บทที่ 3
ลอจิกและวงจรคอมไบเนชั่นลอจิก
Logic Functions and Combinational Logic Circuit

เค้าโครง

- การแทนค่าลอจิก (Logic function Representation)
- อินเวอร์เตอร์ (Inverter)
- แอน (And)
- แนน (Nand)
- ออร์ (Or)
- นอร์ (Nor)
- เอ็กคูสซีพ (Exclusive function : XOR , XNOR)
- วงจรคอมไบเนชั่น (Combination Logic Circuit)

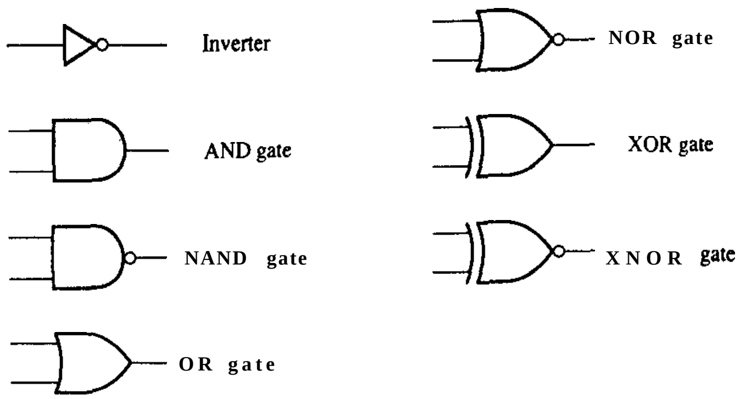
การศึกษาในบทนี้จะเป็นศึกษาการวิเคราะห์ดิจิตอลลอจิก และการออกแบบ การศึกษาขั้นแรกจะเป็นพื้นฐานของดิจิตอลลอจิก และคอมไบเนชั่นลอจิก ส่วนแรกของบทนี้จะเป็นพื้นฐานการศึกษาในบทต่อไป

3.1. การแทนค่าลอจิก (Logic Function Representation)

ดิจิตอลลอจิก คือ การแทนคุณสมบัติด้วยสภาวะลอจิก 2 สภาวะ คือ ลอจิก High และ ลอจิก Low รูปแบบการทำงานของลอจิกจะอยู่ในรูปของคลื่น (WaveForms) พื้นฐานการทำงานของลอจิกมี 7 ชนิด ซึ่งจัดอยู่ส่วนของดิจิตอลอีเลคโทรนิค การนำลอจิกที่มีการกำหนดการทำงานไปประยุกต์ใช้งาน หรือนำไปรวมเป็นวงจรดิจิตอล เช่น วงจรบวกเลข (Adder) วงจรถอดรหัส (Decoder) วงจรเข้ารหัส (Encoder) วงจรเลือกข้อมูล (Multiplexer) วงจรกระจายข้อมูล (Demultiplexer) วงจรนับ (Counter) และหน่วยความจำรีจีสเตอร์ (Storage Register)

พื้นฐานการทำงานของลอจิกทั้ง 7 ชนิด ที่ใช้งานในวงจรดิจิตอลอีเลกโทรนิก แต่ละชนิดเราเรียกว่า Logic Gate ซึ่งมีการกำหนดสัญลักษณ์ (Symbol) สมการลอจิก (Logic Equation) ชื่อลอจิก (Name) และตารางแสดงคุณสมบัติ (Truth Table) ของแต่ละชนิดดังต่อไปนี้

สัญลักษณ์ลอจิก (Logic Symbol) ที่มีการกำหนดแทนสัญลักษณ์ด้วยรูปในการกำหนดหน้าที่ การกำหนดจะเป็นไปตามมาตราฐานของลอจิกเกต ดังแสดงในรูปต่อไปนี้



รูป 3.1 Standard Digital Symbols

สัญลักษณ์ของลอจิกสามารถแทนด้วยสมการทางลอจิก (Logic Equations) ที่ใช้แทนหน้าที่การทำงานของสัญญลักษณ ฟังชั่นของสมการลอจิกจะแทนด้วยตัวอักษรตัวใหญ่ที่กำหนดค่าของอินพุต เอ้าพุตลอจิก หรือหน้าที่ของลอจิกสามารถแสดงด้วยตารางแสดงคุณสมบัติ ที่แสดงความเป็นไปได้ของสถานะทางอินพุต และเอ้าพุต สำหรับค่าของวงจร ถ้ามีตัวแปร n อินพุต ความเป็นไปได้ของสถานะคือ 2^n ค่าของอินพุตเหล่านี้จะแสดงการกำหนดสภาวะด้วยค่าของเลขฐานสองในตารางแสดงคุณสมบัติ

หน้าที่ของลอจิกพื้นฐานทั้ง 7 ชนิด คือ NOT , AND , NAND , OR , NOR , Exclusive NOR แต่ละชนิดจะมีการอธิบายการทำงานดังต่อไปนี้

3.2 อินเวอร์เตอร์ (INVERTER FUNCTION)

Inverter คือลอจิกเกต ที่มีเพียง 1 อินพุต และ 1 เอ้าพุต การทำงานของอินเวอร์เตอร์จะให้ผลลัพธ์ที่มีค่าตรงกันข้ามกับอินพุต ถ้าค่าของอินพุตมีค่า LOW เอ้าพุตของวงจรอินเวอร์เตอร์จะมีค่า HIGH หรือถ้าค่าของอินเวอร์เตอร์มี่ค่า HIGH เอ้าพุตของวงจรจะมีค่า LOW หรือเป็นการทำงานในลักษณะคอมพลีเม้นต์ทางด้านอินพุต ตัวอินเวอร์เตอร์ เราเรียกว่า NOT gate ซึ่งกำหนดการทำงานเป็น NOT Function.

Inverter

Logic Symbol

Equation

$$Y = \overline{A}$$

Truth Table

A	Y
0	1
1	0

รูป 3.2 Inverter Logic Function

จากรูป 3.2 แสดงสัญลักษณ์ สมการลอจิก และตารางแสดงคุณสมบัติสำหรับอินเวอร์เตอร์ ในรูป มีอินพุต A และมีเอ้าพุต Y สมการเอ้าพุตคือ $Y = \overline{A}$ เราอ่านว่า A inverted หรือ A bar หรือ A NOT

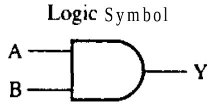
ตารางแสดงคุณสมบัติของอินเวอร์เตอร์ในรายการของสภาวะอินพุตและเอ้าพุต เราจะเห็นว่าเอ้าพุตที่ได้จากตารางจะตรงกันข้ามกับอินพุต

3.3 แอน (AND FUNCTION)

AND Gate ลอจิกชนิดนี้จะมีอินพุต 2 อินพุตหรือมากกว่า และมีเอ้าพุตเพียง 1 เอ้าพุตเท่านั้น ผลลัพธ์ที่เกิดขึ้นจากการทำงานของลอจิกชนิดนี้ถ้าเป็น 1 ค่าของอินพุตทุกตัวจะต้องมีค่าเป็น 1 ทั้งหมด แต่ถ้าอินพุตตัวใดตัวหนึ่งมีค่าเป็น 0 เอ้าพุตจะมีค่าเป็น 0

AND

Logic Symbol



Equation

$$Y = AB$$

Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

รูป 3.3 AND Logic Function

จากรูปที่ 3.4 แสดงสัญลักษณ์ สมการ ตารางแสดงคุณลักษณะ สำหรับลอจิกชนิด 2 อินพุต ที่มีตัวแปร 2 ตัว คือ A และ B จะได้เอ้าพุตคือ AB โดยปกติอ่านว่า A AND B ถ้าในกรณีลอจิกชนิด 4 อินพุต เอ้าพุตจะเป็น A AND B AND C AND D การเขียนเอ้าพุตของลอจิก AND คือ AB , A.B , AxB ตาราง 3.1 แสดงคุณสมบัติของลอจิก AND ชนิด 2 , 3 , 4 อินพุต

ตาราง 3.1 AND Truth tables

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

A	B	C	ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

A	B	C	D	ABCD
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

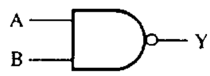
3.4 แนน (NAND FUNCTION)

คำว่า NAND หมายถึงการรวมลอจิก AND + NOT หรือเรียกว่า NOT AND ซึ่งผลลัพธ์ของลอจิกชนิดนี้จะตรงกันข้ามกับลอจิก AND วงจรลอจิกของ NAND เกต จะมี 2 อินพุตหรือมากกว่า และจะมีเพียง 1 เอ้าพุตเท่านั้น เราจะเห็นว่าทุกครั้งที่มีลอจิกเอ้าพุตเป็น 0 อินพุตทั้งหมดของ NAND เกตจะมีค่าเป็น 1

ลอจิก NAND จะอ้างถึงการ Complement ฟังชั่นของลอจิก AND สัญลักษณ์ที่ใช้แสดงใรูป 3.4 ซึ่งเป็นชนิด 2 อินพุต NAND Gate พร้อมสมการและตารางแสดงคุณลักษณะ

NAND

Logic Symbol



Equation

$Y = \overline{AB}$

Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

รูปที่ 3.4 NAND Logic Function

ตาราง 3.2 NAND Truth tables

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	C	$\overline{ABC}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

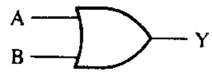
A	B	C	D	ABCD
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

3.5 ออ (OR FUNCTION)

OR Gate เป็นลอจิกที่มีอินพุตตั้งแต่ 2 อินพุตขึ้นไปและมีเพียง 1 เอ้าพุต การทำงานของลอจิก OR Gate ถ้าอินพุตใดอินพุตหนึ่งของลอจิกมีค่าเป็น 1 ผลลัพธ์ของเอ้าพุตก็จะมีค่าเป็น 1 ค่าของเอ้าพุตจะมีค่าเป็น 0 ก็ต่อเมื่ออินพุตทุกตัวมีค่าเป็น 0

OR

Logic Symbol



Equation

$$Y = A + B$$

Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

รูปที่ 3.5 OR Logic Function

ตาราง 3.3 OR Truth tables

A B	A + B
0 0	0
0 1	1
1 0	1
1 1	1

A B C	A + B + C
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

A B C D	A + B + C + D
0 0 0 0	0
0 0 0 1	1
0 0 1 0	1
0 0 1 1	1
0 1 0 0	1
0 1 0 1	1
0 1 1 0	1
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	1
1 0 1 1	1
1 1 0 0	1
1 1 0 1	1
1 1 1 0	1
1 1 1 1	1

3.6 นอ (NOR FUNCTION)

NOR Function หมายถึงการนำ NOT + OR เอ้าพุตของ NOR Gate จะมีค่าเป็น 1 ก็ต่อเมื่ออินพุตทั้ง 2 ของ NOR gate จะมีค่าเป็น 0 ตารางการทำงานของ NOR Gate จะตรงกันข้ามกับ OR Gate

NOR

Logic Symbol

Equation

$Y = A + B$

Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

รูปที่ 3.6 NOR Logic Function

ตาราง 3.4 NOR Truth tables

TABLE 3-4 NOR Truth Tables

AB	A+B	A B C	$\overline{A+B+C}$	A B C D	$\overline{A+B+C+D}$
0 0	1	0 0 0	1	0 0 0 0	1
0 1	0	0 0 1	0	0 0 0 1	0
1 0	0	0 1 0	0	0 0 1 0	0
1 1	0	0 1 1	0	0 0 1 1	0
		1 0 0	0	0 1 0 0	0
		1 0 1	0	0 1 0 1	0
		1 1 0	0	0 1 1 0	0
		1 1 1	0	0 1 1 1	0
				1 0 0 0	0
				1 0 0 1	0
				1 0 1 0	0
				1 0 1 1	0
				1 1 0 0	0
				1 1 0 1	0
				1 1 1 0	0
				1 1 1 1	0

3.7 เอ็กคูสซีฟ (EXCLUSIVE FUNCTIONS : XOR AND XNOR)

Exclusive OR gate คือลอจิกเกตที่มี 2 อินพุตและ 1 เอ้าพุต การทำงานของเอ้าพุต Exclusive OR gate จะมีค่าเป็น 1 หรือ High เมื่ออินพุตทั้งสองไม่เท่ากัน แต่ถ้าอินพุตทั้งสองเป็น 1 หรือเป็น 0 ทั้งคู่ เอ้าพุตจะมีค่าเป็น 0 หรือ Low สัญลักษณ์เป็น XOR

