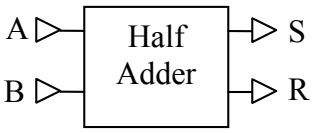
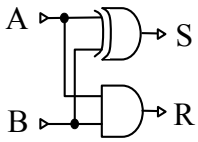
# Laboratorio del 11/11/2010 - I sommatori

1) **Addizionatore Half Adder (senza riporto in ingresso)**:

| A | B | S | R |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



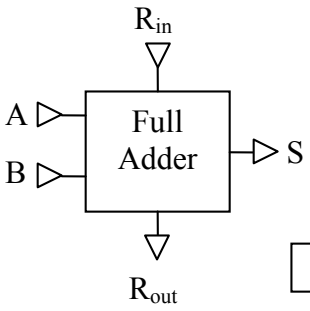



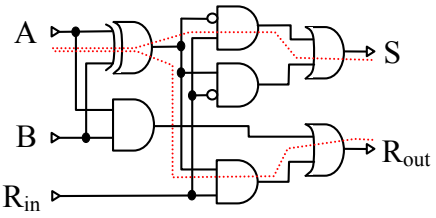

$S = A \oplus B \qquad R = A \cdot B$

N.Porte = 2 *Cammino Critico* S = 1 , R = 1

2) **Addizionatore Full Adder ( con riporto in ingresso )**:

| $R_{in}$ | A | B | S | $R_{out}$ |
|---|---|---|---|---|
| 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 |




N.Porte = 7 *Cammino Critico* S = 3 , R = 3

$S = \sim R_{in} \sim AB + \sim R_{in} A \sim B + R_{in} \sim A \sim B + R_{in} A\, B$
$= \sim R_{in}(\sim AB + A \sim B) + R_{in}(\sim A \sim B + A\, B)$
$= \sim R_{in}(A \oplus B) + R_{in} \sim (A \oplus B)$

$R_{out} = \sim R_{in} AB + R_{in} \sim AB + R_{in} A \sim B + R_{in} AB$
$= AB\,(\sim R_{in} + R_{in}) + R_{in}\,(\sim AB + A \sim B)$
$= AB + R_{in}(A \oplus B)$

Nota: **$\sim A \sim B + A\, B$** = $\sim A \sim B + A\, B = \sim(\sim(\sim A \sim B + A\, B))$
$= \sim(\sim(\sim A \sim B) \sim (A\, B)) = \sim((\sim\sim A + \sim\sim B)(\sim A + \sim B))$
$= \sim((A+B)(\sim A + \sim B)) = \sim(A(\sim A + \sim B) + B(\sim A + \sim B))$
$= \sim(A \sim A + A \sim B + B \sim A + B \sim B) = \sim(A \sim B + B \sim A) = \sim$ **$(A \oplus B)$**

**Semplificazione circuitale:**

$S = \sim R_{in}(A \oplus B) + R_{in} \sim (A \oplus B) = R_{in} \oplus (A \oplus B)$

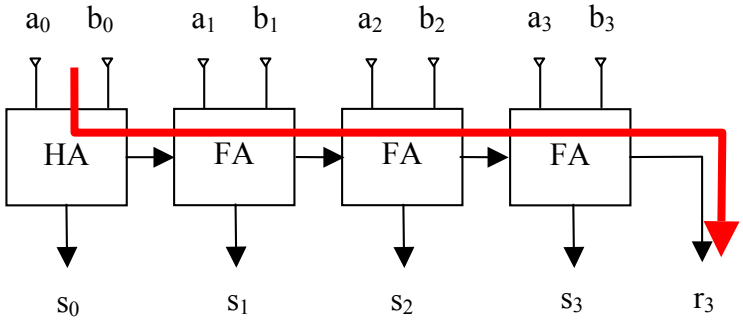

N.Porte = 5 *Cammino Critico* S = 2 , R = 3

***Nota**: le porte **AND e la OR possono essere pensate come "gate"** che lasciano "passare" il segnale presente su un terminale in funzione del segnale presente sull'altro. Diversamente **le porte XOR si comportano come "invertitori"**: il segnale presente su un terminale viene lasciato passare o invertito a seconda del segnale presente sull'altro.*

