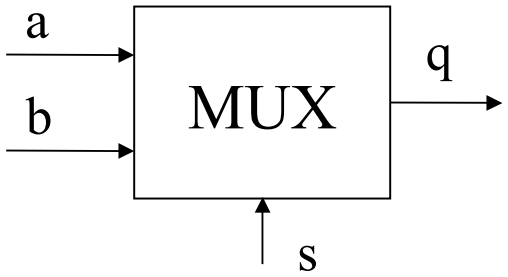
Componenti di un sistema digitale

Il Multiplexer 2x1

Dispositivo che permette di selezionare uno degli *n* ingressi
e
presentarlo in uscita

- Con n linee di ingresso un multiplexer richiede un numero di linee di comando pari a $\lceil \lg_2(n) \rceil$



Descrizione:

```
if ( s==0)
   q = a;
else  q = b;
```

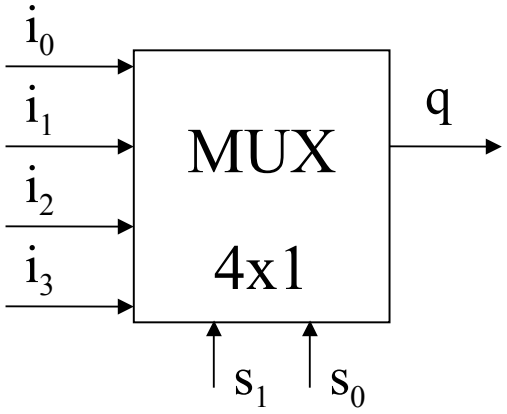
Tabella della verità

a	b	s	q
0	-	0	0
1	-	0	1
-	0	1	0
-	1	1	1

f(a,b,s)= a s'+b s

Il Multiplexer 4x1

Con 4 linee di ingresso sono richieste due linee di comando]



Descrizione:

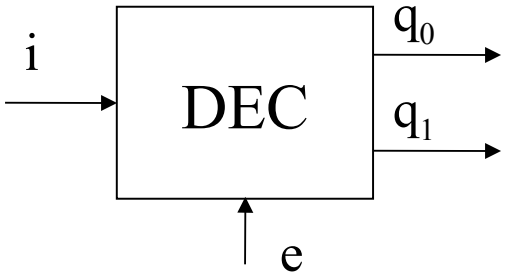
```
if ( s1==0 && s0==0)
   q = i0;
else  if ( s1==0 && s0==1)
   q = i1;
else  if ( s1==1 && s0==0)
   q = i2;
else  q = i3;
```

$f(i_3,i_2,i_1,i_0,s_1,s_0)= i_3s_1s_0+i_2s_1s_0'+i_1s_1's_0+i_0s_1's_0'$

Il decodificatore (in logica diretta)

Un decodificatore (1 su *m*) accetta in ingresso un codice di *n* bit e presenta in uscita *m*=2*n* linee, sulle quali asserisce solo quella che corrisponde alla codifica in ingresso

- Numerando le linee di uscita da 0 a 2*n*-1, viene asserita quella che corrisponde al numero presente in ingresso



$q_0 = ei'$

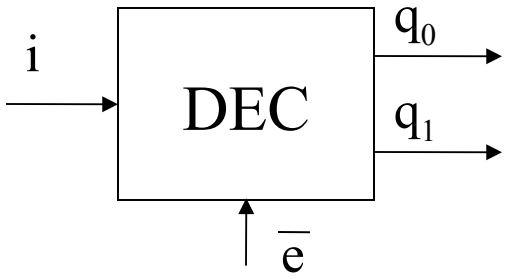
$q_1 = ei$

Descrizione:

```
if ( e==1 && i==0)
    q0= 1
else   q0= 0

if ( e==1 && i==1)
    q1= 1
else   q1= 0
```

Il decodificatore (in logica negata)



Descrizione:

