

Elettronica

Sistemi Digitali

13 / 08. Reti combinatorie particolari

Roberto Roncella



Componenti combinatori

Reti combinatorie particolari

(5.1.1, 5.3-5.8, 5.10)

Reti logiche per operazioni aritmetiche

Decoder ed encoder

Multiplexer

Dispositivi programmabili: PROM e PLA

Reti combinatorie particolari

- › Sono circuiti integrati disponibili in commercio come singolo componente
- › Due tipologie
 - › Realizzano funzioni tipiche di grande diffusione
 - › Funzioni aritmetiche per elaborazioni numeriche
 - › Somma-sottrazione, confronto, moltiplicazione
 - › Funzioni logiche universali
 - › Codifiche e decodifiche, commutatori multipolari
 - › La funzionalità può essere programmata dall'utente
 - › Grazie a elementi di connettività modificabili dopo la fabbricazione, come i **fusibili**

3

Sommatori binari

› Potrebbero essere sintetizzati come UNICA rete comb $2n \rightarrow n+1$

- › Fondamentali per avere **funzioni aritmetiche**
- › Eseguono l'algoritmo della somma binaria

$$x + y = \sum_{i=0}^{n-1} (x_i + y_i) 2^i = c_n 2^n + \sum_{i=0}^{n-1} s_i 2^i$$
 - › Un sommatore tra numeri a n cifre binarie è una rete combinatoria a $2n$ ingressi e $(n + 1)$ uscite
 - › In generale, la somma di **3 cifre binarie** è esprimibile con un **numero binario a 2 cifre**
 - › Infatti assume un valore tra 0 e 3
- › Un sommatore tra numeri a n cifre può essere realizzato con n sommatore da 3 bit
 - › Sono definiti **full-adder**

4

Il full adder (1)

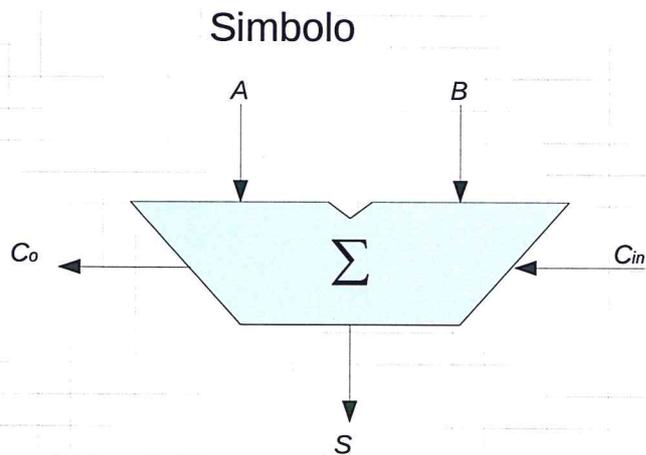


Tabella di verità

A	B	C _{in}	S	C _o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

5

Il full adder (2)

