



I sommatori

Prof. Alberto Borghese
Dipartimento di Informatica
borgnese@di.unimi.it

Università degli Studi di Milano

Riferimenti: Appendice B5 prima parte.



Sommario

Addizionatori

Addizionatori ad anticipazione di riporto



Implementazione di funzioni algebriche



And, Or, Not per ottenere:

Operazioni algebriche (somme, prodotti, sottrazioni e divisioni) su numeri binari.

Operazioni logiche su numeri binari.



Operazione di somma



$$\begin{array}{r} 111 \\ 1011 + \\ 110 = \\ \hline 10001 \end{array}$$

\leftarrow Riporto
 \leftarrow Addendo 1
 \leftarrow Addendo 2

3 Attori: addendo 1, addendo 2, riporto.

Viene eseguita sequenzialmente da dx a sx.

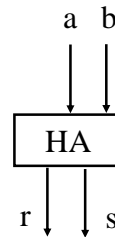


(Half) Adder ad 1 bit



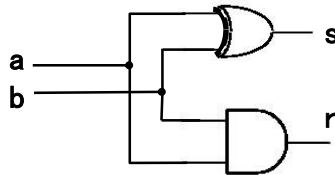
Tabella della verità della somma:

a b	somma	riporto
0 0	0	0
0 1	1	0
1 0	1	0
1 1	0	1



$$s = a \oplus b$$

$$r = ab$$



La somma è diventata un'operazione logica!

Cammini critici:
Somma = 1;
Riporto = 1;

Complessità
Somma = 1 porta;
Riporto = 1 porta;



Full Adder ad 1 bit



Tabella della verità della somma completa:

a b r_{in}	somma	riporto
0 0 0	0	0
0 1 0	1	0
1 0 0	1	0
1 1 0	0	1
0 0 1	1	0
0 1 1	0	1
1 0 1	0	1
1 1 1	1	1

$$s = m_1 + m_2 + m_4 + m_7$$

$$r = m_3 + m_5 + m_6 + m_7$$

$$s = \bar{a} \bar{b} r_{in} + \bar{a} b r_{in} + a \bar{b} r_{in} + a b r_{in} =$$

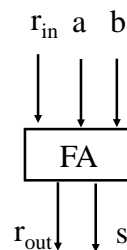
$$= (a \oplus b) r_{in} + (ab + ab) r_{in} =$$

$$= (a \oplus b) r_{in} + (a \oplus b) r_{in}$$

$$r_{out} = a b r_{in} + a \bar{b} r_{in} + a b r_{in} + a b r_{in} = ab + (a \oplus b) r_{in}$$

$$r_{out} = a r_{in} + (a \oplus r_{in}) b$$

Quale è meglio?

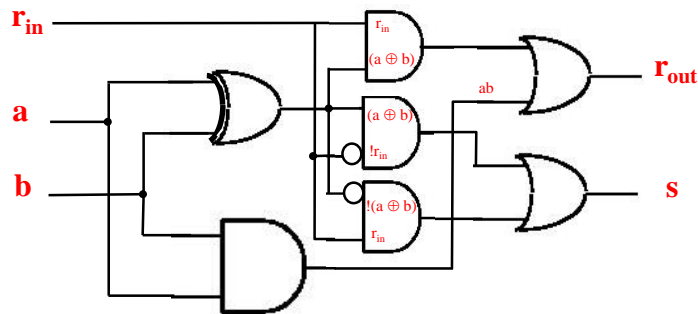




Implementazione circuitale



$$s = (a \oplus b) \overline{r_{in}} + (a \oplus b) r_{in}$$
$$r_{out} = ab + (a \oplus b) r_{in}$$



7 porte logiche.
Cammini critici: $s \rightarrow 3$; $r_{out} \rightarrow 3$



Complessità circuitale



- Definire la complessità circuitale e il cammino critico di HA:
 - $s = m1 + m2$
 - $r = m3$
- Definire la complessità circuitale e il cammino critico di FA:
 - $s = m1 + m2 + m4 + m7$
 - $r = m3 + m5 + m6 + m7$

Traccia: $m1$ è un circuito con 3 ingressi ed un'uscita e si può spezzare in due parte AND in cascata.