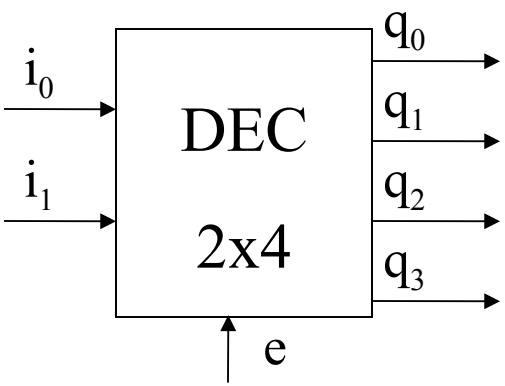
```
if ( e==0 && i==0)
    q0= 0
else  q0= 1

if ( e==0 && i==1)
    q1= 0
else  q1= 1
```

q_0'= e'i' q_0 = e+i

q_1'= e'i q_1 = e+i'

Il decodificatore 2x4 (in logica diretta)



Descrizione:

```
if ( e==1 && i1==0&& i0==0) q0= 1
else   q0= 0
if ( e==1 && i1==0&& i0==1) q1= 1
else   q1= 0
if ( e==1 && i1==1&& i0==0) q2= 1
else   q2= 0
if ( e==1 && i1==1&& i0==1) q3= 1
else   q3= 0
```

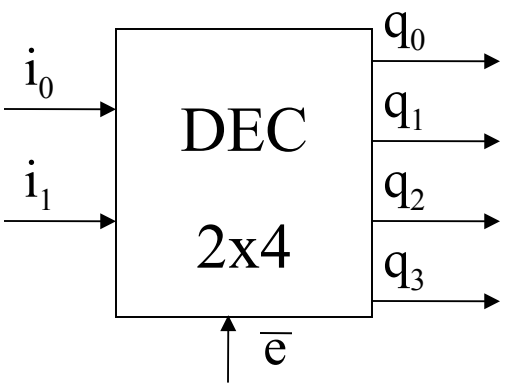
$q_0 = e\ i_1'\ i_0'$

$q_1 = e\ i_1'\ i_0$

$q_2 = e\ i_1\ i_0'$

$q_3 = e\ i_1\ i_0$

Il decodificatore 2x4 (in logica negata)



Descrizione:

```
if ( e==0 && i1==0&& i0==0) q0= 0
else   q0= 1
if ( e==0 && i1==0&& i0==1)q1= 0
else   q1= 1
if ( e==0 && i1==1&& i0==0) q2= 0
else   q2= 1
if ( e==0 && i1==1&& i0==1) q3= 0
else   q3= 1
```

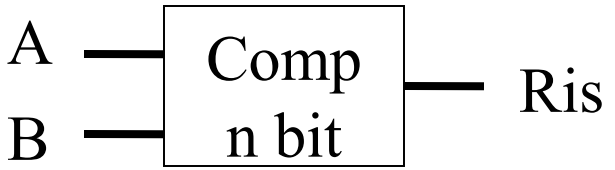
$q_0' = e' i_1' i_0'$ $\quad q_0 = e + i_1 + i_0$

$q_1' = e' i_1' i_0$ $\quad q_1 = e + i_1 + i_0'$

$q_2' = e' i_1 i_0'$ $\quad q_2 = e + i_1' + i_0$

$q_3' = e' i_1 i_0$ $\quad q_3 = e + i_1' + i_0'$

Comparatore a n bit (A==B)

