำผลลัพธ์ทั้งสองมาใช้งานเป็นอินพุตให้นอร์เกตตัวสุดท้ายได้เป็น $\overline{\bar{A}\bar{B}}$ และ จากกฎเดอมอร์แกน (กฎ 4.4.10 ข้อ 1) ได้ว่า $\overline{\bar{A}\bar{B}} = AB$

3. ออร์เกต



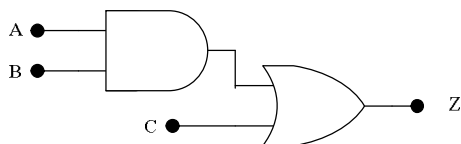
จากรูป อินพุต คือ **A** และ **B** เอาต์พุตคือ **Z** ดังนั้น หากใช้นอร์เกตในการสร้างวงจรออร์เกต ดังกล่าวจะได้ดังต่อไปนี้



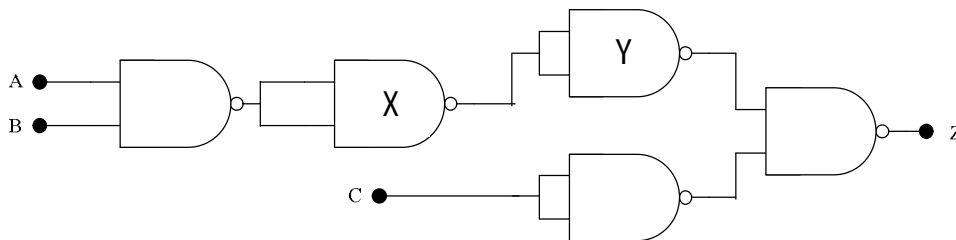
จากรูป อินพุต **A** และ **B** เชื่อมต่อกับ ขาอินพุตทั้งสองขาของนอร์เกตได้เป็น $\bar{A}\bar{B}$ และนำผลลัพธ์ดังกล่าวผ่านนอร์เกตตัวที่สองซึ่งทำหน้าที่เป็นน็อตเกต ดังนั้นจากกฎ 4.4.5 ได้ $\overline{\bar{A}\bar{B}} = A+B$

ตัวอย่างที่ 5-11 และ 5-12 แสดงให้เห็นว่าสามารถใช้แอนด์เกต และ นอร์เกตไปประยุกต์ใช้แทนเกตพื้นฐานอื่นๆ ได้ทั้งหมด

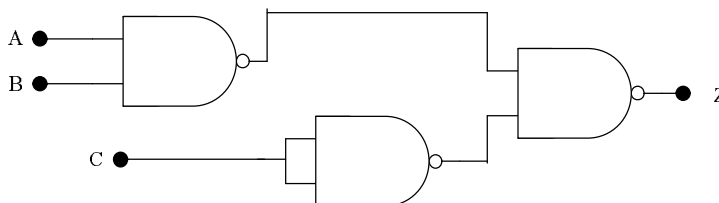
ตัวอย่างที่ 5-13 จากวงจรถลอจิกต่อไปนี้ จงออกแบบวงจรถดังกล่าวใหม่โดยใช้แนนด์เกตเท่านั้น



วิธีทำ จากรูป เป็นวงจรถลอจิกที่ใช้ แอนด์เกต 1 ตัว และ ออร์เกต 1 ตัว ดังนั้นจากตัวอย่างที่ 5-11 ที่กล่าวถึงการใช้อนินเวอร์ชันเกตแทนเกตพื้นฐานชนิดอื่นๆ ได้วงจรถลอจิกใหม่เป็นดังนี้



เนื่องจากแนนด์เกต X และ แนนด์เกต Y ทำหน้าที่เป็นคอมพลิเมนต์ที่อยู่ติดกัน ซึ่งจากกฎพื้นฐานพีชคณิตบูลีนพบว่าเมื่อนำอินพุตมาผ่าน คอมพลิเมนต์ สองครั้ง ผลลัพธ์ที่ได้จะเป็นอินพุตตัวเดิม (กฎ 4.4.5) ดังนั้นสามารถตัด แนนด์เกต X และ แนนด์เกต Y ได้ดังนี้



จากวงจรถพิสูจน์ได้ดังนี้

$$\begin{aligned} Z &= \overline{ABC} \\ &= \overline{AB} + \overline{C} \\ &= AB + C \end{aligned}$$

5.7 บทสรุป

กฎพีชคณิตบูลีนสามารถนำมาใช้สำหรับการลดรูปของวงจรถเพื่อให้วงจรมีขนาดเล็กลง และทำให้สมการของวงจรถอยู่ในรูปที่ง่ายขึ้น โดยสมการพีชคณิตบูลีนสามารถแปลงได้จากสมการที่อยู่ในรูปของผลรวมของผลคูณ (Sum of Product) สมการที่อยู่ในรูปของผลคูณของผลรวม (Product of Sum) หรือจากตารางความจริง สำหรับการออกแบบวงจรถรวมเชิงจัดหมุ่นั้นสามารถใช้เพียงแนนด์เกตหรือนอร์เกตในการออกแบบแทนเกตพื้นฐานชนิดอื่นได้ทั้งหมด จึงเรียกแนนด์เกต และ นอร์เกตว่ายูนิเวอร์แซลเกต

คำถามท้ายบท

1. จงลดรูปวงจรถัดไปนี้โดยใช้กฎบูลีนเพื่อให้อยู่ในรูปแบบที่ง่ายที่สุด
 - 1.1 $\overline{AB} + \overline{AB} + B$
 - 1.2 $(\overline{AB} + \overline{A})BC$
 - 1.3 $\overline{B}(\overline{A} + B)(\overline{A} + \overline{C})$
 - 1.4 $\overline{(A + B)(\overline{B} + C)(A + C)}$
 - 1.5 $ABC + \overline{AB}(\overline{AC})$

2. จงแปลงจากฟังก์ชันต่อไปนี้ให้อยู่ในรูปของสมการพีชคณิตบูลีน และลดรูปให้อยู่ในรูปที่ง่ายที่สุด
 - 2.1 $f(A, B, C) = \sum m(0, 1, 4, 6)$
 - 2.2 $f(A, B, C) = \sum m(1, 3, 4, 5, 7)$
 - 2.3 $f(A, B, C, D) = \sum m(0, 1, 4, 5, 6, 9, 10, 13)$
 - 2.4 $f(A, B, C) = \prod M(1, 2, 4, 7)$
 - 2.5 $f(A, B, C) = \prod M(0, 3, 5, 7)$

3. จงออกแบบวงจรถลอจิกที่สถานะของเอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่ออินพุตอย่างน้อย 3 ตัวจากทั้งหมด 4 ตัวมีสถานะเป็น 0

4. จากสมการพีชคณิตบูลีนที่กำหนดให้ต่อไปนี้ จงออกแบบวงจรถโดยใช้แอนด์เกตเพียงอย่างเดียว และใช้แอนด์เกตเพียงอย่างเดียว
 - 4.1 $\overline{AB} + C$
 - 4.2 $(A + B)\overline{C}$
 - 4.3 $AB(C + \overline{D})$
 - 4.4 $\overline{AB} + \overline{CD}$
 - 4.5 $\overline{AB} + \overline{CD}$

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.
- สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุตรธานี: มหาวิทยาลัยราชภัฏอุตรธานี.**

แผนบริหารการสอนประจำบทที่ 6 แผนผังคาร์โนท์ 3 ชั่วโมง

หัวข้อเนื้อหา

- 6.1 รูปแบบของแผนผังคาร์โนท์
 - 6.1.1 แผนผังคาร์โนท์ขนาด 2 ตัวแปร
 - 6.1.2 แผนผังคาร์โนท์ขนาด 3 ตัวแปร
 - 6.1.3 แผนผังคาร์โนท์ขนาด 4 ตัวแปร
- 6.2 การอ่านค่าลงแผนผังคาร์โนท์
- 6.3 การสร้างรูป
- 6.4 เรื่องไขที่ไม่สนใจ
- 6.5 การออกแบบวงจรรวม และลบเลขฐานสอง
 - 6.5.1 วงจรรวมแบบไม่คิดตัวทด
 - 6.5.2 วงจรรวมแบบคิดตัวทด
 - 6.5.3 วงจรลบแบบไม่คิดตัวยืม
 - 6.5.4 วงจรลบแบบคิดตัวยืม
 - 6.5.5 วงจรรวมแบบหลายบิต
 - 6.5.6 วงจรลบแบบหลายบิต
- 6.6 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับแผนผังคาร์โนท์
2. เพื่อให้ผู้เรียนสามารถใช้แผนผังคาร์โนท์เพื่อลดรูปสมการพีชคณิตบูลีนได้
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับสถานะที่ไม่สนใจ
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับวงจรรวมและวงจรถลบเลขฐานสอง

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 6

แผนผังคาร์โนท์

แผนผังคาร์โนท์ (Karnaugh Map) เป็นเทคนิควิธีที่ใช้สำหรับลดรูปสมการพีชคณิตบูลีนอีกวิธีหนึ่ง โดยใช้ตารางเป็นตัวช่วยในการลดรูปเพื่อให้อยู่ในรูปที่ง่ายขึ้น การใช้แผนผังคาร์โนท์จะแก้ปัญหาได้ง่าย และรวดเร็วกว่าการใช้กฎพื้นฐานพีชคณิตบูลีน ในกรณีที่สมการพีชคณิตบูลีนมีอินพุตขนาด 2 - 4 ตัวแปร

6.1 รูปแบบแผนผังคาร์โนท์

แผนผังคาร์โนท์ เป็นเทคนิควิธีที่เหมาะสมกับอินพุตขนาด 2 - 4 ตัวแปร ซึ่งหากสมการพีชคณิตบูลีนมีจำนวนอินพุตที่มากกว่า 4 ตัวแปรขึ้นไป วิธีอื่นจะมีความเหมาะสมกว่า ดังนั้นในบทนี้จะกล่าวถึงแผนผังคาร์โนท์ที่มีขนาด 2 - 4 ตัวแปรเท่านั้น ซึ่งจำนวนช่องตารางของแผนผังคาร์โนท์คือ $2^{\text{จำนวนอินพุต}}$ ดังนี้

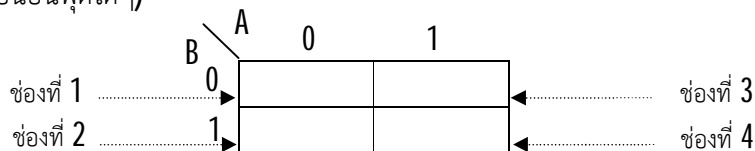
อินพุต 2 ตัวแปร มีจำนวนช่องทั้งหมด $2^2 = 4$ ช่อง

อินพุต 3 ตัวแปร มีจำนวนช่องทั้งหมด $2^3 = 8$ ช่อง

อินพุต 4 ตัวแปร มีจำนวนช่องทั้งหมด $2^4 = 16$ ช่อง

6.1.1 แผนผังคาร์โนท์ขนาด 2 ตัวแปร

แผนผังคาร์โนท์ขนาด 2 ตัวแปร ภายในตารางจะมีจำนวนช่องทั้งหมด 4 ช่องดังต่อไปนี้ (สมมติ A และ B เป็นอินพุตใดๆ)



รูปที่ 6.1 แผนผังคาร์โนท์แบบ 2 อินพุต

จากรูป 6.1 แสดงแผนผังคาร์โนท์แบบ 2 อินพุต มีทั้งหมด 2 แถว 2 คอลัมน์ ซึ่งแต่ละแถวจะใช้สำหรับการแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร B และแต่ละคอลัมน์ใช้สำหรับแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร A ดังนี้

ช่องที่ 1 (แถวที่ 1 คอลัมน์ที่ 1): เป็นการแทนค่าของ $A = 0$ และ $B = 0$

ช่องที่ 2 (แถวที่ 2 คอลัมน์ที่ 1): เป็นการแทนค่าของ $A = 0$ และ $B = 1$

ช่องที่ 3 (แถวที่ 1 คอลัมน์ที่ 2): เป็นการแทนค่าของ $A = 1$ และ $B = 0$

ช่องที่ 4 (แถวที่ 2 คอลัมน์ที่ 2): เป็นการแทนค่าของ $A = 1$ และ $B = 1$

6.1.2 แผนผังคาร์โนห์ขนาด 3 ตัวแปร

แผนผังคาร์โนห์ขนาด 3 ตัวแปร ภายในตารางจะมีจำนวนช่องทั้งหมด 8 ช่อง ซึ่งสามารถเขียนได้ 2 แบบคือตารางแนวนอนและตารางแนวตั้งดังต่อไปนี้ (สมมติ A, B และ C เป็นอินพุตใดๆ)

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

รูปที่ 6.2 แผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวนอน

จากรูป 6.2 แสดงแผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวนอน มีทั้งหมด 2 แถว 4 คอลัมน์ ซึ่งแต่ละแถวจะใช้สำหรับการแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร C และแต่ละคอลัมน์ใช้สำหรับแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร A และตัวแปร B ดังนี้

แถวที่ 1 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0$ และ $C = 0$

แถวที่ 2 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0$ และ $C = 1$

แถวที่ 1 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1$ และ $C = 0$

แถวที่ 2 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1$ และ $C = 1$

แถวที่ 1 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1$ และ $C = 0$

แถวที่ 2 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1$ และ $C = 1$

แถวที่ 1 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0$ และ $C = 0$

แถวที่ 2 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0$ และ $C = 1$

| | | | |
|----|----|---|---|
| | A | 0 | 1 |
| BC | 00 | | |
| | 01 | | |
| | 11 | | |
| | 10 | | |

รูปที่ 6.3 แผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวตั้ง

จากรูป 6.3 แสดงแผนผังคาร์โนห์แบบ 3 อินพุตแบบตารางแนวนอน มีทั้งหมด 4 แถว 2 คอลัมน์ ซึ่งแต่ละแถวจะใช้สำหรับการแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร B และ ตัวแปร C และแต่ละคอลัมน์ใช้สำหรับแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร A ดังนี้

แถวที่ 1 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0$ และ $C = 0$

แถวที่ 1 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 1, B = 0$ และ $C = 0$

แถวที่ 2 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0$ และ $C = 1$

แถวที่ 2 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 1, B = 0$ และ $C = 1$

แถวที่ 3 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 1$ และ $C = 1$

แถวที่ 3 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 1, B = 1$ และ $C = 1$

แถวที่ 4 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 1$ และ $C = 0$

แถวที่ 4 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 1, B = 1$ และ $C = 0$

สำหรับแผนผังคาร์โนทที่มีช่องในแนวแถว หรือ คอลัมน์ 4 ช่องต่อ 1 แถวหรือ 1 คอลัมน์ จะเรียงข้อมูลอินพุตเป็น 00, 01, 11 และ 10 ตามลำดับเสมอ ดังแสดงรูปที่ 6.2 และ 6.3 เนื่องจากว่าสถานะของอินพุตที่อยู่แต่ละช่องติดกันในแผนผังคาร์โนทจะต้องมีสถานะของอินพุตต่างกันเพียง 1 ตัวเท่านั้น จึงใช้วิธีการเรียงค่าแบบรหัสเกรย์แทนรหัสเลขฐานสอง

6.1.3 แผนผังคาร์โนทขนาด 4 ตัวแปร

แผนผังคาร์โนทขนาด 4 ตัวแปร ภายในตารางจะมีจำนวนช่องทั้งหมด 16 ช่อง ดังต่อไปนี้ (สมมติ A, B, C และ D เป็นอินพุตใดๆ)

| | | | | | |
|----|----|----|----|----|----|
| | AB | 00 | 01 | 11 | 10 |
| CD | 00 | | | | |
| | 01 | | | | |
| | 11 | | | | |
| | 10 | | | | |

รูปที่ 6.4 แผนผังคาร์โนทแบบ 4 อินพุต

จากรูป 6.4 แสดงแผนผังคาร์โนทแบบ 4 อินพุตซึ่ง มีทั้งหมด 4 แถว 4 คอลัมน์ ซึ่งแต่ละแถวจะใช้สำหรับการแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร C และตัวแปร D และแต่ละคอลัมน์ใช้สำหรับแทนค่าที่เป็นไปได้ทั้งหมดของตัวแปร A และตัวแปร B ดังนี้

แถวที่ 1 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0, C = 0$ และ $D = 0$

แถวที่ 2 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0, C = 0$ และ $D = 1$

แถวที่ 3 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0, C = 1$ และ $D = 1$

แถวที่ 4 คอลัมน์ที่ 1: เป็นการแทนค่าของ $A = 0, B = 0, C = 1$ และ $D = 0$

แถวที่ 1 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1, C = 0$ และ $D = 0$

แถวที่ 2 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1, C = 0$ และ $D = 1$

แถวที่ 3 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1, C = 1$ และ $D = 1$

แถวที่ 4 คอลัมน์ที่ 2: เป็นการแทนค่าของ $A = 0, B = 1, C = 1$ และ $D = 0$

แถวที่ 1 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1, C = 0$ และ $D = 0$

แถวที่ 2 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1, C = 0$ และ $D = 1$

แถวที่ 3 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1, C = 1$ และ $D = 1$

แถวที่ 4 คอลัมน์ที่ 3: เป็นการแทนค่าของ $A = 1, B = 1, C = 1$ และ $D = 0$

แถวที่ 1 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0, C = 0$ และ $D = 0$

แถวที่ 2 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0, C = 0$ และ $D = 1$

แถวที่ 3 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0, C = 1$ และ $D = 1$

แถวที่ 4 คอลัมน์ที่ 4: เป็นการแทนค่าของ $A = 1, B = 0, C = 1$ และ $D = 0$

6.2 การอ่านค่าลงแผนผังคาร์โนห์

การอ่านค่าลงแผนผังคาร์โนห์สามารถทำได้ 2 วิธี ดังนี้ อ่านจากสมการพีชคณิตบูลีนที่แปลงมาจากผลรวมของผลคูณ (หรือแปลงจากตารางความจริงโดยเลือกเฉพาะสถานะเอาต์พุตที่เป็น 1) และอ่านจากสมการพีชคณิตบูลีนที่แปลงมาจากผลคูณของผลรวม (หรือแปลงจากตารางความจริงโดยเลือกเฉพาะสถานะเอาต์พุตที่เป็น 0) ดังต่อไปนี้

กรณีที่ 1 สมการพีชคณิตบูลีนอยู่ในรูปผลรวมของผลคูณ: ให้พิจารณามินเทอมของแต่ละพจน์เพื่อนำไปใส่ในแต่ละช่องของตาราง โดยหากอินพุตไม่มีคอมพลิเมนต์ ค่าของอินพุตตัวดังกล่าวจะมีค่าเป็น 1 แต่หากมีคอมพลิเมนต์ที่อินพุตอินพุตตัวดังกล่าวจะมีค่าเป็น 0 เมื่อทราบสถานะอินพุตทั้งหมดของพจน์ใด ๆ แล้วให้ใส่ "1" ลงช่องในแผนผังคาร์โนห์ที่สถานะของอินพุตทั้งหมดมีค่าตรงกัน เมื่อใส่ "1" ครบทุกช่องแล้ว ช่องที่เหลือจะมีค่าเป็น "0"

กรณีที่ 2 สมการพีชคณิตบูลีนอยู่ในรูปผลคูณของผลรวม: ให้พิจารณาแมกเทอมของแต่ละพจน์เพื่อนำไปใส่ในแต่ละช่องของตาราง โดยหากอินพุตไม่มีคอมพลิเมนต์ค่าของอินพุตตัวดังกล่าวจะมีค่าเป็น 0 แต่หากมีคอมพลิเมนต์ที่อินพุตอินพุตตัวดังกล่าวจะมีค่าเป็น 1 เมื่อทราบสถานะอินพุตทั้งหมดของพจน์ใด ๆ แล้วให้ใส่ "0" ลงช่องในแผนผังคาร์โนห์ที่สถานะของอินพุตทั้งหมดมีค่าตรงกัน เมื่อใส่ "0" ครบทุกช่องแล้ว ช่องที่เหลือจะมีค่าเป็น "1"

ตัวอย่างที่ 6-1 จากสมการต่อไปนี้ $Z = \overline{A}B + AB$ จงเขียนค่าสถานะของอินพุตทั้งหมดลงแผนผังคาร์โนห์

วิธีทำ สมการนี้อยู่ในรูปผลรวมของผลคูณแบบ 2 อินพุต ดังนั้นนำ มินเทอมแต่ละพจน์มาใส่ค่าลงแผนผังคาร์โนห์ดังนี้

$\overline{A}B$ ใส่ "1" ลงช่อง $A = 0$ และ $B = 1$

AB ใส่ "1" ลงช่อง $A = 1$ และ $B = 1$

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

ตัวอย่างที่ 6-2 จากสมการต่อไปนี้ $Z = (\overline{A}+B+C)(A+B+C)(A+B+C)(A+B+C)(A+B+C)$ จงเขียนค่าสถานะของอินพุตทั้งหมดลงแผนผังคาร์โนห์

วิธีทำ สมการนี้อยู่ในรูปผลคูณของผลรวมแบบ 3 อินพุต ดังนั้นนำ แมกเทอมแต่ละพจน์มาใส่ค่าลงแผนผังคาร์โนห์ดังนี้

$\overline{A}+B+C$ ใส่ "0" ลงช่อง $A = 1$ และ $B = 1$ และ $C = 0$

$A+B+C$ ใส่ "0" ลงช่อง $A = 0$ และ $B = 1$ และ $C = 0$

$A+B+C$ ใส่ "0" ลงช่อง $A = 0$ และ $B = 1$ และ $C = 1$

$\overline{A}+B+C$ ใส่ "0" ลงช่อง $A = 1$ และ $B = 0$ และ $C = 0$

- $A+B+C$ ใส่ "0" ลงช่อง $A = 0$ และ $B = 0$ และ $C = 0$

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

6.3 การสร้างรูป

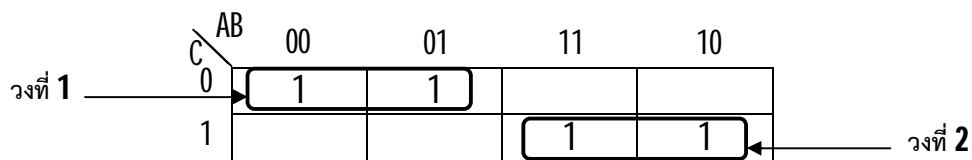
การสร้างรูปเป็นขั้นตอนของการแก้สมการให้อยู่ในรูปที่ง่ายขึ้น โดยเป็นค่ายุบรวมตำแหน่งของช่องที่มีค่าเป็น 1 ติดกันให้ครบทุกช่อง (หรือ 0 ติดกัน) สำหรับขั้นตอนที่จะกล่าวต่อไปนี้เป็นกรณีที่พิจารณาเฉพาะช่องที่มีค่าเป็น 1 โดยมีหลักการดังนี้

1. เลือกช่องดำเนินการ คือเลือกเฉพาะช่องที่มีค่าเป็น 1 ทั้งหมด
2. ยุบรวมช่องที่มี 1 ติดกัน โดยแต่ละวงจะยุบได้จำนวน 2^n ช่องเท่านั้น เช่น 1, 2, 4, 8, 16,... แต่เนื่องจากแผนผังคาร์โนห์ที่กล่าวถึงในเอกสารเล่มนี้มีขนาด 2 - 4 อินพุต ดังนั้น จะยุบช่องที่มี 1 ติดกันได้สูงสุด 16 ช่อง
3. การยุบรวมจะใช้วิธีการวงรอบกลุ่มที่มีค่า 1 อยู่ติดกัน โดยจะต้องวงให้ได้วงที่ใหญ่ที่สุดเท่าที่จะเป็นไปได้ เช่นถ้าวงได้ 4 ช่อง ห้ามวงเป็นวงเล็ก 2 วง ๆ ละ 2 ช่อง
4. ช่องที่อยู่ตำแหน่งขอบของตาราง หากสมมติว่าพับแผ่นตารางแล้วพบว่าช่องทับกัน ถือว่าอยู่ติดกัน
5. ช่องที่ถูกวงแล้วสามารถถูกนำไปวงกับวงอื่นได้อีก เพื่อให้วงดังกล่าวมีขนาดใหญ่ขึ้น
6. นำวงแต่ละวงที่ยุบได้มาแปลงเป็นมินเทอม วิธีการแปลงให้ดูสถานะของอินพุตทั้งหมดที่ตำแหน่งของวงดังกล่าว (มองขึ้น และมองไปทางซ้าย) โดยกรณีที่พบอินพุตตัวเดียวกันเป็นทั้ง 0 และ 1 ให้หากล้างทิ้ง กรณีที่อินพุตตัวเดียวกันมีค่าเป็น 1 เหมือนกันทั้งหมดให้ใส่อินพุตตัวดังกล่าวเพียงตัวเดียว แต่ถ้าพบอินพุตตัวเดียวกันมีค่าเป็น 0 ทั้งหมดให้ใส่อินพุตตัวดังกล่าวเพียงตัวเดียว และให้ใส่คอมพลิเมนต์ให้อินพุตตัวดังกล่าว
7. นำมินเทอมทั้งหมดมาผ่านตัวดำเนินการ ออร์

ตัวอย่างที่ 6-3 จงหาสมการพีชคณิตบูลีนที่อยู่ในรูปที่ง่ายที่สุดจากแผนผังคาร์โนห์ต่อไปนี้

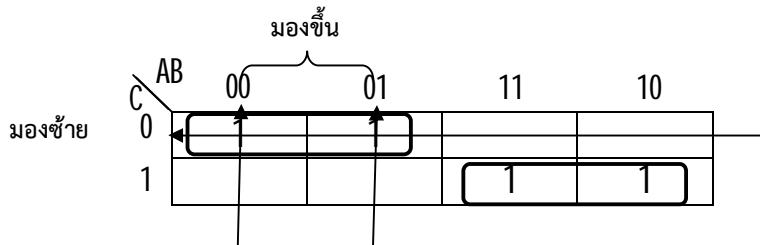
| | | | | | |
|---|----|----|----|----|----|
| | AB | 00 | 01 | 11 | 10 |
| C | 0 | 1 | 1 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 1 |

วิธีทำ เนื่องจากพิจารณาเฉพาะช่องที่มีค่าเป็น 1 ทั้งหมด จึงสามารถลบช่องที่มีค่าเป็น 0 ออกก่อนได้ และจากตัวอย่างพบว่ามี 1 ติดกัน 2 ช่อง อยู่ 2 จุด จึงมีการวงเพื่อยุบรวม 2 ช่องอยู่ 2 วงดังนี้



การแปลงเป็นมินเทอมแต่ละพจน์ให้มองขึ้น และมองไปทางซ้ายตามตำแหน่งของช่องทั้งหมดที่ถูกวง
ดังนี้

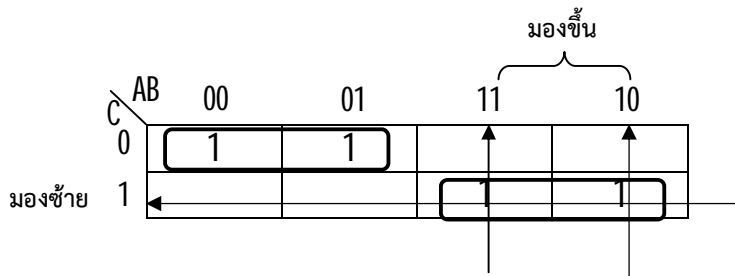
วงที่ 1



มองขึ้น: พบ 00 และ 01 สังเกตว่าตัวหน้าเป็นของ A และมีค่าเป็น 0 จึงใส่คอมพลิเมนต์ด้วยได้เป็น **A** ส่วนตัวหลังเป็นของ B และเนื่องจากว่าพบทั้ง 0 และ 1 (00, 01) จึงหักล้างออกไป

มองซ้าย: พบ 0 ซึ่งเป็นของ C และใส่คอมพลิเมนต์จึงได้เป็น **C**
ดังนั้นได้มินเทอมของพจน์วงที่ 1 เป็น **AC**

วงที่ 2



มองขึ้น: พบ 11 และ 10 สังเกตว่าตัวหน้าเป็นของ A และมีค่าเป็น 1 ได้เป็น **A** ส่วนตัวหลังเป็นของ B และเนื่องจากว่าพบทั้ง 0 และ 1 (11, 10) จึงหักล้างออกไป

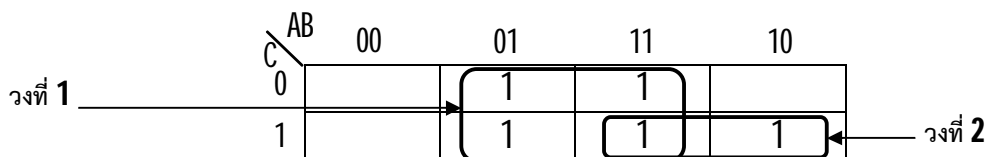
มองซ้าย: พบ 1 ซึ่งเป็นของ C ได้เป็น **C**
ดังนั้นได้มินเทอมของพจน์วงที่ 2 เป็น **AC**

เพราะฉะนั้นเมื่อรวมมินเทอมทั้งสองพจน์ได้สมการพีชคณิตบูลีนเป็น **AC + AC**

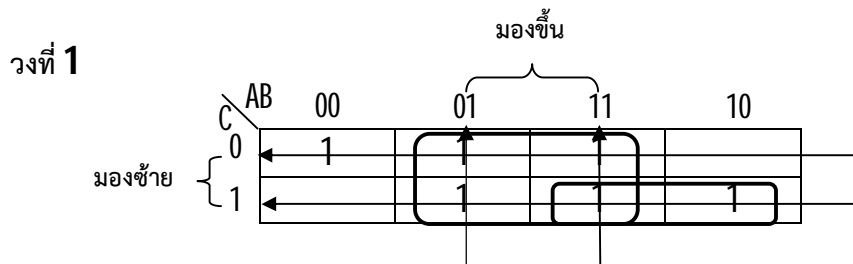
ตัวอย่างที่ 6-4 จงหาสมการพีชคณิตบูลีนที่อยู่ในรูปที่ง่ายที่สุดจากแผนผังคาร์โนห์ต่อไปนี้

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

วิธีทำ จากตัวอย่างพบว่ามี 1 ติดกัน 4 ช่อง อยู่ 1 จุด และติดกัน 2 ช่อง 1 จุด จึงยุบรวม ได้ 2 วง ดังนี้

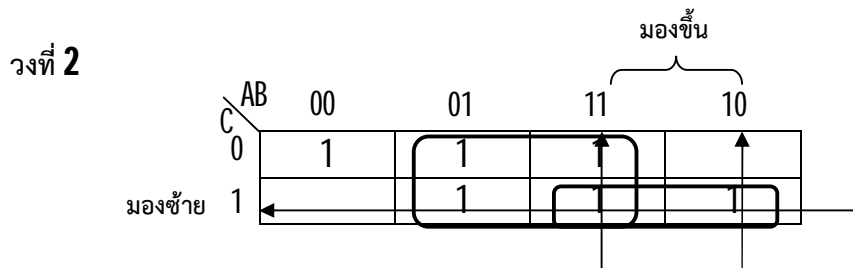


การแปลงเป็นมินเทอมแต่ละพจน์ให้มองขึ้น และมองไปทางซ้ายตามตำแหน่งของช่องทั้งหมดที่ถูกวง ดังนี้



มองขึ้น: พบ 01 และ 11 สังเกตว่าตัวหน้าเป็นของ A และเนื่องจากว่าพบทั้ง 0 และ 1 (01, 11) จึงหักล้างออกไปส่วนตัวหลังเป็นของ B ซึ่งมีค่าเป็น 1 จึงได้เป็น B

มองซ้าย: พบ 0 และ 1 ซึ่งเป็นของ C จึงหักล้างออกไป
 ดังนั้นได้มินเทอมของพจน์วงที่ 1 เป็น B



มองขึ้น: พบ 11 และ 10 สังเกตว่าตัวหน้าเป็นของ A และมีค่าเป็น 1 ได้เป็น A ส่วนตัวหลังเป็นของ B และเนื่องจากว่าพบทั้ง 0 และ 1 (11, 10) จึงหักล้างออกไป

มองซ้าย: พบ 1 ซึ่งเป็นของ C ได้เป็น C
 ดังนั้นได้มินเทอมของพจน์วงที่ 2 เป็น AC

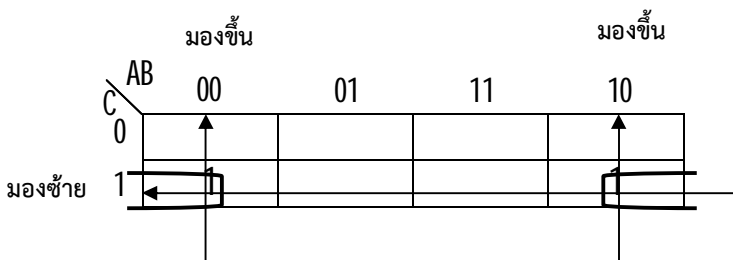
เพราะฉะนั้นเมื่อรวมมินเทอมทั้งสองพจน์ได้สมการพีชคณิตบูลีนเป็น $B + AC$

ตัวอย่างที่ 6-4 แสดงให้เห็นว่าช่องที่ถูกวงแล้วสามารถถูกนำไปยุบรวมกับวงอื่นได้อีกเพื่อให้มินเทอมมีขนาดที่เล็กลง (จากตัวอย่างคือช่องที่ A, B, C มีค่าเป็น 1, 1, 1 ตามลำดับ)

ตัวอย่างที่ 6-5 จงหาสมการพีชคณิตบูลีนที่อยู่ในรูปที่ง่ายที่สุดจากแผนผังคาร์โนห์ต่อไปนี้

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

วิธีทำ จากตัวอย่าง หากลองพิจารณาตามแนวคอลัมน์พบว่าตำแหน่งที่ A, B, C มีค่าเป็น 001 และ 101 จะทับกันพอดีเพราะฉะนั้นสรุปได้ว่า 2 ช่องดังกล่าวอยู่ติดกัน ดังนี้



มองขึ้น: พบ 00 และ 10 สังเกตว่าตัวหน้าเป็นของ A และเนื่องจากว่าพบทั้ง 0 และ 1 (00, 10) จึงหักล้างออกไปส่วนตัวหลังเป็นของ B และมีค่าเป็น 0 ได้เป็น B

มองซ้าย: พบ 1 ซึ่งเป็นของ C ได้เป็น C

ดังนั้นได้มินเทอมเป็น BC

เนื่องจากการยุบรวมเพียงวงเดียวจึงได้สมการพีชคณิตบูลีนคือ BC

ตัวอย่างที่ 6-6 จงแก้สมการพีชคณิตบูลีน $Z = ABCD + ABCD + ABCD + ABCD + ABCD + ABCD$

ให้อยู่ในรูปที่ง่ายที่สุดโดยใช้แผนผังคาร์โนห์

วิธีทำ เนื่องจากเป็นสมการแบบ 4 อินพุตจึงใช้แผนผังคาร์โนห์ขนาด 16 ช่อง และเติม 1 ใส่ช่องที่ตรงกับมินเทอมของสมการพีชคณิตบูลีนข้างต้นในแต่ละพจน์ ดังนี้

มินเทอมตัวที่ 1: ABCD เติม 1 ใส่ช่อง A = 0, B = 0, C = 0 และ D = 0

มินเทอมตัวที่ 2: ABCD เติม 1 ใส่ช่อง A = 0, B = 0, C = 0 และ D = 1

มินเทอมตัวที่ 3: ABCD เติม 1 ใส่ช่อง A = 0, B = 1, C = 0 และ D = 0

มินเทอมตัวที่ 4: ABCD เติม 1 ใส่ช่อง A = 0, B = 1, C = 0 และ D = 1

มินเทอมตัวที่ 5: ABCD เติม 1 ใส่ช่อง A = 1, B = 0, C = 0 และ D = 0

มินเทอมตัวที่ 6: ABCD เติม 1 ใส่ช่อง A = 1, B = 1, C = 1 และ D = 1

นำมินเทอมทั้ง 6 ค่าใส่ลงในแผนผังคาร์โนห์ ดังนี้

| | | | | |
|---------|----|----|----|----|
| AB \ CD | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | | 1 |
| 01 | 1 | 1 | | |
| 11 | | | 1 | |
| 10 | | | | |

จากแผนผังคาร์โนท์สามารถยุบรวมได้ดังนี้

| | | | | |
|---------|----|----|----|----|
| CD \ AB | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | | 1 |
| 01 | 1 | 1 | | |
| 11 | | | 1 | |
| 10 | | | | |

วงที่ 1: (00,01) in (00,01) and (01,01) in (00,01)
 วงที่ 2: (00,10) in (00,00) and (01,10) in (00,00)
 วงที่ 3: (11,11) in (11,11)

วงที่ 1

| | | | | |
|---------|----|----|----|----|
| CD \ AB | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | | 1 |
| 01 | 1 | 1 | | |
| 11 | | | 1 | |
| 10 | | | | |

มองขึ้น: (00,01) in (00,01) and (01,01) in (00,01)
 มองซ้าย: (00,01) in (00,00) and (01,01) in (00,00)

มองขึ้น: พบ 00 และ 01 สังเกตว่าตัวหน้าเป็นของ A และมีค่าเป็น 0 จึงใส่คอมพลิเมนต์ด้วยได้เป็น \bar{A} ส่วนตัวหลังเป็นของ B และเนื่องจากว่าพบทั้ง 0 และ 1 (00, 01) จึงหักล้างออกไป

มองซ้าย: พบ 00 และ 01 สังเกตว่าตัวหน้าเป็นของ C และมีค่าเป็น 0 จึงใส่คอมพลิเมนต์ด้วยได้เป็น \bar{C} ส่วนตัวหลังเป็นของ D และเนื่องจากว่าพบทั้ง 0 และ 1 (00, 01) จึงหักล้างออกไป

ดังนั้นได้มีนเทอมของพจน์วงที่ 1 เป็น $\bar{A}\bar{C}$

วงที่ 2

| | | | | |
|---------|----|----|----|----|
| CD \ AB | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | | 1 |
| 01 | 1 | 1 | | |
| 11 | | | 1 | |
| 10 | | | | |

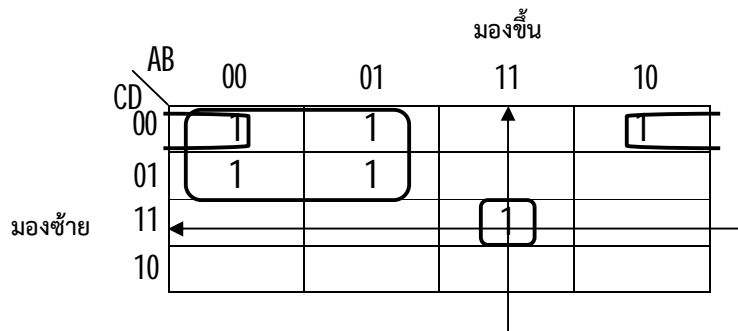
มองขึ้น: (00,01) in (00,01) and (01,01) in (00,01)
 มองซ้าย: (00,10) in (00,00) and (01,10) in (00,00)

มองขึ้น: พบ 00 และ 10 สังเกตว่าส่วนตัวหน้าเป็นของ A และเนื่องจากว่าพบทั้ง 0 และ 1 (00, 10) จึงหักล้างออกไปส่วนตัวหลังเป็นของ B และมีค่าเป็น 0 ได้เป็น B

มองซ้าย: พบ 00 ซึ่งเป็นของ C และ D ได้เป็น CD

ดังนั้นได้มินเทอมของพจน์วงที่ 2 เป็น BCD

วงที่ 3



มองขึ้น: พบ 11 ซึ่งเป็นของ A และ B ได้เป็น AB

มองซ้าย: พบ 11 ซึ่งเป็นของ C และ D ได้เป็น CD

ดังนั้นได้มินเทอมของพจน์วงที่ 3 เป็น ABCD

เพราะฉะนั้นเมื่อรวมมินเทอมทั้งสามพจน์ได้สมการพีชคณิตบูลีนเป็น $AC + BCD + ABCD$

ตัวอย่างที่ 6-7 จงเขียนสมการพีชคณิตบูลีนจากตารางความจริงต่อไปนี้

| อินพุต | | | เอาต์พุต |
|--------|---|---|----------|
| A | B | C | Z |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

วิธีทำ จากตารางความจริงเลือกเฉพาะมินเทอมของอินพุตที่สถานะของเอาต์พุตมีค่าเป็น 1 ได้ดังนี้

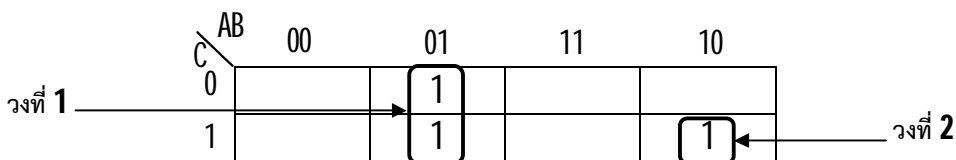
มินเทอมตัวที่ 1: $A = 0, B = 1$ และ $C = 0$

มินเทอมตัวที่ 2: $A = 0, B = 1$ และ $C = 1$

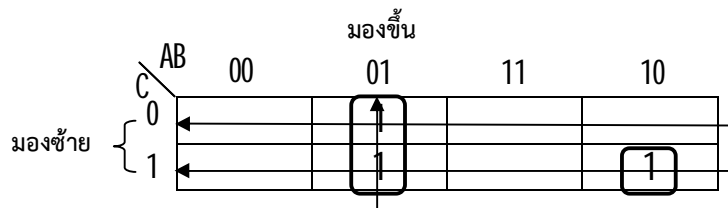
มินเทอมตัวที่ 3: $A = 1, B = 0$ และ $C = 1$

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

จากแผนผังคาร์โนท์สามารถยุบรวมได้ดังนี้



วงที่ 1

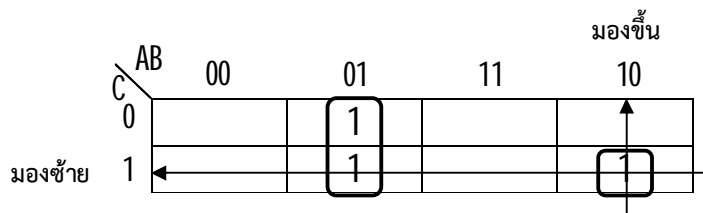


มองขึ้น: พบ 01 ซึ่งเป็นของ A และ B ได้เป็น AB

มองซ้าย: พบ 0 และ 1 ซึ่งเป็นของ C จึงหักล้างออกไป

ดังนั้นได้มินเทอมของพจน์วงที่ 1 เป็น AB

วงที่ 2



มองขึ้น: พบ 10 ซึ่งเป็นของ A และ B ได้เป็น AB

มองซ้าย: พบ 1 ซึ่งเป็นของ C ได้เป็น C

ดังนั้นได้มินเทอมของพจน์วงที่ 2 เป็น ABC

เพราะฉะนั้นเมื่อรวมมินเทอมทั้งสองพจน์ได้สมการพีชคณิตบูลีนเป็น $AB + ABC$

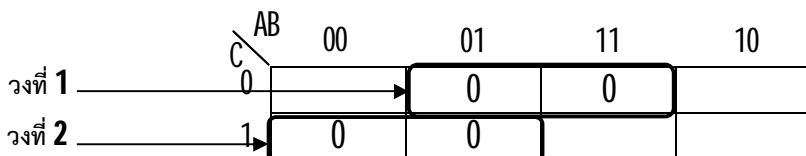
ตัวอย่างที่ 6-8 จงหาสมการพีชคณิตบูลีนที่อยู่ในรูปที่ง่ายที่สุดจากแผนผังคาร์โนท์ต่อไปนี้ โดยให้อยู่ในรูปของผลคูณของผลรวม

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

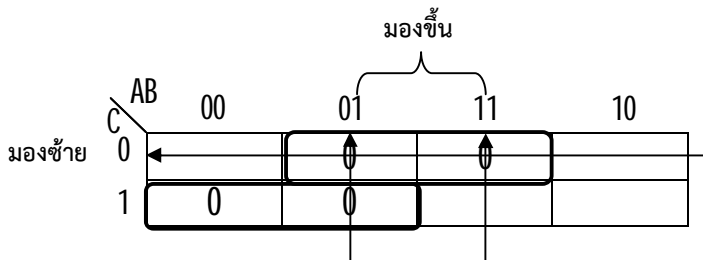
วิธีทำ จากโจทย์กำหนดให้หาสมการพีชคณิตบูลีนโดยให้อยู่ในรูปของผลคูณของผลรวม ซึ่งจะแตกต่างจากตัวอย่างก่อนหน้านี้ทั้งหมดที่ได้เคยแสดงมา เนื่องจากตัวอย่างก่อนหน้านี้ทั้งหมดจะหาสมการพีชคณิตบูลีนซึ่งอยู่ในรูปผลรวมของผลคูณทั้งหมด โดยวิธีการหาสมการพีชคณิตบูลีนในรูปผลคูณของผลรวม จะแตกต่างจากวิธีการหาสมการพีชคณิตบูลีนในรูปผลรวมของผลคูณ ดังนี้

1. เลือกช่องดำเนินการ 0 ทั้งหมด โดยที่ช่องดำเนินการ 1 ทั้งหมดให้อาออก
2. ขั้นตอนการยุบรวมเหมือนกับแบบวิธีการหาแบบผลรวมของผลคูณทุกประการ
3. นำวงแต่ละวงที่ยุบได้มาแปลงเป็นแมกเทอม
4. นำแมกเทอมทั้งหมดมาผ่านตัวดำเนินการ แอนด์

ดังนั้น จากหลักการขั้นต้น เลือกเฉพาะช่องดำเนินการ 0 ทั้งหมดได้ ดังนี้



วงที่ 1

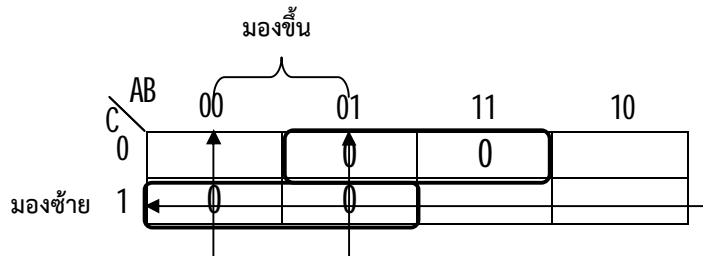


มองขึ้น: พบ 01 และ 11 สังเกตว่าตัวหน้าเป็นของ A และเนื่องจากว่าพบทั้ง 0 และ 1 (01, 11) จึงหักล้างออกไป ส่วนตัวหลังเป็นของ B และมีค่าเป็น 1 จึงใส่คอมพลิเมนต์ด้วยได้เป็น B

มองซ้าย: พบ 0 ซึ่งเป็นของ C ได้เป็น C

ดังนั้นได้แมกเทอมของพจน์วงที่ 1 เป็น $B + C$

วงที่ 2



มองขึ้น: พบ 00 และ 01 สังเกตว่าตัวหน้าเป็นของ A และ มีค่าเป็น 0 ได้เป็น A ส่วนตัวหลังเป็นของ B เนื่องจากว่าพบทั้ง 0 และ 1 (00, 01) จึงหักล้างออกไป

มองซ้าย: พบ 1 ซึ่งเป็นของ C ได้เป็น C

ดังนั้นได้แมกเทอมของพจน์วงที่ 2 เป็น A+C

เพราะฉะนั้นเมื่อรวมแมกเทอมทั้งสองพจน์ได้สมการพีชคณิตบูลีนเป็น $(B+C)(A+C)$

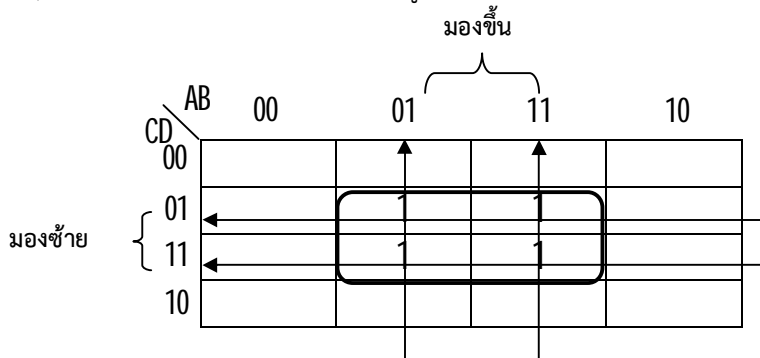
6.4 เงื่อนไขที่ไม่สนใจ (Don't Care Condition)

เงื่อนไขที่ไม่สนใจคือ การกำหนดให้สถานะของมินเทอม หรือแมกเทอมเป็นได้ทั้ง 0 หรือ 1 ประโยชน์ของการนำเงื่อนไขที่ไม่สนใจมาใช้งานคือจะช่วยให้สามารถลดรูปสมการพีชคณิตบูลีนมีขนาดที่เล็กลงได้ ข้อควรระวังคือ เงื่อนไขที่ไม่สนใจอาจทำให้สมการพีชคณิตบูลีนมีขนาดใหญ่ขึ้น เพราะฉะนั้นจึงต้องระวังไม่ให้เกิดกรณีดังกล่าวขึ้นด้วย สำหรับมินเทอม หรือแมกเทอมใดที่มีสถานะเป็นเงื่อนไขที่ไม่สนใจให้ใช้สัญลักษณ์ "x" หรือ สัญลักษณ์ "d" แทนสถานะดังกล่าว โดยที่เอกสารประกอบการสอนเล่มนี้จะเลือกใช้เฉพาะสัญลักษณ์ "x" เท่านั้น

ตัวอย่างที่ 6-9 จงเขียนสมการพีชคณิตบูลีนจากแผนผังคาร์โนห์ต่อไปนี้

| | | | | | |
|----|----|----|----|----|----|
| | AB | 00 | 01 | 11 | 10 |
| CD | 00 | | | | |
| | 01 | | 1 | x | |
| | 11 | | x | 1 | |
| | 10 | | | x | |

วิธีทำ จากแผนผังคาร์โนห์สังเกตว่ามีสถานะที่เป็นเงื่อนไขที่ไม่สนใจทั้งหมด 3 ค่า ซึ่งอยู่ตำแหน่งที่มีมินเทอม "ABCD" มีสถานะอินพุตเป็น "0111", "1101" และ "1110" เนื่องจากสามารถมองให้สถานะที่เป็นเงื่อนไขที่ไม่สนใจเป็นได้ทั้ง "0" หรือ "1" จึงกำหนดให้สถานะที่เป็นเงื่อนไขที่ไม่สนใจที่อยู่ตำแหน่ง "ABCD" ที่มีสถานะอินพุตเป็น "0111" และ "1101" มีค่าเป็น "1" เพื่อให้สามารถยุบรวมได้วงที่ใหญ่ขึ้น (ยุบ 4 ช่อง) ส่วนสถานะที่เป็นเงื่อนไขที่ไม่สนใจอีกค่าหนึ่ง ("ABCD" มีค่าเป็น "1110") ให้กำหนดเป็น "0" เพื่อที่จะไม่ได้ถูกนำมาใช้งาน ดังนี้

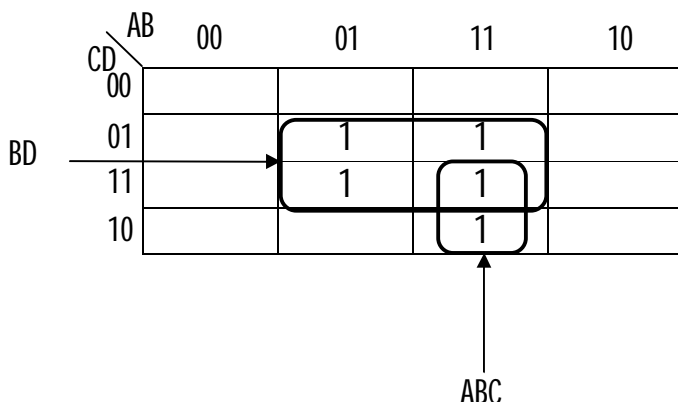


มองขึ้น: พบ 01 และ 11 สังเกตว่าตัวหน้าเป็นของ A เนื่องจากว่าพบทั้ง 0 และ 1 (01, 11) จึงหักล้างออกไป ส่วนตัวหลังเป็นของ B และ มีค่าเป็น 1 ได้เป็น B

มองซ้าย: พบ 01 และ 11 สังเกตว่าตัวหน้าเป็นของ C เนื่องจากว่าพบทั้ง 0 และ 1 (01, 11) จึงหักล้างออกไป ส่วนตัวหลังเป็นของ D และ มีค่าเป็น 1 ได้เป็น D

เพราะฉะนั้นได้สมการพีชคณิตบูลีนเป็น BD

จากตัวอย่างนี้หากมองสถานะที่เป็นเงื่อนไขที่ไม่สนใจของ "ABCD" ที่มีสถานะอินพุตเป็น "1110" ให้มีสถานะเป็น "1" จะพบว่าจะทำให้สมการพีชคณิตบูลีนมีขนาดที่ใหญ่ขึ้น ดังนี้



จากแผนผังคาร์โนห์ข้างต้นจะได้สมการพีชคณิตบูลีนเป็นดังนี้ $ABC + BD$ ซึ่งสังเกตได้ว่าสมการที่ได้เป็นสมการที่มีขนาดใหญ่กว่าสมการที่เป็นคำตอบที่ถูกต้อง (ถึงแม้จะได้สถานะเอาต์พุตที่เหมือนกัน) ซึ่งส่งผลให้มีการใช้งานจำนวนไอซีมากขึ้น

ตัวอย่างที่ 6-10 กำหนดให้ A, B, C และ D แทนเลขฐานสองที่เรียงกันตามลำดับความสำคัญจากมากไปหาน้อย โดย A เป็นบิตที่มีลำดับความสำคัญสูงสุดและ D เป็นบิตที่มีลำดับความสำคัญต่ำที่สุด จงเขียนสมการพีชคณิตบูลีนที่ทำให้เอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่อสถานะของอินพุตซึ่งมีการใช้งานเพียงแค่ 10 ค่าคือ 0000, 0001, ..., 1001 มีค่าเป็น 1 พร้อมกัน 2 ค่า

วิธีทำ จากตัวอย่างโจทย์กำหนดให้มีการใช้งานอินพุตเพียงแค่ 10 ค่า แต่เนื่องจากอินพุตมีจำนวนทั้งหมด 4 บิตซึ่งมีค่าที่เป็นไปได้ทั้งหมด 16 ค่า ดังนั้นอีก 6 ค่าที่เหลือ (1010, 1011, ..., 1111) จะเป็นเงื่อนไขที่ไม่สนใจนั่นเอง และหากเอาต์พุตจะมีค่าเป็น 1 ได้ สถานะของอินพุตจะต้องเป็น 1 พร้อมกัน 2 ค่าเท่านั้น ดังตารางต่อไปนี้

| สถานะอินพุต | | | | สถานะเอาต์พุต (Z) |
|-------------|---|---|---|----------------------|
| A | B | C | D | |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |

| สถานะอินพุต | | | | สถานะเอาต์พุต (Z) |
|-------------|---|---|---|----------------------|
| A | B | C | D | |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

จากตารางความจริงสามารถนำมาเขียนลงแผนผังคาร์โนห์ได้ดังนี้

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | 0 | 0 | X | 0 |
| 01 | 0 | 1 | X | 1 |
| 11 | 1 | 0 | X | X |
| 10 | 0 | 1 | X | X |

จากที่ได้เคยกล่าวมาแล้วว่า การเขียนสมการการบูลีนจากแผนผังคาร์โนห์สามารถทำได้ 2 วิธี คือเลือกช่องที่มีค่าเป็น 1 ทั้งหมด และ เงื่อนไขที่ไม่สนใจบางช่อง หรือ เลือกช่องที่มีค่าเป็น 0 ทั้งหมด และเงื่อนไขที่ไม่สนใจบางช่อง ดังนั้นจากตัวอย่างผู้เขียนเลือกใช้ช่องที่มีค่าเป็น 1 ทั้งหมดและเงื่อนไขที่ไม่สนใจบางช่องดังนี้

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| BCD | | | X | |
| BCD | | 1 | X | 1 |
| BCD | 1 | | X | X |
| BCD | | 1 | X | X |

← AD

เพราะฉะนั้นได้สมการพีชคณิตบูลีนเป็น $AD + BCD + BCD + BCD$

6.5 การออกแบบวงจรรวมและเลขฐานสอง

วงจรรวม และวงจรถลเลขฐานสอง คือวงจรมีการนำอินพุต 2 ตัว ที่มีขนาด 1 บิตและมีลำดับความสำคัญที่เท่ากัน มาดำเนินการบวก หรือ ลบกันโดยกรณีที่เป็นวงจรรวมผลลัพธ์ที่ได้จะเป็นผลรวม และตัวทดในบิตถัดไป แต่หากเป็นวงจรถลผลลัพธ์ที่ได้จะเป็นผลลบ และตัวยืมจากบิตที่มี

ลำดับความสำคัญที่สูงกว่า 1 ตำแหน่ง โดยแบ่งออกเป็น 4 วงจรประกอบด้วยวงจรบวกแบบไม่คิดตัวทด (Half Adder), วงจรบวกแบบคิดตัวทด (Full Adder), วงจรลบแบบไม่คิดตัวยืม (Half Subtractor) และวงจรลบแบบคิดตัวยืม (Full Subtractor)

6.5.1 วงจรบวกแบบไม่คิดตัวทด

วงจรบวกแบบไม่คิดตัวทดคือ วงจรที่เกิดจากการนำบิตที่มีลำดับความสำคัญต่ำที่สุดมาผ่านการดำเนินการบวกกันซึ่งผลลัพธ์ที่ได้จะมี 2 ค่าคือผลบวก และตัวทดในบิตถัดไป



รูปที่ 6.5 โครงสร้างวงจรบวกแบบไม่คิดตัวทด

รูปที่ 6.5 แสดงโครงสร้างวงจรบวกแบบไม่คิดตัวทด การทำงานของวงจรคือจะนำอินพุต A และ B ซึ่งมีขนาด 1 บิตมาบวกกัน โดยเอาต์พุตที่ได้จะมี 2 ค่าคือผลบวก (Sum) และตัวทด (Carry) เนื่องจากอินพุตมี 2 ตัว ดังนั้นค่าที่เป็นไปได้ทั้งหมดมี 4 กรณี ดังนี้