Semplificazione circuitale

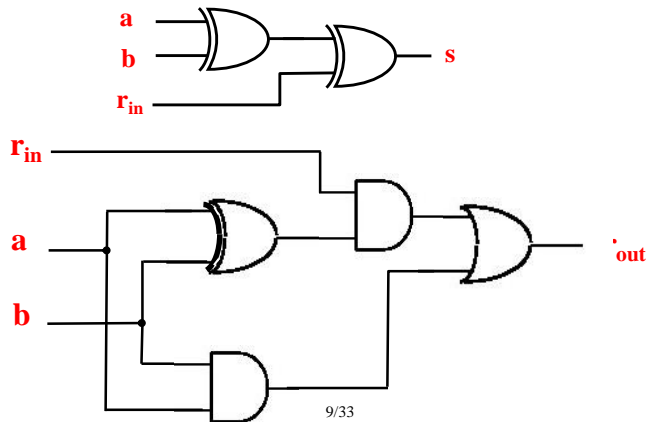


$$s = (a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} = (a \oplus b) \oplus r_{in}$$

$$r_{out} = ab + (a \oplus b)r_{in}$$

6 porte logiche.

Cammini critici: $s \rightarrow 2$; $r_{out} \rightarrow 3$



Semplificazione ulteriore

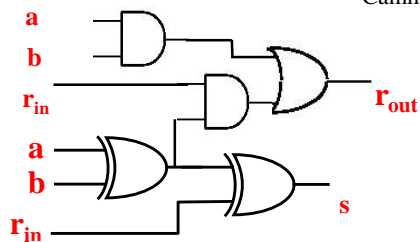


$$s = (a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} = (a \oplus b) \oplus r_{in}$$

$$r_{out} = ab + (a \oplus b)r_{in}$$

5 porte logiche.

Cammini critici: $s \rightarrow 2$; $r_{out} \rightarrow 3$



s - rilevatore di (dis)parità

r_{out} - riporto se generato ($a=b=1$) o se r_{in} propagato ($a \oplus b = 1$)



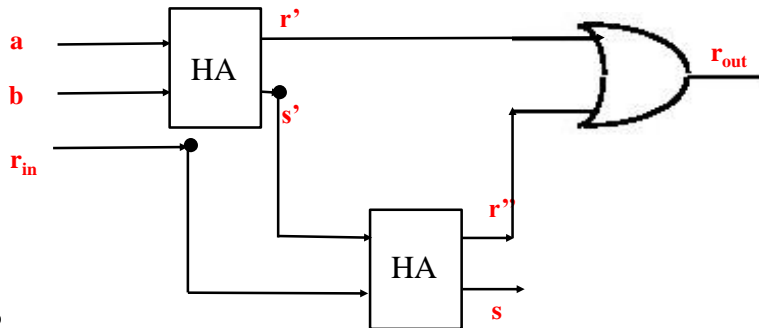
Circuito equivalente

$$s = (a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} = (a \oplus b) \oplus r_{in}$$

$$r_{out} = ab + (a \oplus b)r_{in}$$

5 porte logiche.

