

---

# Progetto di Circuiti Aritmetici

Maurizio Palesi

## Introduzione

---

### ■ Caratteristiche principali di valutazione

- Velocità
  - ✓ Valutata per il caso peggiore
- Costo
- Precisione
  - ✓ Es., operazioni in virgola mobile
- Affidabilità
  - ✓ Codici di rilevazione e correzione di errore
- Consumo di potenza

## Realizzazione

---

- Due diversi aprocci
  - Approccio hardware
  - Approccio firmware

## Presentazione dei Dati

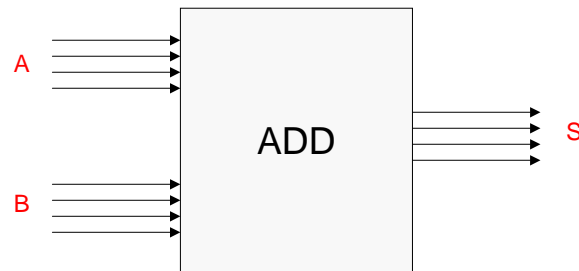
---

- Parallelo
- Seriale pura
- Seriale a byte
- Mista

## Presentazione dei Dati

### ■ Parallelo

→ Tutti i bit degli operandi sono presentati agli ingressi *simultaneamente*

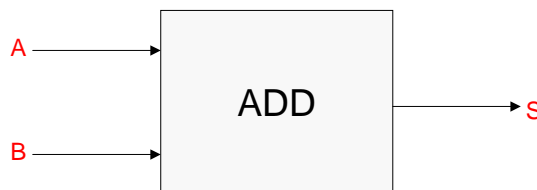


## Presentazione dei Dati

### ■ Seriale pura

→ Alle linee di entrata vengono presentati, *sequenzialmente* nel tempo (in *serie*) i bit di pari posizione dei due addendi

- ✓ Presentazione *normale* (dal meno significativo al più significativo)
  - Little endian
  - Sommatore, moltiplicatori
- ✓ Presentazione *seriale on-line* (dal più significativo al meno significativo)
  - Big endian
  - Divisori



## Presentazione dei Dati

---

### ■ Seriale a byte

- Per *byte* qui si intende un gruppo di pochi bit (non necessariamente 8)
- Es., Se i due addendi hanno ognuno 32 bit, questi vengono divisi in otto gruppi di 4 bit
  - ✓ Le due coppia di 4 bit di pari posizione vengono applicate contemporaneamente

## Presentazione dei Dati

---

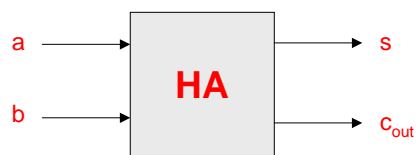
### ■ Mista

- Uno dei due operandi viene presentato con modalità in parallelo, l'altro con modalità seriale

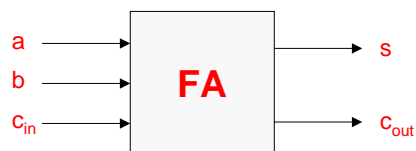
# Presentazione dei Dati

- Parallelo (combinatorio)
  - La più usata per operandi di lunghezza “normale” (16, 32, o 64 bit)
- Seriale e mista (sequenziale)
  - Utilizzata per operandi molto lunghi
    - ✓ Es., operazioni crittografiche