กรณีที่ 1: $A = 0$ และ $B = 0$

$$\begin{array}{r}
 \text{Carry} \\
 \downarrow \\
 0 \quad (A) \\
 + \\
 0 \quad (B) \\
 \hline
 0 \quad (\text{Sum})
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 0, \text{Carry} = 0$

กรณีที่ 2: $A = 0$ และ $B = 1$

$$\begin{array}{r}
 \text{Carry} \\
 \downarrow \\
 0 \quad (A) \\
 + \\
 1 \quad (B) \\
 \hline
 1 \quad (\text{Sum})
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 1, \text{Carry} = 0$

กรณีที่ 3: $A = 1$ และ $B = 0$

$$\begin{array}{r}
 \text{Carry} \\
 \downarrow \\
 0 \\
 \underline{1} \quad (A) \\
 + \\
 \underline{0} \quad (B) \\
 \hline
 \underline{1} \quad (\text{Sum})
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 1, \text{Carry} = 0$

กรณีที่ 4: $A = 1$ และ $B = 1$

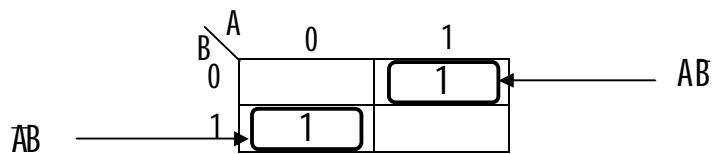
$$\begin{array}{r}
 \text{Carry} \\
 \downarrow \\
 1 \\
 \underline{1} \quad (A) \\
 + \\
 \underline{1} \quad (B) \\
 \hline
 \underline{0} \quad (\text{Sum})
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 0, \text{Carry} = 1$

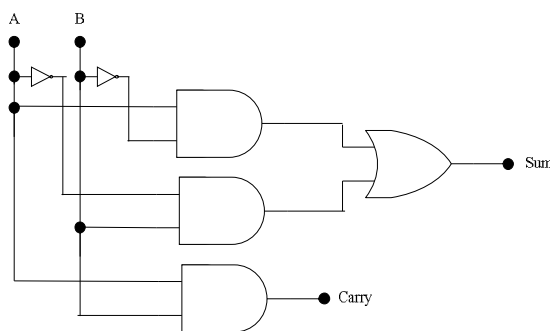
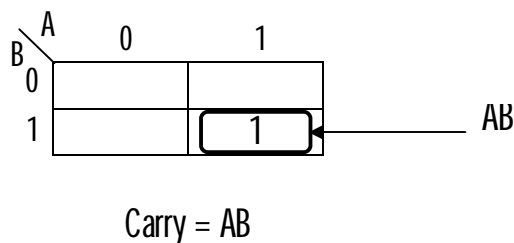
ตารางที่ 6.1 ตารางความจริงของวงจรวกแบบไม่คิดตัวทด

| อินพุต | | เอาต์พุต | |
|--------|---|----------|-------|
| A | B | Sum | Carry |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

จากตารางความจริงสามารถคำนวณคำนวณหาสมการพีชคณิตบูลีนของ Sum และ Carry ได้ดังนี้



$$\text{Sum} = AB + AB$$



รูปที่ 6.6 วงจรบวกแบบไม่คิดตัวทด

6.5.2 วงจรบวกแบบคิดตัวทด

วงจรบวกแบบคิดตัวทคือ วงจรที่เกิดจากการนำอินพุตขนาด 1 บิต 2 ตัวมาบวกกัน แต่ อินพุตทั้ง 2 ตัวจะต้องเป็นบิตที่มีลำดับความสำคัญที่เท่ากันและเป็นบิตที่ไม่ใช่ตำแหน่งบิตที่มีลำดับ ความสำคัญต่ำที่สุด ดังนั้นการบวกจึงมีความเป็นไปได้ที่จะต้องนำตัวทดเข้า (Carry_in) เข้ามาคิดร่วม ด้วย อธิบายได้ดังนี้สมมติว่าต้องการบวกเลขขนาด 2 บิตคือ $A_1A_0 + B_1B_0$ โดยบิตที่เราสนใจคือการ บวกระหว่าง A_1 และ B_1 เท่านั้น แต่เนื่องจาก A_1 และ B_1 เป็นบิตที่ไม่ได้เป็นบิตที่มีลำดับความสำคัญ ต่ำที่สุด ดังนั้นจึงมีความเป็นไปได้ที่จะต้องนำตัวทดเข้าซึ่งเกิดจากการบวกกันเกินของ A_0 และ B_0 เข้า มาบวกร่วมด้วย ซึ่งผลลัพธ์ที่ได้จะมี 2 ค่าคือผลบวก และตัวทดในบิตถัดไป



รูปที่ 6.7 โครงสร้างวงจรบวกแบบไม่คิดตัวทด

รูปที่ 6.7 แสดงโครงสร้างวงจรบวกแบบคิดตัวทด การทำงานของวงจรก็จะนำอินพุต A, B และ Carry_in ซึ่งมีขนาด 1 บิตมาบวกกัน โดยเอาต์พุตที่ได้จะมี 2 ค่าคือผลบวก (Sum) และตัวทด (Carry_out) เนื่องจากอินพุตมี 3 ตัว ดังนั้นค่าที่เป็นไปได้ทั้งหมดมี 8 กรณี ดังนี้

กรณีที่ 1: $A = 0, B = 0$ และ $\text{Carry}_{in} = 0$

$$\begin{array}{r}
 \text{Carry}_{out} \quad \text{Carry}_{in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 0 \quad 0 \\
 \text{0} \quad (\text{A}) \\
 + \\
 \underline{\text{0}} \quad (\text{B}) \\
 \underline{\text{0}} \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 0, \text{Carry}_{out} = 0$

กรณีที่ 2: $A = 0, B = 0$ และ $\text{Carry}_{in} = 1$

$$\begin{array}{r}
 \text{Carry}_{out} \quad \text{Carry}_{in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 0 \quad 1 \\
 \text{0} \quad (\text{A}) \\
 + \\
 \underline{\text{0}} \quad (\text{B}) \\
 \underline{\text{1}} \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 1, \text{Carry}_{out} = 0$

กรณีที่ 3: $A = 0, B = 1$ และ $\text{Carry}_{in} = 0$

$$\begin{array}{r}
 \text{Carry}_{out} \quad \text{Carry}_{in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 0 \quad 0 \\
 \text{0} \quad (\text{A}) \\
 + \\
 \underline{\text{1}} \quad (\text{B}) \\
 \underline{\text{1}} \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sum} = 1, \text{Carry}_{out} = 0$

กรณีที่ 4: $A = 0, B = 1$ และ $\text{Carry}_{in} = 1$

$$\begin{array}{r}
 \text{Carry}_{out} \quad \text{Carry}_{in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 1 \quad 1 \\
 \text{0} \quad (\text{A}) \\
 + \\
 \underline{\text{1}} \quad (\text{B}) \\
 \underline{\text{0}} \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: Sum = 0, Carry_out = 1

กรณีที่ 5: A = 1, B = 0 และ Carry_in = 0

$$\begin{array}{r}
 \text{Carry_out} \quad \text{Carry_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 0,0 \\
 1 \quad (A) \\
 + \\
 0 \quad (B) \\
 \hline
 1 \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: Sum = 1, Carry_out = 0

กรณีที่ 6: A = 1, B = 0 และ Carry_in = 1

$$\begin{array}{r}
 \text{Carry_out} \quad \text{Carry_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 1,1 \\
 1 \quad (A) \\
 + \\
 0 \quad (B) \\
 \hline
 0 \quad (\text{Sum})
 \end{array}
 \end{array}$$

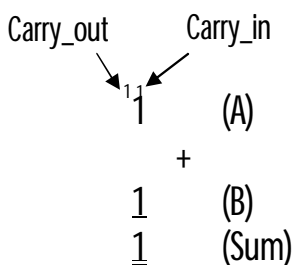
ผลลัพธ์ที่ได้: Sum = 0, Carry_out = 1

กรณีที่ 7: A = 1, B = 1 และ Carry_in = 0

$$\begin{array}{r}
 \text{Carry_out} \quad \text{Carry_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 1,0 \\
 1 \quad (A) \\
 + \\
 1 \quad (B) \\
 \hline
 0 \quad (\text{Sum})
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: Sum = 0, Carry_out = 1

กรณีที่ 8: $A = 1, B = 1$ และ $Carry_{in} = 1$

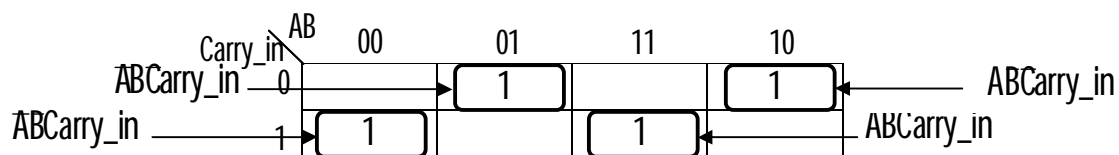


ผลลัพธ์ที่ได้: $Sum = 1, Carry_{out} = 1$

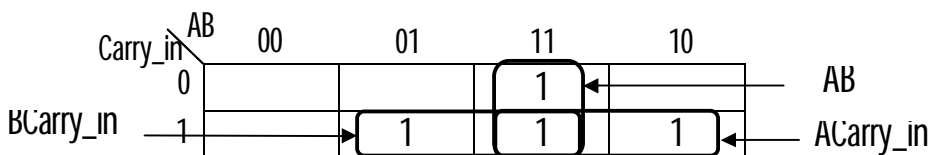
ตารางที่ 6.2 ตารางความจริงของวงจรวกแบบคิดตัวทด

| อินพุต | | | เอาต์พุต | |
|--------|---|----------|----------|-----------|
| A | B | Carry_in | 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 |

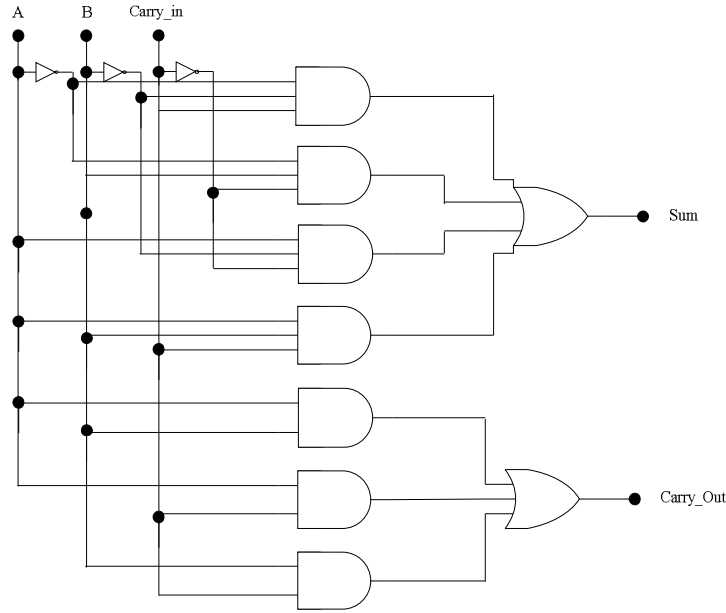
จากตารางความจริงสามารถคำนวณคำนวณหาสมการพีชคณิตบูลีนของ Sum และ Carry_out ได้ดังนี้



$$Sum = \overline{A}B\overline{Carry_in} + A\overline{B}\overline{Carry_in} + \overline{A}B\overline{Carry_in} + \overline{A}B\overline{Carry_in}$$



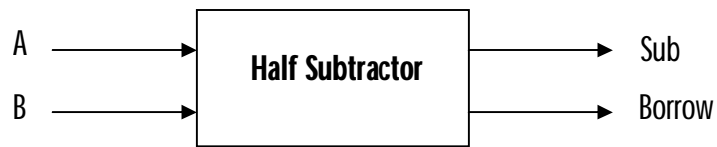
$$Carry_{out} = \overline{A}B\overline{Carry_in} + A\overline{B}\overline{Carry_in} + \overline{A}B\overline{Carry_in}$$



รูปที่ 6.8 วงจรบวกแบบคิดตัวทด

6.5.3 วงจรลบแบบไม่คิดตัวยืม

วงจรถลบแบบไม่คิดตัวยืมคือ วงจรที่เกิดจากการนำบิตที่มีลำดับความสำคัญต่ำที่สุดมาผ่านการดำเนินการลบกันซึ่งผลลัพธ์ที่ได้จะมี 2 ค่าคือผลลบ และตัวยืมจากบิตถัดไป



รูปที่ 6.9 โครงสร้างวงจรถลบแบบไม่คิดตัวยืม

รูปที่ 6.9 แสดงโครงสร้างวงจรถลบแบบไม่คิดตัวยืม การทำงานของวงจรคือจะนำอินพุต A ซึ่งมีขนาด 1 บิตลบออกด้วย อินพุต B ซึ่งมีขนาด 1 บิต โดยเอาต์พุตที่ได้จะมี 2 ค่าคือผลลบ (Sub) และตัวยืม (Borrow) เนื่องจากอินพุตมี 2 ตัว ดังนั้นค่าที่เป็นไปได้ทั้งหมดมี 4 กรณี ดังนี้

กรณีที่ 1: $A = 0$ และ $B = 0$

$$\begin{array}{r}
 \text{Borrow} \\
 \downarrow \\
 0 \quad (A) \\
 - \\
 0 \quad (B) \\
 \hline
 0 \quad (\text{Sub})
 \end{array}$$

ผลลัพธ์ที่ได้: $\text{Sub} = 0, \text{Borrow} = 0$

กรณีที่ 2: $A = 0$ และ $B = 1$

| | | |
|--------|----------|-------|
| Borrow | | |
| | ↓ | |
| | 0 | (A) |
| | | - |
| | 1 | (B) |
| | <u>1</u> | (Sub) |

ผลลัพธ์ที่ได้: Sub = 1, Borrow = 1

จากผลลัพธ์ที่ได้ในกรณีที่ 2 อธิบายได้ดังนี้ เนื่องจากตัวตั้ง (A) มีค่าน้อยกว่าตัวลบ (B) ซึ่งลบกันไม่ได้ A จึงต้องไปขอยืมค่าจากบิตติดกันที่มีลำดับความสำคัญสูงกว่าทำให้ Borrow มีค่าเป็น 1 และเนื่องจากการลบกันแบบเลขฐานสอง ค่าที่ A ยืมมาได้จึงมีค่าเป็น 2 ดังนั้นผลลัพธ์ของ $A - B$ (Sub) คือ $10_2 - 1_2 = 1$

กรณีที่ 3: $A = 1$ และ $B = 0$

| | | |
|--------|----------|-------|
| Borrow | | |
| | ↓ | |
| | 0 | (A) |
| | | - |
| | 0 | (B) |
| | <u>1</u> | (Sub) |

ผลลัพธ์ที่ได้: Sub = 1, Borrow = 0

กรณีที่ 4: $A = 1$ และ $B = 1$

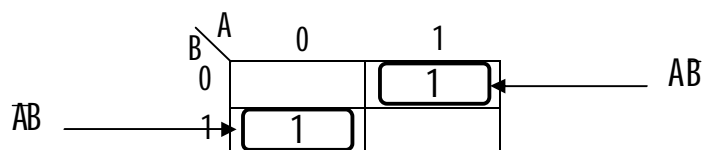
| | | |
|--------|----------|-------|
| Borrow | | |
| | ↓ | |
| | 0 | (A) |
| | | - |
| | 1 | (B) |
| | <u>0</u> | (Sub) |

ผลลัพธ์ที่ได้: Sum = 0, Carry = 0

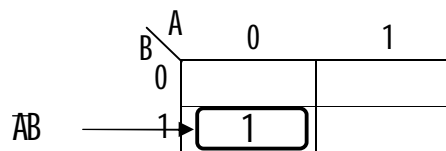
ตารางที่ 6.3 ตารางความจริงของวงจรถบแบบไม่คิดตัวยืม

| อินพุต | | เอาต์พุต | |
|--------|---|----------|--------|
| A | B | Sub | Borrow |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

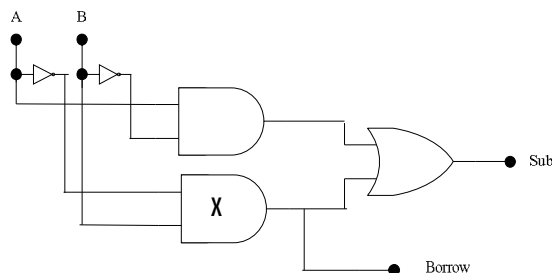
จากตารางความจริงสามารถคำนวณคำนวณหาสมการพีชคณิตบูลีนของ Sub และ Borrow ได้ ดังนี้



Sum = AB + AB



Carry = AB

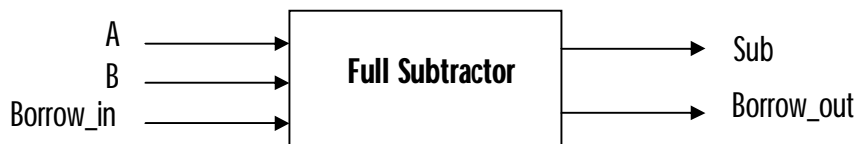


รูปที่ 6.10 วงจรถบแบบไม่คิดตัวยืม

จากรูป 6.10 สังเกตได้ว่าไม่จำเป็นต้องสร้างวงจรถบ Borrow ขึ้นมาใหม่ เนื่องจากผลลัพธ์ของเกต X มีค่าเป็น AB ซึ่งตรงกับค่าของ Borrow ดังนั้นจึงสามารถใช้ร่วมกันได้ ทำให้ประหยัดการใช้งานเกตไปได้ 1 ตัว

6.5.4 วงจรลบแบบคิดตัวยืม

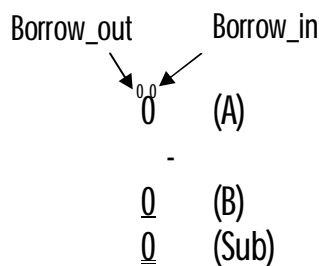
วงจรลบแบบคิดตัวยืมคือ วงจรที่เกิดจากการนำอินพุตขนาด 1 บิต 2 ตัวมาลบกัน แต่อินพุตทั้ง 2 ตัวจะต้องเป็นบิตที่มีลำดับความสำคัญที่เท่ากันและเป็นบิตที่ไม่ใช่ตำแหน่งบิตที่มีลำดับความสำคัญต่ำที่สุด ดังนั้นการลบจึงมีความเป็นไปได้ที่บิตของตัวตั้งจะถูกยืม (Borrow_in) โดยบิตติดกันที่มีลำดับความสำคัญที่ต่ำกว่า เนื่องจากบิตตัวตั้งมีค่าน้อยกว่าบิตตัวลบ อธิบายได้ดังนี้สมมติว่าต้องการลบเลขขนาด 2 บิตคือ $A_1A_0 - B_1B_0$ โดยบิตที่เราสนใจคือการลบระหว่าง A_1 และ B_1 เท่านั้น แต่เนื่องจาก A_1 และ B_1 เป็นบิตที่ไม่ใช่บิตที่มีลำดับความสำคัญต่ำที่สุด ดังนั้นจึงมีความเป็นไปได้ที่จะ A_1 อาจจะถูกยืมจาก A_0 เนื่องจาก A_0 มีค่าน้อยกว่า B_0 ซึ่งลบโดยตรงไม่ได้จึงต้องขอยืมจาก A_1 ซึ่งผลลัพธ์ที่ได้จะมี 2 ค่าคือผลลบ และตัวยืมในบิตถัดไป



รูปที่ 6.11 โครงสร้างวงจรถลบแบบคิดตัวยืม

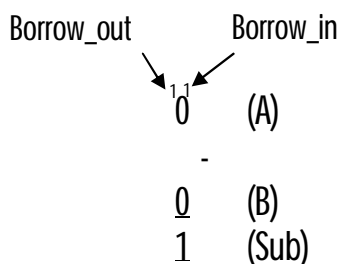
รูปที่ 6.11 แสดงโครงสร้างวงจรถลบแบบคิดตัวยืม การทำงานของวงจรถคือจะนำอินพุต A_1 ซึ่งมีขนาด 1 บิตมาลบออกด้วย B_1 ซึ่งมีขนาด 1 บิตเช่นกัน และหาก Borrow_in ซึ่งมีขนาด 1 บิตมีค่าเป็น 1 จะต้องนำ Borrow_in มาลบออกด้วย โดยเอาต์พุตที่ได้จะมี 2 ค่าคือผลลบ (Sub) และตัวยืม (Borrow_out) เนื่องจากอินพุตมี 3 ตัว ดังนั้นค่าที่เป็นไปได้ทั้งหมดมี 8 กรณี ดังนี้

กรณีที่ 1: $A = 0, B = 0$ และ $Borrow_in = 0$



ผลลัพธ์ที่ได้: $Sub = 0, Borrow_out = 0$

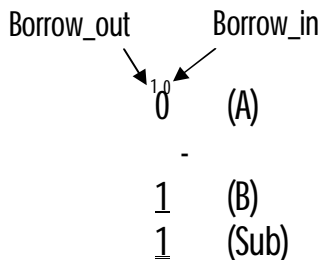
กรณีที่ 2: $A = 0, B = 0$ และ $Borrow_in = 1$



ผลลัพธ์ที่ได้: Sub = 1, Borrow_out = 1

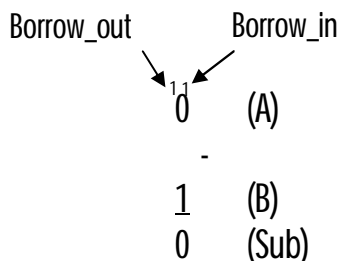
จากผลลัพธ์ที่ได้ในกรณีที่ 2 อธิบายได้ดังนี้ เนื่องจากตัวตั้ง (A) ถูกยืมจากบิตที่มีลำดับความสำคัญต่ำกว่าไป 1 ค่า (Borrow_in = 1) ทำให้ A มีค่าน้อยกว่าตัวลบ (B) ซึ่งลบกันไม่ได้ A จึงต้องไปขอยืมค่าจากบิตติดกันที่มีลำดับความสำคัญสูงกว่าทำให้ Borrow_out มีค่าเป็น 1 และเนื่องจากการลบกันแบบเลขฐานสอง ค่าที่ A ยืมมาได้จึงมีค่าเป็น 2 แต่โดนหักจากบิตที่มีลำดับความสำคัญต่ำกว่าที่ขอยืมไป 1 ค่าดังนั้นผลลัพธ์ของ A - B (Sub) คือ $1 - 0 = 1$

กรณีที่ 3: A = 0, B = 1 และ Borrow_in = 0



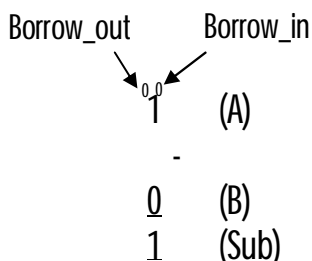
ผลลัพธ์ที่ได้: Sub = 1, Borrow_out = 1

กรณีที่ 4: A = 0, B = 1 และ Borrow_in = 1



ผลลัพธ์ที่ได้: Sub = 0, Borrow_out = 1

กรณีที่ 5: A = 1, B = 0 และ Borrow_in = 0



ผลลัพธ์ที่ได้: Sub = 1, Borrow_out = 0

กรณีที่ **6**: $A = 1, B = 0$ และ $Borrow_in = 1$

$$\begin{array}{r}
 \text{Borrow_out} \quad \text{Borrow_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 01 \\
 \underline{1} \quad (A) \\
 - \\
 \underline{0} \quad (B) \\
 \underline{0} \quad (Sub)
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $Sub = 0, Borrow_out = 0$

กรณีที่ **7**: $A = 1, B = 1$ และ $Borrow_in = 0$

$$\begin{array}{r}
 \text{Borrow_out} \quad \text{Borrow_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 00 \\
 \underline{1} \quad (A) \\
 - \\
 \underline{1} \quad (B) \\
 \underline{0} \quad (Sub)
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $Sub = 0, Borrow_out = 0$

กรณีที่ **8**: $A = 1, B = 1$ และ $Borrow_in = 1$

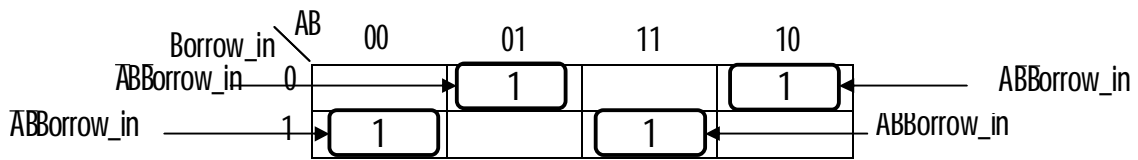
$$\begin{array}{r}
 \text{Borrow_out} \quad \text{Borrow_in} \\
 \swarrow \quad \searrow \\
 \begin{array}{r}
 11 \\
 \underline{1} \quad (A) \\
 - \\
 \underline{1} \quad (B) \\
 \underline{1} \quad (Sub)
 \end{array}
 \end{array}$$

ผลลัพธ์ที่ได้: $Sub = 1, Borrow_out = 1$

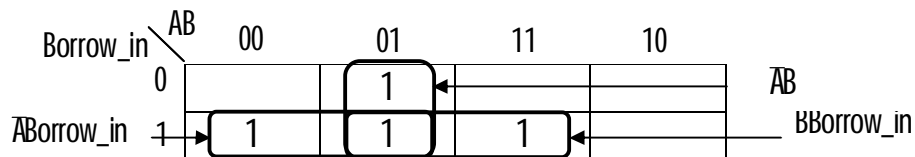
ตารางที่ 6.4 ตารางความจริงของวงจรถลแบบคิดด้วยืม

| อินพุต | | | เอาต์พุต | |
|--------|---|-----------|----------|------------|
| A | B | Borrow_in | Sub | Borrow_out |
| 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 |

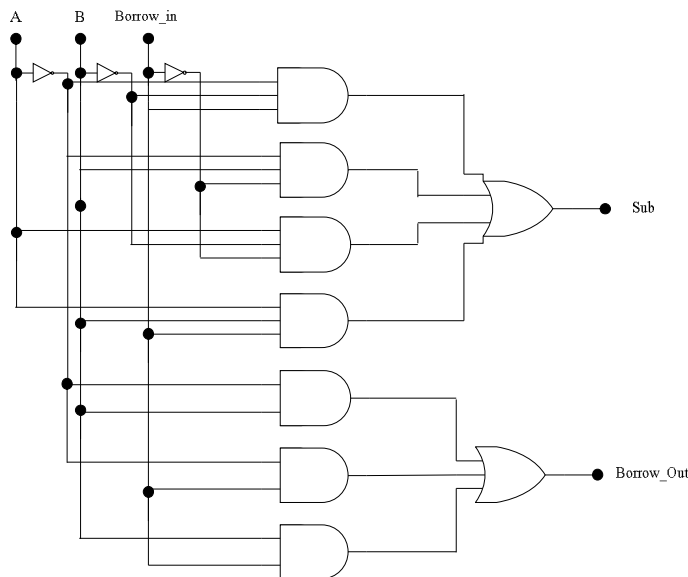
จากตารางความจริงสามารถคำนวณหาสมการพีชคณิตบูลีนของ Sub และ Borrow_out ได้ดังนี้



$$\text{Sub} = \bar{A}B\text{Borrow_in} + \bar{A}B\bar{\text{Borrow_in}} + A\bar{B} + AB\text{Borrow_in}$$



$$\text{Borrow_out} = \bar{A}B\text{Borrow_in} + A\bar{B} + A\bar{B}\text{Borrow_in}$$



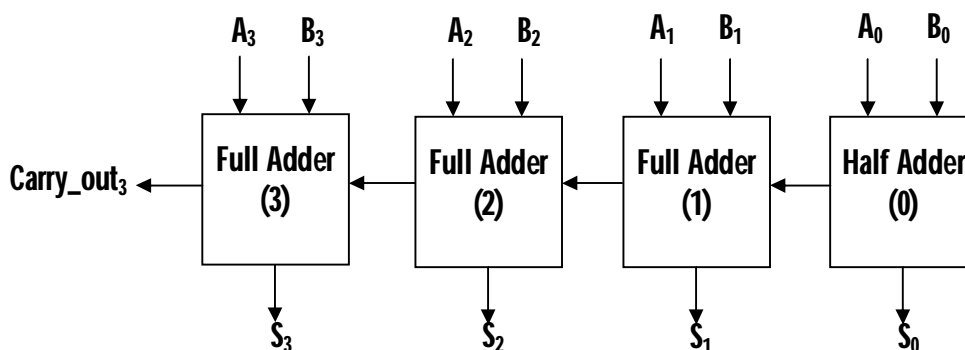
รูปที่ 6.12 วงจรลบแบบคิดตัวยืม

6.5.5 วงจรบวกแบบหลายบิต

วงจรวกเลขฐานสองแบบคิดตัวทด และแบบไม่คิดตัวทอนั้นใช้สำหรับคำนวณหาผลบวกเลขฐานสองของอินพุต 2 ตัวที่มีขนาดเพียง 1 บิตเท่านั้น โดยวงจรวกแบบไม่คิดตัวทอนั้นจะใช้สำหรับการบวกเลขที่ตำแหน่งบิตที่มีลำดับความสำคัญต่ำที่สุด ส่วนวงจรวกแบบคิดตัวทดจะเป็นวงจรที่ใช้สำหรับการบวกเลขที่ตำแหน่งบิตที่ไม่ได้อยู่ในตำแหน่งที่มีลำดับความสำคัญต่ำที่สุด ดังนั้นกำหนดให้จำนวนบิตของค่าที่จะนำมาบวกกันมีค่าเท่ากัน หากต้องการออกแบบวงจรวกเลขฐานสองที่มีขนาดมากกว่า 1 บิตต้องใช้จำนวนวงจรวกเลขฐานสองตามจำนวนบิตของอินพุตทั้งหมด เช่นหากอินพุตตัวที่มีขนาดมากที่สุดคือ 4 บิต ต้องใช้วงจรวกเลขฐานสองทั้งหมด 4 ตัว ประกอบไปด้วยวงจรวกแบบไม่คิดตัวทด 1 ตัว และวงจรวกแบบคิดตัวทอนอีก 3 ตัว ดังตัวอย่างที่ 6-11

ตัวอย่างที่ 6-11 การออกแบบวงจรวกเลขฐานสองขนาด 4 บิต

วิธีทำ การออกแบบวงจรวกเลขฐานสองขนาด 4 บิต โดยบิตตัวทอนออกของแต่ละวงจรวกจะเชื่อมกับตัวทอนเข้าของ วงจรวกที่มีลำดับความสำคัญที่สูงกว่า ดังนี้



ทดสอบการทำงาน กำหนดให้ $A = 1011$ และ $B = 0110$ การคำนวณหา $S = A+B$ จะถูกแบ่งออกเป็น 4 ส่วนซึ่งแต่ละส่วนมีการคำนวณเป็นดังนี้

ส่วนที่ 1: $A_0 = 1$ และ $B_0 = 0$ ดังนั้น $S_0 = A_0 + B_0 = 1 + 0 = 1$

ส่วนที่ 2: $A_1 = 1$ และ $B_1 = 1$ ดังนั้น $S_1 = A_1 + B_1 = 1 + 1 = 0$ แต่มีตัวทดออกไปยังบิตที่ 2

ส่วนที่ 3: $A_2 = 0$ และ $B_2 = 1$ และมีตัวทดเข้า (Carry_{in2} = 1) ดังนั้น $S_2 = A_2 + B_2 + Carry_{in2} = 0 + 1 + 1 = 0$ แต่มีตัวทดออกไปยังบิตที่ 3

ส่วนที่ 4: $A_3 = 1$ และ $B_3 = 0$ และมีตัวทดเข้า (Carry_{in3} = 1) ดังนั้น $S_3 = A_3 + B_3 + Carry_{in3} = 1 + 0 + 1 = 0$ แต่มีตัวทดออกคือ Carry_{out3} = 1

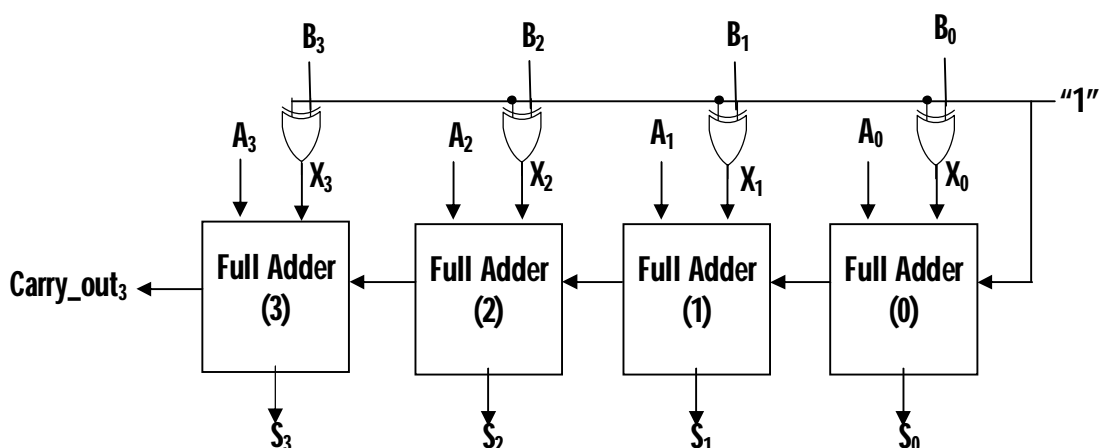
ดังนั้น $S = S_3S_2S_1S_0 = 0001$ และมีตัวทดออก

6.5.6 วงจรลบแบบหลายบิต

การออกแบบวงจรลบแบบหลายบิต สามารถทำได้โดยการนำวงจรลบแบบคิดตัวยืม และวงจรลบแบบไม่คิดตัวยืมมาต่อรวมกัน กำหนดให้จำนวนบิตของค่าที่จะนำมาลบกันมีค่าเท่ากัน หากต้องการออกแบบวงจรลบเลขฐานสองที่มีขนาดมากกว่า 1 บิตต้องใช้จำนวนวงจรลบเลขฐานสองตามจำนวนบิตของอินพุตทั้งหมด อย่างไรก็ตามสามารถนำวงจรบวกเลขฐานสองแบบคิดตัวทดมาออกแบบเป็นวงจรลบเลขฐานสองได้โดยการนำบิตตัวลบทั้งหมดมาผ่านตัวดำเนินการเอ็กออร์กับค่า "1" และกำหนดให้ตัวทดเข้าของวงจรบวกเลขฐานสองของตำแหน่งบิตที่มีลำดับความสำคัญต่ำที่สุดมีค่าเป็น "1" ข้อดีของการนำวงจรบวกเลขฐานสองมาออกแบบแทนวงจรลบเลขฐานสองคือวงจรชนิดเดียวสามารถสร้างได้ทั้งวงจรบวก และวงจรลบเลขฐานสองซึ่งจะช่วยให้จำนวนเกตที่จะนำมาใช้งานลดลง

ตัวอย่างที่ 6-12 การออกแบบวงจรลบเลขฐานสองขนาด 4 บิต

วิธีทำ การออกแบบวงจรลบเลขฐานสองขนาด 4 บิต โดยการนำวงจรบวกเลขฐานสองแทนทั้งหมดสามารถทำได้ดังนี้



ทดสอบการทำงาน กำหนดให้ $A = 1011$ และ $B = 0110$ การคำนวณหา $S = A - B$ จะถูกแบ่งออกเป็น 4 ส่วนซึ่งแต่ละส่วนมีการคำนวณเป็นดังนี้

ส่วนที่ 1: $A_0 = 1$ และ $B_0 = 0$, ได้ $X_0 = B_0 \oplus 1 = 0 \oplus 1 = 1$

ดังนั้น $S_0 = A_0 + X_0 + \text{Carry_in}_0 = 1 + 1 + 1 = 1$ และมีตัวทดออกไปยังบิตที่ 1

ส่วนที่ 2: $A_1 = 1$ และ $B_1 = 1$, ได้ $X_1 = B_1 \oplus 1 = 1 \oplus 1 = 0$ และมีตัวทดเข้าจากบิตที่ 0

ดังนั้น $S_1 = A_1 + X_1 + \text{Carry_in}_1 = 1 + 0 + 1 = 0$ และมีตัวทดออกไปยังบิตที่ 2

ส่วนที่ 3: $A_2 = 0$ และ $B_2 = 1$, ได้ $X_2 = B_2 \oplus 1 = 1 \oplus 1 = 0$ และมีตัวทดเข้าจากบิตที่ 1

ดังนั้น $S_2 = A_2 + X_2 + \text{Carry_in}_2 = 0 + 0 + 1 = 1$ และไม่มีตัวทดออกไปยังบิตที่ 3

ส่วนที่ 4: $A_3 = 1$ และ $B_3 = 0$, ได้ $X_3 = B_3 \oplus 1 = 0 \oplus 1 = 1$ เนื่องจากไม่มีตัวทดเข้าจากบิตที่

2

ดังนั้น $S_3 = A_3 + X_3 + \text{Carry_in}_3 = 1 + 1 + 0 = 0$ และมีตัวทดออกคือ $\text{Carry_out}_3 = 1$

ดังนั้น $S = S_3S_2S_1S_0 = 0101$ และมีตัวทดออกโดยตัวทดออกเปรียบเสมือนเป็นตัวยืมจาก

บิตถัดไป

6.6 บทสรุป

นอกจากการใช้กฎบูลีนแล้ว แผนผังคาร์โนห์ (Karnaugh Map) เป็นวิธีการลดรูปสมการบูลีนอีกวิธีหนึ่งซึ่งเป็นวิธีที่ง่าย และมีข้อผิดพลาดที่น้อยกว่าหากเปรียบเทียบกับการใช้กฎบูลีนสำหรับการลดรูปสมการ โดยแผนผังคาร์โนห์จะใช้ตารางสี่เหลี่ยมที่มีจำนวนช่องเท่ากับ $2^{\text{จำนวนตัวแปร}}$ โดยการยุบรวมนั้นจะต้องเลือกช่องที่มีค่าเป็น 1 ทั้งหมดเพียงอย่างเดียว หรือเลือกช่องที่มีค่าเป็น 0 ทั้งหมดเพียงอย่างเดียว และทำการยุบรวมช่องที่อยู่ติดกันโดยแต่ละกลุ่มที่ถูกยุบรวมนั้นจะมีจำนวนช่องอยู่ในรูปของ 2^n เท่านั้น เมื่อ n คือจำนวนเต็มบวกใดๆ โดยกรณีที่เลือกช่องที่มีค่าเป็น 1 ทั้งหมดให้เขียนสมการบูลีนของแต่ละกลุ่มโดยใช้ตัวดำเนินการแอนด์เพื่อเชื่อมแต่ละตัวแปร และนำแต่ละกลุ่มมาเชื่อมกันด้วยตัวดำเนินการออร์ แต่หากเป็นกรณีที่เลือกช่องที่มีค่าเป็น 0 ทั้งหมด ให้เขียนสมการบูลีนของแต่ละกลุ่มโดยใช้ตัวดำเนินการออร์เพื่อเชื่อมแต่ละตัวแปร และนำแต่ละกลุ่มมาเชื่อมกันด้วยตัวดำเนินการแอนด์ อย่างไรก็ตามแผนผังคาร์โนห์จะเหมาะสมกับการใช้ลดรูปสมการที่มีตัวแปรอยู่ระหว่าง 2 - 4 ตัวเท่านั้น สถานะที่ไม่สนใจ (Don't Care Condition) คือสถานะที่มีค่าเป็นได้ทั้ง 0 หรือ 1 ซึ่งสามารถช่วยลดรูปให้วงจรมีขนาดเล็กลงได้ และสุดท้ายในบทนี้ได้กล่าวถึงการออกแบบวงจรบวก และลบเลขฐานสองโดยใช้แผนผังคาร์โนห์สำหรับการลดรูปวงจร โดยวงจรบวก และลบเลขฐานสองเป็นวงจรที่สามารถนำมาใช้สำหรับการคำนวณด้านการบวก และการลบระหว่าง 2 ตัวแปรที่มีขนาด 1 บิต แต่หากต้องการให้มีการบวก หรือลบระหว่างตัวแปรที่มีขนาดมากกว่า 1 บิตสามารถทำได้โดยนำวงจรดังกล่าวมาต่อเชื่อมกันตามจำนวนบิตที่ต้องการ

คำถามท้ายบท

1. จงลดรูปสมการต่อไปนี้โดยใช้แผนผังคาร์โนห์
 - 1.1 $f(A, B, C) = \sum m(0, 1, 4, 6)$
 - 1.2 $f(A, B, C) = \sum m(1, 3, 4, 5, 7)$
 - 1.3 $f(A, B, C, D) = \sum m(0, 1, 4, 5, 6, 9, 10, 13)$
 - 1.4 $f(A, B, C) = \prod M(1, 2, 4, 7)$
 - 1.5 $f(A, B, C) = \prod M(0, 3, 5, 7)$
 - 1.6 $f(A, B, C) = \sum m(0, 4)$
 - 1.7 $f(A, B, C, D) = \sum m(1, 2, 5, 11) + d(3, 4, 8, 9)$
 - 1.8 $f(A, B, C, D) = \sum m(0, 2, 12, 15)$
 - 1.9 $f(A, B, C) = \prod M(2, 3, 5) + d(1, 6)$
 - 1.10 $f(A, B, C, D) = \prod M(1, 3, 10, 12) + d(2, 6, 11, 15)$

2. จากตารางความจริงต่อไปนี้จงหาสมการบูลีนที่อยู่ในรูปที่ง่ายที่สุด และอยู่ในรูปของผลรวมของผลคูณ
 - 2.1)

| อินพุต | | เอาต์พุต (Z) |
|--------|---|-----------------|
| A | B | |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

2.2)

| อินพุต | | เอาต์พุต (Z) |
|--------|---|-----------------|
| A | B | |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

2.3)

| อินพุต | | | เอาต์พุต (Z) |
|--------|---|---|-----------------|
| A | B | C | |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

2.4)

| อินพุต | | | เอาต์พุต (Z) |
|--------|---|---|-----------------|
| A | B | C | |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

3. กำหนดให้ A, B, C และ D แทนเลขฐานสองที่เรียงกันตามลำดับความสำคัญจากมากไปหาน้อย โดย A เป็นบิตที่มีลำดับความสำคัญมากที่สุดและ D เป็นบิตที่มีลำดับความสำคัญน้อยที่สุด จงเขียนสมการพีชคณิตบูลีนที่ทำให้เอาต์พุตมีค่าเป็น 1 ก็ต่อเมื่อ สถานะของอินพุตซึ่งมีการใช้งานเพียงแค่ 8 ค่าคือ 0000, 0001, ..., 1000 มีค่าเป็น 1 พร้อมกันอย่างน้อย 2 ค่า

4. จงออกแบบวงจรรวมเลขฐานสองระหว่าง 2 ตัวแปรโดยที่ตัวแปรตัวหนึ่งมีขนาด 5 บิต และตัวแปรอีกตัวหนึ่งมีขนาด 3 บิต

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

แผนบริหารการสอนประจำบทที่ 7 ฟลิปฟล็อป 3 ชั่วโมง

หัวข้อเนื้อหา

7.1 ฟลิปฟล็อปแบบ RS

7.1.1 ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ RS

7.1.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ RS

7.2 ฟลิปฟล็อปแบบ D

7.2.1 ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ D

7.2.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D

7.3 ฟลิปฟล็อปแบบ T

7.3.1 ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ T

7.3.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ T

7.4 ฟลิปฟล็อปแบบ JK

7.4.1 ตารางเอ็กไซเทชั่นของฟลิปฟล็อปแบบ JK

7.4.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ JK

7.5 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับนิยาม และคุณสมบัติของฟลิปฟล็อปชนิดต่างๆ
2. เพื่อให้ผู้เรียนสามารถคำนวณหาสมการพีชคณิตบูลีนสถานะเอาต์พุตใหม่ของฟลิปฟล็อปชนิดต่างๆ ได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

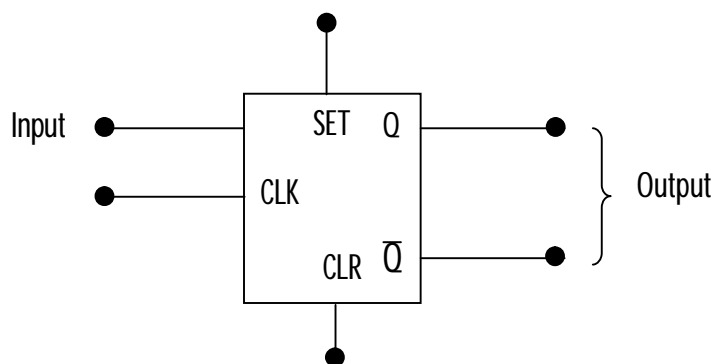
การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 7

ฟลิปฟล็อป

ฟลิปฟล็อป เป็นอุปกรณ์ทางอิเล็กทรอนิกส์ที่มีลักษณะเป็นแบบมีหน่วยความจำขนาด 1 บิต ซึ่งสามารถสร้างได้โดยใช้เกตพื้นฐานต่างๆ โดยเฉพาะอย่างยิ่ง แนนด์เกต หรือ นอร์เกตโดยที่เอาต์พุตของฟลิปฟล็อปจะคงสถานะเดิมไว้ ถึงแม้ว่าจะไม่มีสถานะของอินพุตเกิดขึ้นเลยก็ตาม โดยหากต้องการเปลี่ยนสถานะของเอาต์พุตจะต้องทำการเปลี่ยนสัญญาณอินพุต พร้อมกับการเกิดสัญญาณนาฬิกา หรือเกิดสัญญาณเซต หรือ สัญญาณเคลียร์สัญญาณใดสัญญาณหนึ่งเท่านั้นโดยวงจรที่เกิดจากการนำฟลิปฟล็อปมาใช้งาน ซึ่งอาจจะมีหรือไม่มีเกตพื้นฐานต่อใช้งานร่วมด้วย เรียกว่า วงจรเชิงลำดับ (Sequential Logic)



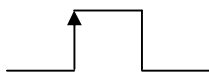
รูปที่ 7.1 โครงสร้างฟลิปฟล็อป

รูปที่ 7.1 แสดงโครงสร้างของฟลิปฟล็อปซึ่งมีสายสัญญาณที่สำคัญทั้งหมด 5 ตัวประกอบด้วย สัญญาณอินพุต สัญญาณเอาต์พุต สัญญาณนาฬิกา (CLK) สัญญาณเคลียร์ค่า (CLR) และ สัญญาณเซตค่า (SET)

สัญญาณอินพุตของฟลิปฟล็อปจะถูกแบ่งออกเป็น 2 ชนิดคือ ฟลิปฟล็อปที่มีอินพุต 1 ค่ามี 2 ชนิดคือ ฟลิปฟล็อปแบบ D และ ฟลิปฟล็อปแบบ T และ ฟลิปฟล็อปที่มีอินพุต 2 ค่ามี 2 ชนิดคือ ฟลิปฟล็อปแบบ RS และ ฟลิปฟล็อปแบบ JK

สัญญาณเอาต์พุตของฟลิปฟล็อปจะมี 2 ค่าคือ Q และ \bar{Q} สังเกตว่าทั้ง 2 ค่านี้เป็นนิเสธซึ่งกันและกันเพราะฉะนั้นหาก Q มีค่าเป็น 1 แล้ว \bar{Q} จะต้องเป็น 0 เท่านั้น ในทางกลับกันหาก Q มีค่าเป็น 0 แล้ว \bar{Q} จะต้องเป็น 1 เท่านั้น

สัญญาณ CLK คือสัญญาณที่ทำให้มีการเปลี่ยนสถานะที่สัญญาณเอาต์พุต โดยหากไม่มีสัญญาณ CLK เกิดขึ้น สัญญาณเอาต์พุตจะไม่มีเปลี่ยนแปลง (หากไม่เกิดสัญญาณ SET หรือ CLR) ถึงแม้ว่าจะมีการเปลี่ยนสถานะของสัญญาณอินพุตก็ตาม โดยสัญญาณ CLK สามารถเกิดได้ 2 ตำแหน่งคือ ตำแหน่งที่เกิดจากการเปลี่ยนสถานะจาก "0" เป็น "1" เรียกว่าเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และตำแหน่งที่เกิดจากการเปลี่ยนสถานะจากสถานะ "1" เป็น "0" เรียกว่าเกิดสัญญาณนาฬิกาที่ขอบขาลง

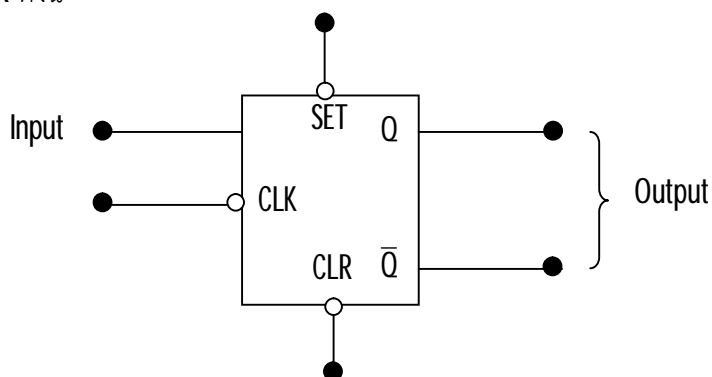


รูปที่ 7.2 การเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น



รูปที่ 7.3 การเกิดสัญญาณนาฬิกาที่ขอบขาลง

สำหรับกรณีที่ใช้ฟลิปฟล็อปที่มีการเปลี่ยนสถานะของเอาต์พุตที่ขอบขาลง จะมีสัญลักษณ์ "0" ที่หน้าสัญญาณ CLK ดังนี้



รูปที่ 7.4 โครงสร้างฟลิปฟล็อปที่มีการเปลี่ยนสถานะเอาต์พุตที่ขอบขาลง

สัญญาณ CLR คือสัญญาณที่ทำให้สถานะของเอาต์พุตมีค่าเป็น 0 ทันทีที่ได้รับสัญญาณดังกล่าว โดยที่ไม่สนใจว่าในขณะนั้นมีสัญญาณ CLK เกิดขึ้นหรือไม่ เนื่องจากสัญญาณ CLR มีลำดับความสำคัญที่สูงกว่าสัญญาณ CLK

สัญญาณ SET คือ สัญญาณที่ทำให้สถานะของเอาต์พุตมีค่าเป็น 1 ทันทีที่ได้รับสัญญาณดังกล่าว โดยที่ไม่สนใจว่าในขณะนั้นมีสัญญาณ CLK เกิดขึ้นหรือไม่ เนื่องจากสัญญาณ SET มีลำดับความสำคัญที่สูงกว่าสัญญาณ CLK

จากผลลัพธ์ข้างต้นของสัญญาณ CLR และสัญญาณ SET สรุปได้ว่าไม่สามารถที่จะทำให้เกิดสัญญาณ CLR และ สัญญาณ SET พร้อมกันได้ เนื่องจากสถานะของเอาต์พุตมีค่าออกมาตรงกันข้ามกัน โดยเงื่อนไขที่เป็นไปได้ทั้งหมดของการเกิดสัญญาณทั้ง 2 ค่าเป็นดังนี้ (กำหนดให้สถานะ "1" คือการส่งสัญญาณ และสถานะ "0" คือยังไม่มีสัญญาณ)

กรณีที่ 1 (CLR = 0, SET = 0): ยังไม่มีการส่งสัญญาณ CLR และสัญญาณ SET ดังนั้นเอาต์พุตที่ได้จะขึ้นกับสัญญาณ CLK และ อินพุต

กรณีที่ 2 (CLR = 0, SET = 1): มีการส่งสัญญาณ SET ดังนั้นเอาต์พุตที่ได้จะมีค่าเป็น 1 ทันทีที่ได้รับสัญญาณ SET

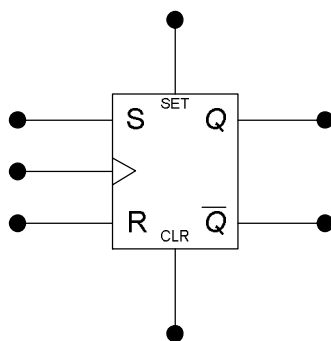
กรณีที่ 3 ($CLR = 1, SET = 0$): มีการส่งสัญญาณ CLR ดังนั้นเอาต์พุตที่ได้จะมีค่าเป็น 0 ทันทีที่ได้รับสัญญาณ CLR

กรณีที่ 4 ($CLR = 1, SET = 1$): มีการส่งสัญญาณ CLR และ สัญญาณ SET พร้อมกัน แต่เนื่องจากพบว่าเอาต์พุตที่ได้ขัดแย้งซึ่งกันและกันดังนั้น เงื่อนไขนี้จึงไม่ให้เกิดขึ้น

หมายเหตุ หากมีสัญลักษณ์ "0" อยู่ที่ตัวสัญญาณ ความหมายคือ "0" คือการส่งสัญญาณ และสถานะ "1" คือยังไม่มีสัญญาณ

7.1 ฟลิปฟลอปแบบ RS (RS FlipFlop)

ฟลิปฟลอปแบบ RS คือฟลิปฟลอปที่มี 2 อินพุตคือ R และ S โดยสามารถสร้างได้จากการใช้นอร์เกต (หรือแอนด์เกต) จำนวน 2 ตัว สำหรับหัวข้อนี้จะกล่าวเฉพาะการสร้างฟลิปฟลอปแบบ RS โดยใช้ นอร์เกต เท่านั้น



รูปที่ 7.5 สัญลักษณ์ของฟลิปฟลอปแบบ RS

ตารางที่ 7.1 ตารางความจริงของฟลิปฟลอปแบบ RS

| อินพุต | | เอาต์พุต | | สถานะ |
|--------|---|------------|------------------|-----------------|
| R | S | Q_{next} | \bar{Q}_{next} | |
| 0 | 0 | Q | \bar{Q} | ไม่เปลี่ยนสถานะ |
| 0 | 1 | 1 | 0 | เซต |
| 1 | 0 | 0 | 1 | เคลียร์ |
| 1 | 1 | - | - | ไม่เกิดขึ้น |

ตารางที่ 7.1 คือตารางความจริงของฟลิปฟลอปแบบ RS กำหนดให้ Q_{next} คือสถานะของค่า Q ที่เกิดขึ้นมาใหม่หลังจากเกิดสัญญาณ CLK ซึ่งสามารถอธิบายเอาต์พุตที่เกิดขึ้นได้ดังนี้

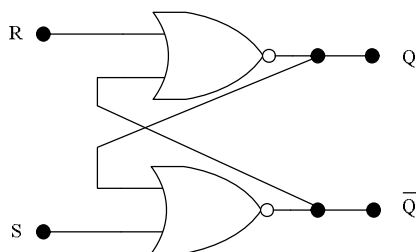
กรณีที่ 1 $R = 0$ และ $S = 0$: ค่า Q_{next} จะมีค่าคงค่าเดิม ความหมายคือ ค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีสถานะเป็นอย่างไร ค่า Q ใหม่ที่เกิดขึ้นหลังเกิดสัญญาณ CLK จะยังคงเป็นค่าเดิม

กรณีที่ 2 $R = 0$ และ $S = 1$: ค่า Q_{next} จะมีค่าเป็น 1 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 3 $R = 1$ และ $S = 0$: ค่า Q_{next} จะมีค่าเป็น 0 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 4 $R = 1$ และ $S = 1$: เป็นสถานะที่ไม่ยอมให้เกิดขึ้น เนื่องจากทำให้สถานะของ Q และ \bar{Q} มีค่าเหมือนกันซึ่งขัดแย้งกับความเป็นจริงที่ว่า Q และ \bar{Q} จะต้องเป็นนิเสธซึ่งกันและกันเสมอซึ่งจะแสดงให้เห็นต่อไปในส่วนของโครงสร้างที่อยู่ภายในของฟลิปฟล็อปแบบ RS

ค่าเอาต์พุต Q และ \bar{Q} มาจากโครงสร้างภายในของฟลิปฟล็อปแบบ RS ซึ่งสามารถพิจารณาได้ดังรูปที่ 7.6