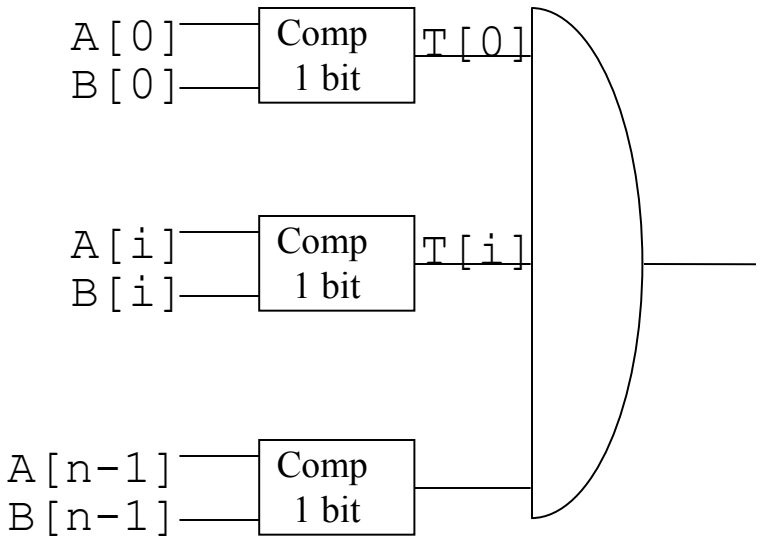



```
if(A==B) Ris=1;
else Ris=0;
```

Descrizione del comparatore

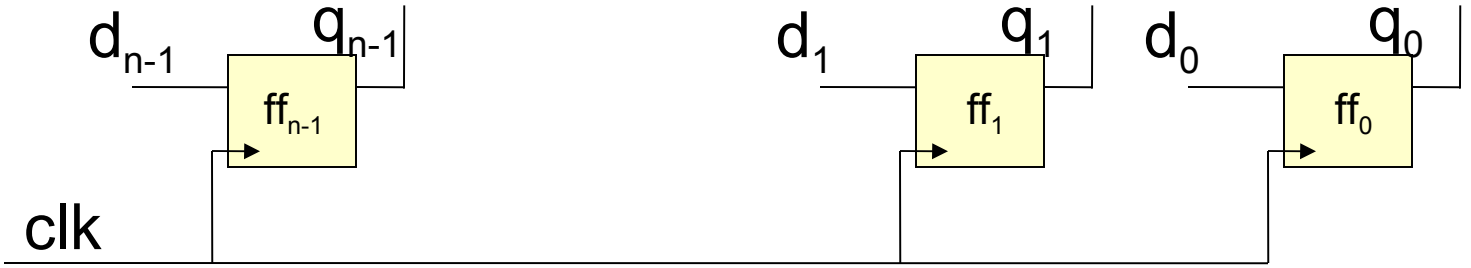
```
for(i=0;i<n-1;i++)
 T[i]=Comparatore_1Bit(A[i],B[i])

Ris[0]=T[0]T[1] ... T[n-2]T[n-1]
for(i=1;i<=n-1;i++)
  Ris[i]=0;
```



