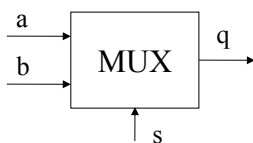


Componenti di un sistema digitale

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:

```
q <= a when s = '0' else
  b;
```

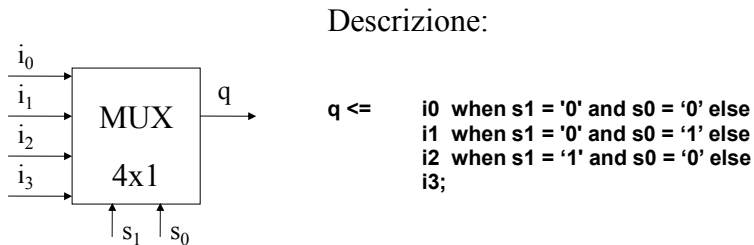
Tabella della verità

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

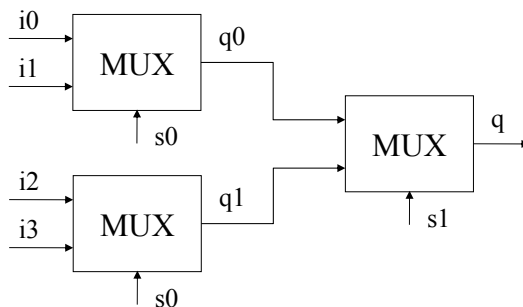
Con 4 linee di ingresso sono richieste due linee di comando



$$f(i_3, i_2, i_1, i_0, s_1, s_0) = i_3 s_1 s_0 + i_2 s_1 s_0' + i_1 s_1' s_0 + i_0 s_1' s_0'$$

Il Multiplexer 4x1

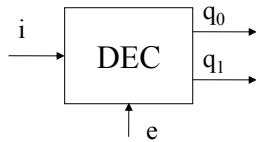
Può essere realizzato a partire da 3 mux 2x1



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



$$q_0 = ei'$$

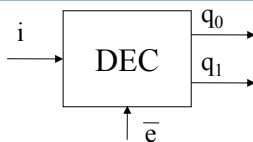
$$q_1 = ei$$

Descrizione:

```
q0 <= '1' when e = '1' and i='0' else '0';
```

```
q1 <= '1' when e = '1' and i='1' else '0';
```

Il decodificatore (in logica negata)



Descrizione:

```
q0 <= '0' when e = '0' and i='0' else '1';
```

```
q1 <= '0' when e = '0' and i='1' else '1';
```

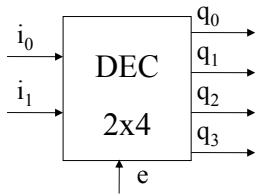
$$q_0' = e'i'$$

$$q_0 = e+i$$

$$q_1' = e'i$$

$$q_1 = e+i'$$

Il decodificatore 2x4 (in logica diretta)



Descrizione:

```
q0 <= '1' when e = '1' and i0='0' and i1='0' else '0';
```

```
q1 <= '1' when e = '1' and i0='0' and i1='1' else '0';
```

```
q2 <= '1' when e = '1' and i0='1' and i1='0' else '0';
```

```
q3 <= '1' when e = '1' and i0='1' and i1='1' else '0';
```

$$q_0 = e \cdot i_1' \cdot i_0'$$

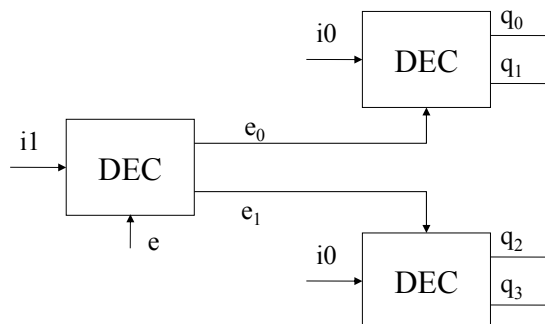
$$q_1 = e \cdot i_1' \cdot i_0$$

$$q_2 = e \cdot i_1 \cdot i_0'$$

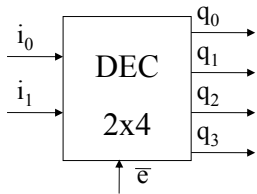
$$q_3 = e \cdot i_1 \cdot i_0$$

Il decodificatore 2x4

Può essere realizzato a partire da 3 decoder 1x2



Il decodificatore 2x4 (in logica negata)



