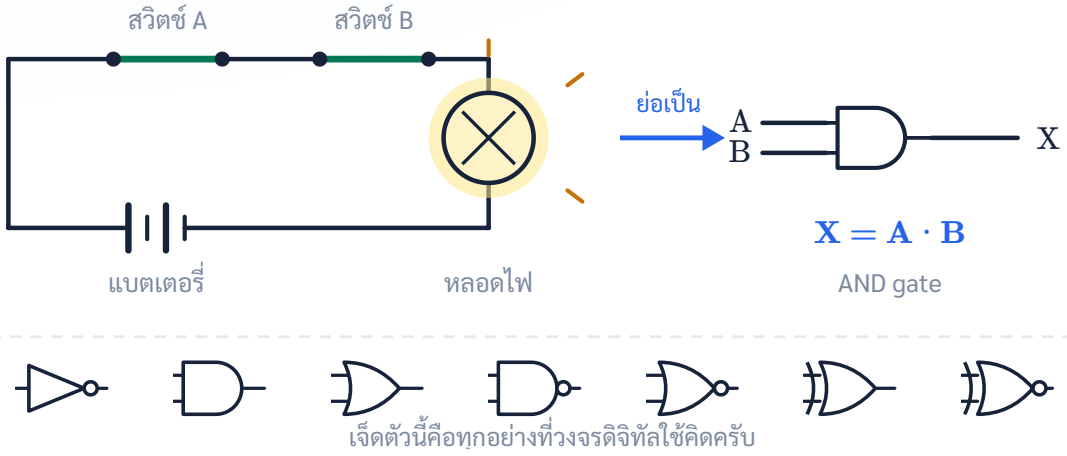
# 05 EP

# ลอจิกเกต 7 ตัว
# จากสวิตช์ไฟบนผนัง สู่วงจรดิจิทัล

NOT · AND · OR · NAND · NOR · XOR · XNOR — ทุกตัวมีครบ สัญลักษณ์ ตารางค่าความจริง นิพจน์บูลีน timing diagram และของใกล้ตัว · negative-AND / negative-OR และ dual symbol




พี่สอนน้องแบบไม่ต้องมีพื้นฐานมาก่อนเลยครับ · อ้างอิงสไลด์ EES271 Lecture 03 หน้า 1–20 + Assignment #3 ข้อ 1–2

# ตอนนี้เราจะเดินไปทางไหนกัน

ก่อนเริ่ม พี่ขอกางแผนที่ให้น้องดูก่อนครับ ว่า EP นี้มีอะไรบ้าง จะได้ไม่หลงทาง

| | | |
|---|---|---|
| 1 | **สวิตช์ไฟสองตัว คือเกตตัวแรกของน้อง** | ของใกล้ตัว |
| 2 | **เครื่องมือ 3 อย่าง: นิพจน์ · ตารางค่าความจริง · timing diagram** | พื้นฐานสุด |
| 3 | **สามตัวแรก: NOT · AND · OR** | ต้องจำให้ได้ |
| 4 | **เอา NOT ไปแปะท้าย: NAND · NOR** | ออกสอบชัวร์ |
| 5 | **XOR — เกตที่น้องใช้มาแล้วตั้งแต่ EP04** | ตัวเอกของ EP |
| 6 | **เกตตัวเดียว แต่วาดได้สองแบบ (dual symbol)** | ยากที่สุด |
| 7 | **ทั้ง 7 เกตในหน้าเดียว + opposing logic** | ทวนก่อนสอบ |
| 8 | **เกตพวกนี้เอาไปใช้ทำอะไรจริง ๆ — 3 ตัวอย่าง** | ของจริง |
| 9 | **ซ้อมมือ: อ่านวงจร → เขียนตารางค่าความจริง** | I DO / WE DO / YOU DO |
| 10 | **เฉลย Assignment 3 ข้อ 1 — AND 3 ขาเข้า** | เฉลยเต็ม 6 ขั้น |
| 11 | **เฉลย Assignment 3 ข้อ 2 — OR 4 ขาเข้า** | เฉลยเต็ม 6 ขั้น |
| 12 | **กับดักรวม + สรุปหน้าเดียว + เช็คลิสต์ก่อนสอบ** | ทวนก่อนสอบ |

## ชีทนี้ใช้ยังไงให้ได้ผลที่สุด

ในชีทจะมีป้าย 3 แบบสลับกันไปครับ อยากให้น้องใช้มันคนละแบบ

**I DO**

**พี่ทำให้ดู** — น้องอ่านอย่างเดียวพอครับ แต่ขอให้อ่าน**เหตุผล**ในแต่ละขั้นด้วย ไม่ใช่แค่ดูว่าทำอะไร

**WE DO**

**ทำด้วยกัน** — มีช่องว่างให้เติม ลองเติมในใจก่อนแล้วค่อยดูเฉลยที่อยู่ถัดลงไปทันทีครับ

**YOU DO**

**ตาน้อง** — เฉลยอยู่ติดกันเลย แต่พี่ขอให้**ปิดไว้ก่อน** ทำเองให้จบแล้วค่อยเปิดเทียบครับ

**เรื่องสีและสัญลักษณ์ในชีทนี้**

EP นี้มีรูปวงจรเยอะมากครับ พี่เลยใช้สีช่วยแยกให้:

**เขียว = ค่า 1 (HIGH)** · **แดง = ค่า 0 (LOW)** · **เส้นประน้ำเงินแนวตั้ง = ที่คั่นช่วงเวลาใน timing diagram**

ในตารางค่าความจริง แถวที่ระบายเขียว คือแถวที่เอาต์พุตเป็น 1 และแถวขาวคือเอาต์พุตเป็น 0 ครับ

ป้ายค่าบนสายไฟใช้แบบนี้: `1` คือ HIGH และ `0` คือ LOW

ส่วน**วงกลมเล็ก ๆ (bubble)** ที่เห็นตามรูปเกต แปลว่า “กลับค่าตรงนี้” เสมอ — เป็นสัญลักษณ์ที่สำคัญที่สุดของ EP นี้ครับ

**มาจากไหน จะไปไหนต่อ**

**มาจาก EP04:** น้องเพิ่งเรียนรหัสและการตรวจจับข้อผิดพลาด — BCD · Gray code · parity · CRC · Hamming code ทั้งหมดนั้นน้องคำนวณด้วย**มือ**บนกระดาษ

**EP05 นี้:** พี่จะให้น้องเห็นว่า**วงจรจริงมันคิดยังไง** และ Gray code กับ parity ที่น้องทำไป ที่จริงคือเกตตัวเดียวชื่อ XOR ครับ

**จะไปต่อที่ EP06:** พีชคณิตบูลีน — วิธี**ย่อ**วงจรให้ใช้เกตน้อยลงโดยผลลัพธ์เหมือนเดิมเป๊ะ

# 1 สวิตช์ไฟสองตัว คือเกตตัวแรกของน้อง

**ตะขอเกี่ยว**

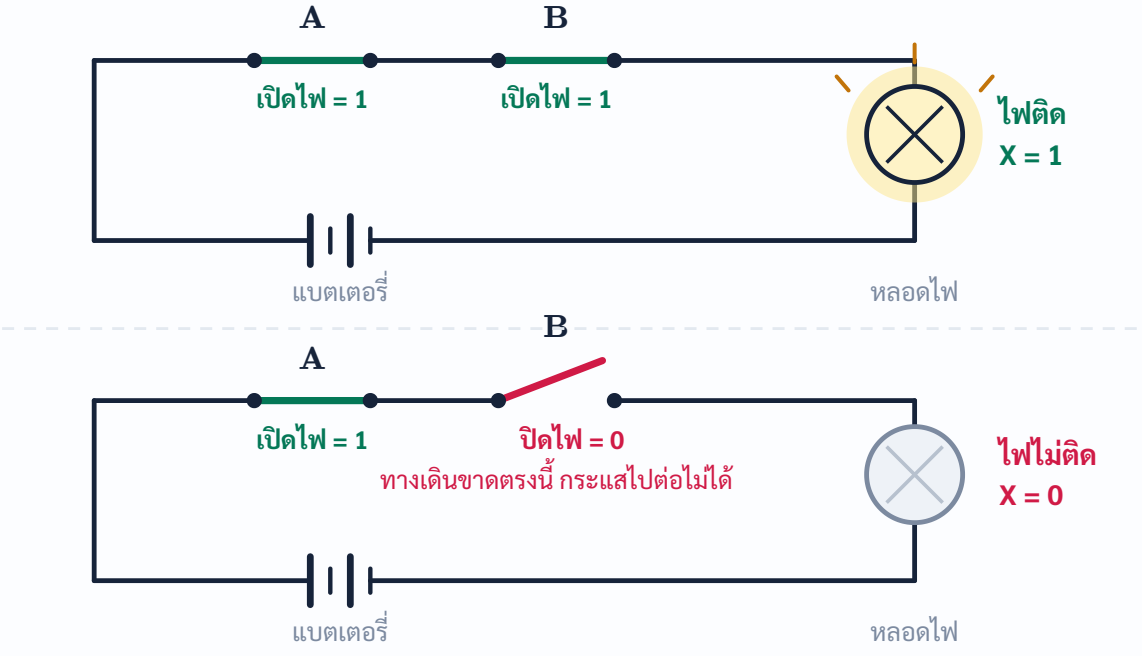
ตอนนี้น้องจัดการเลข 0 กับ 1 ได้หมดแล้วครับ — EP01 แปลงฐาน · EP02 บวกลบด้วย 2's complement · EP03 ยัดเลขทศนิยมลงกล่อง 32 ช่อง · EP04 ใส่พาริตี ทำ CRC ทำ Hamming code จับบิตที่ผิดได้

**ทีนี้ถึงเวลาถามว่า วงจรจริง ๆ มันเอา 0 กับ 1 มาคิดยังไงครับ** ที่ผ่านมาน้องเป็นคนคิดเองบนกระดาษทั้งหมด แต่ในเครื่องจริงไม่มีใครนั่งคิดให้ มีแต่ก้อนเหล็กเล็ก ๆ ที่รับไฟเข้าไปแล้วปล่อยไฟออกมา สมมติว่าถ้าน้องต้องสร้างของที่ “ปล่อยไฟออกก็ต่อเมื่อขาเข้าทั้งสองมีไฟ” ละครับ น้องจะทำยังไง คำตอบอยู่ที่**สวิตช์ไฟบนผนังห้องน้อง**ครับ

## ของใกล้ตัว: เอาสวิตช์ไฟสองตัวมาต่อกัน

ในห้องน้องมีสวิตช์ไฟใช่ไหมครับ สวิตช์มีอยู่สองสถานะเท่านั้น — **เปิด** กับ **ปิด** ไม่มีอะไรอยู่ตรงกลาง นี่คือของแบบ digital เป๊ะ ๆ ตามที่ EP01 สอนไว้ครับ ทีนี้พี่ขอเอาสวิตช์**สองตัว**มาต่อเข้าวงจรเดียวกัน ต่อได้ 2 แบบเท่านั้น แล้วมันให้ผลไม่เหมือนกันเลย

### แบบที่ 1 — ต่ออนุกรม (series): ต้องเปิดทั้งคู่ ไฟถึงจะติด




สวิตช์ต่ออนุกรม — กระแสต้องวิ่งผ่านสวิตช์**ทั้งสองตัว** ถ้ามีตัวใดตัวหนึ่งเปิดวงจร ทางเดินขาด ไฟดับทันทีครับ

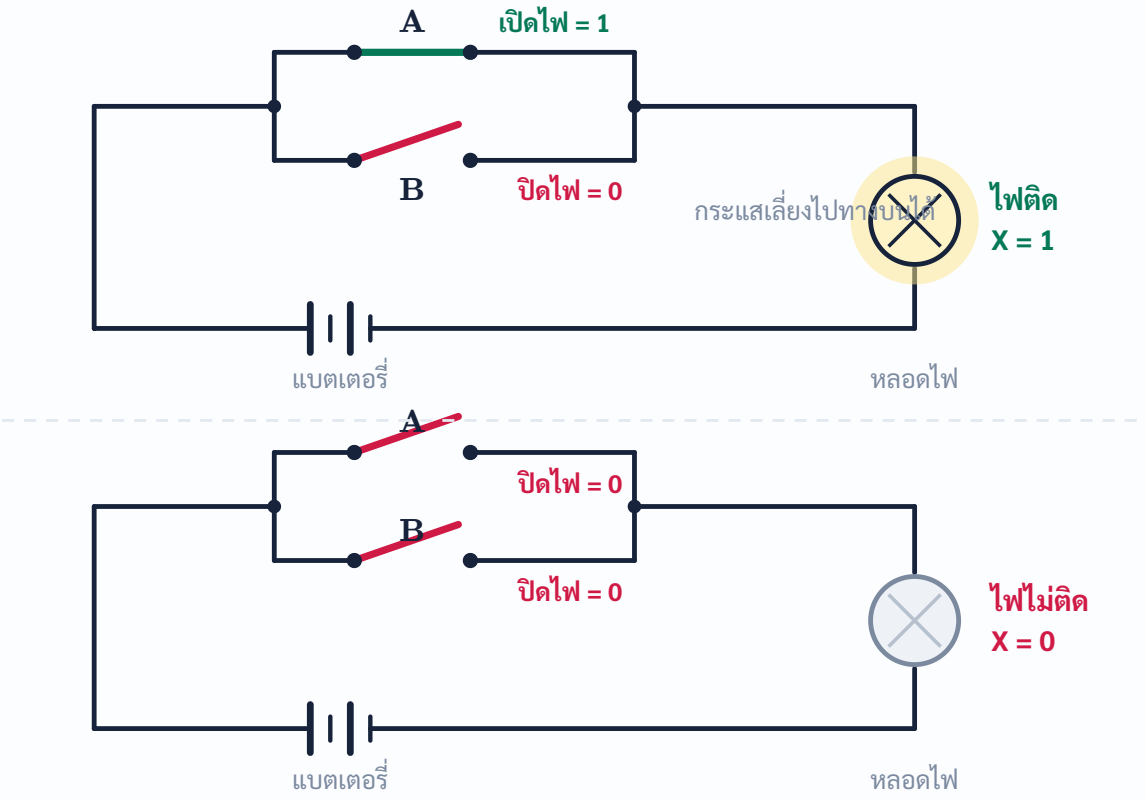
สังเกตให้ดีครับ กระแสไฟฟ้าออกจากแบตเตอรี่แล้วต้องเดินผ่าน A ก่อน แล้วต่อไป B แล้วถึงจะถึงหลอด **ถ้ามีจุดใดจุดหนึ่งขาด ทั้งเส้นก็จบ** เพราะฉะนั้นหลอดจะติดได้กรณีเดียวคือ A เปิด **และ** B เปิดครับ

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

**ภาษาไทยของตารางนี้**

“ไฟติดก็ต่อเมื่อเปิด**ทั้งคู่**” ครับ
คำว่า **ทั้งคู่ / และ** ในภาษาอังกฤษคือ `AND`
ตารางนี้แหละคือ **ตารางค่าความจริงของ AND gate** เป๊ะ ๆ น้องเพิ่งสร้างเกตตัวแรกได้แล้วครับ โดยไม่ต้องรู้จักสัญลักษณ์อะไรเลย

## แบบที่ 2 — ต่อขนาน (parallel): เปิดตัวใดตัวหนึ่ง ไฟก็ติดแล้ว




สวิตช์ต่อขนาน — มี**ทางเดินสองเส้น** ขอแค่เส้นใดเส้นหนึ่งต่อถึงกัน กระแสก็ไหลได้ ไฟติดครับ

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

**ภาษาไทยของตารางนี้**

“ไฟติดถ้าเปิด**ตัวใดตัวหนึ่ง**ก็พอ” ครับ
คำว่า **หรือ** ในภาษาอังกฤษคือ `OR`
ตารางนี้คือ **ตารางค่าความจริงของ OR gate** เป๊ะ ๆ ครับ แถวเดียวที่ไฟดับคือแถวที่ปิดหมดทั้งคู่

เข้าเรื่อง — logic function คืออะไร

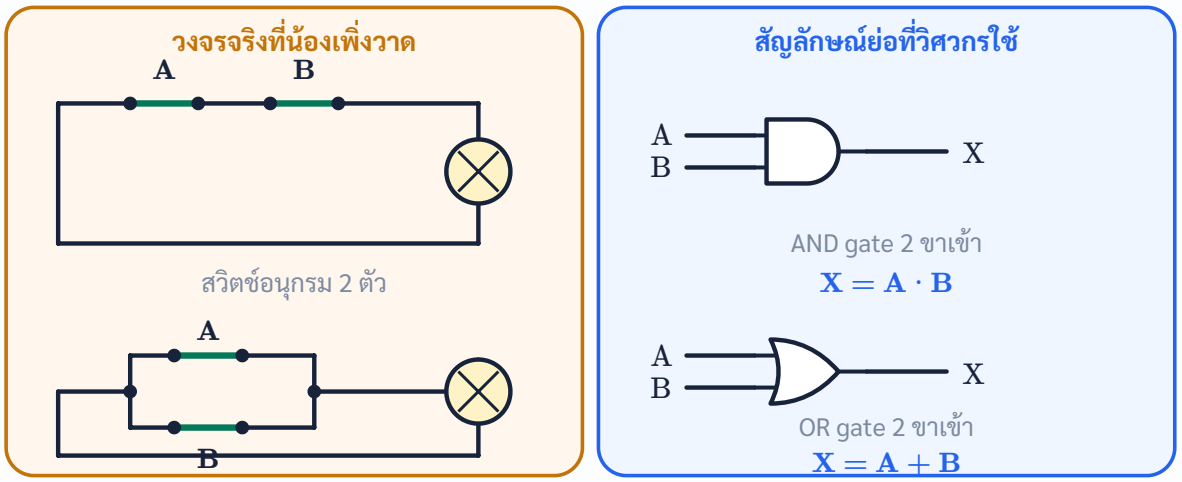
**Many situations and processes that you encounter in your daily life can be expressed in the form of propositional, or logic, functions. Propositions or logic can be classified as true or false.**

แปลเป็นภาษาพี่: เรื่องรอบตัวน้องเยอะมากที่ตัดสินได้แค่ **จริง** หรือ **เท็จ** อย่างเดียว ไม่มีครึ่ง ๆ กลาง ๆ ครับ และของพวกนี้เขียนเป็นสูตรได้

สไลด์อาจารย์ยกตัวอย่างไว้เป๊ะ ๆ ว่า *“The light is on only if the bulb is not burned out and the switch is on.”* — ประโยค “ไฟติด” จะเป็นจริงก็ต่อเมื่อประโยคย่อยอีกสองประโยคเป็นจริง**พร้อมกัน** เห็นคำว่า and ไหมครับ นั่นคือสิ่งที่น้องเพิ่งวาดเป็นวงจรอนุกรมไปเมื่อกี้เลย

## แล้วสัญลักษณ์เกตมันมาจากไหน

วิศวกรขี้เกียจวาดแบตเตอรี่กับหลอดไฟทุกครั้งครับ เลยตกลงกันว่าจะย่อวงจรทั้งวงเหลือ**รูปเดียว** แล้วเขียนแค่ “ขาเข้า” กับ “ขาออก” — นี่คือที่มาของสัญลักษณ์เกตครับ ของสองอันข้างล่างนี้คือ**เรื่องเดียวกัน**




ซ้ายกับขวาคือของอันเดียวกันครับ — สัญลักษณ์เกตคือวงจรสวิตช์ที่ถูกย่อจนเหลือแค่ “ขาเข้า → รูป → ขาออก”

เทียบของเก่า

EP01 พี่บอกว่า digital คือของที่มีแค่ 2 ระดับ — HIGH กับ LOW แล้วพี่ยกตัวอย่างสวิตช์ไฟไปแล้วรอบหนึ่ง ตอนนั้นสวิตช์เป็นแค่ “ตัวอย่างของสัญญาณ” ครับ

EP05 นี้สวิตช์**อันเดิม**กลับมาอีกครั้ง แต่คราวนี้ไม่ใช่ตัวอย่าง — มันคือ**ตัวคำนวณ** เอาสวิตช์สองตัวมาต่อกัน แล้วมันตัดสินใจแทนน้องได้เลยว่าไฟควรติดหรือไม่ นี่แหละครับคือหัวใจของวงจรดิจิทัลทั้งหมด

**กับดักคำว่า “เปิด/ปิด” ในภาษาไทย**

ภาษาไทยงงมากครับ — เราพูดว่า “**เปิด**ไฟ” แปลว่าไฟติด แต่ในทางไฟฟ้าเรียกว่า “**ปิด**วงจร” ( closed circuit ) เพราะวงจรครบวง ส่วน “**ปิด**ไฟ” คือ “**เปิด**วงจร” ( open circuit ) เพราะทางเดินขาด ในชีทนี้พี่จะพูดแบบชาวบ้านคือ “เปิดไฟ = 1” “ปิดไฟ = 0” ตลอด แต่**ในข้อสอบภาษาอังกฤษ** ถ้าเจอคำว่า switch is closed ให้แปลว่า **1** และ switch is open ให้แปลว่า **0** ครับ ตรงข้ามกับความรู้สึกภาษาไทยเป๊ะ ๆ

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| logic function | ฟังก์ชันตรรกะ | expressed in the form of logic functions |
| proposition | ข้อความที่ตัดสินจริง/เท็จได้ | Propositions can be classified as true or false |
| series connection | ต่ออนุกรม (เรียงต่อกัน) | two switches connected in series |
| parallel connection | ต่อขนาน (แยกสองทาง) | two switches connected in parallel |
| closed switch | สวิตช์ที่ต่อถึงกัน = 1 | when the switch is closed |
| open switch | สวิตช์ที่ขาด = 0 | when the switch is open |
| input / output | ขาเข้า / ขาออก | The lines connected to each symbol are the inputs and outputs |

# 2 เครื่องมือ 3 อย่างที่ใช้บรรยายเกตหนึ่งตัว

**ตะขอเกี่ยว**

หัวข้อที่แล้วน้องได้ตารางค่าความจริงของ AND กับ OR มาแล้วครับ สมมติว่าถ้าอาจารย์ไม่ให้ตาราง แต่ให้**กราฟรูปคลื่นไฟฟ้า**มาแทนละครับ น้องจะอ่านออกไหม หรือถ้าให้แค่**สูตร**มาบรรทัดเดียว น้องจะเขียนตารางกลับได้ไหม สไลด์อาจารย์บอกไว้ชัดเลยว่าเกตหนึ่งตัวมองได้ **3 ทาง** และข้อสอบชอบสลับให้ทั้งสามแบบครับ

**เข้าเรื่อง — logic gate และ 3 วิธีมอง**

**Logic functions are indicated by standard distinctive shape symbols. The lines connected to each symbol are the inputs and outputs. Logic symbols used to represent the logic gates are in accordance with ANSI/IEEE Standard 91-1984.**