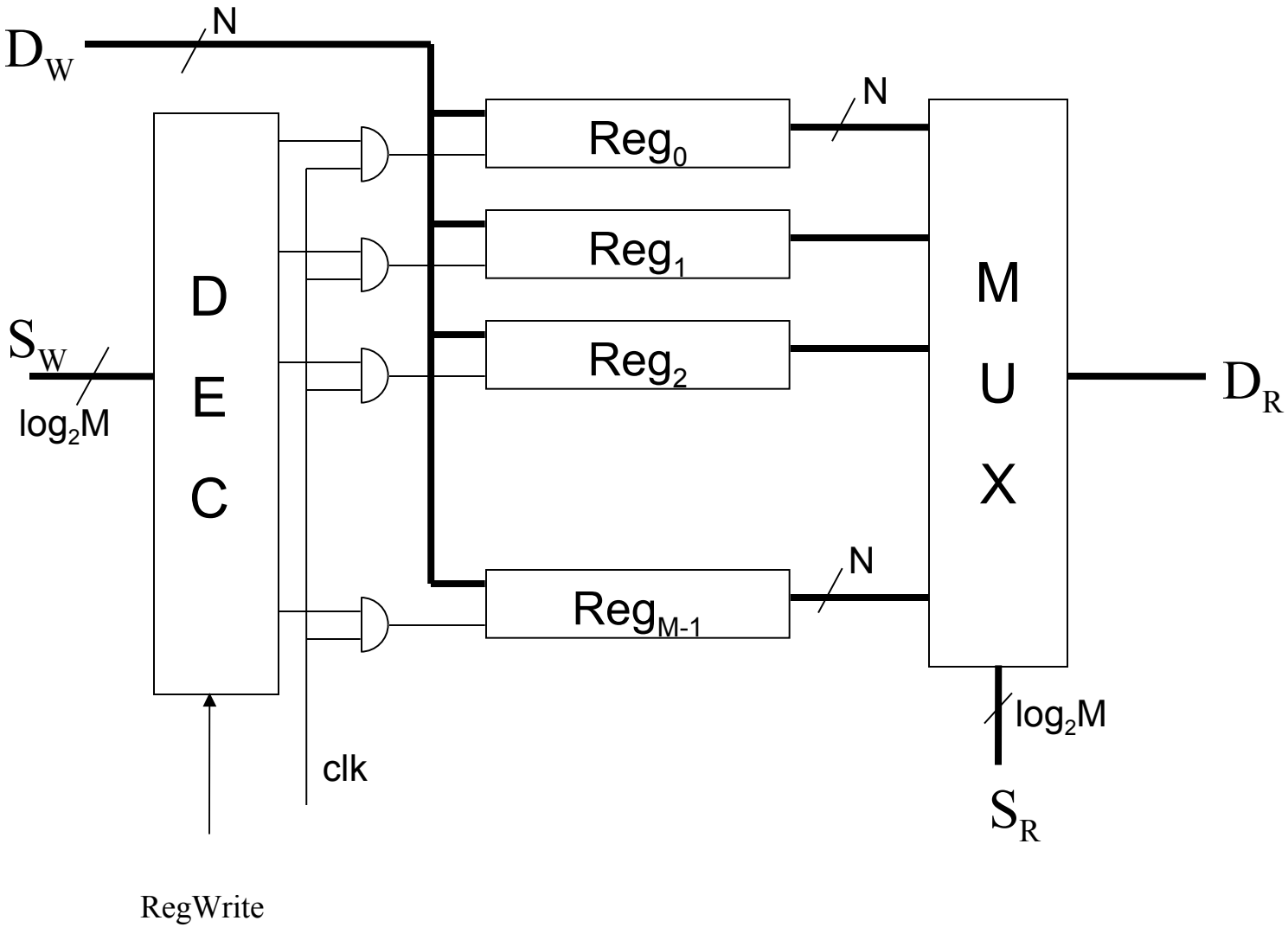
Registro

Un registro di n bit e' un vettore di n ff



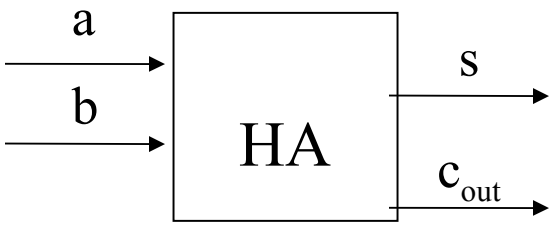
Il registro di n bit lo rappresenteremo come segue:

Il Register file



Sommatore 1 bit (Half/Full Adder)