Descrizione:

$q_0 \leftarrow '0'$ when $e = '0'$ and $i_0 = '0'$ and $i_1 = '0'$ else $'1'$;

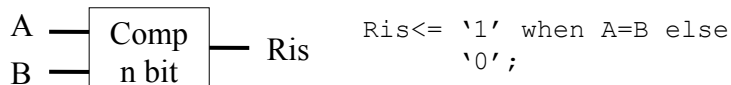
$q_1 \leftarrow '0'$ when $e = '0'$ and $i_0 = '0'$ and $i_1 = '1'$ else $'1'$;

$q_2 \leftarrow '0'$ when $e = '0'$ and $i_0 = '1'$ and $i_1 = '0'$ else $'1'$;

$q_3 \leftarrow '0'$ when $e = '0'$ and $i_0 = '1'$ and $i_1 = '1'$ else $'1'$;

$$\begin{aligned} q_0' &= e' i_1' i_0' & q_0 &= e + i_1 + i_0 \\ q_1' &= e' i_1' i_0 & q_1 &= e + i_1 + i_0' \\ q_2' &= e' i_1 i_0' & q_2 &= e + i_1' + i_0 \\ q_3' &= e' i_1 i_0 & q_3 &= e + i_1' + i_0' \end{aligned}$$

Comparatore a n bit (A==B)



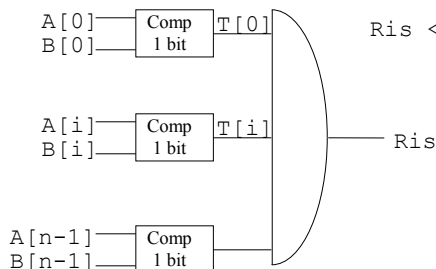
$Ris \leftarrow '1'$ when $A=B$ else $'0'$;

Descrizione del comparatore

for i IN 0 to $n-1$ LOOP

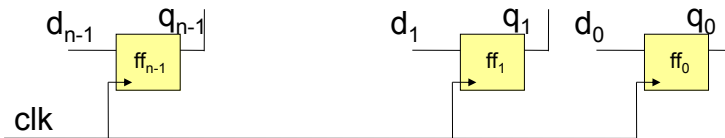
$T[i] := \text{Comparatore_1Bit}(A[i], B[i])$

$Ris \leftarrow T[0] \text{ and } T[1] \text{ and } \dots \text{ and } T[n-1]$



Registro

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



Il registro di n bit lo rappresenteremo come segue:



Il register File: array di registri

Letture

S_R = indice reg. da leggere

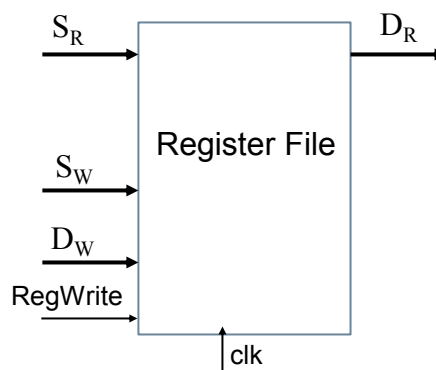
$D_R = \text{Reg}(S_R)$

Scrittura

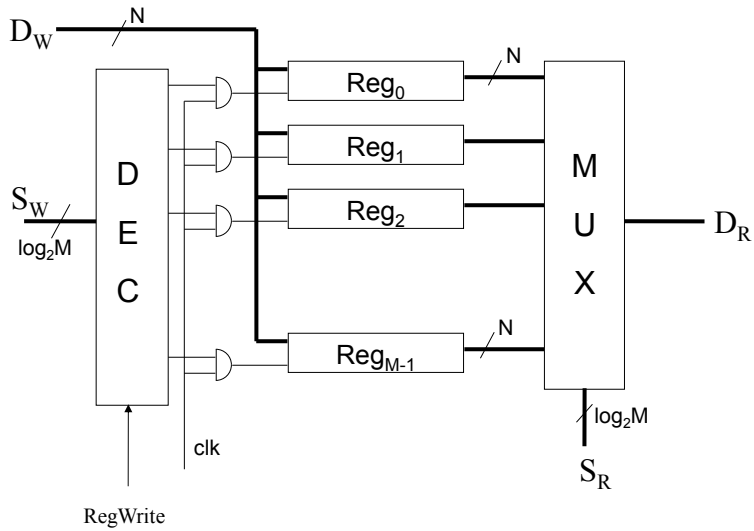
S_W = indice reg. in cui scrivere

$\text{Reg}(S_W) = D_W$

RegWrite: abilita scrittura

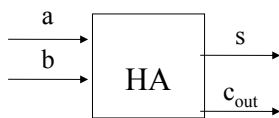


II Register file



Sommatore 1 bit (Half/Full Adder)

Half adder

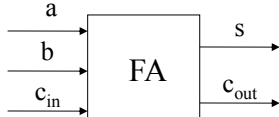


$$s = a'b + ab'$$

$$c_{out} = ab$$

a	b	s	c _{out}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full adder

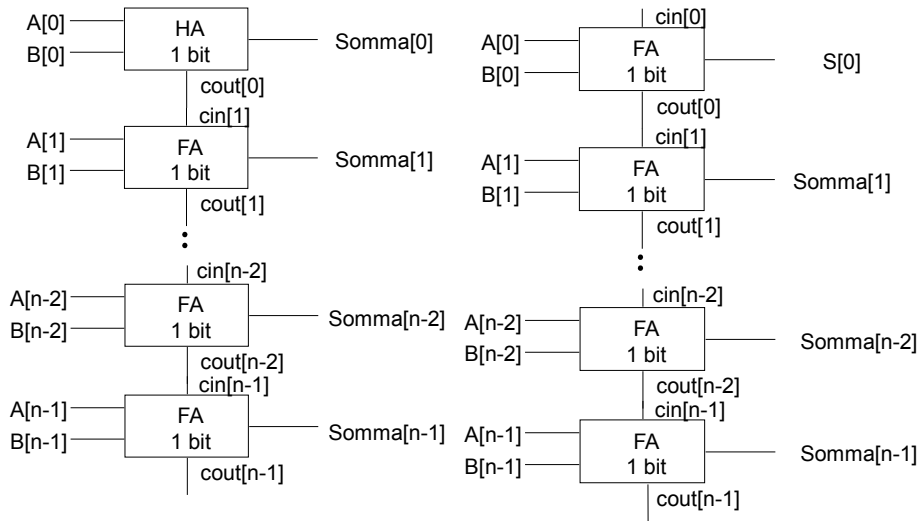


$$s = a'b'c_{in} + a'bc_{in}' + abc_{in} + ab'c_{in}'$$

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

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

$$C_{i+1} = a_i b_i + a_i' b_i C_i + a_i b_i' C_i = a_i b_i + (a_i \oplus b_i) C_i$$