Cammini critici: $s \rightarrow 2$; $r_{out} \rightarrow 3$

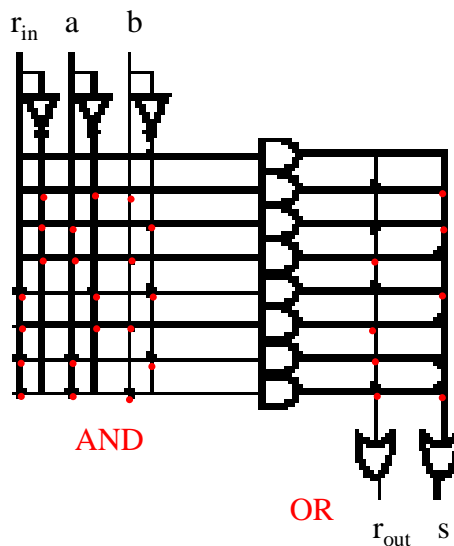


Raggruppo
AND e XOR in un HA



Implementazione mediante PLA

a	b	r_{in}	somma	r_{out}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1



SOP: costruisco i
mintermini e li sommo.



Esercizi con ROM e PLA



Implementare il circuito del Full Adder mediante ROM

Scrivere il circuito che esegue la somma di: $3 + 4$ in base 2.

Riportare tutte le uscite delle porte logiche.

Scrivere il circuito che esegue la seguente sottrazione: $5-2$ in base

2. Riportare tutte le uscite delle porte logiche.

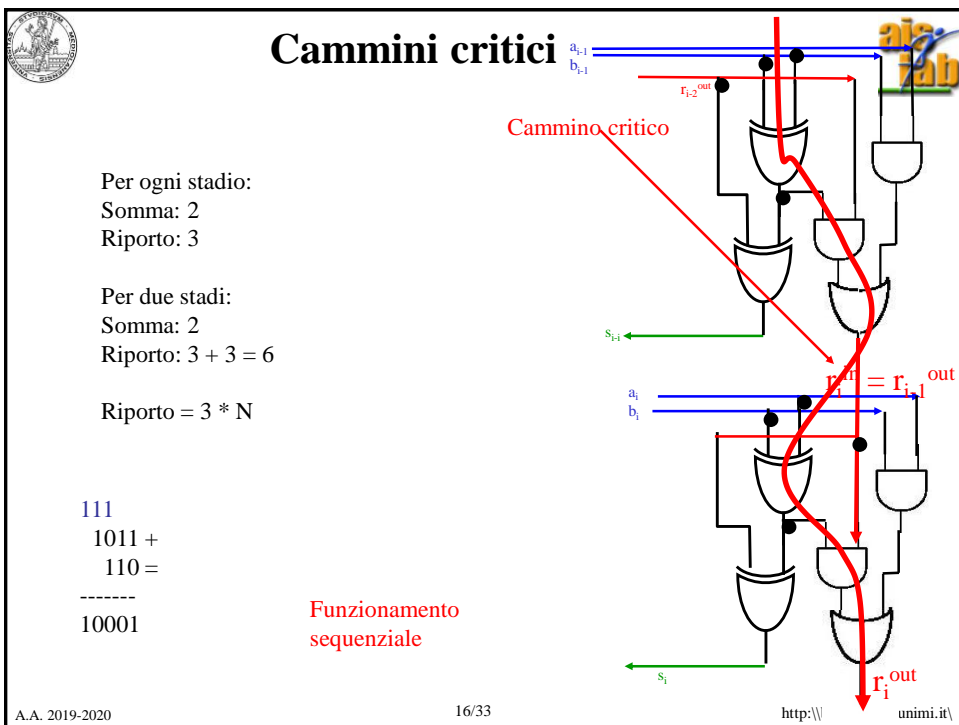
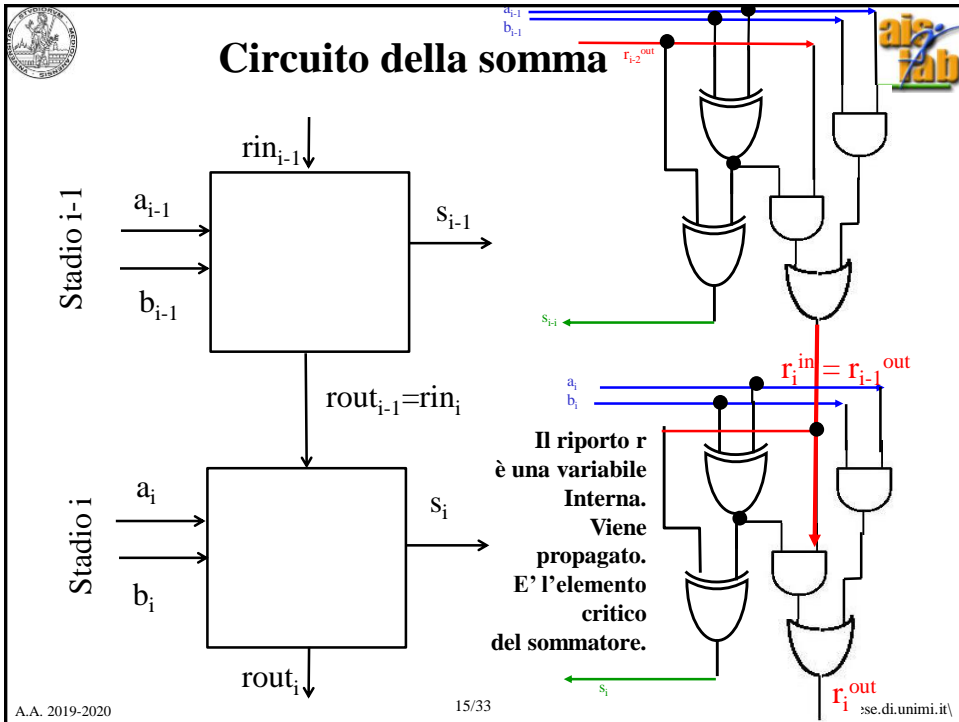