Half adder

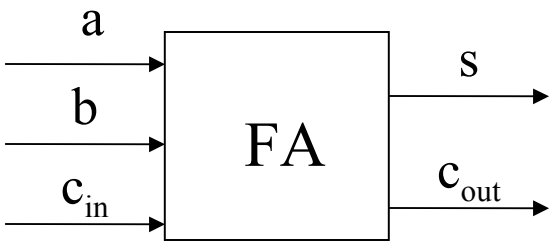


$s = a'b + ab'$

$c_{out} = ab$

a	b	s	c_{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full adder

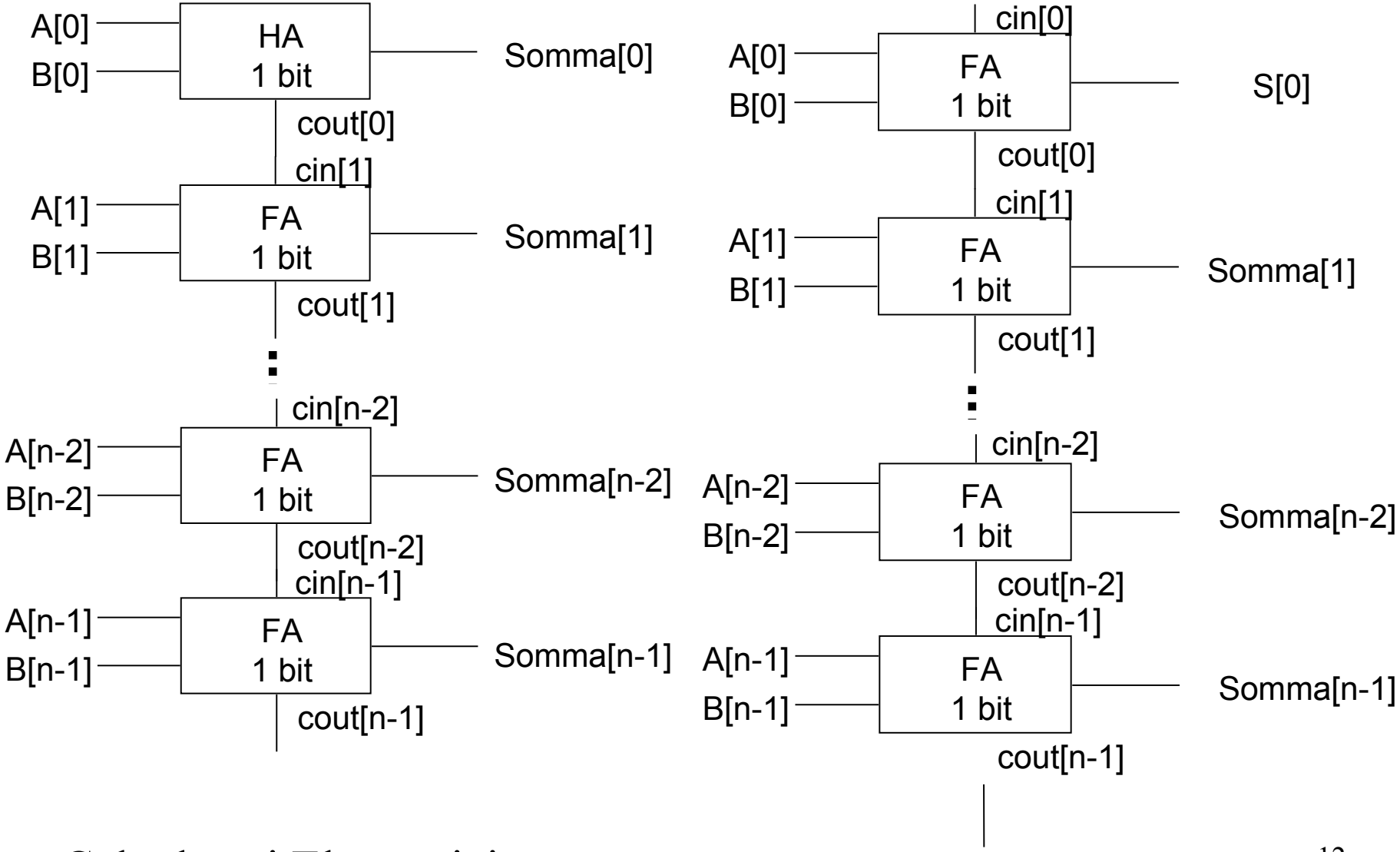


$s = a'b'c_{in} + a'bc_{in}' + abc_{in} + ab'c_{in}'$

$c_{out} = \mathbf{bc_{in}} + ab + ac_{in}$

a	b	c_{in}	s	c_{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

