

บทที่ 6
วงจรรรกกประสม
COMBINATIONAL LOGIC CIRCUIT

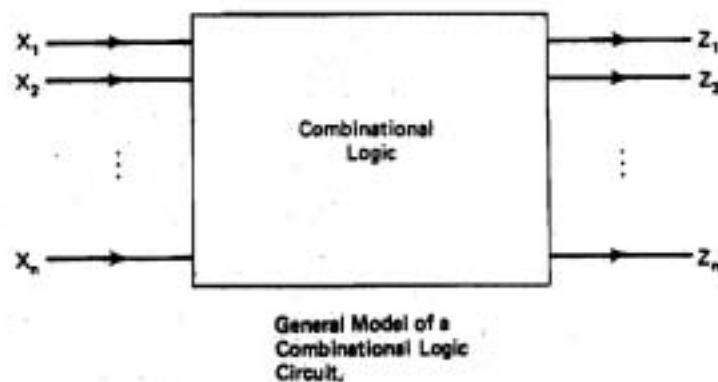
วัตถุประสงค์

เมื่อศึกษาจบบทนี้แล้ว นักศึกษาสามารถ

1. อธิบายความหมายของวงจรรรกกประสมได้
2. อธิบายความแตกต่างของวงจรรรกกประสม และวงจรรรกกเชิงควนเชื่อม
3. ออกแบบวงจรรรกกประสมได้

6.1 ความนำ

วงจรตรรกสำหรับระบบดิจิทัลอาจเป็นแบบประสม (combinational) หรือแบบซีควนเชียล (sequential) วงจรตรรกประสมประกอบด้วยเกทตรรกซึ่งเอาต์พุต ณ ขณะเวลาใด ๆ ถูกกำหนดโดยตรงจากสภาวะประสมปัจจุบันของอินพุตโดยไม่คำนึงถึงอินพุตเมื่อก่อนหน้านั้น และดำเนินการในกระบวนการข้อมูลตามบูลีนฟังก์ชัน วงจรซีควนเชียลมีชิ้นส่วนความจำ (ซึ่งเป็นเซลล์ฐานสอง (binary cell)) ทำงานร่วมกับเกทตรรก เอาต์พุตของมันเป็นฟังก์ชันของอินพุตและสถานะของชิ้นส่วนความจำ โดยที่สถานะของชิ้นส่วนความจำเป็นฟังก์ชันของอินพุตก่อนหน้านั้น กล่าวโดยสรุปก็คือเอาต์พุตของวงจรซีควนเชียล อินพุตปัจจุบัน อินพุตในอดีต และพฤติกรรมของวงจรต้องถูกป้องกันโดยลำดับเวลาของอินพุตและสถานะภายใน



รูป 6.1 แผนภาพวงจรประสม

วงจรประสมประกอบด้วย ตัวแปรอินพุต เกทตรรก และตัวแปรเอาต์พุต เกทตรรกรับสัญญาณจากอินพุตและสร้างสัญญาณสู่เอาต์พุต กระบวนการนี้แปลงข้อมูลฐานสองจากอินพุตให้กลายเป็นข้อมูลเอาต์พุตที่ต้องการ ทั้งข้อมูลอินพุตและเอาต์พุตเป็นสัญญาณฐานสองคือมีเพียง 2 ค่า 0 และ 1

สำหรับตัวแปรของอินพุต n ตัว จะมีสภาวะประสม 2^n สภาวะ แต่ละสภาวะประสมของอินพุต จะมีเพียง 1 สภาวะประสมของเอาต์พุต วงจรประสมจะถูกพรรณนาโดย m บูลีนฟังก์ชัน แต่ละฟังก์ชันสำหรับตัวแปรเอาต์พุตแต่ละตัว เอาต์พุตฟังก์ชันแต่ละอันแสดงอยู่ในเทอมของตัวแปรอินพุต n ตัว

6.2 กระบวนการออกแบบ

Design Procedure

การออกแบบวงจรตรรกะผสมเริ่มจากปัญหา และจบด้วยวงจรตรรกะหรือบูลีนฟังก์ชันซึ่งจะนำไปสร้างวงจรต่อไป ขั้นตอนการออกแบบมีดังนี้

1. อ่านปัญหาให้เข้าใจ
2. พิจารณาจำนวนตัวแปรอินพุต และเอาต์พุตที่ต้องการ
3. กำหนดสัญลักษณ์ให้กับตัวแปรอินพุตและเอาต์พุต
4. เขียนตารางความจริงซึ่งแสดงความสัมพันธ์ระหว่างอินพุตและเอาต์พุต
5. ทำบูลีนฟังก์ชันสำหรับแต่ละเอาต์พุตให้ง่ายขึ้น
6. สร้างแผนภาพตรรกะ

เอาต์พุตในตารางความจริงอาจมีค่า 0 หรือ 1 สำหรับแต่ละสภาวะประสมของอินพุตที่เป็นจริง อย่างไรก็ตามอาจมีบางสภาวะประสมของอินพุตที่ไม่เกิดขึ้น ซึ่งเป็นเงื่อนไขไม่สนใจนั่นเอง

6.3 วงจรบวกเลข

Adder

ดิจิทัลคอมพิวเตอร์ทำงานหลายๆ อย่างเกี่ยวกับการประมวลผลข้อมูล หนึ่งในหน้าที่ทั้งหลายนี้คือการดำเนินการเลขคณิตอันได้แก่การบวก การลบ เป็นต้น การบวกเลขฐานสอง 2 บิตมีหลัก 4 ข้อคือ $0+0=0$, $0+1=1$, $1+0=1$ และ $1+1=10$ 3 ข้อแรกให้ผลบวก (sum) เป็นเลขฐานสอง 1 บิต สำหรับข้อที่ 4 นั้นเมื่อตัวตั้ง (augend) และตัวบวก (addend) ต่างมีค่าเป็น 1 ผลบวกจะมี 2 บิต บิตที่มีนัยสำคัญสูงกว่าเรียกว่าตัวทด (carry) ตัวทดจะไปบวกเข้ากับคู่ตัวตั้งและตัวบวกที่มีนัยสำคัญสูงกว่าถัดไป วงจรประสมซึ่งทำการบวกเลขฐานสองขนาด 2 บิต (ตัวตั้ง 1 บิต ตัวบวก 1 บิต) เรียกว่า วงจรบวกครึ่ง (half-adder) วงจรซึ่งบวกเลขฐานสอง 3 บิต (ตัวตั้ง, ตัวบวก และตัวทดก่อนหน้านั้น) เรียกว่าวงจรบวกเต็ม (full-adder) ซึ่งวงจรบวกครึ่งพอจะบอกได้ว่า วงจรชนิดนี้ 2 วงจรสามารถรวมกันได้เป็นวงจรบวกเต็ม 1 วงจร