XOR

Logic Symbol

Equation

$$Y = A \oplus B$$

Truth Table

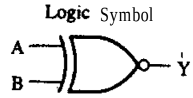
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

รูปที่ 3.7 XOR Logic Function

ส่วนการทำงานของ Exclusive NOR นั้นจะทำงานตรงกันข้ามกับ XOR ถ้าอินพุตของ XNOR มีค่าเท่ากันไม่ว่าจะเป็น 1 หรือ 0 ทั้งคู่ เอ้าพุตจะเป็น 1 หรือ High

XNOR

Logic Symbol



Equation

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

Truth Table

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

รูป 3.8 EX-OR LOGIC FUNCTION

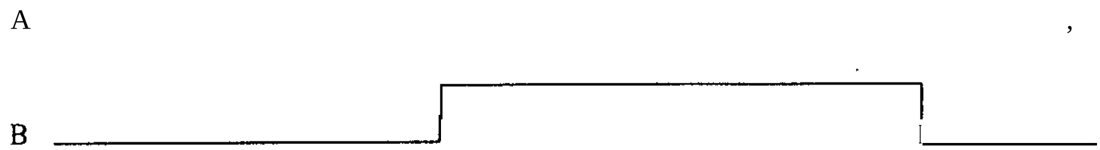
3.8 การทำงานของพัลซ์ลอจิก(PULSED LOGIC GATE OPERATION)

การวิเคราะห์สัญญาณหรือพัลซ์ เราจะต้องดูการทำงานของวงจรลอจิกเกตภายใต้เงื่อนไขที่กำหนด เมื่อสัญญาณอินพุตเปลี่ยนแปลง สัญญาณเอ้าพุตก็จะเปลี่ยนแปลงตามเงื่อนไข ตามกฎเกณฑ์ของลอจิกแต่ชนิด หรือตารางแสดงคุณลักษณะของเกตแต่ละตัว

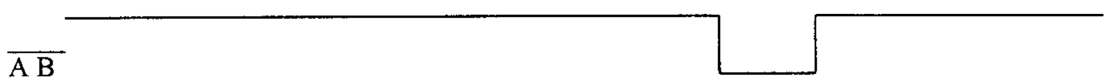
ตารางแสดงคุณลักษณะใช้กำหนดหน้าที่ของลอจิก แต่ละสัญญาณของอินพุตจะต้องนำมารวมกันเพื่อก่อให้เกิดสัญญาณเอ้าพุต การประยุกต์ใช้งานของค่าคงที่หรือค่า Steady state ของลอจิกเกต บางครั้งจะอ้างถึง Static Operation ตัวอย่างของค่า 0 ของอินพุตหนึ่งของ Nand gate และค่า

1 ของอินพุตอีกอินพุตหนึ่งของ Nand gate ผลลัพธ์ที่ตารางเอ้าพุตจะเป็น 1 การทำงานของลอจิกเกต โดยการป้อนอินพุตในรูปของรูปคลื่นดิจิตอล ที่เรารู้จักกันใน Pulsed Operation การเกิดเอ้าพุตของเกตแต่ละตัวได้จากตารางแสดงคุณลักษณะของเกตนั้นๆ ดังตัวอย่าง

ตัวอย่าง 3.1 การเกิดรูปคลื่นของ Pulsed NAND Operation. จงหารูปคลื่นเอ้าพุต จาก 2 อินพุต NAND Gate ด้วยค่าของอินพุตที่กำหนดให้

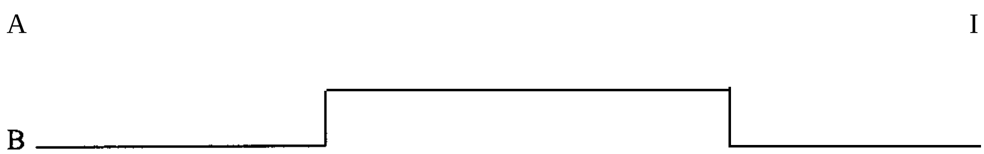


ผลลัพธ์ที่เกิดขึ้นจาก อินพุตของ NAND Gate มีดังนี้ เอ้าพุตจะมีค่าเป็น 0 ถ้าอินพุตเป็น 1 ทั้งคู่ ผลลัพธ์ดูจากรูปคลื่นดังต่อไปนี้

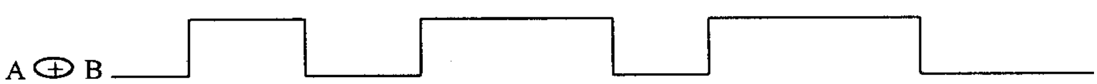


ตัวอย่าง 3.2 การทำงานของรูปคลื่นที่เกิดจาก XOR Operations

ให้รูปคลื่นที่เกิดจากการทำงานของเอ้าพุตของ XOR Gate ซึ่งป้อนข้อมูลจากอินพุต A และ B ตามรูปคลื่นที่กำหนดให้



การทำงานของ XOR Gate เอ้าพุตจะมีค่าเป็น 1 เมื่อสัญญาณอินพุตทั้งสองมีค่าไม่เท่ากัน ดังแสดงผลลัพธ์ของรูปคลื่นดังต่อไปนี้



3.9 วงจรคอมไบเนชั่น (COMBINATION LOGIC CIRCUITS)

พื้นฐานการทำงานของวงจรลอจิก NOT , AND , NAND , OR , NOR , XOR , XNOR เป็นลอจิกเกตที่ใช้ในวงจรดิจิตอล ฟังชั่นการทำงานของลอจิกต่อไปนี้ จะนำมารวมกันในการออกแบบวงจรที่ใช้งานตามที่กำหนดเราเรียกว่าวงจรรวม (Integrated Circuits)

3.9.1 หน้าที่ของวงจรคอมไบเนชั่นและตารางแสดงคุณลักษณะ

(Combinational Logic Circuit Functions and Truth Table Analysis)

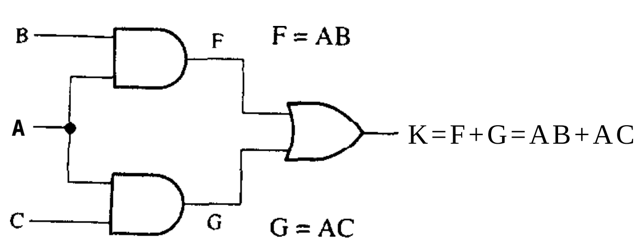
วงจรคอมไบเนชั่น คือ มีโครงสร้างการเชื่อมต่อของลอจิกเกตต่างๆ เข้าด้วยกัน เพื่อกำหนดหน้าที่การทำงานใหม่ กฎเกณฑ์ของลอจิกจะให้ผลลัพธ์ตามที่ต้องการ โดยที่เอ้าพุตของลอจิกเกตจะเกิดขึ้นตามอินพุตที่ป้อนให้กับลอจิกเกต

วงจรคอมไบเนชั่น คือการวิเคราะห์การทำงานจากอินพุตสู่เอ้าพุต เมื่อเราได้เรียนรู้การวิเคราะห์วงจรคอมไบเนชั่น จะช่วยให้ท่านเขียนสมการและวิเคราะห์เอ้าพุตของลอจิกเกตแต่ละตัวในวงจร ก่อนที่รู้ผลลัพธ์สุดท้าย

วงจรดิจิตอลลอจิก สามารถวิเคราะห์การใช้จากตารางแสดงคุณลักษณะ ที่กำหนดสภาวะทางอินพุตที่เปลี่ยนแปลงและผลลัพธ์ทางเอ้าพุตที่เกิดจากการเปลี่ยนแปลงทางอินพุต ลอจิกเกตแต่ละตัว จะมีผลลัพธ์ที่เกิดขึ้นตามคุณสมบัติของลอจิกเกตนั้นๆ ตารางแสดงคุณสมบัติจะใช้ในการวิเคราะห์การทำงานดังตัวอย่างต่อไปนี้

ตัวอย่าง 3-3 การวิเคราะห์วงจรคอมไบเนชั่น

ปัญหา จงวิเคราะห์วงจรตามสมการลอจิกที่กำหนดให้และหาเอ้าพุต

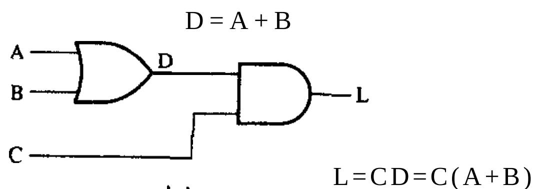


Inputs			Intermediate nodes		Output
A	B	C	F	G	K
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

K=AB+AC

รูปที่ 3.9 วงจรคอมไบเนชั่นตัวอย่างที่ 3.1

ตัวอย่าง 3.4 การวิเคราะห์วงจร

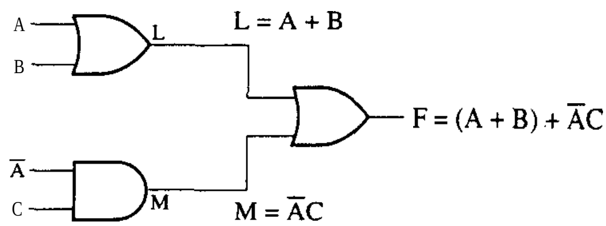


Inputs			Intermediate node	Output
A	B	C	D	L
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

L = C(A + B)

รูปที่ 3.10 วงจรคอมไบเนชั่นตัวอย่างที่ 3.2.

ตัวอย่าง 3.5 การวิเคราะห์วงจร



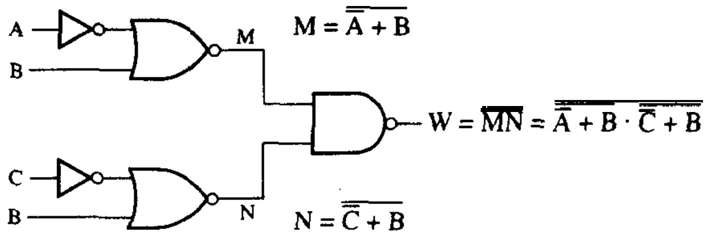
Inputs			Intermediate nodes		output
A	B	C	L	M	F
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	1

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

Note: F = A + B + C by inspection of the truth table.

รูปที่ 3.11 วงจรคอมไบเนชั่นตามตัวอย่างที่ 3.3

ตัวอย่าง 3.6 การวิเคราะห์วงจร

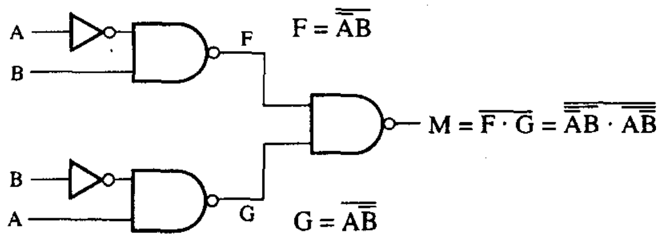


Inputs			Intermediate nodes		output
A	B	C	M	N	W
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	0	0	1

$W = \overline{\overline{\bar{A} + B} \cdot \overline{\bar{C} + B}}$

รูปที่ 3.12 วงจรคอมไบเนชั่นตามตัวอย่างที่ 3.4

ตัวอย่าง 3.7 การวิเคราะห์วงจร



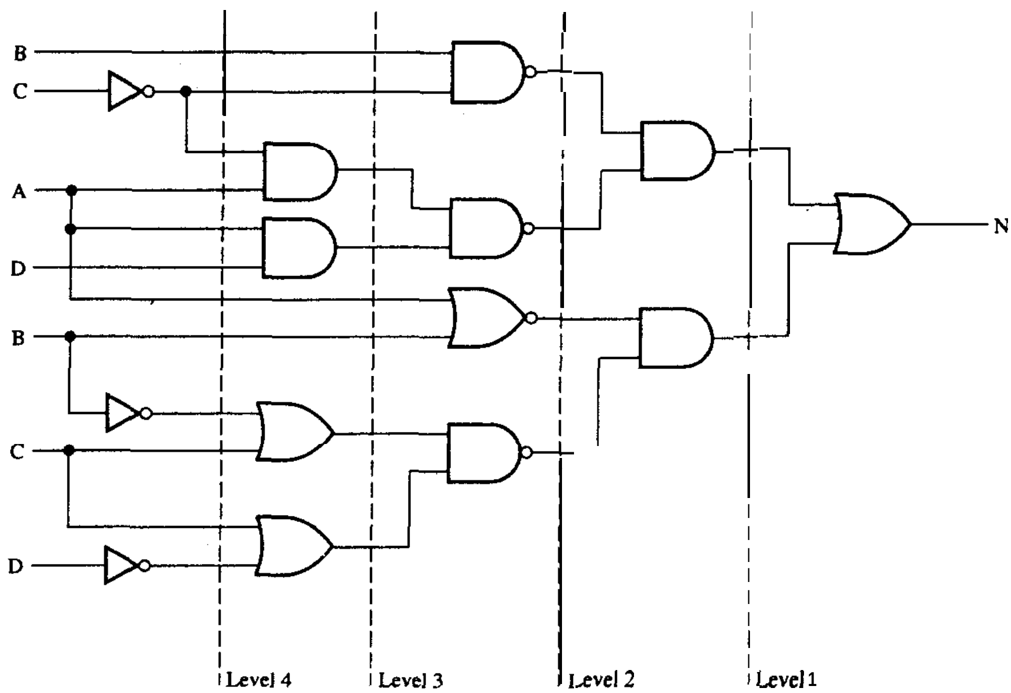
Inputs		Intermediate nodes		output
A	B	F	G	M
0	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	1	1	0

$M = \overline{AB} \cdot A\overline{A}\oplus B$ truth table.