Sintesi

		AB			
		00	01	11	10
S	0	0	1	0	1
	1	1	0	1	0

		AB			
		00	01	11	10
C _o	0	0	0	1	0
	1	0	1	1	1

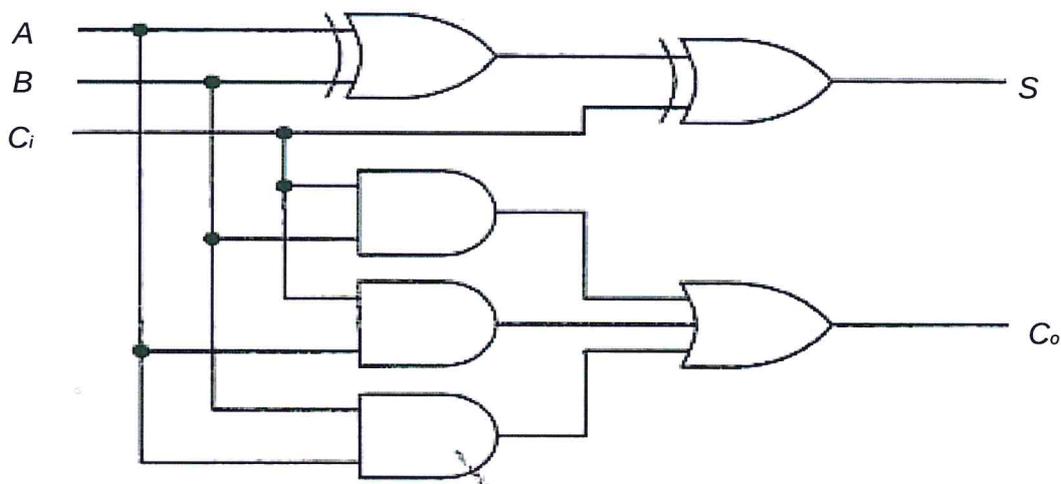
$$S = \bar{A}\bar{B}C_i + \bar{A}B\bar{C}_i + A\bar{B}\bar{C}_i + ABC_i = A \oplus B \oplus C_i$$

$$C_o = AB + AC_i + BC_i$$

6

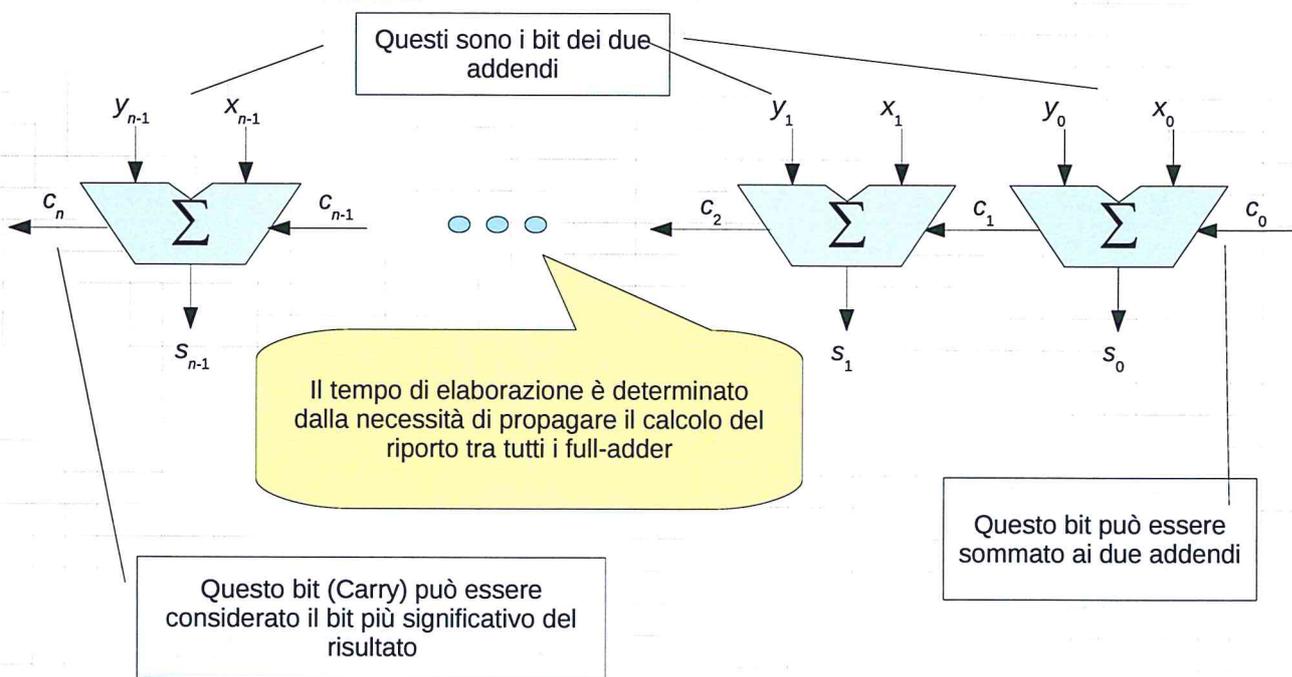
Il full adder (3)