# Half/Full Adder

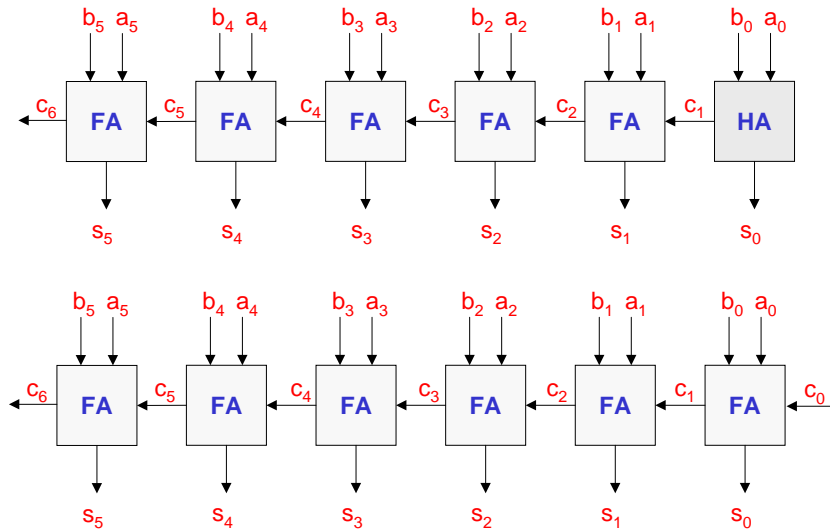


a	b	s	c <sub>out</sub>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



a	b	c <sub>in</sub>	s	c <sub>out</sub>
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

## Sommatore a Propagazione di Riporto



Maurizio Palesi

11

## Overflow

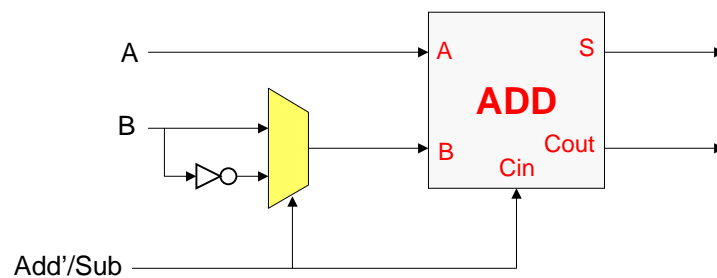
- L'overflow nella somma di due interi a  $n$  bit si ha se il risultato supera  $2^{n-1}$ 
  - Tale situazione viene segnalata dal valore 1 del riporto in uscita del FA in posizione più significativa

Maurizio Palesi

12

## Differenza

- $A - B = A + (-B) = A + C2(B)$   
→  $C2(B) = \text{NOT}(B) + 1$

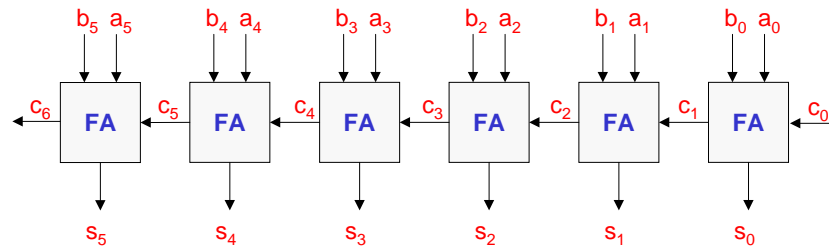


## Overflow nelle Somme Algebriche

- Dati due numeri a  $n$  bit di **segno diverso**  
→ Il risultato è **sempre corretto** e quindi occorre **ignorare** il riporto in uscita dello stadio più significativo
- Se i due numeri hanno lo stesso segno è possibile avere overflow  
→ Come capire se c'è stato overflow?
  - ✓ La somma di due numeri negativi è positiva oppure
  - ✓ La somma di due numeri positivi è negativa

$$\text{Overflow} = (a_{n-1}b_{n-1})s_{n-1} + (a_{n-1}\bar{b}_{n-1})s_{n-1}$$

## Velocità del Sommatore a Propagazione del Riporto



- Esiste un percorso dei segnali dalla posizione meno significativa a quella più significativa tramite i segnali di riporto
- Se  $\tau$  è il ritardo di propagazione dagli ingressi alle due uscite del FA, il ritardo di propagazione massimo è  $n\tau$

## Sommatori *Carry-Lookahead*

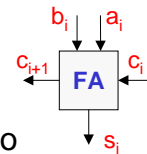
- Nella generica cella  $i$ -esima di un sommatore a propagazione di riporto, il riporto in uscita  $c_{i+1}$  deriva da

→ Una componente generata localmente

- ✓ Vale 1 se i bit  $i$ -esimi degli addendi valgono 1

→ E una propagata dovuta al riporto di ingresso

- ✓ Vale 1 se  $c_i=1$  e se almeno uno dei bit  $i$ -esimi degli addendi è 1



$$c_{i+1} = G_i + P_i c_i$$

$\swarrow$                        $\swarrow$   
 $a_i b_i$                        $a_i \oplus b_i$