**Logic gates can be analyzed in three ways: Boolean expression · Truth table · Timing diagrams.**

แปลเป็นภาษาพี่: รูปเกตที่น้องเห็นไม่ใช่รูปที่ใครวาดเล่น ๆ ครับ มันเป็น**มาตรฐานสากล**ชื่อ ANSI/IEEE 91-1984 ทั้งโลกใช้รูปเดียวกัน (แบบที่มีรูปทรงเฉพาะตัวนี้เรียกว่า `distinctive shape` ซึ่งตำรา Floyd ที่วิชานี้ใช้ก็ใช้แบบนี้) และเกตแต่ละตัวบรรยายได้ 3 ทาง ซึ่ง**ทั้งสามทางบอกเรื่องเดียวกันเป๊ะ** แค่คนละภาษาครับ

### ① Boolean expression

**สูตรสั้น ๆ บรรทัดเดียว** เช่น $X = AB$

ข้อดี: เขียนเร็ว เอาไปย่อต่อได้ (EP06)

ข้อเสีย: มองไม่เห็นภาพว่าเกิดอะไรขึ้น

### ② Truth table

**ตารางไล่ทุกความเป็นไปได้**

ข้อดี: ครบถ้วน ไม่มีทางตกกรณี

ข้อเสีย: $n$ ขาเข้า ต้องเขียน $2^n$ แถว

### ③ Timing diagram

**กราฟไฟฟ้าเทียบกับเวลา**

ข้อดี: เห็นของจริงที่วัดได้ด้วยออสซิลโลสโคป

ข้อเสีย: ต้องหัดอ่านสักหน่อย

## เครื่องมือที่ 2 — ตารางค่าความจริง เขียนยังไงไม่ให้ตกแถว

ตารางค่าความจริงคือการ**ไล่ทุกความเป็นไปได้ของขาเข้า**ครับ ขาเข้า $n$ ขา แต่ละขาเป็นได้ 2 ค่า เพราะฉะนั้นมี $2^n$ แถว — 2 ขาได้ 4 แถว · 3 ขาได้ 8 แถว · 4 ขาได้ 16 แถว ตัวเลขนี้ใช้**เช็คตัวเองว่าเขียนครบไหม**ครับ ถ้าเขียนได้ 7 แถวสำหรับ 3 ขา แปลว่าตกไป 1 แถวแน่นอน

**1**

| A | B | C | |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | เร็ว |
| 0 | 1 | 0 | |

**คอลัมน์ขวาสุดสลับทุกแถว**

เพราะเราจะได้ไม่ต้องคิดเลย — คอลัมน์ขวาสุดเขียน 0 1 0 1 0 1 ... สลับไปเรื่อยครับ

2




**คอลัมน์ถัดไปสลับทุก 2 แถว**

เพราะแต่ละค่าของคอลัมน์นี้ต้องอยู่คู่กับ 0 และ 1 ของคอลัมน์ขวาให้ครบก่อนครับ

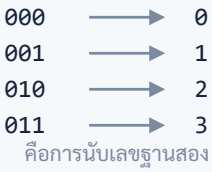
3




**คอลัมน์ซ้ายสุดสลับทุก 4 แถว**

เพราะเลขนับซ้ำสองเท่าทุกครั้งที่ขยับไปทางซ้ายหนึ่งคอลัมน์ครับ — ขวาสุดทุก 1 · ถัดมาทุก 2 · ถัดมาทุก 4

4




**ที่จริงมันคือการนับเลขฐานสองจาก EP01**

ตารางค่าความจริงที่เรียงถูก คือการนับ 000, 001, 010, 011, ... เป๊ะ ๆ ครับ เพราะฉะนั้นถ้าไม่แน่ใจว่าเรียงถูกไหม ให้แปลงเป็นฐานสิบดู ต้องได้ 0, 1, 2, 3, ... เรียงกันไม่ขาดครับ

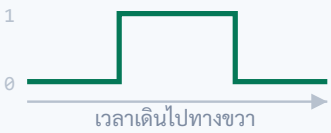
**ทางลัดเช็คตารางก่อนส่ง**

นับจำนวนแถวก่อนเสมอครับ — ต้องได้ $2^n$ พอดี แล้วดูคอลัมน์ซ้ายสุด: **ครึ่งบนต้องเป็น 0 ทั้งหมด ครึ่งล่างต้องเป็น 1 ทั้งหมด** ถ้าไม่ใช่แบบนี้ แปลว่าเรียงผิดหรือตกแถวครับ

## เครื่องมือที่ 3 — timing diagram อ่านยังไง

ในโลกจริงขาเข้าของเกตไม่ได้อยู่นิ่ง ๆ ครับ มันเปลี่ยนไปเรื่อยตามเวลา `timing diagram` คือกราฟที่**เวลาเดินจากซ้ายไปขวา** เส้นอยู่**ระดับบน** = สัญญาณเป็น `1` (HIGH) · อยู่**ระดับล่าง** = เป็น `0` (LOW) ตรงกับ digital pulse ที่ EP01 สอนไว้เป๊ะครับ

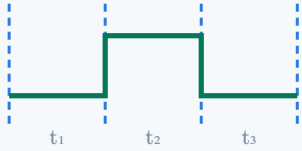
1




**อ่านความสูงของเส้น ไม่ใช่รูปร่าง**

สูง = 1 · ต่ำ = 0 ครับ ส่วนเส้นตั้งที่พุ่งขึ้น-ลงคือ**จังหวะเปลี่ยนค่า** ไม่ได้มีความหมายอะไรในตัวมันเอง เพราะในโลกอุดมคติเราถือว่าเปลี่ยนทันทีครับ

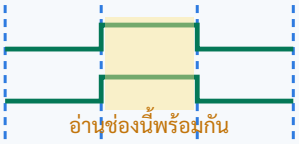
2




**เส้นประแนวตั้งคือ “ที่คั่นช่อง”**

เขาลากเส้นประไว้เพื่อบอกว่า**ช่วงเวลานี้สัญญาณคงที่**ครับ ในชีทนี้พี่ตั้งชื่อแต่ละช่องว่า $t_1, t_2, t_3, \ldots$ เพื่อให้น้องชี้ได้ว่า “ช่อง $t_2$ ตรงกับแถวไหนของตาราง”

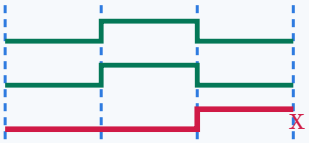
**3**




**อ่านทีละช่อง อ่านทุกเส้นพร้อมกัน**

เอานิ้วชี้ตั้งฉากลงมาที่ช่องหนึ่ง แล้วอ่านค่าของ A และ B **ในช่องเดียวกัน** จะได้ค่าคู่หนึ่ง เช่น $(A, B) = (1, 0)$ ครับ แล้วเอาคู่นั้นไปเปิดตารางค่าความจริง จะได้ค่าของ $X$ ในช่องนั้นทันที

**4**




**เขียนเส้นเอาต์พุตทีละช่อง**

ทำซ้ำทุกช่องจนครบ แล้วต่อจุดให้เป็นเส้นเดียว นั่นคือรูปคลื่นของเอาต์พุตครับ **เอาต์พุตจะเปลี่ยนค่าได้เฉพาะตรงที่เส้นประเท่านั้น** เพราะระหว่างช่องขาเข้ายังคงที่อยู่

**กติกาของ timing diagram ในชีทนี้ (ใช้เหมือนกันทุกเกต)**

พี่จะใช้รูปแบบเดียวกันตลอด EP นี้ครับ เพื่อให้น้องเทียบเกตทั้ง 7 ตัวได้ทันที:
ช่วงเวลามี 4 ช่อง $t_1 \dots t_4$ และค่าขาเข้าคือ $(A, B) = (0,0) \to (0,1) \to (1,0) \to (1,1)$
**ซึ่งตรงกับแถวที่ 1, 2, 3, 4 ของตารางค่าความจริงพอดี** เพราะฉะนั้นถ้าน้องอ่านตารางได้ น้องวาด timing diagram ได้ทันทีโดยไม่ต้องคิดใหม่ครับ

`1` = HIGH = เส้นอยู่ระดับบน · `0` = LOW = เส้นอยู่ระดับล่าง

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| truth table | ตารางค่าความจริง | Write a truth table for the following gate |
| Boolean expression | นิพจน์บูลีน (สูตร) | stated by the following Boolean expression |
| timing diagram | แผนภาพเวลา (รูปคลื่น) | Sketch the output waveform / timing diagram |
| distinctive shape symbol | สัญลักษณ์รูปทรงเฉพาะ | standard distinctive shape symbols |
| HIGH / LOW | ระดับสูง / ระดับต่ำ | a HIGH on the output |
| waveform | รูปคลื่นสัญญาณ | the input waveforms shown |

# 3 สามตัวแรก: NOT · AND · OR

**ตะขอเกี่ยว**

หัวข้อ 1 น้องสร้าง AND กับ OR ได้เองจากสวิตช์แล้วครับ สมมติว่าถ้าน้องอยากได้ของที่ “**กลับค่า**” ละครับ — ใส่ 0 เข้าไปแล้วได้ 1 ออกมา ของแบบนี้ต่อสวิตช์ธรรมดาไม่ได้ ต้องมีเกตอีกตัวหนึ่ง และมันคือเกตที่**เล็กที่สุดแต่ใช้บ่อยที่สุด**ในทั้งวิชานี้ครับ

ในหัวข้อนี้พี่จะเล่าเกตทีละตัว และ**ทุกตัวจะมีครบ 5 อย่างเสมอ**: ① สัญลักษณ์ ② ตารางค่าความจริง ③ นิพจน์บูลีน ④ timing diagram ⑤ ของใกล้ตัว ถ้าน้องจับ 5 อย่างนี้ได้ทุกตัว ข้อสอบส่วนเกตจะไม่มีอะไรเซอร์ไพรส์เลยครับ

## เกตที่ 1 — NOT gate (inverter)

**เข้าเรื่อง**

**The NOT gate performs an inversion operation, changing the input from TRUE to FALSE (1 to 0), or from FALSE to TRUE (0 to 1).**

แปลเป็นภาษาพี่: NOT gate คือ**ตัวกลับค่า**ครับ ใส่อะไรเข้าไปมันเอาตรงข้ามออกมาให้ มีขาเข้าแค่ **ขาเดียว** — เป็นเกตเดียวใน 7 ตัวที่มีขาเข้าขาเดียว อีกชื่อหนึ่งที่เจอบ่อยกว่าคือ `inverter` ครับ

**⑤ ของใกล้ตัว:** เหรียญบาทในกระเป๋าน้องครับ วางหงายอยู่แล้วพลิกหนึ่งที หัวกลายเป็นก้อย ก้อยกลายเป็นหัว — **พลิกทีเดียวได้ค่าตรงข้ามเสมอ** และถ้าพลิกสองทีติดกัน มันกลับมาเหมือนเดิม (นี่คือกฎ $\overline{\overline{A}} = A$ ที่ EP06 จะเรียกว่า involution ครับ)




วงกลมเล็กที่ปลาย = ตัวกลับค่า

① สัญลักษณ์ NOT — สามเหลี่ยม + วงกลมเล็ก

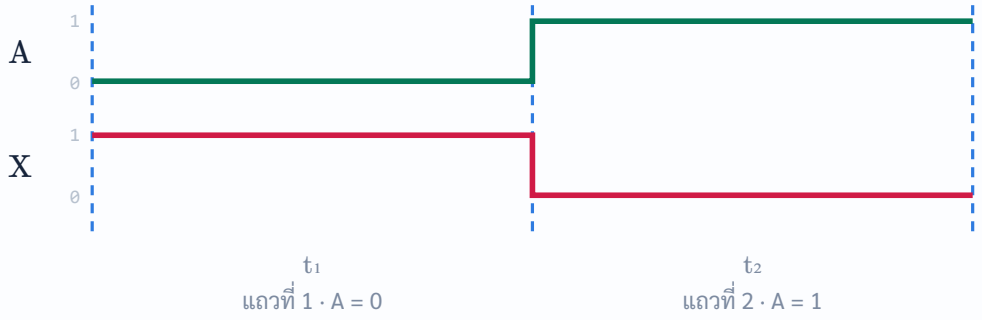
| A | $X = \overline{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

$$X = \overline{A}$$

③ อ่านว่า “X เท่ากับ A บาร์” · ขีดบนหัว A แปลว่ากลับค่า

วงกลมเล็กนั้นสำคัญมาก

วงกลมเล็ก ๆ ที่ปลายเรียกว่า bubble ครับ **เจอ bubble ที่ไหนแปลว่ากลับค่าที่นั่น** — กฎข้อนี้ใช้ได้กับทุกเกตในวิชานี้ NAND ก็คือ AND ที่มี bubble ที่ขาออก · NOR ก็คือ OR ที่มี bubble ที่ขาออก ถ้าน้องจำกฎนี้ข้อเดียว น้องจะไม่ต้องท่องเกตทีละตัวเลยครับ




④ timing diagram ของ NOT — เส้นเขียวคือขาเข้า เส้นแดงคือขาออก สังเกตว่าเส้นแดง**อยู่ตรงข้ามกับเส้นเขียวเสมอ** ช่อง $t_1$ ตรงกับแถวแรกของตาราง ช่อง $t_2$ ตรงกับแถวที่สองครับ

## เกตที่ 2 — AND gate

เข้าเรื่อง

**The AND gate produces a TRUE result only when all inputs to the gate are TRUE (HIGH or 1). Otherwise, it produces a FALSE result (LOW or 0). The AND gate performs a multiplication operation.**

แปลเป็นภาษาพี่: AND ให้ 1 ออกมาเฉพาะตอนที่ขาเข้า**เป็น 1 หมดทุกขา** ขาดไปขาเดียวก็ได้ 0 ครับ คำสำคัญคือ only when all
และสไลด์บอกว่ามันคือ **การคูณ** — ที่มาคือ $0 \times 0 = 0, 0 \times 1 = 0, 1 \times 0 = 0, 1 \times 1 = 1$ ลองเทียบกับตารางข้างล่างสิครับ **เหมือนกันเป๊ะทั้ง 4 แถว** เพราะฉะนั้นเราถึงเขียน AND ด้วยเครื่องหมายคูณครับ

⑤ **ของใกล้ตัว:** ลิ้นชักที่ต้องไขกุญแจสองดอกครับ — ดอกแรกปลดสลักบน ดอกที่สองปลดสลักล่าง ไขดอกเดียวลิ้นชักยังไม่เปิด ต้องไข**ครบทั้งสองดอก**ถึงจะดึงออกมาได้ หรือจะนึกถึงสวิตช์อนุกรมสองตัวจากหัวข้อ 1 ก็เรื่องเดียวกันครับ



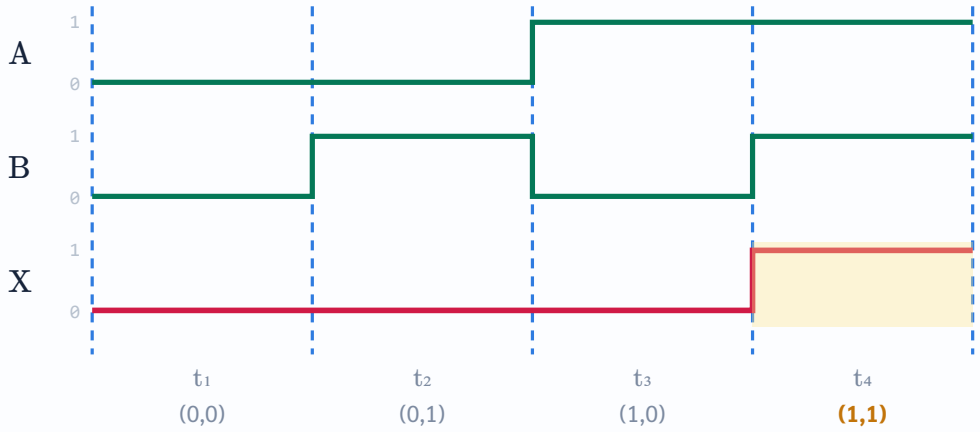

ด้านหน้าโค้งครึ่งวงกลม ด้านหลังตรง

① สัญลักษณ์ AND — จำว่า “หน้าโค้งเหมือนตัว D”

| A | B | $X = AB$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$X = A \cdot B = AB$$

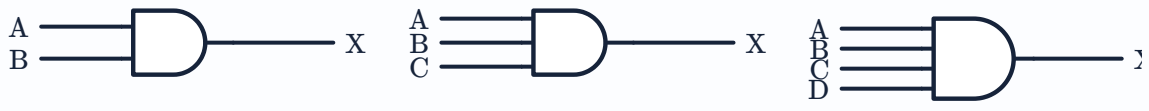
③ จุดคูณละไว้ในฐานที่เข้าใจได้ เหมือน $2x$ ในพีชคณิต




④ timing diagram ของ AND — **ช่วงสีเหลืองคือช่วงเดียวที่เอาต์พุตขึ้นสูง** คือ $t_4$ ที่ทั้ง A และ B สูงพร้อมกัน ตรงกับแถวสุดท้ายของตารางที่ระบายเขียวไว้ครับ

### AND ที่มีมากกว่า 2 ขาเข้า

กฎเดิมเป๊ะครับ — **ทุกขาต้องเป็น 1 หมด** ถึงจะได้ 1 ไม่ว่าจะมีกี่ขาก็ตาม จำนวนแถวของตารางจะโตเป็น $2^n$ ตามจำนวนขาครับ




**X = AB**
2 ขาเข้า · 4 แถว

**X = ABC**
3 ขาเข้า · 8 แถว

**X = ABCD**
4 ขาเข้า · 16 แถว

AND 2 · 3 · 4 ขาเข้า — สังเกตว่ารูปสูงขึ้นตามจำนวนขา แต่**ความหมายไม่เปลี่ยน** คือ “ต้อง 1 หมดทุกขา”

**ทางลัดของ AND ทุกขนาด**

ตาราง AND ไม่ว่ากี่ขา มี **แถวเดียว**เท่านั้นที่ได้ 1 คือ**แถวล่างสุด** (1 หมดทุกช่อง) ที่เหลือเป็น 0 ทั้งหมดครับ เพราะฉะนั้นเวลาทำข้อสอบ ให้เติม 0 ลงทั้งคอลัมน์ก่อน แล้วค่อยแก้แถวสุดท้ายเป็น 1 เร็วกว่าไล่ทีละแถวมาก และไม่มีทางผิดครับ

## เกตที่ 3 — OR gate

**เข้าเรื่อง**

**The OR gate produces a TRUE result when one or all of the inputs are TRUE (HIGH or 1). Otherwise, it produces a FALSE result (LOW or 0). The OR gate performs a logical addition operation.**

แปลเป็นภาษาพี่: OR ให้ 1 ออกมาถ้ามีขาเข้า**อย่างน้อยหนึ่งขา**เป็น 1 จะเป็นขาเดียวหรือครบทุกขาก็ได้เหมือนกันครับ คำสำคัญคือ `one or all`
สไลด์บอกว่ามันคือ **การบวก** — $0+0=0, 0+1=1, 1+0=1$, และ... $1+1$ ในโลกบูลีน**เท่ากับ 1** ไม่ใช่ 2 ครับ เพราะเรามีแค่ 0 กับ 1 เท่านั้น ค่านี้แหละที่ทำให้เรียกว่า “logical addition” ไม่ใช่การบวกธรรมดา

⑤ **ของใกล้ตัว:** กระเป๋าใบเดียวที่มีซิปสองด้านครับ จะเปิดซิปด้านซ้ายหรือด้านขวาก็หยิบของได้เหมือนกัน **ขอแค่เปิดสักด้านหนึ่ง** และถ้าเปิดทั้งสองด้านพร้อมกันก็ยังหยิบได้อยู่ดี — ไม่มีอะไรพิเศษขึ้นครับ นี่คือเหตุผลที่ $1+1=1$




หลังเว้าเข้า หน้าแหลมออก

① สัญลักษณ์ OR — จำว่า “ปลายแหลมเหมือนหัวลูกศร”

| A | B | $X = A + B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

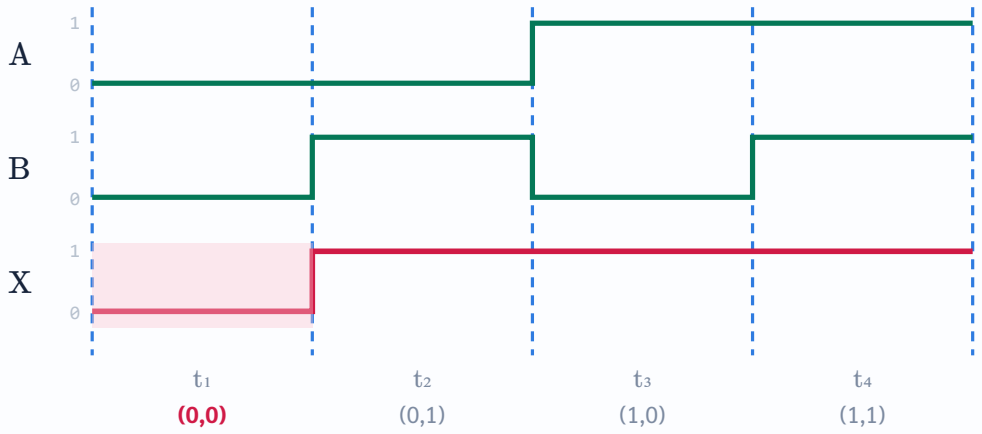
$$X = A + B$$

③ เครื่องหมายบวกในบูลีนแปลว่า OR ไม่ใช่การบวกเลข

**กับดักที่คนพลาดกันทุกปี**

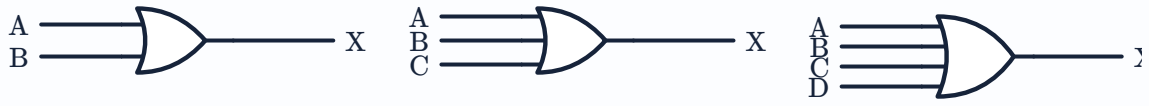
เห็นเครื่องหมาย $+$ แล้วเผลอคิดว่า $1 + 1 = 2$ ครับ — **ผิด** ในพีชคณิตบูลีนมีแค่ 0 กับ 1 เท่านั้น ไม่มีเลข 2 ให้ใช้ เพราะฉะนั้น $1 + 1 = 1$

วิธีกันพลาด: อย่าอ่านว่า “บวก” ครับ ให้**อ่านว่า “ออร์”** ทุกครั้ง $A + B$ อ่านว่า “A ออร์ B” แล้วน้องจะไม่เผลอเอาไปบวกเลขจริงเลยครับ




④ timing diagram ของ OR — คราวนี้กลับกันครับ **ช่วงสีชมพูคือช่วงเดียวที่เอาต์พุตต่ำ** คือ $t_1$ ที่ทั้งคู่ต่ำพร้อมกัน ตรงกับแถวแรกของตารางที่เป็นแถวแดงแถวเดียวครับ

## OR ที่มีมากกว่า 2 ขาเข้า




$\mathbf{X = A + B}$
2 ขาเข้า · 4 แถว

$\mathbf{X = A + B + C}$
3 ขาเข้า · 8 แถว

$\mathbf{X = A + B + C + D}$
4 ขาเข้า · 16 แถว

OR 2 · 3 · 4 ขาเข้า — **รูป 4 ขาเข้าตัวนี้แหละคือโจทย์ Assignment 3 ข้อ 2** ที่เดี๋ยวเราจะทำกันครับ