รูปที่ 7.6 โครงสร้างของฟลิปฟล็อปแบบ RS

รูปที่ 7.6 แสดงโครงสร้างของฟลิปฟล็อปแบบ RS ซึ่งเกิดจากการใช้ NOR เกต 2 ตัวโดยการนำเอาต์พุต Q มาเป็นอินพุตใช้ NOR เกตตัวที่ 2 และนำเอาเอาต์พุต \bar{Q} มาเป็นอินพุตให้ NOR เกตตัวที่ 1 เรียกการต่อวงจรลักษณะนี้ว่า "แลตซ์" ซึ่งเขียนสมการพีชคณิตบูลีนของ Q และ \bar{Q} ได้ดังนี้

$$Q = R + \bar{Q}$$

$$\bar{Q} = S + Q$$

ดังนั้นสถานะของ Q และ \bar{Q} เนื่องจาก R และ S เป็นดังนี้

กรณีที่ 1: $R = 0$ และ $S = 0$

$$Q = 0 + \bar{Q} = \bar{Q} = Q$$

$$\bar{Q} = 0 + Q = Q$$

} คงค่าเดิม

กรณีที่ 2: $R = 0$ และ $S = 1$

$$Q = 1 + \bar{Q} = 1 = 0$$

$$\bar{Q} = 0 + Q = 0 + 0 = 0 = 1$$

} เซตค่า

กรณีที่ 3: $R = 1$ และ $S = 0$

$$Q = 1 + \bar{Q} = 1 = 0$$

$$\bar{Q} = 0 + Q = 0 + 0 = 0 = 1$$

} เคลียร์ค่า

กรณีที่ 4: $R = 1$ และ $S = 1$

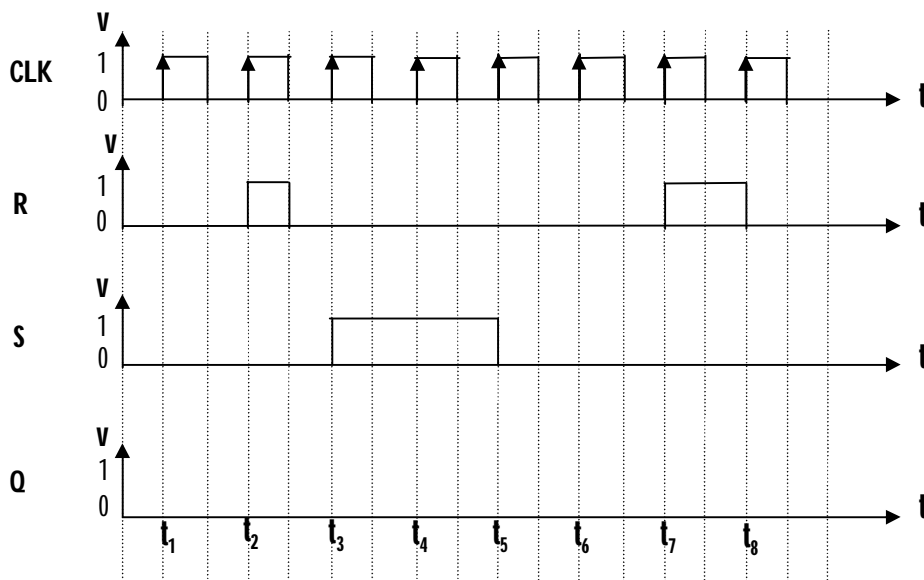
$$Q = 1 + \bar{Q} = 1 = 0$$

$$\bar{Q} = 1 + Q = 1 = 0$$

} ไม่ให้เกิดขึ้น

จากกรณีที่ 4 สังเกตเห็นว่า ค่าเอาต์พุต Q และ \bar{Q} มีค่าเท่ากันคือ 0 ซึ่งขัดแย้งความเป็นจริง จึงเป็นกรณีที่จะไม่เกิดขึ้น

ตัวอย่างที่ 7-1 จากไดอะแกรมแสดงเวลาของฟลิปฟล็อปแบบ RS ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

เวลา t_1 : $R = 0$ และ $S = 0$ ค่า Q จึงคงค่าเดิม แต่เนื่องจากโจทย์ไม่ได้กำหนดค่าเริ่มต้นจึงไม่จำเป็นต้องใส่ผลลัพธ์ของ Q ที่ช่วงเวลา t_1

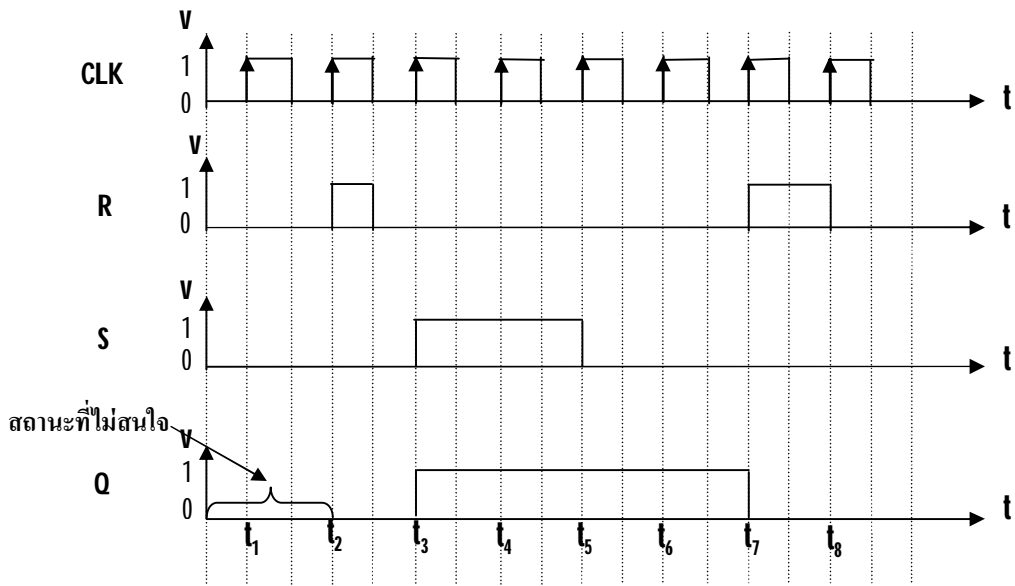
เวลา t_2 : $R = 1$ และ $S = 0$ ได้ค่า Q เป็น 0

เวลา t_3 - t_5 : $R = 0$ และ $S = 1$ ได้ค่า Q เป็น 1

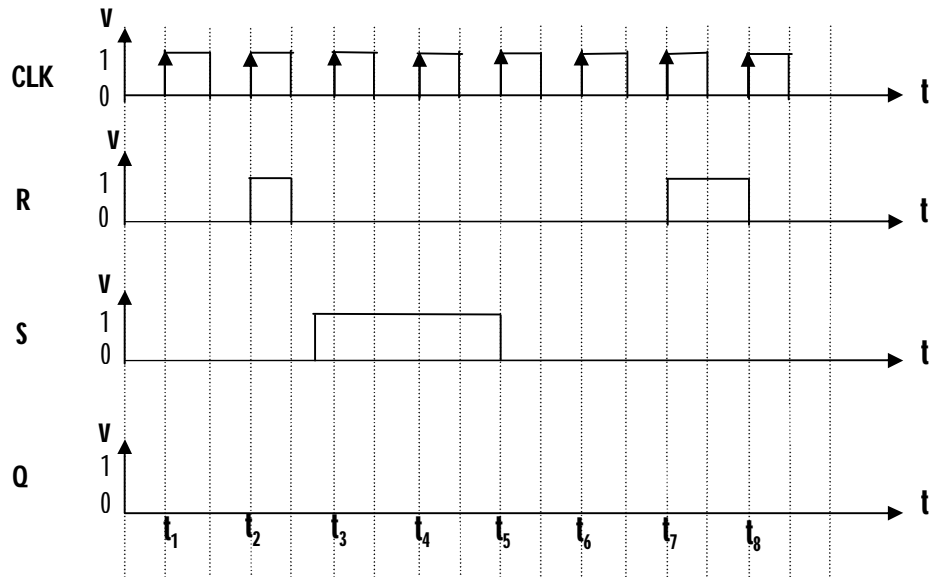
เวลา t_6 : $R = 0$ และ $S = 0$ ค่า Q ต้องคงค่าเดิม เนื่องจากค่า Q ที่เวลา t_5 คือ 1 ดังนั้นได้ค่า Q เป็น 1

เวลา t_7 - t_8 : $R = 1$ และ $S = 0$ ได้ค่า Q เป็น 0

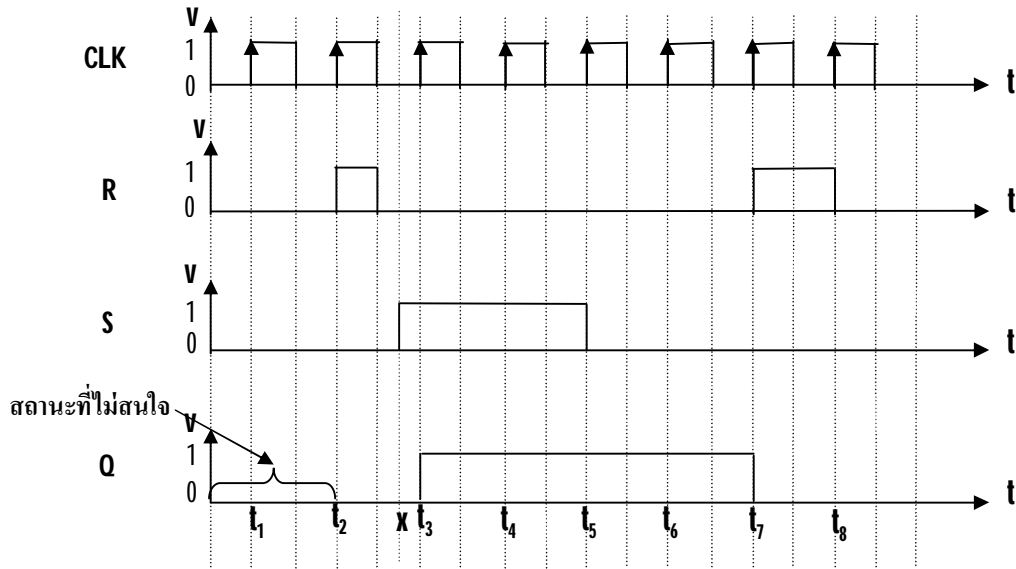
เพราะฉะนั้นจึงได้ค่า Q ในแต่ละช่วงเวลาออกมา ดังนี้



ตัวอย่างที่ 7-2 จากไดอะแกรมแสดงเวลาของฟลิปฟลอปแบบ RS ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา

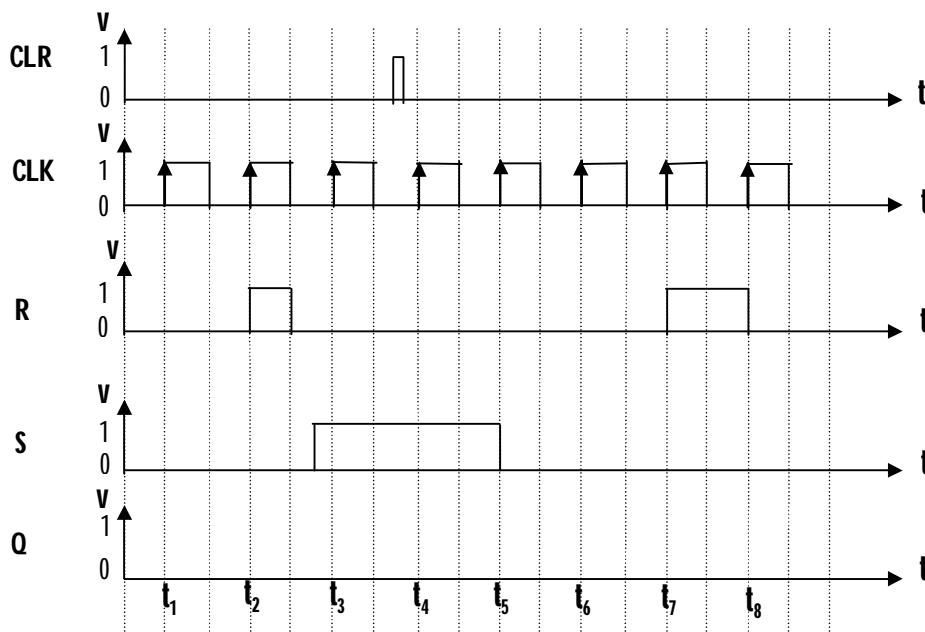


วิธีทำ จากไดอะแกรมแสดงได้ค่า Q ออกมาเป็นดังนี้

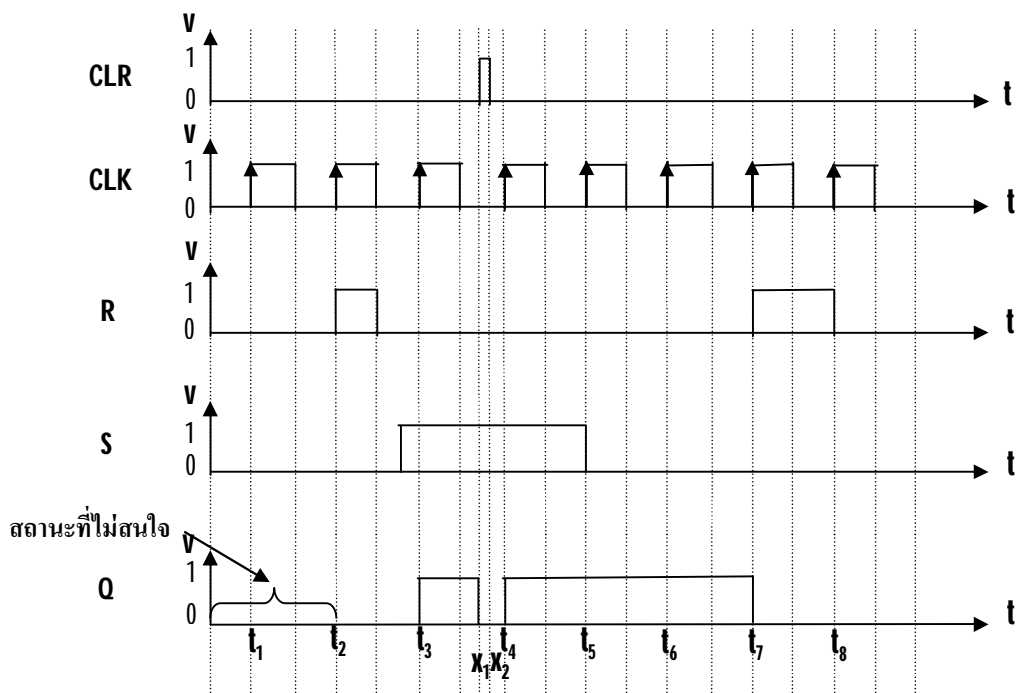


จากค่า Q ที่คำนวณได้พบว่าที่เวลา X ถึงแม้มีการกำหนดค่า $R = 0$ และ $S = 1$ แต่ค่า Q ยังคงมีค่าเป็น 0 เนื่องจากว่าในช่วงเวลาดังกล่าวยังไม่มีการเกิดสัญญาณ CLK จึงทำให้ค่า Q ไม่มีการเปลี่ยนแปลงใดๆ

ตัวอย่างที่ 7-3 จากไดอะแกรมแสดงเวลาของฟลิปฟล็อปแบบ RS ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา



วิธีทำ จากไดอะแกรมแสดงได้ค่า Q ออกมาเป็นดังนี้



จากตัวอย่างที่ 7-3 เนื่องจากช่วงเวลา $x_1 - x_2$ เกิดสัญญาณ CLR ขึ้น จึงทำให้ค่า Q เป็น 0 ในทันทีโดยไม่สนใจว่าในช่วงเวลาดังกล่าวเกิดสัญญาณ CLK หรือไม่ โดยที่ค่า Q จะมีโอกาสที่จะเปลี่ยนแปลงค่าได้อีกก็ต่อเมื่อเกิดสัญญาณ CLK ขึ้นใหม่

7.1.1 ตารางเอ็กโซเทชั่นของฟลิปฟล็อปแบบ RS

ตารางเอ็กโซเทชั่นของฟลิปฟล็อปแบบ RS คือตารางที่ใช้สำหรับการวิเคราะห์หาค่าอินพุต R และ อินพุต S เพื่อให้ได้มาซึ่งค่าของเอาต์พุตใหม่ที่มีสถานะตามต้องการ ซึ่งแบ่งออกเป็น 4 กรณี ดังนี้

กรณีที่ 1: $Q = 0$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.1

กำหนด $R=0, S=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $R=0, S=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $R=1, S=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 0 คือ R มีค่าเป็นได้ทั้ง 0 หรือ 1 จึงกำหนดให้ R เป็นเงื่อนไขที่ไม่สนใจ และ S ต้องมีค่าเป็น 0 เท่านั้น

กรณีที่ 2: $Q = 0$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.1

กำหนด $R=0, S=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $R=0, S=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $R=1, S=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 1 กรณีที่ Q มีค่าเป็น 0 คือ R มีค่าเป็น 0 และ S ต้องมีค่าเป็น 1 เท่านั้น

กรณีที่ 3: $Q = 1$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.1

กำหนด $R=0, S=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $R=0, S=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $R=1, S=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 1 คือ R มีค่าเป็น 1 และ S ต้องมีค่าเป็น 0 เท่านั้น

กรณีที่ 4: $Q = 1$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.1

กำหนด $R=0, S=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $R=0, S=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $R=1, S=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 1 กรณีที่ Q มีค่าเป็น 1 คือ R มีค่าเป็น 0 และ S จะมีค่าเป็นได้ทั้ง 0 หรือ 1 จึงเป็นเงื่อนไขที่ไม่สนใจ

จากเงื่อนไขทั้ง 4 กรณีสามารถสรุปออกมาเป็นตารางเอ็กซ์เทชันได้ดังนี้

ตารางที่ 7.2 ตารางเอ็กซ์เทชันของฟลิปฟล็อปแบบ RS

| อินพุต | | เอาต์พุต | |
|--------|------------|----------|---|
| Q | Q_{next} | R | S |
| 0 | 0 | x | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | x |

7.1.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ RS

การคำนวณหาสมการพีชคณิตบูลีนของ Q_{next} ของฟลิปฟล็อปแบบ RS มีไว้เพื่อใช้สำหรับการวิเคราะห์วงจรเชิงลำดับซึ่งสามารถวิเคราะห์ได้ทั้ง กรณีที่ใช้สัญญาณนาฬิกาชุดเดียวกัน หรือ สัญญาณนาฬิกาคนละชุดกัน โดยแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1: ไม่คิดสัญญาณนาฬิกา ดังนั้นอินพุตทั้งหมดประกอบไปด้วย R, S และ Q ซึ่งมีทั้งหมด 3 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 8 ค่าดังนี้

กำหนด $R=0, S=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$

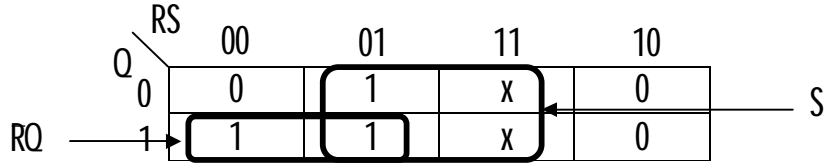
กำหนด $R=0, S=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$

กำหนด $R=0, S=1$ และ $Q = 0$ ดังนั้น $Q_{next} = 1$

กำหนด $R=0, S=1$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$

กำหนด $R=1, S=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$

กำหนด $R=1, S=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 0$
 กำหนด $R=1, S=1$ และ $Q = 0$ -->ไม่ให้เกิดกรณีนี้ จึงเป็นสถานะที่ไม่สนใจ
 กำหนด $R=1, S=1$ และ $Q = 1$ -->ไม่ให้เกิดกรณีนี้ จึงเป็นสถานะที่ไม่สนใจ
 เขียนผลลัพธ์ที่ได้ลงแผนผังคาร์โนห์ได้ดังนี้



รูปที่ 7.7 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟล็อปแบบ RS

ดังนั้นได้ว่า $Q_{next} = RQ + S$

กรณีที่ 2: คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดประกอบไปด้วย R, S, Q และ CLK ซึ่งมีทั้งหมด 4 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 16 ค่าดังนี้

- กำหนด $R=0, S=0, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=0, S=0, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=0, S=0, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=0, S=0, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=0, S=1, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=0, S=1, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=0, S=1, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=0, S=1, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=1, S=0, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=1, S=0, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=1, S=0, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=1, S=0, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=1, S=1, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $R=1, S=1, Q=0$ และ $CLK = 1$ ($R=1, S=1$ ไม่ให้เกิดขึ้น) ดังนั้น $Q_{next} = X$
- กำหนด $R=1, S=1, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $R=1, S=1, Q=1$ และ $CLK = 1$ ($R=1, S=1$ ไม่ให้เกิดขึ้น) ดังนั้น $Q_{next} = X$

| | | | | | | |
|------|------|----|----|----|----|------|
| | | RS | | | | |
| | QCLK | 00 | 01 | 11 | 10 | |
| | 00 | 0 | 0 | 0 | 0 | |
| | 01 | 0 | 1 | X | 0 | SClk |
| RQ | 11 | 1 | 1 | X | 0 | |
| QCLK | 10 | 1 | 1 | 1 | 1 | |

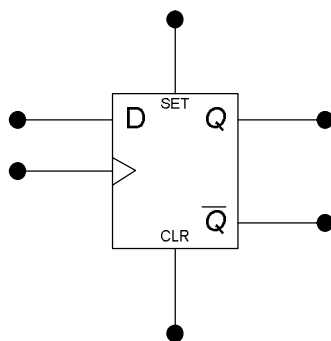
รูปที่ 7.8 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกากรณีใช้ฟลิปฟลอปแบบ RS

ดังนั้นได้ว่า $Q_{next} = RQ + QCLK + SClk$

จากรูปที่ 7.8 ที่ตำแหน่ง R, S, Q และ CLK มีค่าเป็น "1100" และ "1110" ค่า Q_{next} มีค่าเป็น 0 และ 1 ตามลำดับ ถึงแม้ R และ S จะมีค่าเป็น 1 พร้อมกันก็ตาม แต่ยังไม่เกิดสัญญาณ CLK ค่า Q_{next} จึงมีค่าคงเดิม ($Q_{next} = Q$)

7.2 ฟลิปฟลอปแบบ D (D FlipFlop)

ฟลิปฟลอปแบบ D คือฟลิปฟลอปที่มี 1 อินพุตหลักการทำงานของฟลิปฟลอป D คือ กรณีที่เกิดสัญญาณ CLK เอาต์พุตที่ได้จะมีค่าเดียวกับค่า D ($Q_{next} = D$) ซึ่งมีสัญลักษณ์ ดังรูปที่ 7.9 และตารางความจริง ดังตารางที่ 7.3



รูปที่ 7.9 สัญลักษณ์ของฟลิปฟลอปแบบ D

ตารางที่ 7.3 ตารางความจริงของฟลิปฟลอปแบบ D

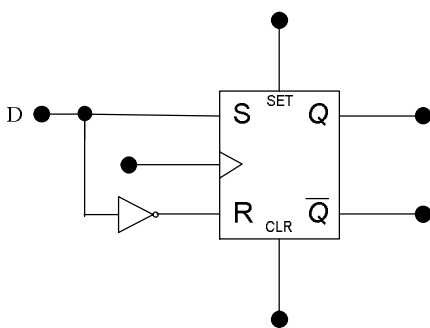
| อินพุต | เอาต์พุต | | สถานะ |
|--------|------------|------------------|---------|
| | Q_{next} | \bar{Q}_{next} | |
| 0 | 0 | 1 | เคลียร์ |
| 1 | 1 | 0 | เซต |

ตารางที่ 7.3 คือ ตารางความจริงของฟลิปฟล็อปแบบ D ซึ่งสามารถอธิบายเอาต์พุตที่เกิดขึ้นได้ดังนี้

กรณีที่ 1 $D = 0$ ค่า Q จะมีค่าเป็น 0 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 2 $D = 1$ ค่า Q จะมีค่าเป็น 1 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

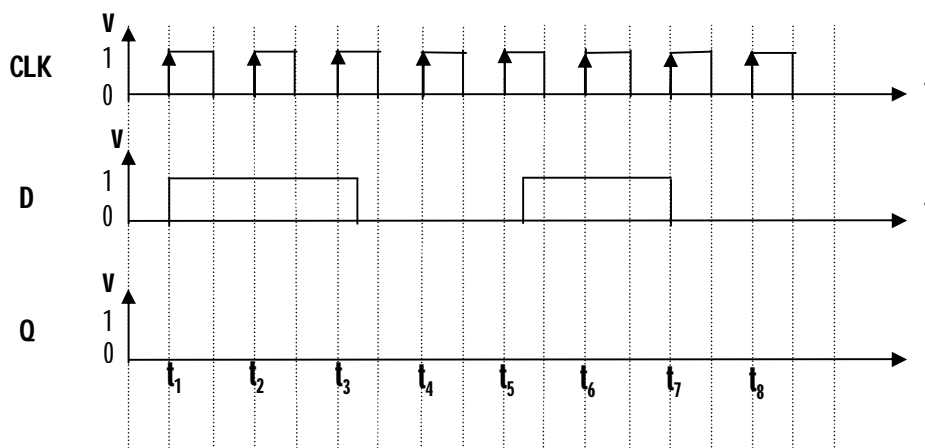
โดยที่สามารถนำโครงสร้างของฟลิปฟล็อปแบบ RS มาสร้างฟลิปฟล็อปแบบ D ได้โดยนำอินพุตของค่า D มาเป็นอินพุตให้กับค่า S และนำอินพุตตัวดังกล่าวมาใส่คอมพลิเมนต์เพื่อเป็นอินพุตให้ค่า R ได้ดังนี้



รูปที่ 7.10 การสร้างฟลิปฟล็อปแบบ D โดยใช้ฟลิปฟล็อปแบบ RS

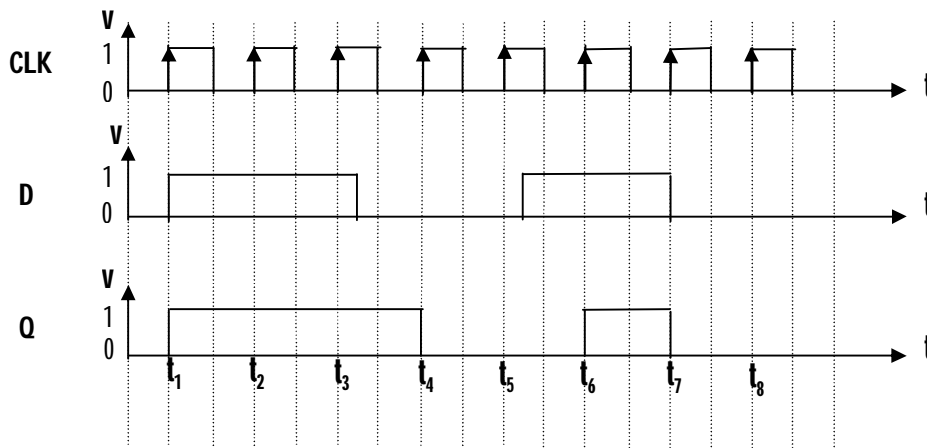
รูปที่ 7.10 แสดงการสร้างฟลิปฟล็อปแบบ D ด้วยฟลิปฟล็อปแบบ RS โดยอินพุตมี 1 ค่าจึงมีทั้งหมด 2 กรณีดังนี้ กรณีที่ 1 กำหนดให้ $D = 0$ จะทำให้ $R = 1$ และ $S = 0$ ซึ่งเป็นการเคลียร์ค่าทำให้ $Q_{next} = 0$ และกรณีที่ 2 กำหนดให้ $D = 1$ จะทำให้ $R = 0$ และ $S = 1$ ซึ่งเป็นการเซตค่าทำให้ $Q_{next} = 1$

ตัวอย่างที่ 7-4 จากไดอะแกรมแสดงเวลาของฟลิปฟล็อปแบบ D ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้
 เวลา t_1-t_3 : $D=1$ ได้ค่า Q_{next} เป็น 1 โดยค่า Q จะเป็นค่าดังกล่าวไปจนถึงเวลา t_4
 เวลา t_4-t_5 : $D=0$ ได้ค่า Q_{next} เป็น 0 โดยค่า Q จะเป็นค่าดังกล่าวไปจนถึงเวลา t_6
 เวลา t_6 : $D=1$ ได้ค่า Q_{next} เป็น 1
 เวลา t_7-t_8 : $D=0$ ได้ค่า Q_{next} เป็น 0

เพราะฉะนั้นจึงได้ค่า Q ในแต่ละช่วงเวลาออกมา ดังนี้



7.2.1 ตารางเอ็ทซเทชั่นของฟลิปฟลอปแบบ D

ตารางเอ็ทซเทชั่นของฟลิปฟลอปแบบ D คือตารางที่ใช้สำหรับการวิเคราะห์หาค่าอินพุต D เพื่อให้ได้มาซึ่งค่าของเอาต์พุตใหม่ที่มีสถานะตามต้องการ ซึ่งแบ่งออกเป็น 4 กรณีดังนี้

กรณีที่ 1: $Q = 0$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.2

กำหนด $D = 0, Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $D = 1, Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นสรุปได้ว่า เมื่อเกิดสัญญาณ CLK เงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 0 คือ D มีค่าเป็น 0

กรณีที่ 2: $Q = 0$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.2

กำหนด $D = 0, Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $D = 1, Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้น สรุปได้ว่า เมื่อเกิดสัญญาณ CLK เงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 1 กรณีที่ Q มีค่าเป็น 0 คือ D มีค่าเป็น 1

กรณีที่ 3: $Q = 1$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.2

กำหนด $D = 0, Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $D = 1, Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข
 ดังนั้น สรุปได้ว่า เมื่อเกิดสัญญาณ CLK เงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0
 กรณีที่ Q มีค่าเป็น 1 คือ D มีค่าเป็น 0

กรณีที่ 4: $Q = 1$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.2

กำหนด $D = 0, Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $D = 1, Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้น สรุปได้ว่า เมื่อเกิดสัญญาณ CLK เงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 1
 กรณีที่ Q มีค่าเป็น 1 คือ D มีค่าเป็น 1

จากเงื่อนไขทั้ง 4 กรณีสามารถสรุปออกมาเป็นตารางอีกโซเทชั่นได้ดังนี้

ตารางที่ 7.4 ตารางอีกโซเทชั่นของฟลิปฟล็อปแบบ D

| อินพุต | | เอาต์พุต |
|--------|------------|----------|
| Q | Q_{next} | D |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

7.2.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D

การคำนวณหาสมการพีชคณิตบูลีนของ Q_{next} ของฟลิปฟล็อปแบบ D มีไว้เพื่อใช้สำหรับการวิเคราะห์วงจรเชิงลำดับซึ่งสามารถวิเคราะห์ได้ทั้ง กรณีที่ใช้สัญญาณนาฬิกาชุดเดียวกัน หรือ สัญญาณนาฬิกาคนละชุดกัน โดยแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1: ไม่คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดมี 2 ตัวคือ D และ Q เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 4 ค่าดังนี้

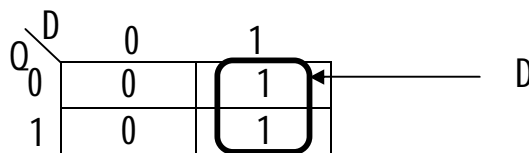
กำหนด $D=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$

กำหนด $D=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 0$

กำหนด $D=1$ และ $Q = 0$ ดังนั้น $Q_{next} = 1$

กำหนด $D=1$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$

เขียนผลลัพธ์ที่ได้ลงแผนผังคาร์โนที่ได้ดังนี้

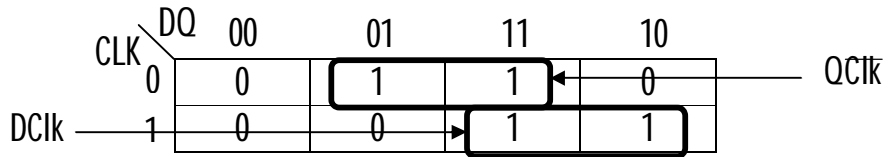


รูปที่ 7.11 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟล็อปแบบ D

ดังนั้นได้ว่า $Q_{next} = D$

กรณีที่ 2: คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดประกอบไปด้วย D, Q และ CLK ซึ่งมีทั้งหมด 3 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 8ค่าดังนี้

- กำหนด $D=0, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $D=0, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $D=0, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $D=0, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $D=1, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $D=1, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $D=1, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $D=1, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$

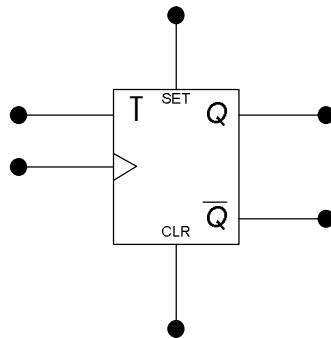


รูปที่ 7.12 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟลอปแบบ D

ดังนั้นได้ว่า $Q_{next} = DClk + QClk$

7.3 ฟลิปฟลอปแบบ T (T FlipFlop)

ฟลิปฟลอปแบบ T คือฟลิปฟลอปที่มี 1 อินพุตหลักการทำงานของฟลิปฟลอป T คือ กรณีที่เกิดสัญญาณ CLK หากกำหนด $T = 0$ แล้วเอาต์พุตที่ได้จะคงเดิม ($Q_{next} = Q$) แต่หากกำหนด $T = 1$ แล้วเอาต์พุตที่ได้จะเกิดจากการกลับสถานะของค่าเดิม ($Q_{next} = \bar{Q}$) โดยฟลิปฟลอปแบบ T มีสัญลักษณ์ดังรูปที่ 7.13 และตารางความจริง ดังตารางที่ 7.5



รูปที่ 7.13 สัญลักษณ์ของฟลิปฟลอปแบบ T

ตารางที่ 7.5 ตารางความจริงของฟลิปฟล็อปแบบ T

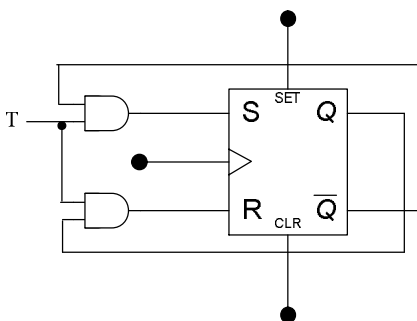
| อินพุต | เอาต์พุต | | สถานะ |
|--------|------------|------------------|-----------------|
| | Q_{next} | \bar{Q}_{next} | |
| T | Q | \bar{Q} | ไม่เปลี่ยนสถานะ |
| 0 | Q | \bar{Q} | ไม่เปลี่ยนสถานะ |
| 1 | \bar{Q} | Q | กลับสถานะ |

ตารางที่ 7.5 คือตารางความจริงของฟลิปฟล็อปแบบ T ซึ่งสามารถอธิบายเอาต์พุตที่เกิดขึ้นได้ดังนี้

กรณีที่ 1 $T = 0$ ค่า Q_{next} จะมีค่าคงเดิม คือสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไร Q_{next} ก็จะมีค่าเป็นเช่นนั้น

กรณีที่ 2 $T = 1$ ค่า Q_{next} จะมีค่าตรงข้ามกับค่าเดิมคือสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไร Q_{next} จะมีค่าตรงข้ามกับค่า Q

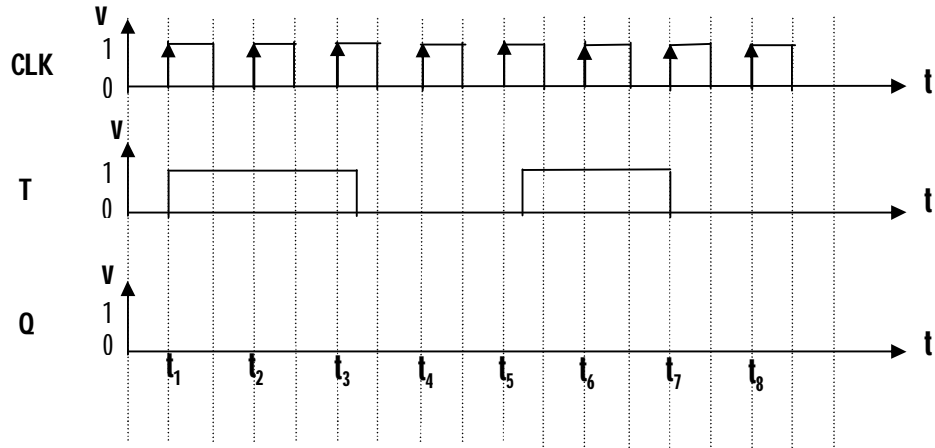
โดยที่สามารถนำโครงสร้างของฟลิปฟล็อปแบบ RS มาสร้างฟลิปฟล็อปแบบ T ได้โดยนำค่า Q มาเป็นอินพุตให้กับค่า R และนำค่า \bar{Q} มาเป็นอินพุตให้ค่า S ได้ดังนี้



รูปที่ 7.14 การสร้างฟลิปฟล็อปแบบ T โดยใช้ฟลิปฟล็อปแบบ RS

รูปที่ 7.14 แสดงการสร้างฟลิปฟล็อปแบบ T ด้วยฟลิปฟล็อปแบบ RS โดยอินพุตมี 1 ค่าจึงมีทั้งหมด 2 กรณีดังนี้ กรณีที่ 1 กำหนดให้ $T = 0$ จะทำให้ $R = 0$ และ $S = 0$ ซึ่งทำให้ Q_{next} ที่ได้มีค่าเท่ากับค่า Q ก่อนที่จะเกิดสัญญาณ CLK และกรณีที่ 2 กำหนดให้ $T = 1$ จะทำให้ $R = \bar{Q}$ และ $S = Q$ ซึ่งทำให้ Q_{next} ที่ได้มีค่าตรงข้ามกับค่า Q ก่อนที่จะเกิดสัญญาณ CLK ยกตัวอย่างเช่น สมมติว่า $Q = 0$, เมื่อเกิดสัญญาณ CLK และกำหนด $T = 1$ จะทำให้ $R = 1 \cdot 0 = 0$ และ $S = 1 \cdot 1 = 1$ ซึ่งทำให้ Q_{next} มีค่าเป็น 1 (ค่าที่ได้ตรงข้ามกับค่า Q เสมอ)

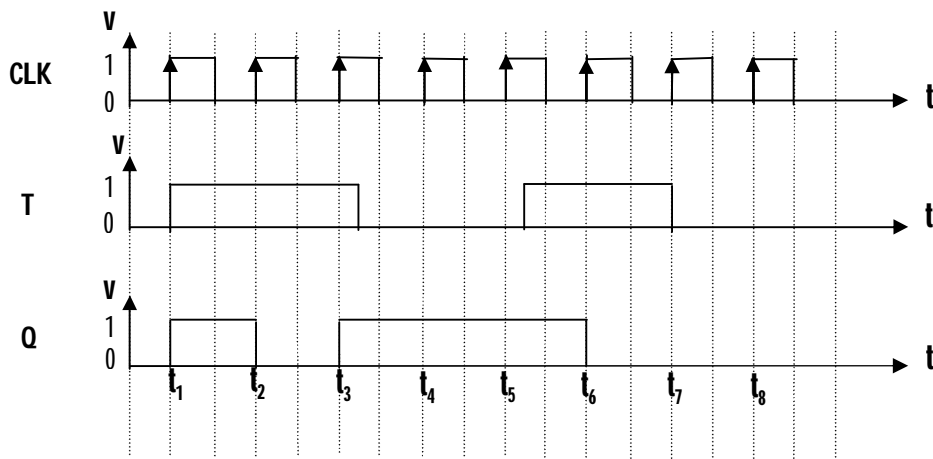
ตัวอย่างที่ 7-5 จากไดอะแกรมแสดงเวลาของฟลิปฟล็อปแบบ T ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา กำหนดให้ค่า Q ก่อนที่จะเกิดสัญญาณ CLK ที่เวลา t_1 มีค่าเป็น 0



วิธีทำจากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

เวลา t_1 : $T = 1$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 0 ทำให้ได้ Q_{next} มีค่าเป็น 1
เวลา t_2 : $T = 1$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 1 ทำให้ได้ Q_{next} มีค่าเป็น 0
เวลา t_3 : $T = 1$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 0 ทำให้ได้ Q_{next} มีค่าเป็น 1
เวลา t_4 : $T = 0$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 1 ทำให้ได้ Q_{next} มีค่าเป็น 1
เวลา t_5 : $T = 0$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 1 ทำให้ได้ Q_{next} มีค่าเป็น 1
เวลา t_6 : $T = 1$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 1 ทำให้ได้ Q_{next} มีค่าเป็น 0
เวลา t_7 : $T = 0$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 0 ทำให้ได้ Q_{next} มีค่าเป็น 0
เวลา t_8 : $T = 0$, เนื่องจากค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็น 0 ทำให้ได้ Q_{next} มีค่าเป็น 0

เพราะฉะนั้นจึงได้ค่า Q ในแต่ละช่วงเวลาออกมา ดังนี้



7.3.1 ตารางเอ็ทซเทชั่นของฟลิปฟลอปแบบ T

ตารางเอ็ทซเทชั่นของฟลิปฟลอปแบบ T คือตารางที่ใช้สำหรับการวิเคราะห์หาค่าอินพุต T เพื่อให้ได้มาซึ่งค่าของเอาต์พุตใหม่ที่มีสถานะตามต้องการ ซึ่งแบ่งออกเป็น 4 กรณีดังนี้

กรณีที่ 1: $Q = 0$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.5

กำหนด $T = 0, Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $T = 1, Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นสรุปได้ว่า เมื่อเกิดสัญญาณ CLK แล้ว Q_{next} จะมีค่าเป็น 0 เสมอ เมื่อ Q มีค่าเป็น 0 และ T มีค่าเป็น 0

กรณีที่ 2: $Q = 0$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.5

กำหนด $T = 0, Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $T = 1, Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้นสรุปได้ว่า เมื่อเกิดสัญญาณ CLK แล้ว Q_{next} จะมีค่าเป็น 1 เสมอ เมื่อ Q มีค่าเป็น 0 และ T มีค่าเป็น 1

กรณีที่ 3: $Q = 1$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.2

กำหนด $T = 0, Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $T = 1, Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้นสรุปได้ว่า เมื่อเกิดสัญญาณ CLK แล้ว Q_{next} จะมีค่าเป็น 0 เสมอ เมื่อ Q มีค่าเป็น 1 และ T มีค่าเป็น 1

กรณีที่ 4: $Q = 1$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.2

กำหนด $T = 0, Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $T = 1, Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นสรุปได้ว่า เมื่อเกิดสัญญาณ CLK แล้ว Q_{next} จะมีค่าเป็น 1 เสมอ เมื่อ Q มีค่าเป็น 1 และ T มีค่าเป็น 0

จากเงื่อนไขทั้ง 4 กรณีสามารถสรุปออกมาเป็นตารางเอ็กซิทเทชันได้ดังนี้

ตารางที่ 7.6 ตารางเอ็กซิทเทชันของฟลิปฟล็อปแบบ T

| อินพุต | | เอาต์พุต |
|--------|------------|----------|
| Q | Q_{next} | T |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

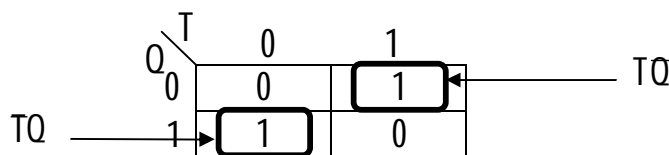
7.3.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ T

การคำนวณหาสมการพีชคณิตบูลีนของ Q_{next} ของฟลิปฟล็อปแบบ T มีไว้เพื่อใช้สำหรับการวิเคราะห์วงจรเชิงลำดับซึ่งสามารถวิเคราะห์ได้ทั้ง กรณีที่ใช้สัญญาณนาฬิกาชุดเดียวกัน หรือ สัญญาณนาฬิกาคนละชุดกัน โดยแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1: ไม่คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดมี 2 ตัวคือ T และ Q เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 4 ค่าดังนี้

- กำหนด $T=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$
- กำหนด $T=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$
- กำหนด $T=1$ และ $Q = 0$ ดังนั้น $Q_{next} = 1$
- กำหนด $T=1$ และ $Q = 1$ ดังนั้น $Q_{next} = 0$

เขียนผลลัพธ์ที่ได้ลงแผนผังคาร์โนที่ได้ดังนี้

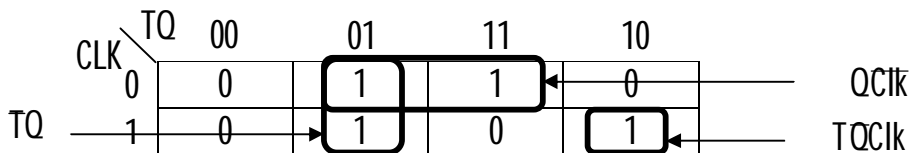


รูปที่ 7.15 การใช้แผนผังคาร์โน้เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่คิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟล็อปแบบ T

ดังนั้นได้ว่า $Q_{next} = TQ + T\bar{Q}$

กรณีที่ 2: คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดประกอบไปด้วย T, Q และ CLK ซึ่งมีทั้งหมด 3 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 8 ค่าดังนี้

- กำหนด $T=0, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $T=0, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $T=0, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $T=0, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $T=1, Q=0$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด $T=1, Q=0$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $T=1, Q=1$ และ $CLK = 0$ (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด $T=1, Q=1$ และ $CLK = 1$ (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$

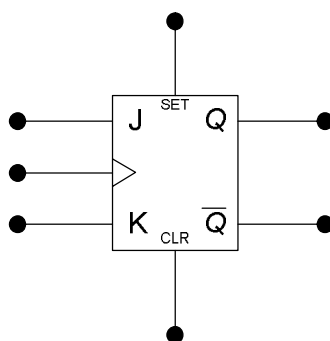


รูปที่ 7.16 การใช้แผนผังคาร์โน้เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีคิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟล็อปแบบ T

$$\text{ดังนั้นได้ว่า } Q_{\text{next}} = TQ + TQCik + QCik$$

7.4 ฟลิปฟลอปแบบ JK (JK FlipFlop)

ฟลิปฟลอปแบบ JK คือฟลิปฟลอปที่มี 2 อินพุตคือ J และ K หลักการทำงานของฟลิปฟลอป JK คือ กรณีที่เกิดสัญญาณ CLK หากกำหนด $J = 0$ และ $K = 0$ แล้วเอาต์พุตที่ได้จะคงเดิม ($Q_{\text{next}} = Q$) หากกำหนด $J = 0$ และ $K = 1$ แล้วเอาต์พุตที่ได้จะมีค่าเป็น 0 เสมอ ($Q_{\text{next}} = 0$) หากกำหนด $J = 1$ และ $K = 0$ แล้วเอาต์พุตที่ได้จะมีค่าเป็น 1 เสมอ ($Q_{\text{next}} = 1$) และหากกำหนด $J = 1$ และ $K = 1$ แล้วเอาต์พุตที่ได้จะเกิดจากการกลับสถานะของค่าเดิม ($Q_{\text{next}} = \bar{Q}$) โดยฟลิปฟลอปแบบ JK มีสัญลักษณ์ดังรูปที่ 7.17 และตารางความจริง ดังตารางที่ 7.7



รูปที่ 7.17 สัญลักษณ์ของฟลิปฟลอปแบบ JK

ตารางที่ 7.7 ตารางความจริงของฟลิปฟลอปแบบ JK

| อินพุต | | เอาต์พุต | | สถานะ |
|--------|---|-------------------|-------------------------|-----------------|
| J | K | Q_{next} | \bar{Q}_{next} | |
| 0 | 0 | Q | \bar{Q} | ไม่เปลี่ยนสถานะ |
| 0 | 1 | 0 | 1 | เซต |
| 1 | 0 | 1 | 0 | เคลียร์ |
| 1 | 1 | \bar{Q} | Q | กลับสถานะ |

ตารางที่ 7.7 คือตารางความจริงของฟลิปฟลอปแบบ JK ซึ่งสามารถอธิบายเอาต์พุตที่เกิดขึ้นได้ดังนี้

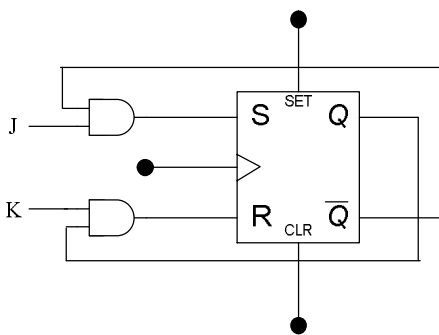
กรณีที่ 1: $J = 0$ และ $K = 0$: ค่า Q_{next} จะมีค่าคงค่าเดิม ความหมายคือ ค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีสถานะเป็นอย่างไร ค่า Q ใหม่ที่เกิดขึ้นหลังเกิดสัญญาณ CLK จะยังคงเป็นค่าเดิม

กรณีที่ 2: $J = 0$ และ $K = 1$: ค่า Q_{next} จะมีค่าเป็น 0 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 3: $J = 1$ และ $K = 0$: ค่า Q_{next} จะมีค่าเป็น 1 เสมอไม่ว่าสถานะของค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 4: $J = 1$ และ $K = 1$: ค่า Q_{next} จะมีค่ากลับสถานะของค่า Q ความหมายคือ ค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีสถานะเป็นอย่างไร ค่า Q ใหม่ที่เกิดขึ้นหลังเกิดสัญญาณ CLK จะมีค่าตรงกันข้ามกับค่า Q ก่อนเกิดสัญญาณ CLK เสมอ

โดยที่สามารถนำโครงสร้างของฟลิปฟล็อปแบบ RS มาสร้างฟลิปฟล็อปแบบ JK ได้โดยนำค่า Q มาเป็นอินพุตที่แอนด์กับ K ให้กับค่า R และนำค่า Q มาเป็นอินพุตที่แอนด์กับ J ให้ค่า S ได้ดังนี้



รูปที่ 7.18 การสร้างฟลิปฟล็อปแบบ JK โดยใช้ฟลิปฟล็อปแบบ RS

รูปที่ 7.18 แสดงการสร้างฟลิปฟล็อปแบบ JK ด้วยฟลิปฟล็อปแบบ RS โดยอินพุตมี 2 ค่าจึงมีทั้งหมด 4 กรณี เนื่องจาก $S = JQ$ และ $R = KQ$ ดังนั้นผลลัพธ์ที่เป็นไปได้ทั้งหมดมีดังนี้

กรณีที่ 1: $J = 0$ และ $K = 0$, จะทำให้ $S = 0 \cdot Q = 0$ และ $R = 0 \cdot Q = 0$ ดังนั้น $Q_{next} = Q$

กรณีที่ 2: $J = 0$ และ $K = 1$, Q_{next} จะถูกพิจารณาออกเป็น 2 กรณีดังนี้

กรณีที่ 2.1: $Q = 0$, จะทำให้ $S = 0 \cdot Q = 0 \cdot 0 = 0$ และ $R = 1 \cdot Q = 1 \cdot 0 = 0$ ดังนั้น

เนื่องจาก Q_{next} จะคงค่าเดิม และค่าเดิมมีค่าเป็น 0 ($Q = 0$) สรุปได้ว่า $Q_{next} = 0$

กรณีที่ 2.2: $Q = 1$, จะทำให้ $S = 0 \cdot Q = 0 \cdot 1 = 0$ และ $R = 1 \cdot Q = 1 \cdot 1 = 1$ ดังนั้น

$Q_{next} = 0$

จากผลลัพธ์ของ Q_{next} ที่ได้ในกรณีที่ 2 สรุปได้ว่า $Q_{next} = 0$ เสมอไม่ว่าค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

กรณีที่ 3: $J = 1$ และ $K = 0$, Q_{next} จะถูกพิจารณาออกเป็น 2 กรณีดังนี้

กรณีที่ 3.1: $Q = 0$, จะทำให้ $S = 1 \cdot Q = 1 \cdot 0 = 0$ และ $R = 0 \cdot Q = 0 \cdot 0 = 0$ ดังนั้น

$Q_{next} = 1$

กรณีที่ 3.2: $Q = 1$, จะทำให้ $S = 1 \cdot Q = 1 \cdot 1 = 1$ และ $R = 0 \cdot Q = 0 \cdot 1 = 0$ ดังนั้น

เนื่องจาก Q_{next} จะคงค่าเดิม และเนื่องจากค่าเดิมมีค่าเป็น 1 ($Q = 1$) สรุปได้ว่า $Q_{next} = 1$

จากผลลัพธ์ของ Q_{next} ที่ได้ในกรณีที่ 3 สรุปได้ว่า $Q_{next} = 1$ เสมอไม่ว่าค่า Q ก่อนที่จะเกิดสัญญาณ CLK มีค่าเป็นอย่างไรก็ตาม

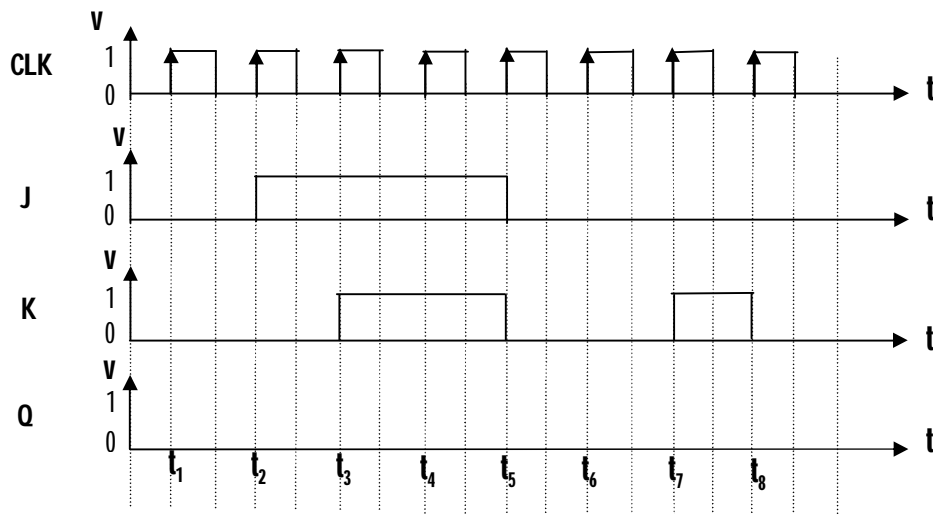
กรณีที่ 4: $J = 1$ และ $K = 1$, Q_{next} จะถูกพิจารณาออกเป็น 2 กรณีดังนี้

กรณีที่ 4.1: $Q = 0$, จะทำให้ $S = 1$ $\bar{Q} = 1.1 = 1$ และ $R = 1.Q = 1.0 = 0$ ดังนั้น $Q_{next} = 1$ ($Q_{next} = \bar{Q}$)

กรณีที่ 4.2 $Q = 1$, จะทำให้ $S = 1.\bar{Q} = 1.0 = 0$ และ $R = 1.Q = 1.1 = 1$ ดังนั้น เนื่องจาก $Q_{next} = 0$ ($Q_{next} = Q$)

จากผลลัพธ์ของ Q_{next} ที่ได้ในกรณีที่ 4 สรุปได้ว่าเมื่อกำหนด $J = 1$ และ $K = 1$, Q_{next} จะเกิดจากการกลับสถานะของค่า Q ในทุกครั้งที่เกิดสัญญาณ CLK

ตัวอย่างที่ 7-6 จากไดอะแกรมแสดงเวลาของฟลิปฟล็อปแบบ JK ต่อไปนี้จงหาเอาต์พุตที่เกิดขึ้นในแต่ละช่วงเวลา กำหนดให้ค่า Q ก่อนที่จะเกิดสัญญาณ CLK ที่เวลา t_1 มีค่าเป็น 0



วิธีทำ จากไดอะแกรมแสดงเวลาสามารถคำนวณหาเอาต์พุตแต่ละช่วงเวลาได้ดังนี้

เวลา t_1 : $J = 0$ และ $K = 0$, ค่า $Q_{next} = Q = 0$

เวลา t_2 : $J = 1$ และ $K = 0$, ค่า $Q_{next} = 1$

เวลา t_3 : $J = 1$ และ $K = 1$ ค่า $Q_{next} = \bar{Q} = 0$

เวลา t_4 : $J = 1$ และ $K = 1$ ค่า $Q_{next} = \bar{Q} = 1$

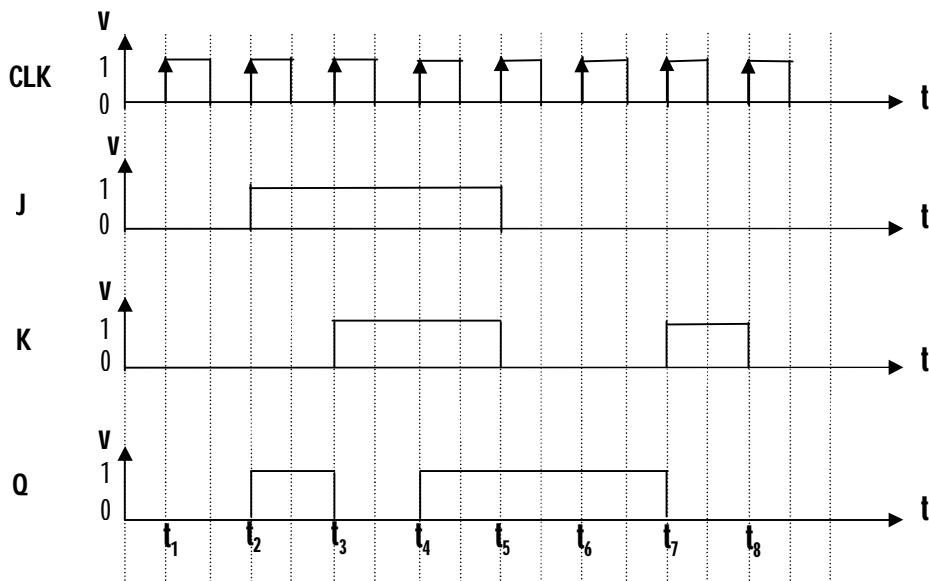
เวลา t_5 : $J = 0$ และ $K = 0$, ค่า $Q_{next} = Q = 1$

เวลา t_6 : $J = 0$ และ $K = 0$, ค่า $Q_{next} = Q = 1$

เวลา t_7 : $J = 0$ และ $K = 1$, ค่า $Q_{next} = 0$

เวลา t_8 : $J = 0$ และ $K = 0$, ค่า $Q_{next} = Q = 0$

เพราะฉะนั้นจึงได้ค่า Q ในแต่ละช่วงเวลาออกมา ดังนี้



7.4.1 ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ JK

ตารางเอ็กไซเทชั่นของฟลิปฟลอปแบบ JK คือตารางที่ใช้สำหรับการวิเคราะห์หาค่าอินพุต J และ อินพุต K เพื่อให้ได้มาซึ่งค่าของเอาต์พุตใหม่ที่มีสถานะตามต้องการ ซึ่งแบ่งออกเป็น 4 กรณี ดังนี้

กรณีที่ 1: $Q = 0$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.7

กำหนด $J=0, K=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $J=0, K=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $J=1, K=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $J=1, K=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 0 คือ J มีค่าเป็น 0 และ K มีค่าเป็นได้ทั้ง 0 หรือ 1 จึงกำหนดให้ K เป็นเงื่อนไขที่ไม่สนใจ

กรณีที่ 2: $Q = 0$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.7

กำหนด $J=0, K=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $J=0, K=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $J=1, K=0$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $J=1, K=1$ ดังนั้นเมื่อ $Q = 0$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข

ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 1 กรณีที่ Q มีค่าเป็น 0 คือ J มีค่าเป็น 1 และ K มีค่าเป็นได้ทั้ง 0 หรือ 1 จึงกำหนดให้ K เป็นเงื่อนไขที่ไม่สนใจ

กรณีที่ 3: $Q = 1$ ต้องการได้ $Q_{next} = 0$

จากตารางที่ 7.7

กำหนด $J=0, K=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข

กำหนด $J=0, K=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข

กำหนด $J=1, K=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ไม่ตรงตามเงื่อนไข
 กำหนด $J=1, K=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ตรงตามเงื่อนไข
 ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 0 คือ J มีค่าเป็นได้ทั้ง 0 หรือ 1 จึงกำหนดให้ J เป็นเงื่อนไขที่ไม่สนใจและ K มีค่าเป็น 1

กรณีที่ 4: $Q = 1$ ต้องการได้ $Q_{next} = 1$

จากตารางที่ 7.7

กำหนด $J=0, K=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข
 กำหนด $J=0, K=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข
 กำหนด $J=1, K=0$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 1 \rightarrow$ ตรงตามเงื่อนไข
 กำหนด $J=1, K=1$ ดังนั้นเมื่อ $Q = 1$ จะได้ $Q_{next} = 0 \rightarrow$ ไม่ตรงตามเงื่อนไข
 ดังนั้นเงื่อนไขที่เป็นไปได้ที่จะทำให้ได้ Q_{next} มีค่าเป็น 0 กรณีที่ Q มีค่าเป็น 0 คือ J มีค่าเป็นได้ทั้ง 0 หรือ 1 จึงกำหนดให้ J เป็นเงื่อนไขที่ไม่สนใจและ K มีค่าเป็น 0
 จากเงื่อนไขทั้ง 4 กรณีสามารถสรุปออกมาเป็นตารางอีกโซเทชั่นได้ดังนี้

ตารางที่ 7.8 ตารางอีกโซเทชั่นของฟลิปฟล็อปแบบ JK

| อินพุต | | เอาต์พุต | |
|--------|------------|----------|---|
| Q | Q_{next} | J | K |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | X | 1 |
| 1 | 1 | X | 0 |

7.4.2 การคำนวณหาสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ JK

การคำนวณหาสมการพีชคณิตบูลีนของ Q_{next} ของฟลิปฟล็อปแบบ JK มีไว้เพื่อใช้สำหรับการวิเคราะห์วงจรเชิงลำดับซึ่งสามารถวิเคราะห์ได้ทั้ง กรณีที่ใช้สัญญาณนาฬิกาชุดเดียวกัน หรือ สัญญาณนาฬิกาคนละชุดกัน โดยแบ่งออกเป็น 2 กรณีดังนี้

กรณีที่ 1: ไม่คิดสัญญาณนาฬิกา ดังนั้นอินพุตทั้งหมดประกอบไปด้วย J, K และ Q ซึ่งมีทั้งหมด 3 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 8 ค่าดังนี้

กำหนด $J=0, K=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$

กำหนด $J=0, K=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$

กำหนด $J=0, K=1$ และ $Q = 0$ ดังนั้น $Q_{next} = 0$

กำหนด $J=0, K=1$ และ $Q = 1$ ดังนั้น $Q_{next} = 0$

กำหนด $J=1, K=0$ และ $Q = 0$ ดังนั้น $Q_{next} = 1$

กำหนด $J=1, K=0$ และ $Q = 1$ ดังนั้น $Q_{next} = 1$

กำหนด $J=1, K=1$ และ $Q = 0$ ดังนั้น $Q_{next} = 1$

กำหนด $J=1, K=1$ และ $Q = 1$ ดังนั้น $Q_{next} = 0$

เขียนผลลัพธ์ที่ได้ลงแผนผังคาร์โนที่ได้ดังนี้

| | | | | | | |
|----|----|----|----|----|----|------|
| | JK | 00 | 01 | 11 | 10 | |
| Q | 0 | 0 | 0 | 1 | 1 | ← JQ |
| 1 | 1 | 0 | 0 | 1 | | |
| KQ | 1 | 1 | 0 | 0 | 1 | |

รูปที่ 7.19 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีไม่เกิดสัญญาณนาฬิกา กรณีใช้ฟลิปฟล็อปแบบ JK

ดังนั้นได้ว่า $Q_{next} = JQ + KQ$

กรณีที่ 2: คิดสัญญาณนาฬิกาดังนั้นอินพุตทั้งหมดประกอบไปด้วย J, K, Q และ CLK ซึ่งมีทั้งหมด 4 ตัว เพราะฉะนั้นค่าที่เป็นไปได้ทั้งหมดจึงมี 16 ค่าดังนี้

- กำหนด J=0, K=0, Q=0 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=0, K=0, Q=0 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=0, K=0, Q=1 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=0, K=0, Q=1 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=0, K=1, Q=0 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=0, K=1, Q=0 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=0, K=1, Q=1 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=0, K=1, Q=1 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=1, K=0, Q=0 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=1, K=0, Q=0 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=1, K=0, Q=1 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=1, K=0, Q=1 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=1, K=1, Q=0 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$
- กำหนด J=1, K=1, Q=0 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=1, K=1, Q=1 และ CLK = 0 (ไม่เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 1$
- กำหนด J=1, K=1, Q=1 และ CLK = 1 (เกิดสัญญาณนาฬิกา) ดังนั้น $Q_{next} = 0$

| | | | | | | |
|------|----|----|----|----|----|---------|
| | JK | 00 | 01 | 11 | 10 | |
| QCLK | 00 | 0 | 0 | 0 | 0 | ← JQClk |
| 01 | 0 | 0 | 0 | 1 | 1 | |
| KQ | 11 | 1 | 0 | 0 | 1 | |
| QClk | 10 | 1 | 1 | 1 | 1 | |

รูปที่ 7.20 การใช้แผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของ Q_{next} ในกรณีเกิดสัญญาณนาฬิกากรณีใช้ฟลิปฟล็อปแบบ JK

ดังนั้นได้ว่า $Q_{next} = KQ + QClk + JQClk$

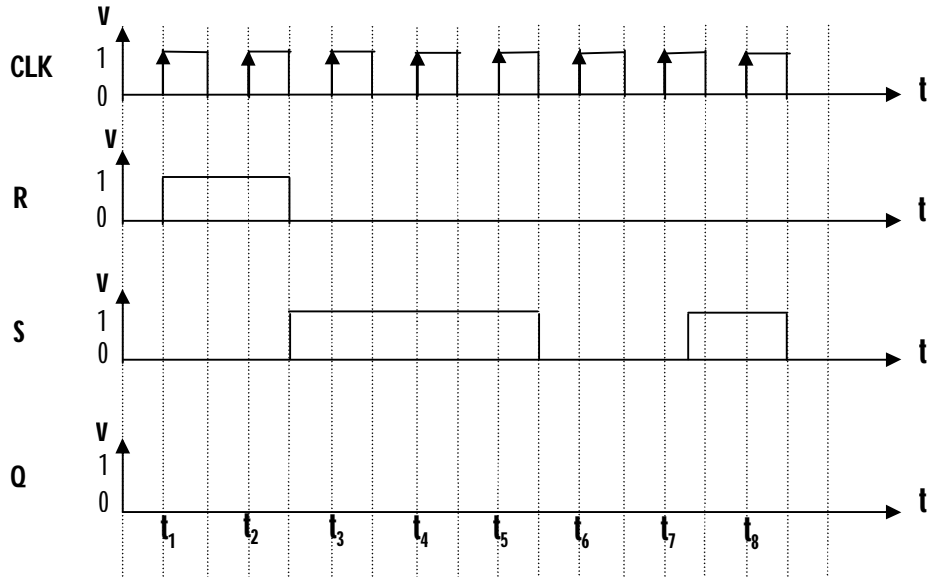
7.5 บทสรุป

ฟลิปฟล็อป คืออุปกรณ์ทางอิเล็กทรอนิกส์ที่มีหน่วยความจำขนาด 1 บิต ซึ่งสามารถถูกสร้างได้โดยใช้เกตพื้นฐานเช่นแนนด์เกต หรือ นอร์เกต โดยจะมีสัญญาณเอาต์พุต 2 สัญญาณคือ Q และ \bar{Q} โดยที่ทั้ง 2 สัญญาณนี้จะต้องเป็นนิเสธซึ่งกันและกันเสมอ ส่วนอินพุตจะมี 2 แบบ คือฟลิปฟล็อปแบบ 1 อินพุตมี 2 ชนิด คือ ฟลิปฟล็อปแบบ D และ ฟลิปฟล็อป แบบ T และฟลิปฟล็อปแบบ 2 อินพุตแบ่งเป็น 2 ชนิด คือ ฟลิปฟล็อปแบบ RS และ ฟลิปฟล็อปแบบ JK โดยฟลิปฟล็อปจะใช้สัญญาณนาฬิกา CLK เป็นสัญญาณที่ใช้สำหรับการกำหนดจังหวะของการเปลี่ยนแปลงสัญญาณเอาต์พุต ใช้สัญญาณ CLR เพื่อกำหนดให้สัญญาณ Q มีค่าเป็น 0 และใช้สัญญาณ SET เพื่อกำหนดให้สัญญาณ Q มีค่าเป็น 1

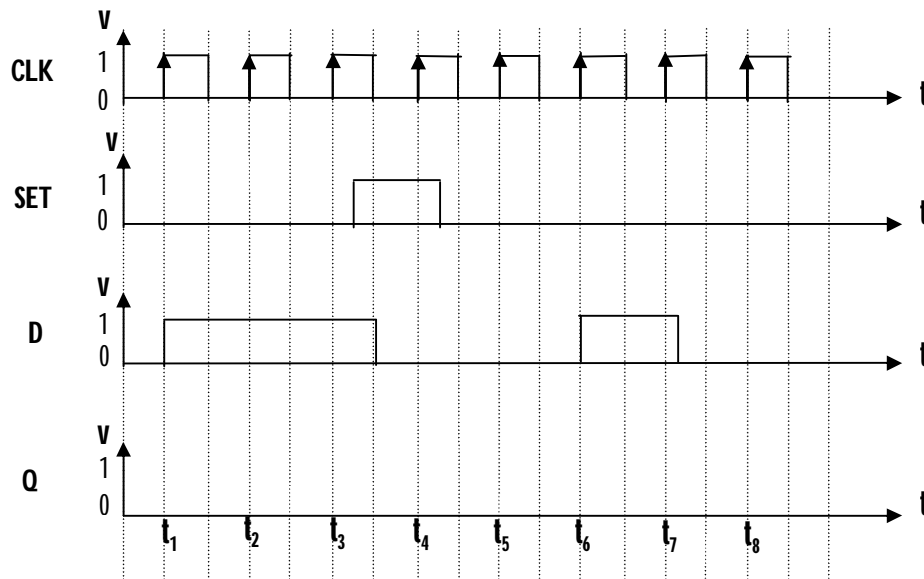
คำถามท้ายบท

1. ฟลิปฟล็อปคืออะไร
2. สัญญาณนาฬิกาคืออะไร
3. สัญญาณเซตคืออะไร
4. สัญญาณเคลียร์คืออะไร
5. จงหาสถานะของ Q ที่เกิดขึ้นในแต่ละช่วงเวลาดังต่อไปนี้

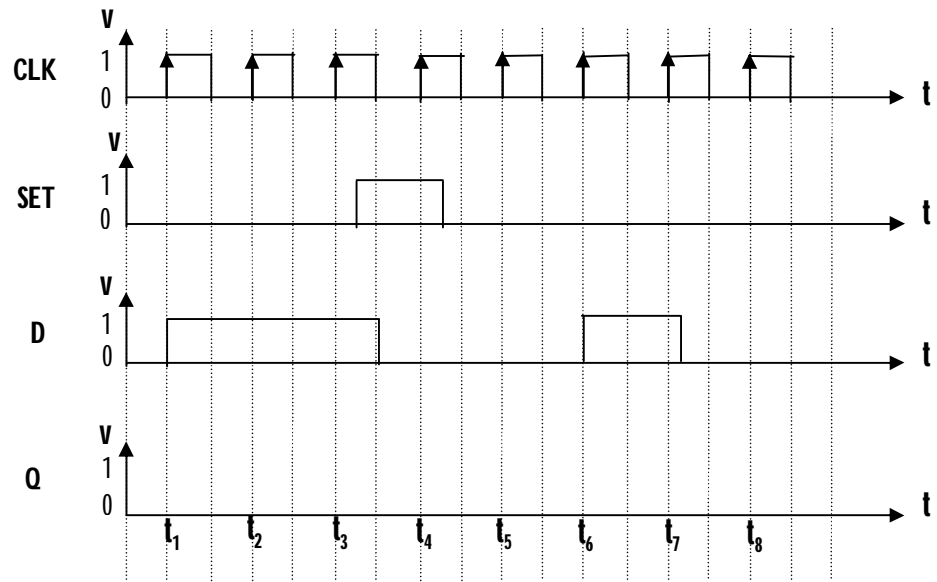
5.1)



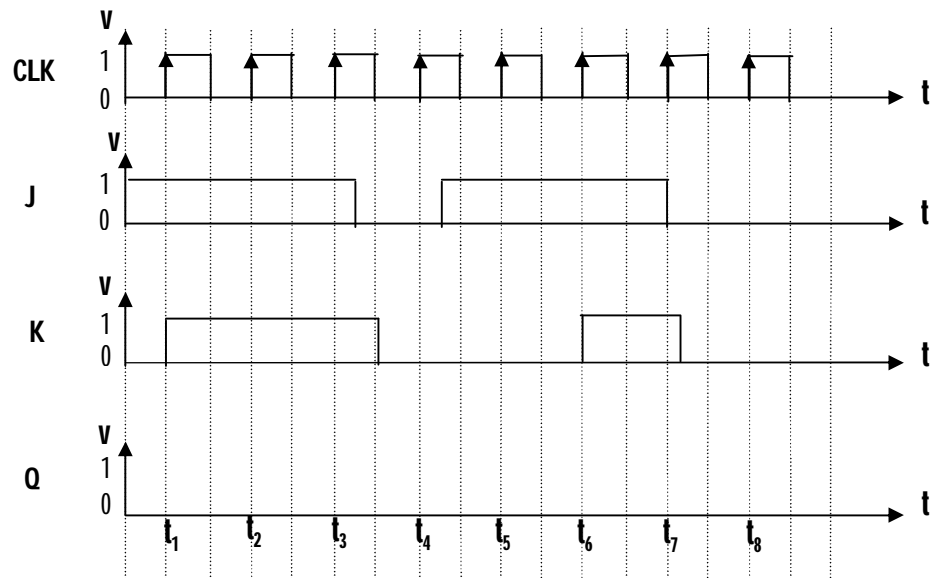
5.2)



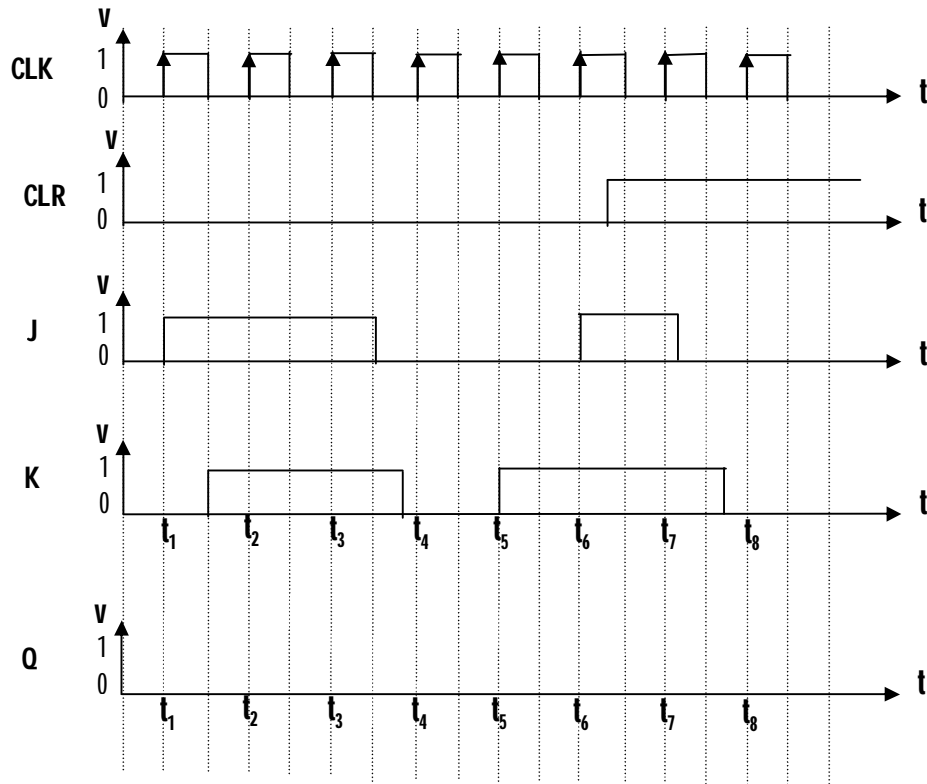
5.3)



5.4)



5.5)



6. เหตุใดจึงไม่สามารถกำหนดให้สัญญาณเซต และสัญญาณเคลียร์เกิดขึ้นพร้อมกันได้
7. จากรูปที่ 7.6 จงออกแบบวงจรใหม่โดยใช้แชนด์เกต แทนนอร์เกตทั้งหมดโดยการทำงานของวงจรใหม่ยังคงเหมือนดังตารางที่ 7.1

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.
- สมชาย ชื่นวัฒนาประณี. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุตรธานี: มหาวิทยาลัยราชภัฏอุตรธานี.**

แผนบริหารการสอนประจำบทที่ 8

การออกแบบวงจรเชิงลำดับ 6 ชั่วโมง

หัวข้อเนื้อหา

- 8.1 การออกแบบแผนภาพสถานะและตารางสถานะ
 - 8.1.1 รูปแบบของเมลลี
 - 8.1.2 รูปแบบของมัวร์
- 8.2 การวิเคราะห์ห้วงจรเชิงลำดับ
 - 8.2.1 วิธีการวิเคราะห์ห้วงจรเชิงลำดับแบบเข้าจังหวะ
 - 8.2.2 วิธีการวิเคราะห์ห้วงจรเชิงลำดับแบบไม่เข้าจังหวะ
- 8.3 การออกแบบวงจรเชิงลำดับ
- 8.4 การลดสถานะเอาต์พุตของฟลิปฟล็อป
- 8.5 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับแผนภาพสถานะและตารางสถานะเพื่อนำไปใช้สำหรับการวิเคราะห์และออกแบบวงจร
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการวิเคราะห์ห้วงจรเชิงลำดับทั้งแบบเข้าจังหวะและแบบไม่เข้าจังหวะ
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการออกแบบวงจรเชิงลำดับ
4. เพื่อให้ผู้เรียนสามารถลดสถานะเอาต์พุตของฟลิปฟล็อปกรณีที่มีมากเกินไปได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 8

การออกแบบวงจรเชิงลำดับ

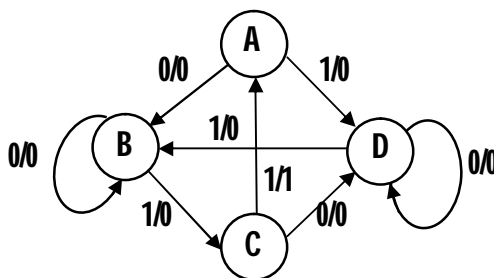
วงจรเชิงลำดับ คือวงจรที่เกิดจากการต่อใช้งานร่วมกันระหว่างเกตพื้นฐานต่างๆ กับฟลิปฟล็อป อย่างน้อย 1 ตัว โดยแบ่งออกเป็น 2 ประเภท ได้แก่ วงจรเชิงลำดับแบบเข้าจังหวะ (Synchronous Sequential Logic) ลักษณะการทำงานของวงจร คือ ฟลิปฟล็อปทุกตัวจะถูกระตุ้นด้วยสัญญาณนาฬิกาตัวเดียวกันเพื่อให้ทำงานไปพร้อมกัน และ วงจรเชิงลำดับแบบไม่เข้าจังหวะ (Asynchronous Sequential Logic) ลักษณะการทำงานของวงจร คือฟลิปฟล็อปอย่างน้อย 1 ตัวที่ไม่ถูกระตุ้นด้วยสัญญาณนาฬิกาเดียวกับฟลิปฟล็อปตัวอื่นๆ

8.1 การออกแบบแผนภาพสถานะและตารางสถานะ

แผนภาพสถานะ (State Diagram) และตารางสถานะ (Table Diagram) คือ เครื่องมือที่ถูกนำมาใช้สำหรับการแสดงการเปลี่ยนแปลงสถานะของวงจรเชิงลำดับในแต่ละช่วงเวลา เพื่อช่วยให้การวิเคราะห์และการออกแบบวงจรเชิงลำดับสามารถทำได้ง่ายยิ่งขึ้น โดยใช้ตัวอักษรภาษาอังกฤษแสดงถึงสถานะต่างๆ ของวงจร ซึ่งมีทั้งหมด 2 รูปแบบคือรูปแบบของเมลลี (Mealy Model) และรูปแบบของมัวร์ (Moore Model)

8.1.1 รูปแบบของเมลลี

สถานะเอาต์พุตสำหรับรูปแบบของเมลลีจะขึ้นอยู่กับค่าอินพุตที่ป้อนเข้าวงจร และสถานะของวงจรในปัจจุบัน โดยตัวอย่างแผนภาพสถานะเป็นดังต่อไปนี้



รูปที่ 8.1 ตัวอย่างแผนภาพสถานะรูปแบบเมลลี

รูปที่ 8.1 แสดงตัวอย่างแผนภาพสถานะรูปแบบของเมลลี ซึ่งจะเก็บสถานะไว้ภายในวงกลม ส่วนการเชื่อมโยงระหว่างสถานะปัจจุบันและสถานะใหม่ของวงจรจะใช้เส้นโค้งหรือเส้นตรงสำหรับการเชื่อมโยงโดยแต่ละเส้นจะมีเลขกำกับอยู่ในรูปแบบ x/y โดยที่ x คือ อินพุตที่ป้อนเข้าสู่วงจร และ y คือสถานะของเอาต์พุตซึ่งจะขึ้นอยู่กับสถานะวงจรปัจจุบันและค่าอินพุตซึ่งอธิบายการทำงานได้เป็นดังนี้

สถานะ A: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ B และมีสถานะเอาต์พุตเป็น 0

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ D และมีสถานะเอาต์พุตเป็น 0

สถานะ B: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ B และมีสถานะเอาต์พุตเป็น 0

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ C และมีสถานะเอาต์พุตเป็น 0

สถานะ C: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ D และมีสถานะเอาต์พุตเป็น 0

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ A และมีสถานะเอาต์พุตเป็น 1

สถานะ D: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ D และมีสถานะเอาต์พุตเป็น 0

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรถือ B และมีสถานะเอาต์พุตเป็น 0

ตารางที่ 8.1 ตัวอย่างตารางสถานะรูปแบบเมลลี

| สถานะปัจจุบัน | สถานะที่เกิดขึ้นใหม่/เอาต์พุต | |
|---------------|-------------------------------|---------|
| | $x = 0$ | $x = 1$ |
| A | B/0 | D/0 |
| B | B/0 | C/0 |
| C | D/0 | A/1 |
| D | D/0 | B/0 |

ตารางที่ 8.1 แสดงตารางสถานะที่แปลงมาจากแผนภาพสถานะของรูปที่ 8.1 คอลัมน์ที่ 1 แสดงสถานะปัจจุบันซึ่งมีทั้งหมด 4 สถานะคือ A – D คอลัมน์ที่ 2 และ 3 แสดงผลลัพธ์ของสถานะที่เกิดขึ้นใหม่ และเอาต์พุตซึ่งเขียนได้เป็น สถานะที่เกิดขึ้นใหม่/เอาต์พุต ซึ่งขึ้นอยู่กับสถานะปัจจุบันและอินพุต โดยที่คอลัมน์ที่ 2 เกิดจากกรณีที่อินพุตมีค่าเป็น 0 ($x = 0$) และ คอลัมน์ที่ 3 เกิดจากกรณีที่อินพุตมีค่าเป็น 1 ($x = 1$)

ตัวอย่างที่ 8-1 จากแผนภาพสถานะดังรูปที่ 8.1 หรือ ตารางสถานะดังตารางที่ 8.1 กำหนดให้สถานะปัจจุบันอยู่ที่ B จงหาสถานะที่เกิดขึ้นใหม่ และเอาต์พุตที่เกิดขึ้น กรณีที่ลำดับของอินพุตที่ป้อนเข้ามามีค่าเป็น 11010

วิธีทำ เนื่องจากอินพุตถูกป้อนเข้ามาเป็นลำดับทั้งหมด 5 ค่า การคำนวณหาสถานะที่เกิดขึ้นใหม่ และเอาต์พุต เป็นดังนี้

ลำดับที่ 1: สถานะปัจจุบัน คือ **B** และ อินพุตมีค่าเป็น **1** ดังนั้นสถานะที่เกิดขึ้นใหม่จึงมีค่าเป็น **C** และเอาต์พุตมีค่าเป็น **0**

ลำดับที่ 2: สถานะที่เกิดขึ้นใหม่ (**C**) จากลำดับที่ 1 กลายมาเป็นสถานะปัจจุบันของลำดับที่ 2 และอินพุตมีค่าเป็น **1** ดังนั้น สถานะที่เกิดขึ้นใหม่มีค่าเป็น **A** และเอาต์พุตมีค่าเป็น **1**

ลำดับที่ 3: สถานะที่เกิดขึ้นใหม่ (**A**) จากลำดับที่ 2 กลายมาเป็นสถานะปัจจุบันของลำดับที่ 3 และอินพุตมีค่าเป็น **0** ดังนั้น สถานะที่เกิดขึ้นใหม่มีค่าเป็น **B** และเอาต์พุตมีค่าเป็น **0**

ลำดับที่ 4: สถานะที่เกิดขึ้นใหม่ (**B**) จากลำดับที่ 3 กลายมาเป็นสถานะปัจจุบันของลำดับที่ 4 และอินพุตมีค่าเป็น **1** ดังนั้น สถานะที่เกิดขึ้นใหม่มีค่าเป็น **C** และเอาต์พุตมีค่าเป็น **0**

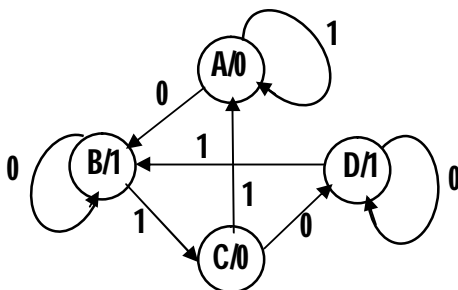
ลำดับที่ 5: สถานะที่เกิดขึ้นใหม่ (**C**) จากลำดับที่ 4 กลายมาเป็นสถานะปัจจุบันของลำดับที่ 5 และอินพุตมีค่าเป็น **0** ดังนั้น สถานะที่เกิดขึ้นใหม่มีค่าเป็น **D** และเอาต์พุตมีค่าเป็น **0**

จากผลลัพธ์ที่ได้ทั้งหมด ถูกนำมาเขียนเป็นลำดับได้ดังนี้

| | | | | | |
|-----------------------|---|---|---|---|---|
| อินพุต: | 1 | 1 | 0 | 1 | 0 |
| สถานะปัจจุบัน: | B | C | A | B | C |
| สถานะที่เกิดขึ้นใหม่: | C | A | B | C | D |
| เอาต์พุต: | 0 | 1 | 0 | 0 | 0 |

8.1.2 รูปแบบของมัวร์

สถานะเอาต์พุตสำหรับรูปแบบของมัวร์จะขึ้นอยู่กับ สถานะของวงจรในปัจจุบันเท่านั้น โดยตัวอย่างแผนภาพสถานะเป็นดังต่อไปนี้



รูปที่ 8.2 ตัวอย่างแผนภาพสถานะรูปแบบมัวร์

รูปที่ 8.2 แสดงตัวอย่างแผนภาพสถานะรูปแบบของมัวร์ ซึ่งจะเก็บสถานะ และเอาต์พุตไว้ภายในวงกลม ส่วนการเชื่อมโยงระหว่างสถานะปัจจุบันและสถานะใหม่ของวงจรจะใช้เส้นโค้งหรือเส้นตรงสำหรับการเชื่อมโยงโดยแต่ละเส้นจะใช้อินพุตเป็นตัวเชื่อมโยงซึ่งอธิบายการทำงานได้เป็นดังนี้

สถานะ A: เมื่ออินพุตมีค่าเป็น **0** จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ **B** และ มีสถานะเอาต์พุตเป็น **0**

เมื่ออินพุตมีค่าเป็น **1** จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ **A** และ มีสถานะเอาต์พุตเป็น **0**

สถานะ B: เมื่ออินพุตมีค่าเป็น **0** จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ **B** และ มีสถานะเอาต์พุตเป็น **1**

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ C และมีสถานะเอาต์พุตเป็น 1

สถานะ C: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ D และมีสถานะเอาต์พุตเป็น 0

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ A และมีสถานะเอาต์พุตเป็น 0

สถานะ D: เมื่ออินพุตมีค่าเป็น 0 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ D และมีสถานะเอาต์พุตเป็น 1

เมื่ออินพุตมีค่าเป็น 1 จะทำให้สถานะที่เกิดขึ้นใหม่ของวงจรคือ B และมีสถานะเอาต์พุตเป็น 1

ตารางที่ 8.2 ตัวอย่างตารางสถานะรูปแบบมัวร์

| สถานะปัจจุบัน | สถานะที่เกิดขึ้นใหม่ | | เอาต์พุต |
|---------------|----------------------|---------|----------|
| | $x = 0$ | $x = 1$ | |
| A | B | A | 0 |
| B | B | C | 1 |
| C | D | A | 0 |
| D | D | B | 1 |

ตารางที่ 8.2 แสดงตารางสถานะที่แปลงมาจากแผนภาพสถานะของรูปที่ 8.2 คอลัมน์ที่ 1 แสดงสถานะปัจจุบันซึ่งมีทั้งหมด 4 สถานะคือ A – D คอลัมน์ที่ 2 และ 3 แสดงผลลัพธ์ของสถานะที่เกิดขึ้นใหม่ โดยที่คอลัมน์ที่ 2 คือสถานะที่เกิดขึ้นใหม่ที่เกิดจากกรณีที่อินพุตมีค่าเป็น 0 ($x = 0$) คอลัมน์ที่ 3 คือสถานะที่เกิดขึ้นใหม่ที่เกิดจากกรณีที่อินพุตมีค่าเป็น 1 ($x = 1$) และคอลัมน์ที่ 4 แสดงผลลัพธ์ของเอาต์พุตซึ่งขึ้นอยู่กับสถานะปัจจุบันเท่านั้น

เนื่องจากรูปแบบเอาต์พุตที่ได้จากรูปแบบมัวร์นั้นจะขึ้นอยู่กับเพียงแค่สถานะปัจจุบัน การใช้รูปแบบมัวร์ในการวิเคราะห์และออกแบบวงจรจึงทำให้มีสถานะที่มากขึ้น เมื่อเปรียบเทียบกับการใช้รูปแบบเมลลี ดังนั้นเอกสารประกอบการสอนเรื่องนี้จึงเลือกใช้เพียงรูปแบบของเมลลีเท่านั้น เนื่องจากการใช้รูปแบบของเมลลีสำหรับการวิเคราะห์และออกแบบวงจรสามารถทำได้ง่ายกว่า

8.2 การวิเคราะห์วงจรเชิงลำดับ

การวิเคราะห์วงจรเชิงลำดับ มีวัตถุประสงค์ในเพื่อใช้สำหรับการวิเคราะห์หาสถานะต่าง ๆ ของวงจร และเอาต์พุตที่จะเกิดขึ้น โดยพิจารณาจากสถานะปัจจุบัน และค่าของอินพุตที่เกิดขึ้นในแต่ละช่วงเวลา แบ่งออกเป็น 2 วิธีคือวิธีการวิเคราะห์วงจรเชิงลำดับแบบเข้าจังหวะ และวิธีการวิเคราะห์วงจรเชิงลำดับแบบไม่เข้าจังหวะ

8.2.1 วิธีการวิเคราะห์วงจรเชิงลำดับแบบเข้าจังหวะ

การวิเคราะห์วงจรเชิงลำดับแบบเข้าจังหวะ ใช้สำหรับการวิเคราะห์วงจรเชิงลำดับในกรณีที่มีฟลิปฟล็อปทุกตัวของวงจรใช้สัญญาณนาฬิกาาร่วมกัน ซึ่งมีขั้นตอนเป็นดังนี้

ขั้นตอนที่ 1: หาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทุกตัว และเอาต์พุตวงจร

ขั้นตอนที่ 2: สร้างแผนผังคาร์โนห์ของอินพุตของฟลิปฟล็อปทั้งหมด และเอาต์พุตวงจร โดยที่แผนผังคาร์โนห์ 1 แผนผังใช้สำหรับแสดงสถานะที่เป็นไปได้ทั้งหมดของอินพุตของฟลิปฟล็อปหรือเอาต์พุตของวงจร 1 ตัว สำหรับค่าผลลัพธ์ที่ได้แต่ละช่องในแผนผังคาร์โนห์ได้มาจากการวิเคราะห์จากสมการพีชคณิตบูลีนที่ได้จากขั้นตอนที่ 1

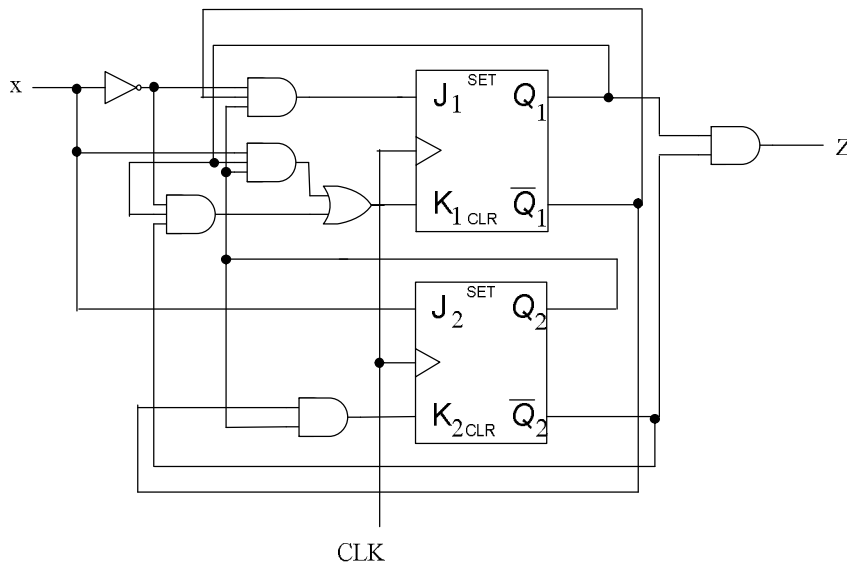
ขั้นตอนที่ 3: นำผลลัพธ์ที่ได้แต่ละช่องของแผนผังคาร์โนห์ไปใส่ลงในตารางเอ็กซ์เทชัน

ขั้นตอนที่ 4: นำตารางเอ็กซ์เทชันที่ได้จากขั้นตอนที่ 3 มาใช้สำหรับการวิเคราะห์หาตารางทรานซิชัน

ขั้นตอนที่ 5: นำตารางทรานซิชันมาสร้างตารางสถานะและแผนภาพสถานะ

ขั้นตอนที่ 6: วิเคราะห์หาสถานะที่เกิดขึ้นใหม่ และเอาต์พุตของวงจร จากลำดับของอินพุต และสถานะปัจจุบันของวงจร

ตัวอย่างที่ 8-2 จากวงจรเชิงลำดับที่กำหนดให้จงวิเคราะห์หาเอาต์พุต และสถานะที่เกิดขึ้น กรณีที่อินพุตที่ถูกป้อนเข้ามาอย่างเป็นลำดับมีค่าเป็น 10010 และสถานะเริ่มต้นของฟลิปฟล็อปทั้ง 2 ตัวมีค่าเป็น 00 ($Q_1 = 0$ และ $Q_2 = 0$)



วิธีทำ เนื่องจากวงจรนี้ใช้สัญญาณ CLK ร่วมกัน จึงเป็นวงจรแบบเข้าจังหวะ ขั้นตอนการวิเคราะห์เป็นดังนี้

ขั้นตอนที่ 1: หาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทุกตัว และเอาต์พุตวงจรได้ดังนี้

$$J_1 = X\bar{Q}_1Q_2$$

$$K_1 = XQ_1Q_2 + XQ_1\bar{Q}_2$$

$$J_2 = X$$

$$K_2 = \bar{Q}_1 Q_2$$

$$Z = Q_1 \bar{Q}_2$$

ขั้นตอนที่ 2: สร้างแผนผังคาร์โนห์ให้อินพุตของฟลิปฟล็อปทุกตัว และ เอาต์พุตของวงจรถัดนี้

1. สร้างแผนผังคาร์โนห์ $J_1 = X\bar{Q}_1 Q_2$ โดย J_1 จะมีค่าเป็น 1 ก็ต่อเมื่อ $X = 0$, $Q_1 = 0$ และ $Q_2 = 1$ สำหรับกรณีอื่นๆ J_1 จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1 Q_2$ | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

$$J_1 = X\bar{Q}_1 Q_2$$

2. สร้างแผนผังคาร์โนห์ $K_1 = XQ_1 Q_2 + XQ_1 \bar{Q}_2$ โดย K_1 จะมีค่าเป็น 1 ก็ต่อเมื่อ $X = 1$, $Q_1 = 1$ และ $Q_2 = 1$ หรือ $X = 0$, $Q_1 = 1$ และ $Q_2 = 0$ สำหรับกรณีอื่นๆ K_1 จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1 Q_2$ | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |

$$K_1 = XQ_1 Q_2 + XQ_1 \bar{Q}_2$$

3. สร้างแผนผังคาร์โนห์ $J_2 = X$ โดย J_2 จะมีค่าเป็น 1 ก็ต่อเมื่อ $X = 1$ สำหรับกรณีอื่นๆ J_2 จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1 Q_2$ | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

$$J_2 = X$$

4. สร้างแผนผังคาร์โนห์ $K_2 = \bar{Q}_1 Q_2$ โดย K_2 จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1 = 0$ และ $Q_2 = 1$ สำหรับกรณีอื่นๆ K_2 จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1 Q_2$ | 00 | 01 | 11 | 10 |
|------------------------|----|----|----|----|
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |

$$K_2 = \bar{Q}_1 Q_2$$

5. สร้างแผนผังคาร์โนท์ $Z = Q_1\bar{Q}_2$ โดย Z จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1=1$ และ $Q_2 = 0$ สำหรับกรณีอื่นๆ Z จะมีค่าเป็น 0 ทั้งหมด

| | | | | | |
|-----|----------|----|----|----|----|
| | Q_1Q_2 | 00 | 01 | 11 | 10 |
| X | 0 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | 1 |

$$Z = Q_1\bar{Q}_2$$

ขั้นตอนที่ 3: สร้างตารางเอ็กไซเทชัน เนื่องจากฟลิปฟล็อปทั้ง 2 ตัวเป็นฟลิปฟล็อปแบบ JK ข้อมูลที่จะนำมาใส่ตารางเอ็กไซเทชันแต่ละคอลัมน์เป็นดังนี้

คอลัมน์ที่ 1 ($Q_1=0, Q_2=0$)

เมื่อ $X=0$ จะได้ $J_1=0, K_1=0, J_2=0, K_2=0$ และ $Z=0$

เมื่อ $X=1$ จะได้ $J_1=0, K_1=0, J_2=1, K_2=0$ และ $Z=0$

คอลัมน์ที่ 2 ($Q_1=0, Q_2=1$)

เมื่อ $X=0$ จะได้ $J_1=1, K_1=0, J_2=0, K_2=1$ และ $Z=0$

เมื่อ $X=1$ จะได้ $J_1=0, K_1=0, J_2=1, K_2=1$ และ $Z=0$

คอลัมน์ที่ 3 ($Q_1=1, Q_2=1$)

เมื่อ $X=0$ จะได้ $J_1=0, K_1=0, J_2=0, K_2=0$ และ $Z=0$

เมื่อ $X=1$ จะได้ $J_1=0, K_1=1, J_2=1, K_2=0$ และ $Z=0$

คอลัมน์ที่ 4 ($Q_1=1, Q_2=0$)

เมื่อ $X=0$ จะได้ $J_1=0, K_1=1, J_2=0, K_2=0$ และ $Z=1$

เมื่อ $X=1$ จะได้ $J_1=0, K_1=0, J_2=1, K_2=0$ และ $Z=1$

จึงได้ตารางเอ็กไซเทชันเป็นดังนี้

| สถานะปัจจุบัน Q_1Q_2 | $X=0$ | | $X=1$ | | Z | |
|---------------------------|----------|----------|----------|----------|-------|-------|
| | J_1K_1 | J_2K_2 | J_1K_1 | J_2K_2 | $X=0$ | $X=1$ |
| 00 | 00 | 00 | 00 | 10 | 0 | 0 |
| 01 | 10 | 01 | 00 | 11 | 0 | 0 |
| 10 | 01 | 00 | 00 | 10 | 1 | 1 |
| 11 | 00 | 00 | 01 | 10 | 0 | 0 |

ขั้นตอนที่ 4: นำผลลัพธ์จากตารางเอ็กไซเทชันมาสร้างตารางทรานซิชันเพื่อหาสถานะที่เกิดขึ้นใหม่ของ Q (Q_{next}) ดังนี้

แถวที่ 1 ($Q_1 = 0, Q_2 = 0$)

เมื่อ $X = 0$ กำหนด $J_1 = 0, K_1 = 0$ ได้ $Q_{next1} = 0$ และ กำหนด $J_2 = 0, K_2 = 0$ ได้ $Q_{next2} = 0$

เมื่อ $X = 1$ กำหนด $J_1 = 0, K_1 = 0$ ได้ $Q_{next1} = 0$ และ กำหนด $J_2 = 1, K_2 = 0$ ได้ $Q_{next2} = 1$

แถวที่ 2 ($Q_1 = 0, Q_2 = 1$)

เมื่อ $X = 0$ กำหนด $J_1 = 1, K_1 = 0$ ได้ $Q_{next1} = 1$ และ กำหนด $J_2 = 0, K_2 = 1$ ได้ $Q_{next2} = 0$

เมื่อ $X = 1$ กำหนด $J_1 = 0, K_1 = 0$ ได้ $Q_{next1} = 0$ และ กำหนด $J_2 = 1, K_2 = 1$ ได้ $Q_{next2} = 0$

แถวที่ 3 ($Q_1 = 1, Q_2 = 0$)

เมื่อ $X = 0$ กำหนด $J_1 = 0, K_1 = 1$ ได้ $Q_{next1} = 0$ และ กำหนด $J_2 = 0, K_2 = 0$ ได้ $Q_{next2} = 0$

เมื่อ $X = 1$ กำหนด $J_1 = 0, K_1 = 0$ ได้ $Q_{next1} = 1$ และ กำหนด $J_2 = 1, K_2 = 0$ ได้ $Q_{next2} = 1$

แถวที่ 4 ($Q_1 = 1, Q_2 = 1$)

เมื่อ $X = 0$ กำหนด $J_1 = 0, K_1 = 0$ ได้ $Q_{next1} = 1$ และ กำหนด $J_2 = 0, K_2 = 0$ ได้ $Q_{next2} = 1$

เมื่อ $X = 1$ กำหนด $J_1 = 0, K_1 = 1$ ได้ $Q_{next1} = 0$ และ กำหนด $J_2 = 1, K_2 = 0$ ได้ $Q_{next2} = 1$

จึงได้ตารางทรานซิชันเป็นดังนี้

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่ | | Z | |
|---------------------------|----------------------|-------|-------|-------|
| | X = 0 | X = 1 | X = 0 | X = 1 |
| 00 | 00 | 01 | 0 | 0 |
| 01 | 10 | 00 | 0 | 0 |
| 10 | 00 | 11 | 1 | 1 |
| 11 | 11 | 01 | 0 | 0 |

ขั้นตอนที่ 5 สร้างตารางสถานะ และ แผนภาพสถานะจากตารางทรานซิชัน

กำหนดให้ A แทนสถานะ 00

B แทนสถานะ 01

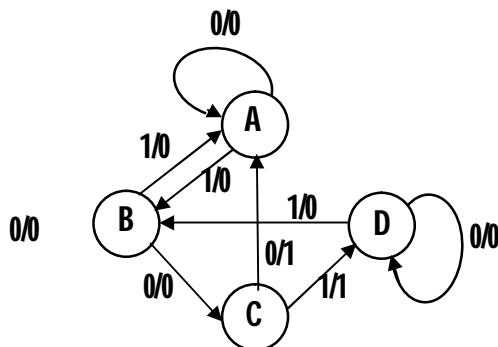
C แทนสถานะ 10

D แทนสถานะ 11

ตารางสถานะเกิดจากการแทนค่าสถานะของ Q_1 และ Q_2 ด้วยตัวอักษรที่กำหนดข้างต้น ได้เป็นดังนี้

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/เอาต์พุต (Q_{next}/Z) | |
|---------------------------|---|-------|
| | X = 0 | X = 1 |
| A | A/0 | B/0 |
| B | C/0 | A/0 |
| C | A/1 | D/1 |
| D | D/0 | B/0 |

จากตารางสถานะ สามารถแปลงเป็นแผนภาพสถานะได้ดังนี้



จากตารางสถานะ หรือแผนภาพสถานะสามารถหาเอาต์พุต และสถานะที่เกิดขึ้นได้ดังนี้ (โดยสถานะเริ่มต้นคือ A และลำดับของข้อมูลอินพุตที่ป้อนเข้ามาคือ 10010)

| | | | | | |
|-----------------------|---|---|---|---|---|
| อินพุต: | 1 | 0 | 0 | 1 | 0 |
| สถานะปัจจุบัน: | A | B | C | A | B |
| สถานะที่เกิดขึ้นใหม่: | B | C | A | B | C |
| เอาต์พุต: | 0 | 0 | 1 | 0 | 0 |

8.2.2 วิธีการวิเคราะห์วงจรเชิงลำดับแบบไม่เข้าจังหวะ

การวิเคราะห์วงจรเชิงลำดับแบบไม่เข้าจังหวะ ใช้สำหรับการวิเคราะห์วงจรเชิงลำดับในกรณีทีฟลิปฟลอปอย่างน้อย 1 ตัวที่ไม่ใช้สัญญาณนาฬิกาพร้อมกับฟลิปฟลอปตัวอื่นมีอยู่ในวงจร นอกจากการวิเคราะห์วงจรแบบไม่เข้าจังหวะแล้ววิธีที่นำเสนอนี้ยังสามารถใช้สำหรับการวิเคราะห์วงจรเชิงลำดับแบบเข้าจังหวะได้เช่นกัน ซึ่งมีขั้นตอนเป็นดังนี้

ขั้นตอนที่ 1: หาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟลอปทุกตัว และเอาต์พุตวงจร โดยที่หากเป็นวงจรเชิงลำดับเป็นวงจรแบบไม่เข้าจังหวะ จะต้องหาสมการบูลีนของสัญญาณนาฬิกาของฟลิปฟลอปแต่ละตัวที่อยู่ภายในวงจรด้วย

ขั้นตอนที่ 2: นำสมการพีชคณิตบูลีนของอินพุตทุกตัวที่หาได้ในขั้นตอนที่ 1 มาแทนค่าลงในสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟลอป

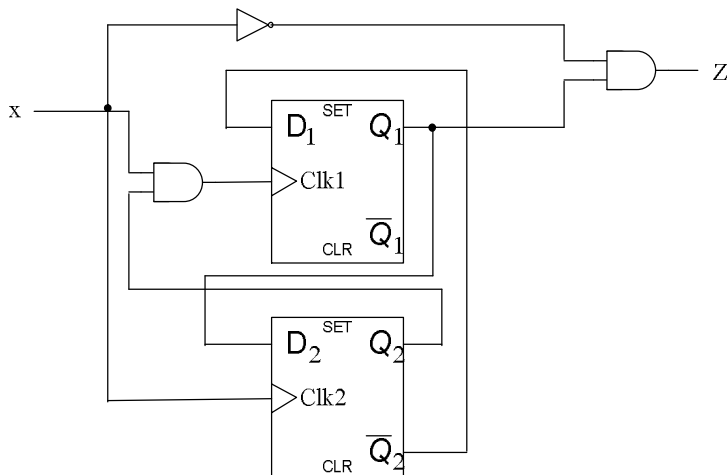
ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์ของเอาต์พุตใหม่ของฟลิปฟลอปทั้งหมด และเอาต์พุตวงจร โดยที่แผนผังคาร์โนห์ 1 แผนผังใช้สำหรับแสดงสถานะที่เป็นไปได้ทั้งหมดของเอาต์พุตใหม่ของฟลิปฟลอปหรือเอาต์พุตของวงจร 1 ตัว สำหรับค่าผลลัพธ์ที่ได้แต่ละช่องในแผนผังคาร์โนห์ได้มาจากการวิเคราะห์จากสมการพีชคณิตบูลีนที่ได้จากขั้นตอนที่ 2

ขั้นตอนที่ 4: นำผลลัพธ์ที่ได้แต่ละช่องของแผนผังคาร์โนห์ไปใส่ลงในตารางทรานซิชัน

ขั้นตอนที่ 5: นำตารางทรานซิชันมาสร้างตารางสถานะและแผนภาพสถานะ

ขั้นตอนที่ 6: วิเคราะห์หาสถานะที่เกิดขึ้นใหม่ และเอาต์พุตของวงจร จากลำดับของอินพุต และสถานะปัจจุบันของวงจร

ตัวอย่างที่ 8-3 จากวงจรเชิงลำดับที่กำหนดให้จงวิเคราะห์หาเอาต์พุต และสถานะที่เกิดขึ้น กรณีที่อินพุตที่ถูกป้อนเข้ามาอย่างเป็นลำดับมีค่าเป็น **11000** และสถานะเริ่มต้นของฟลิปฟล็อปทั้ง 2 ตัวมีค่าเป็น **01** ($Q_1 = 0$ และ $Q_2 = 1$)



วิธีทำ เนื่องจากวงจรนี้ไม่ได้ใช้สัญญาณ CLK ร่วมกัน จึงเป็นวงจรแบบไม่เข้าจังหวะ ขั้นตอนการวิเคราะห์เป็นดังนี้

ขั้นตอนที่ 1: หาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทุกตัวรวมถึงสัญญาณนาฬิกา และเอาต์พุตวงจรได้ดังนี้

$$D_1 = \bar{Q}_2$$

$$D_2 = Q_1$$

$$Clk_1 = XQ_2$$

$$Clk_2 = X$$

$$Z = XQ_1$$

ขั้นตอนที่ 2: วงจรเชิงลำดับที่กำหนด มีการใช้งานฟลิปฟล็อปแบบ D เท่านั้น โดยสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D เป็นคือ

$$Q_{next} = DCIK + Q\bar{C}IK$$

กำหนดให้ Q_{next1} และ Q_{next2} คือสถานะเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D ตัวที่ 1 และ 2 ตามลำดับจึงได้สมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D_1 และ D_2 เป็นดังนี้

$$Q_{next1} = D_1Clk_1 + Q_1\bar{C}Ik_1$$

$$= \bar{Q}_2XQ_2 + Q_1X\bar{Q}_2$$

$$= Q_1X + Q_1\bar{Q}_2$$

$$Q_{next2} = D_2Clk_2 + Q_2\bar{C}Ik_2$$

$$= Q_1X + Q_2\bar{X}$$

ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์ของเอาต์พุตใหม่ของฟลิปฟล็อปทั้งหมด และเอาต์พุตวงจร

1. สร้างแผนผังคาร์โนห์ $Q_{next1} = Q_1\bar{X} + Q_1\bar{Q}_2$ โดย Q_{next1} จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1 = 1$ และ $X = 0$ หรือ $Q_1 = 1$ และ $Q_2 = 0$ สำหรับกรณีอื่นๆ Q_{next1} จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1Q_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |

$$Q_{next1} = Q_1\bar{X} + Q_1\bar{Q}_2$$

2. สร้างแผนผังคาร์โนห์ $Q_{next2} = Q_1X + Q_2X$ โดย Q_{next2} จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1 = 1$ และ $X = 1$ หรือ $Q_2 = 1$ และ $X = 0$ สำหรับกรณีอื่นๆ Q_{next2} จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1Q_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |

$$Q_{next2} = Q_1X + Q_2X$$

3. สร้างแผนผังคาร์โนห์ $Z = XQ_1$ โดย Z จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1 = 1$ และ $X = 0$ สำหรับกรณีอื่นๆ Z จะมีค่าเป็น 0 ทั้งหมด

| $X \backslash Q_1Q_2$ | 00 | 01 | 11 | 10 |
|-----------------------|----|----|----|----|
| 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |

$$Z = XQ_1$$

ขั้นตอนที่ 4: นำผลลัพธ์ที่ได้แต่ละช่องของแผนผังคาร์โนห์ไปใส่ลงในตารางทรานซิชัน ซึ่งพิจารณาเป็น 4 คอลัมน์ดังนี้

คอลัมน์ที่ 1: ($Q_1 = 0$ และ $Q_2 = 0$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 0$, $Q_{next2} = 0$ และ $Z = 0$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0$, $Q_{next2} = 0$ และ $Z = 0$

คอลัมน์ที่ 2: ($Q_1 = 0$ และ $Q_2 = 1$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 0$, $Q_{next2} = 1$ และ $Z = 0$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0$, $Q_{next2} = 0$ และ $Z = 0$

คอลัมน์ที่ 3: ($Q_1 = 1$ และ $Q_2 = 1$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 1, Q_{next2} = 1$ และ $Z = 1$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0, Q_{next2} = 1$ และ $Z = 0$

คอลัมน์ที่ 4: ($Q_1 = 1$ และ $Q_2 = 0$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 1, Q_{next2} = 0$ และ $Z = 1$

เมื่อ $X = 1$ ได้ $Q_{next1} = 1, Q_{next2} = 1$ และ $Z = 0$

จากทั้ง 4 คอลัมน์ได้ตารางทรานซิชันเป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่ (Q_{next}) | | เอาต์พุต (Z) | |
|-------------------------------|-------------------------------------|---------|------------------|---------|
| | $X = 0$ | $X = 1$ | $X = 0$ | $X = 1$ |
| 00 | 00 | 00 | 0 | 0 |
| 01 | 01 | 00 | 0 | 0 |
| 10 | 10 | 11 | 1 | 0 |
| 11 | 11 | 01 | 1 | 0 |

ขั้นตอนที่ 5 สร้างตารางสถานะ และ แผนภาพสถานะจากตารางทรานซิชัน

กำหนดให้ A แทนสถานะ 00

B แทนสถานะ 01

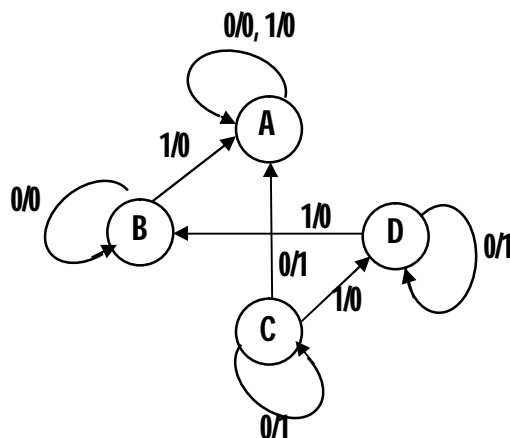
C แทนสถานะ 10

D แทนสถานะ 11

ตารางสถานะเกิดจากการแทนค่าสถานะของ Q_1 และ Q_2 ด้วยตัวอักษรที่กำหนดข้างต้น ได้เป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่/เอาต์พุต (Q_{next}/Z) | |
|-------------------------------|---|---------|
| | $X = 0$ | $X = 1$ |
| A | A/0 | A/0 |
| B | B/0 | A/0 |
| C | C/1 | D/0 |
| D | D/1 | B/0 |

จากตารางสถานะ สามารถแปลงเป็นแผนภาพสถานะได้ดังนี้



จากตารางสถานะ หรือแผนภาพสถานะสามารถหาเอาต์พุต และสถานะที่เกิดขึ้นได้ดังนี้ (โดยสถานะเริ่มต้นคือ B และลำดับของข้อมูลอินพุตที่ป้อนเข้ามาคือ 11000)

| | | | | | |
|-----------------------|---|---|---|---|---|
| อินพุต: | 1 | 1 | 0 | 0 | 0 |
| สถานะปัจจุบัน: | B | A | A | A | A |
| สถานะที่เกิดขึ้นใหม่: | A | A | A | A | A |
| เอาต์พุต: | 0 | 0 | 0 | 0 | 0 |

จากตารางสถานะ หรือแผนภาพสถานะสังเกตได้ว่า หากสถานะปัจจุบันอยู่ที่ A แล้ว สถานะใหม่ที่จะเกิดขึ้นยังคงอยู่ที่ A เสมอ และไม่มีโอกาสที่จะเกิดสถานะอื่นได้อีกเลย ไม่ว่าค่าอินพุต X ที่ป้อนเข้ามาจะมีสถานะเป็นเช่นไรก็ตาม เรียกสภาวะที่เกิดขึ้นนี้ว่า “วงจรรอ” (Deadlock)

ตัวอย่างที่ 8-4 จากตัวอย่างที่ 8-2 ทดลองหาตารางสถานะ และแผนภาพสถานะใหม่อีกครั้งโดยใช้วิธีการที่ใช้สำหรับวิเคราะห์วงจรรหัสลำดับแบบไม่เข้าจังหวะ

วิธีทำ เนื่องจากวงจรรหัสใช้สัญญาณ CLK ร่วมกัน จึงเป็นวงจรรหัสแบบเข้าจังหวะ ขั้นตอนการวิเคราะห์เป็นดังนี้