รูปที่3.13 วงจรคอมไบเนชั่นตามตัวอย่าง 3.5

วงจรคอมไบเนชั่นตามตัวอย่างที่กล่าวมาแล้วนั้น ประกอบด้วยลอจิกหลายระดับ เริ่มต้นจากระดับหนึ่งที่เอ้าพุต และมีระดับอื่นๆ ที่อินพุตเพิ่มขึ้นเรื่อยๆ ตัว Inverter ทุกตัวที่ระดับอินพุตไม่ถือว่าเป็นระดับที่แยกเป็นระดับอิสระ

ผลรวมของจำนวนระดับต่างๆที่กำหนดจำนวนเกตสูงสุด นั่นคือสัญญาณดิจิตอลจะผ่านก่อนที่จะออกสู่เอ้าพุต จากรูป 3.14 แสดงระดับต่างๆของวงจรดิจิตอล



รูปที่ 3.14 วงจรดิจิตอล 4 ระดับ

3.9.2 วงจรลอจิกจากสมการพีชคณิตบูลีน

(Logic Circuits from Boolean Equations)

วงจรลอจิก สามารถกำหนดโครงสร้างจากกฎเกณฑ์ของลอจิก โดยอยู่ในรูปของ กฎเกณฑ์ลอจิก (Logic Expression) กับพื้นฐานการทำงานของลอจิกเกต ซึ่งเป็นการรวมเอ้าพุตของเกตเหล่านี้กับลอจิกเกตอื่นๆ จะได้ผลลัพธ์สุดท้าย มีขั้นตอนดังต่อไปนี้

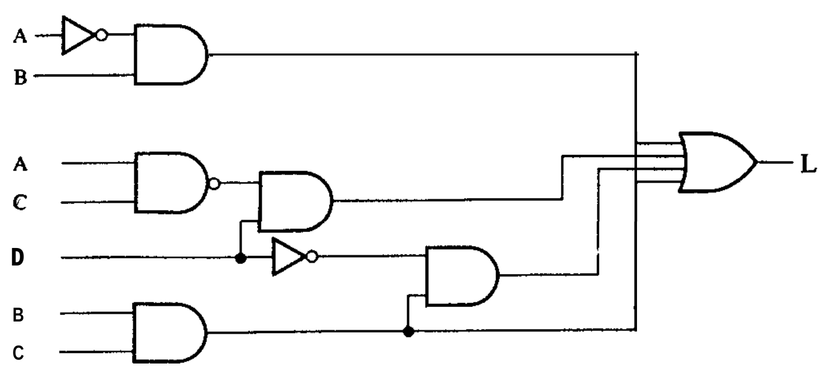
1. กำหนดค่าของตัวแปรอินพุต ทั้งเป็นค่า True และค่า Inverted ที่กำหนดในวงจร
2. จับกลุ่มของกฎเกณฑ์ที่กำหนดหน้าที่ของลอจิกตามต้องการ
3. กำหนดรูปแบบของเทอมอินพุตกับพื้นฐานของลอจิกเกต
4. ทำการรวมเอ้าพุตของลอจิกเกตกับเกตตัวอื่นๆที่เป็นกฎที่ต้องการ ของเอ้าพุต

ตัวอย่าง 3.8 การเขียนวงจรลอจิก

ปัญหา การออกแบบวงจรลอจิกจากสมการลอจิก

การแก้ปัญหา แทนค่าจากเทอมที่เป็นอิสระของสมการด้วยลอจิกเกต และอยู่ในรูปแบบของเอ้าพุต

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

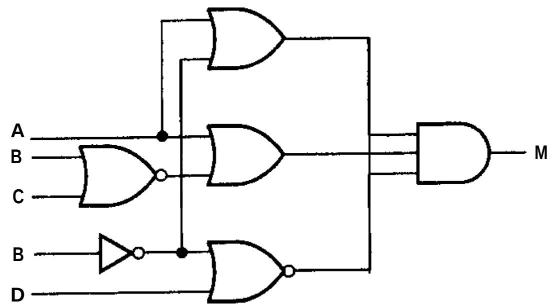


$L = \overline{A}B + \overline{AC}D + BC + BC\overline{D}$

รูปที่ 3.15. การเขียนวงจรลอจิกตามตัวอย่าง 3.6

ตัวอย่าง 3.9 การเขียนวงจรจากสมการ

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

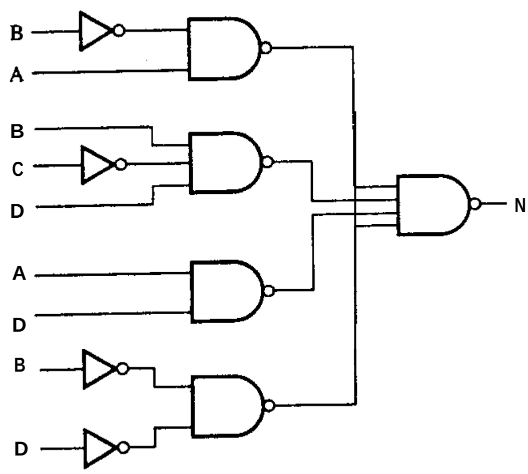


$$M = (A + \overline{B})(A + \overline{B + C})(\overline{\overline{B} + D})$$

รูปที่ 3.16 การเขียนวงจรตามตัวอย่างที่ 3.7

ตัวอย่าง 3.10 การเขียนวงจรจากสมการ

$$N = \overline{\overline{A\overline{B}}\;\overline{B\overline{C}D}\;\overline{AD}\;\overline{\overline{B}\,\overline{D}}}$$

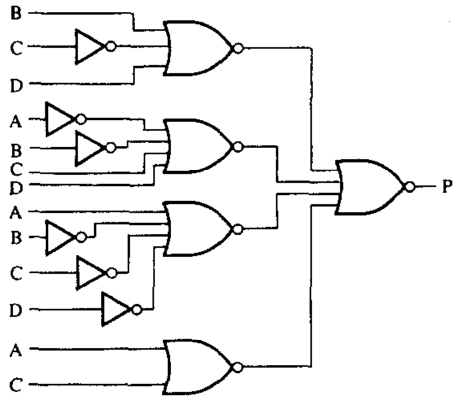


$$N = \overline{\overline{A\overline{B}}\;\overline{B\overline{C}D}\;\overline{AD}\;\overline{\overline{B}\,\overline{D}}}$$

รูปที่ 3.17 การเขียนวงจรตามตัวอย่างที่ 3.8

ตัวอย่าง 3.11 **การเขียนวงจรจากสมการ**

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

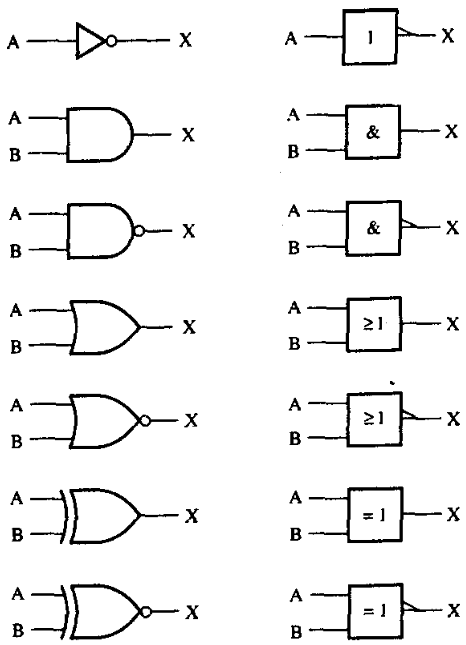


$$P = \overline{\overline{\overline{(B + \overline{C} + D)} + \overline{(\overline{A} + \overline{B} + C + D)} + \overline{(A + \overline{B} + \overline{C} + \overline{D})} + \overline{(A + C)}}}$$

รูปที่ 3.18 การเขียนวงจรตามตัวอย่างที่ 3.9

3.10 มาตรฐานของสัญลักษณ์ลอจิก IEEE / ANSI LOGIC GATE SYMBOLS

ในปี 1973 สัญลักษณ์ของลอจิกใหม่ได้แนะนำเกี่ยวกับอุตสาหกรรมดิจิตอลอิเลคโทรนิคในสหรัฐอเมริกา ผ่านคณะกรรมการด้านเทคนิคของ IEEE (Institute of Electrical and Electronics Engineers) มาตราฐานสุดท้ายเป็นที่ยอมรับของนานาชาติในปี 1984 ที่อ้างถึงมาตราฐาน IEEE 91-1984



รูปที่ 3.19 Traditional and IEEE/ANSI logic Gate Symbols

3.11 BOOLEAN ALGEBRA

บูลีนอัลจีบา คือพีชคณิตบูลีนที่ใช้กำหนดการทำงานของวงจรลอจิก ใช้ประโยชน์ในการลดรูปวงจรลอจิกแต่คุณสมบัติการทำงานยังเหมือนเดิม ท่านจะต้องทำความเข้าใจพื้นฐานของพีชคณิตบูลีน ซึ่งจะนำไปประยุกต์ใช้กับวงจรดิจิตอลอีเลคโทรนิค

ทฤษฎีพื้นฐานของพีชคณิตบูลีน

กฎพื้นฐานของพีชคณิตบูลีน คือสมการที่เป็นกฎเกณฑ์ทางคณิตศาสตร์ จะแตกต่างกันที่สัญลักษณ์ตัวทำงาน (Operator) หมายความว่าลอจิก AND จะใช้ (dot) ขั้นกลางระหว่างตัวแปรของสมการ เช่น A and B and C จะเขียนเป็น A.B.C หรือเขียนเป็น ABC ตัวแปรของ AND เรารู้จักกันในนามผลคูณ (Product term) เช่น AB , A B C , WXYZ

หน้าที่ของลอจิก OR จะแทนด้วยเครื่องหมายบวก (+) เช่น A or B เขียนเป็น A + B การทำงานของ OR จะอยู่ในรูปของผลบวก (SUM terms) เช่น A + B , D + E + F , and (W + X + Y + Z) ที่จัดอยู่ในรูป Sum terms

สัญลักษณ์สุดท้ายคือ คอมพลีเม้นต์ (Complement) สัญลักษณ์เป็น - (Bar) ที่กำหนดไว้เหนือตัวแปร เช่นค่าคอมพลีเม้นต์ของ A จะเป็น $\overline{A}$ และค่าคอมพลีเม้นต์ของ (A + C) จะเขียนเป็น $(\overline{A + C})$ เครื่องหมาย bar ในสัญลักษณ์ลอจิกเกตจะแทนด้วย วงกลม อยู่ทางด้านอินพุตหรือเอ้าพุต

วงเล็บที่ใช้พีชคณิตบูลีน ใช้กำหนดกลุ่มของตัวแปร มาตราฐานของ อัลจีบา คือกฎ Commutative Law สำหรับพีชคณิตบูลีน Commutayive law สำหรับตัวแปร 2 ตัว มีการทำงานดังต่อไปนี้

Eq. 3-1	$AB = BA$	AND operation
Eq. 3-2	$\overline{AB} = \overline{BA}$	NAND operation
Eq. 3-3	$A + B = B + A$	OR operation
Eq. 3-4	$\overline{A + B} = \overline{B + A}$	NOR operation
Eq. 3-5	$A \oplus B = B \oplus A$	XOR operation
Eq. 3-6	$\overline{A \oplus B} = \overline{B \oplus A}$	XNOR operation

