

Binary arithmetic
Operazioni aritmetiche fra numeri binari

Somma

$$0 + 0 \rightarrow 0$$

$$0 + 1 \rightarrow 1$$

$$1 + 0 \rightarrow 1$$

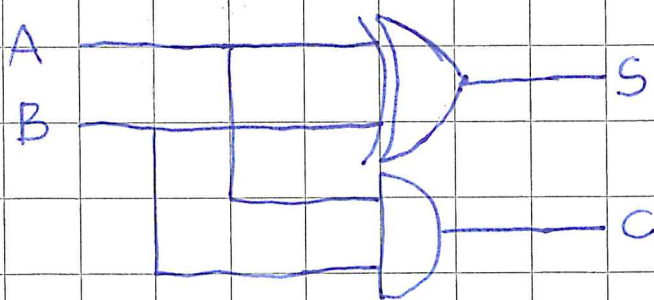
$$1 + 1 \rightarrow 0, \text{ riporto carry } 1 \text{ cioè } 0 \times 2^0 + 1 \times 2^1$$

stessa cosa del decimale

$$5 + 5 \rightarrow 0, \text{ carry } 1 \quad 0 \times 10^0 + 1 \times 10^1$$

$$7 + 8 \rightarrow 5, \text{ carry } 1 \quad 5 \times 10^0 + 1 \times 10^1$$

tornando al binario, posso gestire la somma con operazioni bitwise e con il circuito logico mostrato in questo diagramma e tabella di verità



Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Si chiama "half adder" *

* il "full adder" tiene conto anche del carry fra gli input

La rappresentazione dei numeri negativi con il MSB mi mette nei guai con le somme!

Es.

0	0	0	0	1	0	1	0	$\xrightarrow{8+2}$	1	0 ₁₀
1	0	0	0	1	0	1	0	\rightarrow	-	10 ₁₀

SOMMA

1	0	0	1	0	1	0	0	$\xrightarrow{16+4}$	-	20 ₁₀
---	---	---	---	---	---	---	---	----------------------	---	------------------

non va bene

RAPPRESENTAZIONI CORRETTE x NUMERI NEGATIVI

One's complement (complemento a uno)

Inverti tutti i bit, in questo modo ogni numero A sommato al suo complemento a uno \bar{A} mi da numero binario con "uni" in tutti i bit

A = 10 ₁₀	\rightarrow	0	0	0	0	1	0	1	0
\bar{A}		1	1	1	1	0	1	0	1
	+	<hr/>							
A + \bar{A}		1	1	1	1	1	1	1	1

Nota: $\bar{A} = 245_{10}$ che è uguale a $255 - 10$:

per un unsigned, il complemento a uno è il più grande numero disponibile con quei bit meno il numero stesso: max 8 bit $2^8 - 1 = 255$

Se invece prendo il complemento a uno ⁽³⁾
 unito alla convenzione del MSD
 per il segno

1 1 1 1 0 1 0 1
 ↑

sarebbe un
 numero negativo

Quale numero? -10_{10} perché
 il suo complemento a uno è $00001010 = 10_{10}$

In questo modo sistema $10 + (-10)$

0 0 0 0 1 0 1 0
 1 1 1 1 0 1 0 1

+
 1 1 1 1 1 1 1 1

è negativo,
 quindi lo complemento
 per capire cos'è

0 0 0 0 0 0 0 0

0_{10} ok

Proviamo ancora $3_{10} + (-1_{10})$

1 1 1 1 1 1 1 0 ← carry
 0 0 0 0 0 0 1 1
 1 1 1 1 1 1 1 0

+
 1 0 0 0 0 0 0 1
 ↑

→ $+1_{10}$

non è corretto

esce dagli 8 bit → se prendessi 9 bit sarebbe 257_{10}

riprovo con $3_{10} + (-2_{10})$

④

1 1 1 1 1 1 1
0 0 0 0 0 0 1 1
1 1 1 1 1 1 0 1

+
1 0 0 0 0 0 0 0 0 → + 0₁₀

↑ esce dagli 8 bit
se prendessi 9 bit
sarebbe 256₁₀

Sembra che ottengo
v sempre un numero sfornato
di 1

Come potrei risolvere il problema?

Sommando il carry finale 9^a posizione
alla prima posizione.

Svantaggio: devo usare un bit in più per tener
conto del carry.

Ad un certo momento i computer usavano
questo "trucco" (o meglio algoritmo)

Un algoritmo non necessita del bit in più x carry
maggiormente ottimizzato
è il complemento a due.

↳ invece
ha solo una
rappresentazione
per lo 0 e
quindi qualunque
1 numero: 8 bit
-128 a +127

TWO'S COMPLEMENT

Si ottiene facendo il complemento a uno
e si somma ancora 1.

Sommando un binario con il suo compl. a due trova tutti 0