Sommatori e Moltiplicatori

Prof. Alberto Borghese
Dipartimento di Scienze dell'Informazione
borghese@di.unimi.it

Università degli Studi di Milano

Riferimenti: Appendice C5 prima parte. Per approfondimenti e HW della moltiplicazione consultare il Fummi.

Sommario

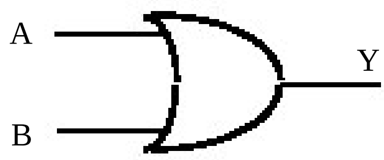
Sommatori

Moltiplicatori

Somma e prodotto logico

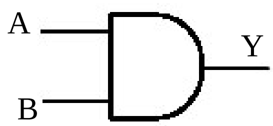
Somma => OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



Moltiplicazione => AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



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

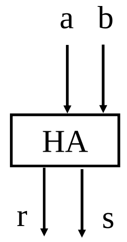
=

1	1	0	1

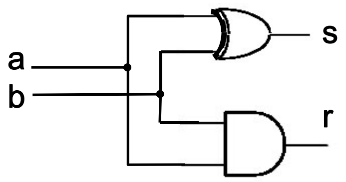
(Half) Adder ad 1 bit

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:

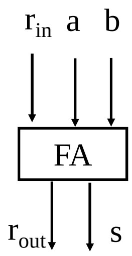
a b	r_{in}	somma	riporto
0 0	0	0	0
0 1	0	1	0
1 0	0	1	0
1 1	0	0	1
0 0	1	1	0
0 1	1	0	1
1 0	1	0	1
1 1	1	1	1

$s = m_1 + m_2 + m_4 + m_7$
$r = m_3 + m_5 + m_6 + m_7$

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

$$= (a \oplus b)\, \bar{r}_{in} + (\bar{a}\bar{b} + ab)\, r_{in} =$$

$$= (a \oplus b)\, \bar{r}_{in} + \overline{(a \oplus b)}\, r_{in}$$



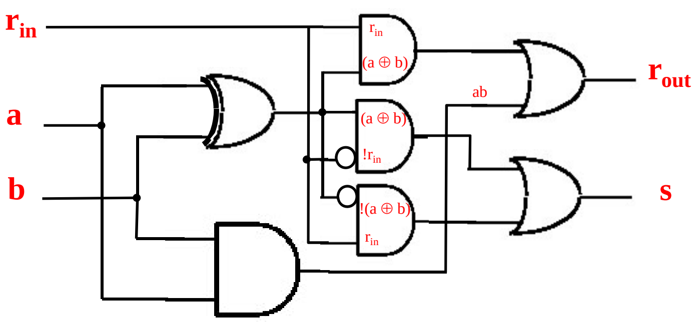
$r_{out} = a\, b\, \bar{r}_{in} + \bar{a}\, b\, r_{in} + a\, \bar{b}\, r_{in} + a\, b\, r_{in} = ab + (a \oplus b)\, r_{in}$
$r_{out} = a\, r_{in} + (a \oplus r_{in})\, b$

Quale è meglio?

Implementazione circuitale

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

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



7 porte logiche.

Cammini critici: s -> 3; r_{out} -> 3

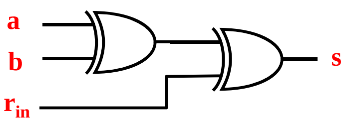
Semplificazione circuitale

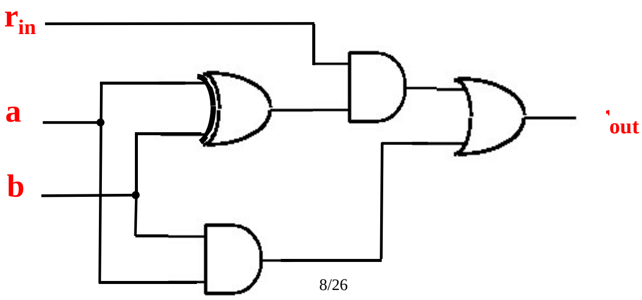
$$s = (a \oplus b)\overline{r_{in}} + \overline{(a \oplus b)}r_{in} = (a \oplus b) \oplus r_{in}$$

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

5 porte logiche.

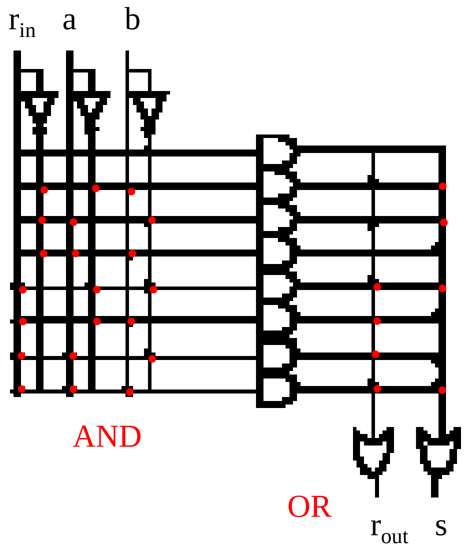
Cammini critici: s –> 2; r_{out} -> 3





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.

