



# ESERCITAZIONE 4.5

Approfondimento Circuiti Logici e Sequenziali

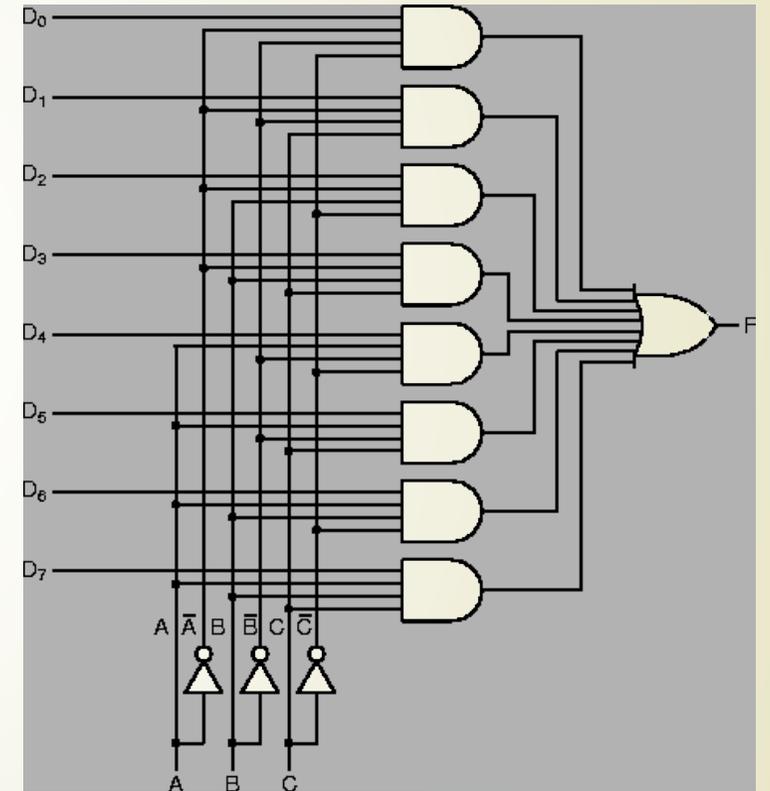


2

# Approfondimento: multiplexer

# Multiplexer: soluzione alternativa

Multiplexer: composizione interna

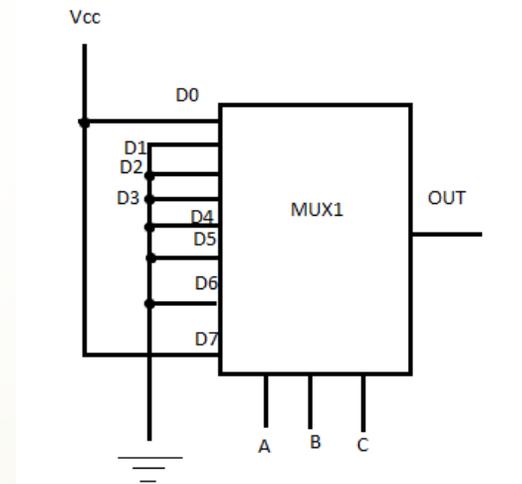


# Multiplexer: soluzione alternativa

Soluzione alternativa al problema già analizzato dell'uso dei multiplexer a 3 input di controllo per espressioni logiche a 4 variabili.

