



การจัดองค์การคอมพิวเตอร์

Hardware simulations

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 2 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิติศรีวรพันธุ์

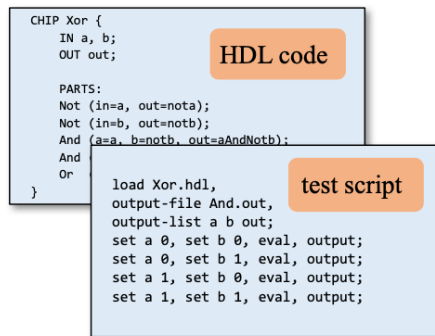
songrit@npu.ac.th

สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

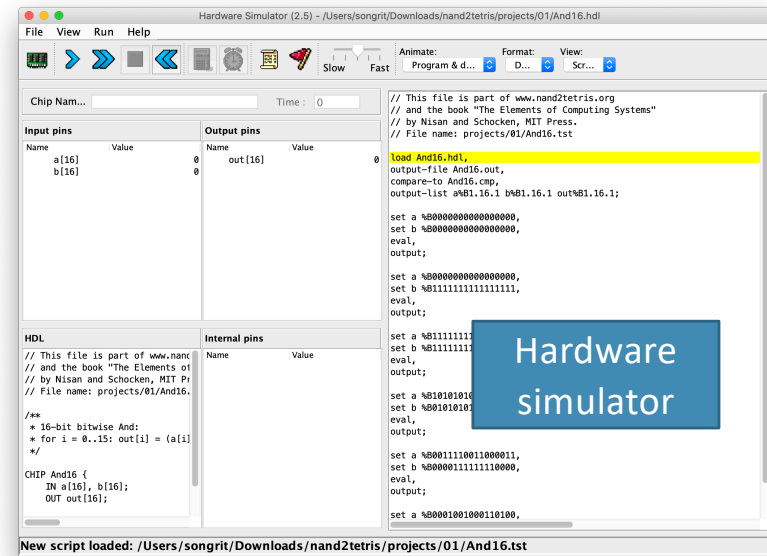
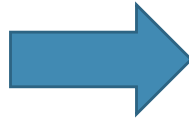
Lecture plan

- 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

การใช้โปรแกรม Hardware sim



โหลดเข้าโปรแกรม

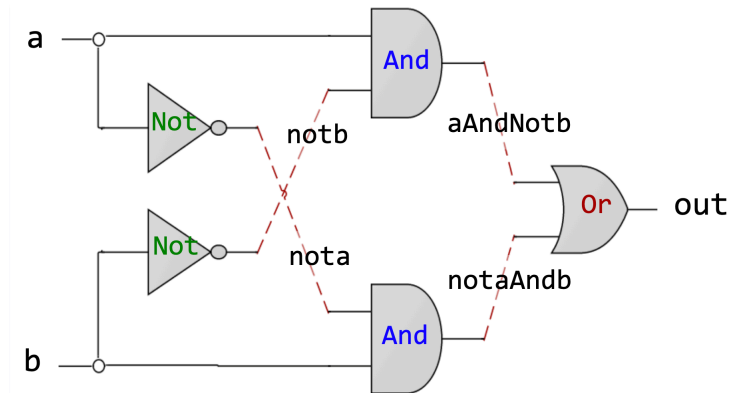


- เขียนโค้ดด้วยภาษา HDL
- คอมไฟล์และจำลองการทำงาน (Script based simulation)
- ตอบสนองต่อการเปลี่ยนอินพุตได้
- สามารถตรวจผลลัพธ์กับไฟล์ทดสอบได้

Interactive simulation

- Xor.hdl

```
/** 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);  
}
```



- เปิดโปรแกรม Hardware simulator
- โหลดโค้ด Xor.hdl ลงโปรแกรม Hardware simulator
- ทดลองเซตค่า 0 หรือ 1 กับอินพุต a, b
- ใช้สังเกตการทำงาน

Hardware Simulator (2.5) - /Users/songrit/Downloads/nand2tetris/projects/demo/Xor.hdl

File View Run Help

Animate: Program flow Format: D... View: Scr...