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.

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

Moltiplicatori

Moltiplicazione binaria

Moltiplicando → 1 1 0 1 1 x
Moltiplicatore → 1 1 1 =

```
     1 1 0 1 1 x    27₁₀
         1 1 1 =     7₁₀
   ------------------------
1 1 1 1 1 1
     1 1 0 1 1 +
   1 1 0 1 1 –
 1 1 0 1 1 - -
-------------------------
1 0 1 1 1 1 0 1    189₁₀
```

```
 -----------------------
    1 1 1 1 1
         1 1 0 1 1 +
      1 1 0 1 1 -
 -----------------------
   1
    1 0 1 0 0 0 1 +
    1 1 0 1 1 - -
 -----------------------
   1 0  1 1 1 1 0 1
```

prodotti parziali

prodotto → 1 0 1 1 1 1 0 1

Moltiplicazione mediante shift

Lo shift di un numero a dx, di k cifre, corrisponde ad una divisione per la base elevata alla k-esima potenza.
Lo shift di un numero a sx, di k cifre, corrisponde ad una moltiplicazione per la base elevata alla k-esima potenza.

Esempio:
$213_{10} / 10 = 21.3_{10}$
$213_{10} = (2 \times 10^2 + 1 \times 10^1 + 3 \times 10^0) / 10^1 =$
$(2 \times 10^2 + 1 \times 10^1 + 3 \times 10^0) \times 10^{-1} =$
$(2 \times 10^2 \times 10^{-1} + 1 \times 10^1 \times 10^{-1} + 3 \times 10^0 \times 10^{-1}) =$
$(2 \times 10^1 + 1 \times 10^0 + 3 \times 10^{-1}) = 21.3$ cvd.

Esempio:
23 / 4 = 5, 75 => 10111 / 100 =
$(1x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 1x2^0) \times 2^{-2} =$
$(1x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2}) = 5{,}75$ cvd.

Moltiplicazione binaria

Moltiplicando → 1 1 0 1 1 x
Moltiplicatore → 1 1 1 =

```
 1 1 0 1 1 x   27_10
     1 1 1 =    7_10
-------------------------
1 1 1 1 1 1
     1 1 0 1 1 +
   1 1 0 1 1 –
 1 1 0 1 1 - -
-------------------------
1 0 1 1 1 1 0 1   189_10
```

```
----------------------
1 1 1 1 1
    1 1 0 1 1 +
  1 1 0 1 1 -
----------------------
1
  1 0 1 0 0 0 1 +
  1 1 0 1 1 - -
----------------------
```

Prodotto → 1 0 1 1 1 1 0 1

Moltiplicazione binaria

Moltiplicando → 1 1 0 1 1 x
Moltiplicatore → 1 0 1 1 =

Prodotti parziali

Riporto

Somma parziale

Prodotto

```
        1 1 0 1 1 x     27 x
          1 0 1 1 =     11 =
----------------------  -----
      1 1 1 1 1
          1 1 0 1 1 +    27 +
        1 1 0 1 1 -     27- =
----------------------  ------
    0 0 0 0 0           297
      1 0 1 0 0 0 1 +
      0 0 0 0 0 - -
----------------------
1 1 0 1 0
    1 0 1 0 0 0 1 +
  1 1 0 1 1 - - - =
----------------------
1 0 0  1 0 1 0 0 1    -> 297|_10
```

Moltiplicazione binaria (su 4 bit)

Moltiplicando

Moltiplicatore

Prodotti parziali (AND)

Somma parziale (Sommatori)

Prodotto

```
         1 0 1 1 x     11_10  x
           1 0 1 =      5_10  =
------------------------
       0 0 0 0
         1 0 1 1 +   1011*1 * 2^0 +
       0 0 0 0 –     1011* 0 *2^1 =
--------------------------
         1 0 1 1                   +
     1 0 1 1 - -     1011 *1*2^2 =
--------------------------
     1 1 0 1 1 1     55_10
```

Il prodotto parziale è = { Moltiplicando incolonnato opportunamente / 0

La moltiplicazione binaria

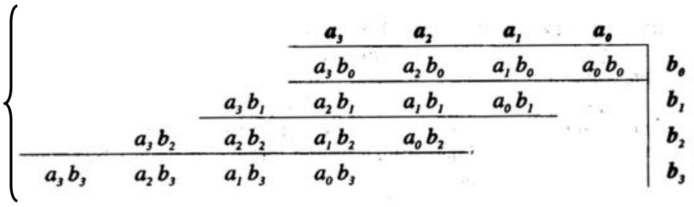
Possiamo vederla come:

Un primo stadio in cui si mette in AND ciascun bit del moltiplicatore con il moltiplicando.