## Sommatori *Carry-Lookahead*

$$c_{i+1} = G_i + P_i c_i = a_i b_i + a_i \oplus b_i c_i$$

- Il procedimento può essere iterato su  $c_i$

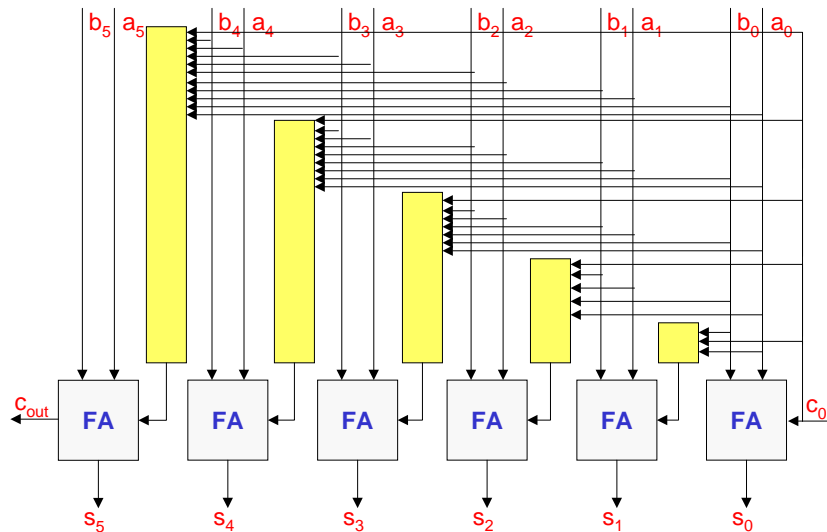
$$c_i = G_{i-1} + P_{i-1} c_{i-1} = a_{i-1} b_{i-1} + a_{i-1} \oplus b_{i-1} c_{i-1}$$

- e, riportando questa espressione in quella che fornisce  $c_{i+1}$  si ricava

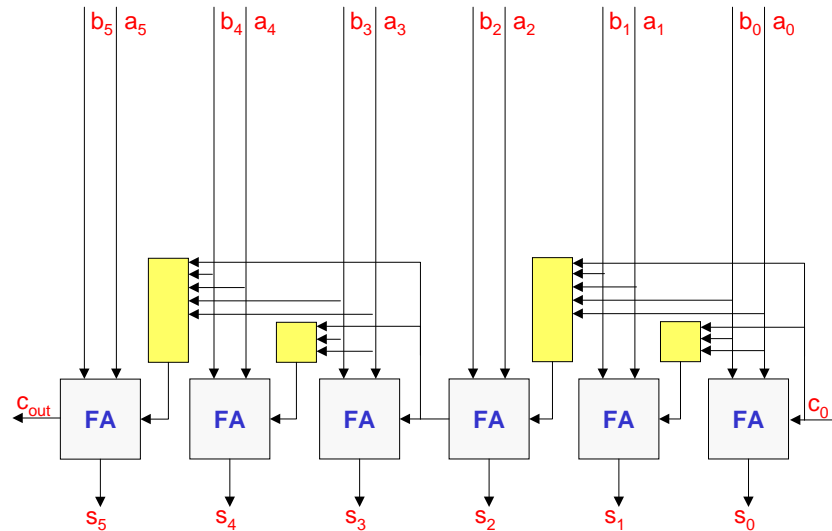
$$c_{i+1} = a_i b_i + a_i \oplus b_i (a_{i-1} b_{i-1} + a_{i-1} \oplus b_{i-1} c_{i-1})$$

- e così via fino ad esprimere il riporto  $c_{i+1}$  in funzione dei bit da  $i$  a  $0$  dei due addendi

## Sommatori *Carry-Lookahead*



## Sommatori *Carry-Lookahead*



Maurizio Palesi

19

## Prodotto

### ■ Due fasi

- Calcolo dei prodotti parziali
  - ✓ Uguale per tutti i moltiplicatori
- Somma dei prodotti parziali
  - ✓ Somma per righe
  - ✓ Somma per diagonali
  - ✓ Somma per colonne
    - Moltiplicatore di Wallace
    - Moltiplicatore di Dadda

