w09-Lec2

Bitwise Operations

Assembled for 204111 by Kittipitch Kuptavanich

204111: Fundamentals of Computer Science

Bitwise Operations

• เราสามารถนำ Boolean Operator มาใช้กับ bit vector ซึ่งก็ คือตัวเลข 0 หรือ 1 จำนวน w ตัว

1000101.....011100101



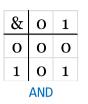
• ให้
$$a$$
 = $[a_{w ext{-}1}, a_{w ext{-}2}, \dots, a_{ ext{o}}]$

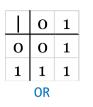
$$b = [b_{w-1}, b_{w-2}, \dots, b_{o}]$$

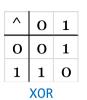
• ผลลัพธ์ของ $a \ \& \ b$ จะได้เป็น bit vector ที่มีความยาว w โดยที่ โดยในหลักที่ i จะมีค่าเท่ากับ $a_i \ \& \ b_i \ (o \le i < w)$

Operations of Boolean Algebra









- Operation ~ corresponds to the logical operation <u>NOT</u>, denoted by the symbol ¬
- Operation & corresponds to the logical operation <u>AND</u>, denoted by the symbol ∧
- Operation | corresponds to the logical operation <u>OR</u>, denoted by the symbol V
- Operation ^ corresponds to the logical operation <u>Exclusive-Or</u> (<u>XOR</u>), denoted by the symbol \bigoplus ($p \land q$ มีค่าเป็น True ก็ต่อเมื่อ p และ q มี<u>ตัวใดตัวหนึ่ง</u>เป็น True แต่ไม่ใช่ทั้งสองตัว ซึ่งก็คือ $0 \land 1$ หรือ $1 \land 0$)

Computer Systems: A Programmer's Perspective - Bryant & O'Hallaron

204111: Fundamentals of Computer Scien

Bitwise Operations [2]

<u>ตัวอย่าง</u>

- ให้ a = [0110] และ b = [1100]
- ในการทำ bitwise operation a & b, $a \mid b$, $a \land b$, และ $\sim b$

จะได้ผลลัพธ์ดังนี้



$$\begin{array}{c}
0110 \\
1100 \\
\hline
1110
\end{array}$$

$$\frac{1100}{1010}$$

0110

$$\sim \frac{1100}{0011}$$

2

ทำ Operation & ใน แต่ละหลักแยกกัน

Practice 1: Bitwise Operations

• เติมตารางให้สมบูรณ์ เพื่อ แสดงค่าที่ได้จากการทำ Bitwise Operation โดยมีค่า ของ bit vector a และ b ดัง กำหนด

Operation	Result
а	[01101001]
b	[01010101]
~ a	
~ b	-
a & b	
$a \mid b$	
$a \hat{b}$	

Computer Systems: A Programmer's Perspective - Bryant & O'Hallaron

cionco

5

204111: Fundamentals of Computer Science

Python bitwise ~ Operator [2]

- เช่นเดียวกันในกรณีจำนวนลบ เช่น -35
 - เขียนแบบ two's complement

• 35 = [010001<u>1]</u> ดังนั้น

• -35 = [1011101] กรณีนี้ไม่ต้องเพิ่ม MSB

• ~-35 จึงมีค่า [0100010] = 34

>>> ~-35
34
>>> int('0100010', 2)
34

- ค่า ~x จะมีค่าเท่ากับ -x 1 ทั้งจำนวน + และ 0
 - สังเกตการใช้ฟังก์ชัน int() เพื่อแปลงเลขฐาน 2 เป็น integer
 - 📍 เลข 2 เป็น optional parameter (default คือ ฐาน 10)

Python bitwise ~ Operator

- Python ใช้การแทนค่ำตัวเลขแบบ two's complement
 - จำนวนเต็มบวกใด ๆ มีค่า MSB เป็น 0
 - จำนวนเต็มลบใด ๆ มีค่า MSB เป็น 1
- พิจารณาค่า 3 (11) ในภาษา Python
 - การแทนค่าจึงอยู่ในรูป [011] เพิ่ม MSB = 0
- ดังนั้น ~3 ใน Python จึงมีค่า [100]
 - ตีความแบบ two's complement ได้เป็น -4

```
>>> ~3
-4
```

Think Python: How to Think Like a Computer Scientist

204111: Fundamentals of Computer Science

Bit-Level Operations in Python

- ใน Python เราสามารถทำ bitwise Boolean operation ได้ เช่นเดียวกัน โดยใช้ operator |, &, ~ และ ^
- โดย operation เหล่านี้สามารถใช้ได้กับ data type จำนวน เต็ม ตัวอย่างเช่น

Python expression	Binary expression	Binary result	2's Complement Value
~0x41	~[0100 0001]	[1011 1110]	
~0x00	~[0000 0000]	[1111 1111]	
0x69 & 0x55	$[0110\ 1001]$ & $[0101\ 0101]$	$[0100\ 0001]$	
0x69 0x55	[0110 1001] [0101 0101]	$[0111\ 1101]$	

เลขฐาน 16 ใน Python ขึ้นต้นด้วย <mark>ØX</mark>

Shift Operations in Python

- ใน Python ยังมี shift operation ที่ทำหน้าที่ เลื่อน bit vector ไป ทาง<u>ซ้าย</u>หรือ<u>ขวา</u> โดยใช้เครื่องหมาย << หรือ >>
- ตัวอย่างเช่น x << 4 หมายถึงการเลื่อน bit ใน x ไปทางซ้าย 4 ตำแหน่ง
- โดยปกติแล้วเมื่อมีการทำ left shift (<<) ตำแหน่งในทางด้าน<u>ขวา</u> ของ bit vector ที่ว่างลง จะถูกแทนด้วยเลขศูนย์

1100101<u>1</u> << 2

// shift ซ้ายไป 2 ตำแหน่ง



00101<u>1</u>00

Computer Systems: A Programmer's Perspective - Bryant & O'Hallaron

9

204111: Fundamentals of Computer Science

Practice 3: Shift Operations

 ให้เปลี่ยนเลขฐาน 16 ที่กำหนดให้ใน column ซ้ายสุดเป็น เลขฐาน 2 แล้วทำ shift operation ตามที่ระบุ จากนั้นให้ เปลี่ยนผลลัพธ์ที่ได้ จากเลขฐาน 2 กลับเป็นเลขฐาน 16

(Arithmetic)

X	x <	< 3	x >:	> 2
Binary	Binary	Hex	Binary	Hex
	Binary		Binary Binary Hex	Binary Binary Hex Binary

Shift Operations in Python [2]

- ในกรณีการทำ right shift (>>) จะต่างออกไปโดยจะเป็นการ ทำ Arithmetic Right Shift
 - Arithmetic Right Shift: ตำแหน่งที่ว่างลงทาง<u>ซ้าย</u>จะถูกแทนด้วย ค่าของ bit <u>ซ้ายสุด</u>ก่อนทำการ shift เพื่อคงเครื่องหมาย + หรือ - ไว้
- ค่าที่ได้จากการทำ left shift (x << n) จะเท่ากับ x *
- ค่าที่ได้จากการทำ right shift (x >> n) จะเท่ากับ x //

 Operation
 Values

 Argument x
 [01100011] [10010101]

 x << 4</td>
 $[0011\underline{0000}]$ $[0101\underline{0000}]$

x >> 4 (arithmetic) [00000110] [11111001]

Computer Systems: A Programmer's Perspective - Bryant & O'Hallaron

204111: Fundamentals of Computer Science

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	В	11
1100	С	12
1101	D	13
1110	E	14
1111	F	15

10