Schema



i dati si presentano contemporaneamente

Sommatore parallelo ripple carry



Soluzione "iterativa" o "modulare"

Sottrattori binari

- › Si realizza allo stesso modo del sommatore
 - › Con n circuiti elementari **full-subtractor**
 - › Eseguono la differenza tra due bit, tenendo conto di un eventuale prestito rappresentato da un terzo bit in ingresso
 - › Il risultato varia da un minimo di -2
 - › Minuendo nullo, sottraendo e prestito a 1
 - › a un massimo di 1
 - › Minuendo a 1, sottraendo e prestito nulli
 - › Il risultato del full-subtractor è rappresentato dal bit della differenza e da un prestito in uscita di peso -2

9

Il full subtracter

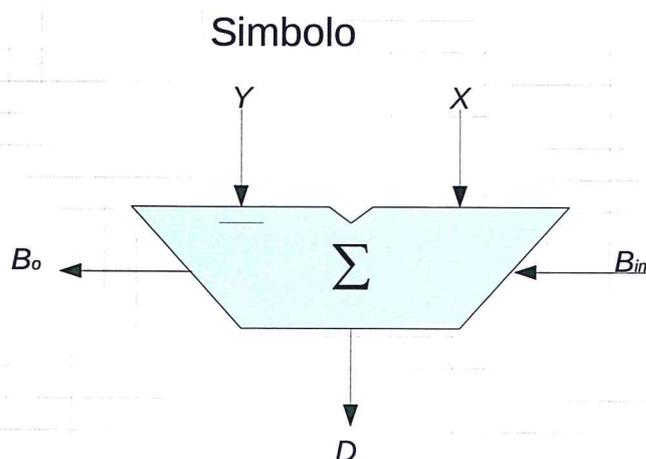
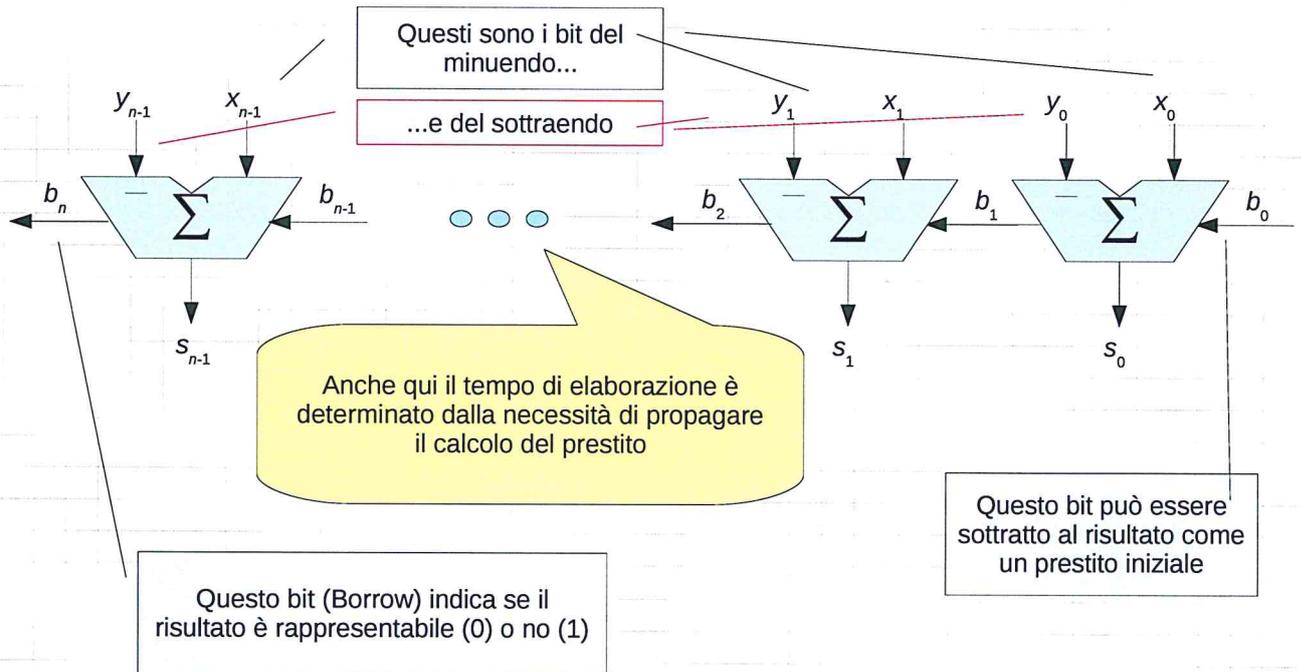