Maurizio Palesi

20

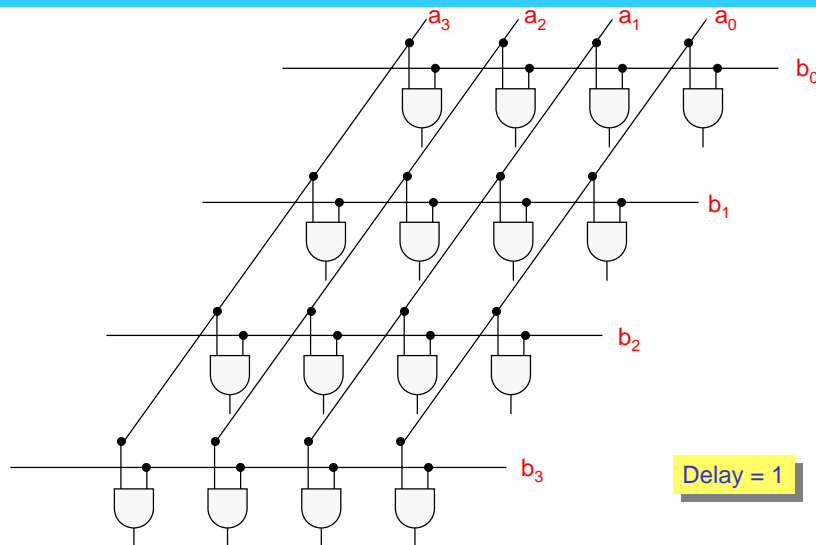
## Matrice dei Prodotti Parziali

	$a_3$	$a_2$	$a_1$	$a_0$
	$b_3$	$b_2$	$b_1$	$b_0$
	$a_3b_0$	$a_2b_0$	$a_1b_0$	$a_0b_0$
	$a_3b_1$	$a_2b_1$	$a_1b_1$	$a_0b_1$
	$a_3b_2$	$a_2b_2$	$a_1b_2$	$a_0b_2$
	$a_3b_3$	$a_2b_3$	$a_1b_3$	$a_0b_3$

