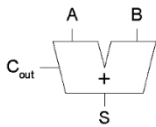


## Bit Adders

### Half Adder

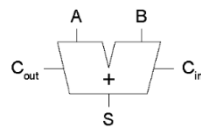


A	B	$C_{out}$	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$S = A \oplus B$$

$$C_{out} = AB$$

### Full Adder



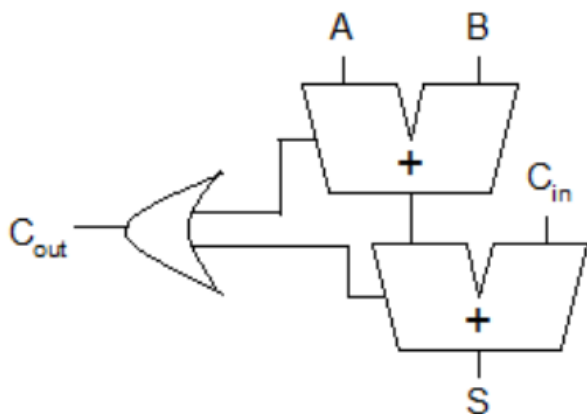
$C_{in}$	A	B	$C_{out}$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

Si fa la somma degli ingressi, il riporto va messo in  $C_{out}$  il risultato in S.

## Full adders from half adders

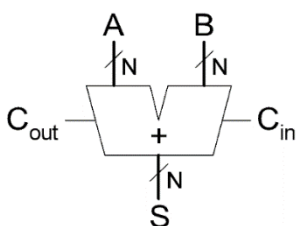


Un full adders può essere composto partendo da 2 half adders, il risultato di un adders alla quale passiamo A e B viene passato come input ad un altro full adders insieme al valore utilizzato come  $C_{in}$ . I 2 riporti di questi vengono confrontati in un OR, dando il riporto finale ( $C_{out}$ ). Il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

## Multi-bit adders ( $CPA_s$ )

Tipi di sommatori a propagazione del riporto

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

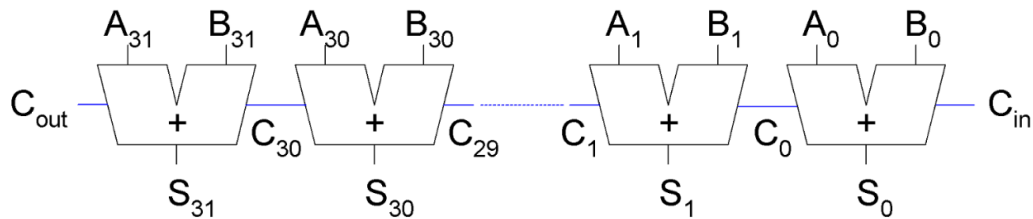


I carry look ahead ed I prefix sono più veloci per gli adders di larga scala ma richiedono più hardware.

## 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.



Come si può notare, passiamo 2 input per ogni adder, più un  $C_{in}$  al primo adder, il risultato del primo adder verrà salvato come  $S_0$ , il riporto verrà passato come  $C_{in}$  all'adder successivo, che lo ripasserà poi al prossimo. L'ultimo adder passerà il  $C_{out}$ .

### Delay temporale del ripple carry adder

$t_{\text{ripple}} = N t_{FA}$  N è il numero degli adder,  $t_{FA}$  è il delay temporale per ogni adder.

