

HOOFDSTUK 5

BIT MANIPULATION

Helga Naessens

Bitoperatoren

operator	beschrijving	Resultaat (voor elke bitpositie)
&	Bitwise AND	1 als beide 1, 0 anders
	Bitwise OR	0 als beide 0, 1 anders
^	Bitwise XOR (Exclusive OR)	0 als beide 1 of beide 0, 1 anders
~	Bitwise NOT (1's complement)	0 wordt 1, 1 wordt 0
>>	shift right	(zie verder)
<<	shift left	(zie verder)

Operand(s): integer types (char, short, int en long)

Voorbeelden:

```
int main() {  
    int a = 12345, b = 67890;  
    int c = ~a; int d = a&b;  
    int e = a|b; int f = a^b;  
    return 0;  
}
```

➡

a =	00000000	00000000	00110000	00111001
b =	00000000	00000001	00001001	00110010
~a =	11111111	11111111	11001111	11000110
a&b =	00000000	00000000	00000000	00110000
a b =	00000000	00000001	00111001	00111011
a^b =	00000000	00000001	00111001	00001011

- Ga na of a oneven is:

`if (a & 1)`

- Wis alle bits van a,
met uitzondering van de laatste (= meest rechtse) 4 bits:

`a &= 0xF`

- Zet de voorlaatste bit van a op 1 (rest blijft ongewijzigd):

`a |= 2`

- Inverteer de voorlaatste bit van a (rest blijft ongewijzigd):

`a ^= 2`

- Zet de voorlaatste bit van a op 0 (rest blijft ongewijzigd):

`a &= ~2`

Shift left

verplaatst de bits *aantal_bits* naar links en
voegt rechts nullen toe ($= *2^{aantal_bits}$)

```
int a = 3; int i;  
for(i=1 ; i<3 ; i++)  
    a<<=i;
```

➡	a = 3	00000000	00000000	00000000	00000011
	a = 6	00000000	00000000	00000000	00000110
	a = 24	00000000	00000000	00000000	00011000
➡	a = -3	11111111	11111111	11111111	11111101
	a = -6	11111111	11111111	11111111	11111010
	a = -24	11111111	11111111	11111111	11101000

Shift right

verplaatst de bits *aantal_bits* naar rechts en
voegt links nullen/enen toe ($= /2^{aantal_bits}$)

- Types zonder teken:
steeds 0-en inschuiven
- Types met teken:
ofwel: tekenbit inschuiven
ofwel: 0-en inschuiven

computerafhankelijk

(JAVA: >> operator schuift 0-en in)

```
int a = 192; int i;  
for(i=1 ; i<3 ; i++)  
    a>>=i;
```

bit_vb.c

➡	a = 192	00000000	00000000	00000000	11000000
	a = 96	00000000	00000000	00000000	01100000
	a = 24	00000000	00000000	00000000	00011000
➡	a = -192	11111111	11111111	11111111	01000000
	a = -96	11111111	11111111	11111111	10100000
	a = -24	11111111	11111111	11111111	11101000

Oefeningen:

- Stop de voorlaatste bit van a in b:

$$b = (a \& 2) \gg 1$$

of
$$b = (a \gg 1) \& 1$$

- Gegeven een geheel getal g.

Schrijf een functie `int reverse(int g)` die (gebruik makend van bitoperatoren) als resultaat het omgekeerde hexadecimale getal teruggeeft.

Bv: als $g = \text{f12ab9}$ is het resultaat = 9ba21f .