**ทางลัดของ OR ทุกขนาด**

ตรงข้ามกับ AND เป๊ะครับ — ตาราง OR ไม่ว่ากี่ขา มี **แถวเดียว**ที่ได้ 0 คือ**แถวบนสุด** (0 หมดทุกช่อง) ที่เหลือเป็น 1 หมด เพราะฉะนั้นเติม 1 ลงทั้งคอลัมน์ก่อน แล้วแก้แถวแรกเป็น 0 ครับ ข้อ 2 ของ Assignment มี 16 แถว ทางลัดนี้ทำให้เขียนเสร็จใน 10 วินาทีครับ

เทียบของเก่า

EP02 น้องบวกเลขฐานสอง $1 + 1 = 10$ (ได้ผลลัพธ์ 0 ทดไป 1) จำได้ไหมครับ
ในบูลีน $1 + 1 = 1$ — **ไม่ขัดกันนะครับ เพราะเป็นคนละเรื่อง** การบวกของ EP02 คือเลขคณิต มีตัวทดได้ ส่วน $+$ ในบูลีนคือ OR ไม่มีตัวทด ให้ดูที่บริบท: ถ้ากำลังพูดถึงเกต $+$ คือ OR เสมอครับ (ที่จริงตัวทดของ $1 + 1$ ใน EP02 คืองานของ AND และผลบวกคืองานของ XOR ซึ่งเดี๋ยวหัวข้อ 5 จะเจอครับ)

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| inverter / NOT gate | ตัวกลับค่า | The NOT gate performs an inversion operation |
| inversion / complement | การกลับค่า | the complement of the input |
| bubble | วงกลมเล็กที่แปลว่ากลับค่า | the bubble on the output |
| only when all inputs are | ก็ต่อเมื่อทุกขาเข้าเป็น | produces a TRUE result only when all inputs are TRUE |
| one or all of the inputs | ขาใดขาหนึ่งหรือทุกขา | when one or all of the inputs are TRUE |
| multiplication operation | การคูณ (คือ AND) | The AND gate performs a multiplication operation |
| logical addition | การบวกเชิงตรรกะ (คือ OR) | The OR gate performs a logical addition operation |

# 4 เอา NOT ไปแปะท้าย: NAND · NOR

**ตะขอเกี่ยว**

หัวข้อที่แล้วน้องมี AND · OR · NOT ครบแล้วครับ สมมติว่าถ้าน้องเอา NOT ไป**แปะท้าย** AND ละครับ จะได้อะไร คำตอบคือได้เกตตัวใหม่ที่มีชื่อของตัวเอง และมันเป็นเกตที่**ใช้จริงในโรงงานมากกว่า AND เสียอีก** (EP08 จะเฉลยว่าทำไม) วิธีจำง่ายมากครับ: เอาชื่อเดิมมาเติม **N** ข้างหน้า แล้วกลับค่าเอาต์พุตให้หมด

## เกตที่ 4 — NAND gate

**เข้าเรื่อง**

**The NAND gate performs AND operation followed by a NOT operation. The result is therefore the inverse of an AND gate.**

แปลเป็นภาษาพี่: NAND = AND แล้วค่อย NOT ครับ ชื่อมันบอกอยู่แล้ว — **N**ot-**AND** เพราะฉะนั้นวิธีทำตารางเร็วที่สุดคือ **เขียนตาราง AND ก่อน แล้วกลับค่าคอลัมน์สุดท้ายทั้งคอลัมน์** ไม่ต้องคิดใหม่เลยครับ




AND แล้วต่อด้วย NOT (2 เกต)

$$X = \text{NOT}(A \cdot B)$$

NAND (เกตเดียว)

$$X = \overline{A \cdot B}$$

bubble ที่ขาออกของ NAND = NOT ที่ถูกยุบเข้าไปในตัวเกตครับ

① สัญลักษณ์ NAND = รูป AND + bubble ที่ขาออก — ประหยัดไปหนึ่งเกตเทียบกับการวาด AND ต่อ NOT

⑤ **ของใกล้ตัว:** กล่องดินสอที่มีตัวล็อกสองอันครับ สมมติว่ามีไฟแดงเตือนว่า “ยังปิดไม่สนิท” ไฟแดงจะ**ติดค้างอยู่ตลอด** ไม่ว่าจะล็อกอันเดียวหรือยังไม่ล็อกเลย มันจะดับ**ก็ต่อเมื่อกดล็อกครบทั้งสองอัน**เท่านั้น — นี่คือ NAND เป๊ะครับ (1 คือปกติ 0 คือกรณีพิเศษที่ครบทุกอย่าง)

| A | B | $AB$ | $X = \overline{AB}$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**อ่านตารางนี้ยังไง**

คอลัมน์ที่ 3 คือ **ผลของ AND ก่อน** ครับ (เหมือนสไลด์ที่เขียนหัวคอลัมน์ว่า $AB$) แล้วคอลัมน์สุดท้ายคือ**กลับค่าจากคอลัมน์ที่ 3**

ข้อสอบชอบใส่คอลัมน์กลางแบบนี้มาให้ เพราะอยากให้เห็นขั้นตอน ถ้าไม่ให้มา น้องเติมเองได้เลยครับ จะได้ไม่พลาด

$$X = \overline{A \cdot B} = \overline{AB}$$

③ ขีดบน**คลุมทั้งก้อน** ไม่ใช่คลุมทีละตัว

**กับดักเรื่องขีดบน**

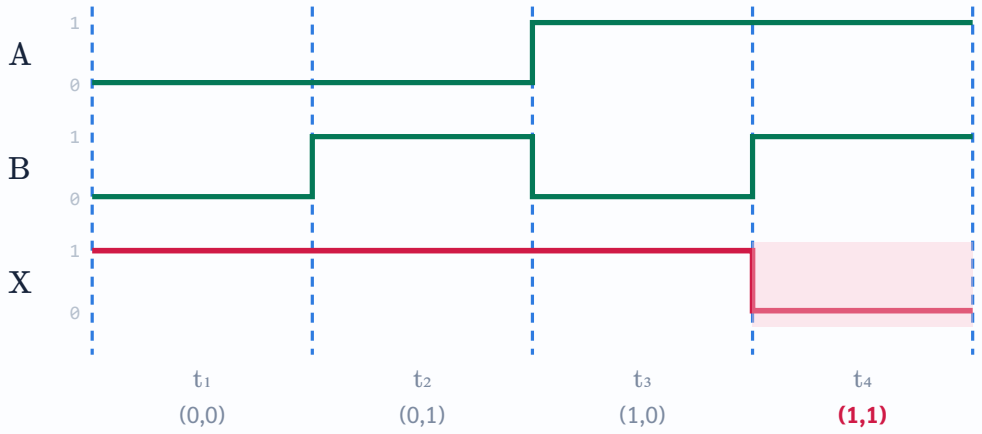
$\overline{AB}$ กับ $\overline{A}\,\overline{B}$ **ไม่เท่ากันครับ**

$\overline{AB}$ = ทำ AND ก่อน แล้วค่อยกลับค่าทีเดียว → ที่ $A = 1, B = 1$ ได้ $\overline{1} = 0$

$\overline{A}\,\overline{B}$ = กลับค่าทีละตัวก่อน แล้วค่อย AND → ที่ $A = 1, B = 1$ ได้ $0 \cdot 0 = 0$ (บังเอิญตรงกัน) แต่ที่ $A = 0, B = 0$: อันแรกได้ $\overline{0} = 1$ ส่วนอันหลังได้ $1 \cdot 1 = 1$ ... ก็ยังตรงกันอีก

ลองที่ $A = 0, B = 1$ สิครับ: อันแรก $\overline{0} = \mathbf{1}$ · อันหลัง $1 \cdot 0 = \mathbf{0}$ — **ต่างกันแล้ว**

**ความยาวของขีดบนคือตัวบอกลำดับการทำ** ขีดยาวคลุมอะไร แปลว่าทำข้างในให้เสร็จก่อนแล้วค่อยกลับค่าครับ




④ timing diagram ของ NAND — เอาเส้นแดงของ AND มา**พลิกกลับหัว**ได้เลยครับ เอาต์พุตสูงตลอด แล้วตกลงมาต่ำแค่ช่วง $t_4$ ช่วงเดียว

## เกตที่ 5 — NOR gate

เข้าเรื่อง

**The NOR gate performs OR operation followed by a NOT operation. The result is therefore the inverse of an OR gate.**

แปลเป็นภาษาพี่: NOR = OR แล้วค่อย NOT ครับ (**N**ot-**OR**) วิธีทำตารางก็เหมือนเดิม — **เขียนตาราง OR ก่อน แล้วกลับค่าคอลัมน์สุดท้ายทั้งคอลัมน์**




OR แล้วต่อด้วย NOT (2 เกต)

$$X = \text{NOT}(A + B)$$

NOR (เกตเดียว)

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

bubble อยู่ที่เดิมเสมอ — ปลายขาออกครับ

① สัญลักษณ์ NOR = รูป OR + bubble ที่ขาออก

**⑤ ของใกล้ตัว:** ลิ้นชักสองใบในโต๊ะเรียนครับ สมมติว่ามีไฟเขียวบอกว่า “เก็บของเรียบร้อยแล้ว” ไฟเขียวจะติด**ก็ต่อเมื่อลิ้นชักปิดสนิททั้งสองใบ** ถ้ามีใบใดใบหนึ่งเปิดค้างไว้ ไฟก็ดับทันที — ให้ 1 แปลว่า “ลิ้นชักถูกดึงออก” แล้วน้องจะได้ตาราง NOR เป๊ะครับ

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

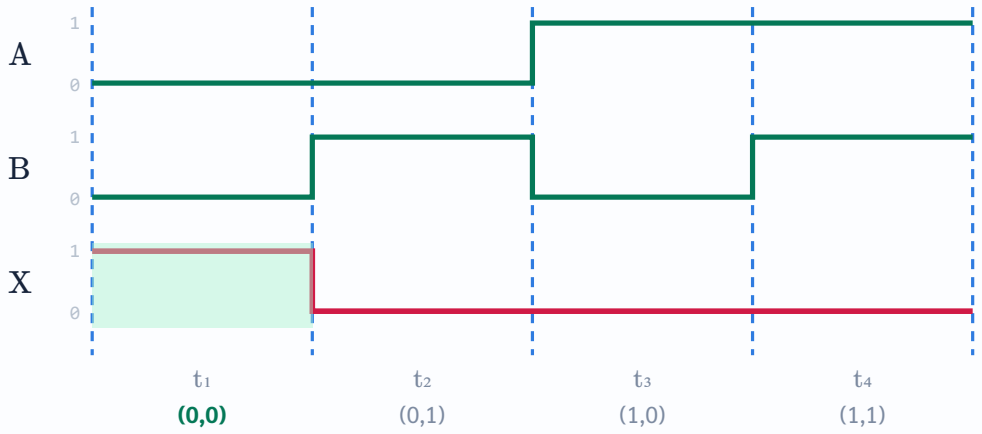
จำ NOR ได้ด้วยประโยคเดียว

**“NOR ให้ 1 แถวเดียว คือแถวที่ 0 หมด”** ครับ

เทียบกัน: AND ให้ 1 แถวเดียว คือแถวที่ 1 หมด · NOR ให้ 1 แถวเดียว คือแถวที่ 0 หมด สองตัวนี้เป็นตาราง “แถวเดียวโดด” เหมือนกัน แค่คนละมุมของตารางครับ

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

③ ขีดบนคลุม $A + B$ ทั้งก้อน — ทำ OR ให้เสร็จก่อนแล้วค่อยกลับค่า




④ timing diagram ของ NOR — เอาเส้นแดงของ OR มาพลิกกลับหัว เอาต์พุตขึ้นสูงแค่ช่วง $t_1$ ช่วงเดียว คือช่วงที่ขาเข้าต่ำทั้งคู่ครับ

### NAND กับ NOR ที่มี 3 ขาเข้า




NAND 3 ขา
$X = \overline{ABC}$
0 แถวเดียว คือแถวล่างสุด




NOR 3 ขา
$X = \overline{A + B + C}$
1 แถวเดียว คือแถวบนสุด

NAND 3 ขาให้ 0 แถวเดียว (แถวที่ 1 หมด) · NOR 3 ขาให้ 1 แถวเดียว (แถวที่ 0 หมด) — กฎเดิมทุกขนาดครับ

**ทางลัด 3 บรรทัดที่ครอบ 4 เกตแรก**

**AND** → ได้ 1 แถวเดียว: แถวล่างสุด · **NAND** → เอาของ AND มาพลิก: ได้ 0 แถวเดียว แถวล่างสุด
**OR** → ได้ 0 แถวเดียว: แถวบนสุด · **NOR** → เอาของ OR มาพลิก: ได้ 1 แถวเดียว แถวบนสุด
เท่านี้น้องเขียนตารางของ 4 เกตนี้ได้ทันทีทุกขนาด โดยไม่ต้องไล่ทีละแถวเลยครับ

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| NAND gate | เกต AND ที่กลับค่าเอาต์พุต | performs AND operation followed by a NOT operation |
| NOR gate | เกต OR ที่กลับค่าเอาต์พุต | performs OR operation followed by a NOT operation |
| the inverse of | ตรงกันข้ามกับ | the result is the inverse of an AND gate |
| equivalent NAND logic | วงจรที่สมมูลกับ NAND | Equivalent NAND logic |
| overbar | ขีดบน (เครื่องหมายกลับค่า) | the overbar covers the entire expression |

# 5 XOR — เกตที่น้องใช้มาแล้วตั้งแต่ EP04

**ตะขอเกี่ยว**

EP04 น้องทำ Gray code ใช่ไหมครับ — จำวิธีได้ไหม “**เอาบิตที่ติดกันมาเทียบกัน ถ้าต่างกันเขียน 1 ถ้าเหมือนกันเขียน 0**” แล้วตอนทำ parity น้องก็นับจำนวน 1 ว่าคู่หรือคี่

สมมติว่าถ้าพี่บอกน้องว่า **ทั้งสองเรื่องนั้นคือเกตตัวเดียวกัน** ละครับ และมันเป็นเกตที่น้อง**ใช้เป็นแล้ว**ตั้งแต่ EP ที่แล้ว แค่ยังไม่รู้ชื่อมันเท่านั้น หัวข้อนี้พี่จะแนะนำชื่อจริงของมันให้รู้จักครับ

## เกตที่ 6 — XOR gate (exclusive-OR)

**เข้าเรื่อง**

**The XOR gate performs an exclusive-OR operation. It produces a TRUE result only when the inputs differ. If the inputs are the same (1,1 or 0,0), the XOR gate produces a FALSE result.**

แปลเป็นภาษาพี่: **ต่างกันได้ 1 · เหมือนกันได้ 0** ครับ จำแค่ประโยคนี้ประโยคเดียวพอ

คำว่า `exclusive` แปลว่า “ยกเว้น” — คือเป็น OR ธรรมดา**แต่ยกเว้นกรณีที่ 1 ทั้งคู่** OR ธรรมดาถือว่า “1 ทั้งคู่ก็ยังนับ” แต่ XOR ไม่นับครับ นี่คือความต่างเพียงจุดเดียวระหว่าง OR กับ XOR — **ต่างกันแค่แถวสุดท้ายแถวเดียว**

**⑤ ของใกล้ตัว:** เหรียญสองเหรียญในกระเป๋าน้องครับ วางลงโต๊ะพร้อมกันแล้วดู ถ้าออก**ต่างกัน** (เหรียญหนึ่งหัว อีกเหรียญก้อย) นับว่าได้ 1 แต้ม ถ้าออก**เหมือนกัน** (หัวทั้งคู่ หรือก้อยทั้งคู่) ได้ 0 แต้ม — XOR คือ**เครื่องตรวจความต่าง** ครับ ไม่ได้สนใจว่าค่าคืออะไร สนใจแค่ว่า “เหมือนหรือต่าง”




รูป OR + เส้นโค้งเพิ่มอีกเส้นด้านหลัง

① สัญลักษณ์ XOR — เหมือน OR แต่มี**เส้นโค้งซ้อนสองชั้น**ด้านหลัง

| A | B | $X = A \oplus B$ | เทียบกัน |
|---|---|---|---|
| 0 | 0 | 0 | เหมือน |
| 0 | 1 | 1 | ต่าง |
| 1 | 0 | 1 | ต่าง |
| 1 | 1 | 0 | เหมือน |

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

③ เครื่องหมาย $\oplus$ คือบวกในวงกลม อ่านว่า “เอ็กซ์ออร์”

**ที่มาของสูตร $\overline{A}B + A\overline{B}$ — ห้ามท่องลอย ๆ**

สูตรนี้ไม่ได้มาจากไหนเลยครับ มันคือ**การอ่านตารางออกมาเป็นคำพูด**:

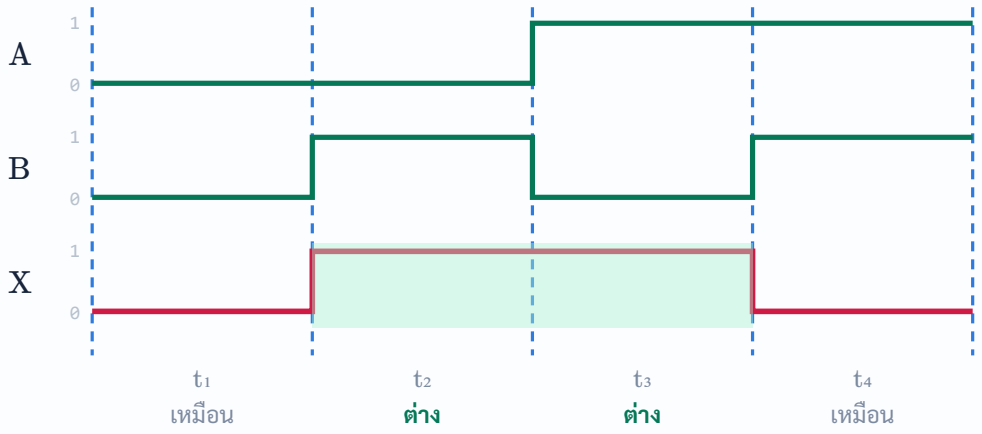
“XOR เป็น 1 ตอนไหน” → ตอบว่า **แถวที่ 2** ($A = 0, B = 1$) **หรือ แถวที่ 3** ($A = 1, B = 0$)

แถวที่ 2 เขียนเป็นสูตรว่า “A เป็น 0 และ B เป็น 1” = $\overline{A}B$

แถวที่ 3 เขียนว่า “A เป็น 1 และ B เป็น 0” = $A\overline{B}$

คำว่า “หรือ” ระหว่างสองแถวคือ $+$ → ได้ $\overline{A}B + A\overline{B}$ พอดีครับ

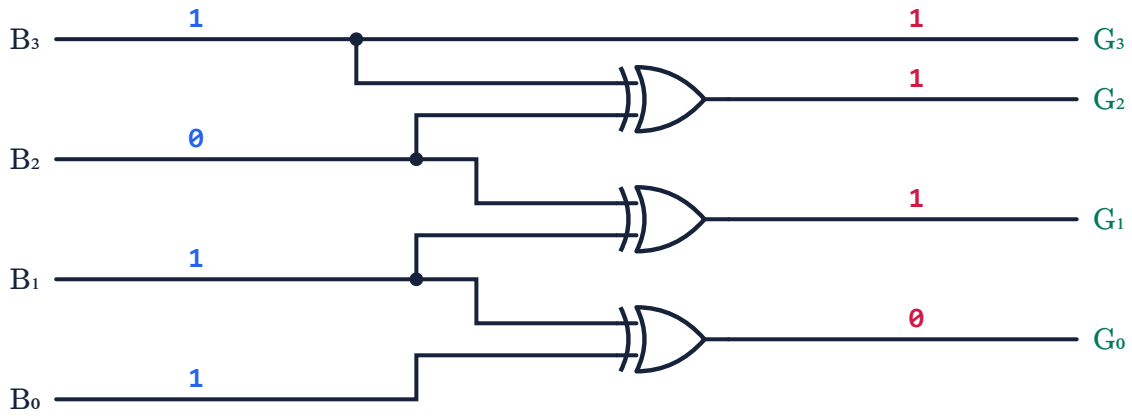
วิธีอ่านตารางเป็นสูตรแบบนี้มีชื่อว่า **SOP (sum-of-products)** ซึ่ง EP07 จะสอนเต็ม ๆ ครับ




④ timing diagram ของ XOR — ดูง่ายที่สุดในบรรดา 7 ตัวครับ **ช่วงไหนที่เส้นเขียวสองเส้นอยู่คนละระดับ เส้นแดงจะขึ้นสูง** (ช่วงเขียว $t_2, t_3$) ช่วงไหนที่เส้นเขียวอยู่ระดับเดียวกัน เส้นแดงจะลงต่ำครับ

**เทียบของเก่า ① — Gray code ของ EP04 คือ XOR ทั้งแถว**

EP04 น้องแปลงเลขฐานสองเป็น Gray code ด้วยกฎ **“บิตซ้ายสุดลอกมาเลย · บิตที่เหลือเอาคู่ที่ติดกันมาเทียบ ต่างเขียน 1 เหมือนเขียน 0”** ประโยค “ต่างเขียน 1 เหมือนเขียน 0” คือ **นิยามของ XOR ตรง ๆ** ครับ เพราะฉะนั้นสูตร Gray code เขียนใหม่ได้ว่า $G_i = B_i \oplus B_{i+1}$



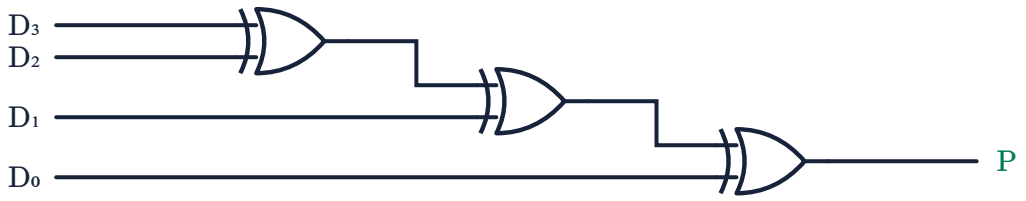

ตัวอย่าง: binary 1011 เข้าไป ออกมาเป็น Gray 1110 — ตรงกับที่คำนวณด้วยมือใน EP04 เป๊ะครับ บิตบนสุดต่อตรงผ่านไปเลย เพราะไม่มีบิตซ้ายให้เทียบด้วย

วงจรแปลง binary → Gray code 4 บิต ใช้ XOR แค่ 3 ตัว — **นี่คือกฎที่น้องทำด้วยมือใน EP04 ถูกทำเป็นวงจรจริง**

**ตรวจให้ดูครับ:** $B = 1011 \rightarrow G_3 = B_3 = 1 \cdot G_2 = 1 \oplus 0 = 1 \cdot G_1 = 0 \oplus 1 = 1 \cdot G_0 = 1 \oplus 1 = 0 \rightarrow G = \mathbf{1110}$ ตรงกับ EP04 ครับ