■ Una componente generata localmente $a_i b_i$

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

$$C_{i+1} = G_i + P_i C_i$$

\downarrow \downarrow
 $a_i b_i$ $a_i \oplus b_i$

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

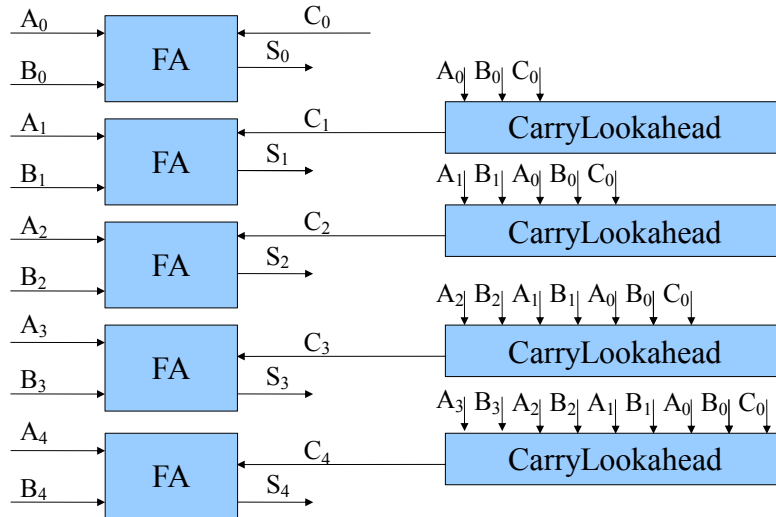
Sommatore con anticipo di riporto

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

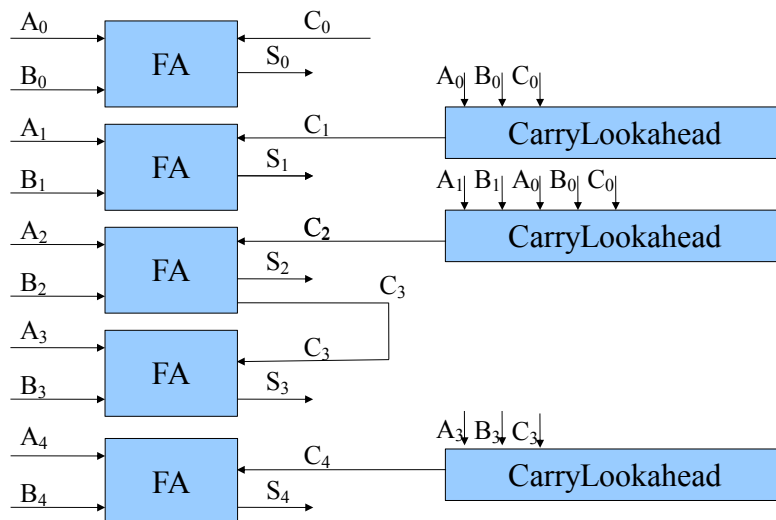
$$\begin{aligned} C_2 &= G_1 + P_1 C_1 = A_1 B_1 + (A_1 \oplus B_1) (A_0 B_0 + (A_0 \oplus B_0) C_0) = \\ &= A_1 B_1 + (A_1 \oplus B_1) (A_0 B_0) + (A_1 \oplus B_1) (A_0 \oplus B_0) C_0 \end{aligned}$$

$$\begin{aligned} C_3 &= G_2 + P_2 C_2 = \\ &= A_2 B_2 + (A_2 \oplus B_2) (A_1 B_1 + (A_1 \oplus B_1) (A_0 B_0 + (A_0 \oplus B_0) C_0)) = \\ &= A_2 B_2 + (A_2 \oplus B_2) (A_1 B_1) + (A_2 \oplus B_2) (A_1 \oplus B_1) (A_0 B_0) \\ &\quad + (A_2 \oplus B_2) (A_1 \oplus B_1) (A_0 \oplus B_0) C_0 \end{aligned}$$

Sommatore con anticipo di riporto



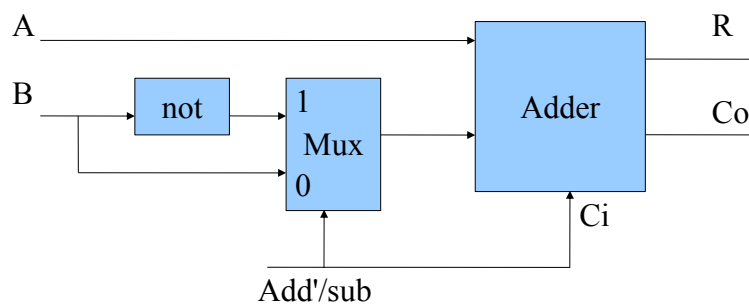
Sommatore con anticipo di riporto



Differenza

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

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



Overflow

- 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

	0	1	0	1	5-	A-
	0	0	1	1	3 =	B =
	0	0	1	0	2	A-B



Overflow						
	0	1	0	1	5+	A+
	1	1	0	1	-3 =	C(B)
0	0	0	1	0	2	A+C(B)

0

Possiamo ignorare il bit di riporto

Overflow

- Se i due numeri hanno lo stesso segno è possibile avere overflow

Overflow						
	0	1	1	1	7+	A+
	0	1	1	1	7=	B=
0	1	1	1	0	14	A+B

Overflow						
	1	0	0	1	-7	C(A)+
	1	0	0	1	-7	C(B)
1	0	0	1	0	-14	C(A)+C(B)