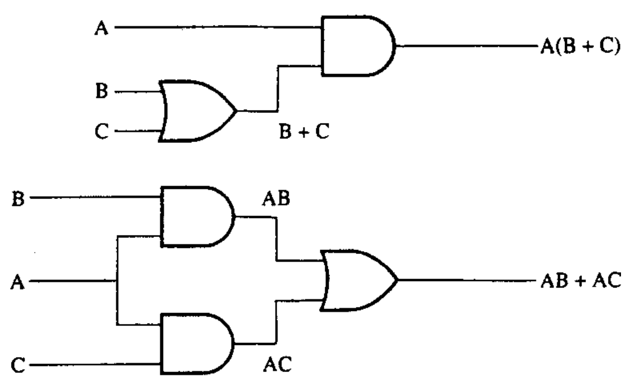
Associative Law เป็นการประยุกต์การทำงานของพีชคณิตบูลีน สำหรับกฎนี้ แสดงการทำงานของ 3 ตัวแปร

Eq. 3-7	$A(BC) = (AB)C$	AND
Eq. 3-8	$\overline{A(BC)} = (AB)C$	NAND
Eq. 3-9	$(A + B) + C = A + (B + C)$	OR
Eq. 3-10	$\overline{(A + B) + C = A + (B + C)}$	NOR

Distributive Law เป็นการประยุกต์การทำงานของ 3 ตัวแปร โดยใช้กฎเกณฑ์ของ AND/OR แสดงดังต่อไปนี้

Eq. 3-11	$A(B + C) = AB + AC$
Eq. 3-12	$(A + B)(C + D) = AC + AD + BC + BD$

Commutative , Associative , Distributive laws สามารถแทนด้วยโครงสร้างของวงจรลอจิก ดังแสดงในรูป วงจรเกตที่เทียบเท่าสมการ



รูปที่ 3.20. Distributive Law Equivalent Circuits

ตัวอย่าง 3.12 Truth Table Verification of the Distributive Law

ปัญหา จงพิสูจน์พีชคณิตบูลีนด้วยกฎของ Distributive Law โดยใช้ตารางแสดงคุณลักษณะจากสมการ $A(B + C) = AB + AC$

การแก้ปัญหา สมการมี 3 ตัวแปรเราจะต้องสร้างสมการอินพุต 3 ตัวแปร คือ A , B , C ซึ่งอินพุต 3 ตัวแปรสามารถกำหนดสภาวะการทำงานได้ 8 สภาวะ คืออยู่ในช่วง 000 ถึง

111 ให้หาเอ้าพุตของค่า A(B + C) ถ้ากฎ Distributive ถูกต้อง ค่าของเอ้าพุตทั้ง 2 จะมีค่าเท่ากัน ดูผลลัพธ์ที่ได้จากตาราง

ตาราง 3.6 Truth Table to Verify the Distributive Law

Input				output # 1			output # 2
A	B	C	B + C	A(B + C)	AB	AC	AB+AC
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

3.12 Boolean Indentities and Theorems

พื้นฐานของกฎ Commutative , Associative , and Distributive ทฤษฎีทางพีชคณิตบูลีนได้นำไปใช้ประโยชน์ในการลดรูปสมการของพีชคณิตบูลีน และวงจรลอจิก ดังตารางที่กำหนดความสัมพันธ์ ดังต่อไปนี้

ตาราง 3.7 Boolean Identifies , Thearms and Rules

Eq. 3-13	$\overline{\overline{A}} = A$
Eq. 3-14	$A + \underline{A} = A$
Eq. 3-15	$A + \overline{A} = 1$
Eq. 3-16	$A + O = A$
Eq. 3-17	$A + 1 = 1$
Eq. 3-18	$A + \underline{A}B = A$
Eq. 3-19	$A + \overline{A}B = A + B$
Eq. 3-20	$A(\underline{A} + B) = A$
Eq. 3-21	$A(\overline{A} + B) = AB$
Eq. 3-22	$A \cdot \underline{A} = A$
Eq. 3-23	$A \cdot \overline{A} = 0$
Eq. 3-24	$A.0 = 0$
Eq. 3-25	$A \cdot 1 = A$
Eq. 3-26	$(A + B)(A + \overline{B}) = A$
Eq. 3-27	$AB + A\overline{B} = A$
Eq. 3-28	$(A + B)(A + C) = A + BC$
Eq. 3-29	$A \oplus B = A\overline{B} + \overline{A}B$
Eq. 3-30	$\overline{A \oplus B} = AB + AB$

ตัวอย่าง 3.13 ทฤษฎี A.0 = 0

ปัญหา ให้แสดงผลทฤษฎีบูลีนที่กำหนด

การแก้ปัญหา วิเคราะห์หน้าที่ของทฤษฎีด้วยตารางแสดงคุณลักษณะ

การวิเคราะห์จากตาราง สังเกตว่าเอ้าพุตจะเป็นค่า 0 เสมอ ฉนั้นทฤษฎีถูกตรวจสอบแล้ว

A	0	A.0
0	0	0
1	0	0

ตัวอย่าง 3.14. ทฤษฎี A + AB = A

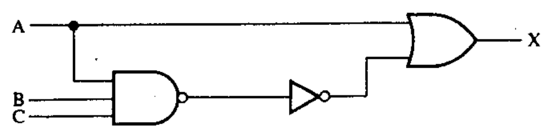
การวิเคราะห์จากตาราง สังเกตุว่าคอลัมน์ A เท่ากับ A + A B ฉะนั้นทฤษฎีถูกตรวจสอบแล้ว

A	B	AB	A + AB
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

ตัวอย่าง 3.15. จงพิสูจน์การใช้ทฤษฎีพีชคณิตบูลีนจากวงจร

ปัญหา จงลดรูปวงจรลอจิกโดยใช้พีชคณิตบูลีน

การแก้ปัญหา ขั้นที่ 1 จากวงจร จะเห็นว่ามี NOT Gate อยู่เอ้าพุต NAND Gate ซึ่ง NOT Gate เป็นอินพุตของ OR Gate ดังที่แสดงวงจรในรูป 3.13



(a) Original circuit

รูปที่ 3.21 จากตัวอย่าง 3.13 .เป็นรูปที่กำหนดเดิม

(b) First simplification step

หลังจากลดรูปแล้วจะได้ ดังรูปที่ 3.22 เป็นการลดรูปขั้นที่ 1

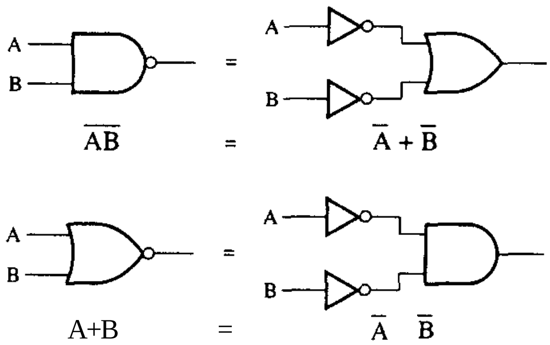
สมการบูลีนที่ได้จากการลดรูปของวงจรในรูปที่ 3.23 คือ A B C + A โดยใช้ทฤษฎี A + A B C = A ฉะนั้นจึงไม่มีวงจรลอจิกเกตที่ให้เอ้าพุต X ซึ่งมีค่าเท่ากับอินพุต A ซึ่งเป็นการลดรูปครั้งสุดท้ายจะได้วงจรดังนี้

รูป 3.23 การลดรูปสุดท้าย

3.13 DeMorgan's Theorems

DeMorgan's Theorems ที่ค้นคิดโดยนักคณิตศาสตร์และนักตรรกวิทยาชาวอังกฤษ ชื่อ Augustus DeMorgan (1806 - 1871) ต่อมาภายหลัง George Boole บุคคลทั้งสองเป็นผู้มีความสำคัญ ในการคิดทฤษฎีพีชคณิตบูลีน ซึ่งทฤษฎีของ DeMorgan สามารถลดรูปได้ทั้ง AND และ OR ดังสมการต่อไปนี้

Eq. 3-31	$\overline{A + B} = \bar{A}\,\bar{B}$	DeMorgan's Theorem 1
Eq. 3-32	$\overline{A\,B} = \bar{A} + \bar{B}$	DeMorgan's Theorem 2



รูปที่ 3.24 การเขียนวงจรลอจิกของทฤษฎี DeMorgan

ทฤษฎีของ DeMorgan สามารถใช้กับตัวแปร 2 ตัวหรือมากกว่า ดังตัวอย่างของสมการ 4 ตัวแปร

Eq. 3-33	$\overline{A + B + C + D} = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$	DeMorgan Theorem 1
Eq. 3-34	$\overline{A\,B\,C\,D} = \bar{A} + \bar{B} + \bar{C} + \bar{D}$	DeMorgan Theorem 2

3.14 การลดรูปพีชคณิตบูลีน (BOOLEAN EQUATION SIMPLIFICATION)

ความสำคัญของสมการพีชคณิตบูลีน คือสามารถนำไปประยุกต์ใช้กับสมการของวงจรลอจิกสำหรับการลดรูป เทคนิคของการลดรูปจะช่วยให้ใช้ไอซีลดลงซึ่งเป็นโครงสร้างของวงจรลอจิกการลดรูปดังตัวอย่างต่อไปนี้

ตัวอย่าง 3.16. การลดรูปรูปโดยใช้พีชคณิตบูลีน

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

SIMPLIFICATION STEPS	EQUATION USED
$A + \overline{A}B + \overline{A}BC$	
$A + \overline{A}B(1 + C)$	Eq. 3-18
$A + \overline{A}B\,1$	Eq. 3-17
$A + \overline{A}B$	Eq. 3-25
$A + B$	Eq. 3-19

ตัวอย่าง 3.17 การลดรูปโดยใช้พีชคณิตบูลีน

Simplify $\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C}$

SIMPLIFICATION STEPS	EQUATION USED
$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C}$	
$\overline{A}\,\overline{B} + A\,\overline{B}\,\overline{C}$	Eq. 3-27
$\overline{B}(\overline{A} + A\,\overline{C})$	Factor out $\overline{B}$

SIMPLIFICATION STEPS	EQUATION USED
$\overline{B}(\overline{A} + \overline{\overline{A}}\,\overline{C})$	Eq. 3-13
$\overline{B}(\overline{A} + \overline{C})$	Eq. 3-19
$\overline{A}\,\overline{B} + \overline{B}\,\overline{C}$	Eq. 3-11

ตัวอย่าง 3.18 **การลดรูปโดยใช้พีชคณิตบูลีน**

Simplify $ABCD + \overline{ABCD}$

SIMPLIFICATION STEPS	EQUATION USED
$ABCD + \overline{ABCD}$	
1	Eq. 3-15

ตัวอย่าง 3.19 การลดรูปโดยใช้พีชคณิตบูลีน

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

SIMPLIFICATION STEPS	EQUATIONS USED
$(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)(A + \overline{B} + \overline{C})$	
$(\overline{A} + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + \overline{C})$	Eq. 3-22
$[(\overline{A} + \overline{B} + \overline{C})(\overline{A} + \overline{B} + C)][(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + \overline{C})]$	
$[\overline{A} + \overline{B}][\overline{B} + \overline{C}]$	Eq. 3-26
$[\overline{B} + \overline{A}][\overline{B} + \overline{C}]$	Eq. 3-3
$\overline{B} + \overline{A}\,\overline{C}$	Eq. 3-28 ■

ตัวอย่าง 3.20 การลดรูปโดยใช้พีชคณิตบูลีน

Simplify $\overline{AB(CD + EF)}$

SIMPLIFICATION STEPS	EQUATION USED
$\overline{AB(CD + EF)}$	
$\overline{AB} + \overline{(CD + EF)}$	Eq. 3-23
$x + \overline{B} + (\overline{C}D)(E\overline{F})$	Eq. 3-32
	Eq. 3-31
$\overline{A} + \overline{B} + (\overline{C} + \overline{D})(\overline{E} + \overline{F})$	Eq. 3-32
$A + B + \overline{C}\,\overline{D} + \overline{C}\,\overline{F} + \overline{D}\,\overline{F}$	Eq. 3-12

ตัวอย่าง 3.21 การลดรูปโดยใช้พีชคณิตบูลีน

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

SIMPLIFICATION STEPS	EQUATION USED
$\overline{(A + B + C + D)} + (A\,\overline{B}\,\overline{C}\,D)$	
$\overline{\overline{A}}\,\overline{B}\,\overline{C}\,\overline{D} + A\,\overline{B}\,\overline{C}\,D$	Eq. 3-31
$A\,\overline{B}\,C\,D + A\,B\,\overline{C}\,D$	Eq. 3-13
$A\,\overline{B}\,\overline{C}$	Eq. 3-27

ตัวอย่าง 3.22 การลดรูปโดยใช้พีชคณิตบูลีน

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

SIMPLIFICATION STEPS	EQUATION USED
$(\bar{A} + \bar{B})(A + \bar{B})(\bar{A} + B)$	
$[(\bar{A} + \bar{B})(A + \bar{B})][(\bar{A} + B)(\bar{A} + \bar{B})]$	Eq. 3-22
$[\bar{B}][\bar{A}]$	Eq. 3-26
$\bar{A}\ \bar{B}$	Eq. 3-1

ตัวอย่าง 3.33 การลดรูปโดยใช้พีชคณิตบูลีน

Negate $(X \cdot \bar{Y} + \bar{X} + Y)$.