Sommario



Addizionatori

Addizionatori ad anticipazione di riporto





I problemi del full-adder



Il full adder con propagazione del riporto è lento:

- Il riporto si propaga sequenzialmente
caratteristica dell'algoritmo di calcolo
- la commutazione dei circuiti non è istantanea (tempo di commutazione)
caratteristica fisica dei dispositivi
- Soluzioni
modificare l'algoritmo
modificare i dispositivi



Prima possibilità: forma tabellare



Riscrivo le equazioni del riporto in modo non sequenziale. Come?

$$r_{out} = f(a_0, b_0, a_1, b_1, a_2, b_2, a_3, b_3, \dots)$$

Scrivo la tabella della verità dove in uscita ho gli N riporti ed
In ingresso $2 * N$ valori (gli N bit dei 2 addendi).

La tabella della verità ha 2^{2N} righe (per $N=32$, ...)



Carry look-ahead (anticipazione di riporto)



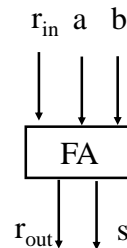
Approccio strutturato per diminuire la latenza della somma.

$$r_{out} = ab + (a \oplus b) r_{in}$$

Analisi del singolo stadio.

Quando si genera un riporto in uscita?

Quando ho almeno due 1, in ingresso; cioè tra r_{in} , a e b .



11000 riporto

1101 +

100 =

10001



Propagazione e generazione

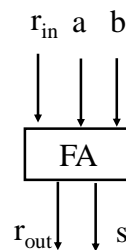


Ho riporto quando ho almeno due 1, in ingresso; cioè tra r_{in} , a e b .

Osservazioni:

- Viene generato un riporto dallo stadio i , qualsiasi sia il riporto in ingresso se $a = b = 1 \Rightarrow g_i = a_i b_i$.
- Viene generato un riporto allo stadio i , se il riporto in ingresso è $= 1$ ed una delle due variabili in ingresso è $= 1 \Rightarrow$ se $p_i = (a_i \oplus b_i) \Rightarrow$ viene generato riporto se $p_i r_i^{in} = 1$ (p_i propaga il segnale di riporto r_i^{in}).

Quando sia la condizione 1) che la condizione 2) è verificata?
Cosa succede se entrambe le condizioni sono verificate?