Tabella di verità

X	Y	B_{in}	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Sottrattore parallelo



11

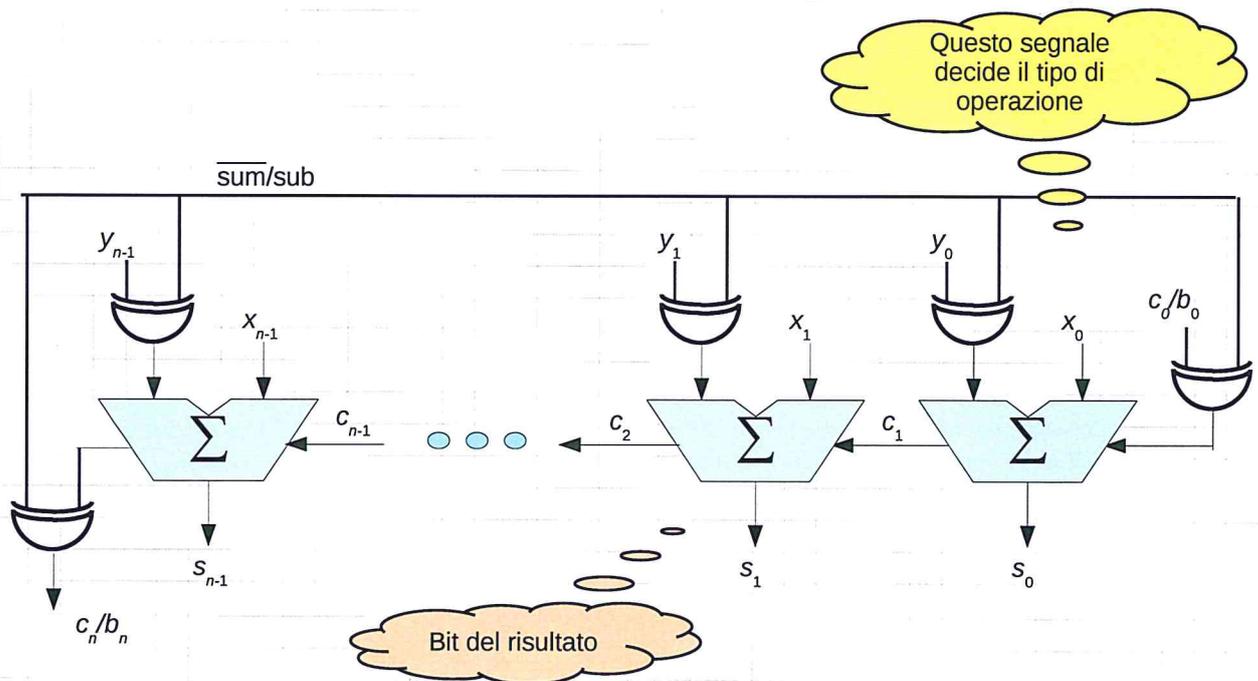
CAMBIARE SEGNO in C2 : INVERTIRE bit e bit e SOMMARE 1