เทียบของเก่า ② — parity ของ EP04 ก็คือ XOR ต่อกันเป็นสาย

EP04 น้องหาบิตพาริตีคู่ด้วยการ**นับจำนวน 1 ว่าคู่หรือคี่** ลองคิดใหม่ครับ: XOR สองตัวคือ “ต่างกันได้ 1” ซึ่งก็คือ “**มี 1 อยู่จำนวนคี่ตัวหรือเปล่า**” นั่นเอง (0,1 มี 1 หนึ่งตัว = คี่ → ได้ 1 · 1,1 มี 1 สองตัว = คู่ → ได้ 0) เพราะฉะนั้น**เอา XOR มาต่อกันเป็นสายยาว ๆ ก็ได้เครื่องนับคี่** ซึ่งก็คือเครื่องกำเนิดพาริตีครับ




$P = D_3 \oplus D_2 \oplus D_1 \oplus D_0$ — ได้ 1 เมื่อจำนวนบิต 1 เป็นจำนวนคี่
นี่คือบิตพาริตีคู่ (even parity) พอดี เพราะเติมเข้าไปแล้วจำนวน 1 รวมจะกลายเป็นคู่ครับ

เครื่องกำเนิดพาริตี 4 บิตด้วย XOR 3 ตัวต่อกันเป็นสาย — ของที่ EP04 ให้น้องนับด้วยมือ วงจรทำให้ในเสี้ยววินาทีครับ

ลองกับข้อมูล 1011 จาก EP04 ครับ: $1 \oplus 0 = 1, 1 \oplus 1 = 0, 0 \oplus 1 = 1 \rightarrow P = 1$ หมายความว่าข้อมูลชุดนี้มีเลข 1 อยู่ **3 ตัว (จำนวนคี่)** ต้องเติมพาริตี 1 เข้าไปให้ครบคู่ — ตรงกับที่น้องนับด้วยมือใน EP04 เป๊ะครับ

ทางลัด: XOR คือ “สวิตช์ควบคุมการกลับค่า”

ลองมองแบบนี้ครับ ถ้าเรา**ล็อกขาหนึ่งไว้**:
ล็อก $B = 0 \rightarrow A \oplus 0 = A$ (ผ่านตรง ๆ ไม่เปลี่ยน)
ล็อก $B = 1 \rightarrow A \oplus 1 = \overline{A}$ (กลับค่า)
เพราะฉะนั้น **XOR ก็คือ NOT ที่เปิด-ปิดได้** ครับ ขา $B$ ทำหน้าที่เป็นสวิตช์สั่งว่า “จะกลับค่าไหม” ความสามารถข้อนี้คือเหตุผลที่วงจรลบเลขใน EP02 ใช้ XOR สั่งให้ทำ 1's complement เมื่อต้องการครับ

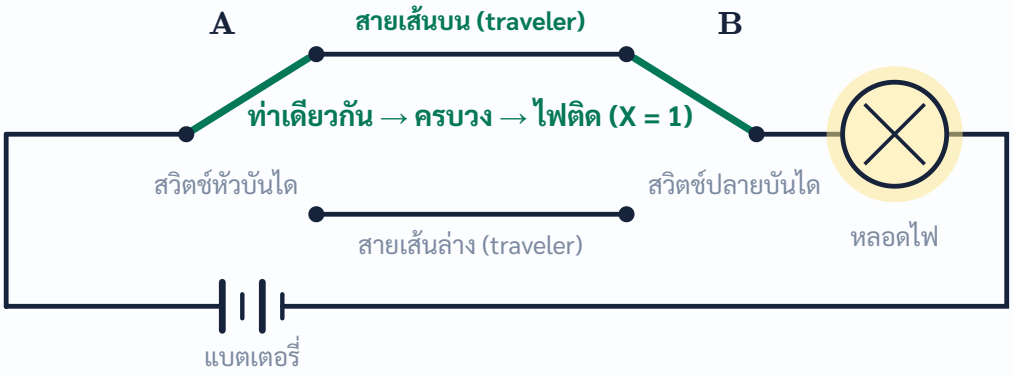
## เกตที่ 7 — XNOR gate (exclusive-NOR)

เข้าเรื่อง

**The XNOR gate performs an exclusive-OR followed by a NOT operation. It produces a TRUE result only when the inputs are the same. If the inputs are different (1,0 or 0,1), the XNOR gate produces a FALSE result.**

แปลเป็นภาษาพี่: กลับด้านกับ XOR เป๊ะครับ — **เหมือนกันได้ 1 · ต่างกันได้ 0** เพราะฉะนั้น XNOR คือ**เครื่องตรวจความเหมือน** หรือเรียกว่า `equality detector` ก็ได้ครับ

⑤ **ของใกล้ตัว:** สวิตช์ไฟบันไดที่มีสวิตช์**สองหัว** — หัวบันไดกับปลายบันไดครับ ลักษณะเด่นของมันคือ **กดตัวไหนก็ได้ ไฟจะสลับสถานะทันที** ไม่ต้องสนใจว่าอีกตัวอยู่ท่าไหน ถ้าเดินสายแบบมาตรฐาน ไฟจะติดตอนที่คันโยกทั้งสอง**อยู่ท่าเดียวกัน** = XNOR ถ้าสลับปลายสายเส้นหนึ่ง ไฟจะติดตอนอยู่**คนละท่า** = XOR ครับ ทั้งสองแบบใช้งานได้จริงทั้งคู่




สวิตช์ไฟบันไดสองหัว — คันโยกทั้งคู่ชี้ไปที่สายเส้นเดียวกัน วงจรครบ ไฟติด ถ้าโยกตัวใดตัวหนึ่ง คันโยกจะไปคนละสาย วงจรขาด ไฟดับทันทีครับ



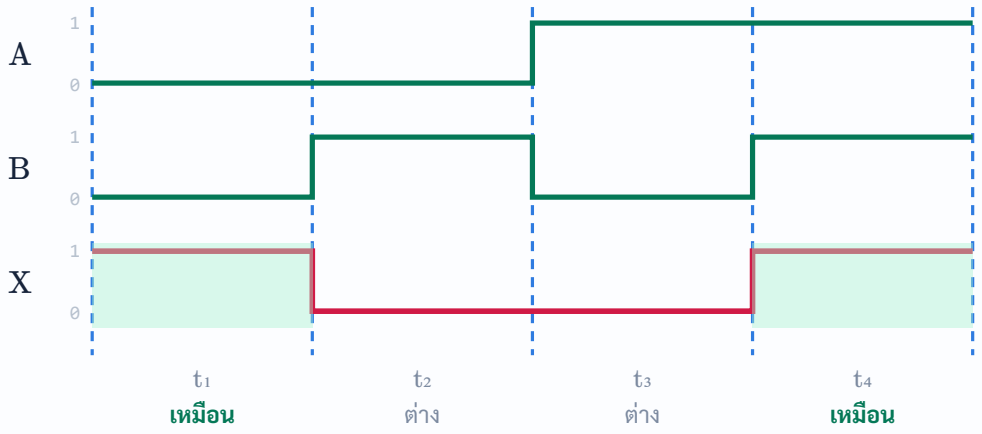

XOR + bubble ที่ขาออก

① สัญลักษณ์ XNOR = รูป XOR + bubble ที่ขาออก

| A | B | $A \oplus B$ | $X$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

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

③ บางตำราเขียน $A \odot B$ (จุดในวงกลม) ก็ตัวเดียวกันครับ




④ timing diagram ของ XNOR — เอาเส้นแดงของ XOR มาพลิกกลับหัวได้เลยครับ ขึ้นสูงในช่วงที่เส้นเขียวสองเส้นอยู่ระดับเดียวกัน คือ $t_1$ กับ $t_4$

**กับดักใหญ่ของ XOR — อย่าสับสนกับ OR**

OR กับ XOR ต่างกัน**แถวเดียว**ครับ คือแถวสุดท้าย $(1, 1)$

$1 + 1 = \mathbf{1}$ (OR) แต่ $1 \oplus 1 = \mathbf{0}$ (XOR)

เวลาทำข้อสอบเร็ว ๆ คนมักลอกตาราง OR มาใส่ช่อง XOR แล้วลืมแก้แถวล่างสุด — **เสียคะแนนฟรี**

วิธีกันพลาด: หลังเขียนตาราง XOR เสร็จ ให้**นับจำนวน 1 ในคอลัมน์เอาต์พุต ต้องได้ 2 ตัวพอดี** ถ้าได้ 3 ตัว แปลว่าน้องเขียนตาราง OR ไปครับ และตาราง XNOR ก็ต้องได้ 1 จำนวน 2 ตัวเหมือนกัน แต่คนละตำแหน่ง

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| exclusive-OR (XOR) | ออร์เฉพาะ (ต่างกันได้ 1) | produces a TRUE result only when the inputs differ |
| exclusive-NOR (XNOR) | ตรงข้ามกับ XOR (เหมือนกันได้ 1) | only when the inputs are the same |
| the inputs differ | ขาเข้าต่างกัน | when the inputs differ |
| parity generator | วงจรสร้างบิตพาริตี | an XOR-based parity generator |
| comparator / equality detector | วงจรเทียบว่าเท่ากันไหม | used as an equality detector |

# 6 เกตตัวเดียว แต่วาดได้สองแบบ

**ตะขอเกี่ยว**

ตอนนี้น้องรู้จักเกตครบ 7 ตัวแล้วครับ และรู้กฎว่า **bubble = กลับค่า** ที่ผ่านมา bubble อยู่ที่**ขาออก**ตลอด (NAND, NOR, NOT)

สมมติว่าถ้าพี่ย้าย bubble ไปไว้ที่**ขาเข้า**แทนละครับ จะเกิดอะไรขึ้น คำตอบจะทำให้น้องประหลาดใจ: **มันกลายเป็นเกตที่น้องรู้จักอยู่แล้ว** แค่คนละหน้าตา หัวข้อนี้คือส่วนที่ยากที่สุดของ EP นี้ครับ แต่เป็นส่วนที่ Assignment 4 ข้อ 6 จะใช้เต็ม ๆ ถ้าเข้าใจตรงนี้ EP08 จะง่ายไปครึ่งวิชาเลย

## เกตหน้าตาใหม่ 2 ตัว: negative-AND และ negative-OR

**เข้าเรื่อง**

**Negative-AND gate** = รูป **AND** ที่มี **bubble ที่ขาเข้าทุกขา** → กลับค่าขาเข้าก่อน แล้วค่อย AND → $X = \overline{A} \cdot \overline{B}$

**Negative-OR gate** = รูป **OR** ที่มี **bubble ที่ขาเข้าทุกขา** → กลับค่าขาเข้าก่อน แล้วค่อย OR → $X = \overline{A} + \overline{B}$

**จุดสำคัญ:** bubble อยู่ที่ขาเข้า แปลว่ากลับค่า**ก่อน**เข้าเกต ต่างจาก NAND/NOR ที่กลับค่า**หลัง**ออกจากเกตครับ




**Negative-AND gate**

รูป AND + bubble ที่ขาเข้าทั้งสองขา

$$X = \overline{A} \cdot \overline{B}$$

ได้ 1 ก็ต่อเมื่อขาเข้า 0 ทั้งคู่




**Negative-OR gate**

รูป OR + bubble ที่ขาเข้าทั้งสองขา

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

ได้ 1 ถ้ามีขาเข้า 0 อย่างน้อยหนึ่งขา

สองตัวนี้คือ “เกตหน้าใหม่” ที่ bubble ย้ายจากขาออกไปอยู่ขาเข้า — ขีดบนอยู่บน $A$ กับ $B$ แยกกัน ไม่ได้คลุมทั้งก้อนแล้วครับ

**Negative-AND**

| A | B | $\overline{A}$ | $\overline{B}$ | $X = \overline{A}\,\overline{B}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |

**Negative-OR**

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

| Negative-AND | | | | |
|---|---|---|---|---|
| A | B | $\overline{A}$ | $\overline{B}$ | $X = \overline{A}\,\overline{B}$ |
| 1 | 1 | 0 | 0 | 0 |

| Negative-OR | | | | |
|---|---|---|---|---|
| A | B | $\overline{A}$ | $\overline{B}$ | $X = \overline{A} + \overline{B}$ |
| 1 | 1 | 0 | 0 | 0 |

ทีนี้ให้น้องดูคอลัมน์ขวาสุดของสองตารางนี้ให้ดี ๆ ครับ แล้วเลื่อนสายตากลับไปดูตาราง NOR กับ NAND ในหัวข้อ 4 **คุ้น ๆ ไหมครับ**

## พิสูจน์ด้วยตารางค่าความจริงวางคู่กัน — คู่ที่ 1: NAND = negative-OR

พี่จะไม่บอกให้น้องเชื่อครับ พี่จะ**วางตารางคู่กันแล้วไล่ทีละแถวให้ดู** ถ้าคอลัมน์เอาต์พุตทั้งสองตรงกันครบทุกแถว แปลว่ามันคือเกตตัวเดียวกันจริง ๆ ไม่มีทางแยกออกจากกันได้เลย

| ขาเข้า | | ทาง NAND | | ทาง negative-OR | | | ตรงกันไหม |
|---|---|---|---|---|---|---|---|
| A | B | $AB$ | $\overline{AB}$ | $\overline{A}$ | $\overline{B}$ | $\overline{A} + \overline{B}$ | — |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | ✓ ตรง |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | ✓ ตรง |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | ✓ ตรง |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | ✓ ตรง |

**ไล่ให้ดูทีละแถว**

**แถว 1 —** $A = 0, B = 0$ ทาง NAND: $AB = 0 \cdot 0 = 0$ แล้วกลับค่าได้ $\mathbf{1}$ · ทาง negative-OR: $\overline{A} = 1, \overline{B} = 1$ แล้ว OR กันได้ $1 + 1 = \mathbf{1}$ **— ตรงกัน**

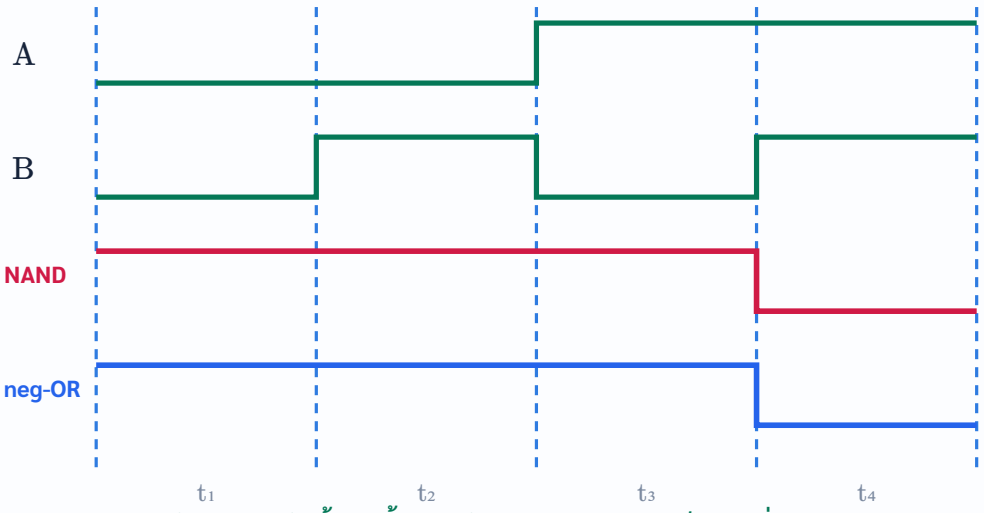
**แถว 2 —** $A = 0, B = 1$ ทาง NAND: $AB = 0$ กลับค่าได้ $\mathbf{1}$ · ทาง negative-OR: $\overline{A} = 1, \overline{B} = 0$ แล้ว $1 + 0 = \mathbf{1}$ **— ตรงกัน** (แถวนี้สำคัญ เพราะทางหนึ่งได้ 1 จาก "$AB$ เป็น 0" อีกทางได้ 1 จาก "$\overline{A}$ เป็น 1" คนละเหตุผลแต่ผลเท่ากันครับ)

**แถว 3 —** $A = 1, B = 0$ ทาง NAND: $AB = 0$ กลับค่าได้ $\mathbf{1}$ · ทาง negative-OR: $\overline{A} = 0, \overline{B} = 1$ แล้ว $0 + 1 = \mathbf{1}$ **— ตรงกัน**

**แถว 4 —** $A = 1, B = 1$ ทาง NAND: $AB = 1$ กลับค่าได้ $\mathbf{0}$ · ทาง negative-OR: $\overline{A} = 0, \overline{B} = 0$ แล้ว $0 + 0 = \mathbf{0}$ **— ตรงกัน**

**สรุป: ตรงกันครบ 4 แถวจาก 4 แถว →** $\overline{AB} = \overline{A} + \overline{B}$ **จริง**

*"The NAND gate and the negative-OR gate have identical truth tables; they are two symbols for the same gate."*




**เส้นแดงกับเส้นน้ำเงินขึ้น-ลงพร้อมกันทุกจังหวะ — เป็นรูปคลื่นเดียวกันเป๊ะ**

พิสูจน์ซ้ำด้วย timing diagram — วัดด้วยออสซิลโลสโคปจริง ๆ ก็แยกไม่ออกครับ ว่าตัวไหนคือ NAND ตัวไหนคือ negative-OR

## คู่ที่ 2: NOR = negative-AND

| ขาเข้า | | ทาง NOR | | ทาง negative-AND | | | ตรงกันไหม |
|---|---|---|---|---|---|---|---|
| A | B | $A+B$ | $\overline{A+B}$ | $\overline{A}$ | $\overline{B}$ | $\overline{A}\,\overline{B}$ | — |
| 0 | 0 | 0 | **1** | 1 | 1 | **1** | ✓ ตรง |
| 0 | 1 | 1 | **0** | 1 | 0 | **0** | ✓ ตรง |
| 1 | 0 | 1 | **0** | 0 | 1 | **0** | ✓ ตรง |
| 1 | 1 | 1 | **0** | 0 | 0 | **0** | ✓ ตรง |

**ไล่ให้ดูทีละแถว**

**แถว 1 —** $A=0, B=0$ ทาง NOR: $A+B=0+0=0$ กลับค่าได้ $\mathbf{1}$ · ทาง negative-AND: $\overline{A}=1, \overline{B}=1$ แล้ว $1\cdot 1=\mathbf{1}$ **— ตรงกัน**

**แถว 2 —** $A=0, B=1$ ทาง NOR: $A+B=1$ กลับค่าได้ $\mathbf{0}$ · ทาง negative-AND: $\overline{A}=1, \overline{B}=0$ แล้ว $1\cdot 0=\mathbf{0}$ **— ตรงกัน** (AND ต้องครบทั้งคู่ พอมีตัวหนึ่งเป็น 0 ก็จบเลยครับ)

**แถว 3 —** $A=1, B=0$ ทาง NOR: $A+B=1$ กลับค่าได้ $\mathbf{0}$ · ทาง negative-AND: $\overline{A}=0, \overline{B}=1$ แล้ว $0\cdot 1=\mathbf{0}$ **— ตรงกัน**

**แถว 4 —** $A=1, B=1$ ทาง NOR: $A+B=1$ กลับค่าได้ $\mathbf{0}$ · ทาง negative-AND: $\overline{A}=0, \overline{B}=0$ แล้ว $0\cdot 0=\mathbf{0}$ **— ตรงกัน**

**สรุป: ตรงกันครบ 4 แถวจาก 4 แถว →** $\overline{A+B}=\overline{A}\cdot\overline{B}$ **จริง**

*“The NOR gate and the negative-AND gate are dual symbols for the same gate.”*

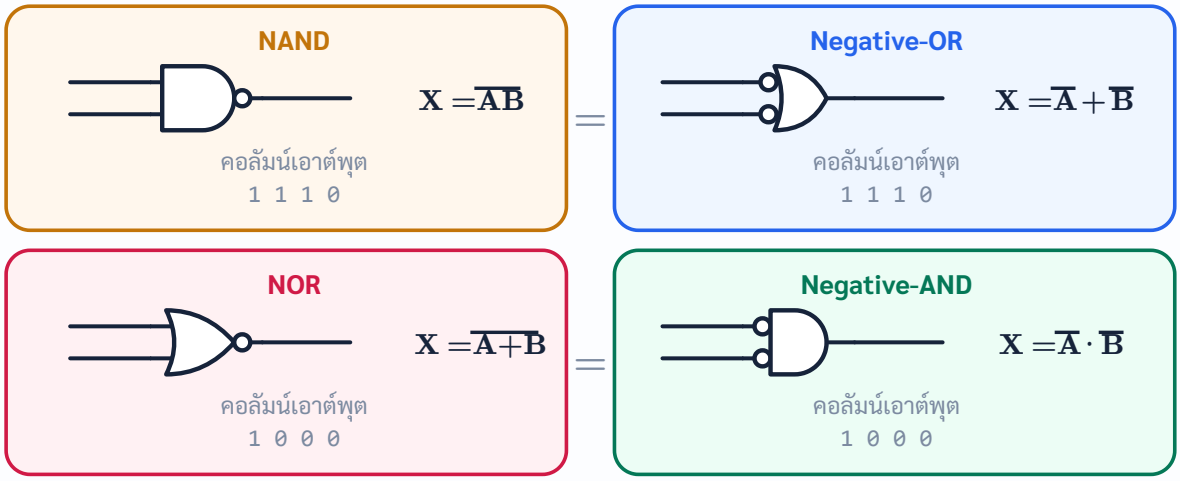
กฎย้าย bubble — จำแค่ประโยคเดียว

**“ย้าย bubble ข้ามตัวเกต ต้องสลับรูป AND ↔ OR ทุกครั้ง”**

NAND (AND + bubble ที่ขาออก) → ย้าย bubble ไปขาเข้า → ต้องเปลี่ยนรูปเป็น OR → ได้ negative-OR

NOR (OR + bubble ที่ขาออก) → ย้าย bubble ไปขาเข้า → ต้องเปลี่ยนรูปเป็น AND → ได้ negative-AND

**ถ้าลืมสลับรูป จะได้เกตผิดตัวทันที** — ย้าย bubble ของ NAND โดยไม่สลับรูป จะได้ negative-AND ซึ่งคือ NOR ครับ ผิดสนิท




คู่ dual symbol ทั้งสองคู่ (ตรงกับสไลด์หน้า 16) — **บนกระดาษวาดคนละรูป แต่ในกล่องชิปคือขาเดียวกัน**

## แล้วทำไมต้องมีสองสัญลักษณ์สำหรับเกตตัวเดียว

คำถามนี้คือหัวใจของหัวข้อครับ ถ้ามันเหมือนกันทุกอย่าง ทำไมไม่ใช้รูปเดียวไปเลยให้จบ คำตอบคือ **สัญลักษณ์ไม่ได้มีไว้บอกแค่ว่าชิปทำอะไร แต่มีไว้บอกว่า “ผู้ออกแบบคิดอะไรอยู่”** ครับ