Sommatore a n bit
Propagazione di riporto



Sommatori con anticipo di riporto

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

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

$$a_i b_i \qquad a_i \oplus b_i$$

Sommatori con anticipo di riporto

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

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

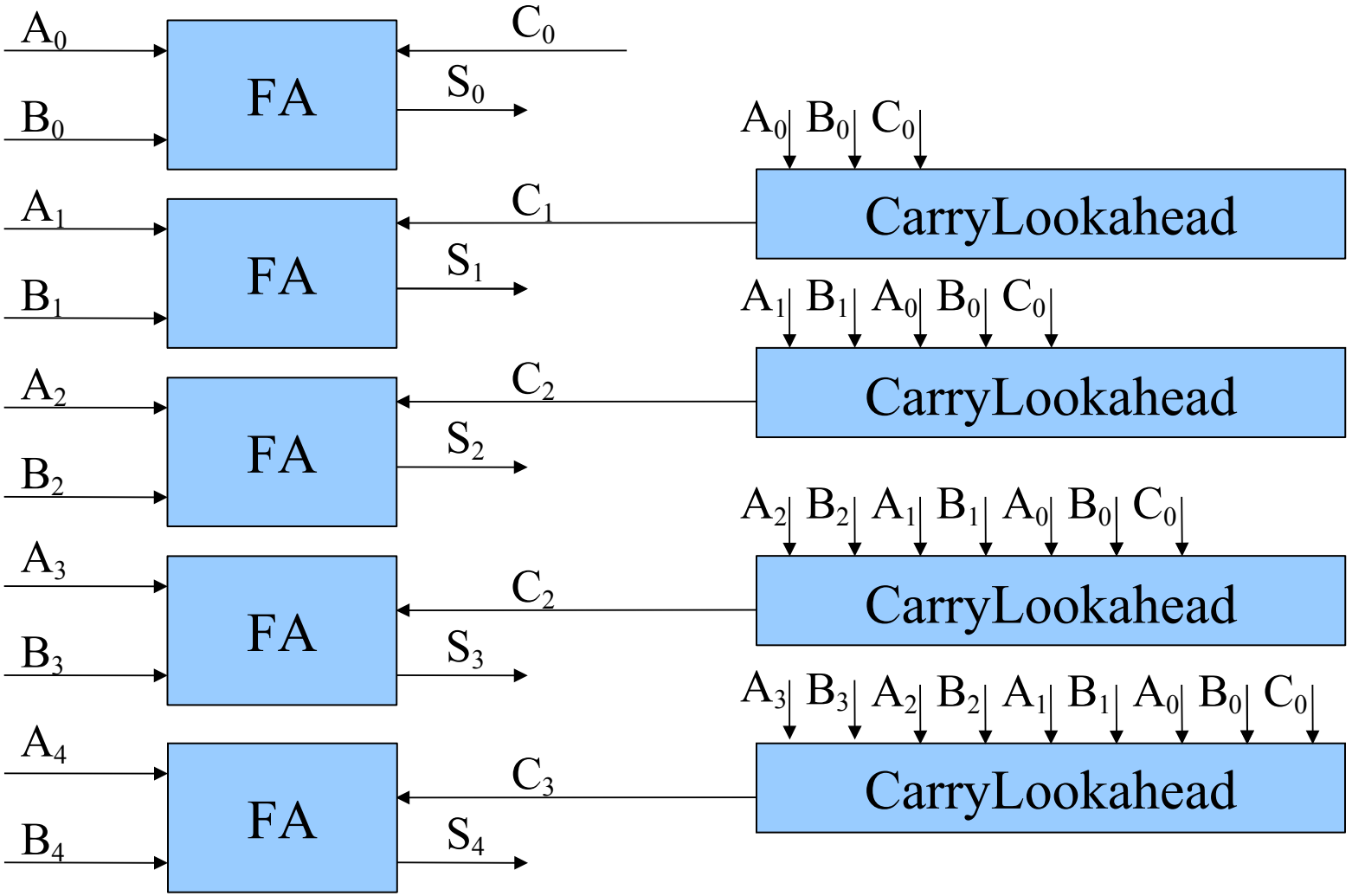
Sommatore con anticipo di riporto

$C_1 = G_0 + P_0 C_0 = A_0 B_0 + (A_0 \oplus B_0) C_0$

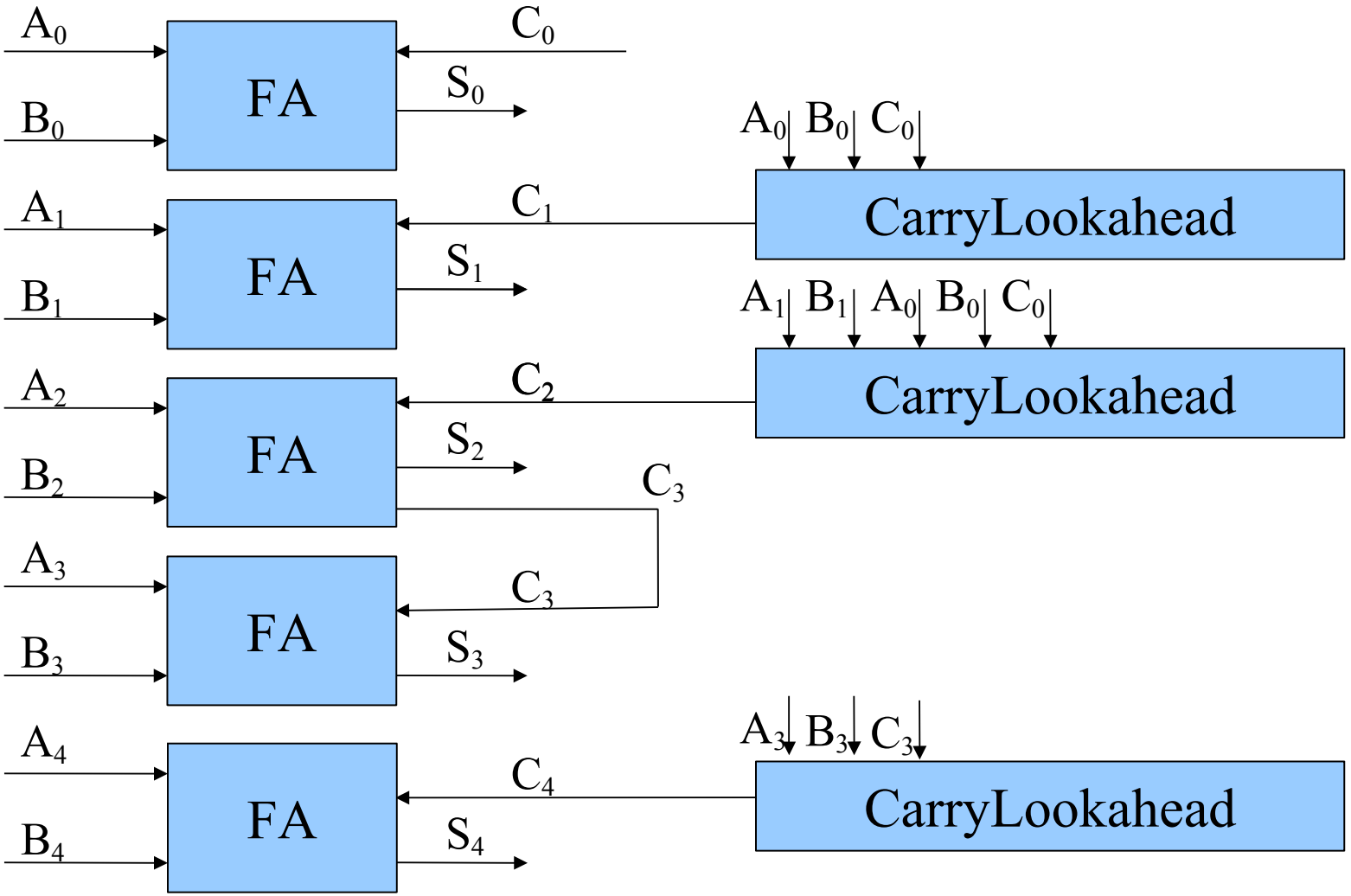
$C_2 = G_1 + P_1 C_1 = A_1 B_1 + (A_1 \oplus B_1)(A_0 B_0 + (A_0 \oplus B_0) C_0) =$
$= A_1 B_1 + (A_1 \oplus B_1)(A_0 B_0) + (A_1 \oplus B_1)(A_0 \oplus B_0) C_0$