Somma/sottrattori binari

- Sommatore e sottrattore possono essere fusi in **un solo circuito**
 - Il sommatore esegue $X + Y + c_0 = c_n 2^n + S$
 - Ponendo $Y = 2^n - 1 - W$ $c_0 = 1 - b_0$
 - Si ottiene $X + (2^n - 1 - W) + (1 - b_0) = c_n 2^n + S$
 - Da cui $X - W - b_0 = -(1 - c_n) 2^n + S = -b_n 2^n + D$
 - Che è l'espressione della differenza se $c_n = 1 - b_n$
- Quindi il sottrattore si ottiene dal sommatore negando bit a bit l'ingresso invertente e i bit dal riporto in ingresso e in uscita
 - Come invertitore comandato si usa la XOR

12

Somma-sottrattore parallelo



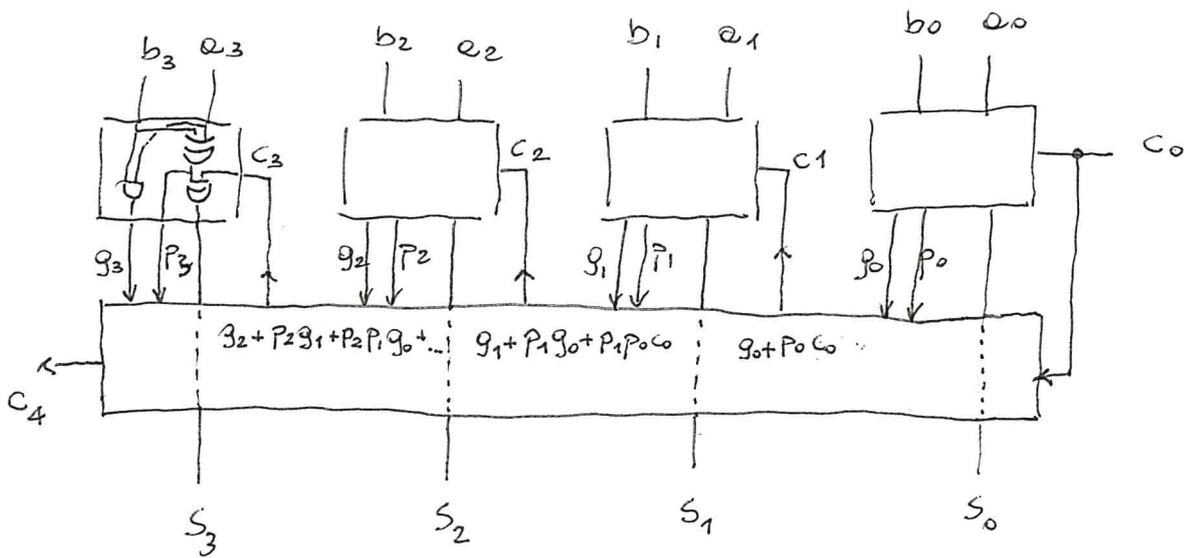
13

Il principio del CARRY look ahead * vedi appunti

Comparatori

- > Algoritmo del confronto (LSB 1st)
 - > Configurazione iniziale di **uguaglianza**
 - > Il confronto esamina due **bit corrispondenti**
 - > Se uguali propaga la configurazione di ingresso
 - > Se diversi
 - > $A_i \bar{B}_i$ propaga la configurazione $A > B$
 - > $\bar{A}_i B_i$ propaga la configurazione $A < B$
- > Modularità
 - > Aumenta però il tempo di propagazione
- > Realizzazioni alternative
 - > E possibile realizzare un comparatore MSB 1st
 - > A parte l'uguaglianza, la configurazione di ingresso **prevale** sull'esito del confronto i -esimo

14



Gli ingressi della rete che calcola i carry sono presenti dopo il RITARDO di una porta AND oppure OR/XOR

La rete calcola i carry in 2 ritardi base T_0 (per una porta elem.)