Esempio



Sono interessato ad r_4^{out} . Supponiamo $r_0^{in} = 0$.

$\begin{array}{r} r_{in} \quad 000000 \\ a \quad 10101101+ \\ b \quad \quad 10000= \\ \hline 10111111 \end{array}$	$\begin{array}{r} 0111000 \\ 10101101+ \\ 11010= \\ \hline 11100111 \end{array}$	$\begin{array}{r} 0111000 \\ 10111101+ \\ 11000= \\ \hline 11010101 \end{array}$
--	--	--

$$r_5^{in} = r_4^{out} = 0$$

$$r_5^{in} = r_4^{out} = 1$$

$$r_5^{in} = r_4^{out} = 1$$

Per propagazione

Per generazione

$$p_4 = (a_4 \oplus b_4)r_4^{in}.$$

$$g_4 = a_4b_4$$



Sviluppo della funzione logica riporto



$$r_i^{out} = ab + (a \oplus b)r_i^{in}$$

$$r_i^{out} = g_i + p_i r_i^{in}$$

$$r_0^{out} = g_0 + p_0 r_0^{in}$$

$$r_1^{out} = g_1 + p_1 r_1^{in} = g_1 + p_1 g_0 + p_1 p_0 r_0^{in}$$

$\begin{array}{r} r_1^{out} \rightarrow 111 \\ 1001 + \\ 10 = \\ \hline 1100 \end{array}$	$\begin{array}{r} r_1^{in} \rightarrow 110 \\ 1001 + \\ 11 = \\ \hline 1100 \end{array}$	$\begin{array}{r} r_1^{out} \rightarrow 10 \\ 1010 + \\ 11 = \\ \hline 1100 \end{array}$
---	--	--

$$g_0 = 0$$

$$p_0 = p_1 = 1$$

$$g_0 = 1$$

$$p_1 = 1$$

$$g_1 = 1$$



Sviluppo della funzione logica riporto



$$r_i^{out} = ab + (a \oplus b) r_i^{in}$$

$$r_i^{out} = g_i + p_i r_i^{in}$$

$$r_0 = g_0 + p_0 r_0$$

$$r_1 = g_1 + p_1 r_0 = g_1 + p_1 g_0 + p_1 p_0 r_0$$

$$r_2 = g_2 + p_2 r_1 = g_2 + p_2(g_1 + p_1 g_0 + p_1 p_0 r_0) = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 r_0$$

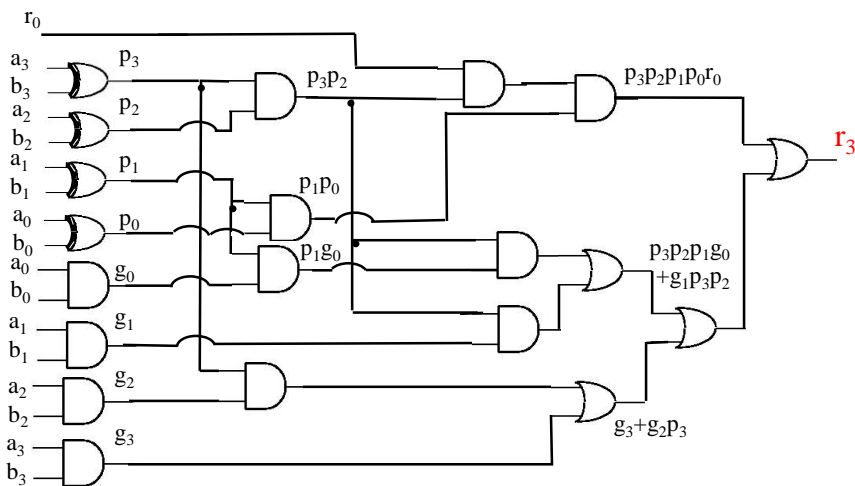
$$r_3 = g_3 + p_3 r_2 = g_3 + p_3(g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 r_0) = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 r_0$$



