



Università degli Studi di Milano
CdL in Informatica per la comunicazione digitale
AA 2025/2026

Architettura degli elaboratori - Lez 6 - A:

Blocchi funzionali
combinatori

Marco Tarini
marco.tarini@unimi.it

1

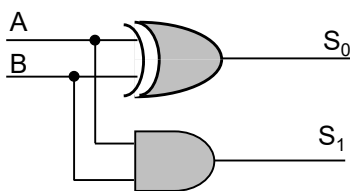


Semi-addizionatore
(«half-hadder»)

- Dati due numeri naturali rappresentati su un solo bit, il circuito «half-hadder» ne calcola la somma S (codificata come numero su due bit: $S_1 S_0$)

Tavola delle verità

A	B	S_1	S_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



il riporto ("carry")

Architettura degli elaboratori

- 2 -

Blocchi funzionali combinatori

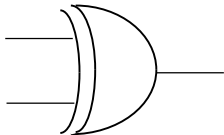
2



Nota: il bit meno significativo dei due, S_0 è dato da un singolo operatore XOR !

XOR
(«or esclusivo»)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0



Significati intuitivi di **A XOR B**:

- A oppure B, ma non entrambi
(in latino: A **aut** B)
- vero se A e B diversi
falso se A e B uguali
- vale $\neg A$ se $B = 1$
vale A se $B = 0$
▶ (e viceversa)
- il contrario di A, se B vale;
A immutato, altrimenti
▶ (e viceversa)
- ...
- **somma di A e B intesi come numeri di... 1 bit! (ignorando il riporto)**

Architettura degli elaboratori

- 3 -

Porte logiche

3



In pratica, lo half-adder implementa la “tabellina” della somma in base 2

+	0	1
0	00	01
1	01	10

il riporto (“carry”)

Architettura degli elaboratori

- 4 -

Aritmetica binaria

4



Full-adder

- Dati *tre* numeri naturali rappresentati su un solo bit, questo circuito ne calcola la somma S (codificata con due BIT)

Tavola delle verità

A	B	C	S ₁	S ₀
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S_1 = ABC + AB/C + /ABC + A/BC$$
$$= AB(C+/C) + C (/AB + A/B)$$
$$= AB + C (A\oplus B)$$
$$S_0 = A \oplus B \oplus C$$

xor

Architettura degli elaboratori

- 5 -

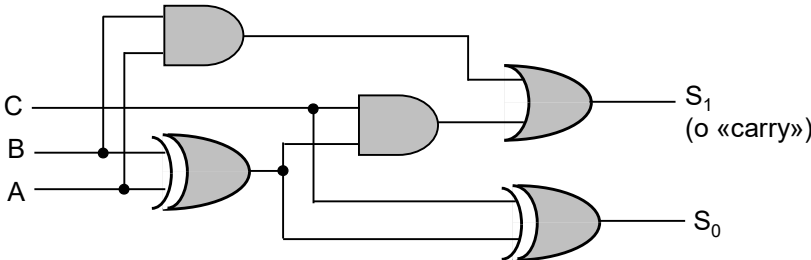
Blocchi funzionali combinatori

5



Full Adder: esempio di implementazione

$$S_1 = ABC + AB/C + /ABC + A/BC$$
$$= AB(C+/C) + C (/AB + A/B)$$
$$= AB + C (A\oplus B)$$
$$S_0 = A \oplus B \oplus C$$