วงจรบวกครึ่ง

จากคำอธิบายของวงจรบวกครึ่ง เราพบว่าวงจรนี้ต้องการ 2 อินพุต และ 2 เอาต์พุต ตัวแปรอินพุตคือตัวตั้ง และตัวบวก ตัวแปรเอาต์พุตคือผลบวก และตัวทด กำหนดสัญลักษณ์ของตัวแปรอินพุตโดยให้ x , y เป็นตัวตั้งและตัวบวกตามลำดับ และให้ s , C เป็นผลบวกและตัวทดตามลำดับ เป็นตัวแปรเอาต์พุต

เขียนตารางความจริงของวงจรวกครั้งได้ดังนี้

ตาราง 6.1 ตารางความจริงของวงจรวกครั้ง

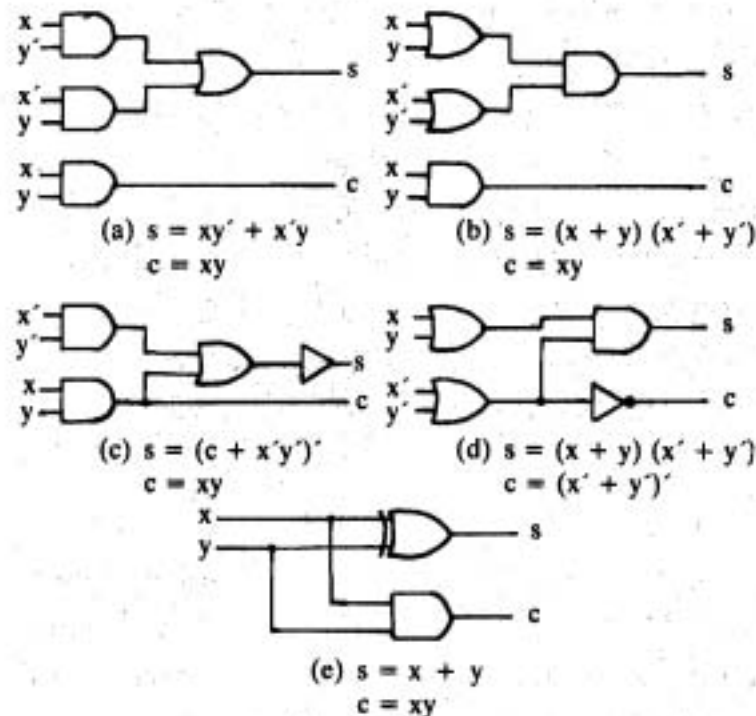
ตัวตั้ง	ตัวบวก	ตัวทด	ผลบวก
x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

จะเห็นว่าตัวทดเป็น 0 ยกเว้นเมื่ออินพุตเป็น 1 ทั้งคู่ และ S นั้น แทนบิตนัยสำคัญน้อยที่สุดของผลบวก

บูลีนฟังก์ชันที่ลดรูปแล้วสำหรับเอาต์พุตทั้งสองสามารถอ่านออกมาได้โดยตรงจากตารางความจริง ในรูปแบบผลบวกของผลคูณจะได้

$$S = x'y + xy'$$

$$C = xy$$



รูป 6.2 วงจรวกครั้งแสดงการจัดวงจรแบบต่างๆ

แผนภาพตรรกสำหรับวงจรวกตรีงปรากฏดังรูป 6.2 รูปนี้แสดงการจัดวงจรแบบต่าง ๆ ตามความต้องการของผู้ออกแบบ แต่ผลที่ได้สอดคล้องกับตารางความจริง

รูป 6.2 (a) เขียนวงจรจากสมการข้างบนนี้ คือในแบบผลบวกของผลคูณ

รูป 6.2 (b) เขียนวงจรในแบบผลคูณของผลบวก :

$$S = (x + y)(x' + y')$$

$$C = xy$$

รูป 6.2 (c) เริ่มจากฟังก์ชันในแบบผลบวกของผลคูณของ S จากตารางความจริงคือ

$$S = x'y + xy'$$

ซึ่งคือเอ็กซ์คลูสิฟ - (X) ของ x และ y คอมพลีเมนต์ของ S คือ อีควิเวเลนซ์ของ x และ y ซึ่งมีฟังก์ชันเป็น

$$S' = xy + x'y'$$

แต่ $C = xy$ ดังนั้น

$$S = (C + x'y')$$

รูป 6.2 (d) เราใช้ผลคูณของผลบวกกับ C จะได้

$$C = xy = (x' + y')$$

รูป 6.2 (e) เป็นวงจรวกตรีงซึ่งใช้เอ็กซ์คลูสิฟ-ออ ในการจัดวงจร ซึ่งจะเห็นต่อไปว่าเรานำวงจรนี้ 2 วงจรมาสร้างวงจรวกเต็มได้

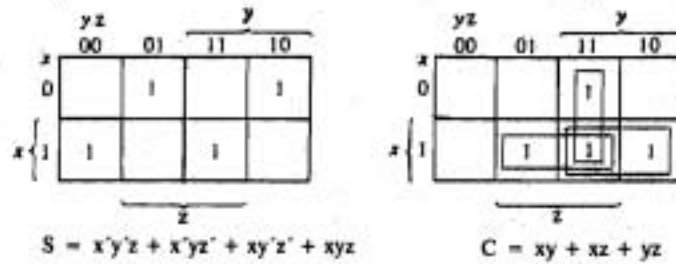
วงจรวกเต็ม

วงจรวกเต็มเป็นวงจรตรรกประสมซึ่งทำการบวกแบบเลขคณิต สำหรับเลขฐานสอง 3 อินพุต วงจรนี้ประกอบด้วย 3 อินพุต และ 2 เอาท์พุต กำหนดให้ x, y เป็นตัวแปรอินพุตคือตัวตั้ง และตัวบวกตามลำดับ และ z เป็นตัวแปรอินพุตอีกตัวหนึ่งซึ่งคือตัวทดที่มาจากตำแหน่งที่มีนัยสำคัญต่ำกว่าก่อนหน้านี้ เอาท์พุตจำเป็นที่ต้องมี 2 อันเพราะผลบวกเลขคณิตของเลขฐานสอง 3 บิต มีเรนจ์จาก 0 ถึง 3 โดยที่เลขฐานสอง 2 และ 3 (เทียบเลขฐานสิบ) มี 2 บิตให้ S และ C เป็นเอาท์พุตคือผลบวกและตัวทดตามลำดับ S เป็นตัวแปรฐานสองของเอาท์พุตที่ให้บิตที่มีนัยสำคัญน้อยที่สุดของผลบวก C คือตัวทดของเอาท์พุต ตารางความจริงของวงจรวกเต็มเป็นดังนี้

ตาราง 6.2 ตารางความจริงของวงจรวกเต็ม

ตัวตั้ง	ตัวบวก	ตัวทดเข้า	ตัวทอดออก	ผลบวก
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