Solution $\overline{(X \cdot \bar{Y} + \bar{X} + Y)} = (\overline{X \cdot \bar{Y}}) \cdot \bar{\bar{X}} \cdot \bar{Y}$

$$= (\bar{X} + Y) \cdot X + \bar{Y}$$
$$= \bar{X} \cdot X \cdot \bar{Y} + Y \cdot X \cdot \bar{Y}$$
$$= (X \cdot \bar{X}) \cdot \bar{Y} + X \cdot (Y \cdot \bar{Y})$$
$$= 0 \cdot \bar{Y} + X \cdot 0$$
$$= 0 + 0$$
$$= 0$$

Negate $\bar{A} \cdot B + A \cdot \bar{B}$.

Solution $\overline{(\bar{A} \cdot B + A \cdot \bar{B})} = (\overline{\bar{A} \cdot B}) \cdot (\overline{A \cdot \bar{B}})$

$$= (A + \bar{B}) \cdot (\bar{A} + B)$$
$$= A \cdot \bar{A} + A \cdot B + \bar{B} \cdot \bar{A} + \bar{B} \cdot B$$
$$= 0 + A \cdot B + \bar{A} \cdot \bar{B} + 0$$
$$= A \cdot B + \bar{A} \cdot \bar{B}$$

(a) $D = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$

(b) W = –Y(X + Z) + Z(–X + Y) + XZ

(a) $D = \bar{A}BC + A\bar{B}C + ABC + B\bar{C}$

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

(b) $W = \bar{Y}(X + Z) + Z(\bar{X} + Y) + XZ$

$$= X\bar{Y} + \bar{Y}Z + \bar{X}Z + YZ + XZ$$
$$= X\bar{Y} + Z(\bar{Y} + \bar{X} + Y + X) = X\bar{Y} + Z(1)$$
$$= X\bar{Y} + Z$$

ตัวอย่างที่ 3.23 การลดรูปโดยใช้พีชคณิตบูลีน

Simplify the following logic equations using Boolean algebra:

(a) $H = \bar{X}\bar{Y} + Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z + X(Y + \bar{X}) + X$

(b) $P = \bar{A}B\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$

(c) $X = R(P + \bar{P}Q + \bar{Q})(\bar{Q} + \bar{R}P)$

(d) $Z = (A\bar{B}\bar{C} + A\bar{B}C + ABC + AB\bar{C})(A + B)$

(e) $W = \overline{AC + A\bar{D}} + \overline{\bar{B}C}$

(f) $A = P + PQR + Q + R$

Solutions

(a)

$H = \bar{X}\bar{Y} + Y\bar{Z} + X\bar{Y}\bar{Z} + X\bar{Y}Z + X(Y + \bar{X}) + X$

$H = \bar{X}\bar{Y} + Y\bar{Z} + X\bar{Y}(\bar{Z} + Z) + X(Y + \bar{X}) + X$ (distributive)

$H = \bar{X}\bar{Y} + Y\bar{Z} + X\bar{Y}1 + X(Y + \bar{X}) + X$ (complementarity)

$H = \bar{X}\bar{Y} + Y\bar{Z} + X\bar{Y} + X(Y + \bar{X}) + X$ (identity)

$H = Y\bar{Z} + \bar{X}\bar{Y} + X\bar{Y} + X(Y + \bar{X}) + X$ (commutative)

$H = Y\bar{Z} + \bar{Y}(\bar{X} + X) + X(Y + \bar{X}) + X$ (distributive)

$H = Y\bar{Z} + \bar{Y}1 + X(Y + \bar{X}) + X$ (complementarity)

$H = Y\bar{Z} + \bar{Y} + X(Y + \bar{X}) + X$ (identity)

$H = \bar{Z} + \bar{Y} + X(Y + \bar{X}) + X$ (miscellaneous)

$H = \bar{Z} + \bar{Y} + XY + X\bar{X} + X$ (distributive)

$H = \bar{Z} + \bar{Y} + XY + 0 + X$ (complementarity)

$H = \bar{Z} + \bar{Y} + XY + X$ (Identity)

$H = \bar{Z} + \bar{Y} + X$ (absorbtion)

(b)

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

$P = \bar{A}\bar{C}(\bar{B} + B) + \bar{A}BC + A\bar{B}(\bar{C} + C) + AB\bar{C}$ (distributive)

$P = \bar{A}\bar{C} + \bar{A}BC + A\bar{B} + AB\bar{C}$ (complementarity / identity)

$P = \bar{A}(\bar{C} + BC) + A(\bar{B} + B\bar{C})$ (distributive)

$P = \bar{A}(\bar{C} + B) + A(\bar{B} + \bar{C})$ (miscellaneous)

$P = \bar{A}\bar{C} + \bar{A}B + A\bar{B} + A\bar{C}$ (distributive)

$P = \bar{A}\bar{C} + A\bar{C} + A\bar{B} + \bar{A}B$ (associative)

$P = \bar{C}(\bar{A} + A) + A\bar{B} + \bar{A}B$ (distributive)

$P = \bar{C} + A\bar{B} + \bar{A}B$ (complementarity / identity)

(c)

$X = R(P + \bar{P}Q + \bar{Q})(\bar{Q} + \bar{R}P)$

$X = R(P + Q + \bar{Q})(\bar{Q} + \bar{R}P)$ (miscellaneous)

$X = R(P + 1)(\bar{Q} + \bar{R}P)$ (complementarity)

$X = R1(\bar{Q} + \bar{R}P)$ (dominance)

$X = R(\bar{Q} + \bar{R}P)$ (identity)

$X = R\bar{Q} + R\bar{R}P$ (distributive)

$X = R\bar{Q} + 0P$ (complementarity)

$X = R\bar{Q} + 0$ (dominance)

$X = R\bar{Q}$ (identity)

(d)

$Z = (ABC + A\bar{B}C + ABC + ABC)(A + B)$

$Z = (A\bar{B}(\bar{C} + C) + AB(C + \bar{C}))(A + B)$ (distributive)

$Z = (A\bar{B} + AB)(A + B)$ (complementarity / identity)

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

$Z = A(A + B)$ (complementarity / identity)

$Z = A$ (absorbtion)

3.15 Sum Of Product Equations and Circuits

กฎเกณฑ์ของ SOP คือสมการของวงจรลอจิก ที่ทำงานตามกำหนดของตารางแสดงคุณลักษณะ ที่กำหนดค่าของอินพุต และค่าของเอ้าพุตในระดับลอจิก 1

การตรวจสอบ SOP จากกฎเกณฑ์ในสมการที่ 3-35 ค่าที่เราเห็นว่ามีเอ้าพุตเป็น 1 เมื่อทุกตัวแปรในเทอมของ Products มีค่าเป็น 1 เพราะฉะนั้น กฎเกณฑ์ของ SOP สามารถ SUM ค่าของอินพุตของทุกเทอมรวมกันที่จะให้ผลลัพธ์เป็น 1

ถ้าไม่มีการพิสูจน์จากกฎของ SOP นั่นหมายถึงข้อมูลที่เป็นไปได้ของอินพุตทั้งหมดรวมกันจะให้ผลลัพธ์เป็น 1 พิจารณาจากการขยายจากกฎ SOP จำนวนเทอมของสมการจะเท่ากับจำนวน Occurrence ที่ให้ค่าเอ้าพุตเป็น 1 ในตารางแสดงคุณลักษณะ หลายกรณีสมการสามารถพิสูจน์ที่จะลดจำนวนเทอมของตัวแปรในสมการ SOP

ตารางที่ **3-7** แสดงกฎของสมการ SOP เทียบกับตารางแสดงคุณลักษณะของลอจิกฟังชั่น

การสร้างวงจรของสมการ SOP ประกอบด้วย 2 ระดับ จะไม่นับตัวอินพุต Inverter สำหรับสมการ SOP ค่าของลอจิกเกต 2 ระดับ สามารถเขียนแทนด้วยวงจร AND=OR หรือวงจร NAND-NAND โดยการประยุกต์ใช้ทฤษฎี De-Morgan ลดรูปจากสมการ SOP เดิม

ตาราง **3-7** แสดงกฎสมการของ SOP จากตารางแสดงคุณลักษณะ

1. เมื่อตัวแปรมีค่าเป็น 1 เขียนตัวแปรในสมการเป็น True form
2. เมื่อตัวแปรมีค่าเป็น 0 เขียนตัวแปรในสมการเป็น Inverted form
3. รวมค่าของอินพุตทั้งหมดจะให้เอ้าพุตอยู่ในระดับที่ต้องการ จะไม่รวมอินพุตในเทอมอื่นๆของสมการ
4. เขียนแต่ละเทอมอย่างอิสระ อินพุตจะรวมกันในรูปของ Product
5. รวมเทอมทั้งหมดที่กำหนดค่าอินพุตเข้าด้วยกัน
6. จัดการลดรูปสมการใหม่ ถ้าต้องการ โดยใช้กฎพีชคณิตบูลีน
7. ตรวจสอบสมการโดยการแทนค่าอินพุตจากตารางแสดงคุณลักษณะเพื่อให้เกิดเอ้าพุตที่ถูกต้อง

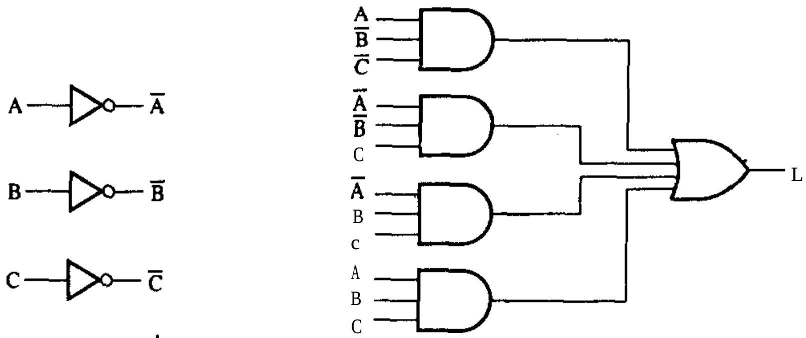
ตัวอย่าง 3-24 **Sum Of Products Expressions and Circuits**

ปัญหา จากกฎของ SOP สำหรับการกำหนดลอจิกฟังชั่นในตารางแสดงคุณลักษณะ ในรูปแบบวงจร AND-OR เพื่อกำหนดฟังชั่นการทำงาน

การแก้ปัญหา จากคุณลักษณะในตาราง 3-7 ของกฎ SOP จะได้สมการ

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$L = \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + A\,B\,C$$