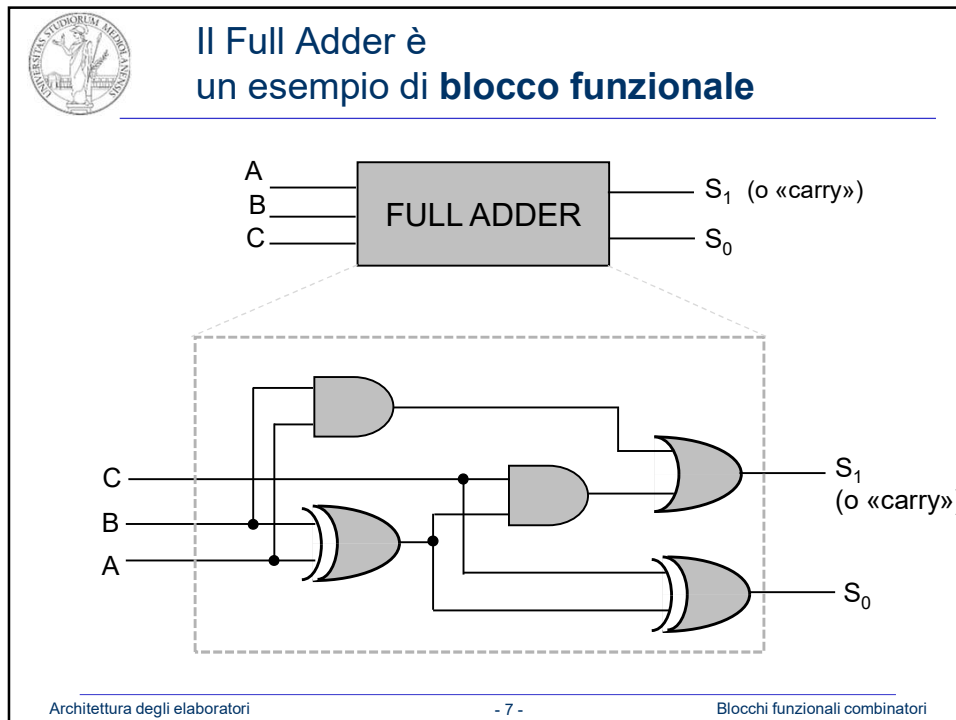


Architettura degli elaboratori

- 6 -

Blocchi funzionali combinatori

6



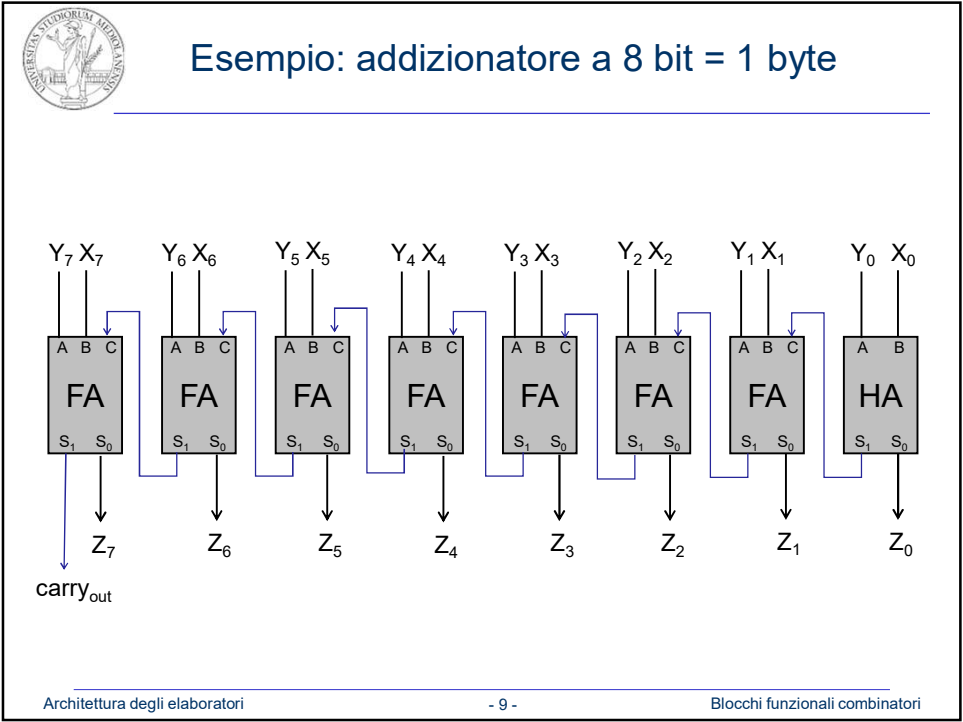
7

 **Circuito addizionatore di interi a n bit: note per il lucido successivo 1/2**

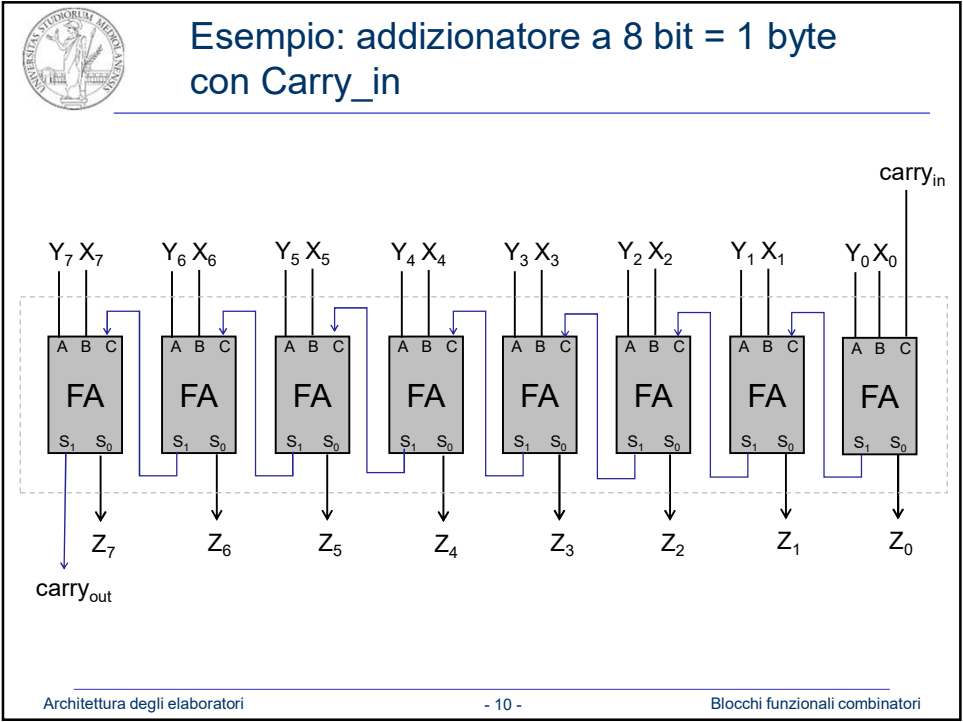
- Componendo insieme full-adder e half-adder posso costruire circuiti che sommano numeri in base 2 a n bit
 - Questo ricalca il procedimento comune per sommare i numeri «in colonna» (con i riporti)
 - Il primo Half-Adder calcola la somma della colonna «più a destra» (i due bit meno significativi)
 - Ogni Full-Adder successivo calcola i due bit successivi, considerando il riporto
- Il circuito risultante
 - prende in ingresso due numeri naturali X e Y da n 1 bit ciascuno
 - presenta in uscita la loro somma a Z , da n bit ...
 - ... e anche anche un «riporto in uscita» (Carry Out)
 - ha quindi $n+1$ uscite in tutto
- Un addizionatore ottenuto in questo modo non è molto efficiente, ma è economico
 - Il riporto deve attraversare tutto il circuito
- Su uso un full-adder anche per la prima cifra, il circuito prende anche un riporto in ingresso (Carry-in) che viene normalmente messo a zero
 - Il circuito somma anche il valore del Carry-in (0 o 1) all'output