ESEMPIO:  $ABC + \bar{A}\bar{B}\bar{C} + E$

CIRCUITO PER  $ABC + \bar{A}\bar{B}\bar{C}$



# Multiplexer: soluzione alternativa

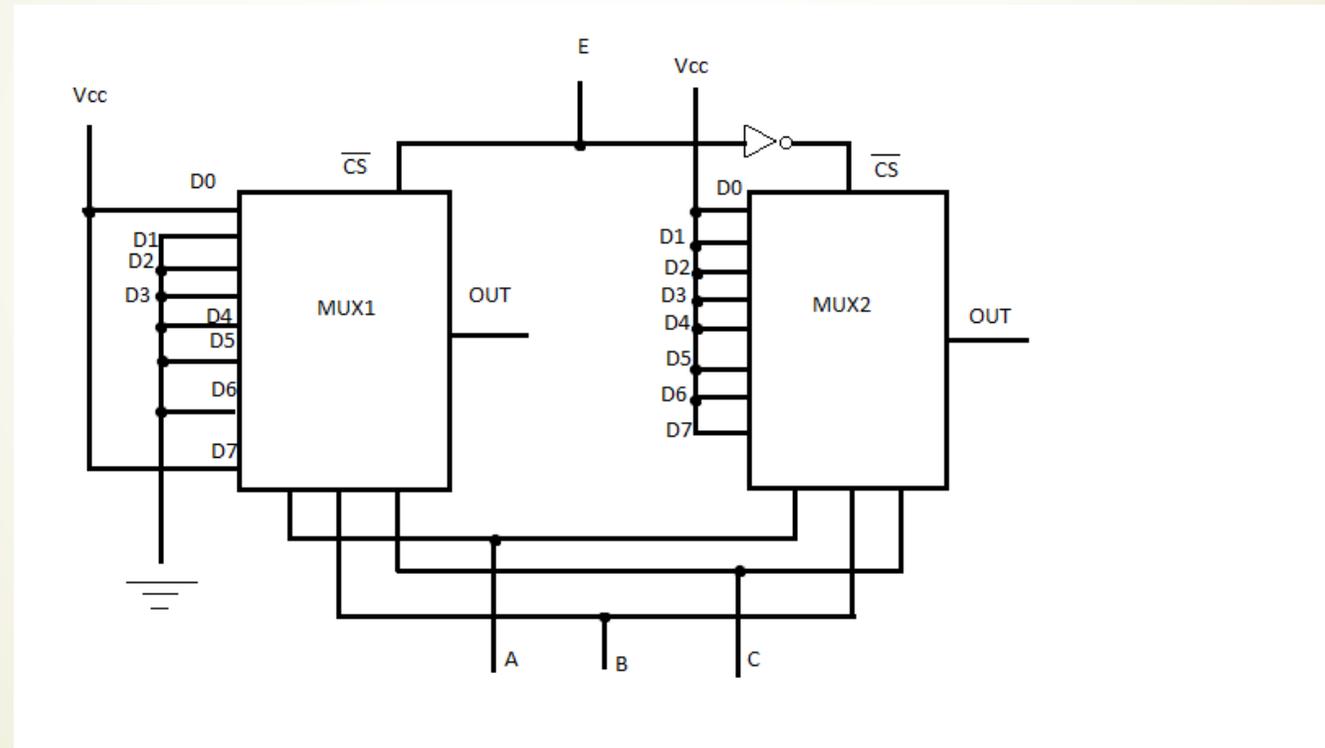
Idea: usare il bit di chip selection (che chiameremo  $\overline{CS}$ ) che ci permette di selezionare il mux o meno.

Utilizzare più mux collegati alla variabile E (o, in alternativa, in forma negata) in modo tale che il risultato dipenda da esso.

Nel caso in analisi: il mux illustrato alla slide precedente dovrà attivarsi quando E è 0, mentre creeremo un secondo mux che restituirà 1 a vuoto che, invece, sarà selezionato nel caso in cui E sia 1.

# Multiplexer: soluzione alternativa

Immagine dei due mux e del loro collegamento a E.



# Multiplexer: soluzione alternativa

Il circuito mostrato in questa seconda figura implementa esattamente quanto detto finora:

- ▶ Se  $E$  ha valore 0, la porta CS del  $\text{mux}_1$  riceve un segnale basso (0), attivando proprio il  $\text{mux}_1$ . Da esso viene calcolato il valore della funzione logica, escludendo la variabile  $E$ . Questo valore viene mandato in output. Allo stesso tempo, il  $\text{mux}_2$  non viene selezionato, (la sua porta logica CS riceve un segnale alto) e quindi non fa nulla.
- ▶ Se  $E$  ha valore 1, la porta CS del  $\text{mux}_2$  riceve un segnale basso (0), che dunque viene attivato. Visto che è un circuito che restituisce sempre 1, farà uscire dal proprio output il valore 1, a prescindere dai valori presenti sui bit di controllo. Contemporaneamente  $\text{mux}_1$  non viene selezionato, e quindi non farà nulla.

