BIT HACKS

Binary & Hexadecimal representation of integers
Set/Clear/Toggle kth bit
Extract/Set Bit Field

Marco Muñoz Otero

Table of contents

01 Binary and Hexadecimal Representation

02 Set/Clear/Toggle the kth bit

03 Extract/Set Bit Field

Binary Representation

Unsigned Integer

Formula & Example

$$x = \sum_{k=0}^{w-1} x_k 2^k$$

$$0b10000101 \rightarrow 133 = 1 + 4 + 128$$

Signed Integer

Formula & Example

$$x = \left(\sum_{k=0}^{w-2} x_k 2^k \right) - x_{w-1} 2^{w-1}$$

$$0b10000101 \rightarrow -123 = 1 + 4 - 128$$

$$-x = \sim x + 1.$$

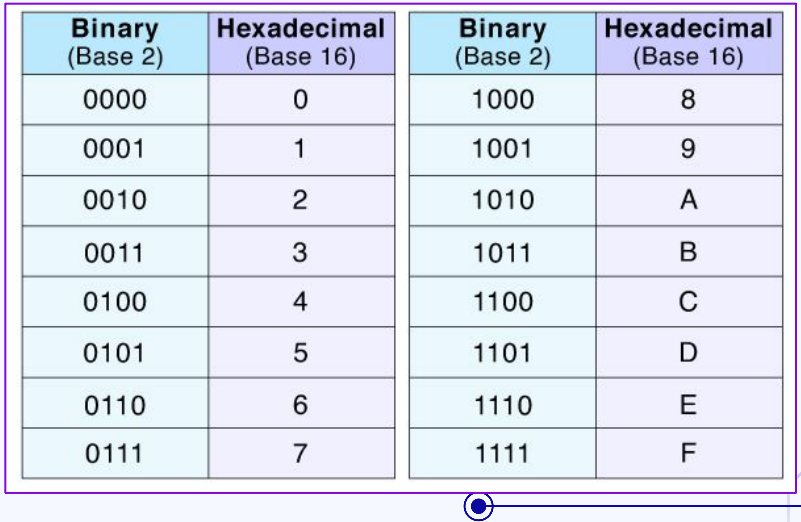
Hexadecimal Representation

Formula

$$x = \sum_{k=0}^{w-1} x_k 16^k$$

Example

$$0xA3_{16} = 10 \times 16^1 + 3 \times 16^0 = 160 + 3 = 163$$

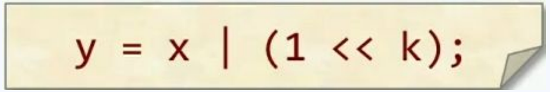


Binary (Base 2)	Hexadecimal (Base 16)	Binary (Base 2)	Hexadecimal (Base 16)
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Set the kth bit

Objective

Set kth bit in a word **x** to **1**.



```
y = x | (1 << k);
```

OPERATOR	DESCRIPTION
&	AND
|	OR
^	XOR
~	NOT
<<	SHIFT LEFT
>>	SHIFT RIGHT

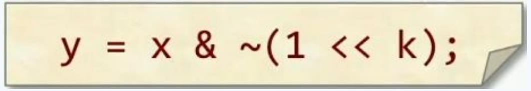
Example

k	3
x	0b 01100101
1 << k	0b 00001000
x | (1 << k)	0b 01101101

Clear the kth bit

Objective

Clear kth bit in a word **x**.



```
y = x & ~(1 << k);
```

OPERATOR	DESCRIPTION
&	AND
|	OR
^	XOR
~	NOT
<<	SHIFT LEFT
>>	SHIFT RIGHT

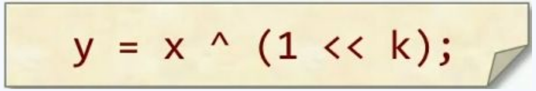
Example

k	2
x	0b 01100101
1 << k	0b 00000100
~(1 << k)	0b 11111011
x & ~(1 << k)	0b 01100001

Toggle the kth bit

Objective

Flip the kth bit in a word **x** to **1**.



```
y = x ^ (1 << k);
```

OPERATOR	DESCRIPTION
&	AND
|	OR
^	XOR
~	NOT
<<	SHIFT LEFT
>>	SHIFT RIGHT

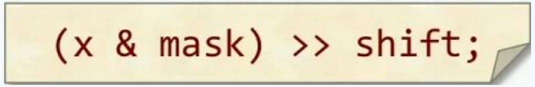
Example

k	**0**
x	**0b 01100101**
1 << k	**0b 00000001**
x ^(1 << k)	**0b 01100100**

Extract a bit field

Objective

Extract a bit field from a work **x.**



```
(x & mask) >> shift;
```

OPERATOR	DESCRIPTION
&	AND
|	OR
^	XOR
~	NOT
<<	SHIFT LEFT
>>	SHIFT RIGHT

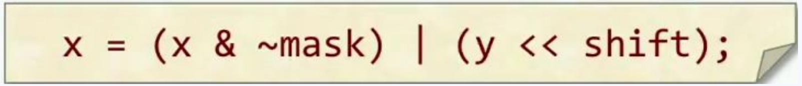
Example

shift	4
x	0b 01100101
mask	0b 11110000
x & mask	0b 01100000
x & mask >> shift	0b 00000110

Set a bit field

Objective

Set a bit in a word **x** to a value **y.**



```
x = (x & ~mask) | (y << shift);
```

<table>
<tr><th>OPERATOR</th><th>DESCRIPTION</th></tr>
<tr><td>&amp;</td><td>AND</td></tr>
<tr><td>|</td><td>OR</td></tr>
<tr><td>^</td><td>XOR</td></tr>
<tr><td>~</td><td>NOT</td></tr>
<tr><td>&lt;&lt;</td><td>SHIFT LEFT</td></tr>
<tr><td>&gt;&gt;</td><td>SHIFT RIGHT</td></tr>
</table>

Example

<table>
<tr><td>shift</td><td>7</td></tr>
<tr><td>x</td><td>0b 01100101001</td></tr>
<tr><td>y</td><td>0b 00000001100</td></tr>
<tr><td>mask</td><td>0b 11110000000</td></tr>
<tr><td>x &amp; ~mask</td><td>0b 00000101001</td></tr>
<tr><td>x = (x &amp; ~mask) | (y <&lt; shift)</td><td>0b 11000101001</td></tr>
</table>

Conclusion

Why bit hacks?

- Binary and Hexadecimal Representation
- Set,bit,toggle the kth bit
- Extract, set a field bit

There are more?

- Multiply by a power of 2
- Divide by a power of 2
- Check if a number is odd

More…

Webgraphie

https://catonmat.net/low-level-bit-hacks

https://youtu.be/ZusiKXcz_ac?si=oWYeFFysxxbOYmcg

https://www.geeksforgeeks.org/bit-tricks-competitive-programming/

Appendix

https://github.com/marcomunoz10/Advanced-Architectures/blob/BitHacks/bithacks.c

Thanks!

Do you have any questions?