Determinazione del cammino critico.



$$r_3 = g_3 + p_3 r_2 = g_3 + p_3(g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 r_0) = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 r_0$$



Cammino critico = 6, senza anticipazione sarebbe $3 * 4 = 12$

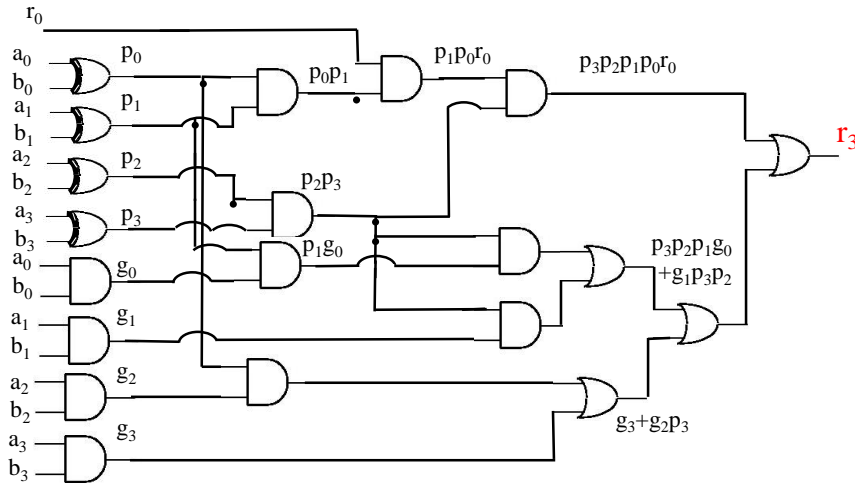


Determinazione la complessità.



$$r_3 = g_3 + p_3 r_2 = g_3 + p_3(g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 r_0) =$$

$$g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 r_0.$$



Complessità di $r_3 = 20$ porte logiche + porte per la somma e i riporti interni



Complessità aggiuntiva per gli altri bit di riporto



$$r_2 = g_2 + p_2 r_1 = g_2 + p_2(g_1 + p_1 g_0 + p_1 p_0 r_0) =$$

$$g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 r_0$$

Complessità aggiuntiva pari a 5 porte logiche.

$$r_1 = g_1 + p_1 r_0 = g_1 + p_1 g_0 + p_1 p_0 r_0$$

Complessità aggiuntiva pari a 2 porte logiche.

Complessità aggiuntiva totale: 7 porte logiche.



Complessità aggiuntiva per i bit di somma



$$s_k = (a_k \oplus b_k) \oplus r_{k\text{in}} = p_k \oplus r_{k\text{in}}$$

Ogni bit di somma aggiunge una porta logica XOR =>
La complessità aumenta di $N * 1 = 4$ porte logiche.

Un CLA su 4 bit ha quindi una complessità di 31 porte logiche.



Quanto si guadagna con l'anticipazione del riporto per N stadi?



Cammino critico per le variabili interne:

$$r_0^{\text{out}} \Rightarrow 3$$

$$r_1^{\text{out}} \Rightarrow 4$$

$$r_2^{\text{out}} \Rightarrow 5$$

Cammino critico per le variabili esterne:

$$r_3^{\text{out}} \Rightarrow 6$$

$$s_3 \Rightarrow 6 \quad \text{NB la prima porta XOR è in comune con } r_2^{\text{out}}$$

$$s_2 \Rightarrow 5 \quad \text{NB la prima porta XOR è in comune con } r_1^{\text{out}}$$

$$s_2 \Rightarrow 4 \quad \text{NB la prima porta XOR è in comune con } r_0^{\text{out}}$$

$$s_0 \Rightarrow 2$$

Cammino critico scala come $CC_{1\text{ stadio}} * \log(N)$



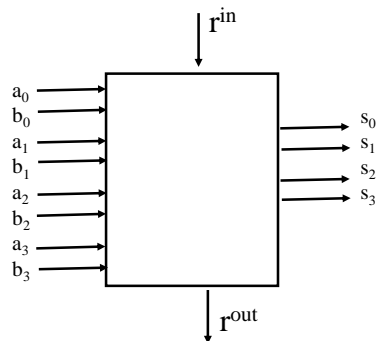
Addizionatori modulari



La complessità del circuito è tollerata per piccoli n .