Un secondo stadio in cui si effettuano le somme (full adder) dei bit sulle righe contenenti i prodotti parziali.

La matrice dei prodotti parziali

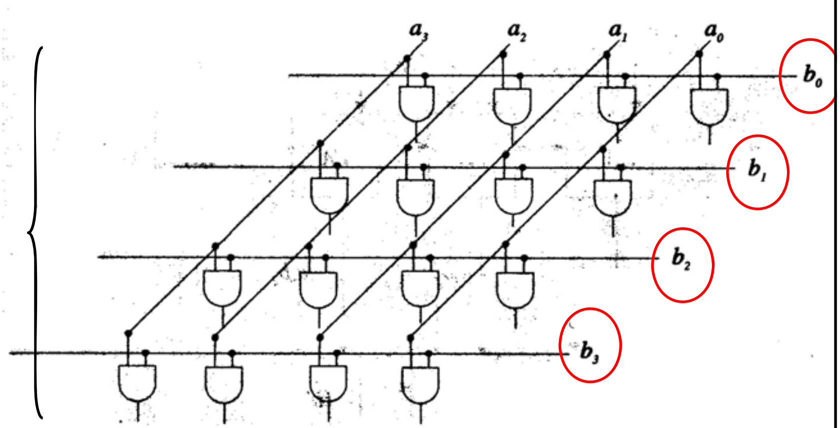
Prodotti parziali



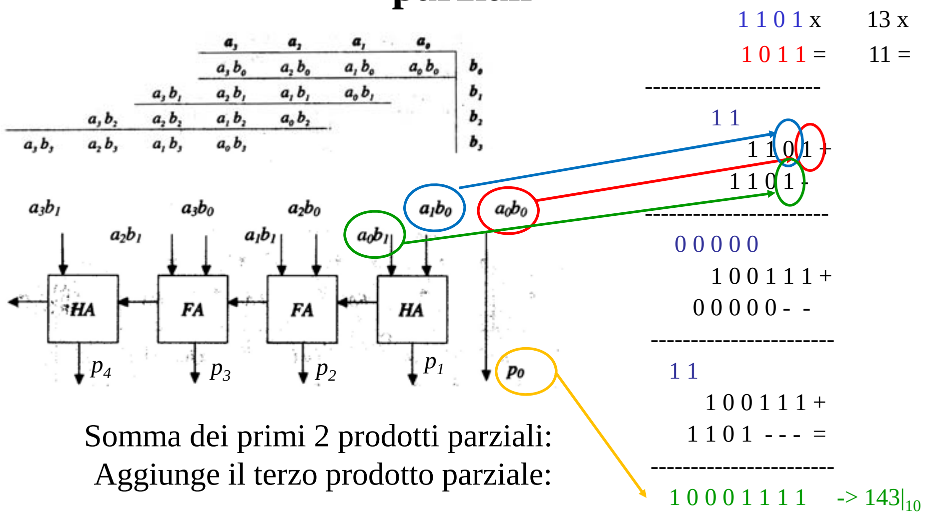
In binario i prodotti parziali sono degli AND.

Il circuito che effettua i prodotti

Prodotti parziali



Somma delle prime 2 righe dei prodotti parziali

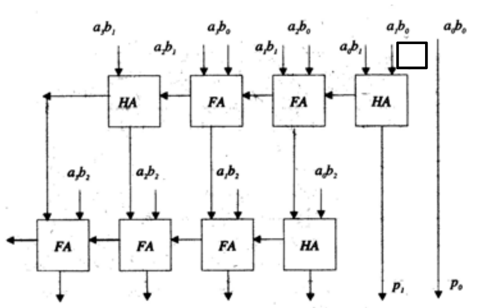


Somma dei primi 2 prodotti parziali:
Aggiunge il terzo prodotto parziale:

HA e FA non sono equivalenti
per i diversi cammini critici.