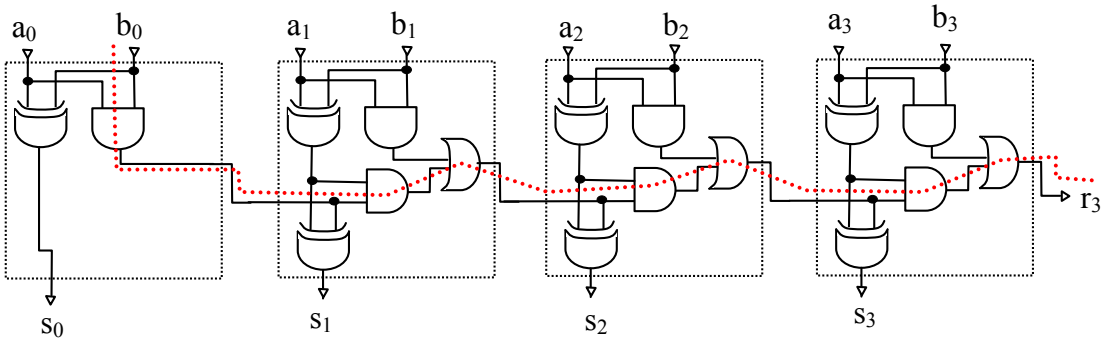
Massimo Marchi – marchi@dsi.unimi.it

### 3) Sommatore ad n bit (senza riporto in ingresso)

E' possibile realizzare un sommatore ad n bit usando un HA e n-1 FA collegati in cascata.




Il cammino critico è quello definito dalla propagazione dei riporti dal primo modulo all'ultimo. Se si esamina il circuito dei FA si nota che la porta XOR e le porta AND colegate direttamente agli ingressi $a_i$ $b_i$ si stabilizzano tutte nella prima unità di tempo. Ne segue che per propagare il riporto dall'ingresso $R_{in}$ all'ingresso $R_{out}$ di ogni sommatore occorrono due unità di tempo per ogni FA.



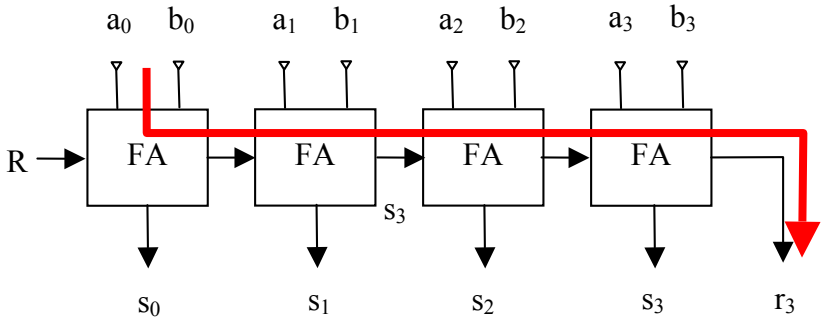

Il **cammino critico** quindi per questo sommatore è **1 + 2 * (n-1)** dove **n** è il numero di bit da sommare.

Nota: per i fini del corso potete considerare gli FA come un modulo monolitico in cui la propagazione del riporto avviene sempre 3 cicli dopo la stabilizzazione di tutti gli ingressi. In questo caso quindi il cammino critico del circuito risulta **1+3*(n-1) = 3*n -2.**

Massimo Marchi – marchi@dsi.unimi.it

### 4) Sommatore ad n bit (con riporto in ingresso)

Normalmente si preferisce adottare un FA anche per il primo modulo. Questo permette di mettere in cascata più sommatori e di realizzare semplicemente un sottrattore binario in complemento a due (vedi più avanti).