รูป 3.26 AND - OR SUM of Product (SOP) Circuit

ตัวอย่าง 3.25 NAND - NAND SOP Circuit

ปัญหา สร้างวงจร NAND - NAND SOP Circuit จากสมการดังต่อไปนี้

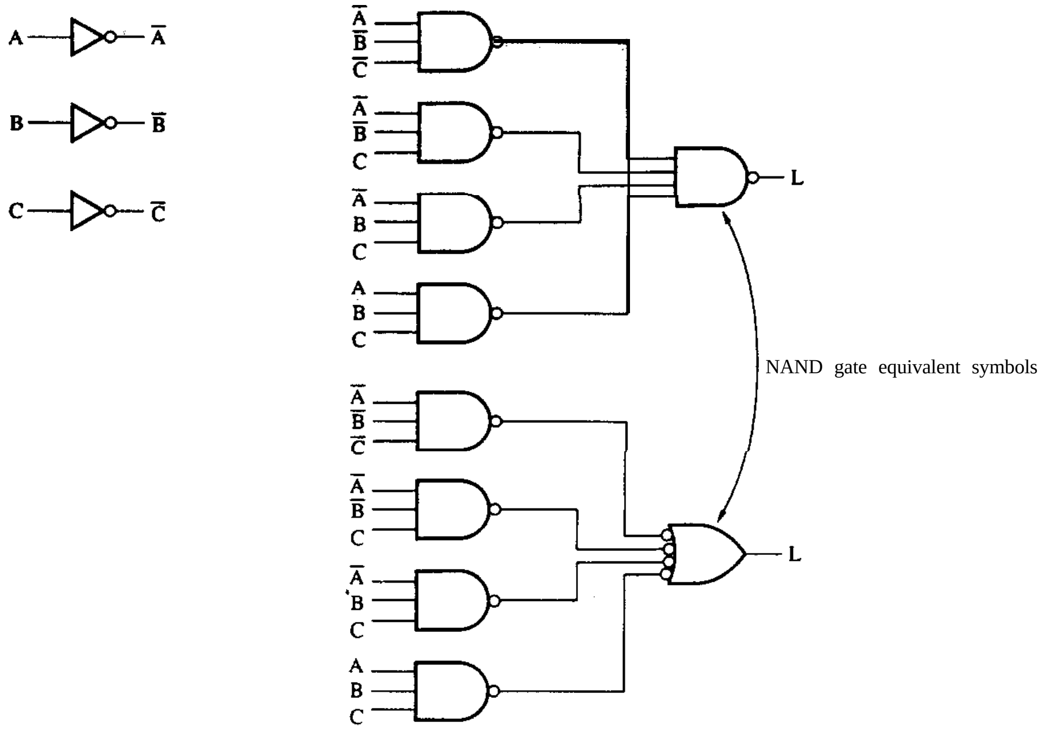
การแก้ปัญหา ประยุกต์ใช้ทฤษฎีของ DeMorgan ในการกลับค่าเป็นตรงกันข้ามของสมการ AND - OR Circuit ดังต่อไปนี้

1. กับค่าของ L เป็นตรงข้าม 2 ครั้ง
2. ประยุกต์ใช้ทฤษฎีของ DeMorgan โดยแยกเป็นสมการ OR
3. ออกแบบ NAND - NAND Circuit

$$L = \bar{A}\bar{B}\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + ABC$$

$$L = \overline{\overline{\bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + A\,B\,C}}$$

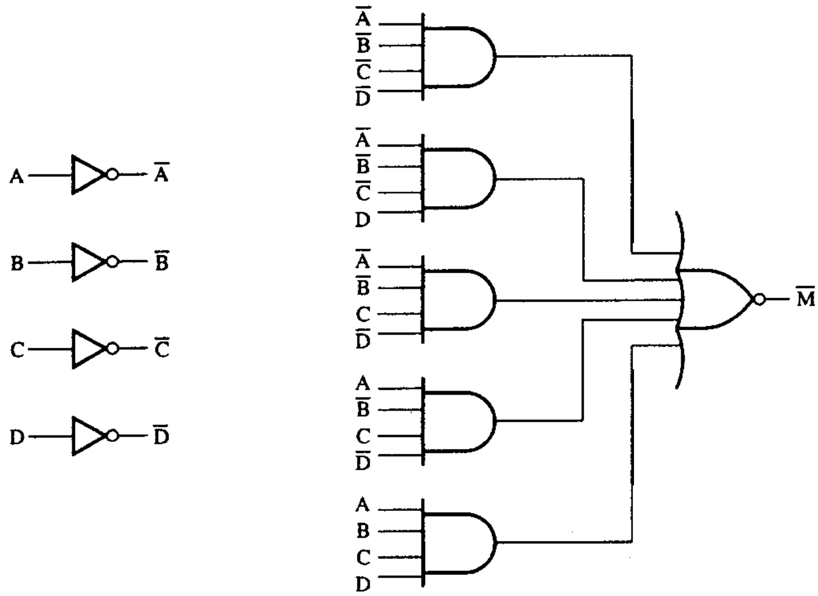
$$L = \overline{\overline{(\bar{A}\,\bar{B}\,\bar{C})}\,\overline{(\bar{A}\,\bar{B}\,C)}\,\overline{(\bar{A}\,B\,C)}\,\overline{(A\,B\,C)}}$$



ตัวอย่าง 3.26 Active LOW Sum of Products Expression Derived from a truth table

A	B	C	D	M
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

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

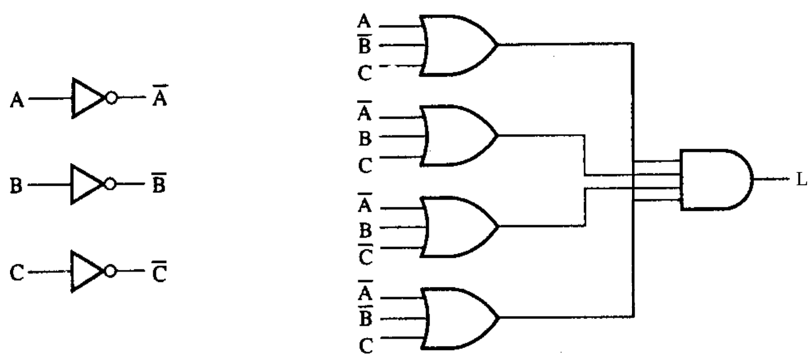


รูป 3.28 Active LOW SOP Circuit

ตัวอย่าง 3.21 Product of Sums Expressions and Circuits ออกแบบโดยใช้ OR AND Circuit

A	B	C	L
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

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



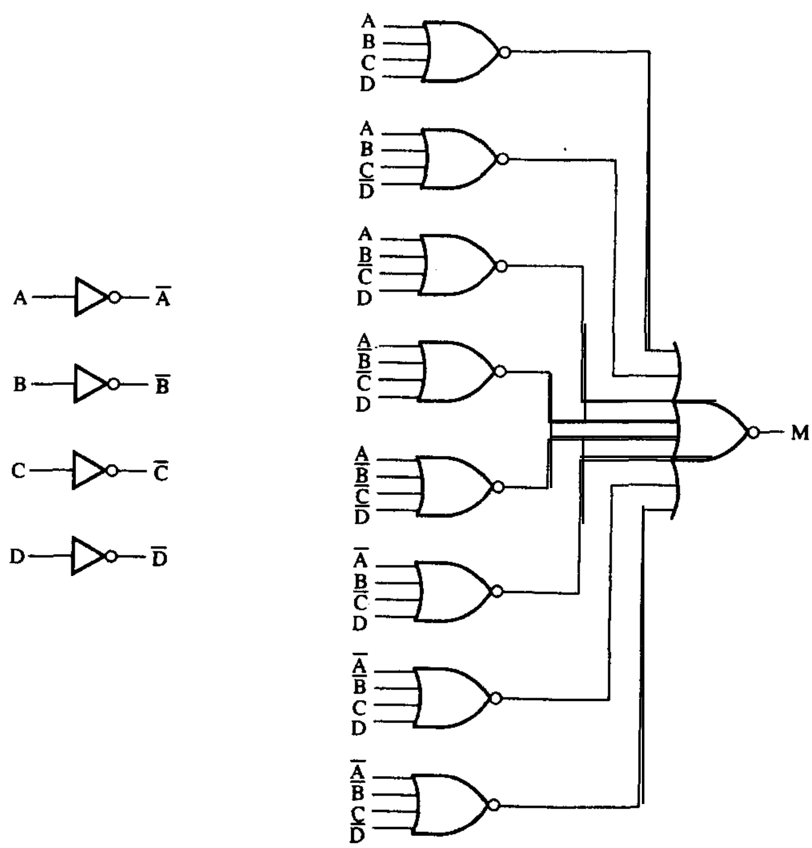
รูป 3.28 OR - AND (Product of Sum) POS Circuit

ตัวอย่าง 3.28 Product of Sums Expression and Circuits

A	B	C	D	M
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

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

■



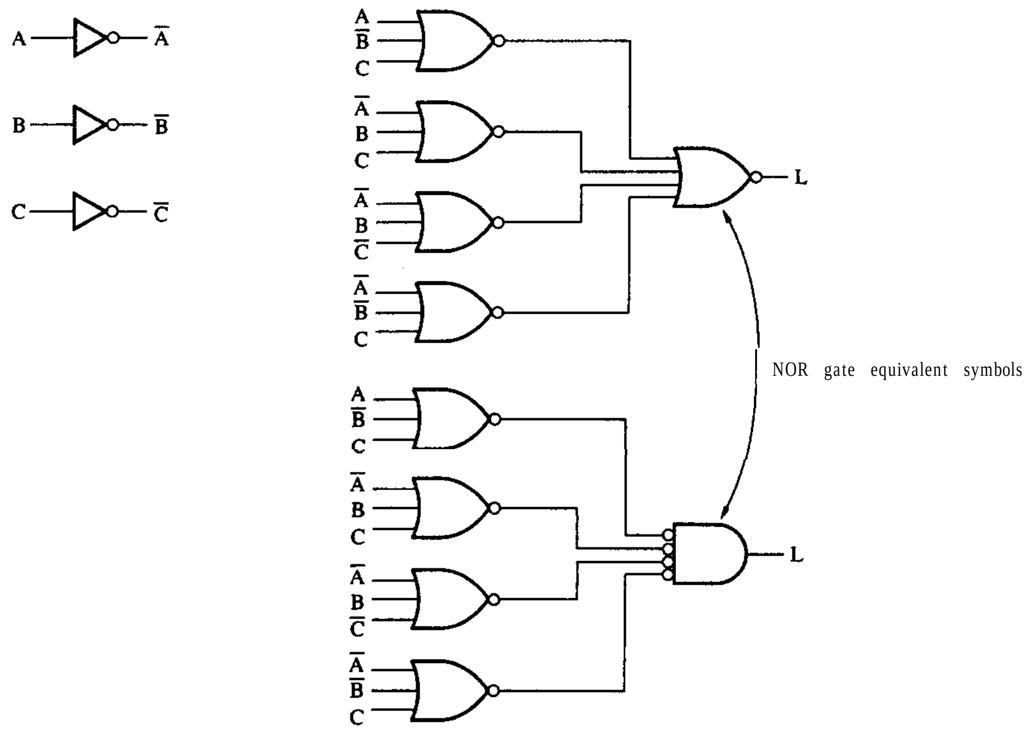
รูป 3.29 NOR - NOR POS Circuit

ตัวอย่าง 3.29 NOR - NOR Product of Sums Circuit

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

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

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



รูป 3.30 NOR-NOR POS Circuit

3.17 การเปรียบเทียบและการเปลี่ยนสมการ SOP และ POS

(Comparision and Conversion of sum of product and product of sum equations)

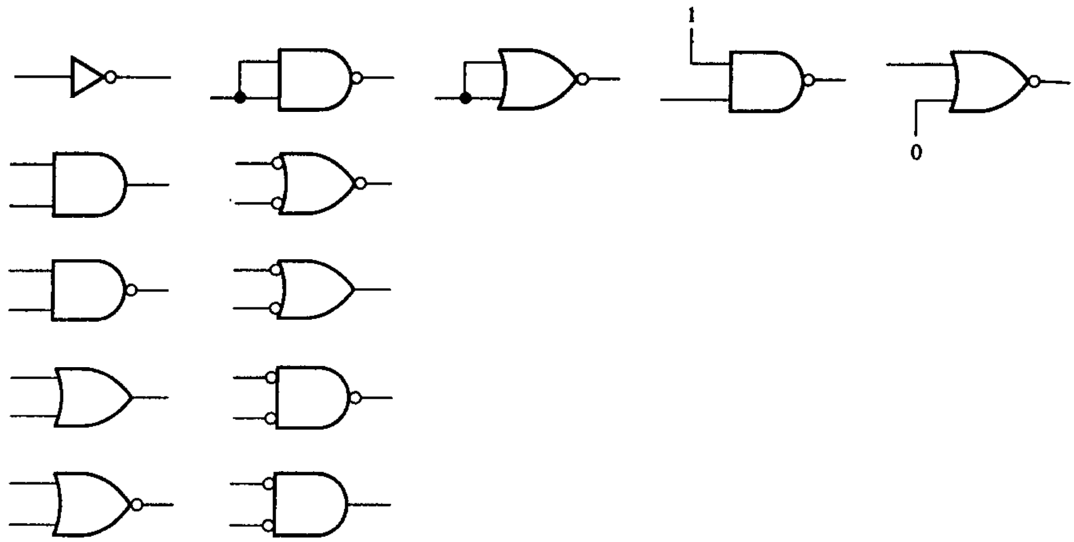
กฎเกณฑ์ของ POS และ SOP จากกฎเกณฑ์ของลอจิกและรูปแบบมาตราฐานในการออกแบบวงจร เราสามารถเปรียบกระบวนการในการใช้ลอจิกของ SOP และ POS ซึ่งทำให้การออกแบบวงจรลอจิกง่ายขึ้น ดังนี้