จากตารางความจริงของวงจรวกเต็มจะเห็นว่า เมื่อทุกอินพุทบิตเป็น 0 เอาท์พุทจะเป็น 0 เอาท์พุท S จะมีค่าเป็น 1 เมื่ออินพุทเพียง 1 อัน ที่มีค่าเป็น 1 หรือ เมื่อทั้งสามอินพุทเป็น 1 ทั้งหมด เอาท์พุท C จะมีตัวทอดเป็น 1 เมื่อ 2 หรือ 3 อินพุทเป็น 1



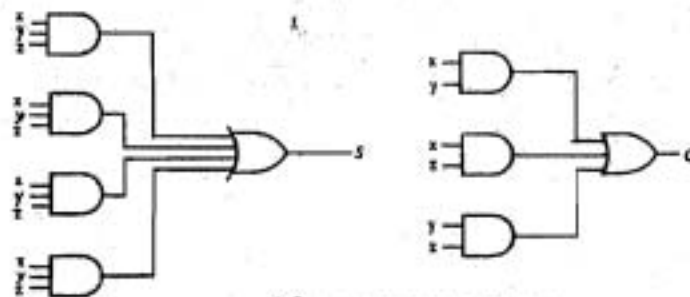
รูป 6.3 คาร์นอร์จแม็พสำหรับวงจรวกเต็ม

เอาท์พุท C และ S นำมาเขียนในคาร์นอร์จแม็พเพื่อลดรูป คาร์นอร์จแม็พแต่ละอันประกอบด้วยอินพุท 3 ตัว จึงมี 8 ช่อง ในที่สุดจะได้บูลีนฟังก์ชันดังนี้

$$S = x'y'z + x'yz' + xy'z' + xyz$$

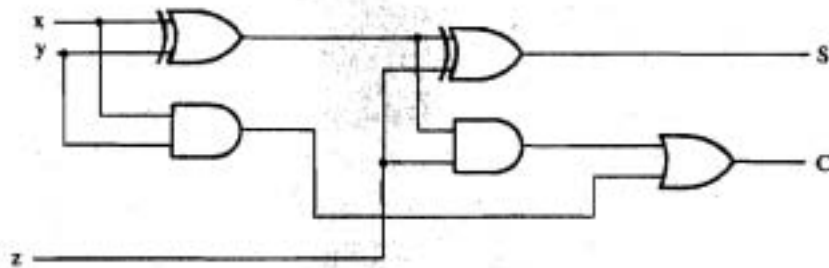
$$C = xy + yz + zx$$

ซึ่งสร้างวงจรได้ดังรูป 6.4



รูป 6.4 วงจรวกเต็มในแบบผลบวกของผลคูณ

วงจรวกเต็มอาจสร้างจากวงจรวกครึ่ง 2 วงจร และออกเกตดังรูป 6.5 เกทพุท S จากวงจรวกครึ่งตัวที่สองคือเอ็กซ์คลูซีฟ-ออร์ ของ z กับเกทพุทของวงจรวกครึ่งตัวแรก โดยมีรายละเอียดดังนี้



รูป 6.5 วงจรวกเต็มสร้างจากวงจรวกครึ่ง 2 วงจรและออกเกต

$$\begin{aligned}
 S &= z \oplus (x \oplus y) \\
 &= z'(xy' + x'y) + z(xy' + x'y) \\
 &= z'(xy' + x'y) + z(xy' + x'y) \\
 &= zy'z' + x'yz' + xyz + x'y'z
 \end{aligned}$$

และเกทพุท C เป็นดังนี้

$$\begin{aligned}
 C &= z(xy' + x'y) + xy \\
 &= xy'z + x'yz + xy \\
 &= (xy'z + xyz) + (x'yz + xyz) + xy \\
 &= xz(y' + y) + yz(x' + x) + xy \\
 &= xy + yz + zx
 \end{aligned}$$

6.4 วงจรลบเลข

Subtractor

การลบเลขฐานสอง 2 จำนวนอาจกระทำโดยใช้คอมพลิเมนต์ช่วย กล่าวคือ แปลงตัวลบให้เป็นคอมพลิเมนต์ แล้วนำไปบวกกับตัวตั้ง ดังรายละเอียดในบทที่ 2 หรืออาจลบเลขฐานสองโดยวิธีตรงไปตรงมา คือเอาตัวลบ (subtrahend) ไปลบออกจากตัวตั้ง (minuend) ถ้าพบกรณีตัวตั้งน้อยกว่าตัวลบ ต้องมีการยืม 1 จากหลักที่มีนัยสำคัญสูงกว่าที่อยู่ถัดไป 1 ที่ถูกขอยืมไปนี้จะถูกนำไปยังบิตคู่ที่มีนัยสำคัญสูงกว่าโดยสัญญาณฐานสองส่งผลออกมาเป็นเกทพุทของระยะ (stage) นั้นว่ามีกรขอยืม และส่งไปเป็นอินพุทสู่อินพุทที่สูงกว่าถัดไป วงจรลบมีทั้งวงจรถลบครึ่ง (half-subtractor) และวงจรถลบเต็ม (full-subtractor)

วงจรถลบครึ่ง

วงจรถลบครึ่งเป็นวงจรถรกรกประสมซึ่งลบเลขฐานสอง 2 บิต (ตัวตั้ง 1 บิต, ตัวลบ 1 บิต) ให้เอาท์พุทเป็นผลลบ (difference) และตัวยืม (borrow) ถ้าเกิดการยืมขึ้น กำหนด x และ y เป็นตัวตั้งและตัวลบตามลำดับ การพิจารณา $x-y$ ต้องตรวจสอบดูว่า $x : y$ ตัวใดมีขนาดสูงกว่า ถ้า $x \geq y$ จะได้กฎการลบ 3 ข้อ คือ $0-0=0$, $1-0=1$ และ $1-1=0$ ถ้า $x < y$ ได้กฎการลบข้อที่ 4 คือ $0-1$ ซึ่งจำเป็นต้องขอยืม 1 จากบิตสูงกว่าถัดไป 1 ที่ขอยืมมาได้นี้มีค่าเป็น 2 บวกเข้ากับตัวตั้ง เช่นเดียวกับในระบบเลขฐานสิบ ตัวยืมจะมีค่าเป็น 10 บวกเข้ากับหลักของตัวตั้ง เมื่อตัวตั้งเป็น 2 ผลลบจะเป็น $2-1=1$ วงจรถลบครึ่งมี 2 เอาท์พุท คือผลลบใช้สัญลักษณ์ D และตัวยืมใช้สัญลักษณ์ B ซึ่งผลิตสัญญาณฐานสองอันเป็นการแจ้งให้บิตถัดไปรู้ว่า 1 ถูกขอยืมไป ตารางความจริงสำหรับวงจรถลบครึ่งมีดังนี้

ตาราง 6.3 ตารางความจริงของวงจรถลบครึ่ง

ตัวตั้ง x	ตัวลบ y	ตัวยืม B	ผลลบ D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