ขั้นตอนที่ 1: หาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทุกตัว และเอาต์พุตวงจรรหัสได้ดังนี้

$$\begin{aligned}
 J_1 &= X\bar{Q}_1Q_2 \\
 K_1 &= XQ_1Q_2 + XQ_1\bar{Q}_2 \\
 J_2 &= X \\
 K_2 &= \bar{Q}_1Q_2 \\
 Z &= Q_1\bar{Q}_2
 \end{aligned}$$

ขั้นตอนที่ 2: วงจรรหัสลำดับที่กำหนด มีการใช้งานฟลิปฟล็อปแบบ JK เท่านั้น โดยสมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ JK เป็นคือ

$$Q_{next} = J\bar{Q} + KQ$$

กำหนดให้ Q_{next1} และ Q_{next2} คือสถานะเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D ตัวที่ 1 และ 2 ตามลำดับจึงได้สมการพีชคณิตบูลีนของเอาต์พุตใหม่ของฟลิปฟล็อปแบบ D_1 และ D_2 เป็นดังนี้

$$\begin{aligned}
 Q_{next1} &= J_1 Q_1 + K_1 \bar{Q}_1 \\
 &= X Q_1 Q_2 + \overline{(X Q_1 Q_2 + X Q_1 Q_2)} Q_1 \\
 &= X Q_1 Q_2 + (\overline{X Q_1 Q_2} \overline{X Q_1 Q_2}) Q_1 \\
 &= X Q_1 Q_2 + (X + \bar{Q}_1 + \bar{Q}_2)(X + Q_1 + Q_2) Q_1 \\
 &= X Q_1 Q_2 + (X + \bar{Q}_1 + \bar{Q}_2)(X Q_1 + Q_1 Q_2) \\
 &= X Q_1 Q_2 + X Q_1 Q_2 + X Q_1 Q_2 \\
 &= X Q_2 + X Q_1 Q_2
 \end{aligned}$$

$$\begin{aligned}
 Q_{next2} &= J_2 Q_2 + K_2 \bar{Q}_2 \\
 &= X Q_2 + \overline{Q_1} \bar{Q}_2 Q_2 \\
 &= X Q_2 + (Q_1 + \bar{Q}_2) Q_2 \\
 &= X Q_2 + Q_1 Q_2
 \end{aligned}$$

ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์ของเอาต์พุตใหม่ของฟลิปฟล็อปทั้งหมด และเอาต์พุตวงจร

- สร้างแผนผังคาร์โนห์ $Q_{next1} = X Q_2 + X Q_1 Q_2$ โดย Q_{next1} จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_2 = 1$ และ $X = 0$ หรือ $Q_1 = 1$ $Q_2 = 0$ และ $X = 1$ สำหรับกรณีอื่นๆ Q_{next1} จะมีค่าเป็น 0 ทั้งหมด

| | | | | | |
|-----|-----------|----|----|----|----|
| | $Q_1 Q_2$ | 00 | 01 | 11 | 10 |
| X | 0 | 0 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 0 | 1 |

$$Q_{next1} = X Q_2 + X Q_1 Q_2$$

- สร้างแผนผังคาร์โนห์ $Q_{next2} = X Q_2 + Q_1 Q_2$ โดย Q_{next2} จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_2 = 0$ และ $X = 1$ หรือ $Q_1 = 1$ และ $Q_2 = 1$ สำหรับกรณีอื่นๆ Q_{next2} จะมีค่าเป็น 0 ทั้งหมด

| | | | | | |
|-----|-----------|----|----|----|----|
| | $Q_1 Q_2$ | 00 | 01 | 11 | 10 |
| X | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 0 | 1 | 1 |

$$Q_{next2} = X Q_2 + Q_1 Q_2$$

- สร้างแผนผังคาร์โนห์ $Z = Q_1 \bar{Q}_2$ โดย Z จะมีค่าเป็น 1 ก็ต่อเมื่อ $Q_1 = 1$ และ $Q_2 = 0$ สำหรับกรณีอื่นๆ Z จะมีค่าเป็น 0 ทั้งหมด

| | | | | | |
|-----|-----------|----|----|----|----|
| | $Q_1 Q_2$ | 00 | 01 | 11 | 10 |
| X | 0 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | 1 |

$$Z = Q_1 \bar{Q}_2$$

ขั้นตอนที่ 4: นำผลลัพธ์ที่ได้แต่ละช่องของแผนผังคาร์โนห์ไปใส่ลงในตารางทรานซิชัน ซึ่งพิจารณาเป็น 4 คอลัมน์ดังนี้

คอลัมน์ที่ 1: ($Q_1 = 0$ และ $Q_2 = 0$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 0, Q_{next2} = 0$ และ $Z = 0$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0, Q_{next2} = 1$ และ $Z = 0$

คอลัมน์ที่ 2: ($Q_1 = 0$ และ $Q_2 = 1$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 1, Q_{next2} = 0$ และ $Z = 0$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0, Q_{next2} = 0$ และ $Z = 0$

คอลัมน์ที่ 3: ($Q_1 = 1$ และ $Q_2 = 1$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 1, Q_{next2} = 1$ และ $Z = 0$

เมื่อ $X = 1$ ได้ $Q_{next1} = 0, Q_{next2} = 1$ และ $Z = 0$

คอลัมน์ที่ 4: ($Q_1 = 1$ และ $Q_2 = 0$)

เมื่อ $X = 0$ ได้ $Q_{next1} = 0, Q_{next2} = 0$ และ $Z = 1$

เมื่อ $X = 1$ ได้ $Q_{next1} = 1, Q_{next2} = 1$ และ $Z = 1$

จากทั้ง 4 คอลัมน์ได้ตารางทรานซิชันเป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่ (Q_{next}) | | เอาต์พุต (Z) | |
|-------------------------------|--|---------|------------------|---------|
| | $X = 0$ | $X = 1$ | $X = 0$ | $X = 1$ |
| 00 | 00 | 01 | 0 | 0 |
| 01 | 10 | 00 | 0 | 0 |
| 10 | 00 | 11 | 1 | 1 |
| 11 | 11 | 01 | 0 | 0 |

ขั้นตอนที่ 5 สร้างตารางสถานะ และ แผนภาพสถานะจากตารางทรานซิชัน

กำหนดให้ A แทนสถานะ 00

B แทนสถานะ 01

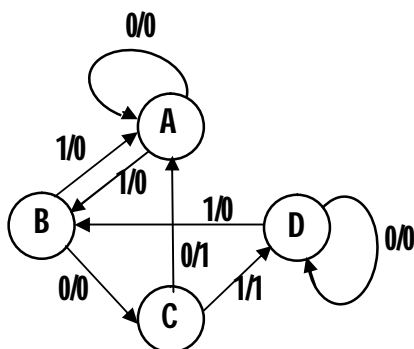
C แทนสถานะ 10

D แทนสถานะ 11

ตารางสถานะเกิดจากการแทนค่าสถานะของ Q_1 และ Q_2 ด้วยตัวอักษรที่กำหนดข้างต้น ได้เป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่/เอาต์พุต (Q_{next}/Z) | |
|-------------------------------|---|---------|
| | $X = 0$ | $X = 1$ |
| A | A/0 | B/0 |
| B | C/0 | A/0 |
| C | A/1 | D/1 |
| D | D/0 | B/0 |

จากตารางสถานะ สามารถแปลงเป็นแผนภาพสถานะได้ดังนี้



จากตารางสถานะ และแผนภาพสถานะที่คำนวณได้ต่างตัวอย่างที่ 8-4 มีค่าตรงกับตารางสถานะ และแผนภาพสถานะที่คำนวณได้จากตัวอย่างที่ 8-2 ทุกประการ

8.3 การออกแบบวงจรเชิงลำดับ

การออกแบบวงจรเชิงลำดับ คือการสร้างวงจรเชิงลำดับขึ้นมาใหม่เพื่อให้ได้ผลการทำงานตามที่ผู้ใช้งานต้องการ ดังนั้นก่อนที่จะสร้างวงจรขึ้นมานั้น จำเป็นต้องกำหนดผลลัพธ์ที่เป็นไปได้ทั้งหมดของวงจรขึ้นมาก่อนแล้วจึงทำการออกแบบ และสร้างวงจรที่ให้ผลลัพธ์ออกมาตรงกับที่ได้กำหนดไว้ โดยขั้นตอนการสร้างวงจรเชิงลำดับมีดังนี้

ขั้นตอนที่ 1: วิเคราะห์ และออกแบบตารางสถานะ และแผนภาพสถานะซึ่งแสดงถึงสถานะของเอาต์พุตของฟลิปฟล็อปที่ให้ผลลัพธ์ตรงกับที่ผู้ใช้งานต้องการ

ขั้นตอนที่ 2: สร้างตารางทรานซิชันจากตารางสถานะที่ออกแบบได้จากขั้นตอนที่ 1

ขั้นตอนที่ 3: จากขั้นตอนที่ 1 เมื่อทราบจำนวนของสถานะของเอาต์พุตของฟลิปฟล็อปที่เป็นไปได้ทั้งหมดแล้ว จะทำให้สามารถคำนวณหาจำนวนของฟลิปฟล็อปทั้งหมดที่ต้องนำมาใช้งานในวงจรได้ (จำนวนสถานะทั้งหมด = $2^{\text{จำนวนฟลิปฟล็อป}}$) ให้เลือกชนิดของฟลิปฟล็อปที่จะนำมาใช้งานและสร้างตารางเอ็ชเทชันจากความสัมพันธ์ระหว่างตารางทรานซิชันและ ชนิดของฟลิปฟล็อปที่เลือกใช้งาน

ขั้นตอนที่ 4: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปแต่ละตัว และเอาต์พุต

ขั้นตอนที่ 5: สร้างวงจรจากสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปและเอาต์พุต

ตัวอย่างที่ 8-5 จงออกแบบวงจรรวมเชิงลำดับโดยที่ค่าเอาต์พุตจะมีค่าเป็น 1 ได้ก็ต่อเมื่ออินพุตที่ป้อนเข้ามามีค่าเป็นลำดับที่ต่อเนื่องคือ "101" สำหรับกรณีอื่นๆ จะทำให้เอาต์พุตมีค่าเป็น 0 ทั้งหมด โดยให้ใช้ฟลิปฟล็อปแบบ RS

วิธีทำขั้นตอนที่ 1: จากโจทย์ที่กำหนดมีเพียง 1 อินพุตจึงสามารถวิเคราะห์ตารางสถานะได้ดังนี้

กำหนดให้ X คือข้อมูลอินพุต

A แทนสถานะเริ่มต้น ดังนั้น A จึงแทนสถานะของข้อมูลอินพุตที่ป้อนเข้ามายังไม่ถูกเลย

B แทนสถานะที่เกิดขึ้นกรณีที่อินพุตถูกป้อนมาถูกแล้ว 1 ค่า (1)

C แทนสถานะที่เกิดขึ้นกรณีที่อินพุตถูกป้อนมาถูกแล้ว 2 ค่า (10)

จากตัวอย่างสังเกตว่าไม่จำเป็นต้องกำหนดสถานะใหม่ที่เกิดขึ้นกรณีอื่นพุทถูกป้อนมาถูกแล้ว 3 ค่า เนื่องจากสถานะสุดท้ายของรูปแบบ "101" คือ "1" ซึ่งพบว่าเป็นสถานะที่เกิดจากการป้อนอินพุตที่ถูกรูปแบบแล้ว 1 ค่าของกรณีที่จะเกิดเอาต์พุตที่มีค่าเป็น 1 ใหม่อีกครั้ง ดังตัวอย่างต่อไปนี้

| | | | | | |
|-----------|---|---|---|---|---|
| ลำดับที่: | 1 | 2 | 3 | 4 | 5 |
| อินพุต: | 1 | 0 | 1 | 0 | 1 |
| เอาต์พุต: | 0 | 0 | 1 | 0 | 1 |

จากตัวอย่างข้างต้น ลำดับที่ 3 คือกรณีที่เกิดจากการใส่ข้อมูลอินพุตในรูปแบบที่ถูกต้อง ("101") ซึ่งทำให้เอาต์พุตที่ได้มีค่าเป็น 1 อย่างไรก็ตามเอาต์พุตสามารถมีสถานะเป็น 1 ได้อีกครั้งในลำดับที่ 5 ซึ่งเกิดจากการป้อนอินพุตที่ต่อจากลำดับที่ 3 ที่ถูกรูปแบบอีกเพียง 2 ค่าดังนั้นสรุปได้ว่าสถานะของลำดับที่ 3 คือสถานะที่เกิดจากการกรณีใส่ข้อมูลอินพุตถูกรูปแบบแล้ว 1 ค่า

ดังนั้นสถานะที่เป็นไปได้มีทั้งหมด 3 สถานะประกอบด้วย A, B และ C การวิเคราะห์หาสถานะที่เกิดขึ้นใหม่สามารถทำได้ดังนี้

กรณีที่ สถานะปัจจุบันคือ A

เมื่อ $X = 0$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามายังไม่ถูกรูปแบบเลย เนื่องจาก A เป็นสถานะที่ยังไม่ถูกรูปแบบเลย ดังนั้นสถานะที่เกิดขึ้นใหม่ยังคงอยู่ที่ A และ เอาต์พุตมีค่าเป็น 0

เมื่อ $X = 1$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามามีความถูกต้องแล้ว 1 ค่าเนื่องจาก B เป็นสถานะที่เกิดจากการป้อนข้อมูลอินพุตถูกรูปแบบแล้ว 1 ค่า ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ B และ เอาต์พุตมีค่าเป็น 0

กรณีที่ สถานะปัจจุบันคือ B

เมื่อ $X = 0$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาเข้ามาที่มีความถูกต้องแล้ว 2 ค่าเนื่องจาก C เป็นสถานะที่เกิดจากการป้อนข้อมูลอินพุตถูกรูปแบบแล้ว 2 ค่า ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ C และ เอาต์พุตมีค่าเป็น 0

เมื่อ $X = 1$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามายังคงมีความถูกต้องอยู่ 1 ค่าเช่นเดิมดังนั้นสถานะที่เกิดขึ้นใหม่ยังคงอยู่ที่ B และ เอาต์พุตมีค่าเป็น 0

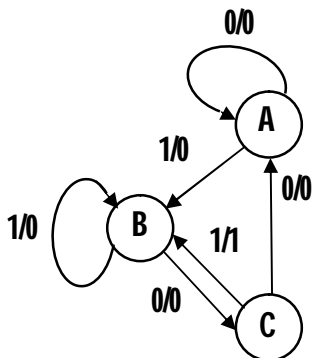
กรณีที่ สถานะปัจจุบันคือ C

เมื่อ $X = 0$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาแล้วนั้นไม่ถูกรูปแบบเลย ดังนั้นสถานะที่เกิดขึ้นใหม่จะต้องกลับมาอยู่ที่ A และ เอาต์พุตมีค่าเป็น 0

เมื่อ $X = 1$ แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาเข้ามาที่มีความถูกต้องแล้ว 3 ค่าทำให้เอาต์พุตมีค่าเป็น 1 แต่อย่างไรก็ตามเนื่องจากสถานะล่าสุดของมีค่าเป็น 1 จึงเปรียบเสมือนว่าเป็นการป้อนข้อมูลอินพุตมาถูกแล้ว 1 ค่า ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ B

จากทั้ง 3 สถานะที่วิเคราะห์ได้ทั้งหมด สามารถสร้างแผนภาพสถานะ และตารางสถานะได้ดังนี้

แผนภาพสถานะ:



ตารางสถานะ:

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|-------|
| | X = 0 | X = 1 |
| A | A/0 | B/0 |
| B | C/0 | B/0 |
| C | A/0 | B/1 |

ขั้นตอนที่ 2: สร้างตารางทรานซิชันจากตารางสถานะได้ดังนี้

กำหนดให้ A แทนสถานะ 00

B แทนสถานะ 01

C แทนสถานะ 10

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่/เอาต์พุต (Q_{next}/Z) | |
|-------------------------------|---|-------|
| | X = 0 | X = 1 |
| 00 | 00/0 | 01/0 |
| 01 | 10/0 | 01/0 |
| 10 | 00/0 | 01/1 |

ขั้นตอนที่ 3: สร้างตารางเอ็กซ์เทชันจากตารางทรานซิชัน

เนื่องจากสถานะที่เป็นไปได้ทั้งหมดของเอาต์พุตของฟลิปฟลอปมี 3 สถานะ หากใช้ฟลิปฟลอป 1 ตัวจะได้ 2 สถานะซึ่งไม่เพียงพอ ดังนั้นจึงต้องใช้ฟลิปฟลอป 2 ตัวซึ่งจะได้ทั้งหมด 4 สถานะ อย่างไรก็ตามจากตัวอย่างต้องการเพียงแค่ 3 สถานะคือ 00, 01 และ 10 ดังนั้นสถานะ 11 ถึงเป็นสถานะที่ไม่สนใจ

จากตารางเอ็กซ์เทชัน ของฟลิปฟลอปแบบ RS ได้ว่า

แถวที่ 1 ($Q_1 = 0, Q_2 = 0$)

เมื่อ X = 0 ต้องการ $Q_{next1} = 0$ จะได้ $R_1 = X$ และ $S_1 = 0$ และ ต้องการ $Q_{next2} = 0$ จะได้ $R_2 = X$ และ $S_2 = 0$

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $R_1 = x$ และ $S_1 = 0$ และ ต้องการ $Q_{next2} = 1$ จะได้ $R_2 = 0$ และ $S_2 = 1$

แถวที่ 2 ($Q_1 = 0, Q_2 = 1$)

เมื่อ $X = 0$ ต้องการ $Q_{next1} = 1$ จะได้ $R_1 = 0$ และ $S_1 = 1$ และ ต้องการ $Q_{next2} = 0$ จะได้ $R_2 = 1$ และ $S_2 = 0$

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $R_1 = x$ และ $S_1 = 0$ และ ต้องการ $Q_{next2} = 1$ จะได้ $R_2 = 0$ และ $S_2 = x$

แถวที่ 3 ($Q_1 = 1, Q_2 = 0$)

เมื่อ $X = 0$ ต้องการ $Q_{next1} = 0$ จะได้ $R_1 = 1$ และ $S_1 = 0$ และ ต้องการ $Q_{next2} = 0$ จะได้ $R_2 = x$ และ $S_2 = 0$

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $R_1 = 1$ และ $S_1 = 0$ และ ต้องการ $Q_{next2} = 1$ จะได้ $R_2 = 0$ และ $S_2 = 1$

แถวที่ 4 ($Q_1 = 1, Q_2 = 1$) เป็นกรณีที่ไม่สนใจจึงกำลังให้ $R_1 = S_1 = R_2 = S_2 = x$

ได้ตารางเอ็กไซเทชันเป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | $X = 0$ | | $X = 1$ | | เอาต์พุต (Z) | |
|-------------------------------|----------|----------|----------|----------|------------------|---------|
| | R_1S_1 | R_2S_2 | R_1S_1 | R_2S_2 | $X = 0$ | $X = 1$ |
| 00 | x0 | x0 | x0 | 01 | 0 | 0 |
| 01 | 01 | 10 | x0 | 0x | 0 | 0 |
| 10 | 10 | x0 | 10 | 01 | 0 | 1 |
| 11 | xx | xx | xx | xx | 0 | 0 |

ขั้นตอนที่ 4: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทั้งหมด และเอาต์พุตของฟลิปฟล็อป

1. แผนผังคาร์โนห์ของ R_1

| | | | | | | |
|-----|----------|----|----|----|----|---------|
| | Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X | 0 | x | 0 | x | 1 | ← Q_1 |
| | 1 | x | x | x | 1 | |

$$R_1 = Q_1$$

2. แผนผังคาร์โนห์ของ S_1

| | | | | | | |
|-----|----------|----|----|----|----|----------|
| | Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X | 0 | 0 | 1 | x | 0 | ← Q_2X |
| | 1 | 0 | 0 | x | 0 | |

$$S_1 = Q_2X$$

3. แผนผังคาร์โนห์ของ R_2

| | | | | | |
|----------|----|----|----|----|-----|
| Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X \ 0 | x | 1 | x | x | ← X |
| 1 | 0 | 0 | x | 0 | |

$$R_2 = X$$

4. แผนผังคาร์โนห์ของ S_2

| | | | | | |
|----------|----|----|----|----|-----|
| Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X \ 0 | 0 | 0 | x | 0 | ← X |
| 1 | 1 | x | x | 1 | |

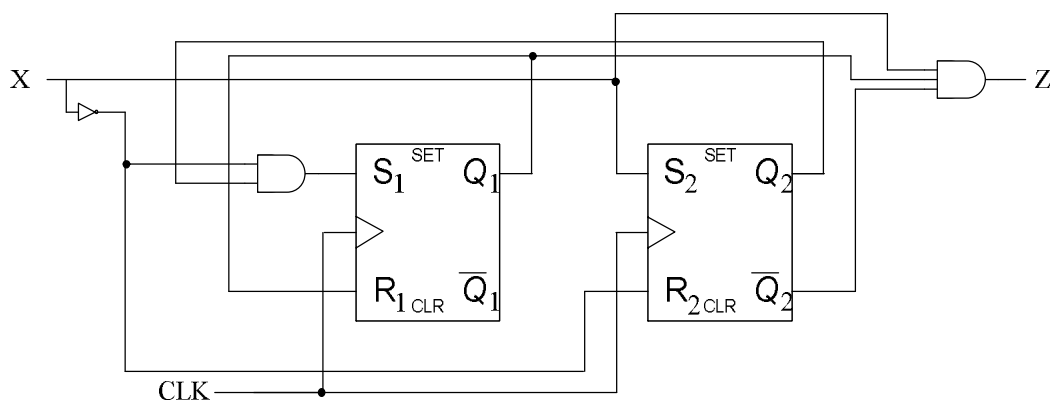
$$S_2 = X$$

5. แผนผังคาร์โนห์ของ Z

| | | | | | |
|----------|----|----|----|----|-------------|
| Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X \ 0 | 0 | 0 | 0 | 0 | ← Q_1Q_2X |
| 1 | 0 | 0 | 0 | 1 | |

$$Z = Q_1Q_2X$$

ขั้นตอนที่ 5: สร้างวงจรรวมเชิงลำดับโดยที่ค่าเอาต์พุตจะมีค่าเป็น 1 ได้ก็ต่อเมื่ออินพุตที่ป้อนเข้ามามีค่าเป็นลำดับที่ต่อเนื่องคือ "101" โดยเมื่อเอาต์พุตมีค่าเป็น 1 แล้วจะต้องเริ่มนับสถานะของอินพุตใหม่ สำหรับกรณีอื่นๆ จะไม่สนใจสถานะเอาต์พุต โดยใช้ฟลิปฟล็อปแบบ JK



ตัวอย่างที่ 8-6 จงออกแบบวงจรรวมเชิงลำดับโดยที่ค่าเอาต์พุตจะมีค่าเป็น 1 ได้ก็ต่อเมื่ออินพุตที่ป้อนเข้ามามีค่าเป็นลำดับที่ต่อเนื่องคือ "101" โดยเมื่อเอาต์พุตมีค่าเป็น 1 แล้วจะต้องเริ่มนับสถานะของอินพุตใหม่ สำหรับกรณีอื่นๆ จะไม่สนใจสถานะเอาต์พุต โดยใช้ฟลิปฟล็อปแบบ JK

วิธีทำ ตัวอย่างนี้จะแตกต่างจากตัวอย่างที่ 8-5 ดังนี้

กรณีที่เอาต์พุตมีค่าเป็น 1 แล้ว วงจรจะต้องเริ่มนับสถานะใหม่ของอินพุตเท่านั้นคือ อินพุตลำดับที่ 3 ที่มีค่าเป็น 1 จะไม่ถูกนำมาพิจารณาสำหรับวงเคราะห้สถานะของเอาต์พุตที่จะมีค่าเป็น 1 ในลำดับถัดไป

โดยตัวอย่างการทำงานของวงจรจะต้องเป็นดังนี้

| | | | | | | | |
|-----------|---|---|---|---|---|---|---|
| ลำดับที่: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| อินพุต: | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| เอาต์พุต: | X | X | 1 | X | X | X | 1 |

ขั้นตอนที่ 1: จากโจทย์ที่กำหนดมีเพียง 1 อินพุตจึงสามารถวิเคราะห์ตารางสถานะได้ดังนี้

กำหนดให้ X คือข้อมูลอินพุต

A แทนสถานะเริ่มต้น ดังนี้ A จึงแทนสถานะของข้อมูลอินพุตที่ป้อนเข้ามายังไม่ถูกเลย

B แทนสถานะที่เกิดขึ้นกรณีที่อินพุตถูกป้อนมาถูกแล้ว 1 ค่า (1)

C แทนสถานะที่เกิดขึ้นกรณีที่อินพุตถูกป้อนมาถูกแล้ว 2 ค่า (10)

ดังนั้นสถานะที่เป็นไปได้มีทั้งหมด 3 สถานะประกอบด้วย A, B และ C การวิเคราะห์หาสถานะที่เกิดขึ้นใหม่สามารถทำได้ดังนี้

กรณีที่ สถานะปัจจุบันคือ A

เมื่อ X = 0 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามายังไม่ถูกรูปแบบเลย เนื่องจาก A เป็นสถานะที่ยังไม่ถูกรูปแบบเลย ดังนั้นสถานะที่เกิดขึ้นใหม่ยังคงอยู่ที่ A และ โดยไม่สนใจสถานะเอาต์พุต

เมื่อ X = 1 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามามีความถูกต้องแล้ว 1 ค่า เนื่องจาก B เป็นสถานะที่เกิดจากการป้อนข้อมูลอินพุตถูกรูปแบบแล้ว 1 ค่า ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ B โดยไม่สนใจสถานะเอาต์พุต

กรณีที่ สถานะปัจจุบันคือ B

เมื่อ X = 0 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาเข้ามามีความถูกต้องแล้ว 2 ค่า เนื่องจาก C เป็นสถานะที่เกิดจากการป้อนข้อมูลอินพุตถูกรูปแบบแล้ว 2 ค่า ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ C โดยไม่สนใจสถานะเอาต์พุต

เมื่อ X = 1 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามายังคงมีความถูกต้องอยู่ 1 ค่า เช่นเดิมนั้นสถานะที่เกิดขึ้นใหม่ยังคงอยู่ที่ B โดยไม่สนใจสถานะเอาต์พุต

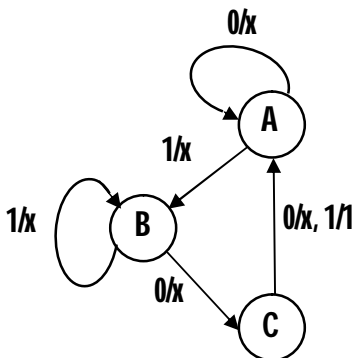
กรณีที่ สถานะปัจจุบันคือ C

เมื่อ X = 0 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาแล้วนั้นไม่ถูกรูปแบบเลย ดังนั้นสถานะที่เกิดขึ้นใหม่จะต้องกลับมาอยู่ที่ A โดยไม่สนใจสถานะเอาต์พุต

เมื่อ X = 1 แสดงให้เห็นว่าข้อมูลอินพุตที่ป้อนเข้ามาเข้ามามีความถูกต้องแล้ว 3 ค่า ทำให้เอาต์พุตมีค่าเป็น 1 หลังจากนั้นวงจรจะเริ่มนับสถานะของอินพุตใหม่อีกครั้ง ดังนั้นสถานะที่เกิดขึ้นใหม่อยู่ที่ A

จากทั้ง 3 สถานะที่วิเคราะห์ได้ทั้งหมด สามารถสร้างแผนภาพสถานะ และตารางสถานะได้ดังนี้

แผนภาพสถานะ:



ตารางสถานะ:

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|-------|
| | X = 0 | X = 1 |
| A | A/x | B/x |
| B | C/x | B/x |
| C | A/x | A/1 |

ขั้นตอนที่ 2: สร้างตารางทรานซิชันจากตารางสถานะได้ดังนี้

- กำหนดให้ A แทนสถานะ 00
- B แทนสถานะ 01
- C แทนสถานะ 10

| สถานะปัจจุบัน (Q_1Q_2) | สถานะที่เกิดขึ้นใหม่/เอาต์พุต (Q_{next}/Z) | |
|-------------------------------|---|-------|
| | X = 0 | X = 1 |
| 00 | 00/x | 01/x |
| 01 | 10/x | 01/x |
| 10 | 00/x | 00/1 |

ขั้นตอนที่ 3: สร้างตารางเอ็ทไซเทชั่นจากตารางทรานซิชัน

เนื่องจากสถานะที่เป็นไปได้ทั้งหมดของเอาต์พุตของฟลิปฟล็อปมี 3 สถานะ หากใช้ฟลิปฟล็อป 1 ตัวจะได้ 2 สถานะซึ่งไม่เพียงพอ ดังนั้นจึงต้องใช้ฟลิปฟล็อป 2 ตัวซึ่งจะได้ทั้งหมด 4 สถานะ อย่างไรก็ตามจากตัวอย่างต้องการเพียงแค่ 3 สถานะคือ 00, 01 และ 10 ดังนั้นสถานะ 11 ถึงเป็นสถานะที่ไม่สนใจ

จากตารางเอ็ทไซเทชั่นของฟลิปฟล็อปแบบ JK ได้ว่า

แถวที่ 1 ($Q_1 = 0, Q_2 = 0$)

เมื่อ X = 0 ต้องการ $Q_{next1} = 0$ จะได้ J = 0 และ K = X และ ต้องการ $Q_{next2} = 0$ จะได้ J =

0 และ K = X

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $J = 0$ และ $K = X$ และ ต้องการ $Q_{next2} = 1$ จะได้ $J = 1$ และ $K = X$

แถวที่ 2 ($Q_1 = 0, Q_2 = 1$)

เมื่อ $X = 0$ ต้องการ $Q_{next1} = 1$ จะได้ $J = 1$ และ $K = X$ และ ต้องการ $Q_{next2} = 0$ จะได้ $J = X$ และ $K = 1$

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $J = 0$ และ $K = X$ และ ต้องการ $Q_{next2} = 1$ จะได้ $J = X$ และ $K = 0$

แถวที่ 3 ($Q_1 = 1, Q_2 = 0$)

เมื่อ $X = 0$ ต้องการ $Q_{next1} = 0$ จะได้ $J = X$ และ $K = 1$ และ ต้องการ $Q_{next2} = 0$ จะได้ $J = 0$ และ $K = X$

เมื่อ $X = 1$ ต้องการ $Q_{next1} = 0$ จะได้ $J = X$ และ $K = 1$ และ ต้องการ $Q_{next2} = 0$ จะได้ $J = 0$ และ $K = X$

แถวที่ 4 ($Q_1 = 1, Q_2 = 1$)

เป็นกรณีที่ไม่สนใจจึงกำลังให้ $J = X$ และ $K = X$

ได้ตารางเอ็กไซเทชันเป็นดังนี้

| สถานะปัจจุบัน (Q_1Q_2) | $X = 0$ | | $X = 1$ | | เอาต์พุต (Z) | |
|-------------------------------|----------|----------|----------|----------|------------------|---------|
| | J_1K_1 | J_2K_2 | J_1K_1 | J_2K_2 | $X = 0$ | $X = 1$ |
| 00 | 0x | 0x | 0x | 1x | x | x |
| 01 | 1x | x1 | 0x | x0 | x | x |
| 10 | x1 | 0x | x1 | 0x | x | 1 |
| 11 | xx | xx | xx | xx | x | x |

ขั้นตอนที่ 4: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทั้งหมด และเอาต์พุตของฟลิปฟล็อป

1. แผนผังคาร์โนห์ของ J_1

| | | | | | | |
|-----|----------|----|----|----|----|--------|
| | Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X | 0 | 0 | 1 | x | x | Q_2X |
| | 1 | 0 | 0 | x | x | |

$$J_1 = Q_2X$$

2. แผนผังคาร์โนห์ของ K_1

| | | | | | | |
|-----|----------|----|----|----|----|---|
| | Q_1Q_2 | 00 | 01 | 11 | 10 | |
| X | 0 | x | x | x | 1 | 1 |
| | 1 | x | x | x | 1 | |

$$K_1 = 1$$

3. แผนผังคาร์โนห์ของ J_2

| | | | | |
|----------------------|----|----|----|----|
| $Q_1 \backslash Q_2$ | 00 | 01 | 11 | 10 |
| X \ 0 | 0 | x | x | 0 |
| 1 | 1 | x | x | 0 |

$$J_2 = Q_1 X$$

4. แผนผังคาร์โนห์ของ K_2

| | | | | |
|----------------------|----|----|----|----|
| $Q_1 \backslash Q_2$ | 00 | 01 | 11 | 10 |
| X \ 0 | x | 1 | x | x |
| 1 | x | 0 | x | x |

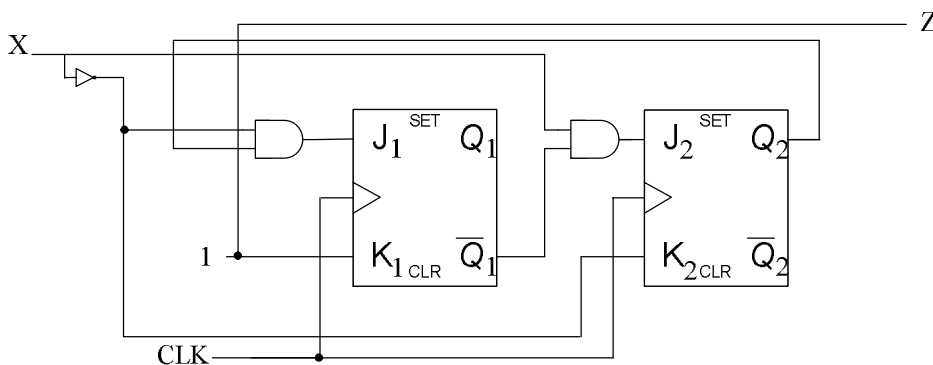
$$K_2 = Q_2 X$$

5. แผนผังคาร์โนห์ของ Z

| | | | | |
|----------------------|----|----|----|----|
| $Q_1 \backslash Q_2$ | 00 | 01 | 11 | 10 |
| X \ 0 | x | x | x | x |
| 1 | x | x | x | 1 |

$$Z = 1$$

ขั้นตอนที่ 5: สร้างวงจรจากสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปและเอาต์พุตได้ดังนี้



8.4 การลดสถานะเอาต์พุตของฟลิปฟล็อป

เนื่องจากการออกแบบวงจรเชิงลำดับนั้น จำเป็นต้องสร้างแผนภาพสถานะ หรือตารางสถานะ ขึ้นมาใหม่ซึ่งมีความเป็นไปได้ที่สถานะเอาต์พุตของฟลิปฟล็อปจะมีมากเกินไปจนความจำเป็น หัวข้อนี้จะกล่าวถึงการตรวจสอบสถานะเอาต์พุตของฟลิปฟล็อปที่ได้ออกแบบไว้ว่ามีมากเกินไปหรือไม่ โดยใช้วิธีที่ถูกรับรองโดย Merger และหากพบว่ามีมากเกินไปจะทำการลดสถานะลงให้

เหลือน้อยที่สุดแต่ผลลัพธ์ที่ได้ยังคงเดิม เนื่องจากหากสถานะเอาต์พุตของฟลิปฟล็อปน้อยลง จะทำให้จำนวนฟลิปฟล็อปที่ต้องใช้งานลดน้อยลงด้วยเช่นกัน ขั้นตอนการตรวจสอบและลดสถานะมีดังต่อไปนี้

ขั้นตอนที่ 1: สร้างตารางอิมพลิเคชันที่มีจำนวนช่องในแนวนอนสูงสุดคือ $n - 1$ เมื่อ n คือจำนวนสถานะทั้งหมดโดยเริ่มจากสถานะแรกเรียงไปจนถึงสถานะรองสุดท้ายและจำนวนช่องในแนวตั้งสูงสุดคือ $n - 1$ โดยเริ่มจากสถานะที่สองไปจนถึงสถานะสุดท้าย ยกตัวอย่างเช่นสมมติว่ามีสถานะทั้งหมด 4 สถานะคือ A, B, C และ D ตารางอิมพลิเคชันจะเป็นดังนี้

| | | | |
|---|---|---|---|
| B | 1 | | |
| C | 2 | 4 | |
| D | 3 | 5 | 6 |
| | A | B | C |

ขั้นตอนที่ 2: สำหรับทุกๆ ช่องที่อยู่ในตารางอิมพลิเคชันให้ทำการเปรียบเทียบสถานะแต่ละสถานะว่ารวมกันได้หรือไม่ โดยการเปรียบเทียบสถานะที่อยู่ในแนวตั้ง และสถานะที่อยู่ในแนวนอนของแต่ละช่อง ยกตัวอย่างเช่นจากตารางอิมพลิเคชัน (Implication) ข้างต้น หากพิจารณาช่องหมายเลข 1 พบว่าในแนวตั้งคือสถานะ A และในแนวนอนคือสถานะ B ดังนั้น ช่องดังกล่าวจะถูกตรวจสอบว่าสามารถรวมสถานะ A และ สถานะ B เข้าด้วยกันได้หรือไม่โดยแบ่งออกเป็น 3 กรณีดังนี้

กรณีที่ 1 (รวมไม่ได้): สถานะที่ถูกพิจารณาอยู่มีอินพุตเหมือนกัน แต่ให้อาต์พุตที่มีสถานะต่างกันโดยให้ใส่เครื่องหมาย " X " ไว้ที่ช่องที่พิจารณา

กรณีที่ 2 (รวมได้แบบไม่มีเงื่อนไข): สถานะที่ถูกพิจารณาอยู่มีอินพุตเหมือนกัน ให้อาต์พุตที่เหมือนกัน และสถานะใหม่ของเอาต์พุตของฟลิปฟล็อปมีสถานะเหมือนกัน โดยให้ใส่เครื่องหมาย " ✓ " ไว้ที่ช่องที่พิจารณา

กรณีที่ 3 (รวมได้แบบมีเงื่อนไข): สถานะที่ถูกพิจารณาอยู่มีอินพุตเหมือนกัน และให้อาต์พุตที่มีสถานะเหมือนกัน แต่สถานะใหม่ของเอาต์พุตของฟลิปฟล็อปมีความแตกต่างกัน โดยให้ใส่สถานะใหม่ของเอาต์พุตของฟลิปฟล็อปทั้ง 2 ค่าไว้ที่ช่องที่พิจารณา ความหมายคือสถานะที่กำลังพิจารณาทั้ง 2 สถานะจะสามารถรวมกันได้ก็ต่อเมื่อ สถานะใหม่ของเอาต์พุตของฟลิปฟล็อปทั้ง 2 ค่าที่เป็นเงื่อนไขต้องสามารถรวมกันได้

ขั้นตอนที่ 3: พิจารณาช่องที่เป็นการรวมได้แบบมีเงื่อนไขทั้งหมด ว่าสามารถรวมกันได้จริงหรือไม่ โดยให้พิจารณาสถานะที่เป็นเงื่อนไข ซึ่งหากสถานะที่เป็นเงื่อนไขสามารถรวมกันได้ แสดงว่าช่องที่กำลังพิจารณาอยู่นั้นสามารถรวมกันได้ แต่หากสถานะที่เป็นเงื่อนไขไม่สามารถรวมกันได้ แสดงว่าช่องที่กำลังพิจารณาอยู่นั้นไม่สามารถรวมกันได้

ขั้นตอนที่ 4: วาดกราฟของสถานะทั้งหมด โดยลากเส้นเชื่อมระหว่างสถานะที่รวมกันได้ โดยหากเป็นกรณีที่รวมได้แบบมีเงื่อนไข ให้ใส่สถานะที่เป็นเงื่อนไขกำกับไว้ที่เส้นเชื่อมดังกล่าวด้วย สำหรับการรวมกันได้นั้นถูกพิจารณาเป็นดังนี้

กรณีที่ 1: หากสถานะที่กำลังพิจารณาไม่มีเส้นเชื่อมไปสถานะใดเลย แสดงว่าสถานะนี้ไม่สามารถรวมกับสถานะอื่น ๆ ได้

กรณีที่ 2: หากมีเส้นเชื่อมถึงกันถือว่ารวมกันได้

กรณีที่ 3: หากมีเส้นเชื่อมสถานะเป็นรูปสามเหลี่ยมแสดงว่า สามารถรวมสถานะทั้ง 3 สถานะนั้นได้

กรณีที่ 4: หากมีเส้นเชื่อมสถานะเป็นรูปหลายเหลี่ยม (มากกว่าสามเหลี่ยม) และมีเส้นทแยงมุมในทุกๆด้าน แสดงว่าสามารถรวมสถานะทั้งหมดที่ทำให้เกิดรูปหลายเหลี่ยมจากเส้นเชื่อมได้

กรณีที่ 5: หากเลือกเส้นเชื่อมที่มีเงื่อนไข ต้องเลือกเส้นที่เป็นเงื่อนไขของเส้นดังกล่าวนี้ด้วย

ขั้นตอนที่ 5: เลือกสถานะมาให้ครบทั้งหมด และเลือกให้น้อยที่สุด โดยหากเลือกสถานะที่เกิดจากการรวมกันมาจะถือว่าเลือกสถานะเหล่านั้นมาทั้งหมด

ขั้นตอนที่ 6: สร้างตารางสถานะจากสถานะที่เลือกมาทั้งหมด

ตัวอย่างที่ 8-7 จากตารางสถานะต่อไปนี้ จงลดสถานะเอาต์พุตของฟลิปฟล็อปให้เหลือน้อยที่สุด

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|---------|
| | $X = 0$ | $X = 1$ |
| A | C/0 | B/0 |
| B | E/0 | D/0 |
| C | A/0 | D/1 |
| D | A/0 | B/1 |
| E | A/0 | B/1 |

วิธีทำ ขั้นตอนที่ 1: สร้างตารางอิมพลีเคชันโดยจากตัวอย่างสถานะทั้งหมดมี 5 สถานะ ดังนั้น ได้จำนวนช่องแนวนอน และแนวตั้งสูงสุดคือ 4 ช่อง ดังนี้

| | | | | |
|---|---|---|---|----|
| B | 1 | | | |
| C | 2 | 5 | | |
| D | 3 | 6 | 8 | |
| E | 4 | 7 | 9 | 10 |
| | A | B | C | D |

ขั้นตอนที่ 2: ตรวจสอบสถานะในแต่ละช่องว่าสามารถรวมกันได้หรือไม่

ช่องที่ 1 (A และ B): อินพุตเหมือนกันให้เอาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 0$ สถานะใหม่ของ A คือ C แต่สถานะใหม่ของ B คือ E และ กรณีที่ $X = 1$ สถานะใหม่ของ A คือ B แต่สถานะใหม่ของ B คือ D ดังนั้นจึงเขียน CE, BD ไว้ช่องที่ 1

ช่องที่ 2 (A และ C): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 2

ช่องที่ 3 (A และ D): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 3

ช่องที่ 4 (A และ E): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 4

ช่องที่ 5 (B และ C): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 5

ช่องที่ 6 (B และ D): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 6

ช่องที่ 7 (B และ E): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 7

ช่องที่ 8 (C และ D): อินพุตเหมือนกันให้เอาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 1$ สถานะใหม่ของ C คือ D แต่สถานะใหม่ของ D คือ B ดังนั้นจึงเขียน BD ไว้ช่องที่ 8 (สำหรับกรณีนี้จะมีเพียงเงื่อนไขเดียวเท่านั้นเนื่องจากว่ากรณีที่ $X = 0$ สถานะใหม่ของ C และ D เป็นตัวเดียวกันคือ A)

ช่องที่ 9 (C และ E): อินพุตเหมือนกันให้เอาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 1$ สถานะใหม่ของ C คือ D แต่สถานะใหม่ของ E คือ B ดังนั้นจึงเขียน BD ไว้ช่องที่ 9

ช่องที่ 10 (D และ E): อินพุตเหมือนกันให้เอาต์พุตและสถานะใหม่ของเอาต์พุตของฟลิปฟล็อปที่เหมือนกัน จึงรวมกันได้แบบไม่มีเงื่อนไข ให้ใส่เครื่องหมาย " ✓ " ไว้ช่องที่ 10

จากทุกๆ ช่องที่พิจารณาออกมา เขียนลงในตารางอิมพลีเคชันได้ดังนี้

| | | | | |
|---|---------------------|----------------|-----------------|-----------------|
| B | CE, BD ₁ | | | |
| C | X ₂ | X ₅ | | |
| D | X ₃ | X ₆ | BD ₈ | |
| E | X ₄ | X ₇ | BD ₉ | ✓ ₁₀ |
| | A | B | C | D |

ขั้นตอนที่ 3: พิจารณาช่องที่เป็นการรวมได้แบบมีเงื่อนไขทั้งหมด ว่าสามารถรวมกันได้จริงหรือไม่โดยช่องที่รวมได้แบบมีเงื่อนไขมีทั้งหมด 3 ช่อง ประกอบไปด้วย ช่องที่ 1, 8 และ 9

ช่องที่ 1: หากจะรวมได้ CE และ BD ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า BD (ช่องที่ 6) ไม่สามารถรวมได้ ดังนั้นสรุปได้ว่า ช่องที่ 1 ไม่สามารถรวมได้

ช่องที่ 8: หากจะรวมได้ BD ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า BD (ช่องที่ 6) ไม่สามารถรวมได้ ดังนั้นสรุปได้ว่า ช่องที่ 8 ไม่สามารถรวมได้

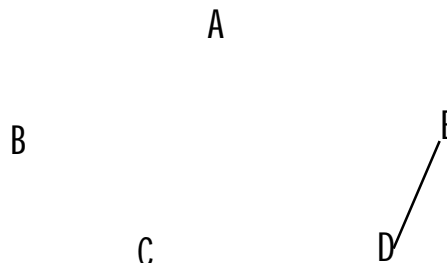
ช่องที่ 9: หากจะรวมได้ BD ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า BD (ช่องที่ 6) ไม่สามารถรวมได้ ดังนั้นสรุปได้ว่า ช่องที่ 9 ไม่สามารถรวมได้

ได้ตารางอิมพลีเคชันใหม่เป็นดังนี้

| | | | | |
|---|--|----------------|----------------------------|-----------------|
| B | CE BD ₁ | | | |
| C | X ₂ | X ₅ | | |
| D | X ₃ | X ₆ | BD ₈ | |
| E | X ₄ | X ₇ | BD ₉ | ✓ ₁₀ |
| | A | B | C | D |

ขั้นตอนที่ 4: วาดกราฟของสถานะทั้งหมด โดยลากเส้นเชื่อมระหว่างสถานะที่รวมกันได้

จากตารางอิมพลีเคชันพบว่ากราฟจะมีเส้นเชื่อมเพียงเส้นเดียวเท่านั้นคือ D-E เนื่องจากสามารถรวมกันได้ แต่กรณีอื่นๆ ไม่สามารถรวมกันได้



ขั้นตอนที่ 5: เลือกสถานะมาให้ครบทั้งหมด และเลือกให้น้อยที่สุด

จากกราฟเนื่องจาก D-E สามารถรวมกันได้ ดังนั้นหากเลือกคู่ดังกล่าวพบว่าเลือก 1 ครั้ง แต่สามารถเลือกมาได้ทั้ง E และ D ดังนั้นจึงเลือกอีกเพียง 3 ค่า คือ A, B และ C เนื่องจากทั้ง 3 ค่าดังกล่าวไม่สามารถรวมกับค่าใดได้เลยจึงจำเป็นต้องเลือกอีกทั้งหมด 3 ครั้งเพื่อเลือกทั้ง 3 ค่า สรุปได้ว่ามีการเลือกทั้งหมด 4 ครั้ง

ครั้งที่ 1 เลือก D-E

ครั้งที่ 2 เลือก A

ครั้งที่ 3 เลือก B

ครั้งที่ 4 เลือก C

ขั้นตอนที่ 6: สร้างตารางสถานะจากสถานะที่เลือกมาทั้งหมดได้ดังนี้

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|-------|
| | X = 0 | X = 1 |
| A | C/0 | B/0 |
| B | D-E/0 | D-E/0 |
| C | A/0 | D-E/1 |
| D-E | A/0 | B/1 |

ตัวอย่างที่ 8-7 จากตารางสถานะต่อไปนี้ จงลดสถานะเอาต์พุตของฟลิปฟล็อปให้เหลือน้อยที่สุด

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|-------|
| | X = 0 | X = 1 |
| A | B/0 | A/1 |
| B | C/0 | A/0 |
| C | C/0 | B/0 |
| D | E/0 | D/1 |
| E | C/0 | D/0 |

วิธีทำ ขั้นตอนที่ 1: สร้างตารางอิมพลีเคชันโดยจากตัวอย่างสถานะทั้งหมดมี 5 สถานะ ดังนั้น ได้จำนวนช่องแนวนอน และแนวตั้งสูงสุดคือ 4 ช่อง ดังนี้

| | | | | |
|---|---|---|---|----|
| B | 1 | | | |
| C | 2 | 5 | | |
| D | 3 | 6 | 8 | |
| E | 4 | 7 | 9 | 10 |
| | A | B | C | D |

ขั้นตอนที่ 2: ตรวจสอบสถานะในแต่ละช่องว่าสามารถรวมกันได้หรือไม่

ช่องที่ 1 (A และ B): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 1

ช่องที่ 2 (A และ C): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 2

ช่องที่ 3 (A และ D): อินพุตเหมือนกันให้อาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 0$ สถานะใหม่ของ A คือ B แต่สถานะใหม่ของ D คือ E ดังนั้นจึงเขียน BE ไว้ช่องที่ 3 (กรณีที่ $X = 1$ สถานะใหม่ของ A คือ A และสถานะใหม่ของ D คือ D ซึ่งแสดงว่า AD คือเงื่อนไขเช่นกัน แต่เนื่องจากช่องที่ 3 ที่เป็นช่องที่กำลังพิจารณาอยู่นั้นคือช่อง AD ความหมายคือเงื่อนไขการรวมกันได้คือ ค่าของตัวเองจึงถือว่ากรณีนี้เป็นการรวมกันได้แบบไม่มีเงื่อนไข)

ช่องที่ 4 (A และ E): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 4

ช่องที่ 5 (B และ C): อินพุตเหมือนกันให้อาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 1$ สถานะใหม่ของ B คือ A แต่สถานะใหม่ของ C คือ B ดังนั้นจึงเขียน AB ไว้ช่องที่ 5

ช่องที่ 6 (B และ D): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 6

ช่องที่ 7 (B และ E): อินพุตเหมือนกันให้อาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 1$ สถานะใหม่ของ B คือ A แต่สถานะใหม่ของ E คือ D ดังนั้นจึงเขียน AD ไว้ช่องที่ 7

ช่องที่ 8 (C และ D): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 8

ช่องที่ 9 (C และ E): อินพุตเหมือนกันให้อาต์พุตที่เหมือนกัน แต่สถานะใหม่ของเอาต์พุตต่างกันคือ กรณีที่ $X = 1$ สถานะใหม่ของ C คือ B แต่สถานะใหม่ของ E คือ D ดังนั้นจึงเขียน BD ไว้ช่องที่ 9

ช่องที่ 10 (D และ E): เนื่องจากกรณีที่ $X = 1$ สถานะเอาต์พุตมีความแตกต่างกัน จึงไม่สามารถรวมกันได้ให้ใส่เครื่องหมาย " X " ไว้ช่องที่ 10

จากทุกๆ ช่องที่พิจารณาออกมา เขียนลงในตารางอิมพลีเคชันได้ดังนี้

| | | | | | |
|---|----|---|----|---|----|
| B | X | 1 | | | |
| C | X | 2 | AB | 5 | |
| D | BE | 3 | X | 6 | X |
| E | X | 4 | AD | 7 | BD |
| | A | B | C | D | |

ขั้นตอนที่ 3: พิจารณาช่องที่เป็นการรวมได้แบบมีเงื่อนไขทั้งหมด ว่าสามารถรวมกันได้จริงหรือไม่โดยช่องที่รวมได้แบบมีเงื่อนไขมีทั้งหมด 4 ช่อง ประกอบไปด้วย ช่องที่ 3, 5, 7 และ 9

ช่องที่ 3: หากจะรวมได้ BE ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า BE (ช่องที่ 7) จะรวมได้ AD ก็ต้องรวมได้เช่นกัน สรุปได้ว่าช่องที่ 3 และ 7 เป็นเงื่อนไขของซึ่งกันและกัน

ช่องที่ 5: หากจะรวมได้ AB ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า AB (ช่องที่ 1) ไม่สามารถรวมได้ ดังนั้นสรุปได้ว่า ช่องที่ 5 ไม่สามารถรวมได้

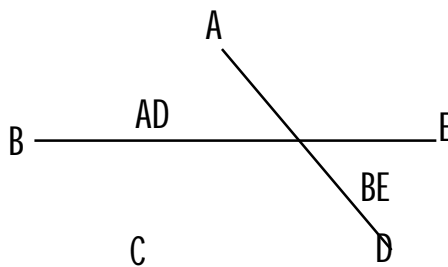
ช่องที่ 9: หากจะรวมได้ BD ต้องรวมได้ แต่จากตารางอิมพลีเคชันพบว่า BD (ช่องที่ 6) ไม่สามารถรวมได้ดังนั้นสรุปได้ว่า ช่องที่ 9 ไม่สามารถรวมได้

ได้ตารางอิมพลีเคชันใหม่เป็นดังนี้

| | | | | | |
|---|----|---|---------------|---|---------------|
| B | X | 1 | | | |
| C | X | 2 | AB | 5 | |
| D | BE | 3 | X | 6 | X |
| E | X | 4 | AD | 7 | BD |
| | A | B | C | D | |

ขั้นตอนที่ 4: วาดกราฟของสถานะทั้งหมด โดยลากเส้นเชื่อมระหว่างสถานะที่รวมกันได้

จากตารางอิมพลีเคชันพบว่ากราฟจะมีเส้นเชื่อมเพียงสองเส้นเท่านั้นคือ A-D และ B-E เนื่องจากสามารถรวมกันได้ แต่กรณีอื่นๆ ไม่สามารถรวมกันได้



ขั้นตอนที่ 5: เลือกสถานะมาให้ครบทั้งหมด และเลือกให้น้อยที่สุด

จากกราฟเนื่องจาก A-D สามารถรวมกันได้ดังนั้นเลือกครั้งที่ 1 จึงเลือกคู่ A-D แต่อย่างไรก็ตามเนื่องจาก A-D ขึ้นอยู่กับ B-E ดังนั้นจึงต้องเลือก B-E ด้วย ซึ่งถือได้ว่าเป็นการเลือกครั้งที่ 2 หลังจากเลือกแล้ว 2 ครั้งพบว่า C เป็นเพียงสถานะเดียวที่ยังไม่ได้เลือก และไม่สามารถรวมกับสถานะอื่นได้ ดังนั้นครั้งที่ 3 ต้องเลือก C

ครั้งที่ 1 เลือก A-D

ครั้งที่ 2 เลือก B-E

ครั้งที่ 3 เลือก C

ขั้นตอนที่ 6: สร้างตารางสถานะจากสถานะที่เลือกมาทั้งหมดได้ดังนี้

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|---------|
| | $X = 0$ | $X = 1$ |
| A-D | B-E/0 | A-D/1 |
| B-E | C/0 | A-D/0 |
| C | C/0 | B-E/0 |

8.5 บทสรุป

วงจรเชิงลำดับ คือ วงจรที่เกิดจากการต่อใช้งานร่วมกันระหว่างฟลิปฟล็อปอย่างน้อย 1 ตัวและไอซีพื้นฐาน วงจรลักษณะนี้จะสามารถนำสถานะเก่ากลับมาพิจารณาใหม่ได้ ซึ่งแบ่งออกเป็น 2 ประเภทคือวงจรเชิงลำดับแบบเข้าจังหวะ ความหมายคือสถานะเอาต์พุตของฟลิปฟล็อปทุกตัวภายในวงจรจะเกิดการเปลี่ยนแปลงพร้อมกันเนื่องจากใช้สัญญาณนาฬิกาตัวเดียวกัน และวงจรเชิงลำดับแบบไม่เข้าจังหวะ ความหมายคือฟลิปฟล็อปแต่ละตัวจะใช้สัญญาณนาฬิกาไม่เหมือนกัน

ตารางสถานะ และ แผนภาพสถานะ คือเครื่องมือที่ช่วยบอกการเปลี่ยนแปลงสถานะของวงจรเชิงลำดับที่เกิดขึ้นในแต่ละช่วงเวลาเพื่อช่วยให้การวิเคราะห์ และการออกแบบวงจรเชิงลำดับสามารถทำได้ง่ายมากยิ่งขึ้น

สำหรับการออกแบบวงจรเชิงลำดับ เมื่อได้ตารางสถานะของวงจรแล้วอาจเป็นไปได้ว่าสถานะที่ได้นั้นอาจมีมากเกินไป ซึ่งสามารถตรวจสอบและลดสถานะของวงจรในกรณีที่วงจรมีสถานะมากเกินไปได้โดยใช้หลักการที่ถูกรับรองโดย Merger

คำถามท้ายบท

1. จากตารางสถานะต่อไปนี้ จงแปลงเป็นแผนภาพสถานะ

1.1)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|---------|
| | $X = 0$ | $X = 1$ |
| A | D/1 | A/1 |
| B | B/0 | C/1 |
| C | A/0 | D/0 |
| D | D/0 | B/1 |

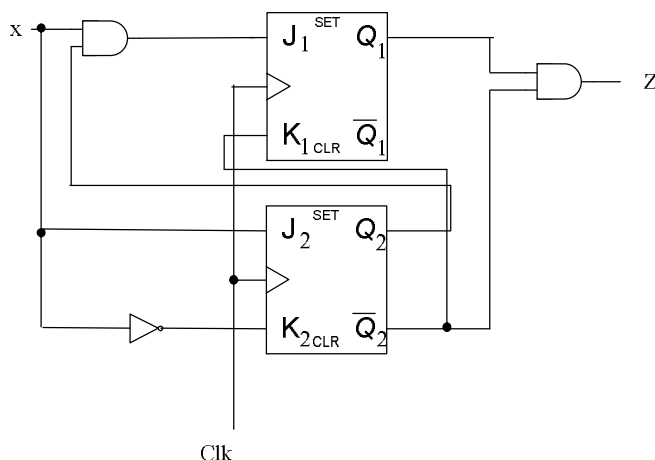
1.2)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|---------|
| | $X = 0$ | $X = 1$ |
| A | E/0 | B/1 |
| B | D/1 | C/0 |
| C | C/0 | A/0 |
| D | E/0 | C/1 |
| E | E/1 | D/0 |