Circuiti sommatore indipendenti si hanno per 4 bit.

Moduli elementari.



Come si ottiene la somma?

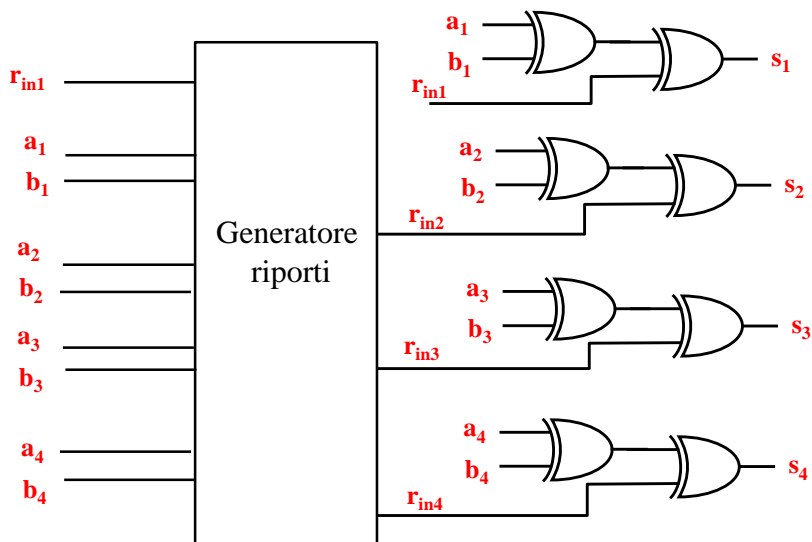
Collegando in cascata i moduli (sommatori elementari).

Cammino critico = $6 * N/4$. Per 32 bit, 48.

Per confronto, senza parallelizzazione, per 32 bit, $N * 3 = 96$.



Architettura interna



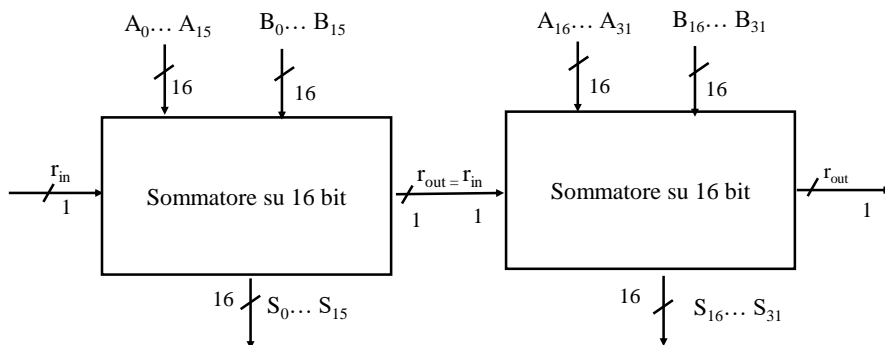


Addizionatori modulari::esempio



Occorre sommare 2 variabili, A e B, su $N = 32$ bit
Ho a disposizione due sommatore su 16 bit.

Come si ottiene la somma?
Fondamento delle estensioni architetturali SSE



AND e OR su più bit



1	0	0	1
---	---	---	---

AND

1	1	0	0
---	---	---	---

=

1	0	0	0
---	---	---	---

1	0	0	1
---	---	---	---

OR

1	1	0	0
---	---	---	---

=

1	1	0	1
---	---	---	---

Ogni bit viene elaborato separatamente



Sommario



Addizionatori

Addizionatori ad anticipazione di riporto