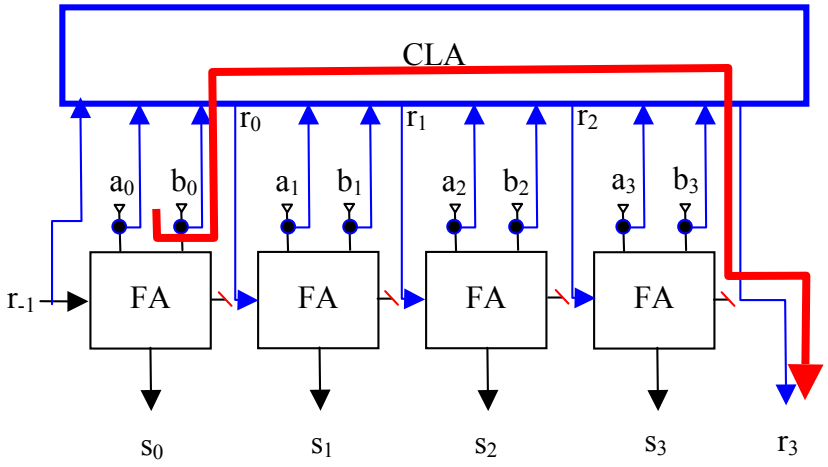



In questo caso quindi il **cammino critico** è **3 + 2 * (n-1) = 1+2*n** poiché al primo FA occorrono 3 unità di tempo per generare il primo riporto.

Nota: per i fini del corso potete considerare gli FA come un modulo monolitico in cui la propagazione del riporto avviene sempre 3 cicli dopo la stabilizzazione di tutti gli ingressi. In questo caso quindi il cammino critico del circuito risulta **3*n.**

### *5)* Sommatori veloci: unità di Carry Look-Ahead

Il tempo di commutazione dei circuito sommatore ad n bit dipende dal tempo di propagazione del riporto dalla coppia di bit meno significativa alla coppia di bit più significativa. Se riusciamo a *prevedere* il riporto $r_i$ che sarà sommato ad ogni coppia di bit prima che avvenga la propagazione dagli stadi precedenti possiamo accelerare il processo di somma. Si tratta di anteporre, o meglio affiancare al circuito sommatore un'unità di **Carry Look-Ahead** (**CLA**), di previsione del riporto.




Consideriamo il generico sommatore Full-Adder ad un bit in posizione i-esima. Il suo cammino critico è determinato dal calcolo del resto in uscita:

$$r_i = a_i b_j + r_{i-1}(a_i \oplus b_i) = g_i + r_{i-1}\, p_i$$

Massimo Marchi – marchi@dsi.unimi.it

Il termine $a_i b_j = g_i$ calcola se nello stadio i-esimo verrà generato un riporto mentre il termine $a_i \oplus b_i = p_i$ calcola se il riporto dello stadio precedente verrà propagato a quello successivo. Da notare che tutti questi termini possono essere calcolati in parallelo con cammino critico uguale a 1.
Per il circuito in esempio quindi vale:

$$\mathbf{r_0 = g_0 + r_{-1}\, p_0} \quad \text{e} \quad r_1 = g_1 + r_0\, p_1$$

Sviluppando:

$$\mathbf{r_1} = g_1 + r_0\, p_1 = g_1 + (g_0 + r_{-1}\, p_0)\, p_1 = \mathbf{g_1 + g_0\, p_1 + r_{-1}\, p_0\, p_1}$$

cioè si verifica un riporto $r_1$ nello stadio 1 se viene generato nello stadio stesso ( termine $g_1$) oppure se si verifica nel precedente e viene propagato dallo stadio attuale ( $g_0\, p_1$) oppure se c'era un riporto in ingresso e i due stadi iniziali lo hanno propagato ($r_{-1}\, p_0\, p_1$). Da notare che il riporto $r_1$ con questa forma algebrica può essere calcolato senza tenere conto del risultato degli stadi FA, al contrario di come accade con i sommatori a propagazione di riporto.