ตาราง วิธีการของ SOP และ POS

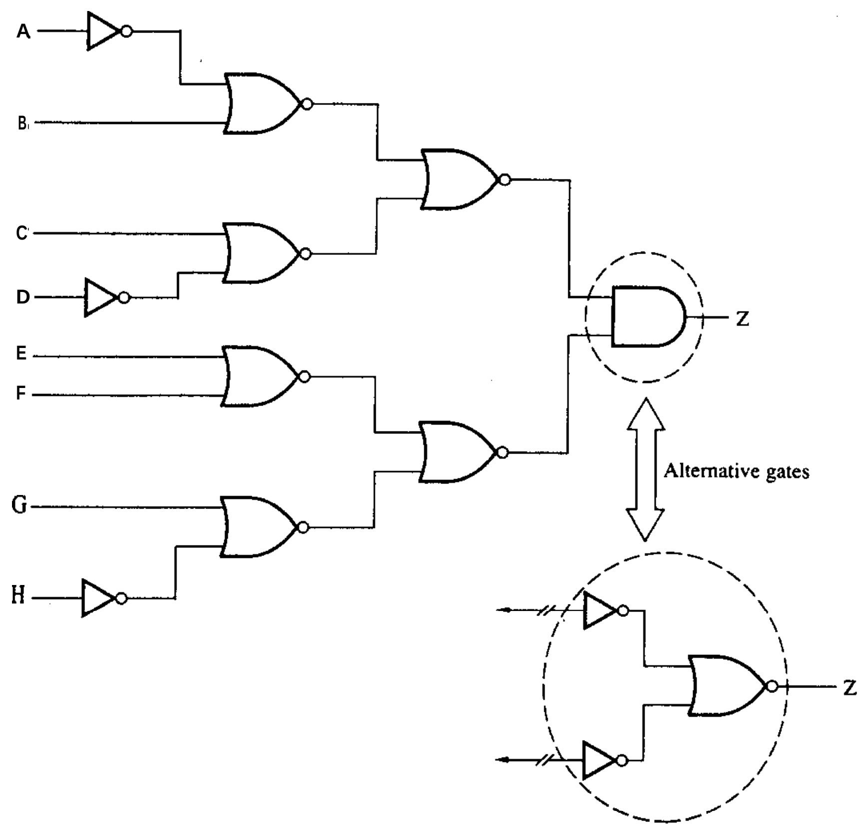
SOP Procedure	POS Procedure
1. Derive the SOP expression using only those **input combinations that result in an output of logic 1.**	1. Derive the POS expression using only those **input combinations that result in the output being logic 0.**
2. Input variables of **1** are written in their **true form.**	2. Input variables of 0 are written in their **true form.**
3. Input variables of 0 are written in their **inverted form.**	3. Input variables of 1 are written in their **inverted form.**
4. Input variables are **ANDed to** form the input terms.	4. Input variables are **ORed to** form the input terms.
5. Input terms are **ORed** together to give the final output result.	5. Input terms are **ANDed together to give the final** output **result.**

Active-LOW SOP
1. Derive the active-LOW SOP expression by using only those **input combinations that give an output of 0.**
2. Follow the same **SOP procedures.**
3. Write the **output variable** in its **inverted form.**

การเปรียบเทียบลอจิกที่เทียบเท่า

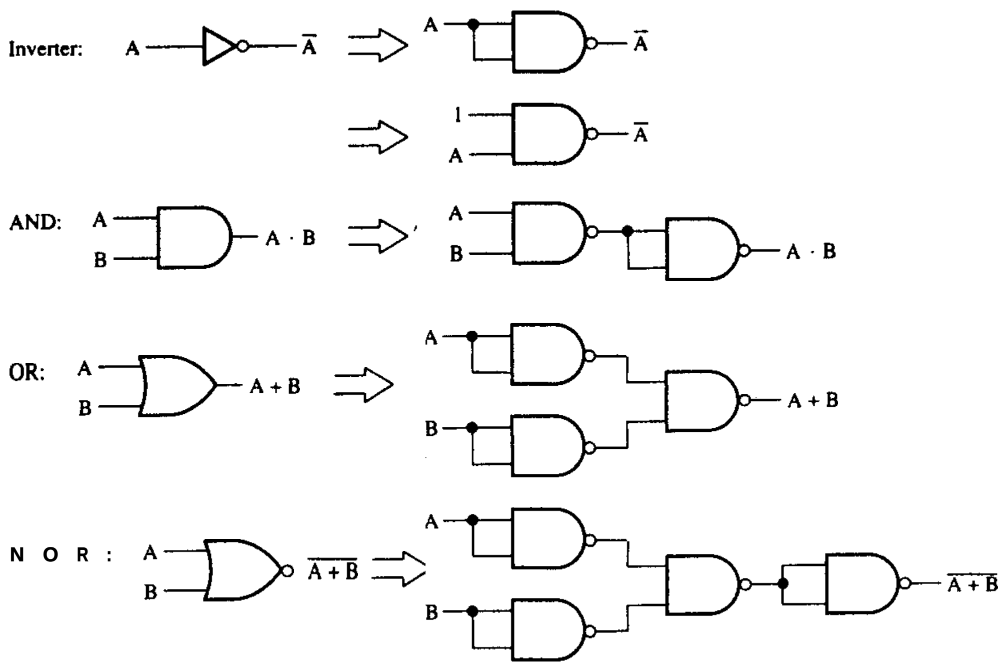


รูป 3.32 Equivalent logic gates symbols



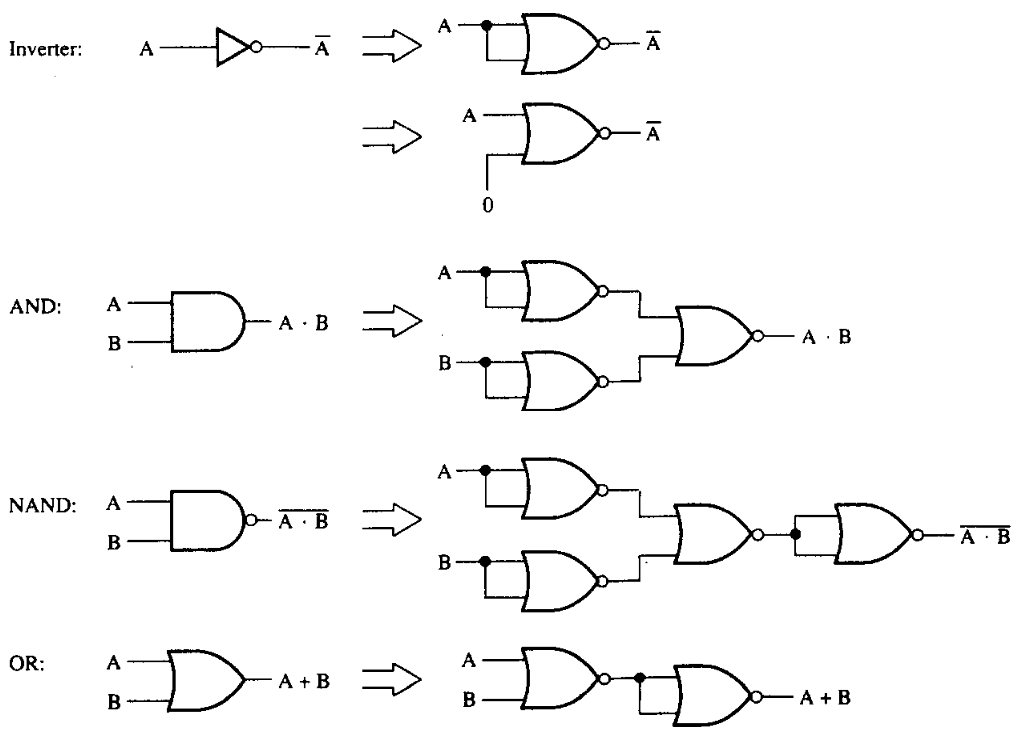
รูป 3.33 Equivalent Combination logic Circuits

