

Lezione 2 – Codifica di numeri in complemento a 2

Architettura degli elaboratori

Modulo 1 – Fondamenti architetturali

Unità didattica 2 – Rappresentazione binaria
delle informazioni

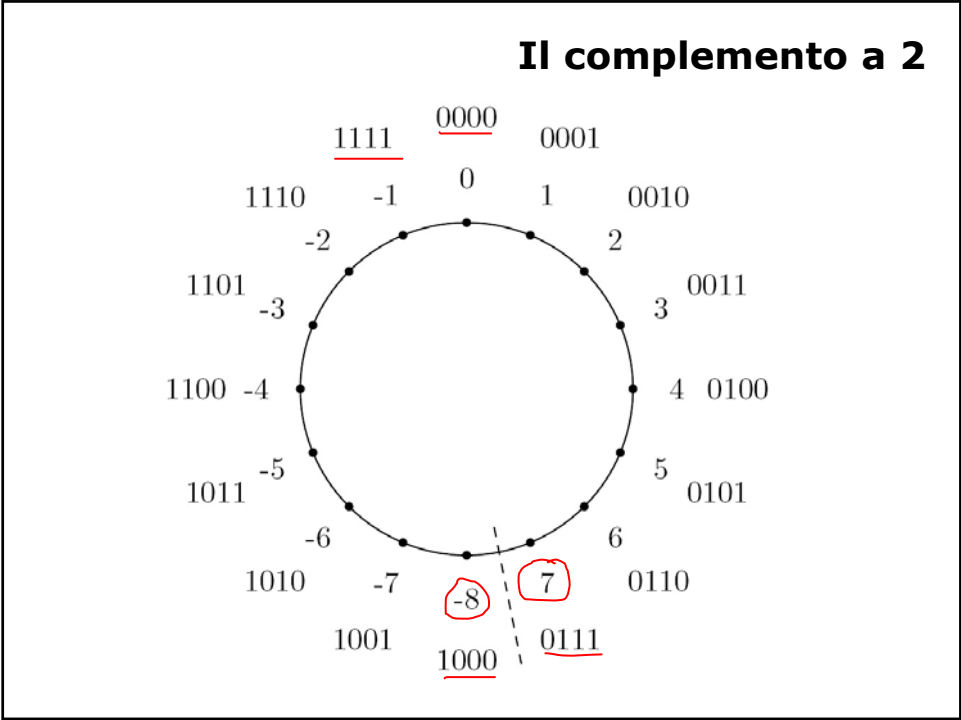
Nello Scarabottolo

Università degli Studi di Milano - Ssri - CDL ONLINE

La notazione in modulo e segno

Sebbene intuitiva, la notazione in modulo e segno presenta due difetti:

- ⊗ ogni operazione di somma e sottrazione dipende dal segno degli operandi:
 - se sono **concordi** (stesso segno) si procede;
 - se sono **discordi** (segno opposto) si devono scambiare le operazioni: per fare una somma si deve in realtà fare una sottrazione e viceversa;
 - in caso si debba fare una sottrazione, serve un **confronto** per stabilire minuendo e sottraendo;
 - il segno del risultato richiede un confronto dei moduli.
- ⊗ esistono **due rappresentazioni** per lo zero: +zero e –zero.



Caratteristiche del complemento a 2

A prima vista, il complemento a 2 ha le seguenti caratteristiche:

- i numeri negativi si distinguono dai positivi per il valore del bit più significativo:
 0 = numeri positivi;
 1 = numeri negativi;
- lo zero ha un'unica rappresentazione:
 0 000
- l'intervallo di numeri ***n*** rappresentabili con ***b*** bit non è più simmetrico: c'è un numero negativo in più ...

$$-2^{b-1} \leq n \leq 2^{b-1} - 1$$

Proviamo a fare le somme ...

Due numeri **positivi**

0101 ₂	+5 ₁₀	
+	0010 ₂	+2 ₁₀
rip. 0000		
=	0111 ₂	+7 ₁₀

Due numeri **negativi**

1100 ₂	-4 ₁₀	
+	1110 ₂	-2 ₁₀
rip. 100		
=	1010 ₂	-6 ₁₀

Due numeri **discordi**

0101 ₂	+5 ₁₀	
+	1010 ₂	-6 ₁₀
rip. 0000		
=	1111 ₂	-1 ₁₀

E le sottrazioni ?

La sottrazione è la somma del minuendo con l'**opposto del sottraendo**:

$A - B = A + (-B)$

Ecco come si trova l'opposto di un numero in complemento a 2:

Numero **positivo**

0101 ₂	+5 ₁₀	
inverto tutti i bit	1010 ₂	
sommo 1	0001 ₂	
=	1011 ₂	-5 ₁₀

Numero **negativo**

1101 ₂	-3 ₁₀	
inverto tutti i bit	0010 ₂	
sommo 1	0001 ₂	
=	0011 ₂	+3 ₁₀

Caratteristiche del complemento a 2

Il complemento a 2 ha dunque altre interessanti caratteristiche:

- la somma funziona correttamente sia fra numeri **concordi** sia fra numeri **discordi**;
- per cambiare il segno di un numero (quindi per fare la sottrazione) basta **invertire tutti i suoi bit** e **sommare al risultato il valore 1** (ancora somma ...);
- anche per codificare e decodificare i numeri non ci sono problemi:
 - se il numero è **positivo**, si usa la notazione posizionale;
 - se il numero è **negativo**, si **cambia segno** alla notazione posizionale.

E se vogliamo aumentare il numero di bit?

Numero **positivo**

su 4 bit: 0101₂ +5₁₀

su 8 bit: 0000 0101₂ +5₁₀

Numero **negativo**

su 4 bit: 1101₂ -3₁₀

su 8 bit: 1111 1101₂ -3₁₀

Basta **ripetere il bit più significativo** per estendere la lunghezza della notazione in complemento a 2.

Però attenzione al traboccamento ...

Due numeri **positivi**

	0101 ₂	+5 ₁₀
+	0110 ₂	+6 ₁₀
rip.	0100	
=	<u>1</u> 011 ₂	-5 ₁₀

Risultato **negativo** (overflow)

Due numeri **negativi**

	1100 ₂	-4 ₁₀
+	1001 ₂	-7 ₁₀
rip.	1000	
=	<u>0</u> 101 ₂	+6 ₁₀

Risultato **positivo** (underflow)

Si individua verificando che la **somma di numeri concordi** dia un risultato **concorde** con gli addendi.

In sintesi...

La notazione in **complemento 2**:

- elimina l'**ambiguità sulla codifica dello zero**;
- consente di utilizzare la **sola somma** per effettuare tutte le operazioni di somma e sottrazione;
- consente di usare la **sola somma** anche per la codifica e la decodifica dei numeri;

ma

- richiede di fare attenzione ai **traboccamenti** ...
- ... perché semplicemente continuando a sommare 1 a un numero positivo prima o poi ci si ritrova con un numero negativo !