เอาท์พุท B เป็น 0 ตรงเวลาที่ $x \geq y$ B จะเป็น 1 เมื่อ $x = 0$ และ $y = 1$ เอาท์พุท D เป็นผลดำเนินการเลขคณิต $2B + x - y$

บูลีนฟังก์ชันสำหรับเอาท์พุททั้งสองนี้ได้มาจากตารางความจริงของวงจรถลบครึ่ง :

$$D = x'y + xy'$$

$$B = x'y$$

น่าสนใจว่า D มีฟังก์ชันเช่นเดียวกับ S ในวงจรถลบครึ่ง

วงจรถลบเต็ม

วงจรถลบเต็มเป็นวงจรถรกรกประสมซึ่งทำการลบเลข 2 บิต และต้องคำนึงถึง 1 ที่อาจถูกยืมโดยบิตที่น้อยสำคัญต่ำกว่า วงจรถลบเต็มมี 3 อินพุท และ 2 เอาท์พุท กำหนดให้ x, y, z เป็นตัวตั้ง ตัวลบ และตัวยืมเข้าตามลำดับ D และ B เป็นผลลบ และ ตัวยืมออก (เป็นเอาท์พุท) ตารางความจริง สำหรับวงจรถลบเต็ม แสดงดังข้างล่าง

ตาราง 6.4 ตารางความจริงของวงจรวกเต็ม

ตัวตั้ง x	ตัวลบ y	ตัวยืมเข้า z	ตัวยืมออก B	ผลลบ D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

ตรรก 1 และ 0 ของเอาต์พุตได้จากการลบของ $x-y-z$

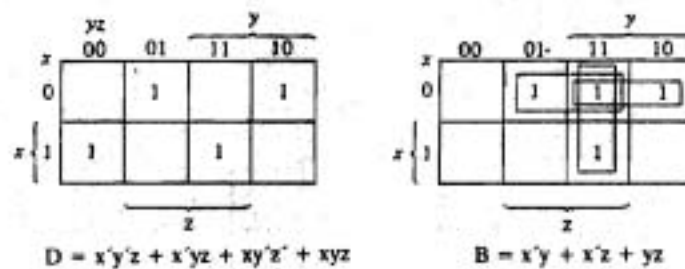
เมื่อสภาวะประสมของอินพุตมี $z = 0$ สถานการณ์เหมือนตารางความจริง ของวงจรถบครึ่ง (ทั้งสี่สภาวะประสมของวงจรถบครึ่ง)

เมื่อ $x = 0, y = 0, z = 1$ เราต้องขอยืม 1 จากบิตที่สูงกว่า ซึ่งทำให้ $B = 1$ และบวก 2 เข้ากับ x ดังนั้นจะได้ว่า $x-y-z = 2-0-1 = 1$ นั่นคือ $D = 1$

เมื่อ $x = 0$ และ $yz = 11$ เราต้องขอยืมอีกครั้ง ทำให้ $B = 1$ และ $x = 2$ จะได้ว่า $x-y-z = 2-1-1 = 0$ นั่นคือ $D = 0$

เมื่อ $x = 1$ และ $yz = 01$ เราได้ $x-y-z = 1-0-1 = 0$ ซึ่งทำให้ $B = 0$ และ $D = 0$

เมื่อ $x = 1, y = 1, z = 1$ เราต้องขอยืมมา 1 ทำให้ $B = 1$ และ $x = 3$ ดังนั้น $x-y-z = 3-1-1 = 1$ ดังนั้น $D = 1$



รูป 6.6 แผนที่สำหรับวงจรถบเต็ม

เมื่อลดรูปบูลีนฟังก์ชันโดยคาร์นอร์แม็พดังรูป 6.6 แล้ว จะได้ฟังก์ชันของเอาต์พุต
ในรูปแบบผลบวกของผลคูณคือ

$$D = x'y'z + x'yz' + xy'z' + xyz$$

$$B = x'y + x'z + yz$$

อีกครั้งหนึ่งที่เราจะสังเกตได้ว่าฟังก์ชันตรรกสำหรับเอาต์พุต D ในวงจรถบเต็มเหมือนกับ
เอาต์พุต S ในวงจรถบเต็ม ยิ่งไปกว่านั้น เอาต์พุต B ก็มีส่วนคล้ายเอาต์พุต C ในวงจรถบเต็ม
เว้นแต่ตัวแปรอินพุต x เท่านั้นที่ถูกคอมพลิเมนต์ ด้วยความคล้ายคลึงกันเช่นนี้จึงเป็นไปได้ที่จะแปลงวงจรถบเต็มให้เป็นวงจรถบเต็ม โดยเพียงแต่คอมพลิเมนต์กับอินพุต x ก่อน
ที่จะส่งเข้าสู่เกทซึ่งให้อาต์พุตเป็นตัวตอออก

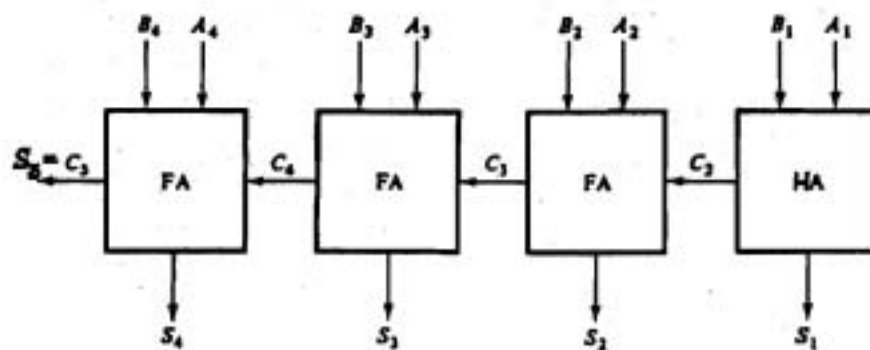
6.5 วงจรถบแบบขนาน Parallel Binary Adder

วงจรถบแบบขนานเป็นวงจรถบซึ่งบวกทุก ๆ บิตของตัวตั้งและตัวบวกพร้อม ๆ กัน

เราอาจนำวงจรถบครึ่งและบวกเต็มมาต่อกันได้เป็นวงจรถบแบบขนาน เช่น
ต้องการบวกตัวตั้ง $A_4 A_3 A_2 A_1$ เข้ากับตัวบวก $B_4 B_3 B_2 B_1$ ดังนี้

$$\begin{array}{r} A_4 A_3 A_2 A_1 \\ + B_4 B_3 B_2 B_1 \\ \hline S_5 S_4 S_3 S_2 S_1 \end{array}$$

ผลลัพธ์ที่ได้อาจมีตัวตอ S_5 ก็ได้ เมื่อลองพิจารณาตลับต่าง ๆ ที่บวกเข้าด้วยกัน จะเห็นว่า
ทุกบิตอาจมีการทดเข้าไปบวกด้วย ยกเว้นบิตขวามือสุดเท่านั้นที่ไม่มีตัวตอเข้า ดังนั้นสำหรับการ
บวกเลขฐานสองขนาด 4 บิตแบบขนาน เราใช้วงจรถบครึ่ง (H.A) 1 ตัว ต่อเข้ากับวงจรถบ
เต็ม (F.A) 3 ตัว ดังรูป 6.7