Sviluppiamo ora i riporti successivi:

$$\begin{aligned}\mathbf{r_2} \quad &= g_2 + r_1\, p_2 = g_2 + (g_1 + g_0\, p_1 + r_{-1}\, p_0\, p_1)\, p_2 \\ &= \mathbf{g_2 + g_1\, p_2 + g_0\, p_1\, p_2 + r_{-1}\, p_0\, p_1\, p_2}\end{aligned}$$

$$\begin{aligned}\mathbf{r_3} \quad &= g_3 + r_2\, p_3 = g_3 + (g_2 + g_1\, p_2 + g_0\, p_1\, p_2 + r_{-1}\, p_0\, p_1\, p_2)\, p_3 \\ &= \mathbf{g_3 + g_2\, p_3 + g_1\, p_2\, p_3 + g_0\, p_1\, p_2\, p_3 + r_{-1}\, p_0\, p_1\, p_2\, p_3}\end{aligned}$$

Riepilogando, un CLA a 4 bit deve realizzare le seguenti funzioni booleane:

$$\mathbf{r_0 = g_0 + r_{-1}\, p_0}$$

$$\mathbf{r_1 = g_1 + g_0\, p_1 + r_{-1}\, p_0\, p_1}$$

$$\mathbf{r_2 = g_2 + g_1\, p_2 + g_0\, p_1\, p_2 + r_{-1}\, p_0\, p_1\, p_2}$$

$$\mathbf{r_3 = g_3 + g_2\, p_3 + g_1\, p_2\, p_3 + g_0\, p_1\, p_2\, p_3 + r_{-1}\, p_0\, p_1\, p_2\, p_3}$$

Calcoliamo il cammino critico per queste funzioni: dividiamo gli operatori inserendo una coppia di parentesi in modo da avere solo operatori binari e valutiamo il numero massimo di parentesi aperte nel punto più profondo:

$\mathbf{r_0} = [g_0 + (r_{-1}\, p_0)]$ → **C.Critico = 2** ***(+1)***

$\mathbf{r_1} = \{[g_1 + (g_0\, p_1)] + [r_{-1}\, (p_0\, p_1)]\}$ → **C.Critico = 3** ***(+1)***

$\mathbf{r_2} = (\{[g_2 + (g_1\, p_2)] + [g_0\, (p_1\, p_2)]\} + [(r_{-1}\, p_0)(\, p_1\, p_2)])$ → **C.Critico = 4** ***(+1)***

$\mathbf{r_3} = [\{[g_3 + (g_2\, p_3)] + [g_1(\, p_2\, p_3)]\} + ([(g_0\, p_1)\, (p_2\, p_3)\,] + \{[r_{-1}\, (p_0\, p_1)](\, p_2\, p_3)\})]$ → **C.Critico = 5** ***(+1)***

Tempo necessario per calcolare i termini $p_i$ e $g_i$

Tutti questi riporti possono essere calcolati in parallelo quindi il cammino critico più lungo è quello corrispondente a $\mathbf{r_3}$ pari a **6** (il *+1* indica il tempo di calcolo dei termini $g_i$ ed $r_i$).
Calcoliamo il cammino critico per $\mathbf{s_i}$:

$$\mathbf{s_i} = (a_i \oplus b_i) \oplus r_{i-1} = [\, p_i \oplus (r_{i-1})] \rightarrow \mathbf{C.Critico = C.Critico(r_{i-1}) + 1}$$