Maurizio Palesi

21

## Calcolo dei Prodotti Parziali



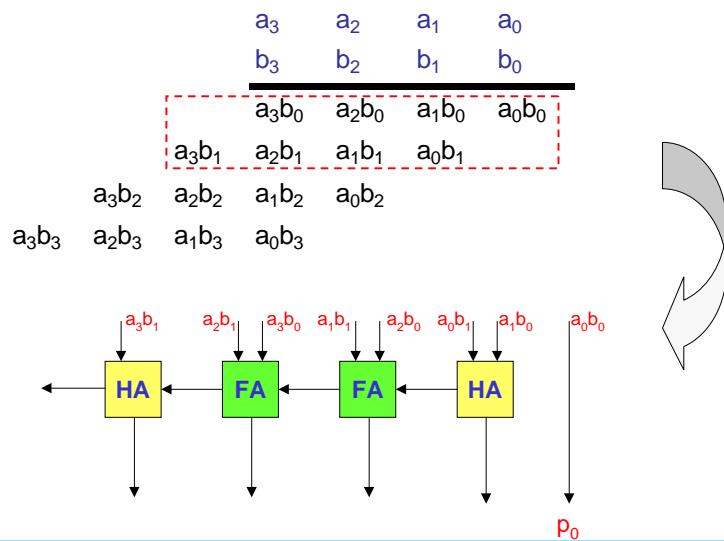
Maurizio Palesi

22

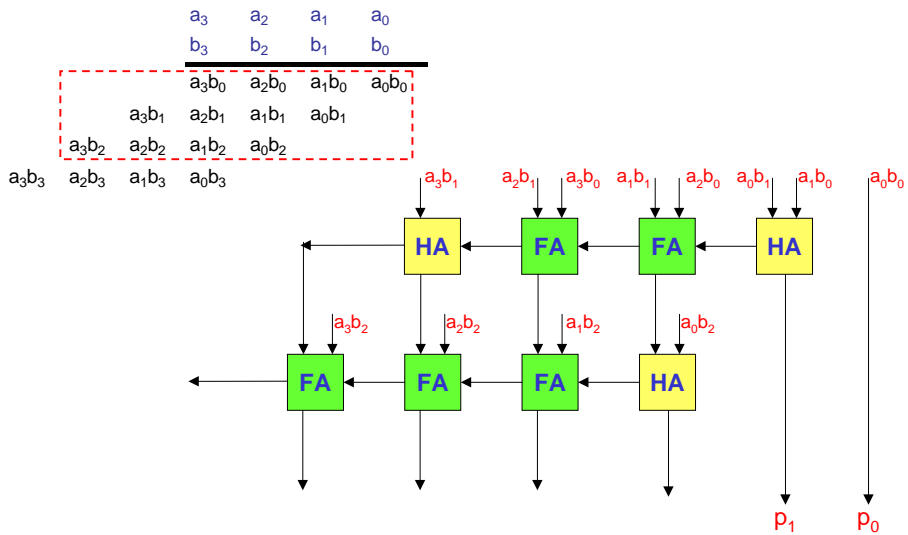
## Somma per Righe

		$a_3$	$a_2$	$a_1$	$a_0$
		$b_3$	$b_2$	$b_1$	$b_0$
		$a_3b_0$	$a_2b_0$	$a_1b_0$	$a_0b_0 +$
		$a_3b_1$	$a_2b_1$	$a_1b_1$	$a_0b_1$
	$a_3b_2$	$a_2b_2$	$a_1b_2$	$a_0b_2$	
$a_3b_3$	$a_2b_3$	$a_1b_3$	$a_0b_3$		