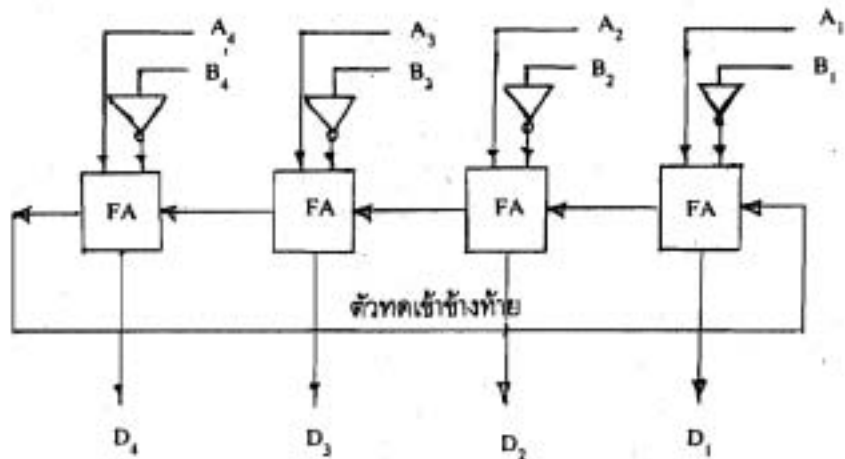
รูป 6.7 วงจรถบแบบขนาน บวกเลขฐานสองขนาด 4 บิต

อาศัยหลักการนี้เราอาจสร้างวงจรบวกแบบขนานสำหรับบวกเลขฐานสองขนาดจำนวนบิตที่ต้องการได้โดยเพิ่มวงจรบวกเต็มเข้าไป

6.6 วงจรลบแบบขนาน Parallel Binary Subtractor

ดังได้กล่าวไว้แล้วว่าเราอาจทำตัวลบให้เป็นคอมพลิเมนต์แล้วบวกเข้ากับตัวตั้ง ได้ผลลัพธ์ เป็นการลบเลข ใช้หลักการนี้มาสร้างวงจรลบเลขแบบขนานเช่น

$$\begin{array}{r} A_4 \ A_3 \ A_2 \ A_1 \\ - \ B_4 \ B_3 \ B_2 \ B_1 \\ \hline D_4 \ D_3 \ D_2 \ D_1 \end{array}$$



รูป 6.8 วงจรลบแบบขนาน ลบเลขฐานสองขนาด 4 บิต

จากรูป 6.8 เราใช้ 1's complement มาช่วยในการลบเลขแบบขนาน ในที่นี้เป็นการลบเลขขนาด 4 บิต จะเห็นว่าเราใช้วงจรบวกเต็ม 4 ตัว ในทำนองเดียวกันหากต้องการลบเลขฐานสองแบบขนานจำนวนกี่บิตที่ต้องการก็อาจนำหลักการนี้ไปใช้ได้

6.7 วงจรถอดรหัสแสดงผลเป็นตัวเลข 7 ส่วน

Seven-Segment Decoder

อุปกรณ์ดิจิทัลสมัยใหม่ให้ผลลัพธ์เป็นตัวเลขฐานสิบที่เอาท์พุทซึ่งให้อ่านค่าได้ทันที เช่นเครื่องคำนวณอิเล็กทรอนิกส์ และดิจิทัลโวลต์มิเตอร์ เราเคยรู้ว่าจะจรดิจิทัลที่อยู่ภายในเครื่องมือเหล่านี้ ผลิตเอาท์พุทเป็นเทอมของ 1 กับ 0 ถ้าเช่นนั้นเลขฐานสิบที่เราอ่านได้จากเครื่องมือเหล่านี้เกิดขึ้นได้อย่างไร

เป็นไปได้ที่จะสร้างตัวเลขฐานสิบโดยการประสมขีดต่างๆ ในแนวตั้ง และแนวนอนสำหรับตัวเลขหนึ่งๆ แม้ว่าภาพตัวเลขที่ได้ออกมาจะไม่ตรงตามต่อสายตาของเราก็ตาม แต่มันก็เหมาะสมสำหรับการสร้างตัวเลขฐานสิบจากอิเล็กทรอนิกส์ เพราะแต่ละขีดสามารถสว่างได้เป็นอิสระ ดังนั้นโดยการทำให้แต่ละขีดที่ประกอบกันเป็นตัวเลขฐานสิบสว่างขึ้นให้เหมาะสมกับแต่ละตัวเลขก็จะได้ภาพของเลขฐานสิบ รูป 6.9 (a) แสดงขีดทั้งเจ็ดที่อาจประกอบเป็นตัวเลขฐานสิบแต่ละตัว ซึ่งเรามักเรียกว่าการแสดงผลเป็นภาพตัวเลข 7 ส่วน (seven-segment display) รูป 6.9 (b) แสดงการแปรผันของสภาวะประสมของขีดที่จะสว่าง (on) เพื่อให้ได้เลขฐานสิบจากเลข 0 ถึง 9



(a) Arrangement of Vertical and Horizontal Bars Needed to Produce the Decimal Digits Zero Through Nine. The Bars are Referred to as "Segments," and are Labelled 'a' Through 'g'.

Decimal Digit	Appearance	Segments On
0		a,b,c,d,e,f
1		b,c
2		a,b,d,e,g
3		a,b,c,d,g
4		b,c,f,g
5		a,c,d,f,g
6		a,c,d,e,f,g
7		a,b,c
8		a,b,c,d,e,f,g
9		a,b,c,d,f,g

(b) Various Combinations of "On" Segments Give the Appearance of Decimal Digits.

รูป 6.9 การแสดงผลเป็นภาพตัวเลข 7 ส่วน

เนื่องจากแต่ละส่วน (segment) ของภาพแสดงผล (display) สามารถสว่างได้เป็นอิสระ ปัญหาของการทำให้เกิดตัวเลขฐานสิบขึ้น จึงลดลงเป็นการสร้างให้เกิดหนึ่งในจำนวนตัวเลขฐานสองขนาด 7 บิต ตำแหน่งของบิตในเลขฐานสองนี้สอดคล้องกับแต่ละส่วนที่ประกอบกันเป็นเลขฐานสิบ และค่าฐานสองที่ตำแหน่งนั้น สอดคล้องกับสถานะของส่วนซึ่งสว่าง ดังนั้น ถ้าตัวอักษรซึ่งแทนแต่ละส่วนสัมพันธ์กับตำแหน่งของบิตของเลขฐานสองขนาด 7 บิตเป็น (a, b, c, d, e, f, g) \rightarrow (B₇, B₆, B₅, B₄, B₃, B₂, B₁) แล้ว ตาราง 6.5 จะแสดงเลขฐานสอง 7 บิตที่เหมาะสมกัน เราสมมติให้ 0 แทนส่วนที่ดับ (off) และ 1 แทนส่วนที่สว่าง (on)

ตาราง 6.5 ความสัมพันธ์ส่วนที่สว่างของเลขฐานสิบ และเลขฐานสอง 7 บิตที่สอดคล้องกัน

The relationship between decimal digits, "on" segments, and the corresponding seven digit binary numbers.