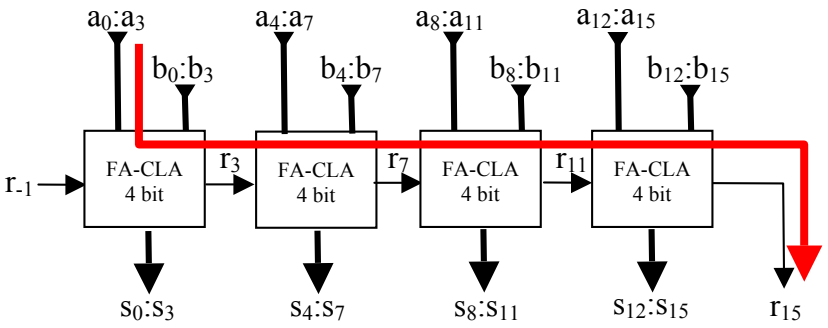
Massimo Marchi – marchi@dsi.unimi.it

Ne segue che il cammino critico più lungo per il calcolo del risultato corrisponde al calcolo di $\mathbf{s_3}$ ed è pari a **6**, come per il calcolo di $\mathbf{r_3}$.

Da quanto visto, ne segue che il cammino critico complessivo per un sommatore a 4 bit con CLA è pari al cammino critico per $\mathbf{r_3}$ (o anche a quello di $\mathbf{s_3}$) che è **6** . Un corrispondente sommatore classico senza anticipazione di riporto ha cammino critico **1+2*4 = 9** (da cui la dicitura "sommatore veloce"!!).

La complessità di circuito di un'unità CLA aumenta all'aumentare del numero di bit da trattare. Questo rende economico sviluppare sommatori veloci solo per basse cardinalità di bit e se necessario utilizzare questi sommatori come blocchi per sommatori modulari di più grandi dimensioni:

Ex: *un sommatore a 16 bit può essere realizzato con 4 moduli CLA a 4 bit:*




*Il cammino critico è 4 * 6 = 24. Il cammino critico del sommatore classico è 1+16 * 2 = 33.*

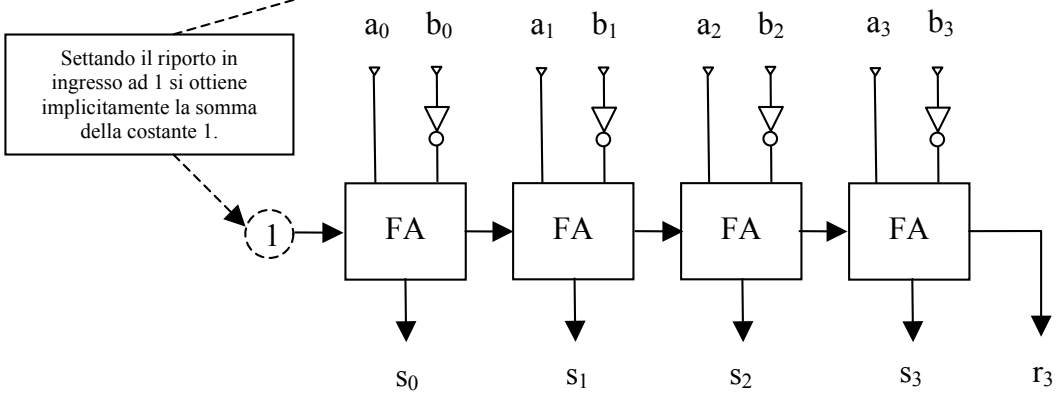
### 6) Sottrattori ad n bit

In binario la sottrazione ad n bit (con segno) può essere realizzata con una somma secondo la regola:

$$\mathbf{A - B = A + \sim B + 1}$$

dove **~B** è l'inversione bit a bit del secondo termine.

Adottando un sommatore con riporto in ingresso è possibile realizzare un sottrattore binario a n bit in questo modo:




Massimo Marchi – marchi@dsi.unimi.it

7) **Problema dell'overflow**

Nel caso che il risultato di un addizione con segno ecceda il limite di rappresentazione della dimensione della parola si verifica un errore di segno (Overflow).