$C_3 = G_2 + P_2 C_2 =$
$= A_2 B_2 + (A_2 \oplus B_2)(A_1 B_1 + (A_1 \oplus B_1)(A_0 B_0 + (A_0 \oplus B_0) C_0) =$
$= A_2 B_2 + (A_2 \oplus B_2)(A_1 B_1) + (A_2 \oplus B_2)(A_1 \oplus B_1)(A_0 B_0)$
$+ (A_2 \oplus B_2)(A_1 \oplus B_1)(A_0 \oplus B_0) C_0$

Sommatore con anticipo di riporto



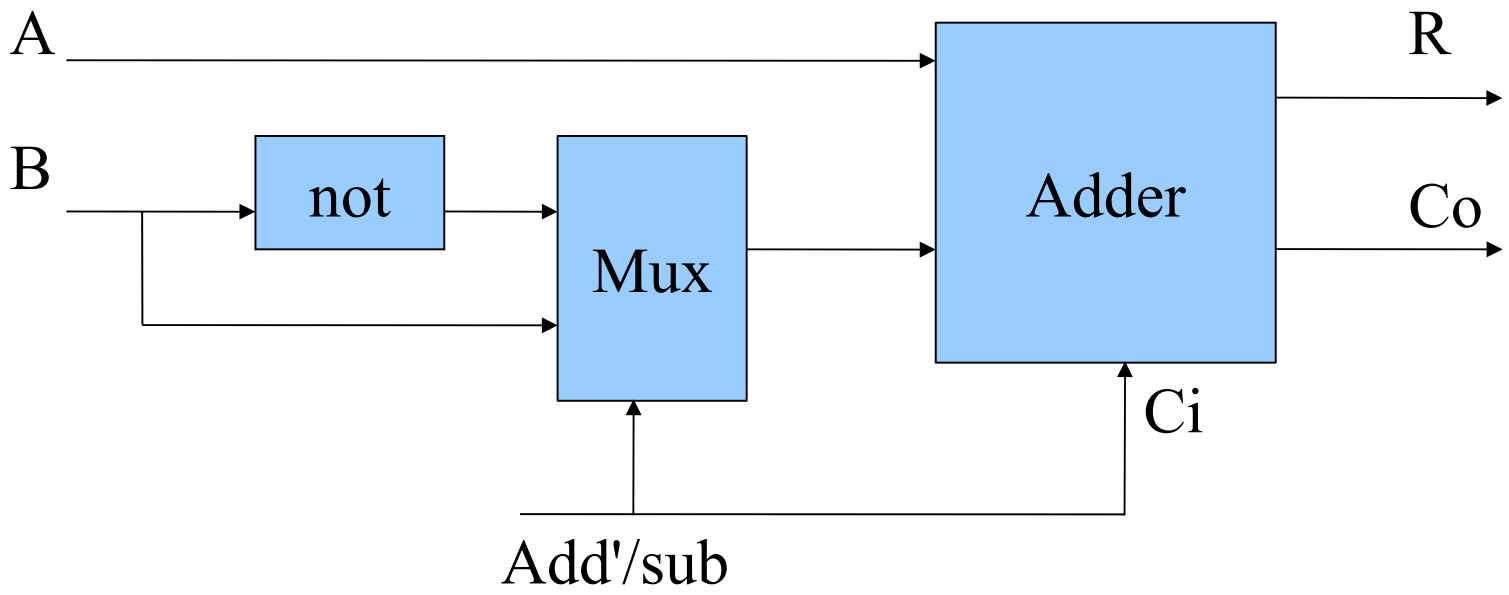
Sommatore con anticipo di riporto



Differenza

A - B = A + (-B) = A + C2(B)

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



Realizzazione di una ALU a n bit