## Somma per Righe



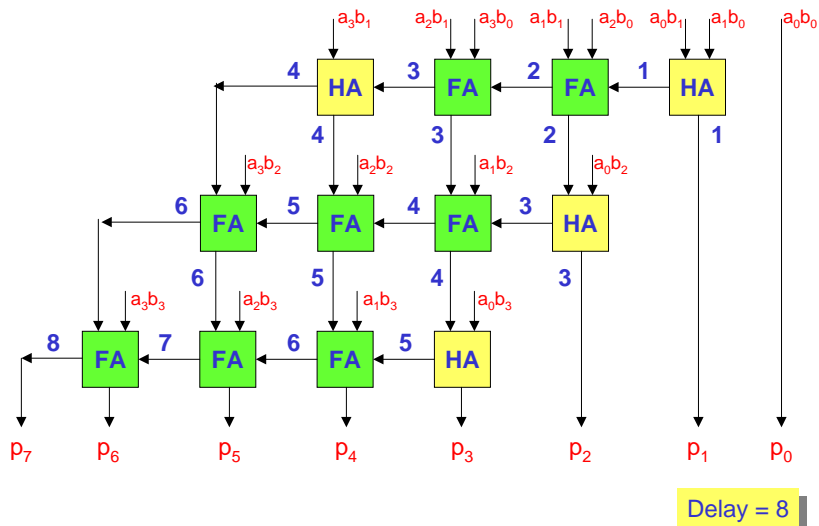
# Somma per Righe



Maurizio Palesi

25

# Somma per Righe



Delay = 8

Maurizio Palesi

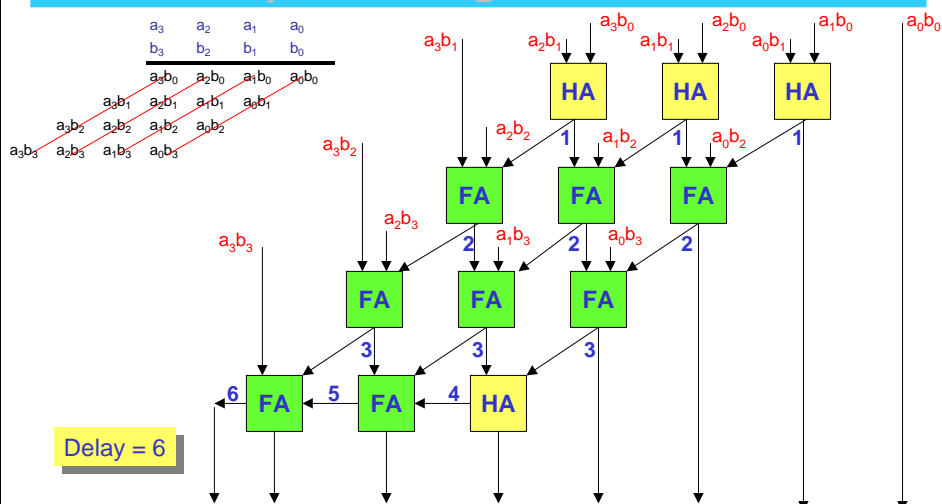
26

## Somma per Righe

- I tempi richiesti per la somma dei prodotti parziali dipendono *linearmente* dal numero dei bit dei fattori

$$\text{Delay}_{\text{SpR}} = 3n - 4$$

## Somma per Diagonali

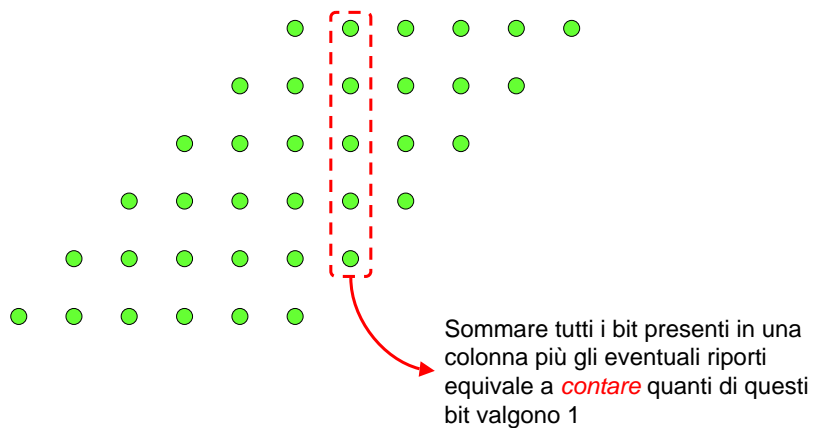


## Somma per Diagonali

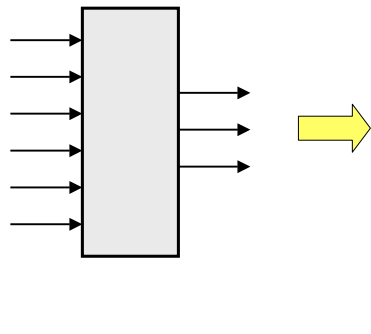
- I tempi richiesti per la somma dei prodotti parziali dipendono *linearmente* dal numero dei bit dei fattori

$$\text{Delay}_{\text{SpD}} = 2n - 2$$

## Somma per Colonne



# Contatore Parallelo



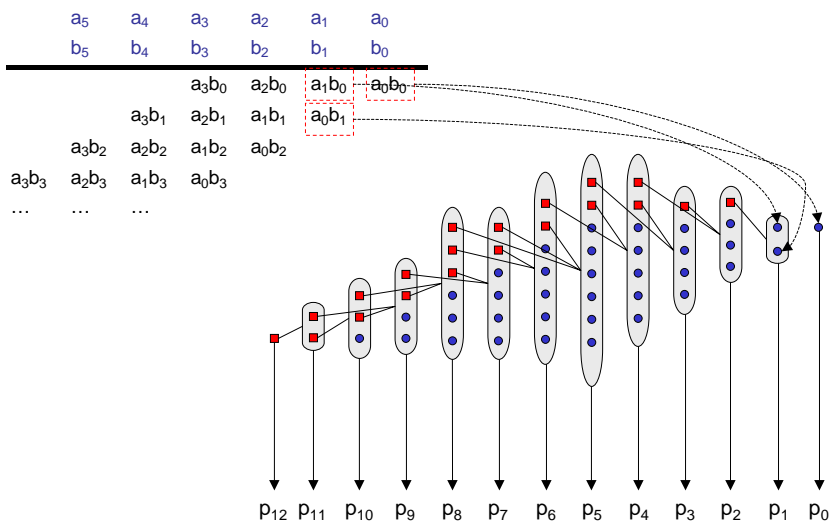
Se le sei entrate appartengono alla colonna  $i$ -esima (peso  $2^i$ ), dei tre bit in uscita