1.3)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | | | |
|---------------------------|------------------------|---------------|---------------|---------------|
| | $X_1X_2 = 00$ | $X_1X_2 = 01$ | $X_1X_2 = 11$ | $X_1X_2 = 10$ |
| A | B/1 | A/1 | D/0 | D/1 |
| B | C/0 | D/1 | A/1 | C/0 |
| C | C/0 | B/1 | D/0 | B/0 |
| D | D/0 | B/1 | A/1 | C/0 |

2. จากวงจรที่กำหนดให้ต่อไปนี้



2.1) จงวิเคราะห์หาเอาต์พุต และสถานะที่เกิดขึ้น กรณีที่อินพุตที่ถูกป้อนเข้ามาอย่างเป็นลำดับมีค่าเป็น 11011 และสถานะเริ่มต้นของฟลิปฟล็อปทั้ง 2 ตัวมีค่าเป็น 00 ($Q_1 = 0$ และ $Q_2 = 0$)

2.2) จงวิเคราะห์หาเอาต์พุต และสถานะที่เกิดขึ้น กรณีที่อินพุตที่ถูกป้อนเข้ามาอย่างเป็นลำดับมีค่าเป็น 11010 และสถานะเริ่มต้นของฟลิปฟล็อปทั้ง 2 ตัวมีค่าเป็น 00 ($Q_1 = 1$ และ $Q_2 = 0$)

2.3) จงวิเคราะห์หาเอาต์พุต และสถานะที่เกิดขึ้น กรณีที่อินพุตที่ถูกป้อนเข้ามาอย่างเป็นลำดับมีค่าเป็น 10010 และสถานะเริ่มต้นของฟลิปฟล็อปทั้ง 2 ตัวมีค่าเป็น 00 ($Q_1 = 1$ และ $Q_2 = 1$)

3. จงออกแบบวงจรรวมเชิงลำดับโดยที่ค่าเอาต์พุตจะมีค่าเป็น 1 ได้ก็ต่อเมื่ออินพุตที่ป้อนเข้ามามีค่าเป็นลำดับที่ต่อเนื่องคือ "0100" สำหรับกรณีอื่นๆ จะทำให้เอาต์พุตมีค่าเป็น 0 ทั้งหมด โดยให้ใช้ฟลิปฟล็อปแบบ RS ทั้งหมด

4. จงออกแบบวงจรที่มีการทำงานเป็นดังแบบฝึกหัดข้อที่ 3 แต่ให้เปลี่ยนเป็นใช้ฟลิปฟล็อปแบบ JK ทั้งหมด

5. จากวงจรที่กำหนดให้ต่อไปนี้ จงลดสถานะของวงจรให้เหลือน้อยที่สุด (ในกรณีที่สามารถลดสถานะได้)

5.1)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|-------|
| | X = 0 | X = 1 |
| A | B/0 | A/1 |
| B | C/0 | A/0 |
| C | C/0 | B/0 |
| D | E/0 | D/1 |
| E | C/0 | D/0 |

5.2)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | |
|---------------------------|------------------------|---------|
| | $X = 0$ | $X = 1$ |
| A | D/0 | B/1 |
| B | F/1 | C/1 |
| C | D/0 | F/1 |
| D | C/0 | E/1 |
| E | C/1 | D/1 |
| F | D/1 | D/1 |
| G | D/1 | C/1 |

5.3)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | | |
|---------------------------|------------------------|---------------|---------------|
| | $X_1X_2 = 00$ | $X_1X_2 = 01$ | $X_1X_2 = 10$ |
| A | A/0 | B/0 | E/0 |
| B | D/0 | D/0 | A/0 |
| C | C/0 | C/0 | E/1 |
| D | C/0 | A/0 | B/0 |
| E | E/0 | A/0 | D/0 |

5.4)

| สถานะปัจจุบัน Q_1Q_2 | สถานะที่เกิดขึ้นใหม่/Z | | |
|---------------------------|------------------------|---------------|---------------|
| | $X_1X_2 = 00$ | $X_1X_2 = 01$ | $X_1X_2 = 10$ |
| A | A/0 | B/0 | A/0 |
| B | B/1 | B/1 | E/0 |
| C | A/0 | E/0 | D/1 |
| D | B/0 | A/0 | B/0 |
| E | D/1 | A/0 | D/0 |

เอกสารอ้างอิง

- Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.
- Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.
- David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.
- Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.
- Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.
- ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**.กรุงเทพฯ: มิตรนราการพิมพ์.
- ธวัชชัย เลื่อนฉวี. (2532). **ดิจิตอลเทคนิคเล่ม 2**.กรุงเทพฯ: มิตรนราการพิมพ์.
- มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**.กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินต์ติ้ง.
- ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**.กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.
- สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิตอลอิเล็กทรอนิกส์.จ. อุตรธานี: มหาวิทยาลัยราชภัฏอุตรธานี.**

แผนบริหารการสอนประจำบทที่ 9 วงจรรีเลย์และชิพทีริจิสเตอร์ 3 ชั่วโมง

หัวข้อเนื้อหา

9.1 วงจรรีเลย์เลขฐานสอง

9.1.1 วงจรรีเลย์เลขฐานสองแบบไม่เข้าจังหวะ

9.1.2 วงจรรีเลย์เลขฐานสองแบบไม่เข้าจังหวะกรณีที่ไม่ลงตัว

9.1.3 วงจรรีเลย์เลขฐานสองแบบเข้าจังหวะ

9.2 ชิพทีริจิสเตอร์

9.2.1 การถ่ายโอนข้อมูลแบบอนุกรม

9.2.2 การถ่ายโอนข้อมูลแบบขนาน

9.3 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการออกแบบวงจรรีเลย์เลขฐานสองทั้งแบบเข้าจังหวะและแบบไม่เข้าจังหวะ
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับชิพทีริจิสเตอร์ เพื่อใช้สำหรับการถ่ายโอนข้อมูล

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 9

วงจรรนับและชิพท์รีจิสเตอร์

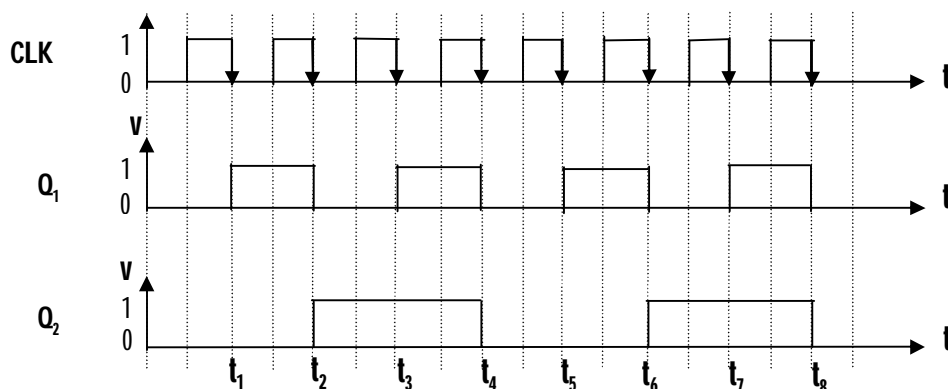
นอกจากวงจรเชิงลำดับแล้ว ฟลิปฟลอปยังสามารถถูกนำไปใช้ในการออกแบบวงจรรนับเลขฐานสอง และชิพท์รีจิสเตอร์ได้ด้วยเช่นกัน สำหรับบทนี้จะกล่าวถึงวิธีการนำฟลิปฟลอปมาใช้สำหรับการออกแบบวงจรรนับเลขฐานสองและชิพท์รีจิสเตอร์

9.1 วงจรรนับเลขฐานสอง

วงจรรนับเลขฐานสอง ถูกแบ่งออกเป็น 2 วิธี คือ วงจรรนับแบบไม่เข้าจังหวะ (Asynchronous Counter) และวงจรรนับแบบเข้าจังหวะ (Synchronous Counter) แต่ละวิธีจะแบ่งวงจรรนับออกเป็น 2 ประเภท คือ วงจรรนับขึ้น และวงจรรนับลง

9.1.1 วงจรรนับเลขฐานสองแบบไม่เข้าจังหวะ

วงจรรนับแบบไม่เข้าจังหวะคือวงจรรนับที่ฟลิปฟลอปแต่ละตัวในวงจรไม่ได้ใช้สัญญาณ CLK ร่วมกัน โดยจำนวนฟลิปฟลอป n ตัวสามารถนับได้ 2^n ค่า (0 ถึง 2^n-1)



รูปที่ 9.1 ตัวอย่างวงจรรนับ 4 แบบนับขึ้น

รูปที่ 9.1 แสดงตัวอย่างวงจรรนับ 4 ซึ่งจะมีการเปลี่ยนสถานะทุกครั้งที่ขอบขาลงของสัญญาณ CLK โดยมีการทำงานเป็นดังต่อไปนี้

จากรูป กำหนดให้สถานะของ Q_1 และ Q_2 ก่อนที่จะถึงช่วงเวลา t_1 มีค่าเป็น 0 ก่อนช่วงเวลา t_1 : สถานะของ " Q_2Q_1 " คือ "00" ซึ่งมีค่าเท่ากับ 0 ในเลขฐานสิบ
 ช่วงเวลา t_1 : สถานะของ " Q_2Q_1 " คือ "01" ซึ่งมีค่าเท่ากับ 1 ในเลขฐานสิบ
 ช่วงเวลา t_2 : สถานะของ " Q_2Q_1 " คือ "10" ซึ่งมีค่าเท่ากับ 2 ในเลขฐานสิบ
 ช่วงเวลา t_3 : สถานะของ " Q_2Q_1 " คือ "11" ซึ่งมีค่าเท่ากับ 3 ในเลขฐานสิบ

ช่วงเวลา 4: สถานะของ " Q_2Q_1 " คือ "00" ซึ่งมีค่าเท่ากับ 0 ในเลขฐานสิบ (กลับมาเริ่มนับใหม่)

ดังนั้นจากรูปที่ 9.1 จะเป็นวงจรที่นับขึ้น 0 - 3 และกลับมาเริ่มนับใหม่

เนื่องจาก Q_1 จะมีการเปลี่ยนสถานะทุกครั้งที่เกิดสัญญาณ CLK ที่ขอบขาลง และ Q_2 จะมีการเปลี่ยนสถานะทุกครั้งที่เกิดสัญญาณที่ขอบขาลงของ Q_1 แสดงว่า Q_1 เปรียบเสมือนเป็นสัญญาณ CLK ให้ Q_2

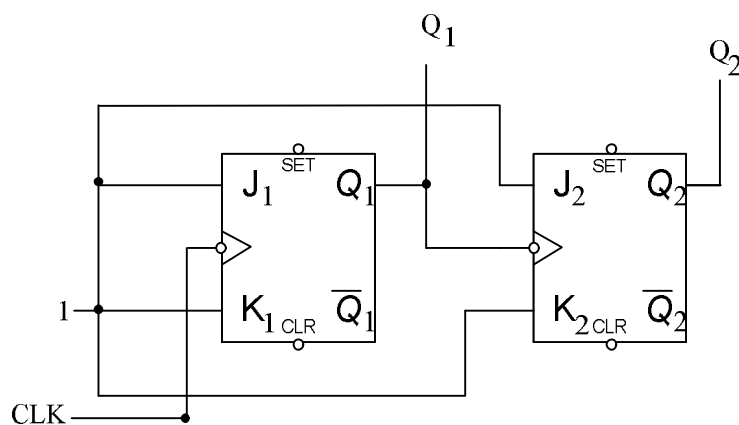
เนื่องจากฟลิปฟล็อปทั้ง 2 ตัวมีการเปลี่ยนสถานะของเอาต์พุตทุกครั้งที่เกิดสัญญาณ CLK ของฟลิปฟล็อปแต่ละตัว จึงต้องกำหนดให้ฟลิปฟล็อปมีการเปลี่ยนสถานะเอาต์พุตทุกครั้งที่เกิดสัญญาณ CLK ซึ่งสามารถทำได้ดังนี้

หากเลือกฟลิปฟล็อป JK ต้องกำหนดให้ J และ K มีค่าเป็น 1

หากเลือกฟลิปฟล็อป T ต้องกำหนดให้ T มีค่าเป็น 1

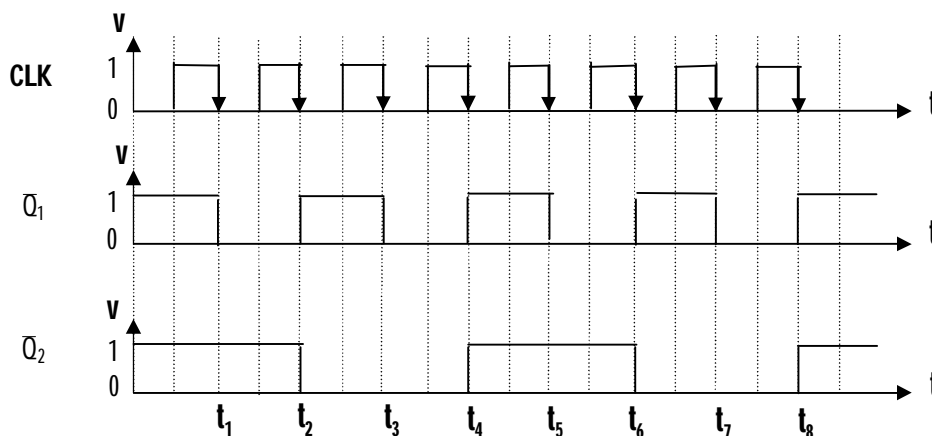
หากเลือกฟลิปฟล็อป D, RS จำเป็นต้องนำมาแปลงให้เป็น T หรือ JK ก่อนแล้วกำหนดให้สัญญาณมีค่าเป็น 1 ทั้งหมด

จากรูปที่ 9.1 หากเลือกฟลิปฟล็อป JK จะได้วงจรนับขึ้นแบบนับ 4 เป็นดังนี้



รูปที่ 9.2 ตัวอย่างวงจรนับ 4 แบบนับขึ้นโดยใช้ฟลิปฟล็อปแบบ JK

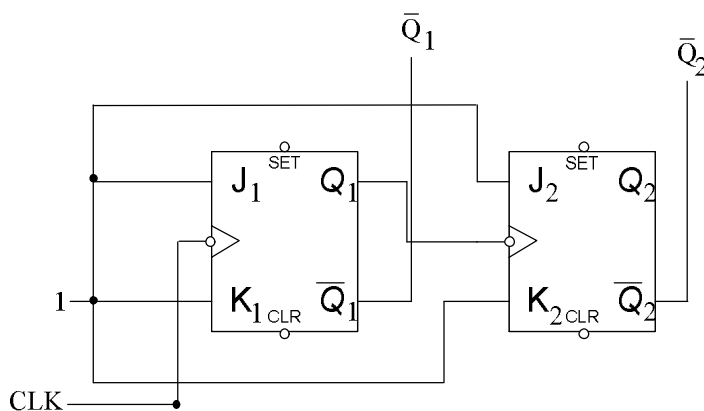
จากรูปที่ 9.2 สังเกตว่าความถี่ของสัญญาณ CLK จะมากกว่า Q_1 อยู่ 2 เท่า และความถี่ของสัญญาณ Q_1 มากกว่าความถี่ของสัญญาณ Q_2 อยู่ 2 เท่า แสดงว่าความถี่ของ Q_1 และ Q_2 จะลดลง $1/2$ และ $1/4$ เท่าตามลำดับ เมื่อเปรียบเทียบกับสัญญาณ CLK ดังนั้นสามารถนำวงจรนับแบบไม่เข้าจังหวะมาใช้งานเพื่อเป็นวงจรหารความถี่ลงได้เช่นกันและจากรูปที่ 9.2 หากเลือกสถานะนิเสธของ Q_1 และ Q_2 มาพิจารณาจะได้ผลลัพธ์เป็นดังนี้



รูปที่ 9.3 ตัวอย่างวงจรนับ 4 แบบนับลง

รูปที่ 9.3 เป็นการนำสถานะนิเสธของ Q_1 และ Q_2 ของรูปที่ 9.1 มาพิจารณาได้ดังนี้
 ก่อนช่วงเวลา t_1 : สถานะของ " $Q_2 Q_1$ " คือ "11" ซึ่งมีค่าเท่ากับ 3 ในเลขฐานสิบ
 ช่วงเวลา t_1 : สถานะของ " $Q_2 Q_1$ " คือ "10" ซึ่งมีค่าเท่ากับ 2 ในเลขฐานสิบ
 ช่วงเวลา t_2 : สถานะของ " $Q_2 Q_1$ " คือ "01" ซึ่งมีค่าเท่ากับ 1 ในเลขฐานสิบ
 ช่วงเวลา t_3 : สถานะของ " $Q_2 Q_1$ " คือ "00" ซึ่งมีค่าเท่ากับ 0 ในเลขฐานสิบ
 ช่วงเวลา t_4 : สถานะของ " $Q_2 Q_1$ " คือ "11" ซึ่งมีค่าเท่ากับ 3 ในเลขฐานสิบ (กลับมาเริ่มนับใหม่)

ดังนั้นสรุปได้ว่าหากเลือกนิเสธของ Q_1 และ Q_2 มาใช้งานแทนวงจรที่ใช้ Q_1 และ Q_2 เป็นเอาต์พุตของวงจร จะกลายเป็นวงจรนับลง ซึ่งจะเป็นวงจรที่นับลง 3-0 และกลับมาเริ่มนับใหม่จากรูปที่ 9.4 หากเลือกฟลิปฟล็อป JK จะได้วงจรนับ 4 แบบนับลงเป็นดังนี้



รูปที่ 9.4 ตัวอย่างวงจรนับ 4 แบบนับลงโดยใช้ฟลิปฟล็อปแบบ JK

หากต้องการสร้างวงจรนับแบบนับ 8 สามารถทำได้โดยเพิ่มฟลิปฟล็อปมาอีก 1 ตัวโดยนำสัญญาณ Q_2 มาเป็นสัญญาณนาฬิกาให้กับฟลิปฟล็อปตัวใหม่ ดังนั้นสรุปได้ว่าหากต้องการสร้างวงจรนับ 2^n จะต้องใช้ฟลิปฟล็อปจำนวน n ตัว

9.1.2 วงจรนับเลขฐานสองแบบไม่เข้าจังหวะกรณีที่ไม่ลงตัว

เนื่องจากวงจรถับแบบไม่เข้าจังหวะนั้นจะสามารถนับได้ดังนี้

ฟลิปฟล็อป 1 ตัว -> วงจรนับ 2

ฟลิปฟล็อป 2 ตัว -> วงจรนับ 4

ฟลิปฟล็อป n ตัว -> วงจรนับ 2^n

ดังนั้นกรณีที่ต้องการออกแบบวงจรถับที่ไม่ได้อยู่ในรูปของ 2^n จะไม่สามารถออกแบบโดยวิธีที่เคยนำเสนอได้ อย่างไรก็ตามหากต้องการออกแบบวงจรถับที่ไม่ได้อยู่ในรูปของ 2^n สามารถทำได้ดังนี้

ขั้นตอนที่ 1: ตรวจสอบจำนวนฟลิปฟล็อปที่น้อยที่สุดที่จำเป็นต้องใช้สำหรับวงจรถับที่ต้องการเช่น หากต้องการออกแบบวงจรถับ 5 พบว่าหากใช้ฟลิปฟล็อป 2 ตัวจะได้เพียงแค่วงจรนับ 4 แต่หากใช้ฟลิปฟล็อป 3 ตัวจะได้วงจรถับ 8 ดังนั้นสำหรับวงจรถับ 5 ฟลิปฟล็อปที่น้อยที่สุดคือ 3 ตัว

ขั้นตอนที่ 2: แบ่งออกเป็น 2 กรณีดังนี้

กรณีเป็นวงจรถับขึ้น: หลังจากวงจรถับถึงค่าสูงที่สุด เมื่อเกิดสัญญาณ CLK ตัวต่อไปให้ส่งสัญญาณ CLR ให้ฟลิปฟล็อปทุกตัวเพื่อให้กลับไปเริ่มนับ 0 ซึ่งเป็นค่าที่ต่ำที่สุดใหม่อีกครั้ง

กรณีเป็นวงจรถับลง: หลังจากวงจรถับ 0 เมื่อเกิดสัญญาณ CLK ตัวต่อไปให้ส่งสัญญาณ CLR หรือ SET ไปให้ฟลิปฟล็อปทุกตัวเพื่อให้กลับไปเริ่มนับที่ค่าเริ่มต้นใหม่อีกครั้ง ยกตัวอย่างเช่น หากเป็นวงจรถับ 5 แบบนับลง (นับ 4 - 0) และมีการเปลี่ยนสถานะเมื่อเกิดสัญญาณนาฬิกาที่ขอบขาลงหลังจากนับ 0 แล้วฟลิปฟล็อปจะกลับไปเริ่มนับ 7 ดังนั้น ให้ส่งสัญญาณ CLR ไปที่ฟลิปฟล็อปตัวที่ 3 และส่งสัญญาณ SET ไปที่ฟลิปฟล็อปตัวที่ 1 และ 2 เนื่องจากจะทำให้สถานะ $Q_3Q_2Q_1$ มีค่าเป็น "011" แต่เนื่องจากสัญญาณที่ใช้คือ $Q_3Q_2Q_1$ ดังนั้นสถานะหลังจากนับ "000" แล้วจะมีค่าเป็น "100"

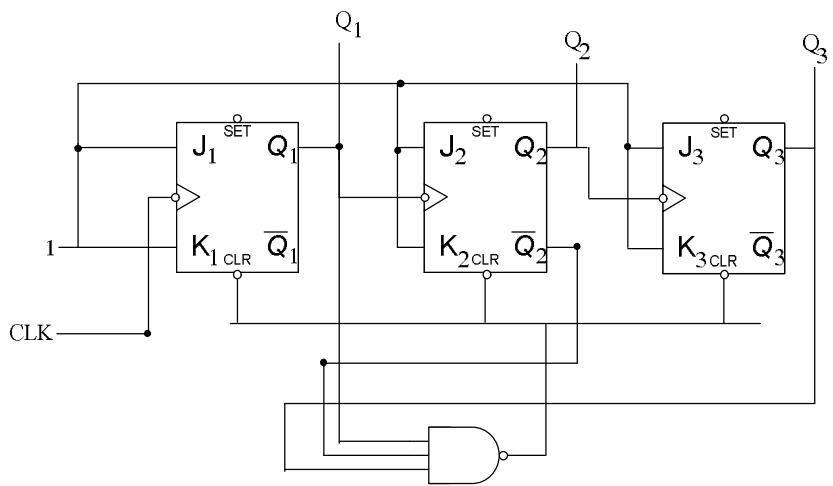
ตัวอย่างที่ 9-1 จงออกแบบวงจรถับ 5 ที่เป็นแบบนับขึ้น และเป็นวงจรถับแบบไม่เข้าจังหวะกำหนดให้ใช้สัญญาณนาฬิกาที่ขอบขาลง

วิธีทำ ตรวจสอบจำนวนฟลิปฟล็อปที่ต้องใช้งาน ดังนี้

จาก $2^2 = 4$ -> น้อยเกินไป

$2^3 = 8$ -> ดังนั้นจึงต้องใช้ฟลิปฟล็อปทั้งหมด 3 ตัว ซึ่งจะเป็นวงจรถับ 8 แต่หากต้องการให้เป็นวงจรถับ 5 สามารถทำได้โดยเมื่อวงจรถับถึง 5 ให้ส่งสัญญาณ CLR ไปยังฟลิปฟล็อปทั้ง 3 ตัว เพื่อให้ฟลิปฟล็อปทั้ง 3 เริ่มนับใหม่ในรอบต่อไป (แทนการนับค่า 5)

การส่งสัญญาณ CLR เมื่อฟลิปฟล็อปนับเป็น "101" เพื่อให้กลับมานับเป็น "000" สามารถทำได้โดยการนำเอาต์พุตที่เกิดจากการนำ Q_1, Q_2 และ Q_3 มาผ่านตัวดำเนินการแนบกันทั้งหมดเพื่อนำเอาต์พุตที่ได้มาใช้งานเป็นอินพุตให้ CLR ของฟลิปฟล็อปทุกตัวโดยเอาต์พุตที่ได้จะมีค่าเป็น 0 ก็ต่อเมื่อ $Q_1=1, Q_2=0(Q_2=1)$ และ $Q_3=1$ ซึ่งเกิดในจังหวะที่เอาต์พุตมีค่าเป็น "101" เท่านั้น โดยวงจรถับที่ได้เป็นดังต่อไปนี้



ตัวอย่างที่ 9-2 จงออกแบบวงจรนับ 5 ที่เป็นแบบนับลง (4 - 0) และเป็นวงจรแบบไม่เข้าจังหวะ กำหนดให้ใช้สัญญาณนาฬิกาที่ขอบขาสูง

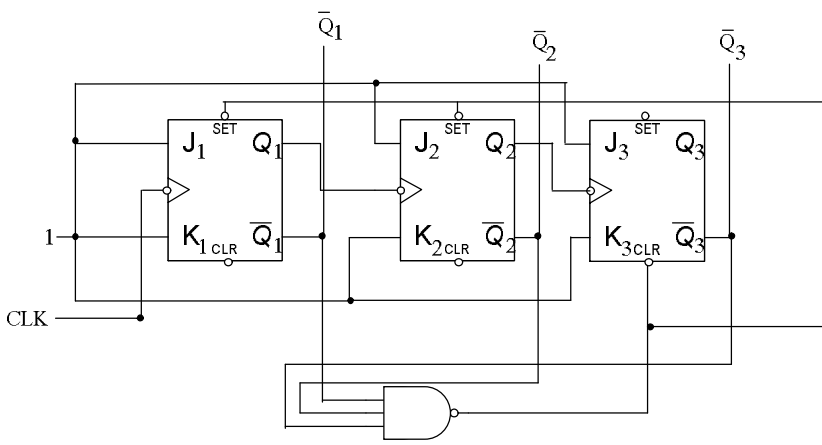
วิธีทำ ตรวจสอบจำนวนฟลิปฟล็อปที่ต้องใช้งาน ดังนี้

จาก $2^2 = 4 \rightarrow$ น้อยเกินไป

$2^3 = 8 \rightarrow$ ดังนั้นจึงต้องใช้ฟลิปฟล็อปทั้งหมด 3 ตัว ซึ่งจะเป็นวงจรนับ 8 โดยที่หลังจาก

นับ 0 แล้วฟลิปฟล็อป จะกลับไปเริ่มนับ 7 ดังนั้น ให้ส่งสัญญาณ RET ไปที่ฟลิปฟล็อปตัวที่ 3 และส่งสัญญาณ SET ไปที่ฟลิปฟล็อปตัวที่ 1 และ 2 เนื่องจากจะทำให้สถานะ $Q_3Q_2Q_1$ มีค่าเป็น "011" แต่เนื่องจากสัญญาณที่ใช้คือ $Q_3Q_2Q_1$ ดังนั้นสถานะหลังจากนับ "000" แล้วสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 จะมีค่าเปลี่ยนเป็น "100" ทันที

การกระตุ้นให้ $Q_3 = 0, Q_2 = 1$ และ $Q_1 = 1$ จะเกิดขึ้นเมื่อ $Q_3Q_2Q_1$ มีค่าเป็น "111" เนื่องจากเอาต์พุตที่เกิดจากการนำอินพุตทั้ง 3 ค่านี้มาผ่านตัวดำเนินการแนบกันแล้วจะได้ค่าเป็น 0 และจะถูกส่งไปที่สัญญาณ RET ของฟลิปฟล็อปตัวที่ 3 และส่งสัญญาณ SET ไปที่ฟลิปฟล็อปตัวที่ 1 และ 2 โดยวงจรที่ได้เป็นดังต่อไปนี้



9.1.3 วงจรนับเลขฐานสองแบบเข้าจังหวะ

วงจรรนับแบบเข้าจังหวะคือวงจรรนับที่ฟลิปฟล็อปทุกตัวในวงจรรใช้สัญญาณนาฬิกาาร่วมกัน โดยจำนวนฟลิปฟล็อป n ตัวสามารถนับได้ 2^n ค่า (0 ถึง 2^n-1) โดยมีขั้นตอนการออกแบบเป็นดังนี้

ขั้นตอนที่ 1: สร้างตารางทรานซิชั่นเพื่อหาสถานะของเอาต์พุตของฟลิปฟล็อปที่จะเกิดขึ้นใหม่

ขั้นตอนที่ 2: เลือกชนิดของฟลิปฟล็อปที่ต้องการใช้งาน แล้วทำการสร้างตารางเอ็กซิเทชันจากตารางทรานซิชั่น

ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทั้งหมด

ขั้นตอนที่ 4: สร้างวงจรรจากสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อป

ตัวอย่างที่ 9-3 จงออกแบบวงจรรนับ 7 ที่เป็นแบบนับขึ้นแบบเข้าจังหวะ

วิธีทำ จากโจทย์ต้องการวงจรรนับ 7 แบบนับขึ้นแสดงว่าเป็นวงจรรนับ 0 – 6 ซึ่งมีสถานะที่เป็นไปได้ทั้งหมด 7 สถานะ ดังนั้นต้องใช้ฟลิปฟล็อปทั้งหมด 3 ตัว

ขั้นตอนที่ 1: สร้างตารางทรานซิชั่นเพื่อหาสถานะของเอาต์พุตของฟลิปฟล็อปที่จะเกิดขึ้นใหม่

ลำดับที่ 1 สถานะปัจจุบันคือ "000" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "001"

ลำดับที่ 2 สถานะปัจจุบันคือ "001" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "010"

ลำดับที่ 3 สถานะปัจจุบันคือ "010" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "011"

ลำดับที่ 4 สถานะปัจจุบันคือ "011" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "100"

ลำดับที่ 5 สถานะปัจจุบันคือ "100" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "101"

ลำดับที่ 6 สถานะปัจจุบันคือ "101" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "110"

ลำดับที่ 7 สถานะปัจจุบันคือ "110" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "000"

ได้ตารางทรานซิชั่น เป็นดังนี้

| สถานะปัจจุบัน ($Q_3Q_2Q_1$) | สถานะที่เกิดขึ้นใหม่ ($Q_{next3}Q_{next2}Q_{next1}$) |
|----------------------------------|---|
| 000 | 001 |
| 001 | 010 |
| 010 | 011 |
| 011 | 100 |
| 100 | 101 |
| 101 | 110 |
| 110 | 000 |

ขั้นตอนที่ 2: เลือกชนิดของฟลิปฟล็อปที่ต้องการใช้งาน แล้วทำการสร้างตารางเอ็ชเทชั่นจากตารางทรานซิชัน

เนื่องจากโจทย์ไม่ได้กำหนดชนิดของฟลิปฟล็อปมาให้ ดังนั้นสามารถกำหนดชนิดของฟลิปฟล็อปได้เอง โดยผู้เขียนเลือกใช้ฟลิปฟล็อป JK ซึ่งต้องกำหนดค่าเป็นดังนี้

ลำดับที่ 1: $Q_3Q_2Q_1 = 000$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 001$

- 1) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
- 2) $Q_2 = 0, Q_{next2} = 0$ จะต้องกำหนด $J_2 = 0$ และ $K_2 = X$
- 3) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_1 = 1$ และ $K_1 = X$

ลำดับที่ 2: $Q_3Q_2Q_1 = 001$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 010$

- 1) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
- 2) $Q_2 = 0, Q_{next2} = 1$ จะต้องกำหนด $J_2 = 1$ และ $K_2 = X$
- 3) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_1 = X$ และ $K_1 = 1$

ลำดับที่ 3: $Q_3Q_2Q_1 = 010$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 011$

- 1) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
- 2) $Q_2 = 1, Q_{next2} = 1$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 0$
- 3) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_3 = 1$ และ $K_3 = X$

ลำดับที่ 4: $Q_3Q_2Q_1 = 011$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 100$

- 1) $Q_3 = 0, Q_{next3} = 1$ จะต้องกำหนด $J_3 = 1$ และ $K_3 = X$
- 2) $Q_2 = 1, Q_{next2} = 0$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 1$
- 3) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 1$

ลำดับที่ 5: $Q_3Q_2Q_1 = 100$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 101$

- 1) $Q_3 = 1, Q_{next3} = 1$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 0$
- 2) $Q_2 = 0, Q_{next2} = 0$ จะต้องกำหนด $J_2 = 0$ และ $K_2 = X$
- 3) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_3 = 1$ และ $K_3 = X$

ลำดับที่ 6: $Q_3Q_2Q_1 = 101$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 110$

- 1) $Q_3 = 1, Q_{next3} = 1$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 0$
- 2) $Q_2 = 0, Q_{next2} = 1$ จะต้องกำหนด $J_2 = 1$ และ $K_2 = X$
- 3) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 1$

ลำดับที่ 7: $Q_3Q_2Q_1 = 110$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 000$

- 1) $Q_3 = 1, Q_{next3} = 0$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 1$
- 2) $Q_2 = 1, Q_{next2} = 0$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 1$
- 3) $Q_1 = 0, Q_{next1} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$

ได้ตารางไค์โฮเทชั่นเป็นดังนี้

| สถานะปัจจุบัน ($Q_3Q_2Q_1$) | J_3 | K_3 | J_2 | K_2 | J_1 | K_1 |
|----------------------------------|-------|-------|-------|-------|-------|-------|
| 000 | 0 | X | 0 | X | 1 | X |
| 001 | 0 | X | 1 | X | X | 1 |
| 010 | 0 | X | X | 0 | 1 | X |
| 011 | 1 | X | X | 1 | X | 1 |
| 100 | X | 0 | 0 | X | 1 | X |
| 101 | X | 0 | 1 | X | X | 1 |
| 110 | X | 1 | X | 1 | 0 | X |

ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทั้งหมด

| | | | | | | |
|-------|----------|----|----|----|----|----------|
| | Q_3Q_2 | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | 0 | 0 | X | X | |
| | 1 | 0 | 1 | X | X | Q_1Q_2 |

$$J_3 = Q_1Q_2$$

| | | | | | | |
|-------|----------|----|----|----|----|-------|
| | Q_3Q_2 | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | X | X | 1 | 0 | |
| | 1 | X | X | X | 0 | Q_2 |

$$K_3 = Q_2$$

| | | | | | | |
|-------|----------|----|----|----|----|-------|
| | Q_3Q_2 | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | 0 | X | X | 0 | |
| | 1 | 1 | X | X | 1 | Q_1 |

$$J_2 = Q_1$$

| | | | | | | | |
|-------|-----------|---|----|----|----|----|-------|
| | $Q_3 Q_2$ | | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | X | 0 | 1 | X | X | Q_3 |
| | 1 | X | 1 | X | X | X | Q_1 |

$$K_2 = Q_1 + Q_3$$

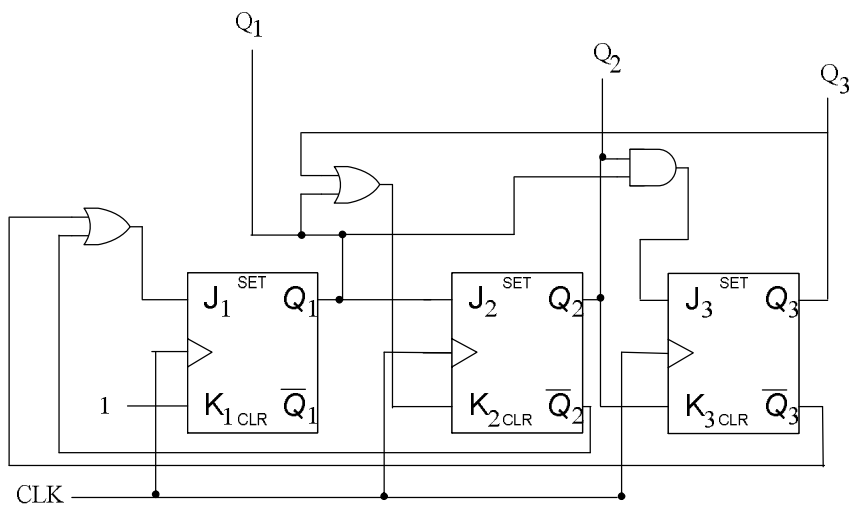
| | | | | | | | |
|-------|-----------|---|----|----|----|----|-------|
| | $Q_3 Q_2$ | | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | 1 | 1 | 0 | 1 | X | Q_2 |
| Q_3 | 1 | X | X | X | X | X | |

$$J_1 = \bar{Q}_2 + \bar{Q}_3$$

| | | | | | | | |
|-------|-----------|---|----|----|----|----|---|
| | $Q_3 Q_2$ | | 00 | 01 | 11 | 10 | |
| Q_1 | 0 | X | X | X | X | X | 1 |
| | 1 | 1 | 1 | X | 1 | 1 | |

$$K_1 = 1$$

ขั้นตอนที่ 4: สร้างวงจรจากสมการพีชคณิตบูลีนของอินพุตของฟลิปฟลอป



ตัวอย่างที่ 9-4 จงออกแบบวงจรนับ 7 ที่เป็นแบบนับลงแบบเข้าจังหวะ

วิธีทำ จากโจทย์ต้องการวงจรรนับ 7 แบบนับลงแสดงว่าเป็นวงจรรนับ 6 - 0 ซึ่งมีสถานะที่เป็นไปได้ทั้งหมด 7 สถานะ ดังนั้นต้องใช้ฟลิปฟล็อปทั้งหมด 3 ตัว

ขั้นตอนที่ 1: สร้างตารางทรานซิชันเพื่อหาสถานะของเอาต์พุตของฟลิปฟล็อปที่จะเกิดขึ้นใหม่

ลำดับที่ 1 สถานะปัจจุบันคือ "110" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "101"

ลำดับที่ 2 สถานะปัจจุบันคือ "101" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "100"

ลำดับที่ 3 สถานะปัจจุบันคือ "100" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "011"

ลำดับที่ 4 สถานะปัจจุบันคือ "011" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "010"

ลำดับที่ 5 สถานะปัจจุบันคือ "010" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "001"

ลำดับที่ 6 สถานะปัจจุบันคือ "001" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "000"

ลำดับที่ 7 สถานะปัจจุบันคือ "000" ดังนั้นสถานะเอาต์พุตของฟลิปฟล็อปทั้ง 3 ตัวที่จะเกิดขึ้นใหม่มีค่าเป็น "110"

ได้ตารางทรานซิชัน เป็นดังนี้

| สถานะปัจจุบัน ($Q_3Q_2Q_1$) | สถานะที่เกิดขึ้นใหม่ ($Q_{next3}Q_{next2}Q_{next1}$) |
|----------------------------------|---|
| 110 | 101 |
| 101 | 100 |
| 100 | 011 |
| 011 | 010 |
| 010 | 001 |
| 001 | 000 |
| 000 | 110 |

ขั้นตอนที่ 2: เลือกชนิดของฟลิปฟล็อปที่ต้องการใช้งาน แล้วทำการสร้างตารางเอ็กไซเทชันจกตารางทรานซิชัน

เนื่องจากโจทย์ไม่ได้กำหนดชนิดของฟลิปฟล็อปมาให้ ดังนั้นสามารถกำหนดชนิดของฟลิปฟล็อปได้เอง โดยผู้เขียนเลือกใช้ฟลิปฟล็อป JK ซึ่งต้องกำหนดค่าเป็นดังนี้

ลำดับที่ 1: $Q_3Q_2Q_1 = 110$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 101$

4) $Q_3 = 1, Q_{next3} = 1$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 0$

5) $Q_2 = 1, Q_{next2} = 0$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 1$

- 6) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_1 = 1$ และ $K_1 = X$
ลำดับที่ 2: $Q_3Q_2Q_1 = 101$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 100$
 4) $Q_3 = 1, Q_{next3} = 1$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 0$
 5) $Q_2 = 0, Q_{next2} = 0$ จะต้องกำหนด $J_2 = 0$ และ $K_2 = X$
 6) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_1 = X$ และ $K_1 = 1$
ลำดับที่ 3: $Q_3Q_2Q_1 = 100$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 011$
 4) $Q_3 = 1, Q_{next3} = 0$ จะต้องกำหนด $J_3 = X$ และ $K_3 = 1$
 5) $Q_2 = 0, Q_{next2} = 1$ จะต้องกำหนด $J_2 = 1$ และ $K_2 = X$
 6) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_1 = 1$ และ $K_1 = X$
ลำดับที่ 4: $Q_3Q_2Q_1 = 011$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 010$
 4) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
 5) $Q_2 = 1, Q_{next2} = 1$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 0$
 6) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_1 = X$ และ $K_1 = 1$
ลำดับที่ 5: $Q_3Q_2Q_1 = 010$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 001$
 4) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
 5) $Q_2 = 1, Q_{next2} = 0$ จะต้องกำหนด $J_2 = X$ และ $K_2 = 1$
 6) $Q_1 = 0, Q_{next1} = 1$ จะต้องกำหนด $J_1 = 1$ และ $K_1 = X$
ลำดับที่ 6: $Q_3Q_2Q_1 = 001$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 000$
 4) $Q_3 = 0, Q_{next3} = 0$ จะต้องกำหนด $J_3 = 0$ และ $K_3 = X$
 5) $Q_2 = 0, Q_{next2} = 0$ จะต้องกำหนด $J_2 = 0$ และ $K_2 = X$
 6) $Q_1 = 1, Q_{next1} = 0$ จะต้องกำหนด $J_1 = X$ และ $K_1 = 1$
ลำดับที่ 7: $Q_3Q_2Q_1 = 000$ ได้ $Q_{next3}Q_{next2}Q_{next1} = 110$
 4) $Q_3 = 0, Q_{next3} = 1$ จะต้องกำหนด $J_3 = 1$ และ $K_3 = X$
 5) $Q_2 = 0, Q_{next2} = 1$ จะต้องกำหนด $J_2 = 1$ และ $K_2 = X$
 6) $Q_1 = 0, Q_{next1} = 0$ จะต้องกำหนด $J_1 = 0$ และ $K_1 = X$

ได้ตารางไค์เทซเป็นดังนี้

| สถานะปัจจุบัน ($Q_3Q_2Q_1$) | J_3 | K_3 | J_2 | K_2 | J_1 | K_1 |
|----------------------------------|-------|-------|-------|-------|-------|-------|
| 110 | X | 0 | X | 1 | 1 | X |
| 101 | X | 0 | 0 | X | X | 1 |
| 100 | X | 1 | 1 | X | 1 | X |
| 011 | 0 | X | X | 0 | X | 1 |
| 010 | 0 | X | X | 1 | 1 | X |
| 001 | 0 | X | 0 | X | X | 1 |
| 000 | 1 | X | 1 | X | 0 | X |

ขั้นตอนที่ 3: สร้างแผนผังคาร์โนห์เพื่อหาสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อปทั้งหมด

| | | | | | |
|-----------|----|----|----|----|-------------|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| 0 | 1 | 0 | x | x | ← $Q_1 Q_2$ |
| 1 | 0 | 0 | x | x | |

$$J_3 = \bar{Q}_1 \bar{Q}_2$$

| | | | | | |
|-----------|----|----|----|----|-------------|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| 0 | x | x | 0 | 1 | ← $Q_1 Q_2$ |
| 1 | x | x | x | 0 | |

$$K_3 = \bar{Q}_1 \bar{Q}_2$$

| | | | | | |
|-----------|----|----|----|----|---------|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| 0 | 1 | x | x | 1 | ← Q_1 |
| 1 | 0 | x | x | 0 | |

$$J_2 = \bar{Q}_1$$

| | | | | | |
|-----------|----|----|----|----|---------|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| 0 | x | 1 | 1 | x | ← Q_1 |
| 1 | x | 0 | x | x | |

$$K_2 = \bar{Q}_1$$

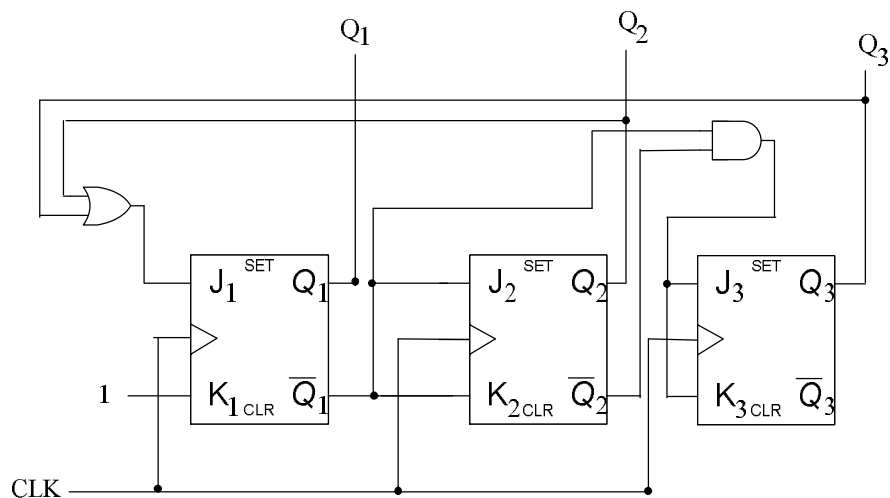
| | | | | | |
|-----------|----|----|----|----|---------|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| Q_2 | 0 | 1 | 1 | 1 | |
| 1 | x | x | x | x | ← Q_3 |

$$J_1 = Q_2 + Q_3$$

| | | | | | |
|-----------|----|----|----|----|-----|
| $Q_3 Q_2$ | 00 | 01 | 11 | 10 | |
| Q_1 | | | | | |
| 0 | x | x | x | x | ← 1 |
| 1 | 1 | 1 | x | 1 | |

$$K_1 = 1$$

ขั้นตอนที่ 4: สร้างวงจรจากสมการพีชคณิตบูลีนของอินพุตของฟลิปฟล็อป

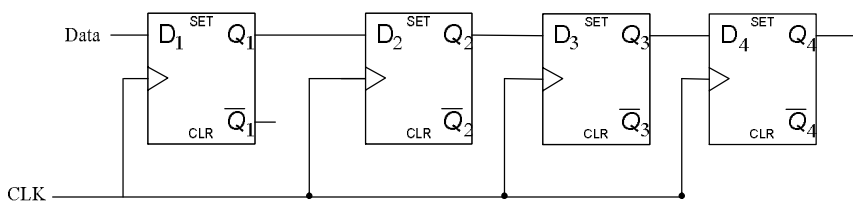


9.2 ชิฟท์รีจิสเตอร์

ชิฟท์รีจิสเตอร์ คือ วิธีที่ใช้สำหรับการถ่ายโอนข้อมูลจากฟลิปฟล็อปตัวหนึ่งไปยังฟลิปฟล็อปอีกตัวหนึ่ง ซึ่งแบ่งออกเป็น 2 วิธี คือ การถ่ายโอนข้อมูลแบบอนุกรม และการถ่ายโอนข้อมูลแบบขนาน

9.2.1 การถ่ายโอนข้อมูลแบบอนุกรม

การถ่ายโอนข้อมูลแบบอนุกรม คือ ข้อมูลจะถูกเลื่อนเข้าไปเป็นอินพุตให้ฟลิปฟล็อปครั้งละ 1 บิต และข้อมูลเอาต์พุตของฟลิปฟล็อปตัวดังกล่าวจะถูกส่งไปเป็นอินพุตให้ฟลิปฟล็อปตัวถัดไป โดยข้อมูลเอาต์พุตของวงจรจะออกมาตามลำดับของข้อมูลอินพุต โดยมีสัญญาณ CLK เป็นตัวควบคุมจังหวะในการป้อนอินพุต



รูปที่ 9.5 การถ่ายโอนข้อมูลแบบอนุกรม

จากรูปที่ 9.5 แสดงการถ่ายโอนข้อมูลแบบอนุกรมขนาด 4 บิตซึ่งจะมีการถ่ายโอนข้อมูลทุกครั้งที่เกิดสัญญาณ CLK โดยเมื่อเกิดสัญญาณ CLK จะมีการทำงานเป็นดังนี้

- ข้อมูลจากฟลิปฟล็อปตัวที่ 1 จะถูกส่งไปยังฟลิปฟล็อปตัวที่ 2
- ข้อมูลจากฟลิปฟล็อปตัวที่ 2 จะถูกส่งไปยังฟลิปฟล็อปตัวที่ 3
- ข้อมูลจากฟลิปฟล็อปตัวที่ 3 จะถูกส่งไปยังฟลิปฟล็อปตัวที่ 4

ตัวอย่างการทำงานเป็นดังนี้ กำหนดให้ **Data** มีที่จะส่งเข้าไปยังฟลิปฟล็อปตัวที่ 1 มีค่าเป็น "1011"

เมื่อเกิดสัญญาณ **CLK** ที่ 1:

1) ข้อมูลตัวที่ 1 "1" จะถูกส่งไปยัง D_1

เมื่อเกิดสัญญาณ **CLK** ที่ 2:

1) ข้อมูลจาก Q_1 "1" จะถูกส่งไปยัง D_2

2) ข้อมูลตัวที่ 2 "0" จะถูกส่งไปยัง D_1

เมื่อเกิดสัญญาณ **CLK** ที่ 3:

1) ข้อมูลจาก Q_2 "1" จะถูกส่งไปยัง D_3

2) ข้อมูลจาก Q_1 "0" จะถูกส่งไปยัง D_2

3) ข้อมูลตัวที่ 3 "1" จะถูกส่งไปยัง D_1

เมื่อเกิดสัญญาณ **CLK** ที่ 4:

1) ข้อมูลจาก Q_3 "1" จะถูกส่งไปยัง D_4

2) ข้อมูลจาก Q_2 "0" จะถูกส่งไปยัง D_3

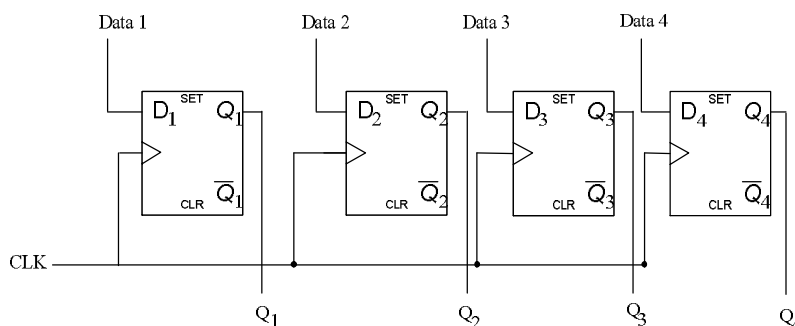
3) ข้อมูลจาก Q_1 "1" จะถูกส่งไปยัง D_2

4) ข้อมูลตัวที่ 4 "1" จะถูกส่งไปยัง D_1

สรุปได้ว่า $Q_1 = 1, Q_2 = 1, Q_3 = 0$ และ $Q_4 = 1$

9.2.2 การถ่ายโอนข้อมูลแบบขนาน

การถ่ายโอนข้อมูลแบบขนาน คือ ชุดข้อมูลจะถูกเลื่อนเข้าไปเป็นอินพุตให้ฟลิปฟล็อป แต่ละตัวพร้อมกันทั้งหมด ข้อมูลเอาต์พุตของวงจรถูกออกมาจะได้ออกมาพร้อมกันทั้งหมดโดยมีสัญญาณ **CLK** เป็นตัวควบคุมจังหวะในการป้อนอินพุต



รูปที่ 9.6 การถ่ายโอนข้อมูลแบบขนาน

จากรูปที่ 9.6 แสดงการถ่ายโอนข้อมูลแบบขนานขนาด 4 บิตซึ่งจะมีการถ่ายโอนข้อมูลเมื่อเกิดสัญญาณ **CLK** โดยเมื่อเกิดสัญญาณนาฬิกาจะมีการทำงานเป็นดังนี้

Data 1 จะถูกส่งไปยัง D_1

Data 2 จะถูกส่งไปยัง D_2

Data 3 จะถูกส่งไปยัง **D₃**

Data 4 จะถูกส่งไปยัง **D₄**

โดยที่ข้อมูลทั้ง 4 ตัวนี้จะถูกส่งไปเป็นอินพุตให้ฟลิปฟล็อปแต่ละตัวพร้อมกันทั้งหมด

9.3 บทสรุป

วงจรรีจิสเตอร์ ถูกสร้างได้โดยการต่อฟลิปฟล็อป และเกตพื้นฐานร่วมกัน ซึ่งมีทั้งแบบวงจรรีจิสเตอร์ขึ้น และวงจรรีจิสเตอร์ลงโดยวงจรรีจิสเตอร์แบ่งออกเป็น 2 ประเภทคือวงจรรีจิสเตอร์แบบไม่เข้าจังหวะคือวงจรรีจิสเตอร์ที่ฟลิปฟล็อปแต่ละตัวจะไม่ใช้สัญญาณ CLK ร่วมกัน การออกแบบจะกำหนดให้ฟลิปฟล็อปแต่ละตัวใช้สัญญาณ CLK ของฟลิปฟล็อปตัวที่อยู่ก่อนหน้า โดยจะเกิดการกลับบิตสัญญาณเอาต์พุตทุกครั้งที่เกิดสัญญาณ CLK ของฟลิปฟล็อปแต่ละตัว ดังนั้นวงจรรีจิสเตอร์นี้จึงเปรียบเสมือนวงจรรีจิสเตอร์ความถี่ได้ เนื่องจากวงจรรีจิสเตอร์แบบไม่เข้าจังหวะจะนับได้ในรูปของ 2^n เท่านั้น ดังนั้นหากต้องการให้วงจรรีจิสเตอร์สามารถนับได้ในรูปแบบอื่นที่ไม่ได้อยู่ในรูปของ 2^n จะต้องใช้สัญญาณ SET หรือ สัญญาณ CLR มาต่อใช้งานร่วมด้วย วงจรรีจิสเตอร์อีกประเภทคือวงจรรีจิสเตอร์แบบเข้าจังหวะ คือวงจรรีจิสเตอร์ที่ฟลิปฟล็อปทุกตัวจะใช้สัญญาณ CLK ร่วมกันโดยการออกแบบจะมีความยุ่งยากมากกว่าวงจรรีจิสเตอร์แบบไม่เข้าจังหวะเนื่องจากต้องมีการใช้มีการคำนวณหาสถานะที่จะเกิดขึ้นในแต่ละช่วงเวลาที่เป็นไปได้ของวงจรรีจิสเตอร์ทั้งหมด

ซีพรีรีจิสเตอร์ คือวิธีที่ใช้สำหรับการถ่ายโอนข้อมูลจากฟลิปฟล็อปตัวหนึ่งไปยังฟลิปฟล็อปอีกตัวหนึ่ง โดยการถ่ายโอนข้อมูลแบ่งออกเป็น 2 ประเภทคือแบบอนุกรมซึ่งข้อมูลจะถูกเลื่อนเข้าไปเป็นอินพุตให้ฟลิปฟล็อปครั้งละ 1 บิต และการถ่ายโอนข้อมูลแบบขนาน คือ ชุดข้อมูลจะถูกเลื่อนเข้าไปเป็นอินพุตให้ฟลิปฟล็อป แต่ละตัวพร้อมกันทั้งหมด

คำถามท้ายบท

1. จงออกแบบวงจรนับขึ้น และนับลงแบบนับ 16 (0 - 15) แบบไม่เข้าจังหวะดังนี้
 - 1.1) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปแบบ T ทั้งหมด
 - 1.2) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปแบบ JK ทั้งหมด
 - 1.3) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปแบบ T ทั้งหมด
 - 1.4) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปแบบ JK ทั้งหมด
 - 1.5) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปอย่างน้อย 2 ชนิด
 - 1.6) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปอย่างน้อย 2 ชนิด
2. จงออกแบบวงจรนับขึ้นแบบนับ 14 (0 - 13) แบบไม่เข้าจังหวะดังนี้
 - 2.1) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปแบบ T ทั้งหมด
 - 2.2) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปแบบ JK ทั้งหมด
 - 2.3) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปแบบ T ทั้งหมด
 - 2.4) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปแบบ JK ทั้งหมด
 - 2.5) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาลง และใช้ฟลิปฟล็อปอย่างน้อย 2 ชนิด
 - 2.6) วงจรเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น และใช้ฟลิปฟล็อปอย่างน้อย 2 ชนิด
3. จงออกแบบวงจรนับขึ้น แบบนับ 14 (0 - 13) แบบเข้าจังหวะ และเป็นวงจรที่เกิดสัญญาณนาฬิกาที่ขอบขาขึ้น ดังนี้
 - 3.1) ใช้ฟลิปฟล็อปแบบ RS ทั้งหมด
 - 3.2) ใช้ฟลิปฟล็อปแบบ D ทั้งหมด
 - 3.3) ใช้ฟลิปฟล็อปแบบ T ทั้งหมด
 - 3.4) ใช้ฟลิปฟล็อปแบบ JK ทั้งหมด
 - 3.5) ใช้ฟลิปฟล็อปอย่างน้อย 2 ชนิด
4. จงออกแบบวงจรนับลงแบบเข้าจังหวะโดยใช้ฟลิปฟล็อปแบบ JK ที่เกิดสัญญาณนาฬิกาที่ขอบขาขึ้น ทั้งหมด กำหนดให้เป็นวงจรที่นับเฉพาะเลขคี่เท่านั้นโดยนับจาก (9 - 1)
5. จงออกแบบวงจรนับขึ้นแบบเข้าจังหวะโดยใช้ฟลิปฟล็อปแบบ D ที่เกิดสัญญาณนาฬิกาที่ขอบขาขึ้น ทั้งหมด กำหนดให้เป็นวงจรที่นับเฉพาะเลขคู่เท่านั้นโดยนับจาก (0 - 12)
6. จากรูปที่ 9-5 กำหนดให้ Data มีที่จะส่งเข้าไปยังฟลิปฟล็อปตัวที่ 1 ในแต่ละช่วงเวลามีค่าเป็น "11001" จงหาสถานะ Q_1 , Q_2 , Q_3 และ Q_4 หลังจากเกิดสัญญาณนาฬิกาทั้งหมด 4 ครั้ง พร้อมอธิบายการทำงานในแต่ละช่วงเวลาอย่างละเอียด

เอกสารอ้างอิง

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.