Non possiamo ignorare il bit di riporto

Calcolatori Elettronici

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

Calcolatori Elettronici

Prodotto

- Due fasi
- Calcolo dei prodotti parziali
 - ▣ Uguale per tutti i moltiplicatori
- Somma dei prodotti parziali
 - ▣ Somma per righe
 - ▣ Somma per diagonali

Calcolatori Elettronici

Prodotti parziali

				A3	A2	A1	A0
				B3	B2	B1	B0
				<hr/>			
				A3 B0	A2 B0	A1 B0	A0 B0
			A3 B1	A2 B1	A1 B1	A0 B1	
		A3 B2	A2 B2	A1 B2	A0 B2		
	A3 B3	A2 B3	A1 B3	A0 B3			
<hr/>							
S7	S6	S5	S4	S3	S2	S1	S0

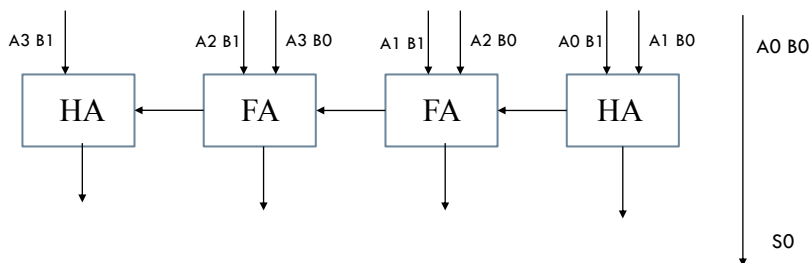
Calcolatori Elettronici

Somma per riga

- $S_0 = A_0 B_0$
- $S_1 = (A_1 B_0 + A_0 B_1)$ con riporto C_{11}
- $S_2 = [(A_2 B_0 + A_1 B_1 + C_{11}) + A_0 B_2]$ con riporti C_{21} e C_{22}
- $S_3 = \{[(A_3 B_0 + A_2 B_1 + C_{21}) + A_1 B_2 + C_{22}] + A_0 B_3\}$ con riporti C_{31} , C_{32} , C_{33}
- $S_4 = \{[(A_3 B_1 + C_{31}) + A_2 B_2 + C_{32}] + A_1 B_3 + C_{33}\}$ con riporti C_{41} , C_{42} , C_{43}
- $S_5 = \{[A_3 B_2 + C_{41} + C_{42}] + A_2 B_3 + C_{43}\}$ con riporti C_{52} , C_{53}
- $S_6 = \{A_3 B_3 + C_{52} + C_{53}\}$ con riporto C_{63}
- $S_7 = C_{63}$