# Multiplexer: soluzione alternativa

Applichiamo ora quanto descritto a un esempio più complesso, come l'esercizio 4 dell'esercitazione 3 (disponibile sul sito del corso).

$$Y = \bar{C}D + AB$$

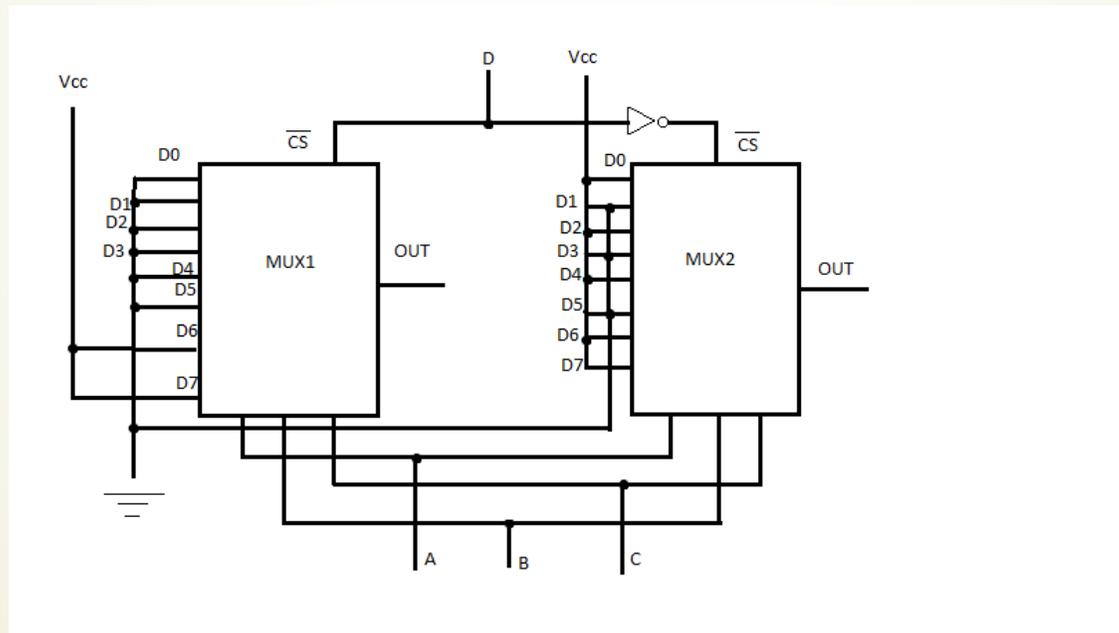
Il mux selezionato da  $D = 0$  dovrà realizzare solo la funzione logica  $AB$ , mentre quello selezionato da  $D = 1$  dovrà realizzare la funzione logica  $AB+C$ .

# Multiplexer: soluzione alternativa

- ▶ Nel mux<sub>1</sub> :
  - ▶ le porte D<sub>6</sub> e D<sub>7</sub> sono collegate alla massa (attivano gli AND del tipo 11x)
  - ▶ tutte le altre porte sono collegate a terra
- ▶ Nel mux<sub>2</sub> :
  - ▶ le porte D<sub>0</sub>, D<sub>2</sub>, D<sub>4</sub>, D<sub>6</sub> sono collegate alla massa (implementano C, attivando tutti gli AND del tipo xx0)
  - ▶ la porta D<sub>7</sub> è collegata alla massa, poiché implementa AB (attiva l'unico ingresso non ancora attivato del tipo 110, cioè 111)
  - ▶ le altre porte sono collegate a terra.

# Multiplexer: soluzione alternativa

Immagine dei due mux e del loro collegamento a D.