Architettura degli elaboratori - 8 - Blocchi funzionali combinatori

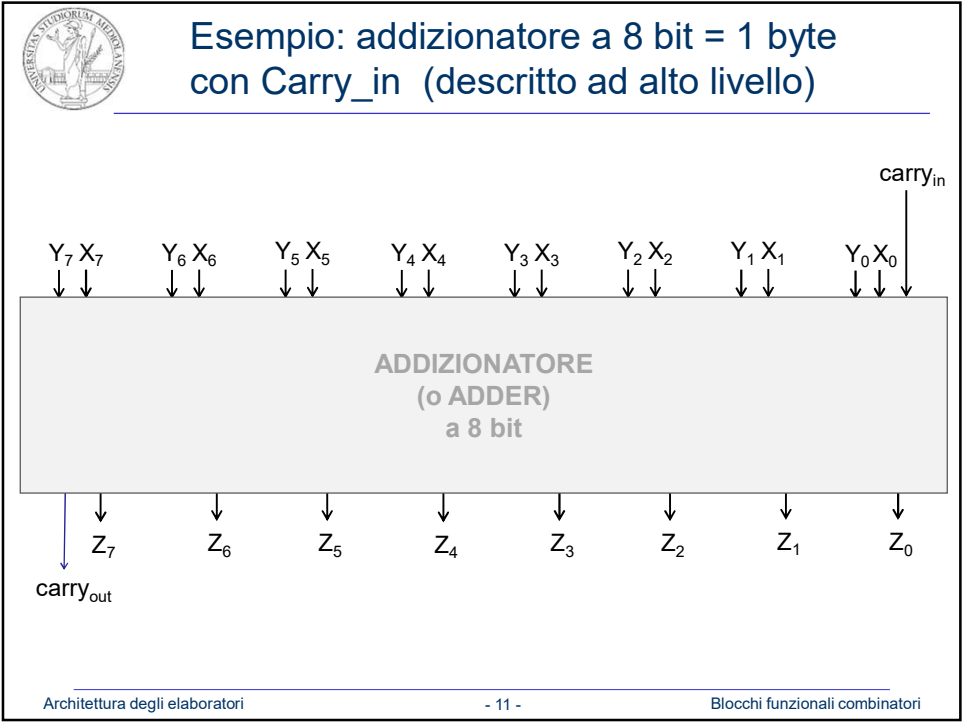
8



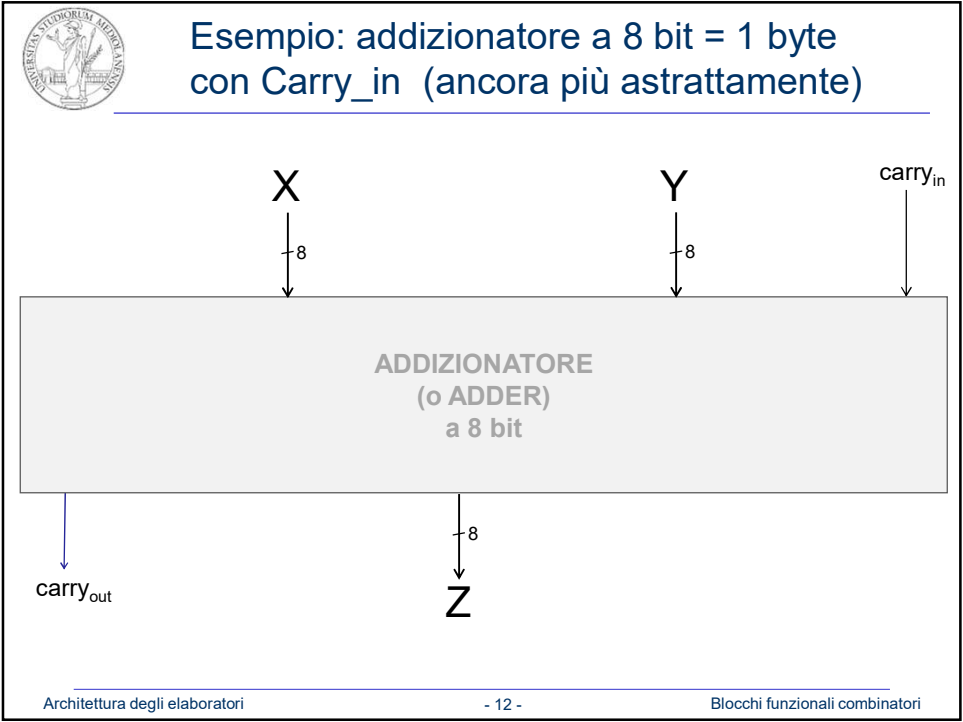
9



10



11



12



Circuito addizionatore di interi a n bit: osservazione

- Come sappiamo, lo stesso circuito funziona anche per **sommare due numeri con segno** in **complemento a due**!
- Unica differenza:
 - ▶ Quando sommo due numeri naturali: il bit «Carry Out» segnala l'overflow
 - ▶ Quando sommo due numeri in CP2: il bit «Carry Out» va ignorato