- Il **LSB** avrà lo stesso peso  $2^i$
- Quello **centrale** avrà peso  $2^{i+1}$
- Il **MSB** avrà peso  $2^{i+2}$

**Note**

Un contatore a 2 bit è un HA  
 Un contatore a 3 bit è un FA

# Somma per Colonne

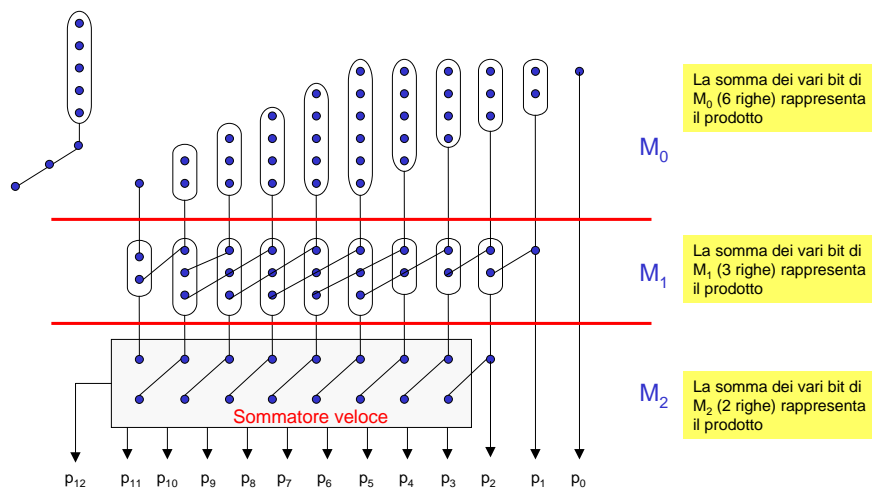


## Somma per Colonne

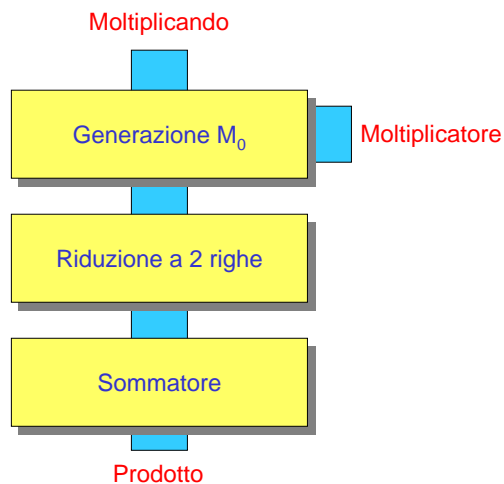
- La struttura ha 13 uscite
  - Un moltiplicatore a 6 bit ha 12 uscite
  - La 13esima uscita è sempre nulla
- Perdita di regolarità dei circuiti e delle connessioni
- Velocità
  - Se  $\text{Delay}_{HA} = \text{Delay}_{FA} = \text{Delay}_{\text{Counter}} = 1$

$$\text{Delay}_{\text{SpC}} = 2n - 1$$

## Somma per Colonne (schema alternativo)



## Somma per Colonne (schema alternativo)



## Somma per Colonne (schema alternativo)

### ■ Osservazioni

- Il sommatore finale somma due righe
- I contatori del livello precedente hanno *2 uscite* al massimo → *3 ingressi* al massimo → *3 righe*
- I contatori del livello precedente hanno *3 uscite* al massimo → *7 ingressi* al massimo → *7 righe*
- I contatori del livello precedente hanno *7 uscite* al massimo → *127 ingressi* al massimo → *127 righe*
- Moltiplicatori con fattori fino a *127 bit riducono* il numero di righe della matrice iniziale a una matrice con due righe *in non più di tre passi!*

## Somma per Colonne (schema alternativo)

### ■ Vantaggi

- Velocità (ritardo dominato dal sommatore finale)
- Un solo addizionatore veloce (e costoso)
  - ✓ La struttura con somma per righe ne richiederebbe  $n-1$