**คนออกแบบคนที่ 1 คิดแบบ active-HIGH**

เขาคิดว่า “**ฉันสนใจตอนที่ขาเข้าทั้งคู่เป็น 1** แล้วอยากให้เอาต์พุตลงต่ำเป็นสัญญาณเตือน”

→ เขาจะวาด **NAND** เพราะรูป AND สื่อคำว่า “และ” และ bubble ที่ขาออกสื่อว่า “ผลลัพธ์ที่ฉันสนใจคือตอนมันเป็น 0”

**คนออกแบบคนที่ 2 คิดแบบ active-LOW**

เขาคิดว่า “**ฉันสนใจตอนที่มีขาเข้าเป็น 0 อย่างน้อยหนึ่งขา** แล้วอยากให้เอาต์พุตขึ้นสูง”

→ เขาจะวาด **negative-OR** เพราะรูป OR สื่อคำว่า “หรือ” และ bubble ที่ขาเข้าสื่อว่า “ขาที่ฉันสนใจคือตอนมันเป็น 0”

**ทั้งสองคนซื้อชิปเบอร์เดียวกัน เสียบขาเดียวกัน วงจรทำงานเหมือนกันเป๊ะ** แต่เขียนคนละรูป เพื่อให้คนที่มาอ่านวงจรทีหลัง**เข้าใจเจตนาได้ทันทีโดยไม่ต้องไล่ตาราง**ครับ นี่คือเหตุผลเดียวที่ dual symbol มีอยู่ — เพื่อการสื่อสาร ไม่ใช่เพื่อการทำงาน

**คำที่ต้องรู้: active-HIGH กับ active-LOW**

`Active-HIGH` = สัญญาณนี้ “ทำงาน / มีความหมาย” ตอนเป็น `1` → ในรูปวงจร**ไม่มี bubble**
`Active-LOW` = สัญญาณนี้ “ทำงาน / มีความหมาย” ตอนเป็น `0` → ในรูปวงจร**มี bubble**
**กฎอ่านวงจรที่ใช้ได้ตลอดชีวิต: เห็น bubble ที่ไหน แปลว่าจุดนั้น active-LOW ครับ** เรื่องนี้ไม่ได้อยู่แค่ในเกต — ขา reset ของไมโครคอนโทรลเลอร์ที่เขียนว่า RESET มีขีดบน ก็คือ active-LOW แบบเดียวกันครับ

**กับดักที่ทำให้ตอบผิดทั้งข้อ**

คนพลาดมากที่สุดคือ **เห็นรูป OR แล้วรีบตอบว่าเป็น OR** ครับ ทั้งที่มัน**มี bubble ที่ขาเข้า** ซึ่งทำให้มันเป็น negative-OR = NAND ไม่ใช่ OR เลยสักนิด
ตาราง OR: $0, 1, 1, 1$ · ตาราง negative-OR: $1, 1, 1, 0$ — **คนละอย่างสิ้นเชิงครับ**
**วิธีกันพลาด:** ก่อนตอบทุกครั้ง ให้**นับ bubble ในรูปก่อน** ① ไม่มี bubble เลย → เกตธรรมดา ② bubble ที่ขาออก → เกตตระกูล N (NAND/NOR) ③ bubble ที่ขาเข้า → เกตตระกูล negative แล้วค่อยแปลงกลับด้วยกฎย้าย bubble ครับ

**เรื่องนี้จะกลับมาอีกแน่นอน**

① **EP06** จะบอกชื่อจริงของสองสมการที่เราเพิ่งพิสูจน์ไปครับ — $\overline{AB} = \overline{A} + \overline{B}$ และ $\overline{A + B} = \overline{A}\,\overline{B}$ มันมีชื่อว่า `De Morgan's theorems` ตอนนี้น้อง**พิสูจน์ด้วยตารางไปแล้ว** เพราะฉะนั้น EP06 น้องแค่จำชื่อเพิ่มอย่างเดียวครับ
② **Assignment 4 ข้อ 6** จะให้วงจรมาแล้วสั่งให้**วาดใหม่ด้วย dual symbol** — งานคือย้าย bubble แล้วสลับรูป AND ↔ OR ตามกฎในหัวข้อนี้เป๊ะ ๆ ครับ
③ **EP08** จะใช้ dual symbol ช่วยอ่านวงจร NAND ล้วนให้กลายเป็นวงจร AND-OR ที่อ่านง่ายครับ

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| negative-AND gate | เกต AND ที่มี bubble ขาเข้า | Negative AND gate symbol |
| negative-OR gate | เกต OR ที่มี bubble ขาเข้า | Negative OR gate symbol |
| dual symbol | สัญลักษณ์คู่ของเกตเดียวกัน | Draw the dual symbol of each gate |
| active-HIGH | ทำงานตอนสัญญาณเป็น 1 | an active-HIGH output |
| active-LOW | ทำงานตอนสัญญาณเป็น 0 | an active-LOW input |
| identical truth tables | ตารางค่าความจริงเหมือนกันทุกแถว | the two gates have identical truth tables |

# 7 ทั้ง 7 เกตในหน้าเดียว

**ตะขอเกี่ยว**

น้องรู้จักเกตครบ 7 ตัวแล้ว บวก dual symbol อีก 2 หน้าตา สมมติว่าถ้าพรุ่งนี้สอบแล้วน้องมีเวลาอ่านทวนแค่ **1 หน้า** ละครับ จะเลือกหน้าไหน หน้านี้แหละครับ — พี่ยัดทุกอย่างที่ต้องจำลงหน้าเดียว ตรงกับสไลด์อาจารย์หน้า 17 พอดี

## ตารางค่าความจริงรวมทั้ง 7 เกต

ตารางนี้อ่านตามคอลัมน์ครับ — เลือกเกตที่อยากรู้ แล้วไล่ลงมา 4 ช่อง **เขียว = 1** **แดง = 0**

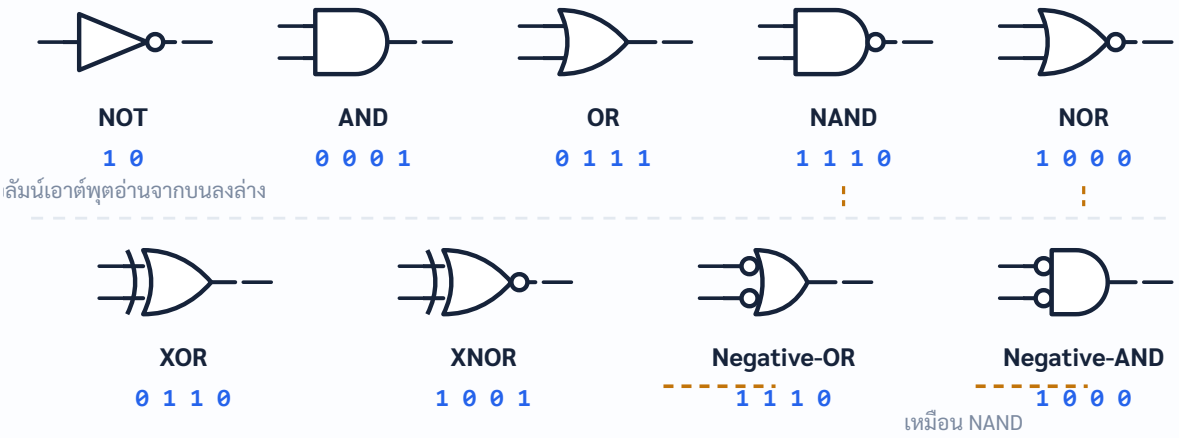
| A | B | NOT $\overline{A}$ | AND $AB$ | OR $A+B$ | NAND $\overline{AB}$ | NOR $\overline{A+B}$ | XOR $A \oplus B$ | XNOR $\overline{A \oplus B}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

**อ่านตารางนี้ยังไงให้เร็ว**

คอลัมน์ NOT ใช้แค่ค่า $A$ ครับ (ไม่สนใจ $B$) เพราะ NOT มีขาเข้าขาเดียว พี่เอามาวางไว้ในตารางเดียวกันเพื่อให้เทียบง่าย

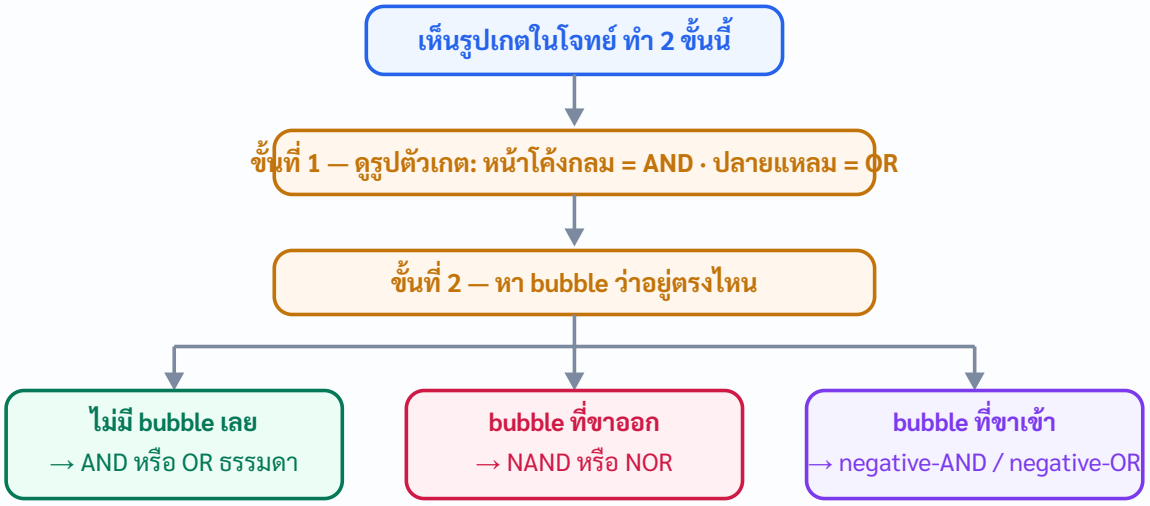
ส่วน **negative-OR ให้ลอกคอลัมน์ NAND** และ **negative-AND ให้ลอกคอลัมน์ NOR** ได้เลยครับ เพราะหัวข้อ 6 พิสูจน์ไปแล้วว่าตรงกันทุกแถว

## สัญลักษณ์ทั้ง 9 หน้าตา




แถวบน = 5 เกตพื้นฐาน · แถวล่าง = XOR, XNOR และ dual symbol อีก 2 ตัว — **เส้นประสีส้มโยง negative-OR กลับไป NAND และ negative-AND กลับไป NOR** เพราะเป็นเกตเดียวกันครับ

## แผนที่ตัดสินใจ: เห็นรูปแบบนี้ = เกตอะไร




ใช้แผนที่นี้ก่อนตอบทุกครั้ง — **ดูรูป แล้วดู bubble** สองขั้นเท่านั้น ไม่ต้องเดา (ส่วน XOR/XNOR สังเกตง่ายอยู่แล้วเพราะมีเส้นโค้งซ้อนสองชั้นด้านหลังครับ)

## Opposing logic — 3 คู่ที่เป็นขั้วตรงข้ามกัน

สไลด์หน้า 15 สรุปไว้สั้น ๆ ว่ามี **3 คู่** ที่คอลัมน์เอาต์พุต**กลับกันทุกช่อง**ครับ คู่พวกนี้ต่างจาก dual symbol นะครับ — dual symbol คือ**เหมือนกัน** ส่วน opposing logic คือ**ตรงข้ามกัน**

| คู่ตรงข้าม | เอาต์พุต | ความสัมพันธ์ |
|---|---|---|
| **AND ↔ NAND** | `0001` ↔ `1110` | $\overline{AB}$ คือ $AB$ ที่ถูกกลับค่าทุกช่อง — ช่องไหนเดิมเป็น 0 กลายเป็น 1 หมดครับ |

| คู่ตรงข้าม | เอาต์พุต | ความสัมพันธ์ |
|---|---|---|
| **OR ↔ NOR** | `0111` ↔ `1000` | $\overline{A+B}$ คือ $A+B$ ที่ถูกกลับค่าทุกช่อง |
| **XOR ↔ XNOR** | `0110` ↔ `1001` | $\overline{A \oplus B}$ คือ $A \oplus B$ ที่ถูกกลับค่าทุกช่อง — “ต่างได้ 1” กลายเป็น “เหมือนได้ 1” |

**อย่าสับสนสองคำนี้**

**Dual symbol** = **เกตเดียวกัน** เขียนคนละรูป → ตารางค่าความจริง**ตรงกันทุกแถว** (NAND กับ negative-OR · NOR กับ negative-AND)

**Opposing logic** = **คนละเกต** ที่เป็นขั้วตรงข้าม → ตารางค่าความจริง**กลับกันทุกแถว** (AND กับ NAND · OR กับ NOR · XOR กับ XNOR)

คำสองคำนี้อยู่ในสไลด์ติดกัน (หน้า 15 กับ 16) และข้อสอบชอบเอามาถามสลับกันครับ **ตรงกันทุกแถว = dual · กลับกันทุกแถว = opposing**

**ทางลัด: จำแค่ 4 ตัว ได้ครบ 7 ตัว**

น้องไม่ต้องท่องตาราง 7 ตารางครับ ท่องแค่ **4 ตัว** พอ:

**NOT** → กลับค่า · **AND** → 1 แถวเดียว แถวล่างสุด · **OR** → 0 แถวเดียว แถวบนสุด · **XOR** → ต่างได้ 1 เหมือนได้ 0

แล้วอีก 3 ตัวที่เหลือ (NAND, NOR, XNOR) ได้จากการ**กลับค่าทั้งคอลัมน์**ของ AND, OR, XOR ตามลำดับครับ ใช้เวลาไม่ถึง 5 วินาทีต่อตัว

## ศัพท์สอบของหัวข้อนี้

| English | แปลว่า | ข้อสอบเขียนมาแบบนี้ |
|---|---|---|
| opposing logic | ลอจิกที่เป็นขั้วตรงข้ามกัน | Opposing logic: AND and NAND |
| summary of truth tables | ตารางสรุปค่าความจริง | Summary of Truth Tables |
| complement of the output | ค่าตรงข้ามของเอาต์พุต | the complement of the output column |

# 8 เกตพวกนี้เอาไปใช้ทำอะไรจริง ๆ

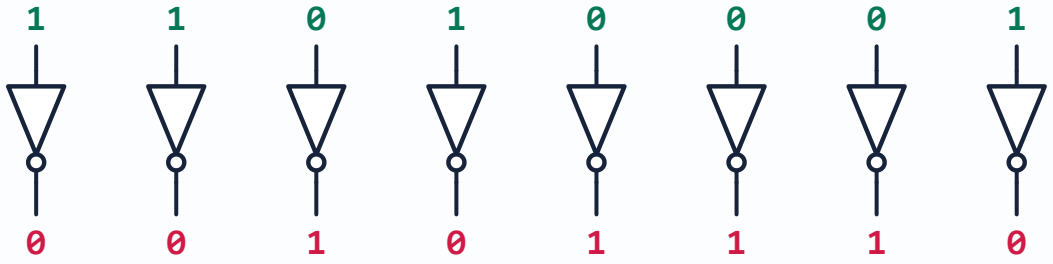
**ตะขอเกี่ยว**

น้องอาจสงสัยว่าเรียนเกตไปทำไม ในเมื่อมันทำได้แค่ตัดสิน 0 กับ 1 สมมติว่าพี่บอกว่า **ทั้งสามตัวอย่างข้างล่างนี้คือของที่ใช้จริงในเครื่องจริง** ละครับ และตัวอย่างแรก**คือสิ่งที่น้องคำนวณด้วยมือมาตั้งแต่ EP02** — วันนี้น้องจะได้เห็นว่าวงจรมันทำงานยังไง

## ตัวอย่างที่ 1 — วงจรหา 1's complement ด้วย NOT ล้วน

EP02 น้องหา 1's complement ด้วยการ**กลับบิตทุกตัว** ใช่ไหมครับ ทีนี้ถ้าจะให้เครื่องทำ ก็ง่ายมาก — **เอา NOT ไปแปะทุกเส้น** เส้นละตัว จบครับ 8 บิตก็ใช้ NOT 8 ตัว ทำงานพร้อมกันทั้ง 8 เส้นในจังหวะเดียว

**Binary number input — เลขฐานสองที่ป้อนเข้า**




**1's complement output — ผลลัพธ์ที่กลับบิตแล้ว**

Using multiple NOT gates to obtain 1's complement of a binary number

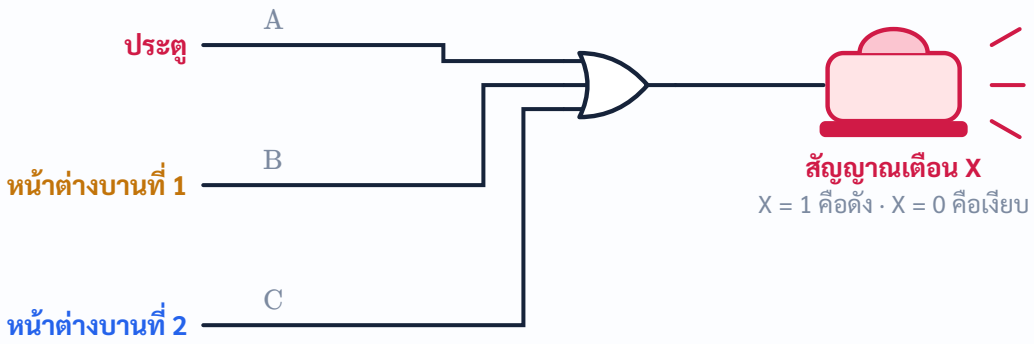
อินเวอร์เตอร์ 8 ตัวเรียงกัน — ป้อน `11010001` เข้าไป ได้ `00101110` ออกมาทันที ตรงกับที่น้องคิดด้วยมือใน EP02 ครับ

**เทียบของเก่า — EP02 กลับมาแล้ว**

EP02 น้องเรียนว่า 2's complement = 1's complement แล้ว **+1**
วงจรข้างบนทำส่วนแรกให้แล้วครับ ส่วน “+1” ต้องใช้วงจรบวก ซึ่งประกอบจาก XOR กับ AND (จำทางลัดในหัวข้อ 5 ได้ไหมครับ — **ผลบวกของ 1 บิตคือ XOR ส่วนตัวทดคือ AND**)
เพราะฉะนั้นทุกอย่างที่น้องคิดด้วยมือใน EP01–EP04 **สร้างจากเกต 7 ตัวนี้ได้หมด**ครับ

## ตัวอย่างที่ 2 — ระบบตรวจจับการบุกรุกด้วย OR gate

บ้านหลังหนึ่งมีเซนเซอร์ 3 จุด: ที่ประตู · ที่หน้าต่างบานที่ 1 · ที่หน้าต่างบานที่ 2 กติกาคือ **ปิดอยู่ให้ค่า 0 · เปิดออกให้ค่า 1** แล้วเราต้องการให้สัญญาณเตือนดัง**ถ้ามีจุดใดจุดหนึ่งถูกเปิด** — คำว่า “จุดใดจุดหนึ่ง” คือ OR เป๊ะครับ




X = A + B + C — เซนเซอร์ตัวไหนขึ้นเป็น 1 ก็พอ สัญญาณเตือนดังทันที
Intrusion detection system using OR gate

ระบบเตือนภัยด้วย OR 3 ขา — ปิดครบทุกจุดเท่านั้นถึงจะเงียบ ตรงกับ “OR ให้ 0 แถวเดียว คือแถวที่ 0 หมด” ครับ

| A<br>ประตู | B<br>หน้าต่าง 1 | C<br>หน้าต่าง 2 | X<br>เตือน |
|---|---|---|---|
| 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 |

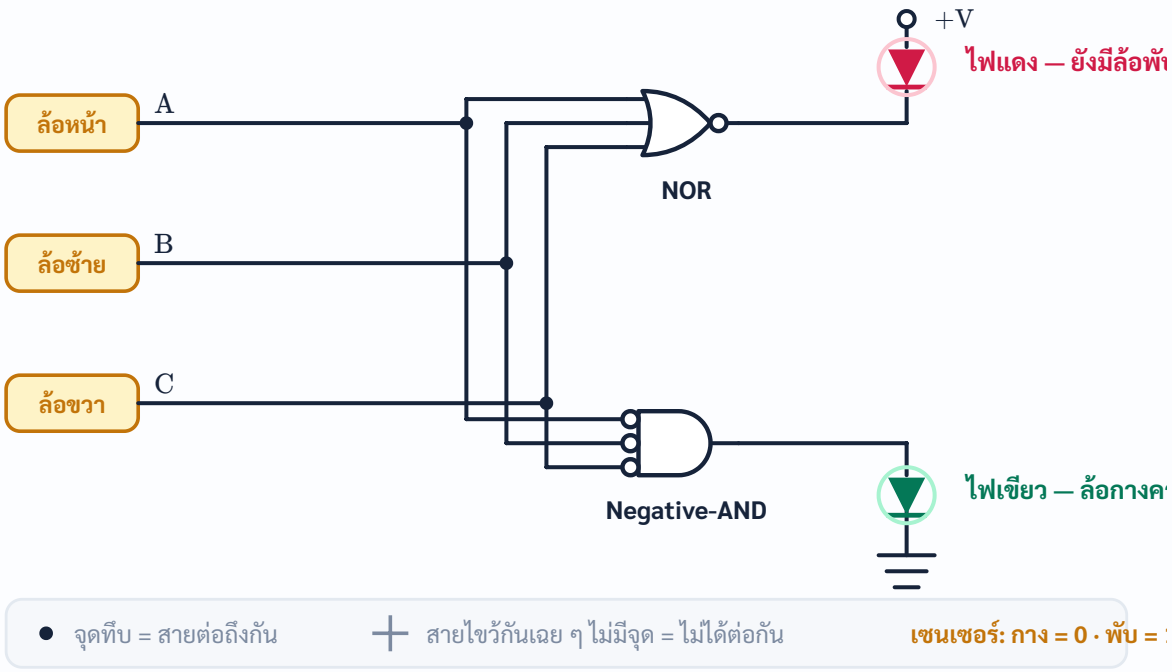
**ทำไมต้องออกแบบให้ “เปิด = 1”**

เพราะเราอยากให้**เหตุการณ์ที่ต้องระวังเป็นค่า 1** ครับ แล้วเอา OR มารวมทุกเหตุการณ์เข้าด้วยกัน จะได้วงจรที่**ตื่นตัวต่อทุกอย่าง**

ถ้าออกแบบกลับกัน (ปิด = 1) ก็ยังทำได้ แต่ต้องเปลี่ยนไปใช้ AND แทน และตอนอ่านวงจรจะงงกว่า — นี่คือเรื่อง active-HIGH / active-LOW จากหัวข้อ 6 ในชีวิตจริงครับ

## ตัวอย่างที่ 3 — ระบบเฝ้าล้อเครื่องบิน ใช้ NOR คู่กับ Negative-AND

ตัวอย่างนี้เป็นตัวอย่างที่**สำคัญที่สุด**ในสไลด์ครับ เพราะมันคือ dual symbol จากหัวข้อ 6 ที่ถูกใช้จริง เครื่องบินมีเซนเซอร์วัดตำแหน่งล้อ 3 จุด กติกาคือ **ล้อกางออกแล้ว (extended) ให้ค่า LOW = 0** และ **ล้อยังพับอยู่ (retracted) ให้ค่า HIGH = 1** แล้วเราต้องการไฟ 2 ดวง: **ไฟแดง** บอกว่ายังมีล้อพับอยู่ · **ไฟเขียว** บอกว่ากางครบทุกล้อแล้ว