## Carry look ahead adder

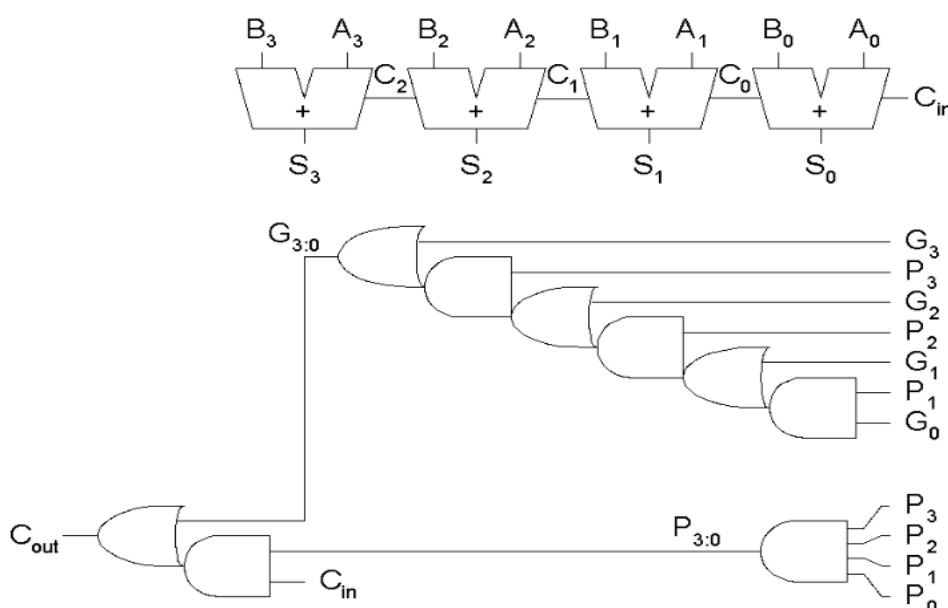
Calcola  $C_{out}$  per k blocchi di bit adder usando segnali di generazione e propagazione. Se  $A_i$  e  $A_b$  valgono uno, si riceverà un riporto (carry).

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

Segnale generato  $G_i = A_i B_i$

Segnale propagato  $P_i = A_i + B_i$

Riporto: 
$$C_i = A_i B_i + (A_i + B_i) C_{i-1} = G_i + P_i C_{i-1}$$



Essendo che i valori G e P sono pre-calcolati, il percorso critico è ciò che parte da  $C_{in}$  a  $C_{out}$ .

$G_i$  e  $P_i$  sono i risultati della propagazione/generazione, essi sono pre-calcolati.

### Delay temporale del look ahead adder

Per N-bit adder con K-blocchi :

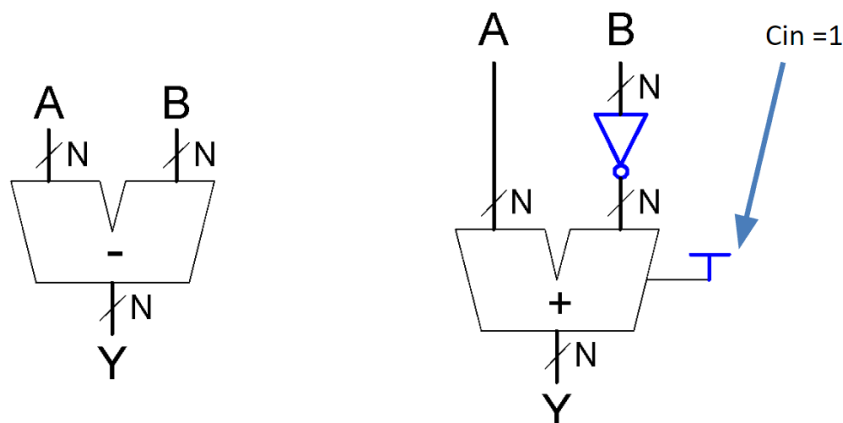
$$t_{CLA} = t_{pg} + t_{pg\_block} + (N/k - 1)t_{AND\_OR} + kt_{FA}$$

$T_{pg}$  è il delay generato ogni  $P_i$   $G_i$ ,  $T_{pg\_block}$  è il delay generato ogni  $P_{i:j}$   $G_{i:j}$

$T_{and\_or}$  = Delay da  $C_{in}$  a  $C_{out}$  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.

### Subtractor



### Equality