Le uscite sono pronte dopo un altro ritardo XOR

TOTALE $4T_0$

Limite dell'idea CARRY look ahead

- Le somme di prodotti aumentano di LETTERALI (ingombro e, in realtà, pure ritardo)
- Come andare verso n grandi? 16, 32 ~ 64 e oltre

a) Soluzione MODULARE "ripple carry" con gruppi "carry look ahead" di 4 oppure 8 bit

b) Applicazione GERARCHICA del principio CLA.

$$P = \prod_{i=0}^{m-1} p_i \quad \text{del modulo da } m \text{ bit}$$

$$G = g_m + p_m g_{m-1} + p_m p_{m-1} g_{m-2} \dots + p_m p_{m-1} \dots p_1 g_0$$

mi permettono di applicare l'idea a livello di MODULO, dando un pedice a P e G

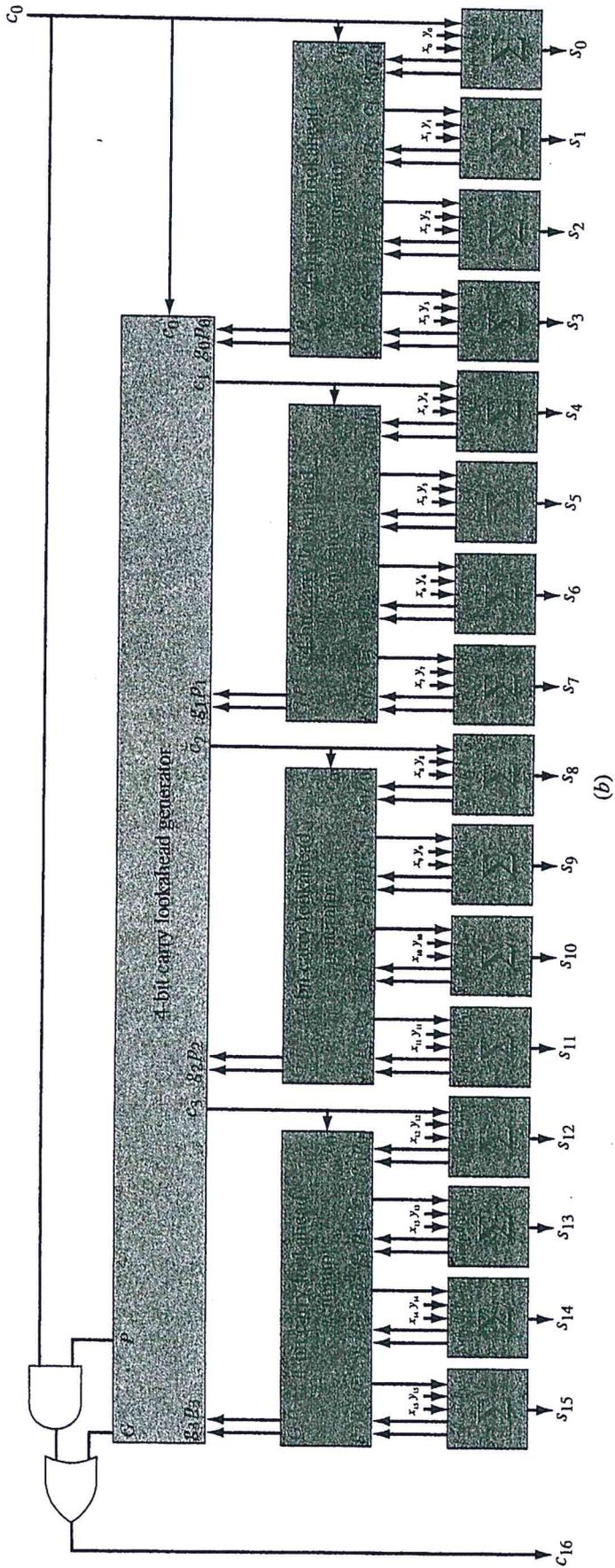
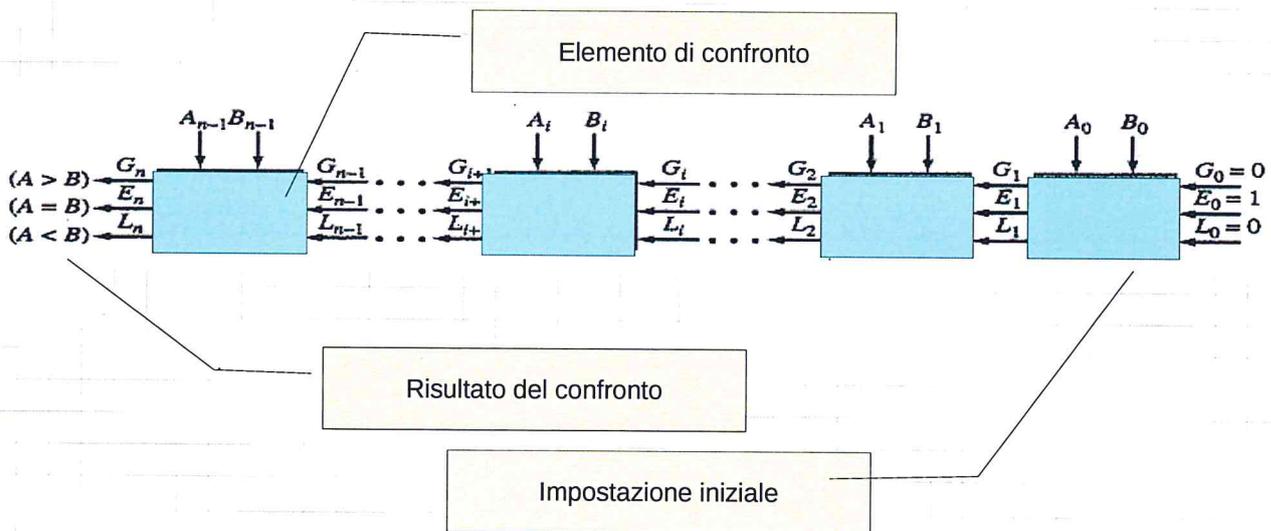