Architettura degli elaboratori

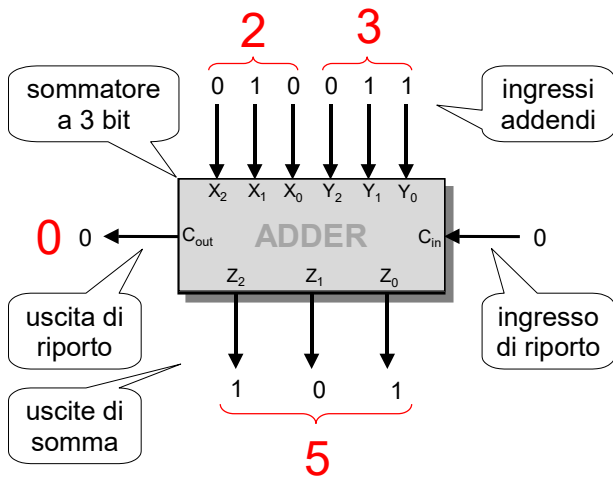
- 13 -

Blocchi funzionali combinatori

13



Esempio di uso di un addizionatore a 3 bit: $2 + 3 = 5$

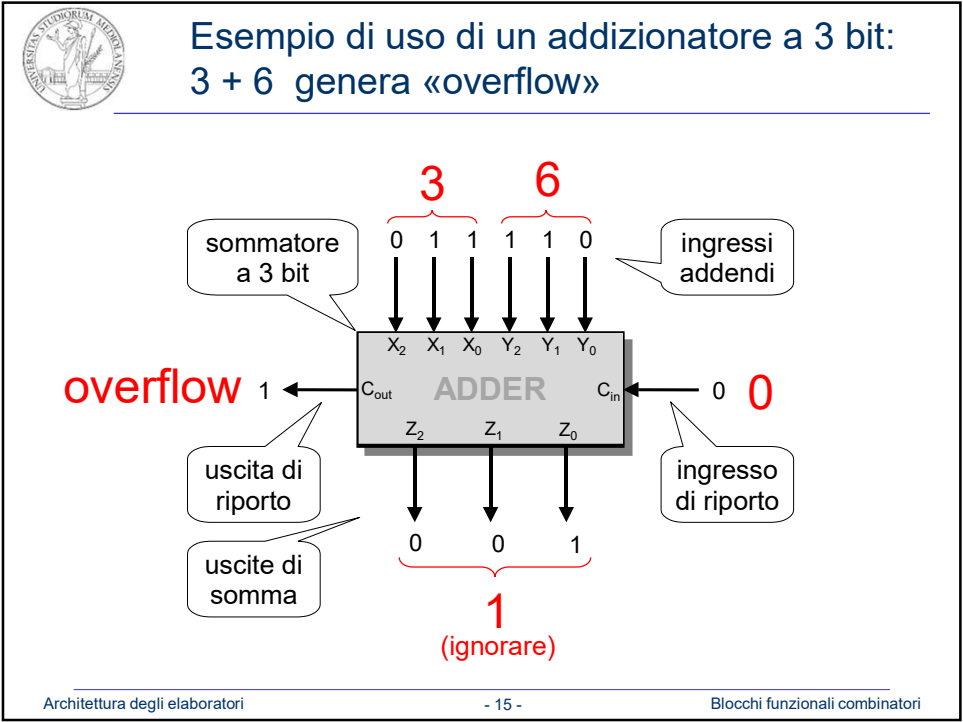


Architettura degli elaboratori

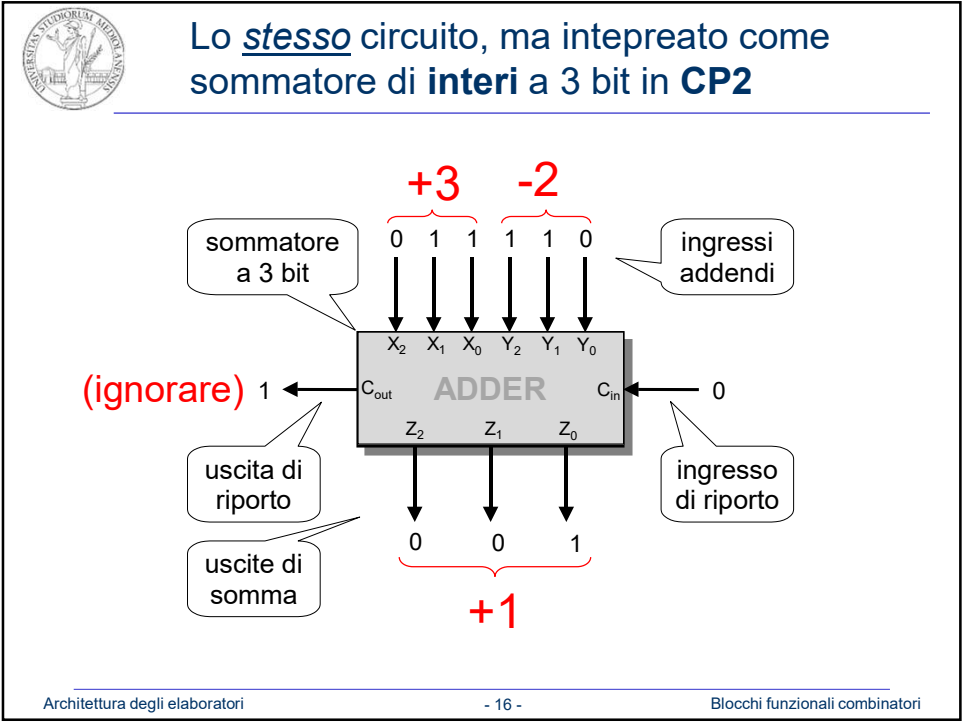
- 14 -

Blocchi funzionali combinatori

14



15



16



Blocchi funzionali combinatori

- La realizzazione di moderni circuiti integrati richiede la sintesi di reti *molto* più grandi di quelle viste fin'ora
- Gestiamo la complessità attraverso livelli di astrazione
- Idea: combinare reti piccole in reti più grandi
 - blocco funzionale** = una sotto rete standard
- Ne sono esempi quelli visti:
 - il **Full Adder**
 - lo **Half-Adder**
(componibile come visto per costruire un addizionatore a n bits)
 - O a livello ancora più alto, un **adder** a n bits (come vedremo, componibile per ottenere altri addizionatori a 2n bit)
- Esiste insieme di blocchi funzionali standard che risolvono problemi comuni, e risultano utili in contesti diversi. Vediamoli.
- (Esistono anche blocchi funzionali di tipo sequenziale, cioè non combinatorio, che vedremo in seguito)