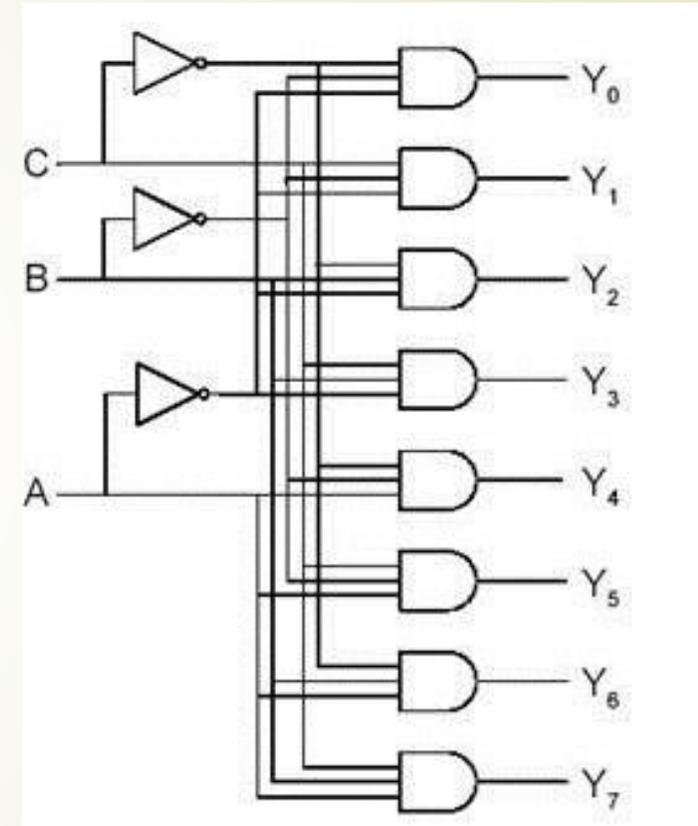
# Approfondimento: Circuiti Logici e sequenziali

11

# Circuiti (1)

## Decoder:

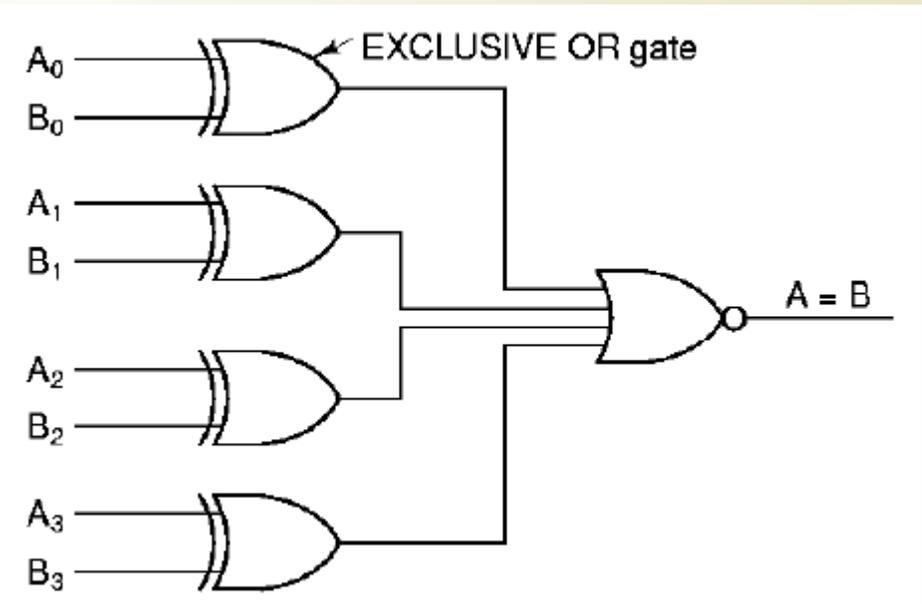
- ▶ Input: numero a  $n$  bit
- ▶ Output: impostare a 1 una sola delle  $2^n$  uscite
- ▶ Esempio uso: Indirizzare piccola memoria a 8 bit



# Circuiti (2)

## Comparator:

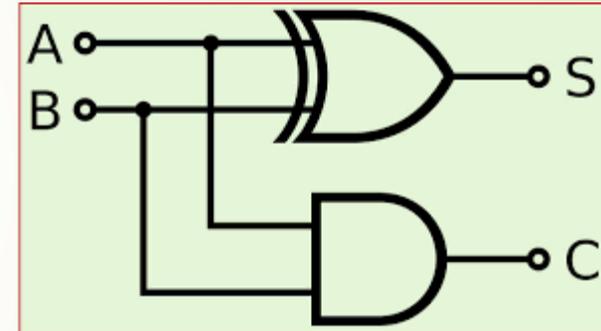
- ▶ Input: 2 numeri a  $n$  bit
- ▶ Output: 0 nel caso in cui gli input siano diversi, uno nel caso in cui siano uguali
- ▶ Esempio uso: calcolare somme di prodotti