*Esempio:*

*$19_{10} + 15_{10} = 34_{10}$ se eseguita in aritmetica binaria con segno a 6 bit dà come risultato:*

| *1* | *1* | *1* | *1* | *1* | | *R* |
|---|---|---|---|---|---|---|
| **0** | **1** | **0** | **0** | **1** | **1** | + |
| **0** | **0** | **1** | **1** | **1** | **1** | = |
| **1** | $^{1}$**0** | $^{1}$**0** | $^{1}$**0** | $^{1}$**1** | $^{1}$**0** | |

*$0\ 10011_2 + 0\ 01111_2 = 1\ 00010_2$ che in rappresentazione in complemento a due è un numero negativo , precisamente $-11110_2 = -30_{10}$ .*

*Analogamente si verifica lo stesso effetto di riporto errato sul bit di segno quando si sommano due numeri negativi in valore assoluto troppo grandi:*

Ex: $-17_{10} + -19_{10} = -35_{10}$ , *se eseguita in aritmetica binaria con segno a 6 bit dà come risultato:*

*$-17_{10} = -\ 10001_2$ (M&S) = $101111_2$ (CA2)*

*$-19_{10} = -\ 10011_2$ (M&S) = $101101_2$ (CA2)*

| *1* | *1* | *1* | *1* | *1* | *1* | | *R* |
|---|---|---|---|---|---|---|---|
| | **1** | **0** | **1** | **1** | **1** | **1** | + |
| | **1** | **0** | **1** | **1** | **0** | **1** | = |
| ~~**1**~~ | $^{1}$**0** | **1** | $^{1}$**1** | $^{1}$**1** | $^{1}$**0** | $^{1}$**0** | |

*$1\ 01111_2 + 1\ 01101_2 = 0\ 11100_2$ che è un numero positivo , precisamente $+28_{10}$ .*

*Nel caso di somme miste, negativo con positivo e viceversa, questo problema non si pone poiché il risultato in valore assoluto sarà sempre al più grande come il più grande in valore assoluto dei due termini sommati.*

**Circuito per rilevare l'overflow:**

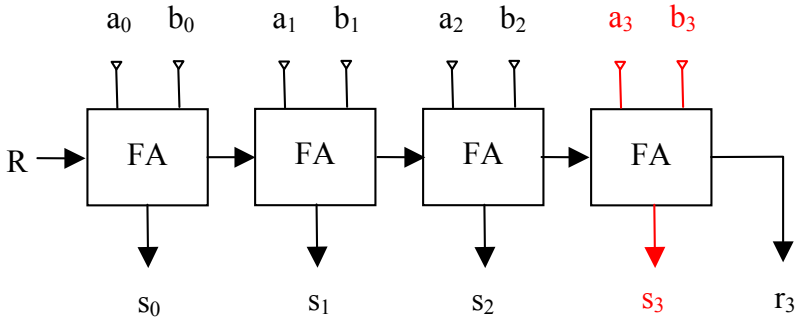
Da quanto visto sopra ne segue che l'overflow si verifica solo nei seguenti due casi:

*Segno di A = 0 e Segno di B = 0 e Segno del risultato = 1*

*oppure*

*Segno di A = 1 e Segno di B = 1 e Segno del risultato = 0*

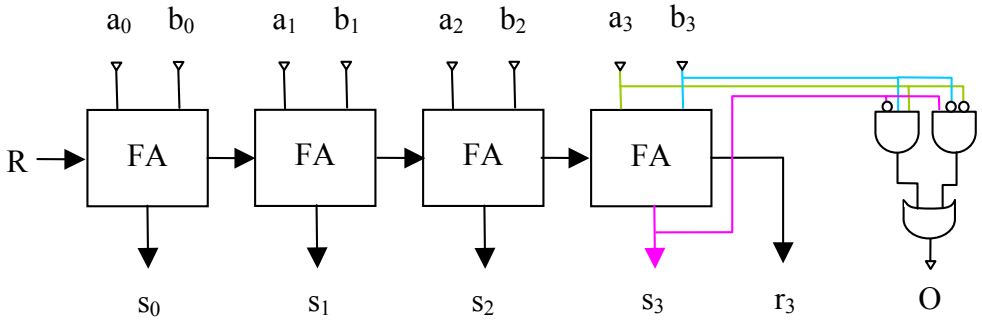
Tenendo presente il circuito Full-Adder a n bit,