Architettura degli elaboratori

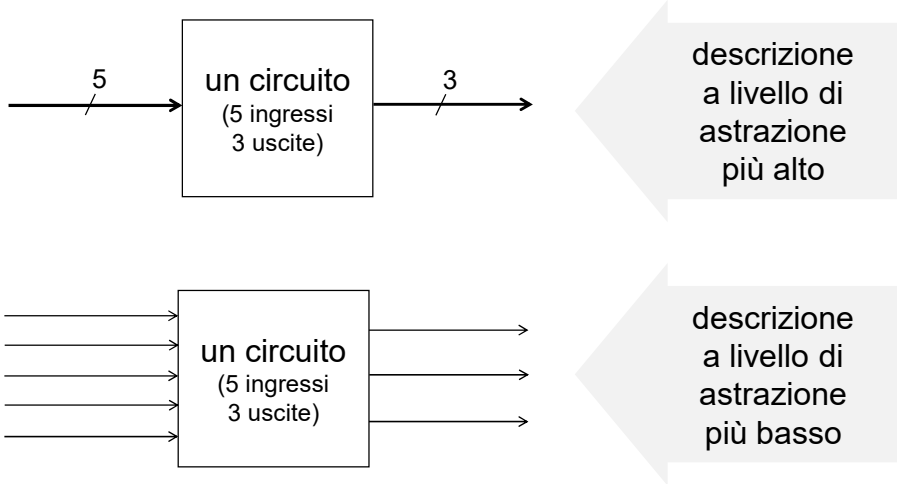
- 24 -

Blocchi funzionali combinatori

24



Livelli di astrazione: per i cavi di ingresso / uscita



Architettura degli elaboratori

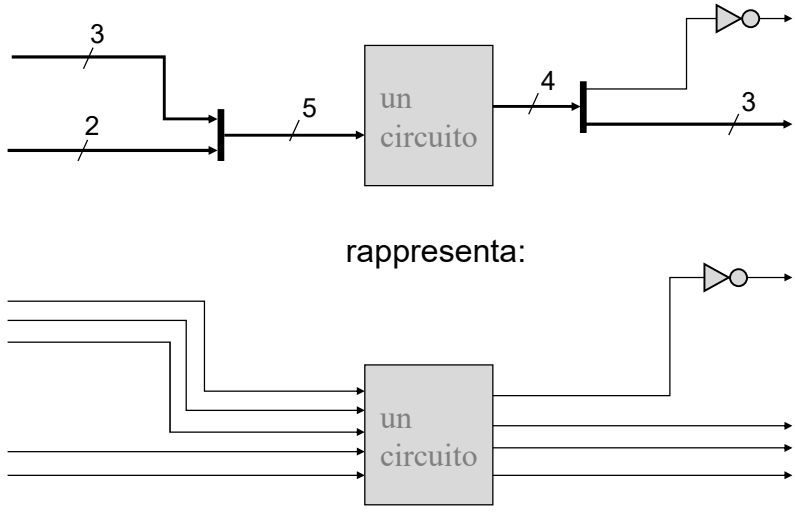
- 25 -

Blocchi funzionali combinatori

25



Livelli di astrazione: per i cavi di ingresso / uscita



Architettura degli elaboratori

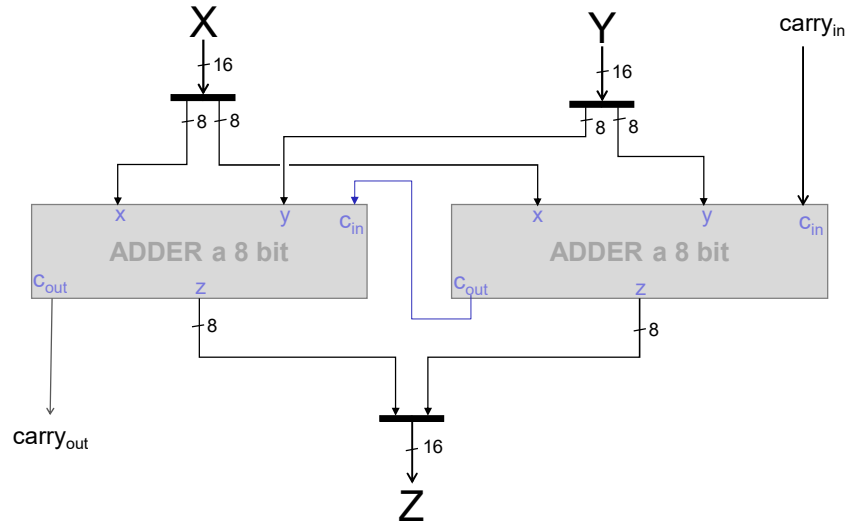
- 26 -

Blocchi funzionali combinatori

26



Esempio: addizionatore a 16 bit costruito con due addizionatori a 8 bits



Architettura degli elaboratori

- 27 -

Blocchi funzionali combinatori

27



Decoder (decodificatore)

