



การจัดองค์การคอมพิวเตอร์

Hardware description language

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิตติศรีวรพันธุ์

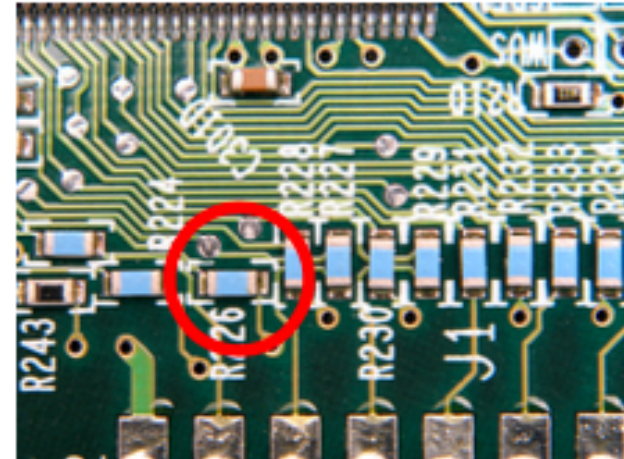
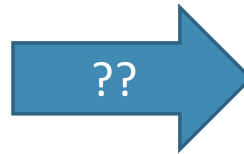
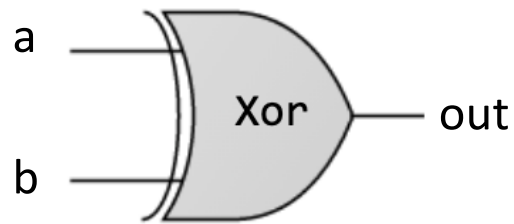
songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Lecture plan

- 1.1 บูลีน ลอจิก
- 1.2 การสังเคราะห์ฟังก์ชันบูลีน
- 1.3 ลอจิกเกต
- **1.4 ภาษา HDL**
- 1.5 โปรแกรมจำลอง Hardware Simulation
- 1.6 โค้ด HDL แบบ Multi-Bit Buses
- 1.7 โปรเจ็ค 1

สร้างลอจิกเกต

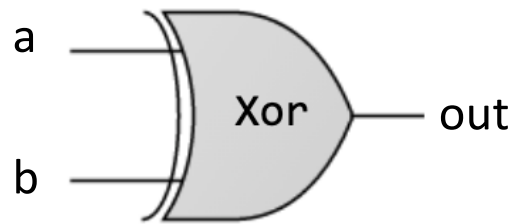


Output เป็น 1 เมื่ออินพุต
ตัวใดตัวหนึ่ง เป็น 1 เท่านั้น

กระบวนการผลิตลอจิกเกต

- ออกแบบคุณสมบัติลอจิกเกต
- กำหนดคุณลักษณะด้วยภาษา HDL
- ทดสอบการทำงานด้วยการจำลองผ่านซอฟต์แวร์จำลอง
- ปรับปรุงประสิทธิภาพ
- สร้างอุปกรณ์จริงในซิลิคอน

Design: ออกแบบคุณสมบัติลอจิกเกต



out เป็น 1 เมื่ออินพุตตัวใดตัวหนึ่ง เป็น 1 เท่านั้น

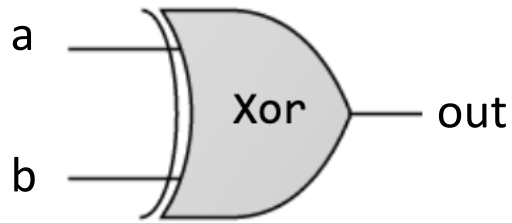
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

requirements

```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b) */  
  
CHIP Xor {  
    IN a, b;  
    OUT out;  
  
    PARTS:  
        // implementation missing  
}
```