Somma della terza riga

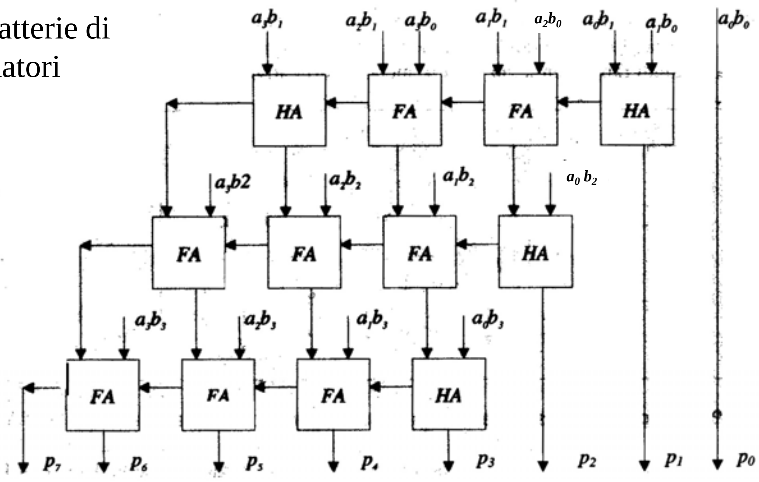
I primi due prodotti parziali sono sommati dalla prima batteria di sommatori.
Ogni altro prodotto parziale è sommato da un'ulteriore batteria di sommatori.



```
  1 1 0 1 x       13 x
  1 0 1 1 =       11 =
----------------------
 1 1
   1 1 0 1 +
  1 1 0 1 -
----------------------
 0 0 0 0 0
   1 0 0 1 1 1 +
  0 0 0 0 0 - -
----------------------
 1 1
   1 0 0 1 1 1 +
  1 1 0 1 - - - =
----------------------
1 0 0 0 1 1 1 1   -> 143|10
```

Circuito completo della somma dei prodotti parziali

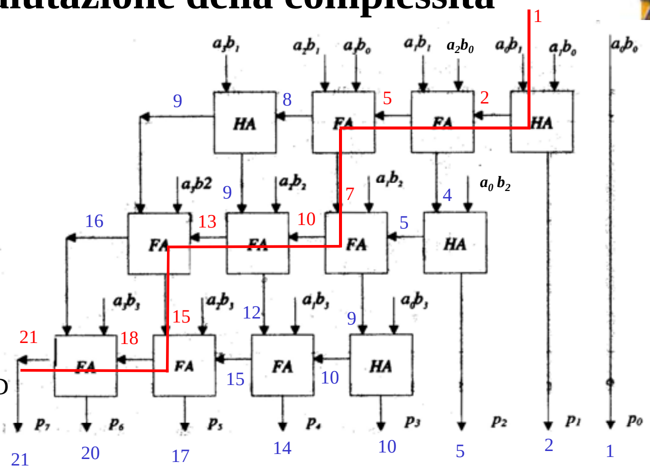
N-1 batterie di sommatori



Problema: overflow: A e B su 32 bit => P su 64 bit.

Valutazione della complessità

Complessità:
Half Adder: 2 porte
Full Adder: 5 porte



Stadio AND:

A su N bit
B su M bit

N * M porte AND

Stadio OR:

Numero porte se N = M = 4 -> 48

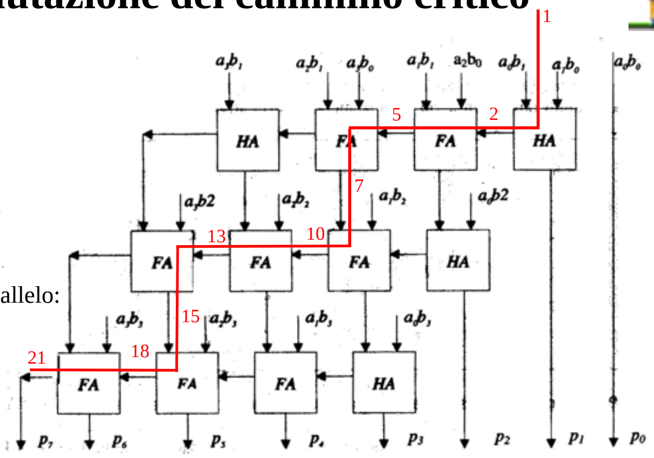
N sommatori per riga
M-1 righe

Numero porte = $\overbrace{(N-2) * 5 + 2*2}^{\text{1° riga}} + \overbrace{(M-2) *[(N-1) * 5 + 1]}^{\text{Altre righe (M-2)}}$

Valutazione del cammino critico

Cammini critici:
- Half Adder:
  - Somma – 1 porta
  - Riporto – 1 porta
- Full Adder:
  - Somma - 2 porte
  - Riporto - 3 porte

Gli AND operano in parallelo: ritardo 1.



21 20 17 14 10 5 2 1

Cammino critico: 21

Sommario

Addizionatori

Moltiplicatori