<i>Decimal Digit</i>	<i>Segments On</i>	<i>Seven Digit Binary Number</i>
0	a b c d e f -	1 1 1 1 1 1 0
1	- b c - - - -	0 1 1 0 0 0 0
2	a b - d e - g	1 1 0 1 1 0 1
3	a b c d - - g	1 1 1 1 0 0 1
4	- b c - - f g	0 1 1 0 0 1 1
5	a - c d - f g	1 0 1 1 0 1 1
6	a - c d e f g	1 0 1 1 1 1 1
7	a b c - - - -	1 1 1 0 0 0 0
8	a b c d e f g	1 1 1 1 1 1 1
9	a b c d - f g	1 1 1 1 0 1 1

แม้ว่าเลขฐานสอง 7 บิตสามารถแทนเลขฐานสิบในเทอมของแต่ละส่วนที่ประกอบกันเป็นเลขฐานสิบนั้นก็ตาม แต่เราต้องการเลขฐานสองที่มีจำนวนบิตน้อยๆ เพื่อแทนเลขฐานสิบจาก 0 ถึง 9 โดยเฉพาะเลขฐานสองขนาด 4 บิต ซึ่งจะมีสภาวะประสม 16 สภาวะ ซึ่งมากเกินไปด้วยซ้ำที่จะแทนเลขฐานสิบทั้ง 10 ตัวนั้น นักออกแบบเครื่องมือดิจิทัลที่ให้เอาท์พุทเป็นตัวเลขต้องตัดสินใจ

(1) ควรจะใช้เลขฐานสองขนาด 7 บิตในการที่จะทำให้เกิดการแสดงผลเป็นเลขฐานสิบหรือไม่

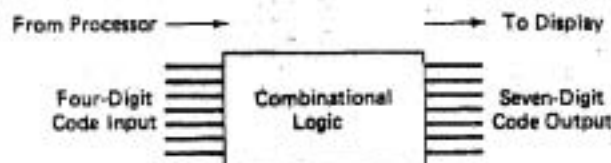
(2) ควรใส่รหัสแก่เลขฐานสิบด้วยเลขฐานสองน้อยๆ บิต เพื่อประหยัดวงจรในดิจิทัลโปรเซสเซอร์ (processor) แล้วสำหรับเอาท์พุทควรให้ผ่านวงจรถอดรหัส (decoder) ให้กลายเป็นเลขฐานสิบเพื่อแสดงผลเป็นเลขฐานสิบ 7 ส่วน ดีไหม

โดยส่วนรวมแล้วจะเป็นการง่ายในกระบวนการวิธีสำหรับเลขฐานสองจำนวนน้อยบิต

ดังนั้นผู้ออกแบบจึงใช้วิธีถอดรหัสให้เป็นเลข 7 ส่วน เมื่อมีความจำเป็นต้องให้เอาท์พุทแสดงผลเป็นเลขฐานสิบ รูป 6.10 (a) แสดงความสัมพันธ์ระหว่างรหัส BCD 4 บิตกับเลขฐานสอง 7 บิต ซึ่งเราต้องการในการให้แสดงผลเป็นตัวเลขฐานสิบ 7 ส่วน เราต้องการวงจรรวมประสมชนิดหลายเอาท์พุท (multiple-output combinational logic circuit) เพื่อที่จะแปลงเลขฐานสอง 4 บิตไปเป็นเลข 7 บิต วงจรรวมนี้แสดงดังรูป 6.10 (b) เราจะออกแบบวงจรรวมประสมนี้

Decimal Digit	Four-Digit Binary Code	Seven-Segment Code
0	0000	1111110
1	0001	0110000
2	0010	1101101
3	0011	1111001
4	0100	0110011
5	0101	1011011
6	0110	1011111
7	0111	1110000
8	1000	1111111
9	1001	1111011
-	1010	0
-	1011	0
-	1100	0
-	1101	0
-	1110	0
-	1111	0

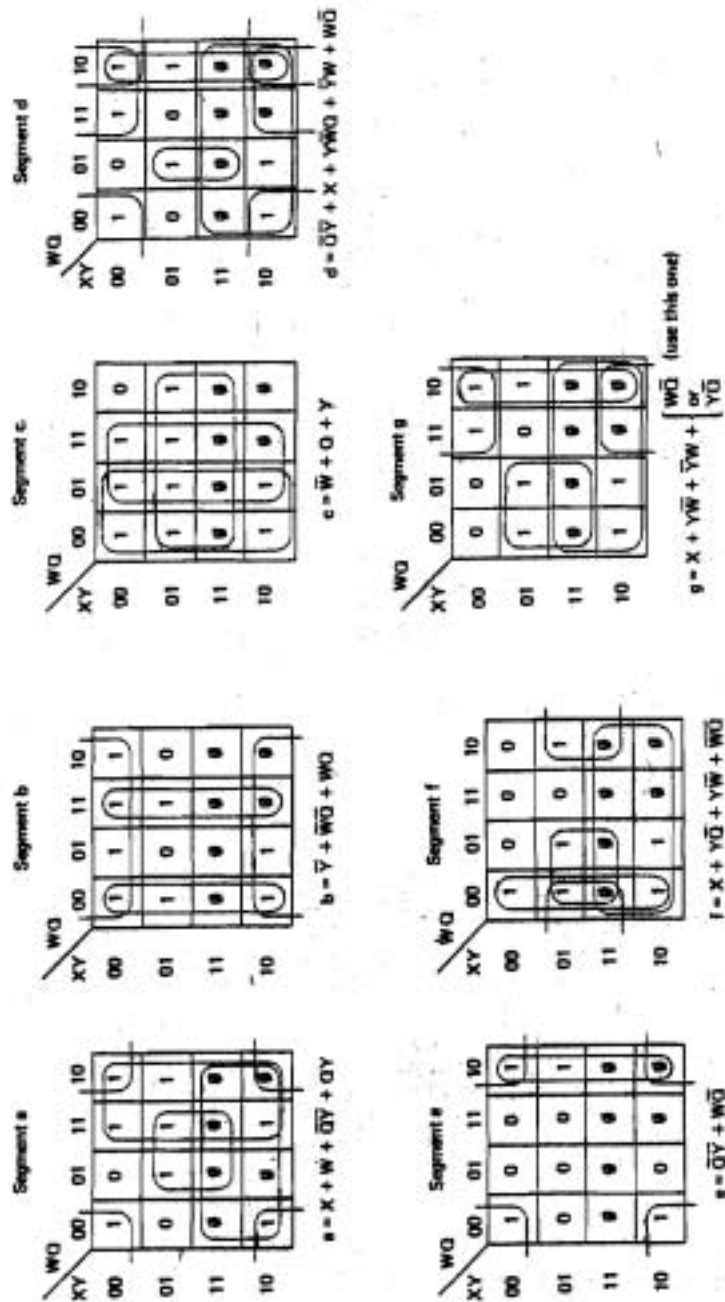
(a) Table Relating the Minimal Decimal Digit Encoding to Seven-Segment Display Code.