ระบบเฝ้าล้อเครื่องบิน (สไลด์หน้า 20) — **เกตสองตัวนี้คือชิปแบบเดียวกัน** แต่วาดคนละสัญลักษณ์ เพื่อสื่อเจตนาของไฟแต่ละดวง ครับ

**อ่านวงจรนี้ให้ออก — นี่คือหัวใจของ dual symbol**

**ตัวบน (NOR):** ผู้ออกแบบคิดว่า “**ถ้ามีล้อไหนพับอยู่ (ขาเข้าเป็น 1) ให้เอาต์พุตลงต่ำ**” เอาต์พุตที่ลงต่ำจะดึงกระแสจาก $+V$ ผ่านไฟแดงลงมา → **ไฟแดงติด** เขาเลยวาดเป็น NOR: รูป OR สื่อว่า “มีตัวไหนก็ได้” และ bubble ที่ขาออกสื่อว่า “ผลที่สนใจคือขาออกเป็น 0” ครับ

**ตัวล่าง (Negative-AND):** ผู้ออกแบบคิดว่า “**ถ้าล้อกางครบทุกล้อ (ขาเข้าเป็น 0 หมด) ให้เอาต์พุตขึ้นสูง**” เอาต์พุตที่ขึ้นสูงจะดันกระแสผ่านไฟเขียวลงกราวด์ → **ไฟเขียวติด** เขาเลยวาดเป็น negative-AND: bubble ที่ขาเข้าสื่อว่า “ขาที่สนใจคือตอนเป็น 0” และรูป AND สื่อว่า “ต้องครบทุกขา” ครับ

**แต่ถ้าไปเปิดกล่องดู จะพบว่าทั้งสองคือชิปเบอร์เดียวกัน** เพราะหัวข้อ 6 พิสูจน์แล้วว่า NOR = negative-AND ทุกแถว — ผู้ออกแบบแค่**เลือกสัญลักษณ์ที่ตรงกับความคิดของตัวเอง**เพื่อให้คนอ่านวงจรเข้าใจง่าย ครับ

| A | B | C | ไฟแดง | ไฟเขียว |
|---|---|---|---|---|
| 0 | 0 | 0 | ดับ | ติด |
| 0 | 0 | 1 | ติด | ดับ |
| 0 | 1 | 0 | ติด | ดับ |
| 0 | 1 | 1 | ติด | ดับ |
| 1 | 0 | 0 | ติด | ดับ |
| 1 | 0 | 1 | ติด | ดับ |
| 1 | 1 | 0 | ติด | ดับ |
| 1 | 1 | 1 | ติด | ดับ |

**กับดักของวงจรนี้**

คนมักคิดว่า “ไฟแดงกับไฟเขียวมาจากเกตคนละตัว เพราะเห็นวาดสองรูป”
**ผิดครับ** — ทั้งสองรูปคือเกตชนิดเดียวกัน (NOR 3 ขา) และเอาต์พุตของทั้งคู่ก็**เท่ากันทุกแถว**ด้วย
ที่ไฟติดคนละดวงเป็นเพราะ**วิธีต่อ LED ต่างกัน**: ไฟแดงต่อจาก $+V$ ลงมา (ติดตอนขาออกเป็น 0) ส่วนไฟเขียวต่อลงกราวด์ (ติดตอนขาออกเป็น 1) — **เกตเดียวกัน แต่ LED อ่านคนละขั้ว**ครับ

**ทำไมสไลด์ถึงเอาตัวอย่างนี้มาปิดท้าย**

เพราะมันรวมทุกอย่างของ EP นี้ไว้ในรูปเดียวครับ — เกตหลายขาเข้า · bubble · active-HIGH vs active-LOW · และ dual symbol ที่พิสูจน์ด้วยตารางค่าความจริง
ถ้าน้องอ่านวงจรนี้ออกทั้งวง แปลว่าน้อง**ผ่าน EP05 แล้วจริง ๆ** ครับ

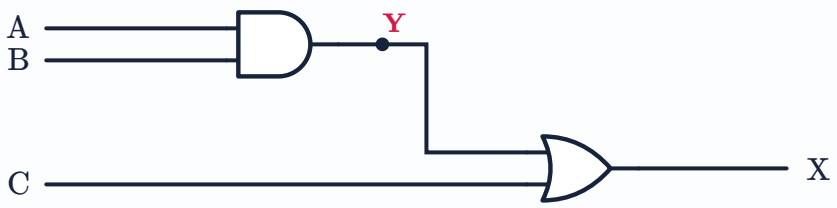
# 9 ซ้อมมือ: อ่านวงจร → เขียนตารางค่าความจริง

**ตะขอเกี่ยว**

ที่ผ่านมาโจทย์ให้เกตมา**ตัวเดียว**ครับ สมมติว่าถ้าเจอวงจรที่มี**สองเกตต่อกัน**ละครับ น้องจะเริ่มจากตรงไหน คำตอบคือ **เริ่มจากซ้ายไปขวาเสมอ ทีละเกต** และ**ตั้งชื่อสายกลางทาง**ไว้ก่อน หัวข้อนี้พี่จะทำให้ดู 1 ข้อ ทำด้วยกัน 1 ข้อ แล้วให้น้องลองเอง 3 ข้อครับ (ข้อพวกนี้เป็นข้อที่พี่แต่งขึ้นเพื่อปูทาง ไม่ใช่ข้อใน Assignment นะครับ)

## I DO

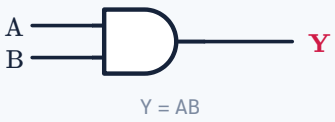
พี่ทำให้ดู — วงจร AND ต่อเข้า OR




พี่ตั้งชื่อสายที่ออกจาก AND ว่า Y เพื่อให้พูดถึงมันได้ครับ

วงจร I DO — AND 2 ขา ต่อเข้า OR 2 ขา โดยขาที่สองของ OR รับ C ตรง ๆ

**1**



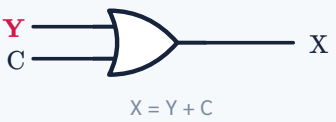

Y = AB

### ตั้งชื่อสายกลางทางก่อน

เพราะเราอ่านวงจรทีละเกต ต้องมีชื่อเรียกสิ่งที่ออกจากเกตแรกครับ พี่ตั้งว่า $Y = AB$

ถ้าไม่ตั้งชื่อ พอมาถึงเกตที่สองจะไม่รู้ว่ากำลังพูดถึงอะไร แล้วจะสับสนทันทีครับ

**2**




X = Y + C

### อ่านเกตที่สอง แล้วแทน Y กลับเข้าไป

เกตที่สองคือ OR รับ $Y$ กับ $C$ → $X = Y + C$

แทน $Y = AB$ กลับเข้าไป ได้ $\mathbf{X = AB + C}$ ครับ เพราะสูตรสุดท้ายต้องเขียนด้วยตัวแปรขาเข้าจริงเท่านั้น ห้ามเหลือชื่อสายกลางทางไว้

**3**

| A | B | C | Y | X |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |

**ทำตาราง 8 แถว แต่ใส่คอลัมน์ Y ไว้ตรงกลาง**

3 ขาเข้า → $2^3 = 8$ แถวครับ พี่แถมคอลัมน์ $Y$ ไว้ระหว่างขาเข้ากับเอาต์พุต

เพราะถ้าเติม $Y$ ก่อน แล้วค่อยเติม $X$ จาก $Y$ กับ $C$ น้องจะ**คิดทีละเกต** ไม่ต้องคิดทั้งวงจรพร้อมกันครับ

**4**

**ตรวจแถวที่ Y = 1**
A=1, B=1 เท่านั้น (2 แถว)

**ตรวจด้วยเหตุผลก่อนส่ง**

$Y = AB$ เป็น 1 แค่ตอน $A$ กับ $B$ เป็น 1 ทั้งคู่ → มีแค่ 2 แถวล่างสุด (แถว 110 และ 111)

เพราะฉะนั้นคอลัมน์ $Y$ ต้องเป็น 0 หกแถวแรกแล้วเป็น 1 สองแถวสุดท้าย ถ้าน้องได้ไม่ตรงนี้ แปลว่าเติมผิดครับ

| A | B | C | $Y = AB$ | $X = Y + C$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

**ทางลัดอ่านวงจรทุกวงในวิชานี้**

**① เดินจากซ้ายไปขวา ② ตั้งชื่อสายที่ออกจากทุกเกต ③ เขียนสูตรของเกตนั้นด้วยชื่อสายที่รู้แล้ว ④ แทนกลับเข้าไปจนเหลือแต่ตัวแปรขาเข้า**

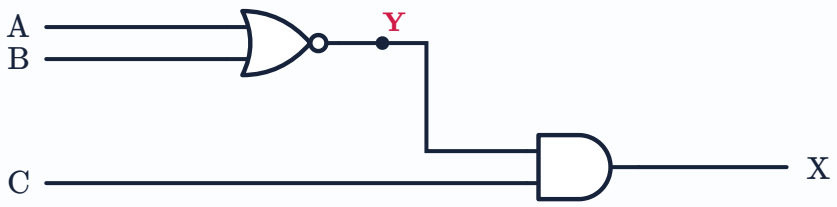
ใช้ได้ทุกวงจรครับ ไม่ว่าจะมี 2 เกตหรือ 20 เกต

$$X = AB + C$$

ตรวจแล้วด้วย `truth_table("(A & B) | C", ["A","B","C"])` ใน `tools/dig.py` ตรงทั้ง 8 แถว

## WE DO

ทำด้วยกัน — คราวนี้เกตแรกมี bubble




เกตแรกเปลี่ยนเป็น NOR และเกตที่สองเปลี่ยนเป็น AND ครับ

วงจร WE DO — NOR 2 ขา ต่อเข้า AND 2 ขา

ลองเติมในใจก่อนดูเฉลยข้างล่างนะครับ: $Y =$ ..... · $X =$ ..... · ตารางจะมีกี่แถว ..... แถว

**เฉลย WE DO — เติมทุกช่องพร้อมเหตุผล**

**ช่องที่ 1 —** $Y = \overline{A + B}$ เพราะเกตแรกเป็น NOR ซึ่ง**มี bubble ที่ขาออก** แปลว่าทำ OR ให้เสร็จก่อนแล้วค่อยกลับค่า ขีดบนจึงคลุมทั้ง $A + B$ ครับ (ระวัง — ไม่ใช่ $\overline{A} + \overline{B}$ นะครับ นั่นคือ negative-OR ซึ่งเป็นคนละตัว)

**ช่องที่ 2 —** $X = \overline{A + B} \cdot C$ เพราะเกตที่สองเป็น AND รับ $Y$ กับ $C \rightarrow X = Y \cdot C$ แล้วแทน $Y$ กลับเข้าไปครับ

**ช่องที่ 3 — 8 แถว** เพราะมีขาเข้า 3 ตัว $(A, B, C) \rightarrow 2^3 = 8$ ครับ

**ไล่ตารางทีละแถวพร้อมเหตุผล** พี่ขอไล่จากคอลัมน์ $Y$ ก่อน เพราะ $Y$ ขึ้นกับ $A, B$ อย่างเดียว:
NOR ให้ 1 แถวเดียวคือแถวที่ $A$ กับ $B$ เป็น 0 ทั้งคู่ $\rightarrow Y = 1$ **เฉพาะสองแถวแรก** (000 และ 001) ที่เหลือเป็น 0
แล้ว $X = Y \cdot C$ ต้องการ**ทั้ง $Y$ และ $C$ เป็น 1** → มีแถวเดียวที่ผ่านคือแถว $A{=}0, B{=}0, C{=}1$ ครับ

| A | B | C | $Y = \overline{A + B}$ | $X = Y \cdot C$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 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 | 0 |

**ตอบ:** $X = \overline{A + B} \cdot C$ **— เป็น 1 แถวเดียว คือแถว** $A = 0, B = 0, C = 1$

*“X is HIGH only when A and B are both LOW and C is HIGH.”*

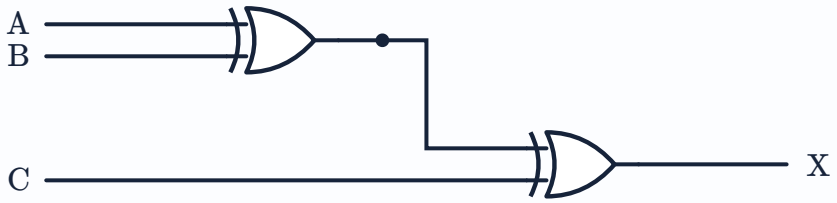
## YOU DO

ตาน้อง — ปิดเฉลยไว้ก่อน ทำเองให้จบแล้วค่อยเปิด

**Y1** พี่แต่งเอง

> Write the truth table and the Boolean expression for the circuit below, which consists of two 2-input XOR gates connected in cascade.

วงจรมี XOR สองตัวต่อกันเป็นสาย ให้เขียนตารางค่าความจริงและนิพจน์บูลีนครับ ใบ้: ตั้งชื่อสายกลางทางก่อนเสมอ




วงจร Y1 — XOR ต่อ XOR

**เฉลย Y1**

**ขั้น 1 — ตั้งชื่อสายกลางทาง** เพราะเราต้องอ่านทีละเกตครับ ให้ $Y = A \oplus B$ (เอาต์พุตของ XOR ตัวแรก)

**ขั้น 2 — อ่านเกตที่สอง** XOR ตัวที่สองรับ $Y$ กับ $C \rightarrow X = Y \oplus C$ แล้วแทนกลับได้ $\mathbf{X = A \oplus B \oplus C}$ ครับ

**ขั้น 3 — เติมคอลัมน์ $Y$ ก่อน** เพราะ $Y$ ขึ้นกับ $A, B$ เท่านั้น กฎ XOR คือ “ต่างได้ 1 เหมือนได้ 0” → $A, B$ = 00 ได้ 0 · 01 ได้ 1 · 10 ได้ 1 · 11 ได้ 0 แต่ละคู่ใช้ซ้ำ 2 แถว (เพราะ $C$ เปลี่ยนได้ 2 ค่า) ครับ

**ขั้น 4 — เติมคอลัมน์ $X$ จาก $Y$ กับ $C$** ใช้กฎ XOR อีกครั้ง เทียบ $Y$ กับ $C$ ทีละแถว

| A | B | C | $Y = A \oplus B$ | $X = Y \oplus C$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |

| A | B | C | $Y = A \oplus B$ | $X = Y \oplus C$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 |

**ขั้น 5 — สังเกตความหมาย** ลองนับจำนวน 1 ในแต่ละแถวของ $A, B, C$ สิครับ: แถวที่ได้ $X = 1$ คือแถวที่มีเลข 1 อยู่ **จำนวนคี่** (1 ตัวหรือ 3 ตัว) พอดีเป๊ะ
นี่คือ **วงจรกำเนิดพาริตี 3 บิต** จากหัวข้อ 5 ครับ — โจทย์นี้ไม่ได้บอก แต่น้องเจอมันโดยบังเอิญ

**ตอบ: $X = A \oplus B \oplus C$ — เป็น 1 เมื่อจำนวนขาเข้าที่เป็น 1 เป็นจำนวนคี่**
*"X = A ⊕ B ⊕ C; the output is HIGH when an odd number of inputs are HIGH (odd-parity function)."*

**Y2** พี่แต่งเอง

> Identify the gate shown below, write its truth table, and name the standard gate that has an identical truth table.

ดูรูปแล้วบอกว่าเป็นเกตอะไร เขียนตารางค่าความจริง แล้วบอกว่าเกตมาตรฐานตัวไหนให้ตารางเหมือนกันเป๊ะครับ ใบ้: ใช้แผนที่ตัดสินใจในหัวข้อ 7 — ดูรูปก่อน แล้วดู bubble




วงจร Y2 — สังเกต bubble ให้ดีว่าอยู่ตรงไหน

**เฉลย Y2**

**ขั้น 1 — ดูรูปตัวเกตก่อน** ตัวเกตปลายแหลมออกมาและด้านหลังเว้าเข้า → **เป็นรูป OR** ครับ

**ขั้น 2 — หา bubble** มี bubble **2 ลูกที่ขาเข้า** ไม่มีที่ขาออก ตามแผนที่ตัดสินใจ: รูป OR + bubble ที่ขาเข้า = **Negative-OR gate** ครับ

**ขั้น 3 — เขียนนิพจน์** bubble ที่ขาเข้าแปลว่ากลับค่าก่อนเข้าเกต → เข้าเกตด้วย $\overline{A}$ กับ $\overline{B}$ แล้วเกตทำ OR → $\mathbf{X = \overline{A} + \overline{B}}$

**ขั้น 4 — ไล่ตารางทีละแถว**
$A{=}0, B{=}0$: $\overline{A}{=}1, \overline{B}{=}1 \rightarrow 1 + 1 = \mathbf{1}$
$A{=}0, B{=}1$: $\overline{A}{=}1, \overline{B}{=}0 \rightarrow 1 + 0 = \mathbf{1}$
$A{=}1, B{=}0$: $\overline{A}{=}0, \overline{B}{=}1 \rightarrow 0 + 1 = \mathbf{1}$
$A{=}1, B{=}1$: $\overline{A}{=}0, \overline{B}{=}0 \rightarrow 0 + 0 = \mathbf{0}$

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

**ขั้น 5 — หาเกตมาตรฐานที่ตารางตรงกัน** คอลัมน์เอาต์พุตคือ $1, 1, 1, 0$ เปิดตารางสรุปหัวข้อ 7 ดู → ตรงกับคอลัมน์ **NAND** เป๊ะทั้ง 4 แถวครับ

**ตอบ: เป็น Negative-OR gate · $X = \overline{A} + \overline{B}$ · ให้ตารางเหมือน NAND gate ทุกแถว**

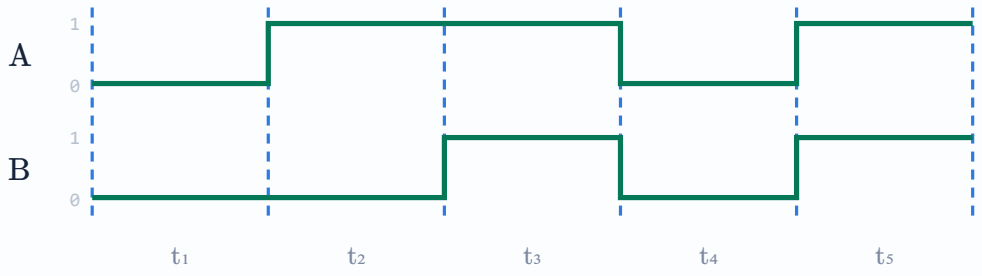
*“This is a negative-OR gate. Its truth table is identical to that of a NAND gate.”*

**กับดักที่ต้องระวัง:** ถ้าตอบว่า “OR gate” เพราะเห็นรูป OR จะได้ตาราง $0, 1, 1, 1$ ซึ่ง**ผิดทุกแถวยกเว้นแถวกลาง**ครับ

**Y3** พี่แต่งเอง

> Given the input waveforms A and B below, sketch the output waveform X for (a) an XOR gate and (b) an XNOR gate. Label the time intervals.

ให้รูปคลื่นขาเข้า A กับ B มา ให้วาดรูปคลื่นขาออกของ (a) XOR และ (b) XNOR พร้อมกำกับช่วงเวลาครับ ใบ้: อ่านทีละช่อง เทียบว่าเส้นสองเส้นอยู่ระดับเดียวกันหรือคนละระดับ




อ่านค่าทีละช่อง แล้วจดคู่ (A,B) ไว้ก่อนวาดครับ

รูปคลื่นขาเข้าของโจทย์ Y3 — มี 5 ช่วงเวลา

**เฉลย Y3**

**ขั้น 1 — อ่านค่าทีละช่อง** เพราะระหว่างเส้นประสัญญาณคงที่ เราจึงอ่านเป็นคู่ค่าได้ครับ

$t_1$: A ต่ำ B ต่ำ → $(0, 0)$ · $t_2$: A สูง B ต่ำ → $(1, 0)$ · $t_3$: A สูง B สูง → $(1, 1)$

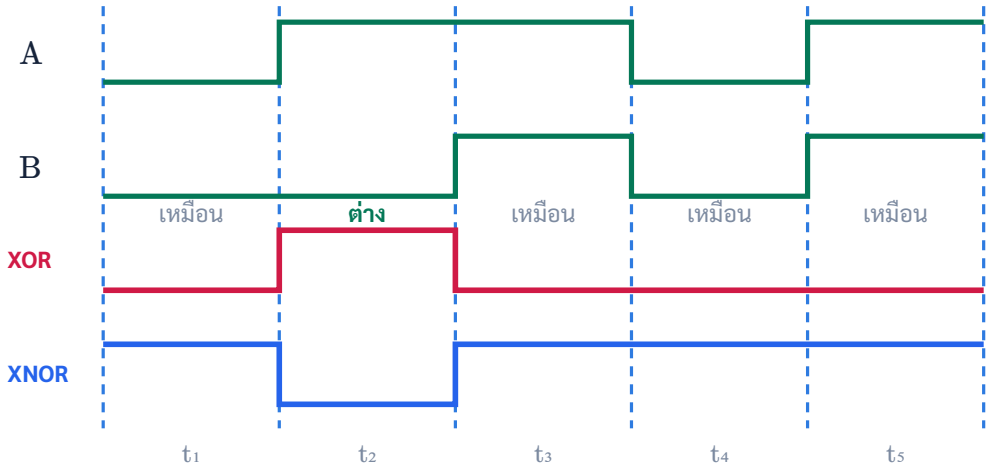
$t_4$: A ต่ำ B ต่ำ → $(0, 0)$ · $t_5$: A สูง B สูง → $(1, 1)$

**ขั้น 2 — เปิดตาราง XOR ใส่ทีละช่อง** กฎคือ “ต่างได้ 1 เหมือนได้ 0” ครับ

$t_1$ เหมือน → 0 · $t_2$ ต่าง → **1** · $t_3$ เหมือน → 0 · $t_4$ เหมือน → 0 · $t_5$ เหมือน → 0

**ขั้น 3 — XNOR คือกลับค่าของ XOR ทุกช่อง** ไม่ต้องคิดใหม่ครับ พลิกทีละช่องได้เลย

$t_1$ → **1** · $t_2$ → 0 · $t_3$ → **1** · $t_4$ → **1** · $t_5$ → **1**




เส้นแดง (XOR) กับเส้นน้ำเงิน (XNOR) เป็นภาพสะท้อนกันเสมอ

เฉลย Y3 — XOR ขึ้นสูงแค่ช่วง $t_2$ ช่วงเดียว ส่วน XNOR ขึ้นสูงทุกช่วงยกเว้น $t_2$

**ขั้น 4 — ตรวจก่อนส่ง** เส้น XOR กับเส้น XNOR ต้อง**อยู่ตรงข้ามกันทุกช่อง** ถ้ามีช่องไหนที่ทั้งคู่อยู่ระดับเดียวกัน แปลว่าน้องวาดผิดครับ

**ตอบ: XOR = 0,1,0,0,0 · XNOR = 1,0,1,1,1 (เรียงตาม $t_1$ ถึง $t_5$)**

*“The XOR output is HIGH only during $t_2$, when A and B are at different levels; the XNOR output is the exact complement of it.”*

# 10 เฉลย Assignment 3 ข้อ 1 — AND 3 ขาเข้า

อ่านตรงนี้ก่อน

สองหัวข้อสุดท้ายนี้คือ**เฉลย Assignment #3 ข้อ 1 กับ ข้อ 2** ครับ พี่จะทำครบ 6 ชั้นทุกข้อ: ① โจทย์ต้นฉบับ + สิ่งที่โจทย์ซ่อนไว้ ② เตรียมก่อนลงมือ ③ ไล่ค่าทีละแถวพร้อมภาพ ④ คำตอบสุดท้าย ⑤ ตรวจด้วยวิธีที่สอง ⑥ กับดักของข้อนี้

**อยากให้น้องลองทำเองก่อนแล้วค่อยเปิดเทียบนะครับ** ข้อพวกนี้เป็นคะแนน 25% ของวิชา

## ชั้น ① โจทย์ต้นฉบับ

**Assignment #3 · ข้อ 1** EES 271 · Assignment 3

> 1. Write a truth table for the following 3-input AND gate.