ธวัชชัย เลื่อนฉวี. (2532). **ดิจิทัลเทคนิคเล่ม 2**. กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิทัล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุดรธานี: มหาวิทยาลัยราชภัฏอุดรธานี.**

แผนบริหารการสอนประจำบทที่ 10 วงจรเข้ารหัสและวงจรถอดรหัส 1 ชั่วโมง 30 นาที

หัวข้อเนื้อหา

- 10.1 วงจรเข้ารหัส
- 10.2 วงจรถอดรหัส
- 10.3 แอลอีดี 7 ส่วน
- 10.4 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการออกแบบวงจรเข้ารหัสและวงจรถอดรหัส
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการใช้งานหลอดแอลอีดี 7 ส่วน และการใช้งานร่วมกับวงจรถอดรหัส

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรถิศจีห์ลและลลจก
2. ภาพเล็ลน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 10

วงจรเข้ารหัสและวงจรถอดรหัส

จากที่เคยได้กล่าวไว้แล้วว่าวงจรดิจิทัลมีการประมวลผลเพียงเลขรหัสฐานสอง (0 และ 1) เท่านั้น แต่ตัวเลขที่มนุษย์ใช้งานในชีวิตประจำวันคือเลขฐานสิบ (0 - 9) ดังนั้นวงจรเข้ารหัส และวงจรถอดรหัส เป็นวงจรที่ใช้สำหรับแปลงสัญญาณ เพื่อให้การสื่อสารระหว่างมนุษย์และวงจรดิจิทัลง่ายขึ้น ตัวอย่างเช่น มนุษย์จะส่งข้อมูลอินพุตที่เป็นเลขฐานสิบ เมื่อวงจรดิจิทัลได้รับข้อมูลดังกล่าวจะแปลงเป็นเลขฐานสองเพื่อนำข้อมูลไปประมวลผล หลังจากผ่านการประมวลผลแล้ว ผลลัพธ์ที่ได้ยังคงเป็นรหัสที่เป็นเลขฐานสอง ดังนั้นวงจรจะแปลงรหัสดังกล่าวกลับเป็นเลขฐานสิบเพื่อที่จะนำผลลัพธ์ที่ได้ส่งไปแสดงผลให้แก่มนุษย์

10.1 วงจรเข้ารหัส

วงจรเข้ารหัส (Encoder) คือวงจรที่รับข้อมูลอินพุตจากผู้ใช้งาน ซึ่งข้อมูลอินพุตอาจเป็นตัวเลขฐานสิบ หรือ ตัวอักษร โดยวงจรเข้ารหัสจะนำข้อมูลอินพุตมาแปลงเป็นรหัสที่เป็นเลขฐานสอง เพื่อที่จะทำให้ข้อมูลดังกล่าวถูกประมวลผลได้ ยกตัวอย่างเช่น เมื่อผู้ใช้งานทำการกดตัวเลข หรือ ตัวอักษรที่อยู่แป้นพิมพ์ ข้อมูลดังกล่าวจะถูกส่งเข้าวงจรเข้ารหัสเพื่อแปลงให้อยู่ในรูปแบบของรหัสเลขฐานสอง เป็นต้น

ตัวอย่างที่ 10-1 จงออกแบบวงจรเข้ารหัสที่ใช้สำหรับแปลงเลขฐานสิบ 4 ตัว (0 - 3) เป็นรหัสเลขฐานสอง

วิธีทำ จากโจทย์ต้องการวงจรแปลงตัวเลขทั้งหมด 4 ตัวคือ 0 - 3 เพื่อเป็นรหัสเลขฐานสอง เนื่องจากสถานะที่เป็นไปได้ทั้งหมดของอินพุตมี 4 กรณี ดังนั้นเอาต์พุตของวงจรจะมีขนาด 2 บิต โดยโครงสร้างของวงจรเป็นดังนี้



จากโครงสร้างของวงจร E คือสัญญาณที่ใช้เพื่อแสดงให้เห็นว่ามีการกดปุ่มที่ทางอินพุตกำหนดให้สถานะ "1" แทนการกดปุ่มอินพุตและ มีการเกิดสัญญาณที่ทางเอาต์พุต และสถานะ "0" แทนสถานะของอินพุตที่ยังไม่มีการกดปุ่ม และ ยังไม่เกิดสัญญาณที่ทางเอาต์พุต ซึ่งสามารถออกแบบตารางความจริงได้ดังนี้

| อินพุต (ปุ่ม) | สถานะอินพุต | | | | เอาต์พุต | | |
|------------------|-------------|---|---|---|----------------|----------------|---|
| | 3 | 2 | 1 | 0 | A ₁ | A ₀ | E |
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

จากตารางความจริง อธิบายการทำงานได้ดังนี้

- แถวที่ 1: ยังไม่มีการกดปุ่มใดๆ เลย, สถานะของเอาต์พุตเป็น 0 ทั้งหมด
- แถวที่ 2: กดปุ่ม "0", สถานะของ A₁ = 0, A₀ = 0 และ E = 1
- แถวที่ 3: กดปุ่ม "1", สถานะของ A₁ = 0, A₀ = 1 และ E = 1
- แถวที่ 4: กดปุ่ม "2", สถานะของ A₁ = 1, A₀ = 0 และ E = 1
- แถวที่ 5: กดปุ่ม "3", สถานะของ A₁ = 1, A₀ = 1 และ E = 1

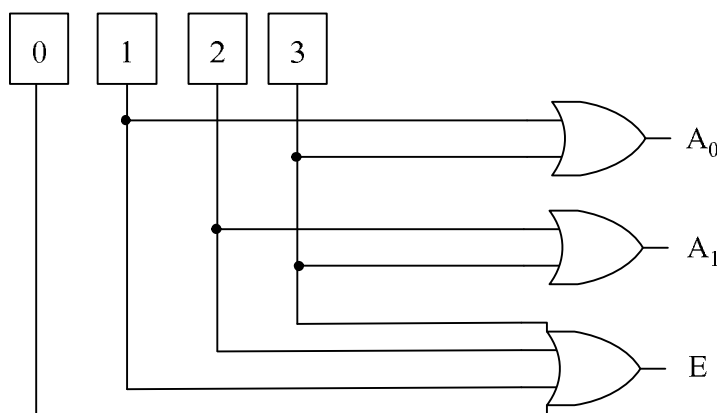
หาสมการพีชคณิตบูลีนของเอาต์พุตทั้ง 3 ตัวได้ ดังนี้

A₀ จะมีสถานะเป็น 1 ได้ก็ต่อเมื่อมีการกดปุ่ม 1 หรือ ปุ่ม 3 ดังนั้น A₀ = ปุ่ม 1 + ปุ่ม 3

A₁ จะมีสถานะเป็น 1 ได้ก็ต่อเมื่อมีการกดปุ่ม 2 หรือ ปุ่ม 3 ดังนั้น A₁ = ปุ่ม 2 + ปุ่ม 3

E จะมีสถานะเป็น 1 ได้ก็ต่อเมื่อมีการกดปุ่ม 0, ปุ่ม 1, ปุ่ม 2 หรือ ปุ่ม 3 ดังนั้น E = ปุ่ม 0 + ปุ่ม 1 + ปุ่ม 2 + ปุ่ม 3

ได้วงจรรหัสที่ใช้สำหรับแปลงเลขฐานสิบ 4 ตัว (0 - 3) เป็นรหัสเลขฐานสองเป็นดังนี้



10.2 วงจรรถอตรหัส

วงจรรถอตรหัส (Decoder) คือวงจรรหัสที่ใช้แปลงจากรหัสเลขฐานสอง เป็นตัวเลขฐานสิบ หรือตัวอักษรก่อนจะแสดงผลที่ได้แก่ผู้ใช้งาน โดยการใช้งานวงจรรถอตรหัสส่วนใหญ่จะเกิดขึ้นหลังจาก

วงจรหลักได้ทำการประมวลผลแล้ว แต่เนื่องจากผลลัพธ์ที่ได้ยังคงเป็นรหัสที่เป็นเลขฐานสองอยู่ จึงนำผลลัพธ์ดังกล่าวมาผ่านวงจรถอดรหัสก่อนจะแสดงผลแก่ผู้ใช้งาน

ตัวอย่างที่ 10-2 จงออกแบบวงจรถอดรหัสที่ใช้สำหรับแปลงรหัสเลขฐานสอง 4 ตัว (00 - 11) เป็นรหัสเลขฐานสิบ

วิธีทำ จากโจทย์ต้องการวงจรแปลงตัวเลขทั้งหมด 4 ตัวคือ 00 - 11 เพื่อเป็นรหัสเลขฐานสิบ ดังนั้นเอาต์พุตของวงจรจะมีจำนวน 4 ตัวโดย โครงสร้างของวงจรเป็นดังนี้



การออกแบบตารางความจริงได้ ดังนี้

| อินพุต | | เอาต์พุต | | | |
|--------|-------|----------|-------|-------|-------|
| A_1 | A_0 | Z_0 | Z_1 | Z_2 | Z_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 |

จากตารางความจริง อธิบายการทำงานได้ดังนี้

แถวที่ 1: อินพุต 00, สถานะเอาต์พุตมีค่าเป็น 0 ($Z_0 = 1, Z_1 = 0, Z_2 = 0$ และ $Z_3 = 0$)

แถวที่ 2: อินพุต 01, สถานะเอาต์พุตมีค่าเป็น 1 ($Z_0 = 0, Z_1 = 1, Z_2 = 0$ และ $Z_3 = 0$)

แถวที่ 3: อินพุต 10, สถานะเอาต์พุตมีค่าเป็น 2 ($Z_0 = 0, Z_1 = 0, Z_2 = 1$ และ $Z_3 = 0$)

แถวที่ 4: อินพุต 11, สถานะเอาต์พุตมีค่าเป็น 3 ($Z_0 = 0, Z_1 = 0, Z_2 = 0$ และ $Z_3 = 1$)

หาสมการพีชคณิตบูลีนของเอาต์พุตทั้ง 3 ตัวได้ ดังนี้

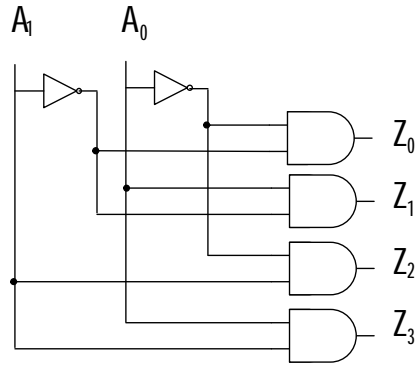
Z_0 จะมีค่าเป็น 1 ได้ก็ต่อเมื่อ $A_1 = 0$ และ $A_0 = 0$ ($Z_0 = \bar{A}_1 \bar{A}_0$)

Z_1 จะมีค่าเป็น 1 ได้ก็ต่อเมื่อ $A_1 = 0$ และ $A_0 = 1$ ($Z_1 = \bar{A}_1 A_0$)

Z_2 จะมีค่าเป็น 1 ได้ก็ต่อเมื่อ $A_1 = 1$ และ $A_0 = 0$ ($Z_2 = A_1 \bar{A}_0$)

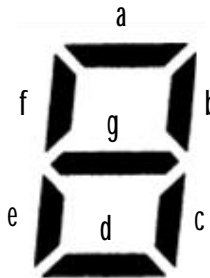
Z_3 จะมีค่าเป็น 1 ได้ก็ต่อเมื่อ $A_1 = 1$ และ $A_0 = 1$ ($Z_3 = A_1 A_0$)

ได้วงจรถอดรหัสที่ใช้สำหรับแปลงรหัสเลขฐานสอง 4 ตัว (00 - 11) เป็นรหัสเลขฐานสิบเป็นดังนี้



10.3 แอลอีดี 7 ส่วน

แอลอีดี 7 ส่วน (Seven – Segment) คือการนำเอาหลอดแอลอีดีจำนวน 7 ตัวมาประกอบกัน เพื่อให้การแสดงผลเปรียบเสมือนว่าเป็นตัวเลขฐานสิบ ซึ่งจะทำให้ผู้ใช้งานเห็นผลลัพธ์ได้ง่าย และชัดเจนมากยิ่งขึ้น



รูปที่ 10.1 โครงสร้างแอลอีดี 7 ส่วน

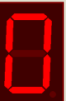









จากรูป 10.1 แสดงโครงสร้างแอลอีดี 7 ส่วน โดยผู้ใช้งานสามารถเลือกการติดตั้งของหลอดแอลอีดีในแต่ละส่วนได้ เพื่อให้การแสดงผลเป็นไปตามที่ต้องการ ปัจจุบันแอลอีดี 7 ส่วนแบ่งออกเป็น 2 ชนิด คือ แอลอีดีแบบคอมมอนแคโทด (Common Cathode) เป็นการนำขาข้างหนึ่งของหลอดแอลอีดีทั้ง 7 ขามาต่อร่วมกันลงกราวด์ แอลอีดี 7 ส่วนลักษณะนี้ มีความหมายคือ สถานะ "1" คือ สถานะไฟติดของหลอดแอลอีดี และ สถานะ "0" คือ สถานะไฟดับของหลอดแอลอีดี และแอลอีดี 7 ส่วนอีกชนิดหนึ่งคือ แอลอีดีแบบคอมมอนแอนโนด (Common Anode) เป็นการนำขาข้างหนึ่งของหลอดแอลอีดีทั้ง 7 ขามาต่อร่วมกันกับไฟเลี้ยง 5 โวลต์ แอลอีดี 7 ส่วนลักษณะนี้ มีความหมายคือ สถานะ "0" คือ สถานะไฟติดของหลอดแอลอีดี และ สถานะ "1" สถานะไฟดับของหลอดแอลอีดี

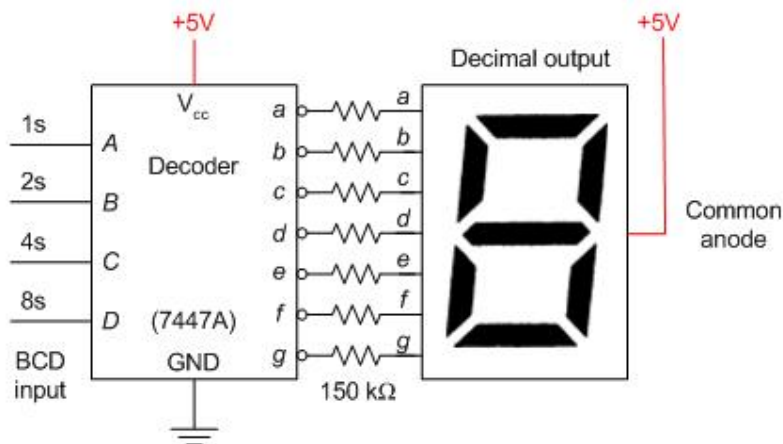
การกำหนดให้แอลอีดี 7 ส่วนแบบคอมมอนแคโทดมาแสดงเป็นเลข 0 – 9 สามารถทำได้ดังตารางที่ 10.1 (หากเป็นแอลอีดี 7 ส่วนแบบคอมมอนแอนโนดสถานะทำได้โดยใช้สถานะเอาต์พุตที่ตรงข้ามกับแอลอีดีแบบคอมมอนแคโทด)

อย่างไรก็ตาม บริษัทผู้ผลิตได้มีการผลิตไอซีตระกูลที่แอลที่ทำหน้าที่เป็นวงจรถอดรหัสที่สามารถแปลงจากรหัสต่าง ๆ บางประเภทที่เป็นรหัสเลขฐานสองเป็นรหัสที่ใช้สำหรับขับแอลอีดี 7 ส่วนได้ เช่น วงจรถอดรหัสที่แปลงจากรหัส BCD เป็นเอาต์พุตที่แสดงเป็นแอลอีดี 7 ส่วนจะใช้ไอซีเบอร์ 7446, 7447, 7448 หรือ 7449 เป็นต้น หรือวงจรถอดรหัสที่แปลงจากรหัสเพิ่ม 3 เป็นเอาต์พุต

ที่แสดงเป็นแอลอีดี 7 ส่วนจะใช้ไอซีเบอร์ 7443 หรือ 7444 เป็นต้น ทำให้ผู้ใช้งานไม่จำเป็นต้องสร้างวงจรถอดรหัสดังกล่าวขึ้นมาใช้งานเอง

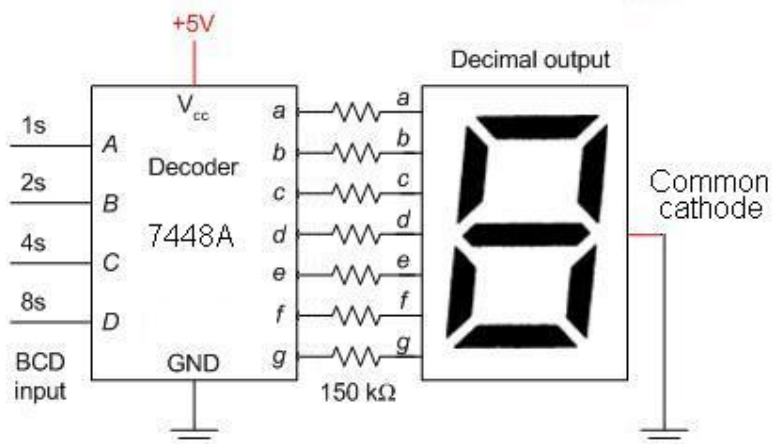
ตารางที่ 10.1 การแสดงผลเลข 0-9 ด้วยแอลอีดี 7 ส่วนแบบคอมมอนแคโทด

| หมายเลข | การแสดงผล | เอาต์พุต | | | | | | |
|---------|---|----------|---|---|---|---|---|---|
| | | g | f | e | d | c | b | a |
| 0 |  | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 |  | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 2 |  | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 3 |  | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 |  | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 5 |  | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 6 |  | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 7 |  | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8 |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 |  | 1 | 1 | 0 | 1 | 1 | 1 | 1 |



รูปที่ 10.2 การใช้งานไอซีเบอร์ 7447A ร่วมกับแอลอีดี 7 ส่วนแบบคอมมอนแอนโนด
ที่มาของภาพ: <http://elektronika-dasar.web.id/teori-elektronika/dekoder-ttl-bcd-ke-7-segment/>

รูปที่ 10.2 แสดงการใช้งานไอซีเบอร์ 7447A ร่วมกับแอลอีดี 7 ส่วนแบบคอมมอนแอนโนดซึ่งอินพุตของไอซีเบอร์ 7447A จะเป็นรหัส BCD ส่วนเอาต์พุตจะใช้สำหรับขับแอลอีดี 7 ส่วน ซึ่งต้องใช้แอลอีดีแบบคอมมอนแอนโนด เนื่องจากสัญญาณเอาต์พุตของไอซีเบอร์ 7447A ใช้สถานะ "0" ในการขับสัญญาณแอลอีดี 7 ส่วนเพื่อให้ไฟติด (มีสัญลักษณ์ "0" ตรงสัญญาณเอาต์พุต)

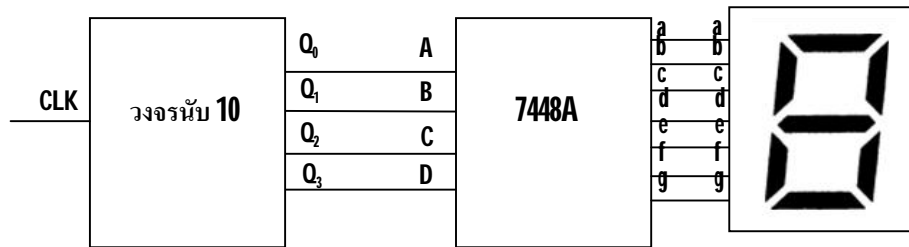


รูปที่ 10.3 การใช้งานไอซีเบอร์ 7448A ร่วมกับแอลอีดี 7 ส่วนแบบคอมมอนแคโทด
ที่มาของภาพ: <http://elektronika-dasar.web.id/teori-elektronika/dekoder-ttl-bcd-ke-7-segment/>

รูปที่ 10.3 แสดงการใช้งานไอซีเบอร์ 7448A ร่วมกับแอลอีดี 7 ส่วนแบบคอมมอนแคโทดซึ่งอินพุตของไอซีเบอร์ 7448A จะเป็นรหัส BCD ส่วนเอาต์พุตจะใช้สำหรับขับแอลอีดี 7 ส่วน ซึ่งต้องใช้แอลอีดีแบบคอมมอนแคโทด เนื่องจากสัญญาณเอาต์พุตของไอซีเบอร์ 7448A ใช้สถานะ "1" ในการขับสัญญาณแอลอีดี 7 ส่วนเพื่อให้ไฟติด

ตัวอย่างที่ 10-3 กำหนดให้มีอุปกรณ์อิเล็กทรอนิกส์อยู่ 3 ชนิดคือ 1 วงจรนับ 10 แบบนับขึ้นและเข้าจังหวะที่มีเอาต์พุต 4 ตัวคือ $Q_0 - Q_3$ โดยที่ Q_0 คือ ตำแหน่งของบิตที่มีลำดับความสำคัญต่ำที่สุด และ Q_3 คือ ตำแหน่งของบิตที่มีลำดับความสำคัญสูงที่สุด อุปกรณ์ที่ 2 คือ ไอซีเบอร์ 7448 ซึ่งใช้แปลงจากรหัส BCD - 8421 เพื่อขับสัญญาณเอาต์พุตเป็นแอลอีดี 7 ส่วน โดยมีอินพุต 4 ตัวคือ A, B, C และ D ซึ่ง A คือ ตำแหน่งของบิตที่มีลำดับความสำคัญต่ำที่สุด และ D คือ ตำแหน่งของบิตที่มีลำดับความสำคัญสูงที่สุดโดยใช้สถานะ "1" ในการขับสัญญาณ และอุปกรณ์ที่ 3 คือแอลอีดี 7 ส่วนแบบคอมมอนแคโทด จงสร้างวงจรวงจรวจรนับ 10 แบบนับขึ้นและเข้าจังหวะที่มีการแสดงผลที่แอลอีดี 7 ส่วน

วิธีทำ สมมติว่าทั้ง 3 วงจรได้มีการเชื่อมต่อที่ไฟเลี้ยงขนาด 5 โวลต์ และกราวด์ทั้งหมดแล้ว ได้วงจรวจรนับ 10 แบบนับขึ้นและเข้าจังหวะที่มีการแสดงผลที่แอลอีดี 7 ส่วน เป็นดังนี้



จากรูป นำตำแหน่งบิตที่มีลำดับความสำคัญเท่ากันระหว่างเอาต์พุตของวงจรวจรนับ 10 และอินพุตของไอซีเบอร์ 7448A มาเชื่อมต่อกัน เนื่องจากเอาต์พุตของวงจรวจรนับ 10 จะต้องถูกนำมาเป็นข้อมูลอินพุตแก่ ไอซีเบอร์ 7448A เป็นรหัส BCD - 8421 และสุดท้ายนำเอาต์พุตของไอซีเบอร์ 7448A เชื่อมต่อกับแอลอีดี 7 ส่วนแบบคอมมอนแคโทดเพื่อให้มีการแสดงผลที่แอลอีดี 7 ส่วนแบบคอมมอนแคโทด

10.4 บทสรุป

วงจรรหัสเข้ารหัส และวงจรรหัสออกรหัส คือ ส่วนที่ช่วยให้การสื่อสารระหว่างมนุษย์และวงจรดิจิทัลสามารถทำได้ง่ายมากยิ่งขึ้น โดยที่วงจรรหัสเข้ารหัสจะเป็นวงจรที่ทำหน้าที่ในการแปลงเลขฐานจากข้อมูลอินพุตซึ่งไม่ใช่เลขฐานสองให้เป็นรหัสที่เป็นเลขฐานสองก่อนที่จะส่งรหัสเลขฐานสองดังกล่าวเข้าไปประมวลผลเนื่องจากวงจรสามารถประมวลผลรหัสที่เป็นเลขฐานสองได้เท่านั้น เมื่อวงจรประมวลผลเสร็จจะได้สัญญาณเอาต์พุตที่เป็นเลขฐานสอง เพื่อให้มนุษย์สามารถอ่านผลการทำงานได้ง่ายมากยิ่งขึ้นจึงใช้วงจรรหัสออกรหัสเพื่อแปลงสัญญาณเอาต์พุตดังกล่าว (สัญญาณอินพุตของวงจรรหัสออกรหัส) ให้เป็นรหัสเลขฐานสิบซึ่งอาจจะใช้แอลอีดีแบบ 7 ส่วนมาช่วยสำหรับการแสดงผลเพื่อให้มีการแสดงผลเป็นตัวเลขได้

คำถามท้ายบท

1. จงบอกประโยชน์ของการใช้งานวงจรถ่ายรหัส และวงจรถอดรหัส
2. จงยกตัวอย่างการนำวงจรถ่ายรหัส และวงจรถอดรหัสไปประยุกต์ใช้งานจริง
3. จงออกแบบวงจรถ่ายรหัสที่ใช้สำหรับแปลงเลขฐานสิบ 8 ตัว (0 – 7) เป็นรหัสเลขฐานสอง
4. จงออกแบบวงจรถ่ายรหัสที่ใช้สำหรับแปลงเลขฐานสิบ 10 ตัว (0 – 9) เป็นรหัสเลขฐานสอง
5. แอลอีตี 7 ส่วนคืออะไร และมีประโยชน์อย่างไร
6. จงยกตัวอย่างอุปกรณ์ที่มีการใช้งานจริง ที่มีการนำแอลอีตี 7 ส่วนมาประยุกต์ใช้งานด้วยอย่างน้อย 3 ชนิด
7. แอลอีตี 7 ส่วนแบบคอมมอนแคโทด และ แอลอีตี 7 ส่วนแบบคอมมอนแอนโนดมีความแตกต่างกันอย่างไร
8. จงเขียนตารางความจริงเพื่อให้แอลอีตี 7 ส่วนแสดงตัวอักษรภาษาอังกฤษ 6 ตัวดังนี้ A, b, c, d, E และ F กำหนดให้ใช้แอลอีตี 7 ส่วนแบบคอมมอนแอนโนด
9. จงออกแบบวงจรถอดรหัสที่ใช้สำหรับแปลงรหัสเลขฐานสอง 8 ตัว (000 – 111) เป็นรหัสเลขฐานสิบ
10. จงออกแบบวงจรถอดรหัสที่ใช้สำหรับแปลงรหัสเลขฐานสอง 16 ตัว (0000 – 1111) เป็นรหัสเลขฐานสิบ
11. จากคำถามท้ายบท บทที่ 9 ข้อ 4 จงใช้แอลอีตี 7 ส่วนแบบคอมมอนแคโทดไปประยุกต์ใช้งานในส่วนของการแสดงผล

เอกสารอ้างอิง

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.

ธวัชชัย เลื่อนฉวี. (2532). **ดิจิทัลเทคนิคเล่ม 2**. กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิทัล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุดรธานี: มหาวิทยาลัยราชภัฏอุดรธานี.**
<http://elektronika-dasar.web.id/teori-elektronika/dekoder-ttl-bcd-ke-7-segment/>

แผนบริหารการสอนประจำบทที่ 11 วงจรมัลติเพล็กซ์และดีมัลติเพล็กซ์ 1 ชั่วโมง 30 นาที

หัวข้อเนื้อหา

- 11.1 วงจรมัลติเพล็กซ์
- 11.2 วงจรดีมัลติเพล็กซ์
- 11.3 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการออกแบบวงจรมัลติเพล็กซ์และวงจรดีมัลติเพล็กซ์

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

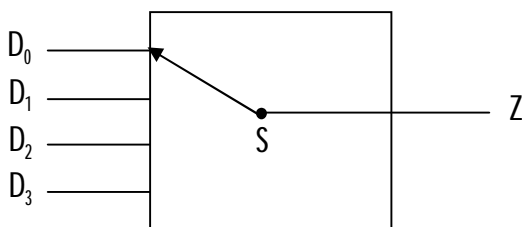
บทที่ 11

วงจรมัลติเพล็กซ์เซอร์และวงจรมัลติเพล็กซ์เซอร์

ในบทนี้จะกล่าวถึงวงจรที่ใช้สำหรับการเลือกข้อมูลอินพุตที่มีหลายช่องทางมาเพียงครั้งละ 1 ค่าจาก 1 ช่องทาง เรียกวงจรลักษณะนี้ว่าวงจรมัลติเพล็กซ์เซอร์ (Multiplexer) และวงจรที่ใช้สำหรับส่งข้อมูลอินพุตที่รับมาจากวงจรมัลติเพล็กซ์เซอร์ 1 ค่าออกไปยังช่องทางที่ถูกต้องเพียง 1 ช่องทางจากจำนวนหลายช่องทางเรียกวงจรลักษณะนี้ว่าวงจรมัลติเพล็กซ์เซอร์ (Demultiplexer) โดยที่วงจรมัลติเพล็กซ์เซอร์ และวงจรมัลติเพล็กซ์เซอร์จะใช้ตัวเลือกข้อมูล (Data selection) สำหรับเลือกข้อมูลอินพุตจากช่องสัญญาณที่ต้องการ และส่งข้อมูลดังกล่าวไปยังช่องสัญญาณเอาต์พุตที่ต้องการได้อย่างถูกต้อง

11.1 วงจรมัลติเพล็กซ์เซอร์

วงจรมัลติเพล็กซ์เซอร์ หรือเรียกสั้นๆว่า MUX เป็นวงจรที่มีหลายอินพุต แต่มีเอาต์พุตเพียง 1 ค่าเท่านั้น โดย ณ เวลาใดเวลาหนึ่ง วงจรจะมีการเลือกอินพุตมาเพียง 1 ค่า และส่งผ่านทางเอาต์พุตที่มีเพียงช่องทางเดียว โดยใช้อินพุตอีกกลุ่มหนึ่งเป็นตัวเลือกของสัญญาณว่าจะเลือกใช้งานอินพุตตัวใด

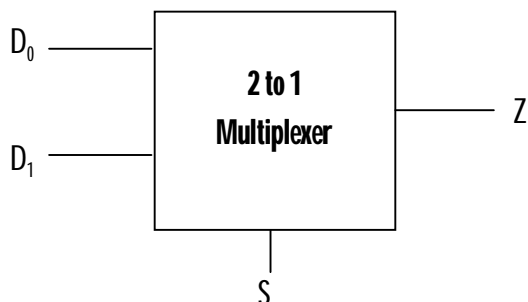


รูปที่ 11.1 โครงสร้างวงจรมัลติเพล็กซ์เซอร์แบบ 4 อินพุต

รูปที่ 11.1 แสดงโครงสร้างของวงจรมัลติเพล็กซ์เซอร์แบบ 4 อินพุตประกอบด้วย D_0 , D_1 , D_2 และ D_3 มีเอาต์พุตคือ Z และมี S (Data selection) เป็นตัวที่ใช้สำหรับการเลือกอินพุต โดยที่จำนวนบิตของ S สามารถคำนวณหาได้จาก $2^n =$ จำนวนอินพุต เมื่อ n คือจำนวนบิตของ S ที่ต้องใช้งานในวงจรดังนั้นจากรูปที่ 11.1 จำนวนบิตของ S ที่จำเป็นต้องใช้งานทั้งหมดคือ 2 บิต ($2^2 = 4$)

ตัวอย่างที่ 11.1: จงออกแบบวงจรมัลติเพล็กซ์เซอร์แบบ 2 อินพุต

วิธีทำ ก่อนออกแบบจำเป็นต้องเลือกจำนวนบิตของ S เพื่อใช้สำหรับเลือกอินพุต จากโจทย์อินพุตมีทั้งหมด 2 ตัวดังนั้น ต้องใช้ S ที่มีขนาด 1 บิต ($2^1 = 2$) จึงได้โครงสร้างเป็นดังนี้

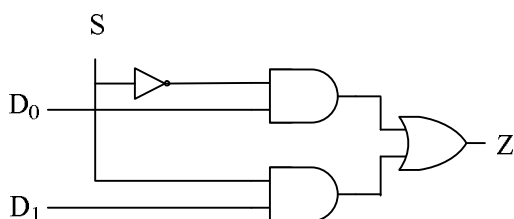


เมื่อได้โครงสร้างของวงจรแล้ว ขั้นตอนต่อไปกำหนดสถานะของ S และสมการพีชคณิตบูลีนของ Z ดังนี้

กรณี $S = 0$: เลือก $D_0 \rightarrow Z = SD_0$

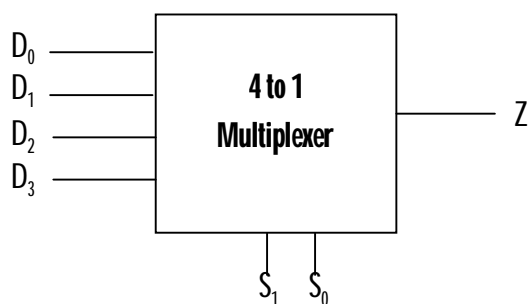
กรณี $S = 1$: เลือก $D_1 \rightarrow Z = SD_1$

ได้วงจรมัลติเพล็กซ์แบบ 2 อินพุต เป็นดังนี้



ตัวอย่างที่ 11.2: จงออกแบบวงจรมัลติเพล็กซ์แบบ 4 อินพุต

วิธีทำ ก่อนออกแบบจำเป็นต้องเลือกจำนวนบิตของ S เพื่อใช้สำหรับเลือกอินพุต จากโจทย์อินพุตมีทั้งหมด 4 ตัว ดังนั้นต้องใช้ S ที่มีขนาด 2 บิต ($2^2 = 4$) กำหนดให้เป็น S_0 และ S_1 ได้โครงสร้างเป็นดังนี้



เมื่อได้โครงสร้างของวงจรแล้ว ขั้นตอนต่อไปกำหนดสถานะของ S_0 และ S_1 และสมการพีชคณิตบูลีนของ Z ดังนี้

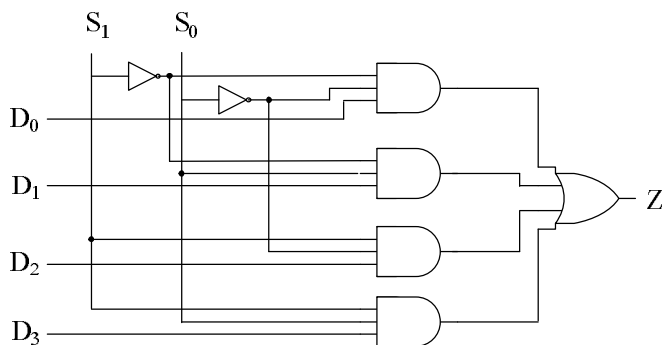
กรณี $S = 00$: เลือก $D_0 \rightarrow Z = S_1 S_0 D_0$

กรณี $S = 01$: เลือก $D_1 \rightarrow Z = S_1 S_0 D_1$

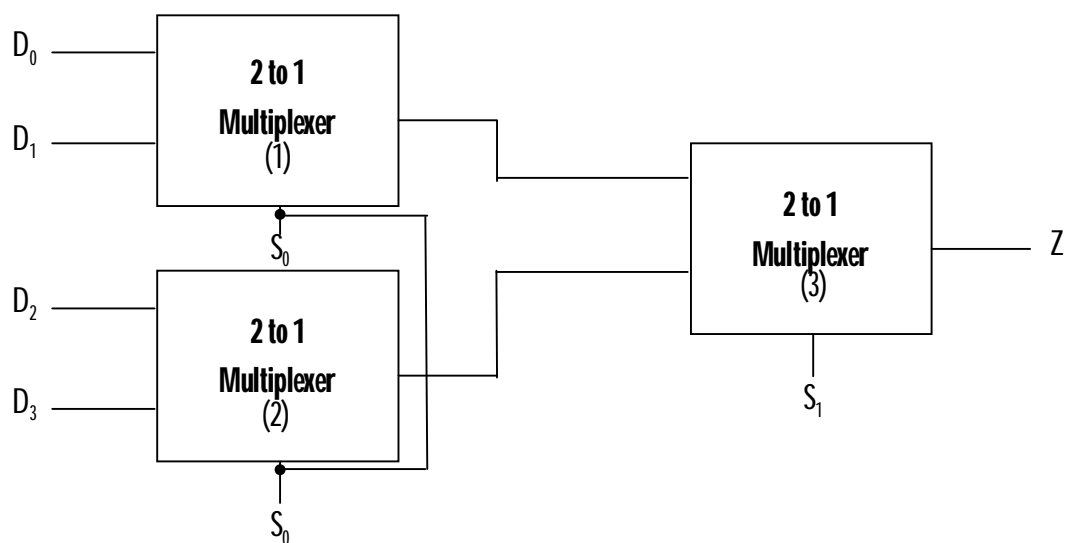
กรณี $S = 10$: เลือก $D_2 \rightarrow Z = S_1 S_0 D_2$

กรณี $S = 11$: เลือก $D_3 \rightarrow Z = S_1 S_0 D_3$

ได้วงจรมัลติเพล็กซ์แบบ 4 อินพุต เป็นดังนี้



ตัวอย่างที่ 11.3: จงออกแบบวงจรมัลติเพล็กซ์แบบ 4 อินพุตโดยใช้มัลติเพล็กซ์แบบ 2 อินพุต
วิธีทำ



จากรูป สรุปได้ว่าการสร้างวงจรมัลติเพล็กซ์แบบ 4 อินพุตโดยใช้วงจรมัลติเพล็กซ์แบบ 2 อินพุตจะต้องใช้วงจรมัลติเพล็กซ์ทั้งหมด 3 ตัว โดยที่วงจรที่ 1 และ วงจรที่ 2 จะใช้สัญญาณ S_0 ร่วมกัน

กำหนดให้ $S_0 = 0$, เป็นการเลือก D_0 จากวงจรที่ 1 และ เลือก D_2 จากวงจรที่ 2 เพื่อเป็นอินพุตให้วงจรที่ 3

กำหนดให้ $S_0 = 1$, เป็นการเลือก D_1 จากวงจรที่ 1 และ เลือก D_3 จากวงจรที่ 2 เพื่อเป็นอินพุตให้วงจรที่ 3

กำหนดให้ $S_1 = 0$, วงจรที่ 3 เลือกอินพุตที่มาจากเอาต์พุตของวงจรที่ 1

กำหนดให้ $S_1 = 1$, วงจรที่ 3 เลือกอินพุตที่มาจากเอาต์พุตของวงจรที่ 2

ดังนั้นการกำหนด S_0 และ S_1 เพื่อเลือกอินพุตทั้ง 4 ตัว คือ

กรณีที่ 1: เลือก D_0 , กำหนด $S_0 = 0$ และ $S_1 = 0$

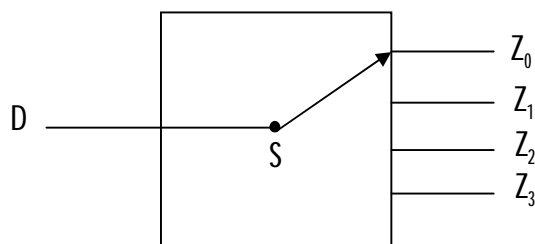
กรณีที่ 2: เลือก D_1 , กำหนด $S_0 = 1$ และ $S_1 = 0$

กรณีที่ 3: เลือก D_2 , กำหนด $S_0 = 0$ และ $S_1 = 1$

กรณีที่ 4: เลือก D_3 , กำหนด $S_0 = 1$ และ $S_1 = 1$

11.2 วงจรดีมัลติเพล็กซ์เซอร์

วงจรดีมัลติเพล็กซ์เซอร์ หรือเรียกสั้นๆว่า **DEMUX** เป็นวงจรที่มีการทำงานตรงข้ามกับวงจรมัลติเพล็กซ์เซอร์ คือ วงจรดีมัลติเพล็กซ์เซอร์จะมีข้อมูลอินพุตเพียง 1 ค่า แต่มีช่องทางของเอาต์พุตมากกว่า 1 ช่องทาง โดย ณ เวลาใดเวลาหนึ่งอินพุตจะถูกส่งไปยังช่องทางใดช่องทางหนึ่งเท่านั้น

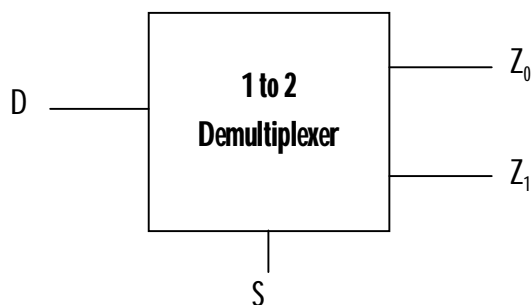


รูปที่ 11.2 โครงสร้างวงจรดีมัลติเพล็กซ์เซอร์แบบ 4 เอาต์พุต

รูปที่ 11.2 แสดงโครงสร้างของวงจรดีมัลติเพล็กซ์เซอร์แบบ 4 เอาต์พุตประกอบด้วย Z_0 , Z_1 , Z_2 และ Z_3 มีอินพุตคือ D และมี S เป็นตัวที่ใช้สำหรับการเลือกช่องทางเอาต์พุตเพื่อที่จะใช้ส่งข้อมูลอินพุต โดยที่จำนวนบิตของ S สามารถคำนวณหาได้จาก $2^n =$ จำนวนเอาต์พุต เมื่อ n คือจำนวนบิตของ S ที่ต้องใช้งานในวงจรดังนั้นจากรูปที่ 11.2 จำนวนบิตของ S ที่จำเป็นต้องใช้งานทั้งหมดคือ 2 บิต ($2^2 = 4$)

ตัวอย่างที่ 11.4: จงออกแบบวงจรดีมัลติเพล็กซ์เซอร์แบบ 2 เอาต์พุต

วิธีทำ ก่อนออกแบบจำเป็นต้องเลือกจำนวนบิตของ S เพื่อใช้สำหรับเลือกเอาต์พุต จากโจทย์เอาต์พุตมีทั้งหมด 2 ตัว ดังนั้น ต้องใช้ S ที่มีขนาด 1 บิต ($2^1 = 2$) ได้โครงสร้างเป็นดังนี้

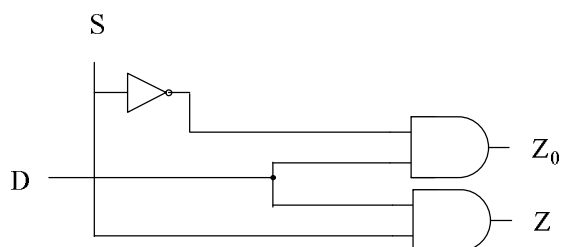


เมื่อได้โครงสร้างของวงจรแล้ว ขั้นตอนต่อไปกำหนดสถานะของ S และสมการพีชคณิตบูลีนของ Z_0 และ Z_1 ดังนี้

กรณี $S = 0$: เลือก D ออกทางช่อง $Z_0 \rightarrow Z_0 = SD$

กรณี $S = 1$: เลือก D ออกทางช่อง $Z_1 \rightarrow Z_1 = SD$

ได้วงจรดีมัลติเพล็กซ์แบบ 2 เอาต์พุต เป็นดังนี้



11.3 บทสรุป

วงจรดีมัลติเพล็กซ์ และวงจรดีมัลติเพล็กซ์เป็นวงจรที่ใช้สำหรับการรับส่งข้อมูลข่าวสารในกรณีที่มีหลายช่องทางผ่านสายส่งเพียงเส้นเดียว โดยวงจรดีมัลติเพล็กซ์ เป็นวงจรที่คัดเลือกข้อมูลอินพุตที่มีหลายค่าโดยแต่ละค่าจะอยู่ในช่องทางที่แตกต่างกันหลายช่องทางมาเพียงค่าเดียวก่อนที่จะส่งข้อมูลดังกล่าวผ่านสายส่ง และวงจรดีมัลติเพล็กซ์เป็นวงจรที่รับข้อมูลจากสายส่งเพื่อส่งออกทางช่องสัญญาณเอาต์พุตที่มีหลายช่องทางเพียงช่องทางเดียว

อย่างไรก็ตาม วงจรดีมัลติเพล็กซ์ และวงจรดีมัลติเพล็กซ์จำเป็นต้องมีสัญญาณอีกชนิดหนึ่งเรียกว่า ตัวเลือกข้อมูล (Data selection) เพื่อจะทำให้สามารถเลือกข้อมูลอินพุตจากช่องสัญญาณที่ต้องการ และส่งข้อมูลดังกล่าวไปยังช่องสัญญาณเอาต์พุตที่ต้องการได้อย่างถูกต้อง

คำถามท้ายบท

1. วงจรมัลติเพล็กซ์เซอร์ และวงจรมัลติเพล็กซ์เซอร์คืออะไร มีประโยชน์อย่างไร
2. วงจรมัลติเพล็กซ์เซอร์ และวงจรมัลติเพล็กซ์เซอร์มีความแตกต่างกันอย่างไร
3. หากต้องการสร้างวงจรมัลติเพล็กซ์เซอร์ที่มีช่องสัญญาณอินพุตทั้งหมด 16 ช่องทางจะต้องใช้สาย Data section ทั้งหมดกี่เส้น
4. จงออกแบบวงจรมัลติเพล็กซ์เซอร์แบบ 8 และ 16 อินพุต
5. จงออกแบบวงจรมัลติเพล็กซ์เซอร์แบบ 8 อินพุตโดยใช้มัลติเพล็กซ์เซอร์แบบ 2 และ 4 อินพุต
6. จงออกแบบวงจรมัลติเพล็กซ์เซอร์แบบ 8 อินพุตโดยใช้มัลติเพล็กซ์เซอร์แบบ 2 อินพุตเท่านั้น
7. จงออกแบบวงจรมัลติเพล็กซ์เซอร์แบบ 8 และ 16 เอาต์พุต

เอกสารอ้างอิง

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.

ธวัชชัย เลื่อนฉวี. (2532). **ดิจิทัลเทคนิคเล่ม 2**. กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิทัล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุดรธานี: มหาวิทยาลัยราชภัฏอุดรธานี.**

แผนบริหารการสอนประจำบทที่ 12

วงจรแปลงสัญญาณ 3 ชั่วโมง

หัวข้อเนื้อหา

- 12.1 ออปแอมป์
 - 12.1.1 วงจรเปรียบเทียบ
 - 12.1.2 อินเวอร์ตติ้งแอมพลิไฟเออร์
- 12.2 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก
 - 12.2.1 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก
- 12.3 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล
 - 12.3.1 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล
- 12.4 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับออปแอมป์เบื้องต้น และการนำออปแอมป์มาประยุกต์ใช้งานเป็นวงจรแปลงสัญญาณ
2. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับหลักการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก และสามารถออกแบบวงจรได้
3. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับหลักการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล และสามารถออกแบบวงจรได้

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน
3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเล็อน

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

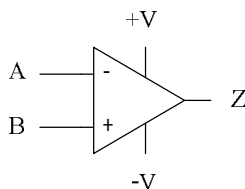
บทที่ 12

วงจรแปลงสัญญาณ

เนื่องจากมีความเป็นไปได้ที่สัญญาณอินพุตของวงจรดิจิทัลอาจจะเป็นสัญญาณแบบอนาลอก เช่นค่าอุณหภูมิ หรือ ค่าความเร็ว เป็นต้น ซึ่งวงจรดิจิทัลไม่สามารถประมวลผลสัญญาณอินพุตที่เป็นแบบอนาลอกได้ ดังนั้นจึงต้องมีวงจรที่ใช้สำหรับแปลงสัญญาณอินพุตจากสัญญาณอนาลอกเป็นสัญญาณดิจิทัลก่อนจึงจะสามารถนำสัญญาณดิจิทัลนี้ส่งเป็นอินพุตให้วงจรเพื่อนำไปประมวลต่อไป โดยหลังจากที่วงจรดิจิทัลประมวลผลเสร็จเรียบร้อยแล้วค่าสัญญาณเอาต์พุตที่ได้ออกมาจะเป็นสัญญาณดิจิทัล อย่างไรก็ตามในการใช้งานจริงก็มีความเป็นไปได้เช่นกันที่จำเป็นต้องใช้ค่าเอาต์พุตที่เป็นสัญญาณอนาลอก และในทางกลับกันวงจรอนาลอกบางชนิดอาจจะรับสัญญาณอินพุตแบบดิจิทัล ดังนั้นจึงต้องมีวงจรที่ใช้สำหรับการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอกด้วยเช่นกัน โดยอุปกรณ์ที่สามารถนำมาประยุกต์ใช้สำหรับการแปลงสัญญาณคือ ออปแอมป์ ในบทนี้จะกล่าวถึง ออปแอมป์เบื้องต้น หลักการแปลงสัญญาณ และวงจรแปลงสัญญาณของทั้งสองโดยใช้ออปแอมป์

12.1 ออปแอมป์

ออปแอมป์ คืออุปกรณ์ทางอิเล็กทรอนิกส์ชนิดหนึ่งที่ได้รับสัญญาณอินพุตเป็นแบบอนาลอก ซึ่งสามารถนำมาประยุกต์ใช้สำหรับสร้างวงจรแปลงสัญญาณได้ โดยมีคุณสมบัติที่สำคัญ 3 ประการ ดังนี้ มีอิมพีแดนซ์ด้านอินพุตสูง มีอัตราขยายแรงดันสูง และ มีอิมพีแดนซ์ด้านเอาต์พุตต่ำ



รูปที่ 12.1 สัญลักษณ์ของออปแอมป์

จากรูปที่ 12.1 แสดงให้เห็นว่าสัญญาณของออปแอมป์มีทั้งหมด 5 สัญญาณประกอบไปด้วยสัญญาณอินพุต 2 สัญญาณ คือ A ซึ่งต่อเข้ากับขาลบเรียกว่า อินเวอร์ตติ้ง และ B ซึ่งต่อเข้ากับขาบวกเรียกว่า นอนอินเวอร์ตติ้ง แหล่งจ่ายไฟ 2 สัญญาณซึ่งเป็นสัญญาณบวก และสัญญาณลบ และสัญญาณเอาต์พุต 1 สัญญาณ

12.1.1 วงจรเปรียบเทียบ

จากรูปที่ 12.1 สามารถนำออปแอมป์มาใช้เป็นตัวเปรียบเทียบได้โดยที่พิจารณาที่ขาของอินพุตทั้ง 2 ค่า โดยแบ่งออกเป็น 3 กรณี ดังนี้

กรณีที่ 1: $A > B$, Z จะมีค่าเป็น $-5V$

กรณีที่ 2: $A < B$, Z จะมีค่าเป็น +5V

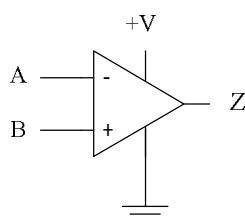
กรณีที่ 3: $A = B$, Z จะมีค่าเป็น 0V

ตัวอย่างที่ 12-1 จงหาค่าแรงดันของ Z เมื่อกำหนดให้ $A = 3V$ และ $B = 1V$

วิธีทำ เนื่องจาก แรงดันของ A มีค่ามากกว่าแรงดันของ B

ดังนั้น $Z = -5V$

อย่างไรก็ตามหากนำออปแอมป์มาใช้เป็นวงจรเปรียบเทียบ อาจกำหนดให้แรงดันที่เอาต์พุตมีสถานะเป็นบวก หรือลบเพียงอย่างเดียวเท่านั้น เช่นหากต้องการให้สถานะเอาต์พุตมีค่าเป็นบวกเพียงอย่างเดียวสามารถทำได้โดยต่อขา -V ลงสายดินแทน ดังรูปที่ 12.2



รูปที่ 12.2 วงจรเปรียบเทียบที่สถานะเอาต์พุตมีค่าเป็นบวกเท่านั้น

จากรูปที่ 12.2 ค่า Z จะเหลือเพียง 2 สถานะเท่านั้น ดังนี้

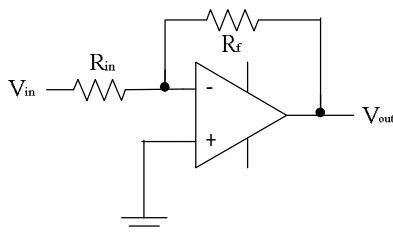
กรณีที่ 1: $A \geq B$, Z จะมีค่าเป็น 0V

กรณีที่ 2: $A < B$, Z จะมีค่าเป็น +5V

ในทางกลับกันหากต้องการให้สถานะเอาต์พุตมีค่าเป็นลบเพียงอย่างเดียวสามารถทำได้โดยต่อขา +V ลงสายดินแทน ซึ่งสถานะเอาต์พุตจะเหลือเพียงสถานะที่เป็นลบเพียงอย่างเดียว

12.1.2 อินเวอร์ตติ้งออปแอมป์ไฟเออร์

อัตราการขยายแรงดันของออปแอมป์มีค่าสูงถึง 200,000 เท่า แต่ไม่ควรทำให้แรงดันที่เอาต์พุตสูงเกิน 80% ดังนั้นการใช้งานจริงจึงควรทำให้อัตราการขยายแรงดันของออปแอมป์มีค่าน้อยกว่า 200,000 เท่า โดยการควบคุมอัตราการขยายด้วยวิธีการป้อนกลับผ่านตัวต้านทานที่ตกคร่อมออปแอมป์ระหว่างอินพุตและเอาต์พุต



รูปที่ 12.3 อินเวอร์ตติ้งออปแอมป์

จากรูปที่ 12.3 อัตราการขยายแรงดันของออปแอมป์หาได้จาก

$$V_{\text{gain}} = \frac{V_{\text{out}}}{V_{\text{in}}}$$

เมื่อ V_{gain} = อัตราการขยายแรงดัน

V_{out} = แรงดันที่เอาต์พุต

V_{in} = แรงดันที่อินพุต

โดยที่ค่า V_{out} สามารถหาได้จากสูตร

$$V_{\text{out}} = \frac{R_f \times V_{\text{in}}}{R_{\text{in}}}$$

เมื่อ R_f = ตัวต้านทานที่ตกคร่อมออปแอมป์ระหว่างอินพุตและเอาต์พุต

R_{in} = ตัวต้านทานที่ต่อที่ขาอินพุต

12.2 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก หรือเรียกสั้นๆว่า D/A คือการแปลงสัญญาณอินพุตที่เป็นแบบดิจิทัลให้เป็นแรงดันแบบอนาลอก ซึ่งมีลักษณะคล้ายกับการแปลงจากเลขฐานสองเป็นเลขฐานสิบนั่นเอง ยกตัวอย่างเช่น สัญญาณ 0010_2 เมื่อแปลงเป็นสัญญาณอนาลอกจะมีค่าเป็น $2V$ หรือ สัญญาณ 0101_2 เมื่อแปลงเป็นสัญญาณอนาลอกจะมีค่าเป็น $5V$ เป็นต้น

การคำนวณหาแรงดันที่เป็นสัญญาณอนาลอกจากสัญญาณดิจิทัลสามารถทำได้ดังสมการต่อไปนี้

$$V_{\text{out}} = K.D$$

เมื่อ V_{out} คือ แรงดันอนาลอก

D คือ รหัสสัญญาณดิจิทัลที่เกิดจากการแปลงจากรหัสฐานสองเป็นเลขฐานสิบ

K คือ ค่าคงที่

ตัวอย่างที่ 12-2 จงคำนวณแรงดันอนาลอก จากวงจร D/A ขนาด 8 บิตโดยอินพุตมีค่าเป็น 01010011 และค่าคงที่แรงดันอนาลอกมีค่าเป็น $10mV$

วิธีทำ เนื่องจากอินพุตคือ 01010011 แปลงเป็นเลขฐานสิบได้เป็น 83

ดังนั้น จาก $V_{\text{out}} = KD$

ได้ $V_{\text{out}} = 10mV \times 83$

$$= 0.83V$$

ตัวอย่างที่ 12-3 จากวงจร D/A ขนาด 8 บิตโดยอินพุตมีค่าเป็น 01100100 จะได้แรงดันออกมาเป็น 4V จงคำนวณหาค่าคงที่แรงดัน และค่าแรงดันสูงสุดของวงจรนี้

วิธีทำ เนื่องจากอินพุตคือ 01100100 แปลงเป็นเลขฐานสิบได้เป็น 100

$$\text{ดังนั้น จาก } V_{\text{out}} = KD$$

$$\text{ได้ } 4 = K \times 100$$

$$K = 0.04V$$

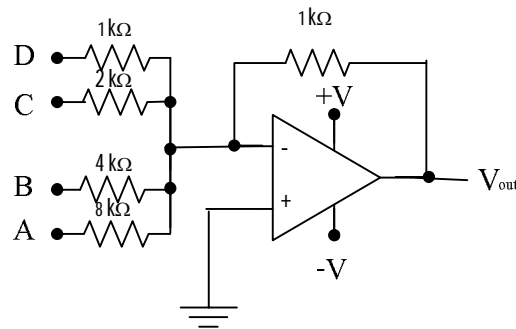
เนื่องจากอินพุตมีขนาด 8 บิต ดังนั้นค่าสูงสุดคือ $11111111_2 = 255_{10}$

$$\text{จาก } V_{\text{max}} = KD_{\text{max}}$$

$$\text{ได้ } = 0.04 \times 255$$

$$= 10.2V$$

12.2.1 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก



รูปที่ 12.4 ตัวอย่างวงจร D/A ขนาด 4 บิต

จากรูปที่ 12.4 แสดงตัวอย่างวงจร D/A ขนาด 4 บิต ซึ่งมีอินพุตประกอบด้วย A, B, C และ D เนื่องจากค่าน้ำหนักจากบิตที่มีลำดับความสำคัญต่ำที่สุดเรียงไปถึงบิตที่มีลำดับความสำคัญสูงที่สุดมีค่าเป็น 1, 2, 4 และ 8 ดังนั้นค่า R ที่นำมาต่อที่ตำแหน่งบิตที่มีลำดับความสำคัญต่ำที่สุดจึงต้องมีค่ามากที่สุด และค่า R ที่นำมาต่อที่ตำแหน่งบิตที่มีลำดับความสำคัญสูงที่สุดจึงต้องมีค่าน้อยที่สุด จึงทำให้ทราบว่าอินพุต A แทนบิตที่มีลำดับความสำคัญต่ำที่สุด และอินพุต D คือตำแหน่งที่มีลำดับความสำคัญสูงที่สุดโดยกำหนดให้เมื่ออินพุตแต่ละตัวได้รับค่าเป็น 1 คือรับแรงดันขนาด 5V แต่หากได้รับค่าเป็น 0 คือยังไม่มีแรงดัน