Figure 5.10 (b) a 16-bit high-speed adder.

Comparatore parallelo (LSB 1st)



L'elemento di confronto (1)

A_i	B_i	G_i	E_i	L_i	G_{i+1}	E_{i+1}	L_{i+1}	A_i	B_i	G_i	E_i	L_i	G_{i+1}	E_{i+1}	L_{i+1}
0	0	0	0	0	-	-	-	1	0	0	0	0	-	-	-
0	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0
0	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0
0	0	0	1	1	-	-	-	1	0	0	1	1	-	-	-
0	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
0	0	1	0	1	-	-	-	1	0	1	0	1	-	-	-
0	0	1	1	0	-	-	-	1	0	1	1	0	-	-	-
0	0	1	1	1	-	-	-	1	0	1	1	1	-	-	-
0	1	0	0	0	-	-	-	1	1	0	0	0	-	-	-
0	1	0	0	1	0	0	1	1	1	0	0	1	0	0	1
0	1	0	1	0	0	0	1	1	1	0	1	0	0	1	0
0	1	0	1	1	-	-	-	1	1	0	1	1	-	-	-
0	1	1	0	0	0	0	1	1	1	1	0	0	1	0	0
0	1	1	0	1	-	-	-	1	1	1	0	1	-	-	-
0	1	1	1	0	-	-	-	1	1	1	1	0	-	-	-
0	1	1	1	1	-	-	-	1	1	1	1	1	-	-	-

Tabella di verità