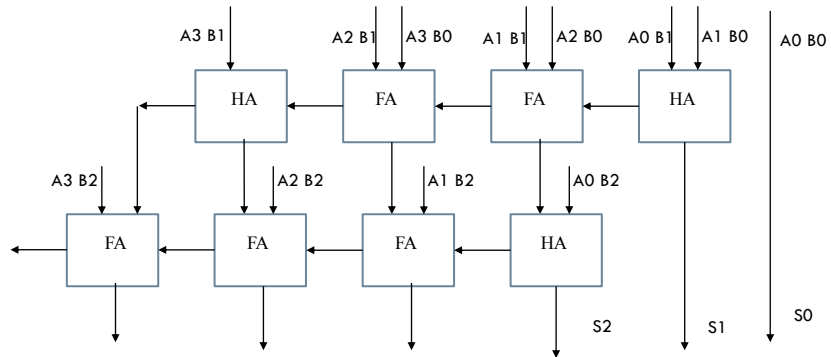
Calcolatori Elettronici

Somma per righe: prima riga



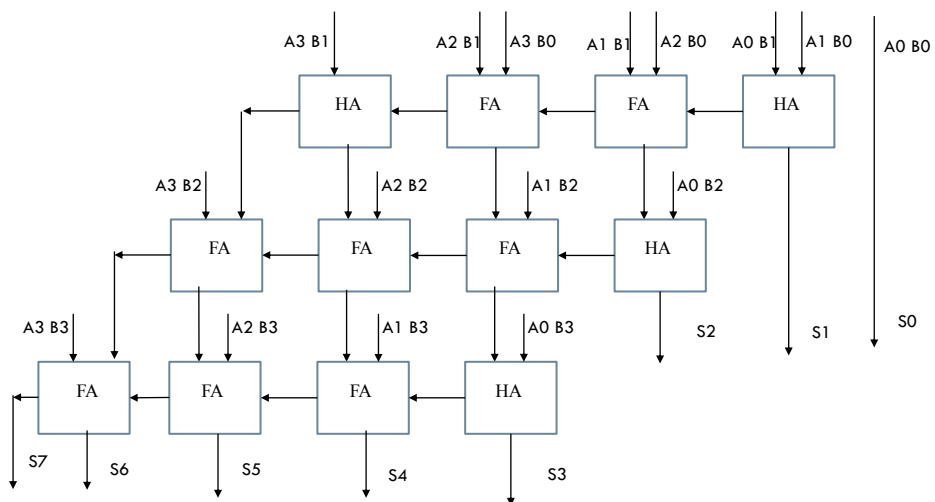
Calcolatori Elettronici

Somma per righe: seconda riga



Calcolatori Elettronici

Somma per righe: terza riga

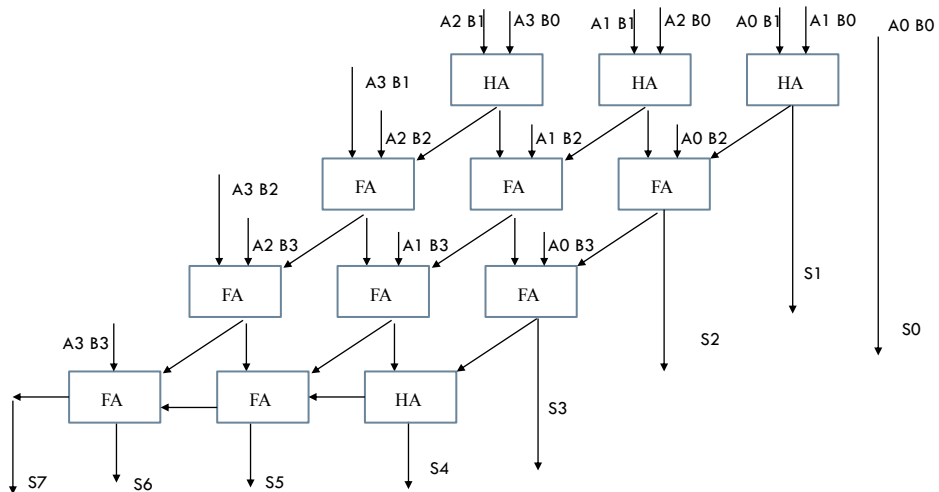


Ritardo somme per righe

- I tempi richiesti per la somma dei prodotti parziali dipendono *linearmente* dal numero dei bit dei fattori
- $\text{DelaySpR} = 3n - 4$

Calcolatori Elettronici

Somma per diagonale



Calcolatori Elettronici

Somma per diagonale

- $S_0 = A_0 B_0$
- $S_1 = (A_1 B_0 + A_0 B_1)$ con riporto C_{12}
- $S_2 = [(A_2 B_0 + A_1 B_1) + A_0 B_2 + C_{12}]$ con riporti C_{22} e C_{23}
- $S_3 = \{[(A_3 B_0 + A_2 B_1) + C_{22} + A_1 B_2] + C_{23} + A_0 B_3\}$ con riporti C_{32} , C_{33} , C_{34}
- $S_4 = \{[(A_3 B_1 + A_2 B_2 + C_{32}) + A_1 B_3 + C_{33}] + C_{34}\}$ con riporti C_{43} , C_{44} , C_{45}
- $S_5 = \{[A_3 B_2 + A_2 B_3 + C_{43}] + C_{44} + C_{45}\}$ con riporti C_{54} , C_{55}
- $S_6 = [A_3 B_3 + C_{54} + C_{55}]$ con riporto C_{65}
- $S_7 = C_{65}$

Calcolatori Elettronici

Ritardo somme per diagonale

- I tempi richiesti per la somma dei prodotti parziali dipendono *linearmente* dal numero dei bit dei fattori
- $\text{DelaySpD} = 2n - 2$

Calcolatori Elettronici

Realizzazione di una ALU a n bit