(b) Combinational Logic is Required to Convert the Processor's Four-Digit Code to a Seven-Segment Code for Display Devices.

รูป 6.10 แสดงตารางความสัมพันธ์ระหว่างรหัส BCD กับเลขฐานสอง 7 บิต (a) และวงจรรวมประสมที่จะแปลง (b)

ให้ XYWO เป็นรหัส BCD ใช้คาร์นอร์จแม็พลดรูปเพื่อให้ได้ฟังก์ชันที่เล็กที่สุดสำหรับตัวแปรเอาต์พุตแต่ละตัว ซึ่งคือแต่ละส่วนใน 7 ส่วนที่ประกอบกันเป็นเลขฐานสิบ เนื่องจากเลขฐานสิบจาก 0 ถึง 9 รหัสเป็น BCD 4 บิตได้ 10 สภาวะประสมของเลขฐานสองขนาด 4 บิต จึงมี 6 สภาวะประสมซึ่งไม่ได้ถูกใช้ เรียกว่าเป็นเงื่อนไขไม่สนใจ แทนด้วย 0 ในคาร์นอร์จแม็พ และสามารถนำมาช่วยลดรูปฟังก์ชันได้



Karnaugh maps for the seven-segment decoder are illustrated; all maps are unique except for that of segment 'g'.

รูป 6.11 คาร์นอร์จแม็พสำหรับแต่ละส่วนของตัวเลขฐานสิบ

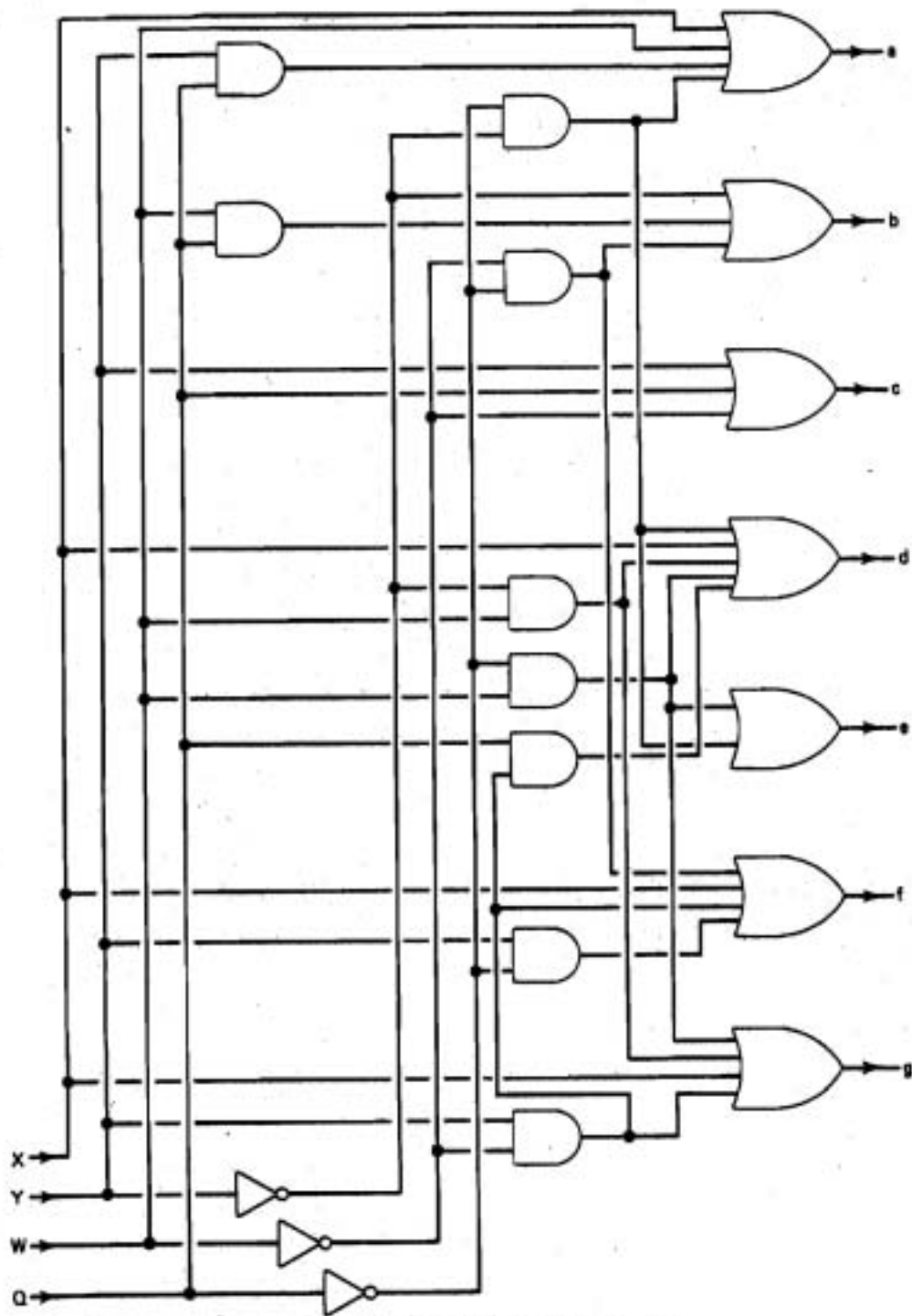
วงจรรกรกประสมหลายเอาร์ท์พุทสามารถเฉลี่ย (share) เทอมผลคูณในระหว่างสมการเอาร์ท์พุทได้ ซึ่งส่งผลเป็นการลดจำนวนเกทให้น้อยลง

ตาราง 6.6 แสดงเทอมผลคูณเฉลี่ยกับแต่ละสมการ

The use of certain product terms in more than one equation is investigated. The use of a product more than once means that it can be shared by several equations.

Product	SEGMENT EQUATIONS USING THIS PRODUCT							Number of Equations Using This Product
	a	b	c	d	e	f	g	
Q \bar{P}	✓			✓	✓			3
QY	✓							1
$\bar{W}Q$		✓				✓		2
WQ		✓						1
$\bar{Y}W$				✓			✓	2
$\bar{W}Q$				✓	✓		✓	3
$\bar{Y}\bar{W}Q$				✓				1
$\bar{Y}\bar{W}$						✓	✓	1
$\bar{Y}Q$						✓	✓	2

ถ้าเทอมผลคูณใดถูกใช้โดยสมการมากกว่า 1 สมการ ก็จะสามารถเฉลี่ย
รูป 6.12 เป็นวงจรที่ได้จากการออกแบบ



Design for a seven-segment decoder.

