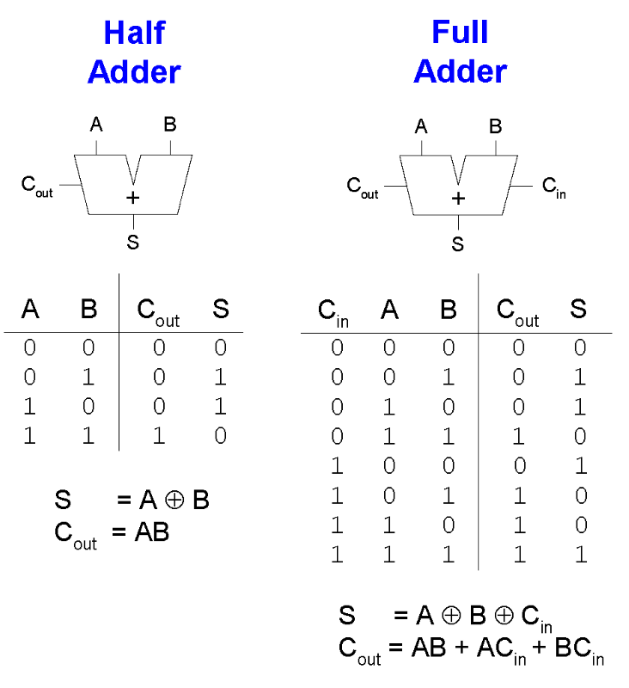
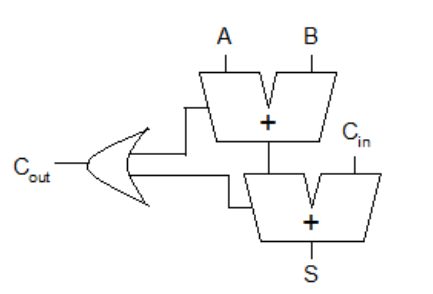
**Bit Adders**



Si fa la somma degli ingressi, il riporto va messo in il risultato in .

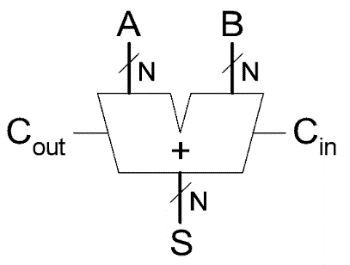
**Full adders from half adders**

Un full adders può essere composto partendo da 2 half adders, il risultato di un adders alla quale passiamo A e B viene passato come input ad un altro full adders insieme al valore utilizzato come . I 2 riporti di questi vengono confrontati in un OR, dando il riporto finale (). Il risultato di quest’ultimo vale come risultato(S).

**Multi-bit adders ()**

Tipi di sommatori a propagazione del riporto

* Ripple carry (lento)
* Carry look ahead (veloce)
* Prefix (velocissimo)

I carry look ahead ed I prefix sono più  
 veloci per gli adders di larga scala   
ma richiedono più hardware.

**Ripple carry adder**

Concatena insieme adders ad 1 bit. Il carry ripple passa attraverso l’intera catena.

Come svantaggio è però più lento rispetto agli altri.

Immagine che contiene testo, orologio

Descrizione generata automaticamente

Come si può notare, passiamo 2 input per ogni adder, più un al primo adder, il risultato del primo adder verrà salvato come , il riporto verrà passato come all’adder successivo, che lo ripasserà poi al prossimo. L’ultimo adder passerà il .

Immagine che contiene testo, clipart

Descrizione generata automaticamente*Delay temporale del ripple carry adder*

N è il numero degli adder, è il delay temporale per ogni adder.

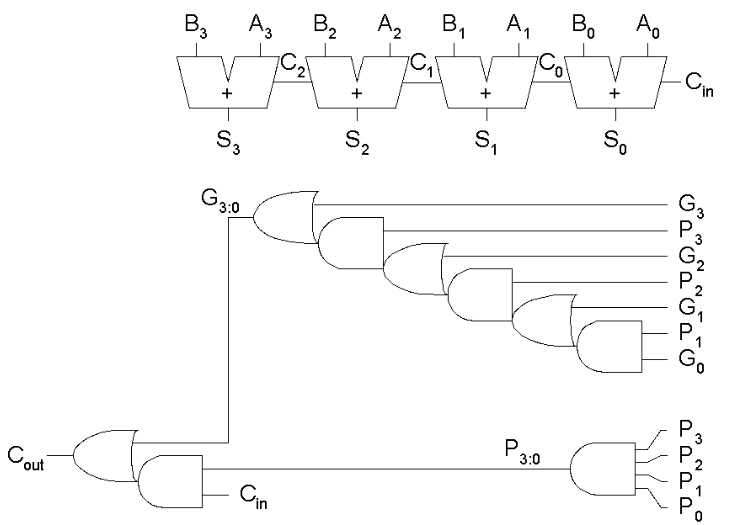
**Carry look ahead adder**

Calcola per k blocchi di bit adder usando segnali di generazione e propagazione. Se e valgono uno, si riceverà un riporto (*carry*).

La colonna i produce un riporto (*carry*) sia generandolo, sia propagandolo dal precedente riporto (*carry*).

Segnale generato Segnale propagato

Riporto :



Essendo che i valori G e P sono pre-calcolati, il percorso critico è ciò che parte da a .

sono i risultati della propagazione/generazione, essi sono pre-calcolati.

*Delay temporale del look ahead adder*

*Per N-bit adder con K-blocchi :*

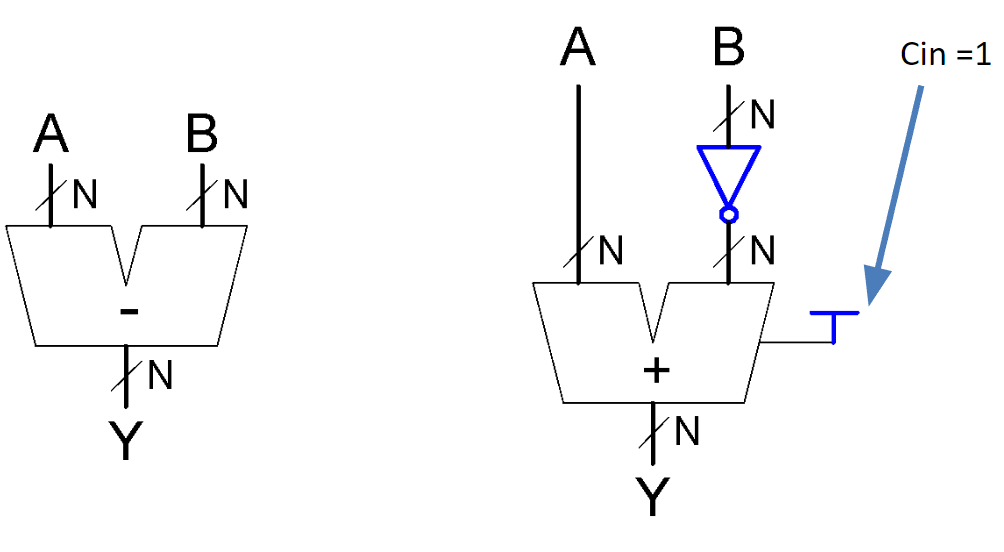


è il delay generato ogni , è il delay generato ogni

= Delay da a del blocco finale AND-OR.

Un look ahead adder è generalmente più veloce di un ripple carry adder quando il numero di adder da utilizzare è superiore a 16.

**Subtracter**



**Equality**

