

ALU

Friday, 18 August 2023

10:25

- Fa tutte le computazioni di un computer
- Ha dentro 2 parti:

- o ALU logica

Abbiamo 2 input, 1 multiplexor, e poi il multiplexor decide quale operazione fare

- o ALU Aritmetica

Per comprendere l'alu logica dobbiamo comprendere 2 parti importanti:

- Half adder

E' un addizione dove non consideriamo possibile resto iniziale.

$$0+0=0$$

$$1+0=1$$

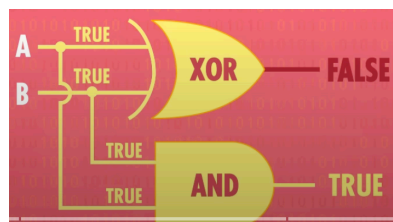
$$0+1=1$$

$$1+1=0 \text{ (con resto)}$$

Tabella:

INPUTS		OUTPUTS	
A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Circuito:



Lo XOR ci dà $1+1=0$

L'AND ci dà il resto

- Full adder

Qui noi aggiungiamo il resto all'input

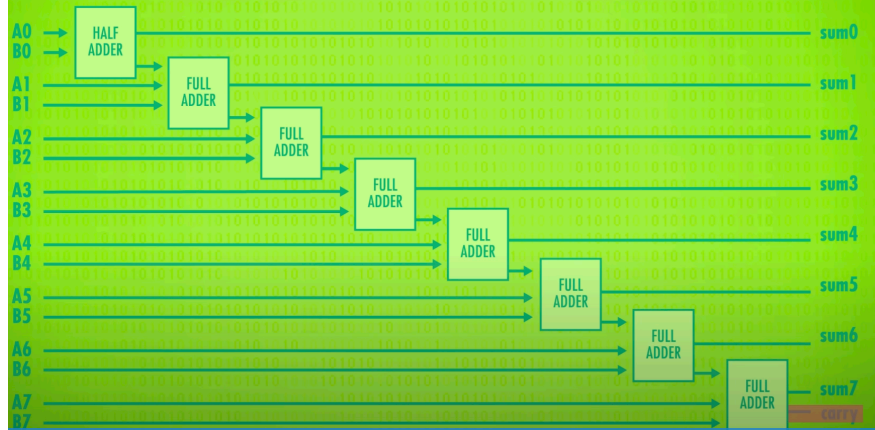
A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1

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

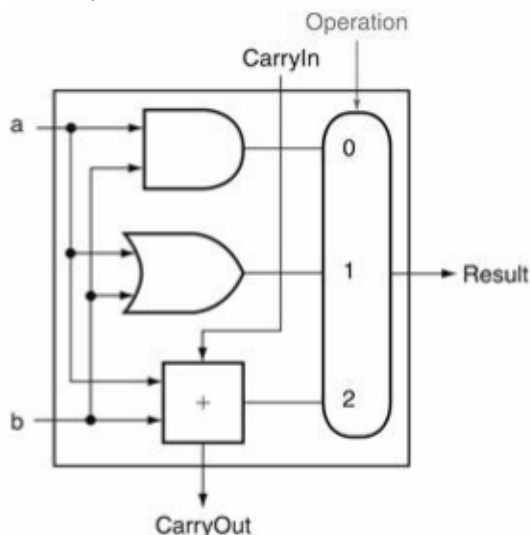
C=Resto di prima, che chiamiamo carry-in

Grazie alla combinazione di half e full adder è possibile creare una ALU

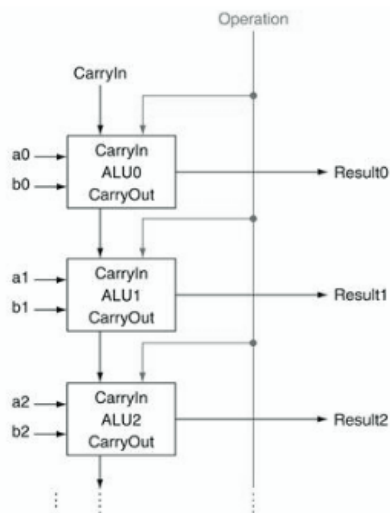
Esempio costruzione ALU da 8 bit:



Ora uniamo le due parti, e per unirle pasta che ci mettiamo un multiplexor dentro al nostro full/half-adder:

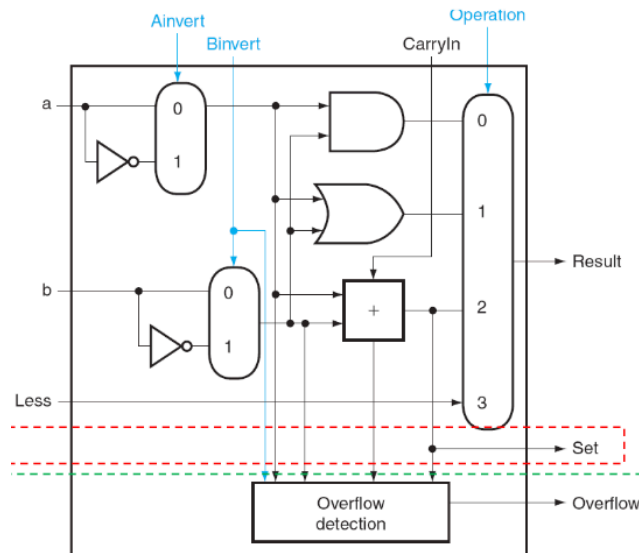


Ed abbiamo un ha alu che è capace di fare sia operazioni logiche che operazioni aritmetiche





Ora aggiungiamo la funzionalità di fare sottrazioni alla ALU, e per farlo ci creiamo un invertitore sia al primo input che al secondo. Dobbiamo aggiungerci un controllo di overflow e la possibilità che l'output sia negativo ($a-b < 0 \rightarrow \text{set} = \text{true}$)



E con questo abbiamo il circuito finale