# Circuiti (3)

## Adder (1bit):

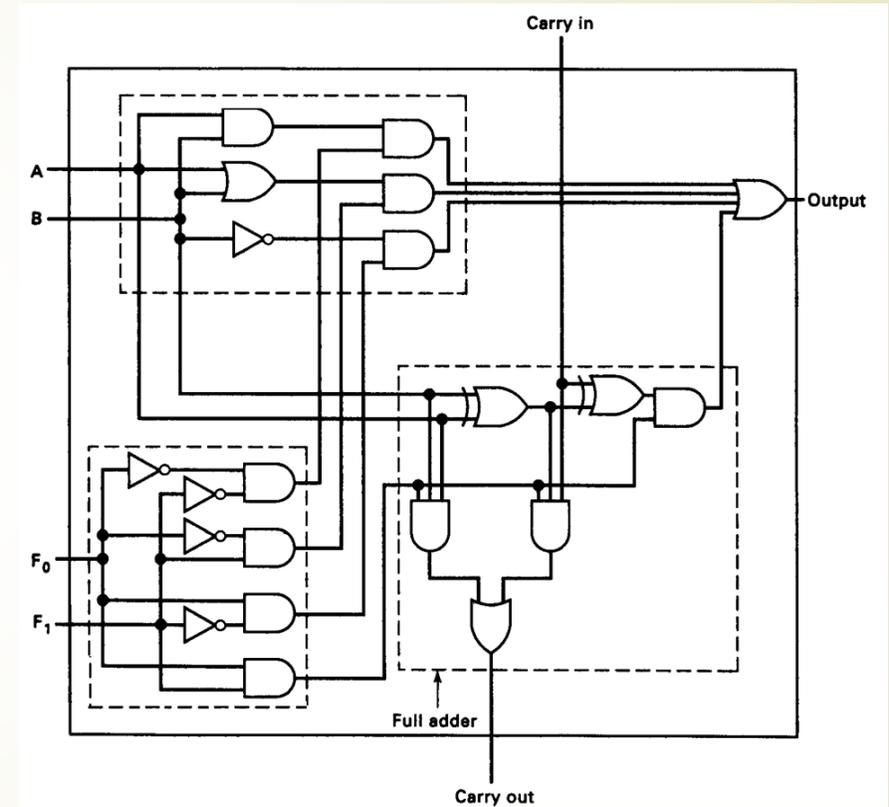
- ▶ Input: 2 numeri a 1 bit
- ▶ Output: Somma dei due numeri all'uscita S, Riporto della somma all'uscita C
- ▶ Esempio uso: Sommare valori
- ▶ Può essere esteso a  $n$  bit ponendone  $n$  in cascata



# Circuiti (4)

## ALU (1bit):

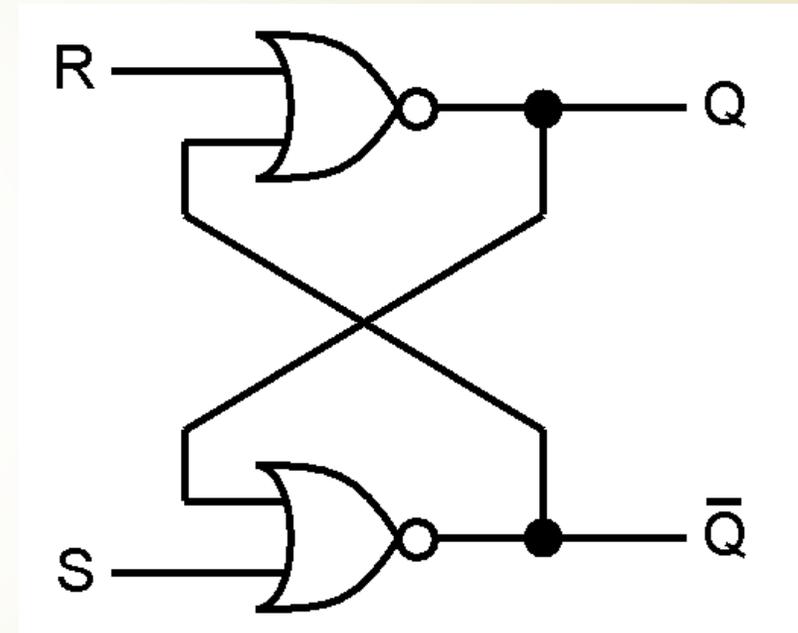
- ▶ Unico circuito per effettuare:
  - ▶ Operazione Logica AND,
  - ▶ Operazione Logica OR,
  - ▶ Somma di due parole.
- ▶ Estendibile a  $n$  bit mettendone  $n$  in cascata