NAND Equivalent Circuits



รูป 3.34 NAND equivalent Gates

NOR Equivalent Circuits



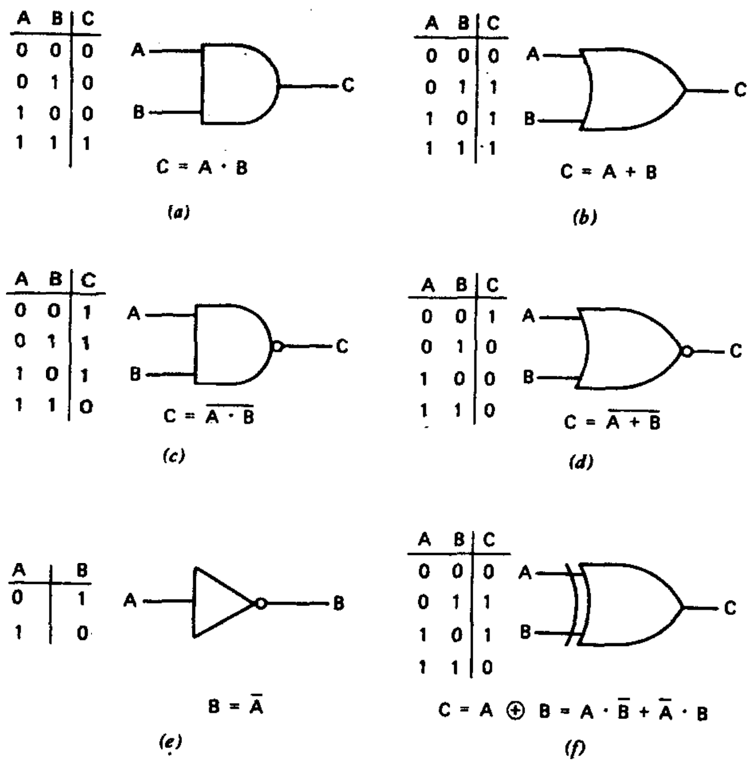
รูป 3.35 NOR Equivalent Gates

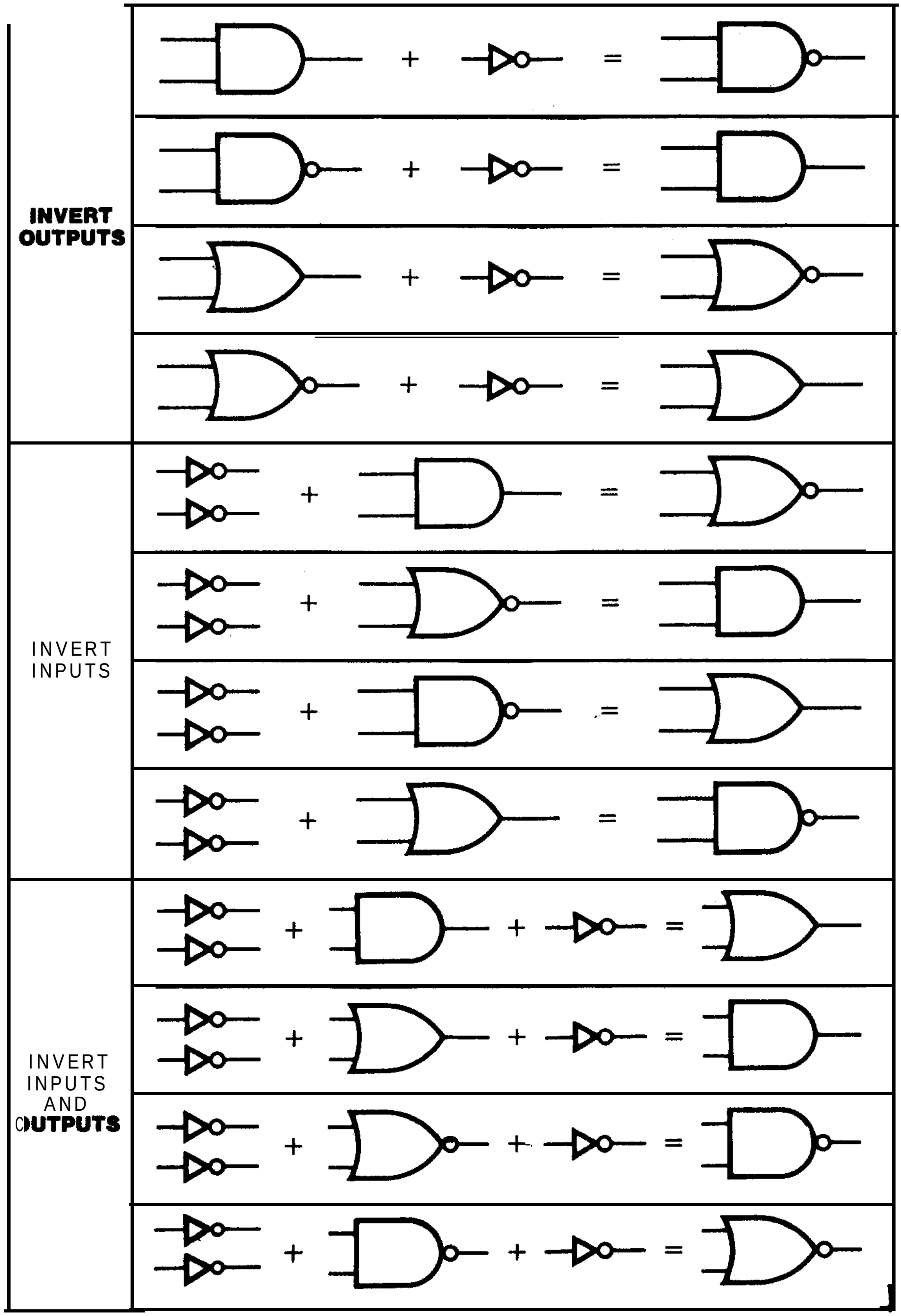
สรุป

วงจรลอจิกแต่ตัวมีคุณสมบัติในการทำงานแต่ละตัวที่แตกต่างกัน เช่น AND , OR , NOT , NOR , NAND , ECLUSIVE OR , EXCLUSIVE NOR เราสามารถนำลอจิกต่างๆเหล่านี้ มาสร้างเป็นวงจรคอมพิวเตอร์เพื่อให้ทำงานตามที่เราออกแบบ

การรวมฟังชั่นของลอจิกแต่ละตัว จะทำให้เกิดวงจรดิจิตอลลอจิก ที่ทำงานตามการออกแบบจากผลที่เกิดขึ้นของอินพุตและเอ้าพุต

ศึกษาถึงการออกแบบที่เรียกว่าวงจรคอมไบเนชั่น สามารถสร้างวงจรที่เหมาะสม โดยใช้หลักการลดรูปที่เรียกว่า พีชคณิตบูลลีน ตามกฎเกณฑ์ของพีชคณิตบูลลีน



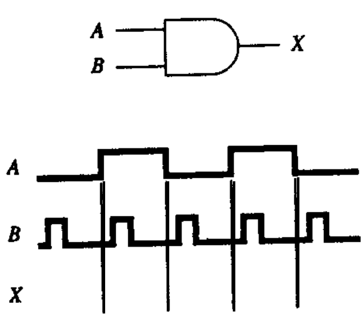

INVERT OUTPUTS
INVERT INPUTS
INVERT INPUTS AND OUTPUTS

แบบฝึกหัด

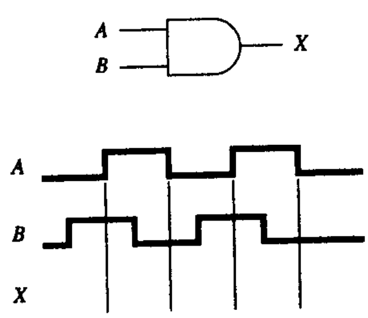
1. จงเขียนตารางในการแสดงคุณลักษณะของลอจิก AND ชนิด 3 อินพุต
2. จงเขียนตารางในการแสดงคุณลักษณะของลอจิก OR ชนิด 3 อินพุต
3. จงเขียนตารางในการแสดงคุณลักษณะของลอจิก XNOR ชนิด 2 อินพุต
4. จงเขียนไดอะแกรมเวลาของเอ้าพุต X จากวงจรลอจิกดังต่อไปนี้

   4.1

4.1 จงเขียนเอ้าพุต X จากลอจิก AND gateชนิด 2 อินพุต

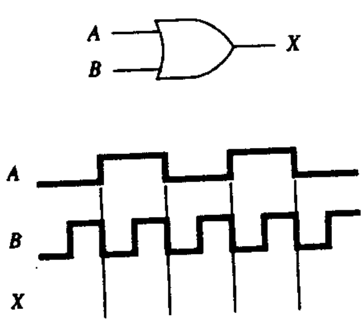


(a)

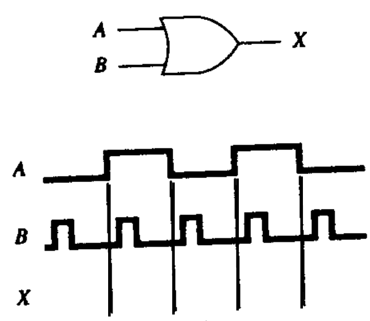


(b)

4.2 **จงเขียนเอ้าพุต** X **จากลอจิก** OR gate **ชนิด** 2 **อินพุต**

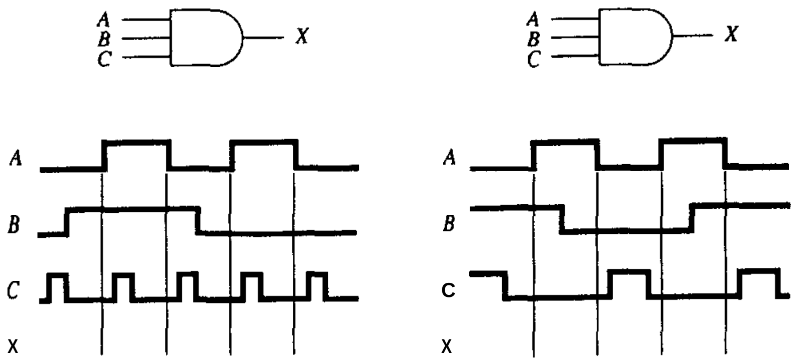


(a)



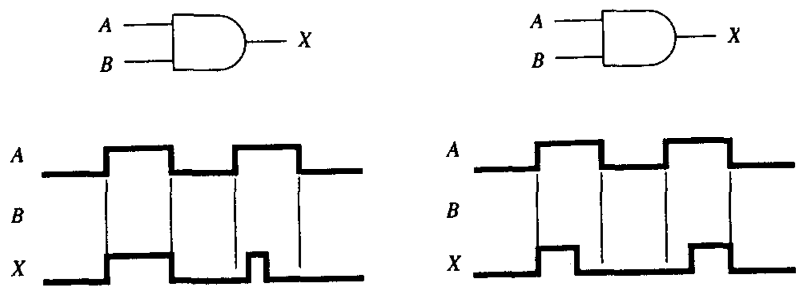
(b)

4.3 จงเขียนเอ้าพุต X จากลอจิก AND gate ชนิด 3 อินพุต



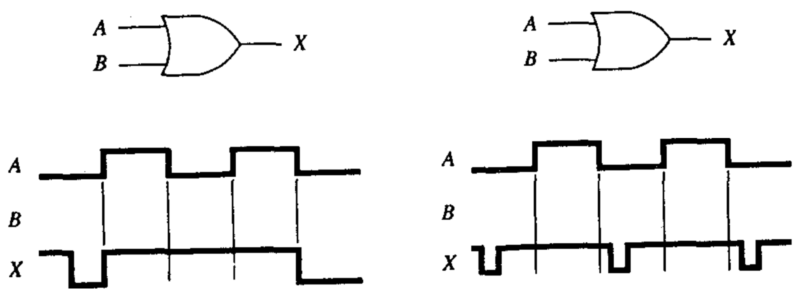
(a) (b)

4.4 จงเขียนเอ้าพุต X จากลอจิก AND gate ชนิด 2 อินพุต



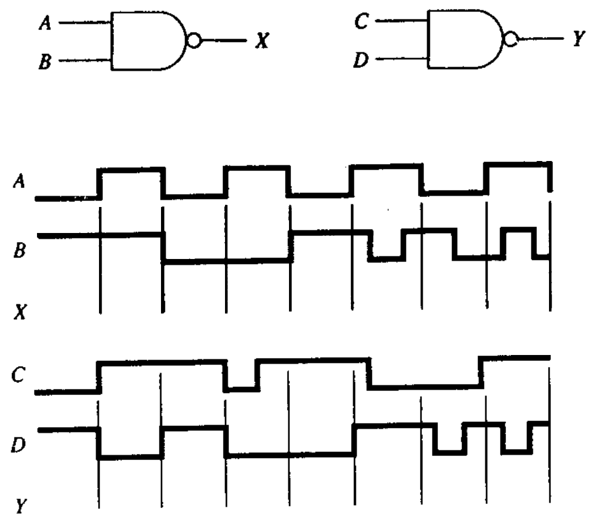
(a) (b)

4.5 จงเขียนเอ้าพุต X จากลอจิก OR gate ชนิด 2 อินพุต

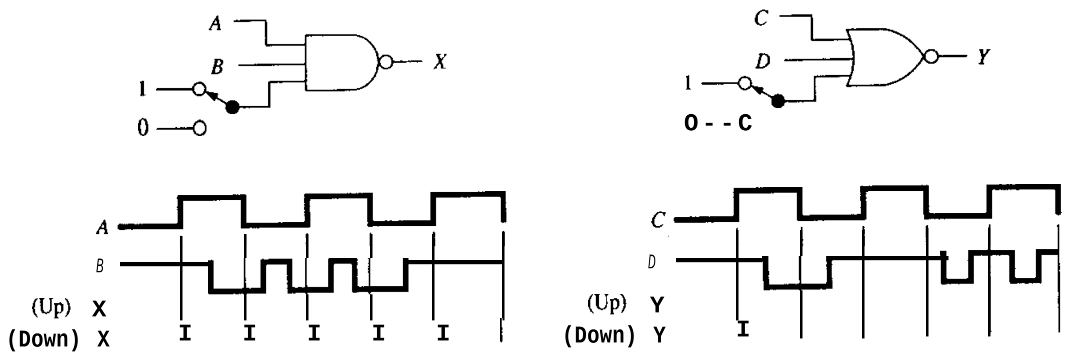


(a) (b)

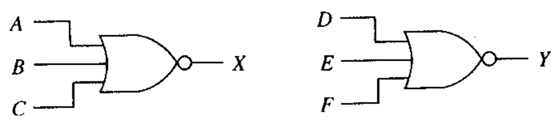
4.6 จงเขียนเอ้าพุต X จากลอจิก NAND gate ชนิด 2 อินพุต



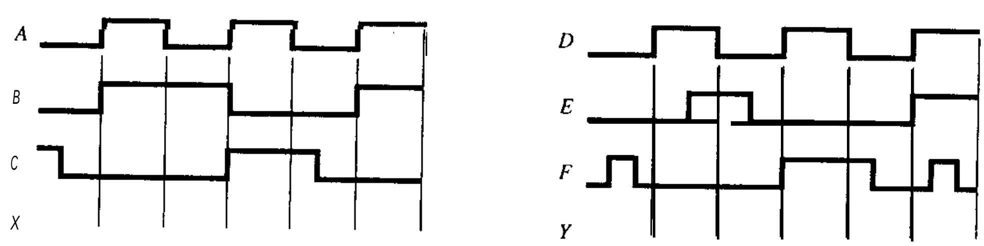
4.7 จงเขียนเอ้าพุต X จากลอจิก NAND gate ชนิด 3 อินพุต



4.8 จงเขียนเอ้าพุต X จากลอจิก NOR gate ชนิด 3 อินพุต



4 Q



4.9 **จงเขียนเอ้าพุต X และเอ้าพุต Z จากลอจิก Inverter** gate