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

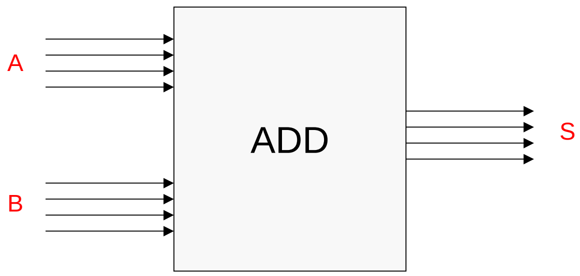
- Due diversi aprocci
  - ➔Approccio Hardware
  - ➔Approccio firmware

Presentazione dei Dati

- Parallelo
- Seriale pura
- Seriale a byte
- Mista

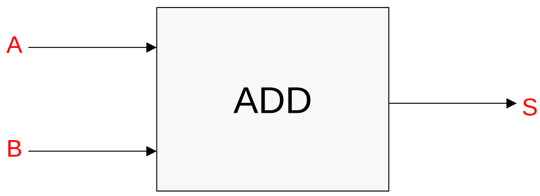
Presentazione dei Dati

- 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 sigificativo)
      - Little endian
      - Sommatori, 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

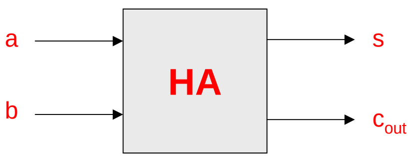
Presentazione dei Dati

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

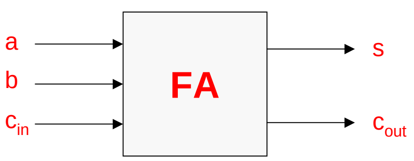
Presentazione dei Dati

- 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

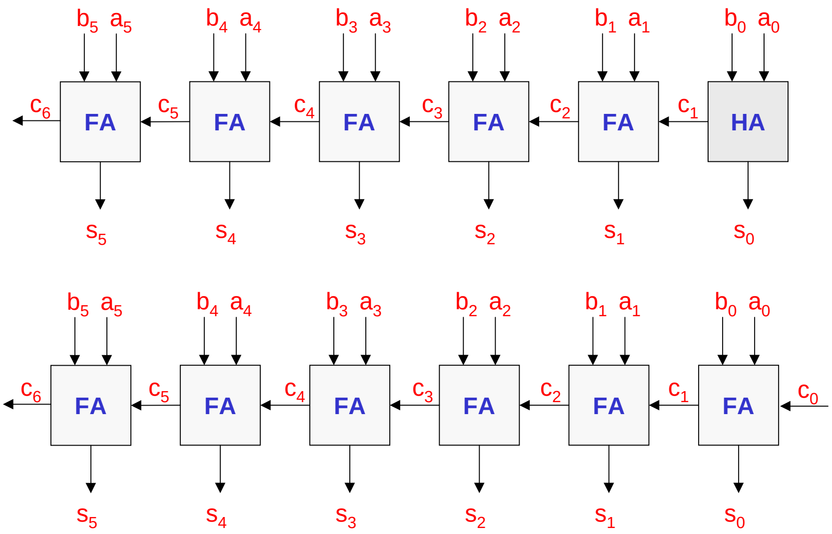


a	b	s	cout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



a	b	cin	s	cout
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

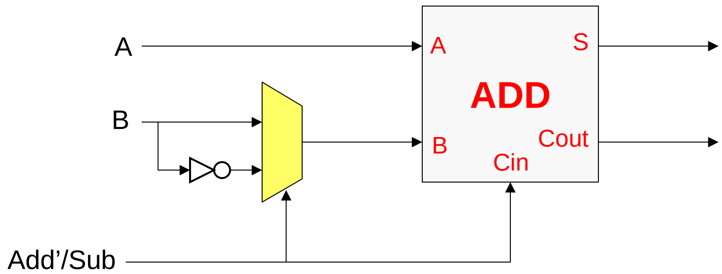


Overflow

- 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

Differenza

- A - B = A + (-B) = A + C2(B)
  - ➔ C2(B) = NOT(B) + 1

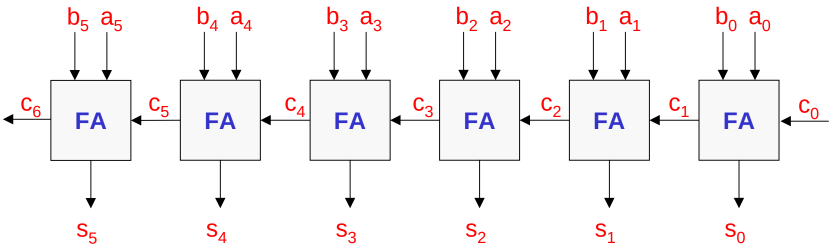


Overflow nelle Somme Algebriche

- 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

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

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 τ 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

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

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

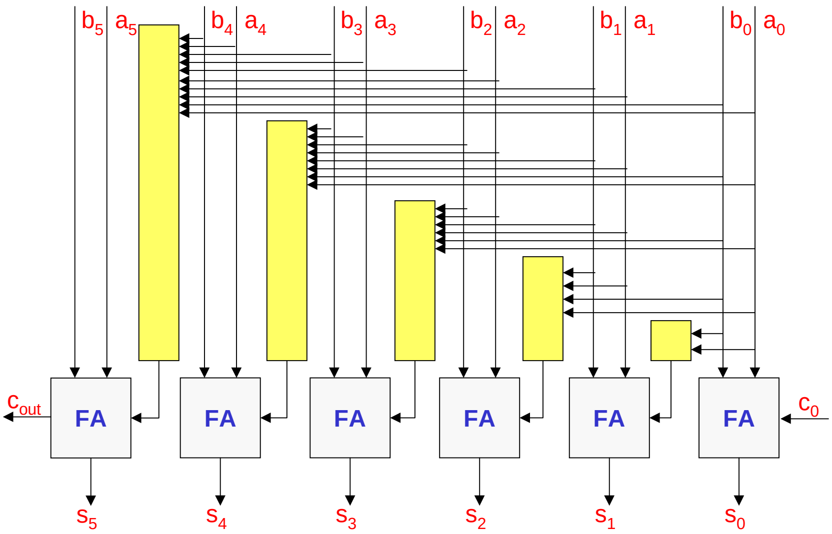
$$c_i = G_{i-1} + P_{i-1} c_{i-1} = a_{i-1} b_{i-1} + a_{i-1} \oplus b_{i-1} c_{i-1}$$

- 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*