เขียนตารางค่าความจริงของ AND gate 3 ขาเข้าตัวนี้ครับ โจทย์ให้ช่องตารางเปล่ามาแล้ว 8 แถว พร้อมกำกับว่า `(inputs go here)` สำหรับ 3 คอลัมน์แรก และ `(output goes here)` สำหรับคอลัมน์สุดท้าย — แปลว่าต้อง**เติมหัวคอลัมน์เองด้วย** ไม่ใช่เติมแค่ค่าครับ




ตรวจแล้วจากรูปในโจทย์ที่ขยาย 400 dpi:
**ไม่มี bubble ที่ขาเข้าและไม่มีที่ขาออก — เป็น AND ธรรมดาล้วน**

วงจรในโจทย์ วาดใหม่ด้วยไลบรารีเกตของชีทนี้ — AND 3 ขาเข้า เอาต์พุต $X = ABC$

โจทย์ซ่อนอะไรไว้

① **ต้องเช็ค bubble ก่อนเสมอ** ครับ พี่ซูมรูปในไฟล์ Assignment ดูที่ 400 dpi แล้ว ยืนยันว่า**ไม่มีวงกลมเล็กที่ขาไหนเลย** ทั้งขาเข้าและขาออก และสมการที่โจทย์เขียนกำกับไว้ก็คือ $X = ABC$ ไม่มีขีดบน — เป็น AND ธรรมดาแน่นอน ถ้ามี bubble ที่ขาออกเมื่อไหร่ คำตอบจะกลับด้านทั้งคอลัมน์ทันที

② **โจทย์ไม่ได้ใส่หัวคอลัมน์มาให้** เขียนว่า `(inputs go here)` เฉย ๆ เพราะฉะนั้นต้องเติม $A$, $B$, $C$ และ $X$ ลงหัวตารางเองครับ ถ้าลืม อาจารย์หักคะแนนได้

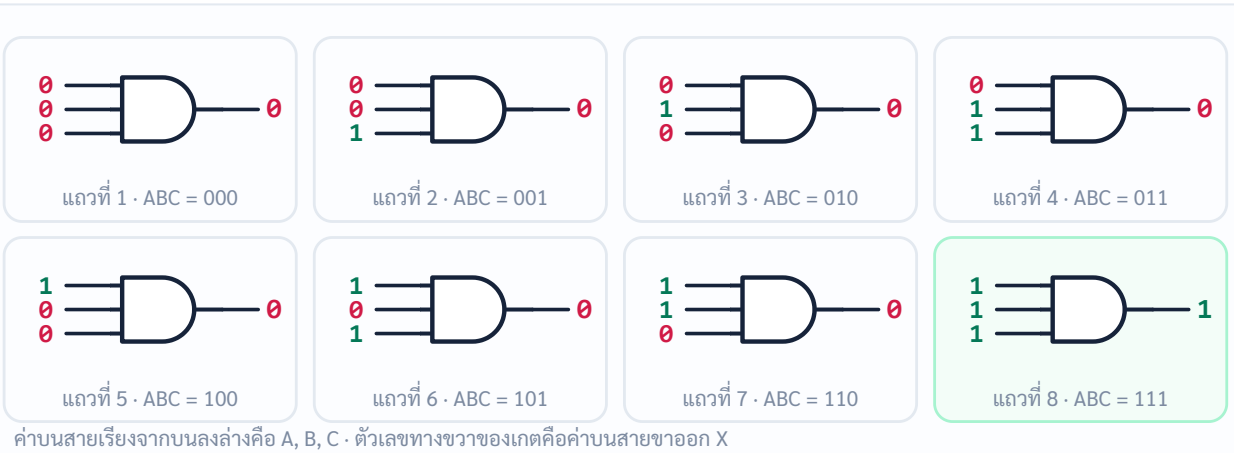
③ **ค่าขาเข้าโจทย์เรียงมาให้แล้ว** เป็น 000 → 111 ตามลำดับ ห้ามสลับ ให้เติมเฉพาะคอลัมน์เอาต์พุตครับ

## ชั้น ② เตรียมก่อนลงมือ

| เตรียมอะไร | รายละเอียด |
|---|---|
| **ใช้วิธีไหน** | ไล่ค่าตรง ๆ ทีละแถว (direct evaluation) ครับ เพราะวงจรมี**เกตเดียว** ไม่มีสายกลางทางให้ตั้งชื่อ จึงไม่ต้องแบ่งขั้น ไม่ต้องใช้ตารางช่วย |
| **กฎที่จะใช้** | **AND ให้ 1 ก็ต่อเมื่อขาเข้าเป็น 1 ครบทุกขา** — ขาดขาเดียวได้ 0 ทันที (สไลด์เขียนว่า `only when all inputs are TRUE`) |
| **จำนวนแถว** | ขาเข้า 3 ตัว → $2^3 = 8$ แถว ตรงกับช่องที่โจทย์ให้มาพอดี 8 ช่องครับ |
| **ลำดับแถว** | เรียงแบบนับเลขฐานสอง 000, 001, 010, 011, 100, 101, 110, 111 (คอลัมน์ขวาสุดสลับทุกแถว · คอลัมน์กลางสลับทุก 2 แถว · คอลัมน์ซ้ายสลับทุก 4 แถว) |
| **สูตรของเอาต์พุต** | $X = A \cdot B \cdot C = ABC$ — เขียนติดกันได้เลย เครื่องหมายคูณละไว้ในฐานที่เข้าใจครับ |

## ชั้น ③ ไล่ค่าทีละแถว พร้อมแสดงค่าบนสายทุกเส้น

พี่จะวาดวงจรเดิม 8 ครั้ง ครั้งละแถว แล้วเขียนค่าลงบนสายทุกเส้นให้เห็นเลยครับ **ตัวเลขสีเขียว = 1 (HIGH)** · **ตัวเลขสีแดง = 0 (LOW)**




ค่าบนสายเรียงจากบนลงล่างคือ A, B, C · ตัวเลขทางขวาของเกตคือค่าบนสายขาออก X

ไล่ครบทั้ง 8 แถว — **สายจากซ้ายเข้าขาเกตทั้งสามขา และสายขวาคือขาออก** สังเกตว่ามีแค่กรอบสุดท้าย (แถวที่ 8) กรอบเดียวที่ขาออกเป็น 1 ครับ

ทีนี้พี่ขอเขียนเหตุผลของแต่ละแถวเป็นตัวหนังสือกำกับด้วย จะได้เห็นว่า “ทำไม” ไม่ใช่แค่ “อะไร” ครับ

**ไล่ทีละแถว — เหตุผลของทุกแถว**

**แถวที่ 1** — A `0` · B `0` · C `0` → มีขาที่เป็น 0 อยู่ (ที่จริง 0 หมดทุกขา) → AND ต้องครบ 1 ทุกขา ไม่ผ่าน → X `0`

**แถวที่ 2** — A `0` · B `0` · C `1` → มี C เป็น 1 ขาเดียว แต่ A กับ B ยังเป็น 0 → ไม่ผ่าน → X `0`

**แถวที่ 3** — A `0` · B `1` · C `0` → A เป็น 0 → พอมีขาไหนเป็น 0 ก็จบทันที ไม่ต้องดูขาอื่นต่อ → X `0`

**แถวที่ 4** — A `0` · B `1` · C `1` → B กับ C เป็น 1 แล้ว แต่ A ยังเป็น 0 → ขาดไปขาเดียวก็ไม่ผ่าน → X `0`

**แถวที่ 5** — A `1` · B `0` · C `0` → A เป็น 1 แล้ว แต่ B กับ C ยังเป็น 0 → X `0`

**แถวที่ 6** — A `1` · B `0` · C `1` → เหลือ B ที่ยังเป็น 0 → X `0`

**แถวที่ 7** — A `1` · B `1` · C `0` → ใกล้แล้วครับ แต่ C ยังเป็น 0 → X `0`
**แถวนี้คือแถวที่คนพลาดบ่อยที่สุด** เพราะเห็น 1 สองตัวแล้วเผลอตอบ 1 ครับ

**แถวที่ 8** — A `1` · B `1` · C `1` → ครบ 1 ทุกขาแล้ว → **ผ่านเงื่อนไข** → X `1`
นี่คือแถวเดียวใน 8 แถวที่เอาต์พุตเป็น 1 ครับ

## ชั้น ④ คำตอบสุดท้าย

| A | B | C | X | minterm |
|---|---|---|---|---|
| 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 | $ABC$ |

**คำตอบที่ต้องเขียนลงกระดาษ**

**คอลัมน์เอาต์พุตจากบนลงล่างคือ**

**`0, 0, 0, 0, 0, 0, 0, 1`**

และนิพจน์บูลีนของเอาต์พุตคือ $X = ABC$

*“X = 1 only for A = B = C = 1; X = 0 for all other input combinations.”*

**อย่าลืมหัวคอลัมน์**

โจทย์เขียน (inputs go here) ไว้เหนือ 3 ช่องแรก และ (output goes here) เหนือช่องสุดท้าย **ให้เติม A, B, C และ X ลงไปด้วยครับ** ไม่งั้นตารางจะไม่สมบูรณ์

$$X = A \cdot B \cdot C = ABC$$

④ นิพจน์บูลีนของเอาต์พุต — ตรงกับที่โจทย์เขียนกำกับไว้ในรูป

## ชั้น ⑤ ตรวจคำตอบด้วยวิธีที่สอง

**1**

1 × 1 × 1 = 1

1 × 1 × 0 = 0

AND = การคูณ

### ตรวจวิธีที่ 1 — เอาไปคูณจริง ๆ

สไลด์บอกว่า AND คือ `multiplication operation` ครับ เพราะฉะนั้นเอาค่าทั้งสามมาคูณกันตรง ๆ ได้เลย

ทุกแถวที่มีเลข 0 อยู่แม้แต่ตัวเดียว ผลคูณย่อมเป็น 0 เสมอ — มีแถวเดียวที่คูณแล้วได้ 1 คือ $1 \times 1 \times 1$ ตรงกับคำตอบครับ

**2**

จำนวน 1 = 1 ตัว

จำนวน 0 = 7 ตัว

### ตรวจวิธีที่ 2 — นับจำนวน 1 ในคอลัมน์เอาต์พุต

กฎที่เรียนในหัวข้อ 3 บอกว่า **AND ไม่ว่ากี่ขา ให้ 1 แถวเดียวเสมอ คือ แถวล่างสุด**

เพราะฉะนั้นคอลัมน์เอาต์พุตต้องมีเลข 1 **1 ตัวพอดี** และเลข 0 อีก 7 ตัว นับแล้วตรงครับ ถ้านับได้ 2 ตัวขึ้นไปแปลว่าเติมผิดแน่นอน

**3**

```
truth_table(
 "A & B & C"
 ["A","B","C"])
```

### ตรวจวิธีที่ 3 — รันด้วย tools/dig.py จริง

พี่รัน `truth_table("A & B & C", ["A","B","C"])` จริงก่อนพิมพ์ลงชีทครับ ผลที่ได้คือ 8 แถว เอาต์พุตเป็น `0 0 0 0 0 0 0 1` และ `minterms(...)` คืนค่า `[7]` แปลว่ามีแถวเดียวที่เป็น 1 คือแถวที่ 7 (นับจาก 0) = แถวสุดท้าย ตรงกับตารางข้างบนเป๊ะครับ

**ผลรันจริงจาก tools/dig.py**

```
--- A3 Q1  X = ABC   expr = A & B & C
    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
Q1 minterms : [7]
Q1 maxterms : [0, 1, 2, 3, 4, 5, 6]
```

## ชั้น ⑥ กับดักของข้อนี้

**กับดัก 4 อย่างที่ทำให้เสียคะแนนข้อนี้**

**① อ่านรูปผิดเป็น NAND** — ถ้าเผลอเห็นว่ามี bubble ที่ขาออก คำตอบจะกลายเป็น `1 1 1 1 1 1 1 0` ซึ่ง**ผิดทั้ง 8 แถว**
วิธีกัน: ดูสมการที่โจทย์เขียนกำกับไว้ครับ — โจทย์เขียน $X = ABC$ **ไม่มีขีดบน** แปลว่าเป็น AND แน่นอน

**② เขียนแค่ 4 แถวเพราะติดภาพเกต 2 ขา** — 3 ขาเข้าต้องมี 8 แถวครับ ถ้าเขียน 4 แถวจะเหลือครึ่งเดียว คำตอบผิดได้ **0 คะแนน**
วิธีกัน: นับจำนวนขาเข้าก่อนแล้วคิด $2^n$ ทุกครั้ง

**③ ตอบ 1 ที่แถว 110** — เพราะเห็นเลข 1 อยู่สองตัวแล้วรู้สึกว่า “ก็เกือบครบแล้ว”
AND ไม่มีคำว่าเกือบครับ ต้องครบ 1 ทุกขาเท่านั้น แถว 110 ยังมี $C = 0$ อยู่ → ตอบ **0**

**④ สลับลำดับแถวเอง** — บางคนเรียง 000, 100, 010, ... ตามใจ ทำให้คำตอบไปไม่ตรงกับช่องที่โจทย์เตรียมไว้
โจทย์**พิมพ์ค่าขาเข้ามาให้แล้ว**ครับ ห้ามแก้ ให้เติมเฉพาะคอลัมน์เอาต์พุตอย่างเดียว

**หัวใจของข้อนี้**

**AND ทุกขนาด ให้ 1 แถวเดียวเสมอ คือแถวที่ขาเข้าเป็น 1 ครบทุกขา — ที่เหลือเป็น 0 ทั้งหมด เขียนคำตอบได้ในสิบวินาที ขอแค่ตรวจ bubble ให้ถูกก่อนครับ**

# 11 เฉลย Assignment 3 ข้อ 2 — OR 4 ขาเข้า

## ชั้น ① โจทย์ต้นฉบับ

**Assignment #3 · ข้อ 2** EES 271 · Assignment 3

> 2. Write a truth table for the following 4-input OR gate.

เขียนตารางค่าความจริงของ OR gate 4 ขาเข้าตัวนี้ครับ โจทย์ให้ช่องเปล่ามา **16 แถว** พร้อมกำกับ `(inputs go here)` 4 คอลัมน์ และ `(output goes here)` อีก 1 คอลัมน์ — ต้องเติมหัวคอลัมน์เองเช่นเดิมครับ




ตรวจแล้วจากรูปในโจทย์ที่ขยาย 400 dpi:
**ไม่มี bubble ที่ขาไหนเลย — เป็น OR ธรรมดา 4 ขาเข้า**

วงจรในโจทย์ วาดใหม่ด้วยไลบรารีเกตของชีทนี้ — OR 4 ขาเข้า เอาต์พุต $X = A + B + C + D$

**โจทย์ซ่อนอะไรไว้ — ข้อนี้มีกับดักทางสายตา**

① **“รอยหยัก” บนสาย A กับสาย D ไม่ใช่ bubble ครับ** ในรูปของโจทย์ สาย A (บนสุด) กับสาย D (ล่างสุด) มี**รอยหักเป็นขั้นบันได**ก่อนเข้าเกต หลายคนเห็นแล้วนึกว่าเป็นวงกลมกลับค่า — **ไม่ใช่ครับ** มันเป็นแค่**วิธีเขียนขาของเกต 4 ขาเข้า** เพราะขอบด้านหลังของรูป OR โค้งเว้า ขาบนสุดกับล่างสุดเลยต้องหักเข้าไปหาขอบ สไลด์หน้า 8 ก็วาดแบบนี้เหมือนกันเป๊ะ
พี่ซูมรูปในไฟล์ Assignment ที่ 400 dpi ยืนยันแล้วว่า**ไม่มีวงกลมเล็กที่ขาไหนเลย** และสมการที่โจทย์เขียนกำกับคือ $X = A + B + C + D$ **ไม่มีขีดบน** ยืนยันซ้ำอีกชั้นครับ
② **16 แถว ไม่ใช่ 8 แถว** — ขาเข้า 4 ตัว → $2^4 = 16$ ครับ ต้องเขียนให้ครบทุกแถว
③ **ต้องเติมหัวคอลัมน์เอง** เหมือนข้อ 1

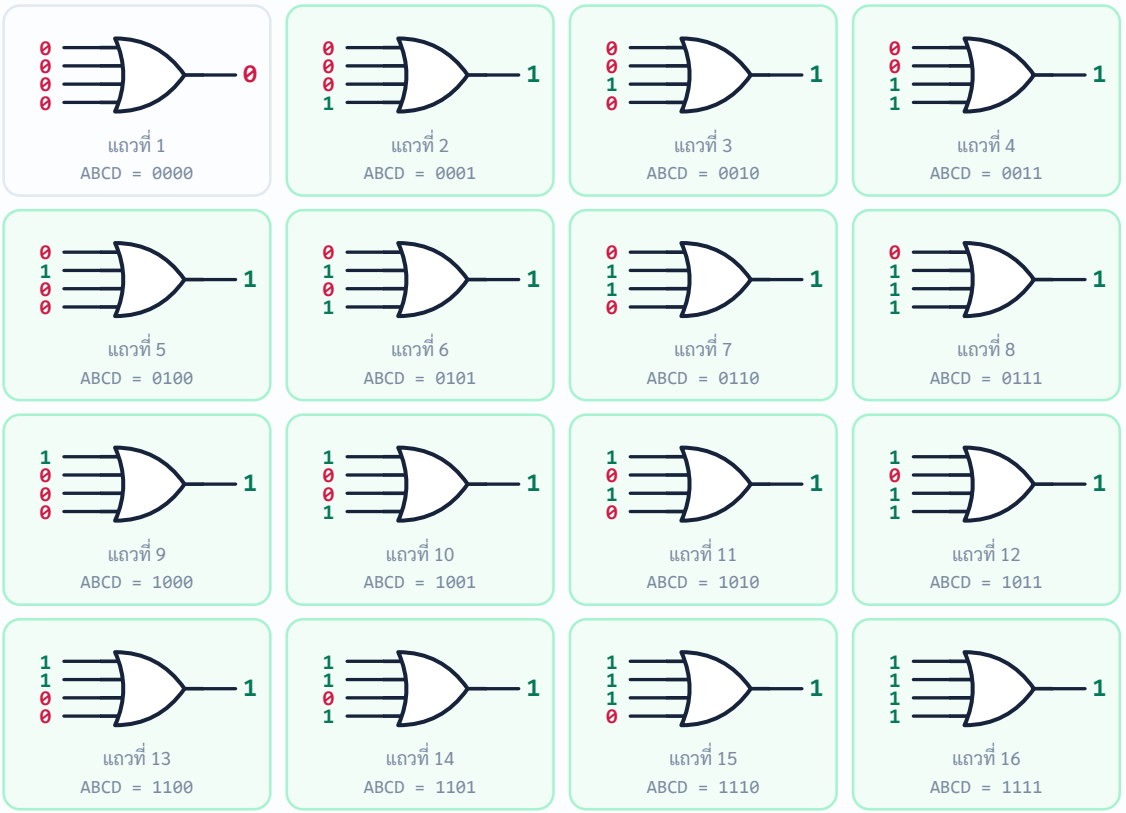
## ชั้น ② เตรียมก่อนลงมือ

| เตรียมอะไร | รายละเอียด |
|---|---|
| **ใช้วิธีไหน** | ไล่ค่าตรง ๆ ทีละแถวเหมือนข้อ 1 ครับ เพราะมีเกตเดียว ไม่มีสายกลางทาง แต่คราวนี้จะใช้**ทางลัดของ OR** เข้าช่วย เพราะ 16 แถวถ้าไล่ทีละตัวจะช้าและพลาดง่าย |

| เตรียมอะไร | รายละเอียด |
|---|---|
| **กฎ + ทางลัด** | **OR ให้ 1 ถ้ามีขาเข้าเป็น 1 อย่างน้อยหนึ่งขา** (สไลด์เขียนว่า `when one or all of the inputs are TRUE`) → ทางลัดคือ **เขียน 1 ลงทั้งคอลัมน์ก่อน แล้วกลับไปแก้แถวแรก (0000) ให้เป็น 0** ครับ |
| **จำนวนแถว** | $2^4 = 16$ แถว · เอาต์พุตจะเป็น 1 จำนวน **15 ตัว** และเป็น 0 จำนวน **1 ตัว** |
| **ลำดับแถว** | นับเลขฐานสอง 0000 → 1111 · คอลัมน์ D สลับทุกแถว · C สลับทุก 2 · B สลับทุก 4 · A สลับทุก 8 |

## ชั้น ③ ไล่ค่าทีละแถว พร้อมแสดงค่าบนสายทุกเส้น

วาดวงจรเดิม 16 ครั้ง ครั้งละแถว เขียนค่าลงบนสายทั้ง 5 เส้น (ขาเข้า 4 + ขาออก 1) ครับ **เขียว = 1 · แดง = 0**




ค่าบนสายเรียงจากบนลงล่างคือ A, B, C, D · ตัวเลขทางขวาของเกตคือค่าบนสายขาออก X

ไล่ครบทั้ง 16 แถว — **มีกรอบเดียวเท่านั้น (แถวที่ 1) ที่ขาออกเป็น 0** เพราะเป็นแถวเดียวที่ไม่มีสายไหนเป็น 1 เลยครับ