- Il blocco funzionale «decodificatore» ha:
 - $n \geq 1$ ingressi
 - 2^n uscite
- Le uscite sono numerate a partire da 0
 - $k = 0, 1, 2, \dots, 2^n - 1$
- Se sugli ingressi è presente il numero binario k
 - la k -esima uscita assume il valore 1
 - e
 - le restanti uscite assumono il valore 0

Architettura degli elaboratori

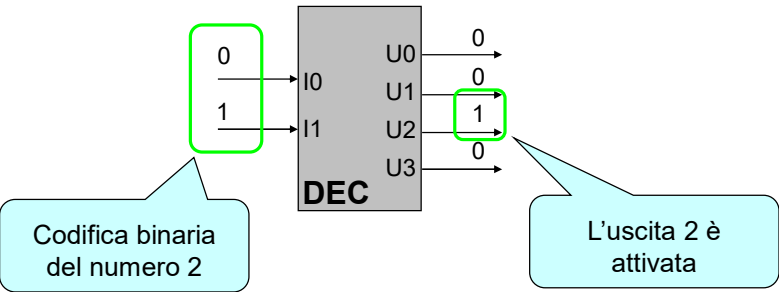
- 28 -

Blocchi funzionali combinatori

28



Decoder a 2 ingressi



Codifica binaria del numero 2

L'uscita 2 è attivata

Architettura degli elaboratori

- 29 -

Blocchi funzionali combinatori

29



Decoder a 2 ingressi

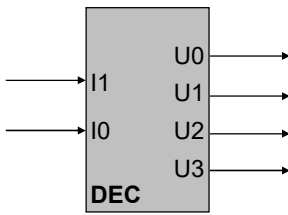


Tabella delle verità

I1	I0	U0	U1	U2	U3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Architettura degli elaboratori

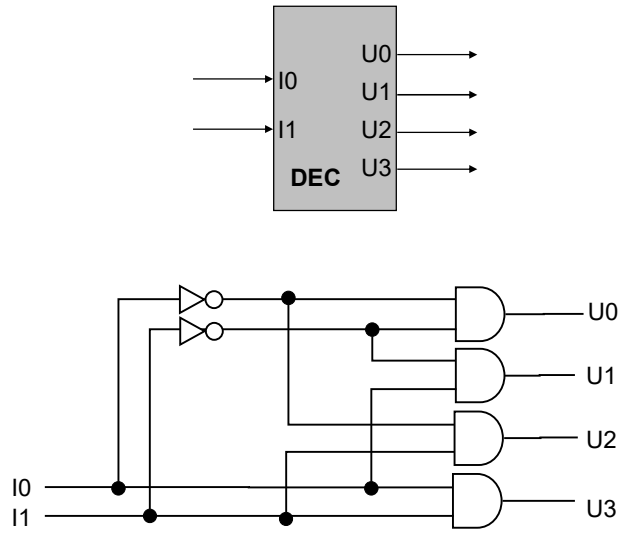
- 30 -

Blocchi funzionali combinatori

30



Decoder a 2 ingressi: una realizzazione



Architettura degli elaboratori

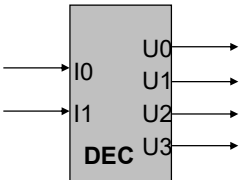
- 31 -

Blocchi funzionali combinatori

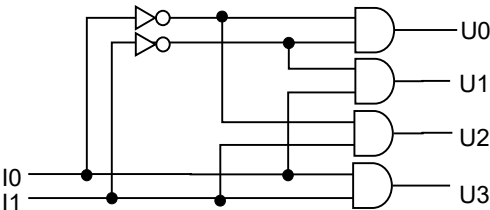
31



Decoder a 2 ingressi: una realizzazione



descrizione
a livello di
astrazione
più alto



descrizione
a livello di
astrazione
più basso

Architettura degli elaboratori

- 32 -

Blocchi funzionali combinatori

32



Codificatore (encoder)

- Il circuito codificatore è l'esatto contrario del decodificatore
- Prende in input n bit, di cui uno solo vale 1 (gli altri valgono tutti 0) e restituisce in output la codifica binaria dell'indice del bit ad 1
- Nota: non importa cosa fa il codificatore per gli input «invalidi» (in cui il numero di bit ad 1 non sia esattamente uno)
 - Nella tabella di verità, segniamo questi bit come X («don't care»)
 - Come sappiamo la presenza di simboli «don't care» può essere utilizzata per ottimizzare i circuiti