# Circuiti (5)

## Latch SR:

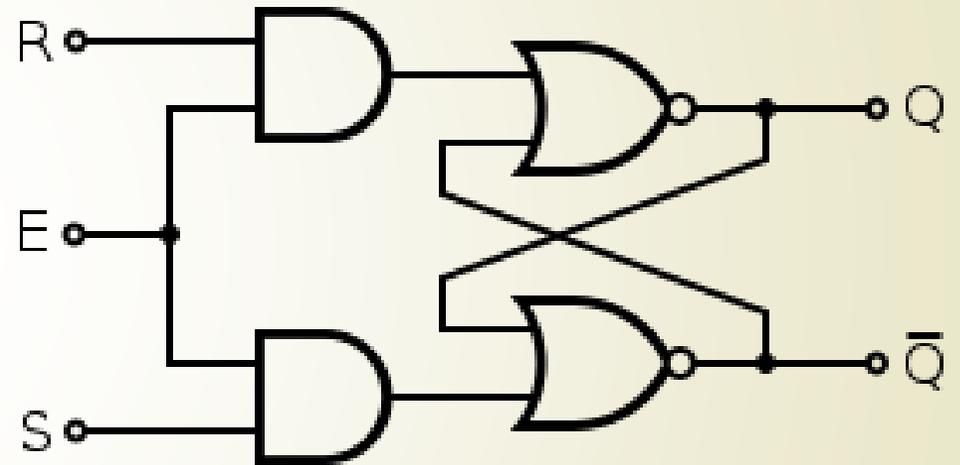
- ▶ Memoria a 1 bit
- ▶ Input:
  - ▶ **S** (*Setting*): imposta il valore in memoria,
  - ▶ **R** (*Resetting*): azzera il valore del latch.
- ▶ Output: Q e  $\bar{Q}$ , complementari
- ▶ Output non determinate unicamente da valori input corrente.



# Circuiti (6)

## Latch SR Temporizzato:

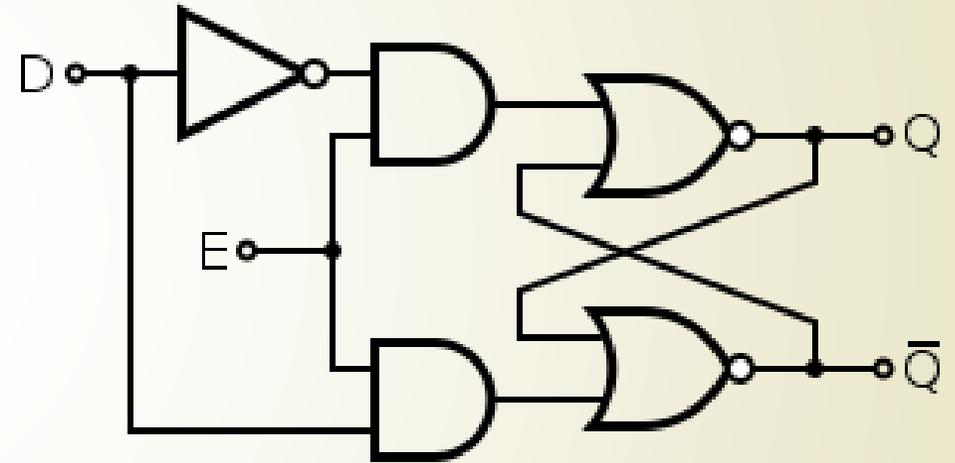
- ▶ Come latch SR, ma cambio stato solo in specifici momenti
- ▶ Cambio stato solo quando clock (E) vale 1



# Circuiti (7)

## Latch D Temporizzato:

- ▶ Come latch SR, ma con un unico input per evitare ambiguità ( $S=R=1$ )
- ▶ Valore D viene campionato quando clock vale 1



# Circuiti (8)

## Flip-flop:

- ▶ Campionamento valore linea e memorizzazione
- ▶ Campionamento durante transizione clock da 0 a 1.