### ■ Svantaggi

- Non regolarità
  - ✓ Potrebbe richiedere contatori con numeri diversi (e potenzialmente alti) di ingressi
  - ✓ Soluzioni
    - Moltiplicatore di *Wallace*
    - Moltiplicatore di *Dadda*

## Moltiplicatore di Wallace

### ■ Usa soltanto contatori paralleli

- A 3 ingressi e 2 uscite (Full Adder)
- A 2 ingressi e 2 uscite (Half Adder)

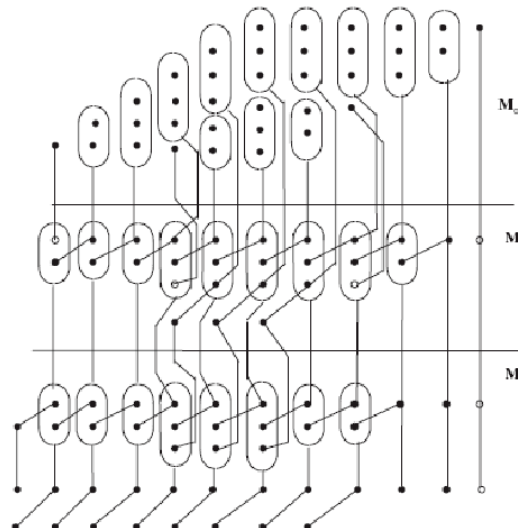
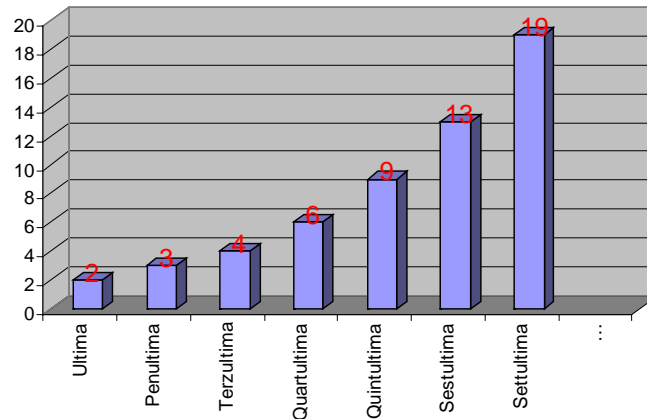


Fig. 7.14, Progettazione Digitale/II edizione, Fummi, Sami, Silvano, Mc Graw Hill

## Moltiplicatore di Wallace

Massimo numero di righe per matrice



## Moltiplicatore di Wallace

- Maggiore lentazza nella riduzione delle righe
- Maggior numero di stadi in cascata
- Contatori semplici (2 e 3 ingressi) → Veloci
  - L'ipotesi che il ritardo di propagazione di un contatore parallelo sia indipendente dal numero degli ingressi non è realistica

# Moltiplicatore di Dadda

- Wallace cerca di contare il *più possibile*
- Dadda cerca di contare il *meno possibile*
- Dadda e Wallace stessa velocità
- Dadda è più economico
  - Il risparmio aumenta all'aumentare della dimensione dei fattori

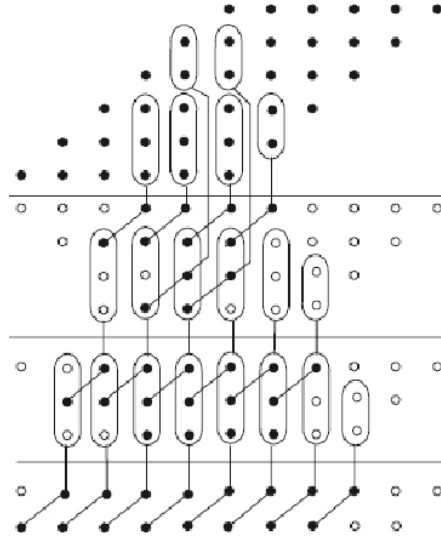


Fig. 7.15, Progettazione Digitale/II edizione, Fummi, Sami, Silvano, Mc Graw Hill