**ไล่ทีละแถว — ค่าบนสายทุกเส้นและเหตุผล**

**แถว 1** · A `0` B `0` C `0` D `0` → ไม่มีสายไหนเป็น 1 เลย สักเส้น → OR ไม่ผ่าน → X `0`

**แถว 2** · A `0` B `0` C `0` D `1` → มีสายที่เป็น 1 อยู่ 1 เส้น → พอแล้วสำหรับ OR → X `1`

**แถว 9** · A `1` B `0` C `0` D `0` → มีสายที่เป็น 1 อยู่ 1 เส้น → พอแล้วสำหรับ OR → X `1`

**แถว 10** · A `1` B `0` C `0` D `1` → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X `1`

**แถว 3** · A 0 B 0 C 1 D 0 → มีสายที่เป็น 1 อยู่ 1 เส้น → พอแล้วสำหรับ OR → X 1

**แถว 4** · A 0 B 0 C 1 D 1 → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X 1

**แถว 5** · A 0 B 1 C 0 D 0 → มีสายที่เป็น 1 อยู่ 1 เส้น → พอแล้วสำหรับ OR → X 1

**แถว 6** · A 0 B 1 C 0 D 1 → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X 1

**แถว 7** · A 0 B 1 C 1 D 0 → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X 1

**แถว 8** · A 0 B 1 C 1 D 1 → มีสายที่เป็น 1 อยู่ 3 เส้น → เกินพอสำหรับ OR → X 1

**แถว 11** · A 1 B 0 C 1 D 0 → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X 1

**แถว 12** · A 1 B 0 C 1 D 1 → มีสายที่เป็น 1 อยู่ 3 เส้น → เกินพอสำหรับ OR → X 1

**แถว 13** · A 1 B 1 C 0 D 0 → มีสายที่เป็น 1 อยู่ 2 เส้น → เกินพอสำหรับ OR → X 1

**แถว 14** · A 1 B 1 C 0 D 1 → มีสายที่เป็น 1 อยู่ 3 เส้น → เกินพอสำหรับ OR → X 1

**แถว 15** · A 1 B 1 C 1 D 0 → มีสายที่เป็น 1 อยู่ 3 เส้น → เกินพอสำหรับ OR → X 1

**แถว 16** · A 1 B 1 C 1 D 1 → มีสายที่เป็น 1 อยู่ 4 เส้น → เกินพอสำหรับ OR → X 1

**สรุป: มีแถวเดียวจาก 16 แถวที่ได้ 0 คือแถวแรก ($A = B = C = D = 0$) ที่เหลืออีก 15 แถวได้ 1 หมดครับ**

## ชั้น ④ คำตอบสุดท้าย

| A | B | C | D | X |
|---|---|---|---|---|
| 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 |

**คำตอบที่ต้องเขียนลงกระดาษ**

**คอลัมน์เอาต์พุตจากบนลงล่างคือ 0, แล้ว 1 อีก 15 ตัวรวด**

และนิพจน์บูลีนของเอาต์พุตคือ $X = A + B + C + D$

*“X = 0 only when A = B = C = D = 0; X = 1 for all other 15 input combinations.”*

**เขียนคำตอบข้อนี้ใน 10 วินาที**

ลากเลข **1** ลงมาให้ครบ 16 ช่องก่อนเลย ครับ แล้วย้อนกลับไป**ลบแถวบนสุดทิ้งแล้วเขียน 0** แทน จบ

ไม่ต้องไล่ทีละแถวให้เสียเวลาในห้องสอบ แต่**ต้องมั่นใจก่อนว่าเป็น OR ธรรมดาจริง** ๆ (ไม่มี bubble) ถึงจะใช้ทางลัดนี้ได้ครับ

| A | B | C | D | X |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |

$$X = A + B + C + D$$

④ นิพจน์บูลีนของเอาต์พุต — ตรงกับที่โจทย์เขียนกำกับไว้ในรูป

## ชั้น ⑤ ตรวจคำตอบด้วยวิธีที่สอง

**1**

จำนวน 1 = 15 ตัว

จำนวน 0 = 1 ตัว

**ตรวจวิธีที่ 1 — นับจำนวนในคอลัมน์เอาต์พุต**

กฎจากหัวข้อ 3 บอกว่า **OR ไม่ว่ากี่ขา ให้ 0 แถวเดียวเสมอ คือแถวบนสุด**

เพราะฉะนั้นจากทั้งหมด 16 แถว ต้องได้ **1 จำนวน 15 ตัว** และ **0 จำนวน 1 ตัว**

นับในตารางแล้วตรงครับ ถ้านับ 0 ได้ 2 ตัวขึ้นไป แปลว่าเติมผิดแน่นอน

**2**

พลิกไปมองอีกด้าน

X = 0 ⇔ 0 หมดทุกขา

มีทางเดียวเท่านั้น

**ตรวจวิธีที่ 2 — คิดกลับด้าน**

ถามใหม่ว่า “**X จะเป็น 0 ได้เมื่อไหร่บ้าง**” ครับ

OR จะได้ 0 ก็ต่อเมื่อ**ไม่มีขาไหนเป็น 1 เลย** ซึ่งเกิดขึ้นได้ทางเดียวคือ $A = B = C = D = 0$

จาก 16 ชุดค่าที่เป็นไปได้ มีชุดเดียวที่เป็นแบบนี้ → เพราะฉะนั้นต้องมี 0 แค่ตัวเดียว ตรงกับคำตอบครับ

**3**

```
truth_table(
 "A | B | C | D",
 ["A","B","C","D"])
```

**ตรวจวิธีที่ 3 — รันด้วย tools/dig.py จริง**

พี่รัน `truth_table("A | B | C | D", ["A","B","C","D"])` จริงก่อนพิมพ์ลงชีทครับ

ได้ 16 แถว · ผลรวมของคอลัมน์เอาต์พุต = **15** · และ `maxterms(...)` คืนค่า `[0]` แปลว่ามีแถวเดียวที่ได้ 0 คือแถวที่ 0 = แถวแรก ตรงกับตารางเป๊ะครับ

ผลรันจริงจาก tools/dig.py

```
--- A3 Q2  X = A+B+C+D   expr = A | B | C | D
    0 0 0 0  ->  0        1 0 0 0  ->  1
    0 0 0 1  ->  1        1 0 0 1  ->  1
    0 0 1 0  ->  1        1 0 1 0  ->  1
    0 0 1 1  ->  1        1 0 1 1  ->  1
    0 1 0 0  ->  1        1 1 0 0  ->  1
    0 1 0 1  ->  1        1 1 0 1  ->  1
    0 1 1 0  ->  1        1 1 1 0  ->  1
    0 1 1 1  ->  1        1 1 1 1  ->  1
Q2 maxterms : [0]
Q2 ones     : 15
A+B+C+D == ABCD : False   (กันสับสนกับข้อ 1)
```

## ชั้น ⑥ กับดักของข้อนี้

**กับดัก 4 อย่างที่ทำให้เสียคะแนนข้อนี้**

**① เห็นรอยหยักบนสาย A กับ D แล้วนึกว่าเป็น bubble** — ถ้าคิดว่า A กับ D ถูกกลับค่า คำตอบจะกลายเป็น $X = \overline{A} + B + C + \overline{D}$ ซึ่ง**ผิด 2 แถว** (ตรวจด้วย `dig.py` แล้ว): แถว `0000` จะได้ 1 แทนที่จะเป็น 0 — ซึ่งเป็น**แถวเดียวที่ควรเป็น 0** พลาดแถวนี้แถวเดียวรูปร่างคำตอบก็ผิดทั้งข้อ — และแถว `1001` จะได้ 0 แทนที่จะเป็น 1
วิธีกัน: **bubble คือวงกลม** ถ้าไม่เห็นวงกลมก็ไม่ใช่ bubble ครับ

**② เขียนแค่ 8 แถว** — เพราะเผลอนับขาเข้าเป็น 3 ครับ ข้อนี้มี 4 ขา ต้อง 16 แถว
วิธีกัน: นับตัวอักษรในสมการ $A + B + C + D$ ได้ 4 ตัว → $2^4 = 16$ ครับ

**③ สับสนกับข้อ 1 แล้วใช้กฎ AND** — ข้อ 1 เป็น AND (1 แถวเดียวที่ได้ 1) ข้อ 2 เป็น OR (1 แถวเดียวที่ได้ 0) **กลับด้านกันสนิท** ถ้าเอากฎข้อ 1 มาใช้จะ**ผิดทั้ง 16 แถว**
วิธีกัน: ดูรูปเกตก่อนเสมอ — **หน้าโค้งกลม = AND · ปลายแหลม = OR**

**④ เรียงลำดับแถวใหม่เอง** — โจทย์พิมพ์ค่าขาเข้ามาให้ครบ 16 แถวแล้ว ห้ามแก้ครับ ให้เติมเฉพาะคอลัมน์เอาต์พุต ไม่งั้นอาจารย์ตรวจไม่ตรงช่อง

**หัวใจของข้อนี้ และทำไมสองข้อนี้ถึงเป็นคู่กัน**

**OR ทุกขนาด ให้ 0 แถวเดียวเสมอ คือแถวที่ขาเข้าเป็น 0 ครบทุกขา — ที่เหลือเป็น 1 ทั้งหมด และรอยหักของสายไฟไม่ใช่ bubble ต้องเห็นวงกลมจริง ๆ เท่านั้นถึงจะนับครับ**

อาจารย์ออกข้อ 1 เป็น AND และข้อ 2 เป็น OR ติดกัน เพราะเป็น**ขั้วตรงข้ามกัน** — AND มี 1 แถวเดียวที่ล่างสุด · OR มี 0 แถวเดียวที่บนสุด ส่วนข้อ 3–5 ของ Assignment ชุดนี้เป็นเรื่องพีชคณิตบูลีน ซึ่งอยู่ใน EP06 ครับ

# 12 กับดักรวม · สรุปหน้าเดียว · เช็คลิสต์ก่อนสอบ

## กับดักรวมของทั้ง EP — 8 ข้อที่ทำให้เสียคะแนนบ่อยที่สุด

กับดัก 1–4

**① เห็นรูป OR ที่มี bubble ขาเข้าแล้วตอบว่า OR**

ต้องดู bubble ก่อนเสมอครับ รูป OR + bubble ขาเข้า = negative-OR = NAND ตาราง OR คือ $0111$ แต่ negative-OR คือ $1110$ — **ต่างกัน 3 แถวจาก 4**

**② คิดว่า $1+1=2$**

ในบูลีนมีแค่ 0 กับ 1 ครับ $1+1=1$ วิธีกันคือ**อ่านเครื่องหมาย $+$ ว่า "ออร์" ทุกครั้ง** อย่าอ่านว่า "บวก"

**③ สับสน $\overline{AB}$ กับ $\overline{A}\,\overline{B}$**

ความยาวของขีดบนคือลำดับการทำครับ ขีดยาวคลุมทั้งก้อน = ทำข้างในให้เสร็จก่อนแล้วค่อยกลับค่า ที่ $A=0, B=1$: $\overline{AB}=1$ แต่ $\overline{A}\,\overline{B}=0$ — คนละค่าเลย

**④ ลอกตาราง OR ไปใส่ช่อง XOR**

สองตัวนี้ต่างกันแถวเดียว คือแถว $(1,1)$: OR ได้ 1 · XOR ได้ 0 วิธีกัน: **นับจำนวน 1 ในคอลัมน์ XOR ต้องได้ 2 ตัวพอดี** ถ้าได้ 3 แปลว่าลอก OR มา

กับดัก 5–8

**⑤ เขียนตารางไม่ครบแถว**

$n$ ขาเข้า ต้องมี $2^n$ แถวเสมอ — 2 ขา 4 แถว · 3 ขา 8 แถว · 4 ขา 16 แถว วิธีกัน: **นับจำนวนตัวแปรในสูตรก่อนลงมือเขียนทุกครั้ง**

**⑥ ย้าย bubble โดยลืมสลับรูป AND ↔ OR**

ย้าย bubble ข้ามเกตต้อง**สลับรูปด้วยเสมอ** ถ้าลืม จะได้เกตผิดตัวไปเลย (NAND ที่ลืมสลับ จะกลายเป็น negative-AND ซึ่งคือ NOR)

**⑦ อ่านวงจรหลายเกตแบบข้ามขั้น**

ต้องตั้งชื่อสายกลางทางก่อนเสมอครับ แล้วอ่านทีละเกตจากซ้ายไปขวา ถ้าพยายามคิดทั้งวงจรพร้อมกันจะพลาดแน่นอน

**⑧ วาด timing diagram โดยไม่ตรงกับตาราง**

เอาต์พุตเปลี่ยนค่าได้**เฉพาะตรงเส้นประ**เท่านั้น เพราะระหว่างช่วง ขาเข้ายังคงที่อยู่ ถ้ามีเส้นเอาต์พุตขึ้น-ลงกลางช่อง แปลว่าวาดผิดครับ

## สรุปหน้าเดียว — อ่านก่อนเข้าห้องสอบ

# EP05 · ลอจิกเกต 7 ตัว — ทุกอย่างที่ต้องจำ

### 1 · เกตทั้ง 7 ในบรรทัดเดียว

**NOT** $X = \overline{A}$ — กลับค่า | **AND** $X = AB$ — 1 หมดทุกขาถึงได้ 1 | **OR** $X = A + B$ — มี 1 สักขาก็พอ

**NAND** $X = \overline{AB}$ — AND แล้วพลิก | **NOR** $X = \overline{A + B}$ — OR แล้วพลิก

**XOR** $X = A \oplus B$ — ต่างได้ 1 เหมือนได้ 0 | **XNOR** $X = \overline{A \oplus B}$ — เหมือนได้ 1 ต่างได้ 0

### 2 · คอลัมน์เอาต์พุตที่ต้องท่องให้ขึ้นใจ (เรียงจากแถว 00 ถึง 11)

**AND** 0 0 0 **1** · **OR** **0** 1 1 1 · **NAND** 1 1 1 **0** · **NOR** **1** 0 0 0

**XOR** 0 **1** **1** 0 · **XNOR** **1** 0 0 **1** · **NOT** 1 0 (มีแค่ 2 แถว)

จำแค่ **AND · OR · XOR** ก็พอ อีก 3 ตัวคือ**กลับค่าทั้งคอลัมน์**ครับ

### 3 · กฎ bubble — ใช้ได้ทั้งวิชา

**เห็น bubble ที่ไหน แปลว่ากลับค่าที่นั่น และจุดนั้นคือ active-LOW**

ไม่มี bubble = เกตธรรมดา · bubble ที่ขาออก = NAND / NOR · bubble ที่ขาเข้า = negative-AND / negative-OR

**ย้าย bubble ข้ามเกต ต้องสลับรูป AND ↔ OR ทุกครั้ง**

### 4 · dual symbol ที่พิสูจน์แล้วในหัวข้อ 6

**NAND = negative-OR** → $\overline{AB} = \overline{A} + \overline{B}$ (ตารางตรงกัน 4/4 แถว)

**NOR = negative-AND** → $\overline{A + B} = \overline{A} \cdot \overline{B}$ (ตารางตรงกัน 4/4 แถว)

เกตเดียวกัน คนละสัญลักษณ์ — เลือกใช้ตัวที่ตรงกับ**เจตนาของผู้ออกแบบ** (active-HIGH หรือ active-LOW)

### 5 · วิธีอ่านวงจรหลายเกต (ใช้ได้ทุกวง)

① เดินจากซ้ายไปขวา ② ตั้งชื่อสายที่ออกจากทุกเกต เช่น $Y$ ③ เขียนสูตรของเกตนั้นด้วยชื่อที่รู้แล้ว ④ แทนกลับจนเหลือแต่ตัวแปรขาเข้า

ในตารางให้**ใส่คอลัมน์ของสายกลางทางไว้ด้วย** จะได้คิดทีละเกต ไม่ต้องคิดทั้งวงพร้อมกัน

### 6 · timing diagram

เส้นสูง = 1 · เส้นต่ำ = 0 · เส้นประแนวตั้ง = ที่คั่นช่วงเวลา $t_1, t_2, \ldots$

อ่านทีละช่อง เอานิ้วชี้ลงมาตั้งฉาก อ่านทุกเส้นพร้อมกัน แล้วเปิดตารางค่าความจริงหาเอาต์พุตของช่องนั้น

**เอาต์พุตเปลี่ยนค่าได้เฉพาะตรงเส้นประเท่านั้น**

### 7 · จำนวนแถวและทางลัดตอบเร็ว

$n$ ขาเข้า → $2^n$ แถว · **AND ทุกขนาด** ให้ 1 แถวเดียว คือแถวล่างสุด · **OR ทุกขนาด** ให้ 0 แถวเดียว คือแถวบนสุด

NAND กับ NOR ก็คือสองอันนั้นที่**พลิกทั้งคอลัมน์**ครับ

### 8 · XOR คือเกตที่น้องใช้มาแล้ว

**Gray code** (EP04): $G_i = B_i \oplus B_{i+1}$ — “ต่างเขียน 1 เหมือนเขียน 0” คือ XOR ตรง ๆ

**Parity** (EP04): $P = D_3 \oplus D_2 \oplus D_1 \oplus D_0$ — XOR ต่อกันเป็นสาย = เครื่องนับจำนวน 1 ที่เป็นคี่

**ทางลัด:** $A \oplus 0 = A$ (ผ่านตรง ๆ) และ $A \oplus 1 = \overline{A}$ (กลับค่า) → **XOR คือ NOT ที่เปิด-ปิดได้**

## เช็คลิสต์ก่อนสอบ — ทำได้ทุกข้อไหม

| ทำได้? | ต้องทำอะไรได้ |
|---|---|
| ..... | วาดสัญลักษณ์ของเกตทั้ง 7 ตัวจากความจำ พร้อมบอกว่า bubble อยู่ตรงไหน |
| ..... | เขียนตารางค่าความจริงของเกตทั้ง 7 ตัวได้ภายใน 2 นาที โดยไม่เปิดชีท |
| ..... | เขียนนิพจน์บูลีนของทุกเกตด้วย**ขีดบน**ได้ถูกต้อง (ไม่ใช้ $A'$ หรือ $\sim A$) |
| ..... | วาด timing diagram ของเกตใดก็ได้ เมื่อโจทย์ให้รูปคลื่นขาเข้ามา |
| ..... | ดูรูปเกตแล้วบอกได้ทันทีว่าเป็น AND / OR / NAND / NOR / negative-AND / negative-OR |
| ..... | พิสูจน์ได้ว่า NAND = negative-OR โดยวางตารางค่าความจริงคู่กันแล้วไล่ทีละแถว |
| ..... | อธิบายได้ว่าทำไมเกตตัวเดียวถึงต้องมีสองสัญลักษณ์ (active-HIGH vs active-LOW) |
| ..... | อ่านวงจรที่มี 2–3 เกตต่อกัน แล้วเขียนทั้งนิพจน์และตารางค่าความจริงได้ |
| ..... | บอกได้ว่าตาราง $n$ ขาเข้ามีกี่แถว และเขียนลำดับ 0000 → 1111 ได้ไม่ตกแถว |
| ..... | โยง XOR กลับไปหา Gray code และ parity ของ EP04 ได้ |
| ..... | ทำ Assignment 3 ข้อ 1 กับ ข้อ 2 ได้เองโดยไม่เปิดเฉลย |

**EP ต่อไปจะเจออะไร**

**ตอนนี้น้องอ่านวงจรได้แล้วครับ** เห็นรูปเกตแล้วเขียนตารางได้ เห็นตารางแล้วเขียนสูตรได้ แต่พี่มีคำถามทิ้งท้าย

**ถ้าวงจรมันใหญ่จนใช้เกต 20 ตัวล่ะครับ จะลดให้เหลือ 5 ตัวโดยที่ผลลัพธ์เหมือนเดิมเป๊ะได้ไหม**

คำตอบคือ **ได้ครับ** และมันคือเนื้อหาของ **EP06 — พีชคณิตบูลีน** ที่จะสอนกฎ 12 ชุดสำหรับ “ย่อ” นิพจน์ให้สั้นลงโดยที่ตารางค่าความจริงไม่เปลี่ยนแม้แต่แถวเดียว เกตน้อยลง = ชิปถูกลง กินไฟน้อยลง ทำงานเร็วขึ้น และของสองอย่างที่น้อง**พิสูจน์ด้วยตารางไปแล้ว**ในหัวข้อ 6 — $\overline{AB} = \overline{A} + \overline{B}$ กับ $\overline{A + B} = \overline{A}\,\overline{B}$ — EP06 จะบอกชื่อจริงของมันว่า `De Morgan's theorems` ซึ่งเป็นกฎที่ใช้บ่อยที่สุดในทั้งวิชาครับ
ส่วน Assignment 3 ข้อ 3 (a–i), ข้อ 4 (a–b) และข้อ 5 ก็อยู่ใน EP06 เหมือนกันครับ

## แหล่งอ้างอิงของ EP นี้

| แหล่ง | ส่วนไหน | ใช้ทำอะไรในชีทนี้ |
|---|---|---|
| **สไลด์ EES 271 Lecture 03**<br>Logic Gates and Boolean Algebra (2026) | หน้า 2–3 | นิยาม logic function · ตัวอย่างหลอดไฟกับสวิตช์ · 3 วิธีวิเคราะห์เกต · ANSI/IEEE 91-1984 |
| — | หน้า 4–13 | เกตทั้ง 7 ตัว (NOT · AND · OR · NAND · NOR · XOR · XNOR) พร้อมตารางค่าความจริง timing diagram และเกต 3–4 ขาเข้า |
| — | หน้า 14–17 | negative-AND / negative-OR · opposing logic · dual symbols · ตารางสรุปทั้ง 9 หน้าตาในหน้าเดียว |
| — | หน้า 18–20 | ตัวอย่างการใช้งาน 3 อัน: 1's complement · ระบบเตือนภัย · ระบบเฝ้าล้อเครื่องบิน |
| **Assignment #3** | ข้อ 1–2 | เฉลยเต็ม 6 ชั้นในหัวข้อ 10 และ 11 (ข้อ 3–5 อยู่ใน EP06) |
| **Floyd, Digital Fundamentals**<br>10th edition | Chapter 3 | สัญลักษณ์เกตแบบ distinctive shape · ตัวอย่างระบบเฝ้าล้อเครื่องบิน |
| **tools/dig.py** | `truth_table` · `same_logic` · `minterms` · `maxterms` | ตรวจตารางค่าความจริงทุกตารางในชีทนี้ก่อนพิมพ์ลงกระดาษ |

**หมายเหตุเรื่องการตรวจคำตอบ**

ตารางค่าความจริงทุกตารางใน EP นี้ผ่านการรัน `tools/dig.py` จริงก่อนพิมพ์ครับ ทั้งเกตทั้ง 7 ตัว · คู่ dual symbol ทั้งสองคู่ (ตรวจด้วย `same_logic` ได้ True ทั้งคู่) · วงจรใน I DO / WE DO / YOU DO · และ Assignment ข้อ 1 กับ ข้อ 2

เวลาน้องเขียนนิพจน์ให้ `dig.py` ตรวจเอง ให้เขียน NOT เป็น `(A ^ 1)` แทน `~A` ครับ เพราะ `~` ของ Python เป็นการกลับบิตของจำนวนเต็ม ให้ค่า $-1$ กับ $-2$ ซึ่งนับเป็น “จริง” ทั้งคู่ — ถ้าใช้ไว้นอกสุดของนิพจน์ ผลจะออกมาเป็น 1 ทุกแถวครับ