Chip Nam... Xor

2. สังจำลอง

1. เซตอินพุต

3. ผลลัพธ์

HDL

```
/**
 * Exclusive-or gate: true if e
 * a is false and b is true; fa
 * QUESTION: how can the simul
 * HDL implementations of the u
 * Answer: since the demo fold
 * files, the simulator reverts
 */

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

Internal pins

Name	Value
nota	1
notb	1
x	0
y	0

ค่าตัวแปรต่างๆ

HDL code

Demo

- เปิดเครื่องคอมพิวเตอร์
- ใช้โปรแกรมไปพร้อมกันได้



Hardware simulators

- สำหรับ Hardware simulator นี้
 - จำลองการทำงานระดับ Logic gate
 - รองรับรูปแบบภาษา HDL (ในวิชานี้)
 - มีฟังก์ชันครบถ้วนสำหรับวิชานี้
- ยังมีอีกหลายฟังก์ชันที่ไม่ได้กล่าวถึง
- ศึกษาเพิ่มเติม
 - www.nand2tetris.org / Hardware Simulator Tutorial

การใช้ไฟล์ช่วยตรวจสอบผล

Xor.hdl

```
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);  
}
```

tested
chip

- เพื่อให้มีระบบทดสอบอัตโนมัติ
- เพื่อใช้ทดสอบซ้ำได้ในอนาคต

Xor.tst

```
load Xor.hdl,  
output-file Xor.out,  
output-list a b out;  
set a 0, set b 0, eval, output;  
set a 0, set b 1, eval, output;  
set a 1, set b 0, eval, output;  
set a 1, set b 1, eval, output;
```

test
script

Xor.out

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

การจำลองพฤติกรรม HDL

```
CHIP Xor {  
  IN a, b;  
  OUT out;  
  
  BUILTIN Xor  
  // Built-in chip implementation,  
  // can execute in the hardware  
  // simulator like any other chip.  
}
```

built-in chip
implementation

- Behavioral simulation:
 - สามารถจำลอง chip logic เป็นรูปแบบภาษาคอมพิวเตอร์ได้

Xor.tst

```
load Xor.hdl,  
output-file Xor.out,  
output-list a b out;  
set a 0, set b 0, eval, output;  
set a 0, set b 1, eval, output;  
set a 1, set b 0, eval, output;  
set a 1, set b 1, eval, output;
```

test
script

Xor.out

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

Xor.cmp

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

การสร้างโปรเจกต์ทางฮาร์ดแวร์

- ส่วนเกี่ยวข้อง
 - สถาปัตยกรรมของระบบคอมพิวเตอร์
 - นักพัฒนา
- สถาปัตยกรรมฯ คือส่วนที่ระบุหน้าที่ logic gate
- นักพัฒนา คือผู้ทำส่วนที่ระบุไว้

- ชิปแต่ละตัวประกอบด้วยไฟล์ HDL ทำหน้าที่ดังนี้
 - **ไฟล์กำหนดการทำงาน (*.hdl)**
 - ไฟล์ทดสอบการทำงาน (*.tst)
 - ไฟล์เปรียบเทียบความถูกต้อง (*.cmp)
 - นักศึกษาไม่ต้องเขียนไฟล์ *.tst และ *.cmp เรามีให้

The developer's view

Xor.hdl

```
/** returns 1 if (a != b) */  
  
CHIP Xor {  
  IN a, b;  
  OUT out;  
  
  PARTS:  
    // Implementation missing  
}
```

stub
file

Xor.tst

```
load Xor.hdl,  
output-file Xor.out,  
compare-to Xor.cmp  
output-list a b out;  
set a 0, set b 0, eval, output;  
set a 0, set b 1, eval, output;  
set a 1, set b 0, eval, output;  
set a 1, set b 1, eval, output;
```

test
script

Xor.cmp

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

compare
file

- ใช้ไฟล์ทั้งสามร่วมกัน ทำให้เอกสารสะดวกขึ้น:
 - ไฟล์กำหนดการทำงาน (*.hdl)
 - ไฟล์ทดสอบการทำงาน (*.tst)
 - ไฟล์เปรียบเทียบความถูกต้อง (*.cmp)