รูป 6.12 วงจรถอยตรัสแสดงผลเป็นตัวเลข 7 ส่วน

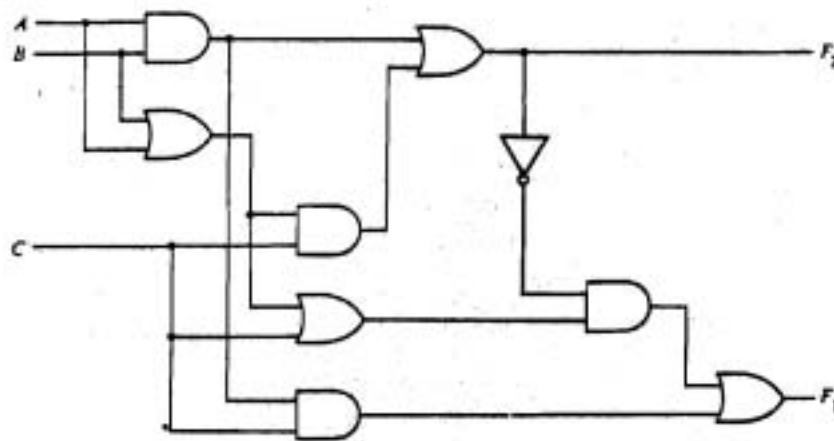
สรุป

อาจคิดว่าวงจรตรรกประสมคือ โครงข่าย (network) ของชิ้นส่วนตรรกที่ไม่มีควมจำ
วงจรแบบนี้มีความสัมพันธ์ระหว่างอินพุตและเอาต์พุตคงที่ และไม่ขึ้นอยู่กับลำดับของอินพุต
ก่อนหน้ามันซึ่งป้อนสู่วงจร

ในบทนี้ได้ยกตัวอย่างวงจรตรรกประสม เป็นวงจรเลขคณิต (arithmetic circuit) ได้แก่
วงจรวก วงจรลบ นอกจากนี้ยังมีวงจรถอดรหัสเป็นเลขฐานสิบซึ่งแสดงผลเป็นตัวเลข 7
ส่วน ได้กล่าวถึงการออกแบบวงจร โดยเริ่มจากปัญหา แยกแยะว่าอะไรคืออินพุต, เอาต์พุต
แล้วเขียนเป็นตารางความจริงซึ่งพรรณาคความสัมพันธ์ระหว่างทุก ๆ สภาวะประสมของ
ตัวแปรอินพุตและเอาต์พุตโดยกำหนดสัญลักษณ์แทนอินพุต เอาต์พุตของวงจร จากนั้น
ลดรูปฟังก์ชันบูลีนของเอาต์พุตโดยวิธีคาร์นอร์จแม็พ เมื่อได้นิพจน์ที่ลดรูปแล้วจึงเขียน
วงจรจากนิพจน์นั้น ๆ

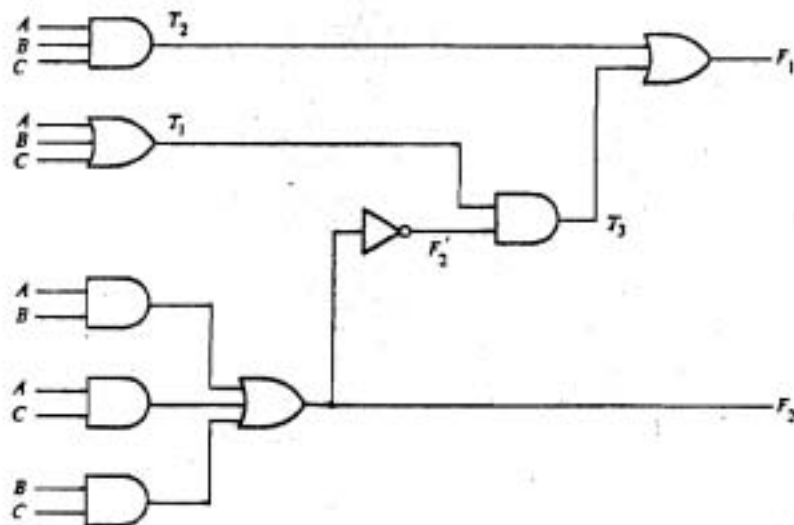
แบบฝึกหัด

- 6.1 วงจรประสมหนึ่งมี 4 อินพุต และ 1 เอาต์พุต เอาต์พุตมีค่า 1 เมื่อ
- (1) ทุกอินพุตเป็น 1
 - (2) ไม่มีอินพุตใดเป็น 1
 - (3) อินพุตเป็นจำนวนคี่มีค่าเป็น 1
- ก. จงเขียนตารางความจริงของวงจรมี
- ข. หาเอาต์พุตฟังก์ชันที่ลดรูป ในแบบ ผลบวกของผลคูณ
- ค. หาเอาต์พุตฟังก์ชันที่ลดรูป ในแบบผลคูณของผลบวก
- ง. เขียนวงจรทั้งสอง
- 6.2 จงออกแบบวงจรซึ่งมีอินพุตเป็นเลขฐานสองขนาด 4 บิต เอาต์พุตเป็น 2's complement ของอินพุต
- 6.3 จงออกแบบวงจร แปลงรหัส 8421 ให้เป็นรหัสเกิน 3
- 6.4 จงวิเคราะห์วงจรในรูป 6.13 เพื่อหาบูลีนฟังก์ชันของเอาต์พุตทั้งสอง แล้วอธิบายการทำงานโดยอาศัยตารางความจริงที่เขียนขึ้น



รูป 6.13 โจทย์แบบฝึกหัด 6.4

6.5 จงวิเคราะห์และอธิบายการทำงานของวงจรในรูป 6.14



รูป 6.14 โจทย์แบบฝึกหัด 6.5

6.6 เลขฐานสิบสามารถแทนได้ด้วยรหัส BCD ซึ่งมี 4 บิต ถ้าต้องการวงจรถอดรหัสเช่นนี้ จะต้องใช้ 4 อินพุต และ 10 เอาต์พุต (ตามจำนวนเลขพื้นฐานในระบบฐานสิบ) จงออกแบบวงจรถอดรหัส BCD ให้เป็นเลขฐานสิบ

6.7 จงออกแบบวงจรถอดเลขฐานสอง ซึ่งมีตัวตั้ง ตัวคูณ และผลคูณ ดังนี้

$$\begin{array}{r}
 b_1 \ b_0 \text{ (ตัวตั้ง)} \\
 \times a_1 \ a_0 \text{ (ตัวคูณ)} \\
 \hline
 P_4 \ P_3 \ P_1 \ P_0 \text{ (ผลคูณ)}
 \end{array}$$