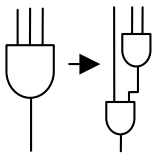



il circuito che realizza questo controllo deve sintetizzare la seguente forma tabellare:

| $s_{n-1}$ | $a_{n-1}$ | $b_{n-1}$ | Overflow |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

In forma SOP si può scrivere: $O = \sim s_{n-1}\, a_{n-1}\, b_{n-1} + s_{n-1} \sim a_{n-1} \sim b_{n-1}$



Per realizzare le porte AND a tre ingressi occorrono due porte in cascata. Ne segue che *il cammino critico di questo circuito per l'overflow* ha cammino critico 2 (le due porte AND a tre ingressi eseguite in parallelo) più la porta OR per un totale di 3.

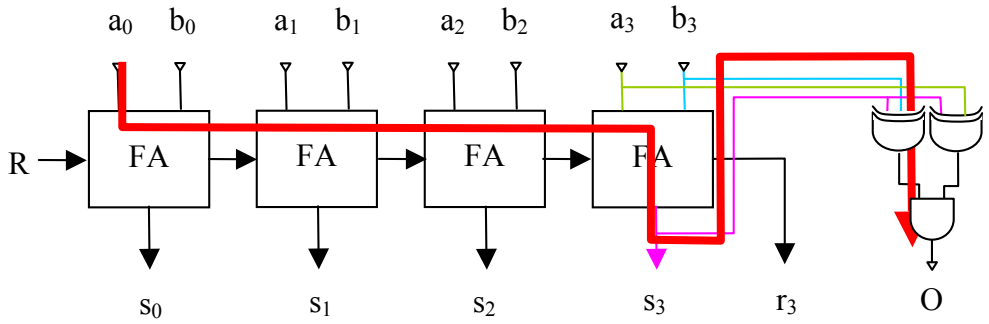
Massimo Marchi – marchi@dsi.unimi.it

Si può fare di meglio (assegnamo per comodità $S=S_{n-1}$ $A=a_{n-1}$ $B=b_{n-1}$):

| | |
|---|---|
| O = ~SAB + S~A~B | |
| = ~SA~SB + S~AS~B | $x = xx$ |
| = ~SA~SB + 0 + 0 + S~AS~B | $x=x+0$ |
| = ~SA~SB + ~SAS~B + S~A~SB + S~AS~B | $0 = x\ {\sim}x\ ,\ x=xy$ |
| = (~SB + S~B)~SA + S~A(~SB + S~B) | $xy+xz=x(y+z)$ |
| = (S⊕B) ~SA + S~A(S⊕B) | $\sim xy + x\sim y = x \oplus y$ |
| = (S⊕B) (~SA + S~A) | $xy+xz=x(y+z)$ |
| = (S⊕B) (S⊕A) | $\sim xy + x\sim y = x \oplus y$ |

che ha cammino critico pari a **2**.

Ne segue che il cammino critico di un circuito Full-Adder con rilevamento dell'overflow definito dal tempo necessario per propagare i segnali in ingresso al circuito di rilevamento dell'overflow più il cammino critico del circuito stesso:




**Cammino critico:** primi n-1 stadi + ultimo stadio + Overflow = 1+2(n-1) + 2 + 2 = **2(n+1) + 1**

*Nota: Le porte XOR in questo caso si comportano come invertitori a comando: quando S è a 0 permettono ad A e B di giungere inalterati alla porta AND e di realizzare la parte alta della forma tabellare mentre quando S è uguale ad 1 invertono A e B realizzando la parte bassa, cioè ~A~B.*

*Nota: Il cammino critico del FA non passa più per l'ultimo resto poiché, dati i termini precedenti, il cammino critico per avere il bit di somma più il tempo del circuito di overflow, in totale 4, è maggiore del cammino critico per calcolare quest'ultimo resto, come visto sopra pari a 3.*

Massimo Marchi – marchi@dsi.unimi.it