ส่วนติดต่อ
interface

Design: จากคุณสมบัติเขียนเป็นไดอะแกรม

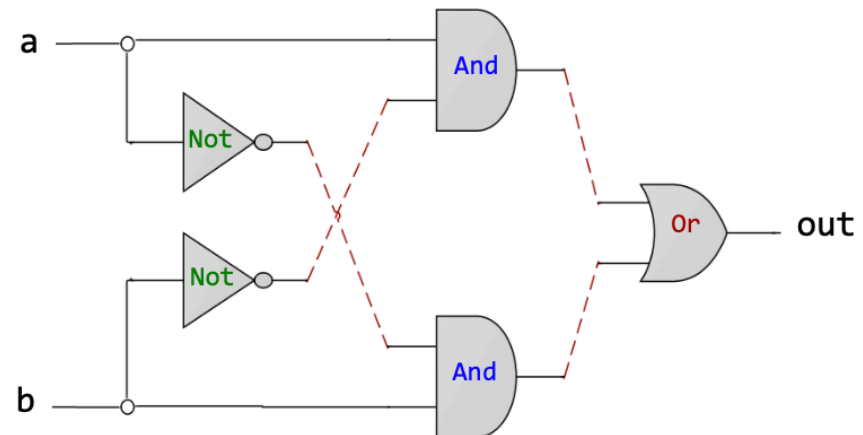
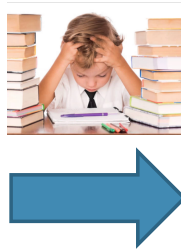


out เป็น 1 เมื่ออินพุตตัวใดตัวหนึ่ง เป็น 1 เท่านั้น

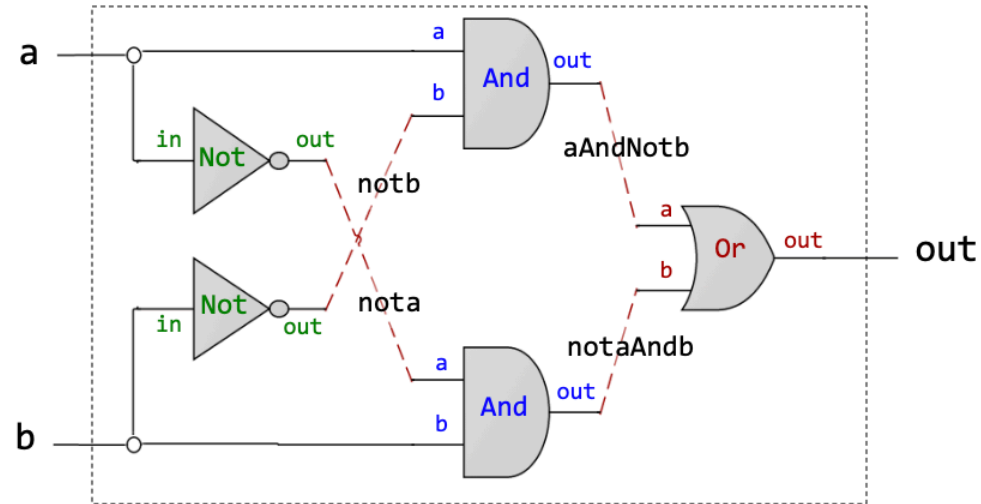
a	b	out
0	0	0
0	1	1
1	0	1
1	1	0

- ไอเดีย

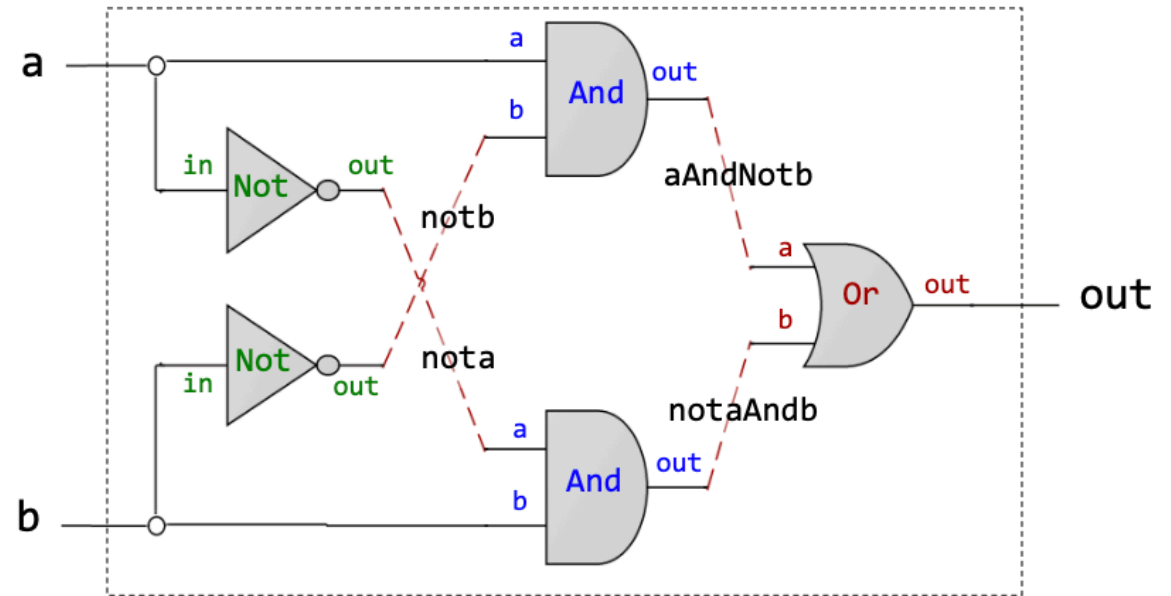
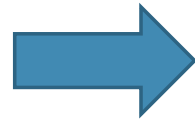
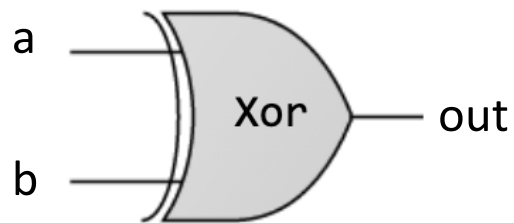
- out=1 เมื่อ
(a And Not(b))
Or
(b And Not(a))