ตัวอย่างที่ 12-4 จากรูป 12.4 จงหาค่า V_{out} หากอินพุตมีค่าเป็น 1010

วิธีทำ เนื่องจากอินพุตมีค่าเป็น 1010 ความหมายคือ

$$D = 1, C = 0, B = 1 \text{ และ } A = 0$$

จึงคำนวณหาค่ากระแสจากแต่ละอินพุตได้ดังนี้

$$I_D = V_D/R_D = 5/1\text{k}\Omega = 5\text{ mA}$$

$$I_C = V_C/R_C = 0/2 \text{ k}\Omega = 0 \text{ mA}$$

$$I_B = V_B/R_B = 5/4 \text{ k}\Omega = 1.25 \text{ mA}$$

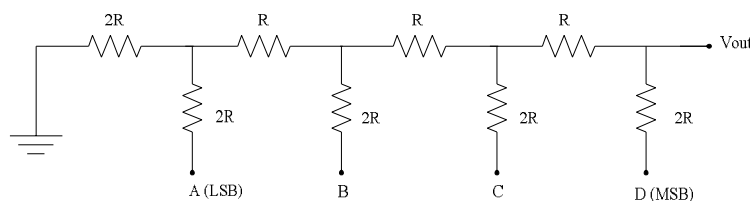
$$I_A = V_A/R_A = 0/8 \text{ k}\Omega = 0 \text{ mA}$$

$$\text{จาก } I_{Rf} = I_D + I_C + I_B + I_A = 6.25 \text{ mA}$$

จึงคำนวณหา V_{out} ได้ดังนี้

$$\begin{aligned} \text{จาก } V_{out} &= I_{Rf} \cdot R_f \\ &= 6.25 \text{ mA} \times 1 \text{ k}\Omega = 6.25\text{V} \end{aligned}$$

วงจร D/A ดังรูปที่ 12.4 เรียกว่า วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอกแบบตัวต้านทานเลขฐานสอง (Binary Weighted Resistor D/A) อย่างไรก็ตามยังมีวงจร D/A อีกประเภทหนึ่งที่นิยมถูกนำมาใช้งานเช่นกัน ซึ่งจะนำตัวต้านทาน 2 ค่ามาต่อสลับกันในรูปแบบซ้ำๆ เดิม ตามจำนวนบิต เรียกววงจรลักษณะนี้ว่า วงจรชุดความต้านทานขั้นบันได (R-2R Ladder Resistor D/A) โดยการเรียงจะอยู่ในรูปแบบของขั้นบันไดที่มีการแบ่งสัดส่วนแรงดันขาออกอย่างเป็นลำดับชั้น คุณสมบัติของวงจร R-2R Ladder คือระดับแรงดันเอาต์พุตจะมีค่าเท่ากับผลรวมแรงดันอินพุตที่เป็นรหัสเลขฐานสอง โดยค่าแรงดันอินพุตแต่ละตัวจะมีค่าเป็น $2^{-1}, 2^{-2}, \dots, 2^{-(n-1)}$ ซึ่งค่าแรงดันนี้เรียงจากค่าแรงดันของอินพุตที่มีลำดับความสำคัญสูงที่สุดไปจนถึงอินพุตที่มีลำดับความสำคัญต่ำที่สุด



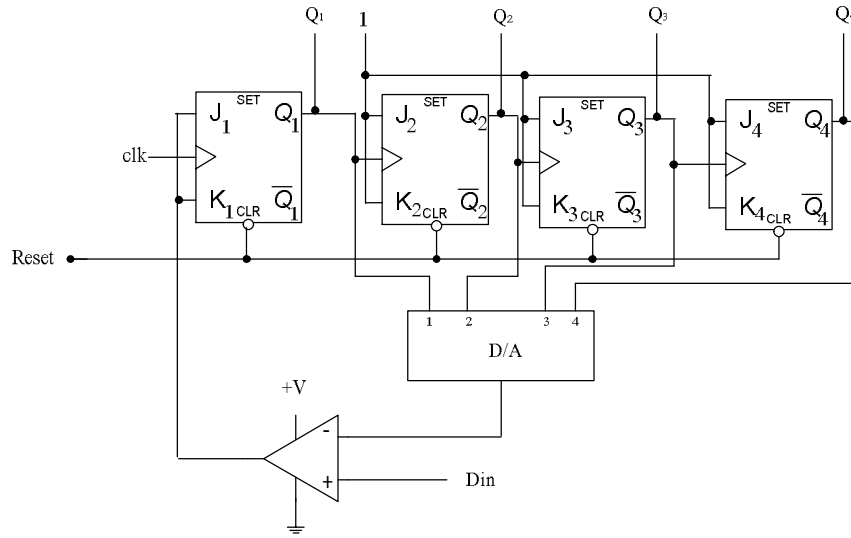
รูปที่ 12.5 ตัวอย่างวงจร D/A แบบ R-2R Ladder ขนาด 4 บิต

จากรูปที่ 12.5 แสดงตัวอย่างวงจร D/A แบบ R-2R Ladder ขนาด 4 บิต โดยอินพุต D เป็นอินพุตที่มีลำดับความสำคัญสูงที่สุดเรียงไปจนถึงอินพุต A ซึ่งเป็นอินพุตที่มีลำดับความสำคัญต่ำที่สุด ดังนั้นค่าแรงดันของอินพุต D อินพุต C อินพุต B และอินพุต A จึงมีค่าเป็น $2^{-1}, 2^{-2}, 2^{-3}$ และ 2^{-4} ตามลำดับ

12.3 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล หรือเรียกสั้นๆว่า A/D คือการแปลงสัญญาณอินพุตที่เป็นแบบอนาลอกให้เป็นสัญญาณแบบดิจิทัล ซึ่งมีลักษณะคล้ายกับการแปลงจากเลขฐานสิบเป็นรหัสเลขฐานสองนั่นเอง

12.3.1 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล



รูปที่ 12.6 ตัวอย่างวงจร A/D ขนาด 4 บิต

จากรูปที่ 12.6 แสดงตัวอย่างวงจร A/D เพื่อแปลงเป็นสัญญาณดิจิทัลขนาด 4 บิต โดยมี D_{in} เป็นสัญญาณอินพุตแบบอนาลอก และ $Q_1 - Q_4$ เป็นสัญญาณเอาต์พุตแบบดิจิทัล โดยแบ่งการอธิบายการทำงานออกเป็น 3 ส่วนได้ดังนี้

ส่วนที่ 1 ออปแอมป์: ทำหน้าที่เป็นวงจรเปรียบเทียบสัญญาณอนาลอกระหว่าง เอาต์พุตของวงจร D/A และ D_{in} โดยที่หากสัญญาณเอาต์พุตจาก D/A มีค่ามากกว่าหรือเท่ากับ D_{in} เอาต์พุตของออปแอมป์จะมีค่าเป็น 0 แต่หากสัญญาณเอาต์พุตจาก D/A มีค่าน้อยกว่า D_{in} เอาต์พุตของออปแอมป์จะมีค่าเป็น 1

ส่วนที่ 2 ฟลิปฟลอปแบบ JK ทั้ง 4 ตัว: จากรูปสังเกตได้ว่าค่าเอาต์พุตของออปแอมป์ถูกส่งไปเป็นอินพุตให้ J_1 และ K_1 ค่า Q_1 จึงถูกแบ่งออกเป็นเพียง 2 กรณีคือ หาก $J_1 = k_1 = 0$ ค่า Q_1 จะคงค่าเดิม แต่หาก $J_1 = k_1 = 1$ ค่า Q_1 จะเกิดการกลับบิตจากค่าเดิม ส่วนฟลิปฟลอปตัวอื่นๆทั้งหมดถูกกำหนดให้อินพุต J และ K มีค่าเป็น 1 ดังนั้นเอาต์พุตของฟลิปฟลอปตัวที่ 2 - 4 จะเกิดการกลับบิตเสมอหากได้รับสัญญาณนาฬิกา ซึ่งสังเกตว่าฟลิปฟลอปตัวที่ n ได้รับสัญญาณนาฬิกาจากเอาต์พุตของฟลิปฟลอปตัวที่ $n-1$ เมื่อ $n = 2, 3, 4$ ดังนั้นการต่อใช้งานฟลิปฟลอปลักษณะนี้เปรียบเสมือนวงจรนับนั่นเอง โดยจะทำการนับไปเรื่อยๆ หากเอาต์พุตของออปแอมป์มีค่าเป็น 1 และหยุดนับเมื่อเอาต์พุตของออปแอมป์มีค่าเป็น 0 เนื่องจาก Q_1 คงค่าเดิม จึงไม่เกิดสัญญาณนาฬิกาที่ Q_2, Q_3 และ Q_4

ส่วนที่ 3 วงจร D/A: รับสัญญาณอินพุตจาก Q_1, Q_2, Q_3 และ Q_4 เพื่อที่จะแปลงเป็นสัญญาณอนาลอกเพื่อนำไปเปรียบเทียบกับ D_{in}

เมื่อรวมทั้ง 3 ส่วนสรุปการทำงานได้ดังนี้ เมื่อมีเกิดสัญญาณ Reset จะทำให้เอาต์พุตของฟลิปฟลอปทั้ง 4 ตัวมีค่าเป็น 0 ทั้งหมด และวงจรจะเริ่มนับขึ้นทีละ 1 จนกระทั่งเอาต์พุตของวงจร D/A มีค่าเท่ากับ D_{in} วงจรจึงจะหยุดนับ และได้ผลลัพธ์ซึ่งเป็นสัญญาณดิจิทัลคือ Q_1, Q_2, Q_3 และ Q_4

12.4 บทสรุป

วงจรแปลงสัญญาณ คือ วงจรที่ทำให้ระบบดิจิทัลสามารถสื่อสารกับอุปกรณ์อิเล็กทรอนิกส์ที่รับสัญญาณเป็นแบบสัญญาณอนาลอกได้ โดยวงจรแปลงสัญญาณแบ่งออกเป็นสองประเภทคือวงจรการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก หรือเรียกสั้นๆว่า **D/A** คือการแปลงสัญญาณอินพุตที่เป็นแบบดิจิทัลให้เป็นแรงดันแบบอนาลอก (มีลักษณะคล้ายกับการแปลงจากรหัสเลขฐานสองเป็นสัญญาณที่อยู่ในรูปแบบของเลขฐานสิบ) และวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล หรือเรียกสั้นๆว่า **A/D** คือการแปลงสัญญาณอินพุตที่เป็นแบบอนาลอกให้เป็นสัญญาณแบบดิจิทัล (มีลักษณะคล้ายกับการแปลงสัญญาณที่เป็นเลขฐานสิบให้อยู่ในรูปแบบของรหัสเลขฐานสอง)

คำถามท้ายบท

1. วงจรแปลงสัญญาณมีประโยชน์อย่างไร
2. วงจร D/A และ A/D คืออะไร มีความแตกต่างกันอย่างไร
3. จากรูปที่ 12.1 จงหาค่าแรงดันของ Z เมื่อ
 - 3.1 กำหนดให้ $A = 1V$ และ $B = 3V$
 - 3.2 กำหนดให้ $A = 4V$ และ $B = 2V$
4. จงคำนวณแรงดันอนาลอก จากวงจร D/A ขนาด 8 บิตโดยอินพุตมีค่าเป็น 01001101 และค่าคงที่แรงดันอนาลอกมีค่าเป็น 10mV
5. จากวงจร D/A ขนาด 8 บิตโดยอินพุตมีค่าเป็น 01101110 จะได้แรงดันออกมาเป็น 5V จงคำนวณหาค่าคงที่แรงดัน และค่าแรงดันสูงสุดของวงจรนี้

เอกสารอ้างอิง

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เลื่อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**. กรุงเทพฯ: มิตรนราการพิมพ์.

ธวัชชัย เลื่อนฉวี. (2532). **ดิจิทัลเทคนิคเล่ม 2**. กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิทัล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**. กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลอิเล็กทรอนิกส์.จ. อุตรธานี**: มหาวิทยาลัยราชภัฏอุตรธานี.

แผนบริหารการสอนประจำบทที่ 13

ภาษา VHDL เบื้องต้น 6 ชั่วโมง

หัวข้อเนื้อหา

13.1 บทนำ

- 13.1.1 ไลบราลี้
- 13.1.2 เอ็นตีตี้
- 13.1.3 อาซิเทคเซอร์
- 13.1.4 แพ็กเกจ
- 13.1.5 คอมพิกกูเรชั่น

13.2 การตั้งชื่อตัวแปร

13.3 ชนิดข้อมูล

- 13.3.1 ชนิดข้อมูล BIT
- 13.3.2 ชนิดข้อมูล BIT_VECTOR
- 13.3.3 ชนิดข้อมูล STD_LOGIC
- 13.3.4 ชนิดข้อมูล STD_LOGIC_VECTOR
- 13.3.5 ชนิดข้อมูล INTEGER
- 13.3.6 ชนิดข้อมูลอื่น

13.4 ตัวดำเนินการทางตรรกะ

13.5 สัญญาณ

13.6 ค่าคงที่

13.7 คำสั่งเชิงลำดับ

13.8 คำสั่งเงื่อนไข

- 13.8.1 คำสั่ง When – Else
- 13.8.2 คำสั่ง With - Select
- 13.8.3 คำสั่ง If - Else
- 13.8.4 คำสั่ง Case - When

13.9 การสร้างสัญญาณนาฬิกา

13.10 บทสรุป

วัตถุประสงค์เชิงพฤติกรรม

1. เพื่อให้ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับการเขียนโปรแกรมภาษาบรรยายฮาร์ดแวร์ด้วยภาษา VHDL เบื้องต้น

วิธีการสอนและกิจกรรมการเรียนการสอนประจำบท

1. บรรยายเนื้อหาในแต่ละหัวข้อ พร้อมยกตัวอย่างประกอบ
2. ศึกษาจากเอกสารประกอบการสอน

3. ผู้สอนสรุปเนื้อหา
4. ทำแบบฝึกหัดเพื่อทบทวนบทเรียน
5. เปิดโอกาสให้ผู้เรียนถามข้อสงสัย
6. ผู้สอนทำการซักถาม

สื่อการเรียนการสอน

1. เอกสารประกอบการสอนวิชาวงจรดิจิทัลและลอจิก
2. ภาพเคลื่อนไหว

การวัดผลและการประเมิน

1. ประเมินจากการซักถามในชั้นเรียน
2. ประเมินจากความร่วมมือและความรับผิดชอบต่อการเรียน
3. ประเมินจากการทำแบบฝึกหัดทบทวนบทเรียน

บทที่ 13

ภาษา VHDL เบื้องต้น

การออกแบบวงจรถิทัศน์ในบทที่ผ่านมา ๆ มาเป็นการออกแบบวงจรถิทัศน์โดยใช้วิธีการวาดผังวงจรถิทัศน์ขึ้นมาเอง ซึ่งเป็นวิธีที่ไม่เหมาะสมในกรณีที่ต้องออกแบบวงจรถิทัศน์ที่มีขนาดใหญ่ การออกแบบวงจรถิทัศน์โดยใช้ภาษาระดับสูง เป็นวิธีการออกแบบวงจรถิทัศน์โดยใช้วิธีการเขียนโปรแกรมเพื่อจำลองการทำงาน ซึ่งสามารถพัฒนาได้ง่าย เนื่องจากรูปแบบของคำสั่งต่างๆ เป็นภาษาที่มนุษย์อ่านได้เข้าใจได้ง่าย ดังนั้นการใช้ภาษาระดับสูงในการออกแบบวงจรถิทัศน์ที่มีขนาดใหญ่จึงเป็นวิธีที่เหมาะสมมากกว่าหากเปรียบเทียบกับวิธีการวาดผังวงจรถิทัศน์ใช้งานเอง สำหรับบทนี้จะกล่าวถึงการเขียนโปรแกรมระดับสูงเพื่อใช้สำหรับการออกแบบวงจรถิทัศน์ด้วยภาษา VHDL เบื้องต้น

13.1 บทนำ

ภาษา VHDL ย่อมาจาก VHSIC Hardware Description Language เป็นภาษาที่ใช้สำหรับการออกแบบทางด้านฮาร์ดแวร์ ประเภทหนึ่งซึ่งถูกพัฒนาโดยกระทรวงกลาโหมของประเทศสหรัฐอเมริกา และได้รับการปรับปรุงเพื่อให้เป็นมาตรฐานโดย IEEE โดยมี IEEE Std 1076 (VHDL87) เป็นมาตรฐานตัวแรกที่มีการใช้งาน และได้มีการปรับปรุงเรื่อยมา โดยภาษานี้จะแตกต่างกับโปรแกรมภาษาอื่นๆ เช่น ภาษา C, C++ หรือ Java เป็นต้น ตรงที่ภาษา VHDL จะเป็นภาษาที่ประมวลผลคำสั่งทุกคำสั่งไปพร้อมๆ กัน (Concurrency Statement) ในขณะที่โปรแกรมภาษาอื่น ๆ จะประมวลผลคำสั่งแต่ละคำสั่งเป็นลำดับ (Sequential Statement) โดยภาษา VHDL จะมองตัวอักษรภาษาอังกฤษตัวพิมพ์เล็ก และตัวอักษรพิมพ์ใหญ่เป็นตัวเดียวกัน (Case Insensitive) โครงสร้างของ VHDL เป็นดังนี้

13.1.1 ไลบรารี (Library)

เป็นส่วนที่ใช้สำหรับเก็บคำสั่ง หรือตัวดำเนินการต่าง ๆ ที่สามารถนำมาใช้งานได้ตามมาตรฐาน IEEE โดยจะอยู่ส่วนบนสุดของโปรแกรม มีโครงสร้างเป็นดังนี้

```
LIBRARY library_name;
USE library_name.package_name.item
```

รูปที่ 13.1 โครงสร้างไลบรารี

จากรูปที่ 13.1 `Library_name` คือ ชื่อของไลบรารีที่เก็บแพ็คเกจ โดยจะขึ้นต้นด้วย "IEEE"

`package_name` คือ ชื่อแพ็คเกจ

`item` คือ จำนวนของข้อมูลที่อยู่ภายในแพ็คเกจที่ต้องการนำมาใช้งานภายในโปรแกรมหลักซึ่งหากต้องการนำมาใช้งานทั้งหมดให้ประกาศในส่วนของ `item` เป็นคำว่า "all"

13.1.2 เอ็นติตี้ (Entity)

เป็นส่วนที่ใช้สำหรับการบอกจำนวนอินพุต และจำนวนของเอาต์พุตทั้งหมดที่จำเป็นต้องใช้ภายในวงจรโดยยังไม่ได้มีการออกแบบวงจรแต่อย่างไร มีโครงสร้างเป็นดังนี้

```
Entity entity_name IS
PORT(input_name1: IN Datatype;
      input_name2: IN Datatype;
      input_namen: IN Datatype;
      output_name1: IN Datatype;
      output_namem: OUT Datatype);
End entity_name;
```

รูปที่ 13.2 โครงสร้างเอ็นติตี้

จากรูปที่ 13.2 entity_name คือ ชื่อของเอ็นติตี้ โดยจะต้องมีชื่อเดียวกับ entity_name ที่อยู่หลังคำว่า End

input_name₁, input_name₂ และ input_name_n คือ ข้อมูลอินพุตของวงจรซึ่งจะมีกี่ตัวก็ได้

IN คือ ส่วนที่อยู่ต่อจากชื่อของข้อมูลอินพุตและเครื่องหมาย ":" เพื่อให้รู้ว่าข้อมูลนี้เป็นข้อมูลอินพุต

OUT คือส่วนที่อยู่ต่อจากชื่อของข้อมูลเอาต์พุตและเครื่องหมาย ":" เพื่อให้รู้ว่าข้อมูลนี้เป็นข้อมูลเอาต์พุต

Datatype คือ ชนิดข้อมูลของสัญญาณ โดยหากอินพุต หรือ เอาต์พุต ที่มีมากกว่า 1 ตัว มี Datatype ชนิดเดียวกันสามารถประกาศสัญญาณเหล่านั้นในบรรทัดเดียวกันได้โดยใช้ "," คั่นสัญญาณแต่ละตัว

ตัวอย่างที่ 13-1 จงเขียนคำสั่งในส่วนเอ็นติตี้เพื่อออกแบบวงจร $Z = A.B$

วิธีทำ จากโจทย์มีอินพุต 2 ตัว และเอาต์พุต 1 ตัว จึงมีเอ็นติตี้เป็นดังนี้

```
Entity EX_01 IS
PORT(A: IN BIT;
      B: IN BIT;
      Z: OUT BIT);
End EX_01;
```

อินพุต 2 ตัวมีชื่อ คือ A และ B และเอาต์พุต 1 ตัวมีชื่อคือ Z ทั้งหมดมีชนิดข้อมูลเป็นแบบ BIT (ชนิดข้อมูลแบบ BIT คือ ชนิดข้อมูลที่มีสถานะสำหรับใช้งานคือ 0 หรือ 1) และมีเอ็นติตี้ชื่อ EX_01

13.1.3 อาซิเทคเซอร์ (Architecture)

เป็นส่วนที่ใช้สำหรับการออกแบบส่วนที่ใช้เขียนบรรยายพฤติกรรมการทำงานของวงจรถูกจากข้อมูลอินพุต และเอาต์พุตที่มาจากส่วนของเอนิตตี้ ซึ่งมีโครงสร้างเป็นดังนี้

```

ARCHITECTURE architecture_name of entity_name IS
Signal, Constant, type, subprogram and component
declarations
Begin
...

End architecture_name;

```

รูปที่ 13.3 โครงสร้างอาซิเทคเซอร์

จากรูปที่ 13.3 architecture_name คือ ชื่อของอาซิเทคเซอร์ ซึ่งจะต้องมีชื่อเดียวกันกับ architecture_name ที่อยู่หลัง End ด้วย

entity_name คือ ชื่อของ entity ซึ่งจะต้องมีชื่อเดียวกันกับชื่อของ เอนิตตี้ที่อยู่ในส่วนของเอนิตตี้

ส่วนที่อยู่ใต้ ARCHITECTURE และ อยู่เหนือ Begin คือการประกาศสัญญาณ ค่าคงที่ หรือ โปรแกรมย่อยไว้ใช้งานซึ่งจะมีหรือไม่ได้ขึ้นอยู่กับการใช้งาน หากมีการใช้งานแต่ละตัวแปรต้องปิดท้ายด้วยเครื่องหมาย ";" เสมอ

การออกแบบส่วนที่ใช้เขียนบรรยายพฤติกรรมการทำงานของวงจรถูกจะเขียนอยู่ระหว่าง Begin และ End เสมอ โดยรูปแบบของการกำหนดสถานะให้เอาต์พุตเป็นดังนี้

กรณีที่ 1: กำหนดค่าคงที่ให้เอาต์พุต

รูปแบบ: ข้อมูลเอาต์พุต <= 'ค่าคงที่';

เช่น: $Z <= '1';$

ความหมายคือ กำหนดให้สัญญาณ Z มีสถานะเป็น 1

กรณีที่ 2: กำหนดสถานะอินพุตให้เอาต์พุต

รูปแบบ: ข้อมูลเอาต์พุต <= ข้อมูลอินพุต;

เช่น: $Z <= A;$

ความหมายคือ กำหนดให้สัญญาณ Z มีสถานะเหมือนสัญญาณ A

กรณีที่ 3: กำหนดสถานะนิเสธของอินพุตให้เอาต์พุต

รูปแบบ: ข้อมูลเอาต์พุต <=not ข้อมูลอินพุต;

เช่น: $Z <= \text{not } A;$

ความหมายคือ กำหนดให้สัญญาณ Z มีสถานะตรงข้ามกับสัญญาณ A

กรณีที่ 4: กำหนดสถานะของผลการดำเนินการระหว่างอินพุตให้เอาต์พุต

รูปแบบ: ข้อมูลเอาต์พุต <= ข้อมูลอินพุต1 ตัวดำเนินการ ข้อมูลอินพุตที่2;

เช่น: $Z <= A \text{ and } B;$

ความหมายคือ กำหนดให้สัญญาณ $Z = AB$

ตัวอย่างที่ **13-2** จงเขียนคำสั่งในส่วนอาชิตะเซอร์เพื่อออกแบบวงจรจากตัวอย่างที่ **13-1**

วิธีทำ เนื่องจากตัวอย่างที่ **13-1** ได้ออกแบบโครงสร้างของเอ็นติตี้ไว้แล้ว จึงออกแบบวงจรในส่วนของอาชิตะเซอร์ได้ดังนี้

```
ARCHITECTURE architecture_EX01 of EX_01 IS
Begin
    Z <= A and B;
End architecture_EX01;
```

อาชิตะเซอร์มีชื่อ `architecture_EX01` ส่วนเอ็นติตี้จะต้องมีชื่อ `EX_01` เสมอซึ่งเป็นชื่อเดียวกันกับชื่อของเอ็นติตี้ที่อยู่ในตัวอย่างที่ **13-1** โดยการเขียนคำสั่ง $Z = AB$ ด้วยภาษา VHDL เป็นดังนี้ $Z <= A \text{ and } B;$

13.1.4 แพ็กเกจ (Package)

แพ็กเกจ คือส่วนที่ใช้สำหรับเก็บข้อมูลต่าง ๆ หรือโปรแกรมย่อยซึ่งส่วนใหญ่แล้วจะเป็นข้อมูลที่มีการเรียกใช้งานบ่อยครั้ง หรือมีการเรียกใช้งานจากหลายส่วนของโปรแกรม โดยในโปรแกรมหลักจะมีการเรียกใช้งานแพ็กเกจหรือไม่ก็ได้ โครงสร้างของแพ็กเกจเป็นดังนี้

```
PACKAGE package_name IS
Signal, Constant, type and component declarations
End package_name;
PACKAGE BODY package_name IS
Signal, Constant, type, component and Subprogram declarations
End package_name;
```

รูปที่ 13.4 โครงสร้างแพ็กเกจ

จากโครงสร้างของแพ็กเกจแบ่งออกเป็น 2 ส่วนคือส่วนของการประกาศแพ็กเกจซึ่งจะใช้สำหรับประกาศสัญญาณ หรือค่าคงที่ต่าง ๆ โดยที่หากมีการสร้างแพ็กเกจขึ้นมาใช้งานจะต้องมีส่วนนี้เสมอ และส่วนที่ 2 คือ แพ็กเกจบอดี้ คือส่วนของคำสั่งต่างๆ ที่ใช้บรรยายการทำงานของโปรแกรมย่อยซึ่งจะมีการทำงานเป็นแบบเชิงลำดับ โดยสัญญาณ หรือค่าคงที่ต่าง ๆ ที่ประกาศภายในแพ็กเกจบอดี้ จะต้องประกาศไว้ในส่วนของการประกาศแพ็กเกจด้วยเสมอ โดยหากมีการประกาศแพ็กเกจขึ้นมาใช้งานจะมีแพ็กเกจบอดี้หรือไม่ก็ได้

หากมีการเรียกใช้งานแพ็กเกจจากโปรแกรมหลักจะต้องมีการใช้คำสั่งดังนี้

USE library_name.package_name.item;

Library_name คือ ชื่อของไลบรารีที่เก็บแพ็คเกจ โดยส่วนใหญ่แล้วแพ็คเกจที่ผู้เขียนสร้างขึ้นเองจะถูกเก็บไว้ในไลบรารีชื่อ "work"

package_name คือ ชื่อแพ็คเกจ

item คือ จำนวนของข้อมูลที่อยู่ภายในแพ็คเกจที่ต้องการนำมาใช้งานภายในโปรแกรมหลัก ซึ่งหากต้องการนำมาใช้งานทั้งหมดใช้ประกาศในส่วนของ **item** เป็นคำว่า "all"

13.1.5 คอนฟิกูเรชัน (Configuration)

ในกรณีที่ต้องการสร้างอาชิตะเซอร์ไว้มากกว่า 1 อาชิตะเซอร์ไว์เพื่อไว้ใช้สำหรับการจำลองการทำงานของเอ็นดีที จำเป็นต้องมีการเขียนคอนฟิกูเรชันไว้เพื่อเลือกใช้งานอาชิตะเซอร์ไว์ที่ต้องการ

13.2 การตั้งชื่อแปร

การตั้งชื่อตัวแปรสัญญาณ หรือค่าคงที่ต่าง ๆ เพื่อไว้ใช้งานจะต้องตั้งตามข้อกำหนดต่อไปนี้เท่านั้น ซึ่งหากไม่ตั้งตามที่กำหนดไว้จะเกิดข้อผิดพลาดของโปรแกรมขึ้น

1. ชื่อตัวแปรจะต้องประกอบด้วย ตัวอักษร ตัวเลข หรือ เส้นใต้ เท่านั้น
2. กรณีที่ชื่อตัวแปร 1 ตัวแปรมีมากกว่า 1 ตัวอักษรจะต้องเขียนติดกัน ห้ามมีช่องว่าง
3. ไม่สามารถใช้เครื่องหมายขีดเส้นใต้ติดกันหลายตัว และไม่สามารถจบด้วยเครื่องหมายขีดเส้นใต้
4. ห้ามใช้คำสงวนที่มีอยู่ในภาษา VHDL ดังนี้

| | | | | |
|---------------|------------|---------|-----------|------------|
| abs | disconnect | label | package | sia |
| access | downto | library | port | sll |
| after | else | linkage | postponed | sra |
| alias | elsif | literal | procedure | srl |
| all | end | loop | process | subtype |
| and | entity | map | protected | then |
| architecture | exit | mod | pure | to |
| array | file | nand | range | transport |
| assert | for | new | record | type |
| attribute | function | next | register | unaffected |
| begin | generate | nor | reject | units |
| block | generic | not | rem | until |
| body | group | null | report | use |
| buffer | guarded | of | return | variable |
| bus | if | on | rol | wait |
| case | impure | open | ror | when |
| component | in | or | select | while |
| configuration | inertial | others | severity | with |
| constant | inout | out | shared | with |
| | is | | signal | xnor |
| | | | | xor |

รูปที่ 13.5 คำสงวนในภาษา VHDL

ที่มาของภาพ: http://hep.uchicago.edu/~tangjian/SVT_sub/FTK_ATLAS/AUX/vhdl-tutorial.pdf

13.3 ชนิดข้อมูล

ข้อมูลที่ต้องใช้งานในภาษา VHDL มีอยู่หลายชนิด ซึ่งแต่ละชนิดจะมีความแตกต่างกันออกไป ดังนี้

13.3.1 ชนิดข้อมูล BIT: เป็นชนิดข้อมูลพื้นฐานที่ใช้สำหรับกำหนดเป็นสถานะของข้อมูลอินพุตหรือข้อมูลเอาต์พุตซึ่งประกอบด้วย 2 สถานะคือ 0 หรือ 1

13.3.2 ชนิดข้อมูลแบบ BIT_VECTOR: เป็นชนิดข้อมูลแบบ BIT ที่ใช้เก็บข้อมูลให้กับตัวแปร 1 ตัวแปรที่มีขนาดมากกว่า 1 บิต ซึ่งสามารถกำหนดได้ 2 แบบคือแบบที่ 1 เรียงจากบิตที่มีลำดับความสำคัญสูงที่สุดไปยังบิตที่มีลำดับความสำคัญต่ำที่สุดโดยใช้ "downto" และแบบที่ 2 เรียงจากบิตที่มีลำดับความสำคัญต่ำที่สุดไปยังบิตที่มีลำดับความสำคัญสูงที่สุดโดยใช้ "to" ตัวอย่างการใช้งานเป็นดังนี้

แบบที่ 1 (ใช้ downto): signal a: BIT_VECTOR (3 downto 0) := "1101";

ความหมายคือ กำหนดให้สัญญาณ a มีขนาด 4 บิตและมีค่าเริ่มต้นเป็น "1101" ($a_3 = 1, a_2 = 1, a_1 = 0$ และ $a_0 = 1$)

แบบที่ 2 (ใช้ to): signal a: BIT_VECTOR (0 to 3) := "1101";

ความหมายคือ กำหนดให้สัญญาณ a มีขนาด 4 บิตและมีค่าเริ่มต้นเป็น "1101" ($a_0 = 1, a_1 = 1, a_2 = 0$ และ $a_3 = 1$)

13.3.3 ชนิดข้อมูล STD_LOGIC: คือ ชนิดข้อมูลที่ใช้สำหรับกำหนดเป็นสถานะของข้อมูลอินพุตหรือข้อมูลเอาต์พุตเช่นเดียวกับชนิดข้อมูลแบบ BIT แต่สามารถเก็บสถานะได้มากกว่า โดยสถานะที่ชนิดข้อมูลแบบ STD_LOGIC สามารถเก็บได้เป็นดังนี้

'U' คือ Uninitialized

'X' คือ Unknown

'0' คือ สถานะ 0

'1' คือ สถานะ 1

'Z' คือ High impedance

'W' คือ Weak unknown

'L' คือ Weak low

'H' คือ Weak high

'-' คือ Don't care

13.3.4 ชนิดข้อมูล STD_LOGIC_VECTOR: คือ ชนิดข้อมูลที่ใช้สำหรับกำหนดเป็นสถานะของข้อมูลอินพุตหรือข้อมูลเอาต์พุตที่มีขนาดมากกว่า 1 บิตเช่นเดียวกับชนิดข้อมูลแบบ BIT_VECTOR แต่สามารถเก็บสถานะได้มากกว่า โดยสถานะที่ชนิดข้อมูลแบบ STD_LOGIC_VECTOR จะเหมือนกับ STD_LOGIC

13.3.5 ชนิดข้อมูล INTEGER: คือ จำนวนเต็มที่มีขนาดระหว่าง -2,147,483,647 - +2,147,483,647 ซึ่งเป็นชนิดข้อมูลที่สามารถใช้ในการคำนวณได้ เช่น การบวก ลบ คูณ หรือหาร เป็นต้น

13.3.6 ชนิดข้อมูลอื่น นอกจากชนิดข้อมูล 5 ชนิดที่กล่าวข้างต้นแล้ว ภาษา VHDL ยังมีชนิดข้อมูลอื่นๆ ให้ใช้งานได้อีก เช่น REAL (จำนวนจริง) BOOLEAN (ตรรกศาสตร์) TIME (เวลา) STRING (ตัวอักษร) เป็นต้น

13.4 ตัวดำเนินการทางตรรกะ

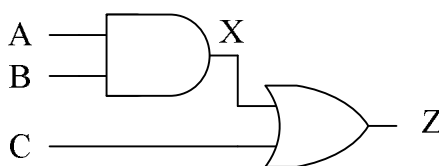
ตัวดำเนินการทางตรรกะคือตัวดำเนินการที่ใช้ในการคำนวณหาสถานะของเอาต์พุตของวงจรดิจิทัลจากความสัมพันธ์ของข้อมูลอินพุตมีทั้งหมด 7 ตัวดังนี้

ตารางที่ 13.1 ตัวดำเนินการทางตรรกะ

| ประเภทตัวดำเนินการ | รูปแบบคำสั่ง | ตัวอย่างการใช้งาน |
|--------------------|--------------|-----------------------------|
| แอนด์ | and | $z \leq a \text{ and } b;$ |
| ออร์ | or | $z \leq a \text{ or } b;$ |
| น็อต | not | $z \leq \text{not } a;$ |
| แนนด์ | nand | $z \leq a \text{ nand } b;$ |
| นอร์ | nor | $z \leq a \text{ nor } b;$ |
| เอ็กออร์ | xor | $z \leq a \text{ xor } b;$ |
| เอ็กนอร์ | xnor | $z \leq a \text{ xnor } b;$ |

13.5 สัญญาณ (Signal)

สัญญาณ คือ ตัวแปรที่ทำหน้าที่ในการส่งผ่านข้อมูล ตัวอย่างเช่นสัญญาณจะนำเอาต์พุตที่ได้จากอุปกรณ์อิเล็กทรอนิกส์ตัวหนึ่งไปเป็นอินพุตให้กับอุปกรณ์อิเล็กทรอนิกส์อีกตัวหนึ่ง ดังตัวอย่างรูปที่ 13.4



รูปที่ 13.6 ตัวอย่างสัญญาณ

จากรูปที่ 13.6 เอาต์พุตที่ได้จากแอนด์เกต (X) ถูกส่งไปเป็นอินพุตให้กับ ออร์เกต แสดงว่า X เป็นตัวแปรสัญญาณ

รูปแบบการเขียนสัญญาณเป็นดังนี้

รูปแบบ: `signal signal_name: Datatype := initial_value;`

โดยที่ `signal_name` คือ ชื่อสัญญาณ

`initial_value` คือค่าเริ่มต้นของสัญญาณ โดยในส่วนี้จะมีการกำหนดหรือไม่ก็ได้

ตัวอย่างที่ 13-3 การออกแบบวงจรรูป 13-4 ด้วยภาษา VHDL โดยใช้ตัวแปรสัญญาณ

```

Entity EX_03 is
  Port ( a, b, c : in BIT;
        Z : out BIT);
end EX_03;
architecture Behavioral of EX_03 is
  signal x: BIT;
begin
  x <= a and b;
  Z <= x or c;
end Behavioral;

```

13.6 ค่าคงที่ (Constant)

ค่าคงที่ คือตัวแปรที่เมื่อถูกสร้างขึ้นมาแล้วต้องมีการกำหนดค่าให้กับตัวแปรนั้นทันทีและไม่สามารถเปลี่ยนแปลงค่าให้กับตัวแปรนี้ได้ภายหลัง มีรูปแบบเป็นดังนี้

รูปแบบ: `constant constant_name: Datatype := value;`

โดยที่ `signal_name` คือ ชื่อสัญญาณ

`value` คือ ค่าคงที่ที่ต้องกำหนดให้ตัวแปร

13.7 คำสั่งเชิงลำดับ (Sequential statement)

การเขียนโปรแกรมด้วยภาษา VHDL มีลักษณะการทำงานเป็นแบบขนานคือแต่ละคำสั่งจะทำงานไปพร้อมกัน อย่างไรก็ตามสามารถกำหนดให้ภาษา VHDL มีการทำงานเป็นลำดับได้โดยการเขียนคำสั่งที่ต้องการให้มีการทำงานเป็นลำดับไว้ภายในโปรเซส (Process) ซึ่งเป็นคำสั่งที่มีการทำงานเป็นแบบขนาน แต่คำสั่งที่อยู่ภายในโปรเซสจะมีการทำงานแบบลำดับโดยโปรเซสจะถูกเขียนไว้ในอาชีเทคเซอร์ มีโครงสร้างเป็นดังนี้

```

Process_name: Process(input1,
input2,...,inputn)
variablevariable_name: Datatype :=
initial_value;
Begin
  Sequential_statement1;
  Sequential_statement2;
End Process;

```

รูปที่ 13.7 โครงสร้างโปรเซส

จากรูปที่ 13.7 **Process_name**: คือชื่อของ **Process** ซึ่งจะมีหรือไม่ก็ได้

Input1, input2,...,inputn คือ จำนวนอินพุตทั้งหมดที่ต้องใช้งานภายในโปรเซสและส่งผลโดยตรงต่อผลลัพธ์ภายในโปรเซส โดยอินพุตแต่ละตัวจะถูกคั่นด้วย “,”

Variable_name คือ ตัวแปรวาริเอเบิล ซึ่งเป็นตัวแปรแบบโลคอล (**Local Variable**) ซึ่งจะถูกใช้งานอยู่ภายในโปรเซสเท่านั้น การใช้งานจะคล้ายกับตัวแปรสัญญาณ แตกต่างกันตรงที่ตัวแปรสัญญาณจะมีการปรับปรุงข้อมูลอยู่ภายนอกโปรเซส แต่ตัวแปรวาริเอเบิลจะมีการปรับปรุงข้อมูลภายในโปรเซส

การกำหนดสถานะของผลการดำเนินการระหว่างอินพุตให้อาต์พุตที่เป็นตัวแปรวาริเอเบิลจะใช้ “:=” ซึ่งแตกต่างกับการกำหนดค่าให้ตัวแปรสัญญาณที่ใช้ “<=”

รูปแบบ: เอาต์พุต := สัญญาณอินพุต

คำสั่งเชิงลำดับจะถูกเขียนไว้ภายในที่อยู่ระหว่าง **Begin** และ **End** โดยที่คำสั่งที่ถูกเขียนก่อนจะถูกประมวลผลก่อน ซึ่งจากรูปที่ 13.5 **Sequential_statement1** ถูกเขียนก่อน **Sequential_statement2** ดังนั้น **Sequential_statement1** จะถูกนำไปประมวลผลก่อน

ตัวอย่างที่ 13-4 การออกแบบวงจรรูป 13.4 ด้วยภาษา VHDL โดยใช้ตัวแปรวาริเอเบิล

```
Entity EX_04 is
  Port ( a, b, c : in BIT;
        Z : out BIT);
end EX_04;
architecture Behavioral of EX_04 is
begin
  Process(a, b)
    variable x: BIT;
  begin
    x := a and b;
    Z <= x or c;
  end Process;
end Behavioral;
```

13.8 คำสั่งเงื่อนไข (Conditional statement)

คำสั่งเงื่อนไขเป็นคำสั่งที่ใช้สำหรับคุมทิศทางการทำงานของโปรแกรม โดยคำสั่งเงื่อนไขถูกแบ่งออกเป็น 2 วิธีคือคำสั่งที่มีการทำงานเป็นแบบขนาน และคำสั่งที่มีการทำงานแบบลำดับ โดยมีทั้งหมด 4 คำสั่งดังนี้

13.8.1 คำสั่ง When - Else

คำสั่งที่มีการทำงานแบบขนาน โดยสถานะของสัญญาณเอาต์พุตจะมีค่าตรงกับสถานะของสัญญาณหรือค่าคงที่ที่อยู่หน้า **When** ในกรณีที่เงื่อนไขที่อยู่หลัง **When** เป็นจริง โครงสร้างคำสั่งเป็นดังนี้

```
signal_name <= expression_1 when condition1 else
    expression_2 when condition2 else
    ...
    Expression_n;
```

รูปที่ 13.8 โครงสร้างคำสั่ง When - Else

จากรูปที่ 13.8 **signal_name** คือ ชื่อสัญญาณเอาต์พุต

expression_1, expression_2, ..., expression_n คือสถานะของค่าคงที่ หรืออินพุตที่จะกำหนดให้สัญญาณเอาต์พุตโดยขึ้นอยู่กับเงื่อนไขโดยจะตรวจสอบเงื่อนไขแรกก่อน (**condition1**) ซึ่งหากเป็นจริงจะกำหนดสถานะของ **expression_1** ให้สัญญาณเอาต์พุตแต่หากเงื่อนไขเป็นเท็จจะไปตรวจสอบเงื่อนไขต่อไป กรณีที่ไม่มีเงื่อนไขใดเลยที่เป็นจริงจะกำหนดสถานะของ **expression** ตัวสุดท้าย (**expression_n**) ให้แก่สัญญาณเอาต์พุตโดย **expression** ตัวสุดท้ายต้องไม่มี คำสั่ง **when - Else** ตามหลัง

ตัวอย่างที่ 13-5 จงออกแบบวงจรมัลติเพล็กซ์แบบ 2 อินพุตโดยใช้คำสั่ง **When - Else** โดยมี **C** เป็นสัญญาณที่ใช้เลือก พฤติกรรมของวงจร ดังนี้

เมื่อ **C = 0** แสดงผลลัพธ์ **Z = A**

C = 1 แสดงผลลัพธ์ **Z = B**

```
entity EX_05 is
    Port ( A, B, C : in BIT;
          Z : out BIT);
end EX_05;
architecture Behavioral of EX_05 is
begin
    Z <= A when C = '0' else
        B;
end Behavioral;
```

จากโปรแกรมมีหลักการทำงานเป็นดังนี้

เมื่อ **C = 0** (เงื่อนไขเป็นจริง) จะได้ **Z = A** แต่หากเงื่อนไขเป็นเท็จจะได้ **Z = B**

เนื่องจาก C เป็นตัวแปรแบบ BIT ซึ่งมีเพียง 2 สถานะคือ 0 หรือ 1 ดังนั้นกรณีที่เงื่อนไขเป็นเท็จจะเกิดขึ้นเมื่อ C มีค่าเป็น 1

13.8.2 คำสั่ง With - Select

คำสั่งที่มีการทำงานแบบขนาน โดยสถานะของสัญญาณเอาต์พุตจะมีค่าตรงกับสถานะของสัญญาณหรือค่าคงที่ที่อยู่หน้า **When** ในกรณีที่เงื่อนไขที่อยู่หลัง **When** มีค่าตรงกับสัญญาณที่อยู่หลัง **With** โครงสร้างของคำสั่งเป็นดังนี้

```
With select_signal select
    signal_name <= expression_1 when condition1;
    expression_2 when condition2;
    ...
    expression_n when condition_n;
```

รูปที่ 13.9 โครงสร้างคำสั่ง With - Select

จากรูปที่ 13.9 **signal_name** คือ สัญญาณเอาต์พุต

expression_1, expression_2,...,expression_n คือสถานะของค่าคงที่ หรืออินพุตที่จะกำหนดให้สัญญาณเอาต์พุตโดยขึ้นอยู่กับเงื่อนไข โดยเงื่อนไขใดที่ตรงกับสัญญาณ **"select_signal"** จะกำหนดให้สถานะของสัญญาณเอาต์พุตมีค่าตรงกับสถานะของสัญญาณที่อยู่ในบรรทัดเดียวกันกับเงื่อนไข เช่น กรณีที่ **condition2** ตรงกับ **select_signal** ดังนั้น **signal_name** จะมีสถานะตรงกับ **expression_2** เป็นต้น

ตัวอย่างที่ 13-6 จงออกแบบวงจรที่มีการทำงานเช่นเดียวกับวงจรในตัวอย่างที่ 13-5 แต่ให้ใช้คำสั่ง **With - Select**

```
entity EX_06 is
    Port ( A,B,C : in BIT;
          Z : out BIT);
end EX_06;
architecture Behavioral of EX_06 is
begin
    with C select
        Z <= A when '0',
            B when '1';
end Behavioral;
```

13.8.3 คำสั่ง If - Else

เป็นคำสั่งที่นิยมใช้มากที่สุด โดยคำสั่ง If - Else จะเป็นคำสั่งที่มีการทำงานเป็นแบบลำดับ ดังนั้นการใช้งานจะต้องอยู่ภายในคำสั่งโปรเซส โดยชุดคำสั่งที่จะถูกนำไปใช้งานอยู่ในบล็อกของเงื่อนไขที่เป็นจริง โครงสร้างของคำสั่งเป็นดังนี้

```

if condition1 then
    sequential_statement1;
elsif condition2 then
    sequential_statement2;
...
else
    sequential_statementn;
endif

```

รูปที่ 13.10 โครงสร้างคำสั่ง If - Else

จากรูปที่ 13.10 การทำงานจะเริ่มตรวจสอบเงื่อนไข condition1 ก่อน ซึ่งหากเงื่อนไขเป็นจริง โปรแกรมจะประมวลผลการทำงานที่ sequential_statement1 แต่หากเงื่อนไขเป็นเท็จ โปรแกรมจะตรวจสอบเงื่อนไขถัดไป อย่างไรก็ตามกรณีที่ไม่มีเงื่อนไขที่เป็นจริง โปรแกรมจะประมวลผล sequential_statementn ที่อยู่ในบล็อกของ else ซึ่งเป็นคำสั่งที่อยู่ส่วนสุดท้ายของคำสั่ง If - Else โดยที่คำสั่ง If จะต้องมีส่วนในโครงสร้างของคำสั่ง If - Else แต่คำสั่ง Else และคำสั่ง Eelsif จะมีหรือไม่ก็ได้

ตัวอย่างที่ 13-7 จงออกแบบวงจรที่มีการทำงานเช่นเดียวกับวงจรในตัวอย่างที่ 13-5 แต่ให้ใช้คำสั่ง If - Else

```

entity EX_07 is
    Port ( A,B,C : in BIT;
          Z : out BIT);
end EX_07;
architecture Behavioral of EX_07 is
begin
    Process(A,B)
    begin
        if C = '0' then
            Z <= A;
        else
            Z <= 'B';
        end if;
    end process;
end Behavioral;

```

13.8.4 คำสั่ง Case - When

คำสั่ง **Case - When** จะเป็นคำสั่งที่มีการทำงานเป็นแบบลำดับเช่นเดียวกับคำสั่ง **If - Else** ดังนั้นการใช้งานจะต้องอยู่ภายในคำสั่ง **Process** โดยชุดคำสั่งที่จะถูกนำไปใช้งานจะอยู่ในบล็อก **when** ที่มีสถานะสัญญาณที่ตรงกับสถานะของสัญญาณที่อยู่หลัง **case** กรณีที่ไม่มีสัญญาณที่อยู่หลัง **when** ที่มีสถานะตรงกับสถานะที่อยู่หลัง **case** โปรแกรมจะเลือกใช้งานชุดคำสั่งที่อยู่หลัง **when others** ซึ่งจะเป็นชุดคำสั่งที่อยู่ตำแหน่งสุดท้ายของคำสั่ง **Case - When** โครงสร้างของคำสั่งเป็นดังนี้

```
case expression is
    when choice_1 => sequential_statement1;
    when choice_2 => sequential_statement1;
    ...
    when others sequential_statementn;
end case
```

รูปที่ 13.11 โครงสร้างคำสั่ง Case - When

จากรูปที่ 13.11 ชุดคำสั่งที่จะถูกนำไปประมวลผลจะอยู่ในบรรทัดเดียวกับ **choice** ที่มีสถานะตรงกับ **expression** เช่น **choice_1** มีสถานะตรงกับ **expression** คำสั่ง **sequential_statement1** จะเป็นคำสั่งที่ถูกนำไปประมวลผล เป็นต้น

ตัวอย่างที่ 13-8 จงออกแบบวงจรมีการทำงานเช่นเดียวกับวงจรมีในตัวอย่างที่ 13-5 แต่ให้ใช้คำสั่ง **Case - When**

```
entity EX_08 is
    Port ( A, B, C : in BIT;
          Z : out BIT);
end EX_08;

architecture Behavioral of EX_08 is
begin
    Process(A,B)
    begin
        case C is
            when '0' => Z <= A;
            when others => Z <= B;
        end case;
    end process;
end Behavioral;
```

13.9 การสร้างสัญญาณนาฬิกา

การสร้างสัญญาณนาฬิกาโดยใช้ภาษา VHDL มีรูปแบบคำสั่งเป็นดังนี้

รูปแบบ: `signal_name'event and signal_name = '0 (or 1)';`

โดยที่ `signal_name` คือ สัญญาณที่จะนำมาใช้เป็นสัญญาณนาฬิกา ซึ่งมี 2 สถานะคือหาก `signal_name` มีค่าเป็น 0 เป็นการเกิดสัญญาณนาฬิกาที่ขอบขาลง แต่หาก `signal_name` มีค่าเป็น 1 เป็นการเกิดสัญญาณนาฬิกาที่ขอบขาขึ้น

ตัวอย่างที่ 13-9 สร้างฟลิปฟล็อปแบบ D ที่ใช้สัญญาณนาฬิกาขอบขาลงโดยใช้ภาษา VHDL

```
entity EX_09 is
  Port ( D,CLK : in BIT;
        Q,notQ : out BIT);
endEX_09;
architecture Behavioral of EX_09 is
begin
  Process(CLK)
  begin
    if CLK'event and CLK = '0' then
      Q <= D;
      notQ <= not D;
    end if;
  end process;
end Behavioral;
```

จากโปรแกรมกำหนดให้ `D` คือ อินพุตของวงจร, `CLK` คือ อินพุตที่ใช้สร้างเป็นสัญญาณนาฬิกา, `Q` และ `notQ` คือเอาต์พุตของวงจรโดยที่ `notQ` คือ นิเสธของ `Q` การทำงานคือเมื่อเกิดสัญญาณนาฬิกา `Q` จะมีค่าเท่ากับสถานะของ `D` และ `notQ` จะมีสถานะตรงข้ามกับ `Q`

ข้อสังเกต สัญญาณ `CLK` เป็นสัญญาณอินพุตเพียงตัวเดียวที่ถูกประกาศเป็นอินพุตของคำสั่งโปรเซส เนื่องจาก `CLK` จะมีผลต่อสัญญาณ `Q` และ `notQ` โดยตรง ในขณะที่ `D` จะขึ้นอยู่กับสัญญาณ `CLK` หากไม่เกิดสัญญาณ `CLK` สถานะ `Q` และ `notQ` จะไม่มีการเปลี่ยนแปลง

13.10 บทสรุป

ภาษา VHDL (ย่อมาจาก VHSIC Hardware Description Language) เป็นภาษาระดับสูงที่ใช้สำหรับการออกแบบทางด้านฮาร์ดแวร์ประเภทหนึ่ง ซึ่งเป็นวิธีการออกแบบวงจรดิจิทัลโดยใช้วิธีการเขียนโปรแกรมเพื่อจำลองการทำงาน ซึ่งสามารถพัฒนาได้ง่าย เนื่องจากรูปแบบของคำสั่งต่างๆ เป็นภาษาที่มนุษย์อ่านได้เข้าใจ ดังนั้นการใช้ภาษาระดับสูงในการออกแบบวงจรดิจิทัลที่มีขนาดใหญ่ จึงเป็นวิธีที่เหมาะสมมากกว่าหากเปรียบเทียบกับวิธีของกาวบัดผังวงจรขึ้นใช้งานเอง

คำถามท้ายการทดลอง

1. ภาษา VHDL คืออะไร และมีประโยชน์อย่างไร
2. Entity คืออะไร และมีไว้เพื่ออะไร
3. จงอธิบายโครงสร้างทั้งหมดของ Entity มาโดยละเอียด
4. จงเขียนคำสั่งในส่วนเอนตีตี้เพื่อออกแบบวงจร $Y = A(B + C)$
5. Architecture คืออะไร และมีไว้เพื่ออะไร
6. จงอธิบายโครงสร้างทั้งหมดของ Architecture มาโดยละเอียด
7. จาก Entity ที่ได้ในข้อ 4 จงนำมาออกแบบวงจรในส่วนของ Architecture
8. Package คืออะไร และมีประโยชน์อย่างไร
9. จงอธิบายความแตกต่างระหว่างชนิดข้อมูลแบบ BIT และ BIT_VECTOR
10. จงอธิบายความแตกต่างระหว่างการเรียงบิตข้อมูลแบบ TO และ DOWNTO
11. จงอธิบายความแตกต่างระหว่าง Concurrency statement และ Sequential statement
12. จงอธิบายความแตกต่างระหว่าง Signal และ Variable
13. จงออกแบบวงจรถอดรหัสโดยใช้ภาษา VHDL ที่แปลงจากอินพุตที่เป็นเลขฐานสองขนาด 10 บิต เป็นรหัสเลขฐานสิบโดยใช้
 - 13.1 คำสั่ง When - Else
 - 13.2 คำสั่ง With - Select
 - 13.3 คำสั่ง If - Else
 - 13.4 คำสั่ง Case - When
14. จงออกแบบฟลิปฟล็อปแบบ JK ที่ใช้สัญญาณนาฬิกาที่ขอบขาลงโดยใช้ภาษา VHDL

เอกสารอ้างอิง

Mark, Z. (2004). **Digital System Design with VHDL**. United Kingdom: Pearson Education.

ชำนาญ ปัญญาใส และ วัชรกร หนูทอง. (2547). ภาษา **VHDL** สำหรับการออกแบบวงจรถิทัศน์. กรุงเทพฯ: ห้างหุ้นส่วนจำกัดเอช-เอน การพิมพ์.

ณรงค์ ทองฉิม และ เจริญ วงษ์ชุ่มเย็น. (2552). ออกแบบไอซีดิจิทัลด้วย **FPGA** และ **CPLD** ภาคปฏิบัติโดยใช้ภาษา **VHDL** ซอฟต์แวร์ชุด **ISE WebPack**. กรุงเทพฯ: บริษัท วี.พี.พี. (1991) จำกัด.

http://hep.uchicago.edu/~tangjian/SVT_sub/FTK_ATLAS/AUX/vhdl-tutorial.pdf

บรรณานุกรม

Morris, M, Charles, R.K. (2007). **Logic and Computer Design Fundamentals**. New Jersey: Prentice-Hall International Inc.

Marcovitz, A. B. (2009). **Introduction to Logic Design**. New York: McGraw-Hill.

Mark, B. (2003). **Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture**. New York: McGraw-Hill.

David, M. H. (2012). **Digital Design and Computer Architecture**. USA: Morgan Kaufmann.

Ramaswamy, P. (2011). **Digital Systems Design**. United Kingdom: London Business School.

Morris, M, Michael, D. C. (2006). **Digital Design**. New Jersey: Prentice-Hall International Inc.

Mark, Z. (2004). **Digital System Design with VHDL**. United Kingdom: Pearson Education.

สมชาย ชื่นวัฒนาประณีต. (2535). **ดิจิทัลลอจิกทรอนิกส์**.จ. อุตรธานี: มหาวิทยาลัยราชภัฏอุตรธานี.

ธวัชชัย เลื่อนฉวี และ อนุรักษ์ เกื้อนศิริ. (2527). **ดิจิทัลเทคนิคเล่ม 1**.กรุงเทพฯ: มิตรนราการพิมพ์.

ธวัชชัย เลื่อนฉวี. (2532). **ดิจิทัลเทคนิคเล่ม 2**.กรุงเทพฯ: มิตรนราการพิมพ์.

มงคล ทองสงคราม. (2544). **ทฤษฎีดิจิตอล**.กรุงเทพฯ: ห้างหุ้นส่วนจำกัด วี.เจ. พรินติ้ง.

ทีมงานสมาร์ทเลิร์นนิ่ง. (2543). **ออกแบบวงจร Digital และประยุกต์ใช้งาน**.กรุงเทพฯ: ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง.

ชำนาญ ปัญญาใส และ วัชรกร หนูทอง. (2547). **ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล**. กรุงเทพฯ: ห้างหุ้นส่วนจำกัดเอช-เอน การพิมพ์.