

## NOTAZIONE POSIZIONALE

Ogni cifra assume un significato diverso a seconda della posizione in cui si trova.

Rappresentazione di un numero su  $n$  cifre in base  $b$

$$a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0$$
$$a_i \in \{0, 1, \dots, b-1\}$$

La  $b$  rappresenta la base da cui poi verranno scelti  $b$  simboli chiamati cifre. Ogni cifra va moltiplicata per un certo peso.

Il valore si calcola nel seguente modo

$$[a_{n-1} a_{n-2} a_{n-3} \dots a_1 a_0]_b = a_0 * 1 + a_1 * b + a_2 * b^2 + a_3 * b^3 + \dots + a_{n-1} * b^{n-1} = \sum_{i=0, 1, \dots, n-1} a_i * b^i$$

*Peso cifra i-esima*

### Notazione binaria

- Base 2  $\rightarrow$  2 cifre:  $\{0, 1\}$
- La cifra nella posizione  $i$ -esima ha peso  $2^i$

### Notazione esadecimale

- Base 16  $\rightarrow$  16 cifre:  $\{0, 1, 2, \dots, 9, A, B, C, D, E, F\}$
- Valore corrispondente delle cifre in decimale 0, 1, 2, ..., 9, 10, 11, 12, 13, 14, 15
- La cifra nella posizione  $i$ -esima ha peso  $16^i$
- Una cifra in base 16 è rappresentata mediante 4 cifre in base 2
- Per ottenere la rappresentazione in base 16 di un numero in base 2 si divide il numero binario in gruppi di 4 a partire da destra e sostituendo ad ogni gruppo la corrispondente cifra.
- Da base 16 in base 2 basta sostituire ciascuna cifra con la sua corrispondente rappresentazione in base 2.

### Intervallo di rappresentabilità

- Con  $N$  cifre in base  $b$  si possono rappresentare numeri naturali nell'intervallo  $[0, b^N - 1]$
- Il tipo naturale (unsigned) su  $N$  bit permette di rappresentare numeri naturali nell'intervallo  $[0, 2^N - 1]$

### Operatori bit a bit

Operano sui numeri naturali intesi come vettori di bit

Operatore	Simbolo	Arietà
AND bit a bit	&	Binario
OR bit a bit		Binario
XOR (OR esclusivo) bit a bit	^	Binario
complemento a 1	~	Unario
traslazione (shift) a sinistra	<<	Binario
traslazione (shift) a destra	>>	Binario

Supponendo di voler calcolare  $z = x \& y$  allora  $z_i = x_i \& y_i$  (vale anche per gli altri operatori).

## Operatore NOT

A differenza degli altri operatori il NOT ( $\sim$ ) è un operatore unario. Scambia tutti i bit di un vettore da 0 a 1 e viceversa.

Con la notazione  $\sim 0$  ci si riferisce a una configurazione con tutti i bit a 1.

## Traslazione

L'operatore di traslazione (shift) a sinistra/destra sposta ogni bit del primo operando di tante posizioni a sinistra/destra quante sono specificate dal secondo operando. Le posizioni lasciate libere in seguito alla traslazione sono riempite con bit a 0.

## Maschere di bit o maschere binarie

Corrispondono a configurazioni di bit usate per controllare/settare/resettare il valore di alcuni bit.

Per scoprire se il bit  $i$ -esimo di un vettore è o meno settato, basta fare l'AND con una maschera in cui solo tale bit è settato ( $1 \ll i$ ). Il risultato sarà diverso da 0 se e solo se l' $i$ -esimo bit è effettivamente settato.

```
Esempio    main() {
              unsigned int x = 53, maschera = 4;
              if ((x & maschera) != 0)
                  cout<<"Bit settato"<<endl;
              else
                  cout<<"Bit non settato"<<endl;
              }
```

Per scoprire se invece più di un bit sono settati basta usare una maschera  $y$  con tutti i bit da controllare settati.

- Se il risultato dell'AND è uguale ad  $y$  i bit sono tutti settati
- Se il risultato è diverso da 0 almeno un bit è settato

Per settare invece un bit in posizione  $i$ -esima basta effettuare l'OR con una maschera in cui solo tale bit è settato.

Per resettare un bit in posizione  $i$ -esima basta effettuare un AND fra  $\sim(1 \ll i)$  e il vettore.

Per cambiare lo stato di un solo bit in un vettore di bit basta effettuare uno XOR con una maschera in cui solo tale bit è settato.

Gli operatori bit a bit non operano su float e double bensì solo su signed e unsigned int. Il risultato per signed e unsigned potrebbe non essere lo stesso in quanto hanno due rappresentazioni differenti.