Design: จากโต๊ะแถมเป็น HDL



```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b) */  
  
CHIP Xor {  
  IN a, b;  
  OUT out;  
  
  PARTS:  
    // implementation missing  
}
```



ส่วนติดต่อ
interface

ส่วนการ
ทำงาน

```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b) */
```

```
CHIP Xor {  
  IN a, b;  
  OUT out;
```

```
  PARTS:
```

```
  Not (in=a, out=nota);
```

```
  Not (in=b, out=notb);
```

```
  And (a=a, b=notb, out=aAndNotb);
```

```
  And (a=nota, b=b, out=notaAndb);
```

```
  Or (a=aAndNotb, b=notaAndb, out=out);
```

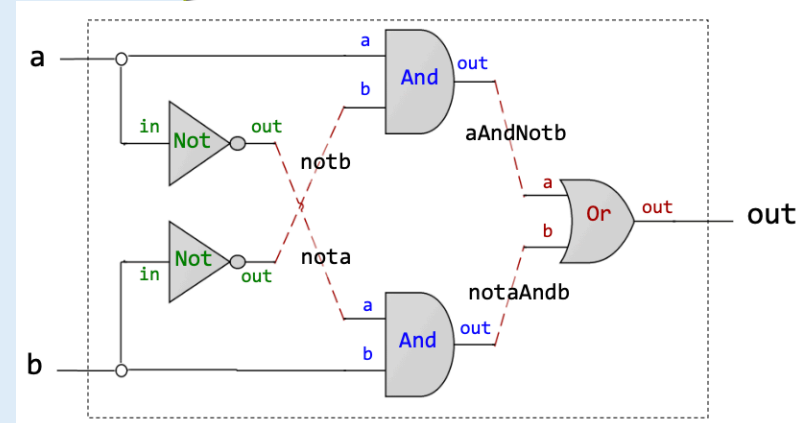
```
}
```

Other Xor
implementations
are possible!

HDL: some comments

```
/** Xor gate: out = (a And Not(b)) Or (Not(a) And b) */
```

```
CHIP Xor {  
  IN a,b;  
  OUT out;  
  
  PARTS:  
    Not (in=a, out=nota);  
    Not (in=b, out=notb);  
    And (a=a, b=notb, out=aAndNotb);  
    And (a=nota, b=b, out=notaAndb);  
    Or (a=aAndNotb, b=notaAndb, out=out);  
}
```



- HDL เป็นฟังก์ชัน / เป็นภาษาคอมพิวเตอร์รูปแบบหนึ่ง
- ลำดับการวางคำสั่ง ไม่มีผลต่อลำดับการทำงาน
- ก่อนใช้ส่วน Parts ควรทราบ interface ของชิป
Not(in= ,out=), And(a= ,b= ,out=), Or(a= ,b= ,out=)
- ต่อสายระหว่างชิปมีชื่อตรงกัน พบได้ปกติ เช่น And(a=a, ...) Or(..., out=out)

Hardware description language

- ภาษา HDL ที่เป็นที่รู้จัก
 - VHDL
 - Verilog
 - และ อื่นๆ
- ภาษา HDL ที่ใช้ในวิชานี้ เป็นอีกรูปแบบหนึ่ง
 - มีคอนเซ็ปต์ ใช้วิธีคิดแบบเดียวกับ HDL ทั่วไป
 - ทำความเข้าใจง่าย
 - ออกแบบให้มีคำสั่งน้อยที่สุด
 - เพียงพอต่อการใช้ในการเรียนวิชานี้
 - www.nand2tetris.org / HDL Survival Guide