Es: Tabella di verità
di un ENCODER a 3 bit

INPUT	OUTPUT
0 0 0	X X
0 0 1	0 0
0 1 0	0 1
0 1 1	X X
1 0 0	1 0
1 0 1	X X
1 1 0	X X
1 1 1	X X

Architettura degli elaboratori

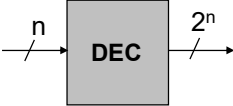
- 33 -

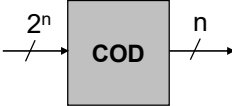
Blocchi funzionali combinatori

33



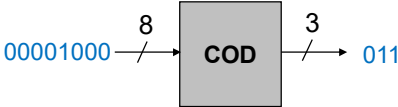
Decodificatore e Codificatore





Esempio con $n = 3$:





Architettura degli elaboratori

- 34 -

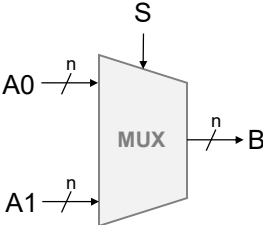
Blocchi funzionali combinatori

34



Multiplexer (selezionatore) a due vie

Rappresentazioni grafica usate:



- E' blocco funzionale che ha in ingresso
 - ▶ un ingresso A0 di n bit
 - ▶ un ingresso A1 di n bit
 - ▶ 1 bit di selezione S (il selezionatore)
- E in uscita:
 - ▶ un canale B di n bit
- Funzionamento:
 - ▶ Il canale B vale A0 oppure A1, a seconda del valore di S
 - ▶ S «sceglie» fra A0 e A1
 - ▶ Se $S = 0$, esce A0
 - ▶ Se $S = 1$, esce A1

Architettura degli elaboratori

- 35 -

Blocchi funzionali combinatori

35



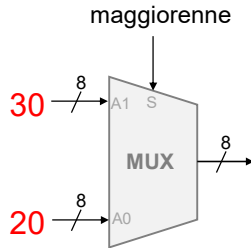
Il Multiplexer a due vie è lo “if-then-else” dell’hardware

“Per i maggiorenni, il prezzo è 30 euro. Per gli altri, è 20 euro”

- Software:

```
funzione prezzo()  
{  
    if (maggiorenne)  
        then return 30  
        else return 20  
}
```

- Hardware:



Architettura degli elaboratori

- 36 -

Blocchi funzionali combinatori

36

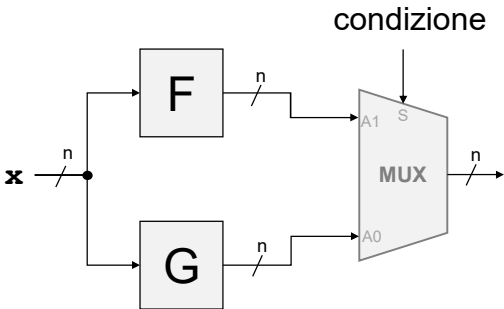


Il Multiplexer a due vie è lo “if-then-else” dell’hardware

- Software

```
if (condizione)  
then return F(x)  
else return G(x)
```

- Hardware



Architettura degli elaboratori

- 37 -

Blocchi funzionali combinatori

37



Il Multiplexer a due vie è lo “if-then-else” dell’hardware

- Osservazioni:
 - ▶ In software, solo uno fra il ramo then e il ramo else viene computato
 - ▶ In hardware, entrambi i casi sono computati, e solo uno dei due risultati è passato in output. L’altro, viene scartato dal Multiplexer.
 - ▶ Questo può sembrare uno spreco, ma va considerato che i due rami sono, almeno, computati *in parallelo*
- Vediamo dalla prossima volta come implementare un multiplexer

Architettura degli elaboratori

- 38 -

Blocchi funzionali combinatori

38



Esercizi e domande di repilogo

- Capire bene e nel dettaglio gli esempi che non abbiamo visto in aula:
 - ▶ l’addizionatore (a tre bit) usato per la somma in Complemento a 2
 - ▶ l’esempio dell’addizionatore a 16 bit progettato usando, come blocchi funzionali, due addizionatori a 8 bit
- Domanda: nell’addizionatore a 8 bit, il passaggio dei riporti costituiscono delle “retroazioni”?
 - ▶ In quello dell’addizionatore a 16 bit, il passaggio del riporto costituisce una “retroazione”?
- Scrivi la tabella di verità di un multiplexer a due vie a 1 bit (cioè, per due segnali di un solo bit bit). Disegna il circuito corrispondente.
 - ▶ [solo per chi ha pazienza] ripetere, per due segnali di 2 bit.

Architettura degli elaboratori

- 39 -

Blocchi funzionali combinatori

39