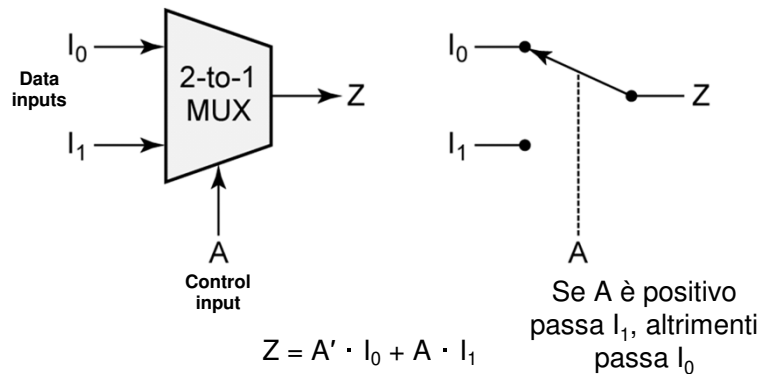


Multiplexer Demultiplexer

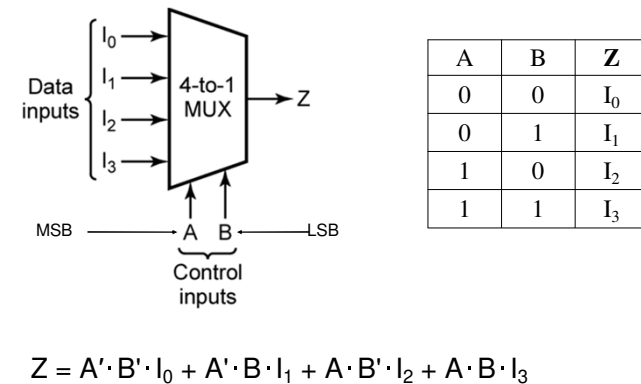
Multiplexer

- Un multiplexer **seleziona un dato** tra possibili input e produce tale dato in output
- E' caratterizzato da
 - **N** input di controllo
 - **2^N** input dati
 - **1** output

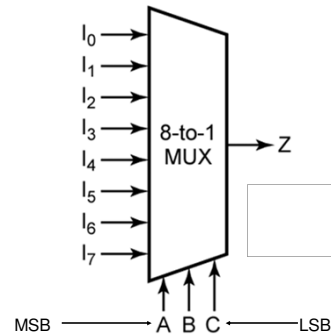
Multiplexer (2 ingressi)



Multiplexer (4 ingressi)



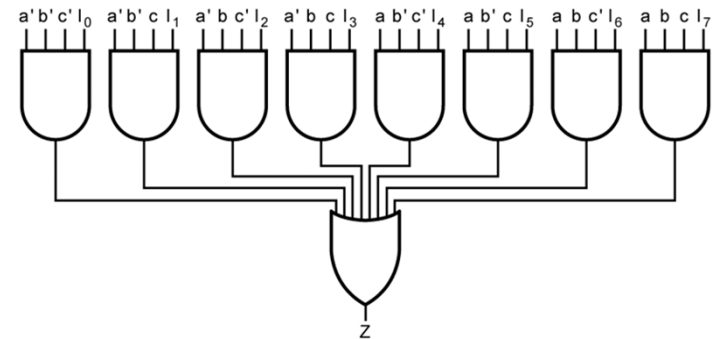
Multiplexer (8 ingressi)



A	B	C	Z
0	0	0	I ₀
0	0	1	I ₁
0	1	0	I ₂
0	1	1	I ₃
1	0	0	I ₄
1	0	1	I ₅
1	1	0	I ₆
1	1	1	I ₇

$$Z = A' \cdot B' \cdot C' \cdot I_0 + A' \cdot B' \cdot C \cdot I_1 + A' \cdot B \cdot C' \cdot I_2 + A' \cdot B \cdot C \cdot I_3 + A \cdot B' \cdot C' \cdot I_4 + A \cdot B' \cdot C \cdot I_5 + A \cdot B \cdot C' \cdot I_6 + A \cdot B \cdot C \cdot I_7$$

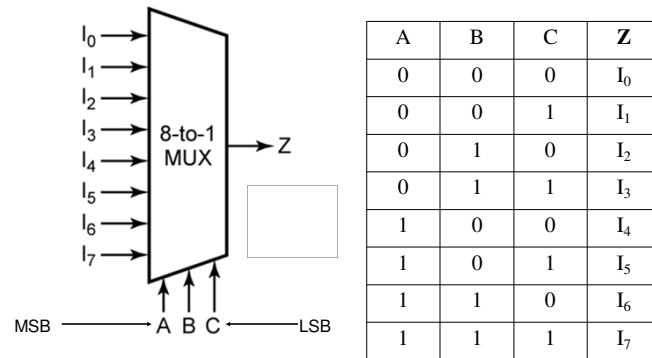
Implementazione



Progettazione di circuiti logici
mediante multiplexer

Analizziamo meglio
l'espressione di uscita di un
multiplexer....

Multiplexer (8 ingressi)



$$Z = A' \cdot B' \cdot C' \cdot I_0 + A' \cdot B' \cdot C \cdot I_1 + A' \cdot B \cdot C' \cdot I_2 + A' \cdot B \cdot C \cdot I_3 + A \cdot B' \cdot C' \cdot I_4 + A \cdot B' \cdot C \cdot I_5 + A \cdot B \cdot C' \cdot I_6 + A \cdot B \cdot C \cdot I_7$$

Uso di n-input Multiplexer

- E' possibile utilizzare un **multiplexer a n ingressi dati per realizzare una funzione logica con n mintermini**
- Il numero di variabili della funzione (literal, corrispondenti agli input di controllo) è $m = \log_2 n$
- **Ciascun mintermine può essere mappato a un ingresso del multiplexer**

Come procedere?

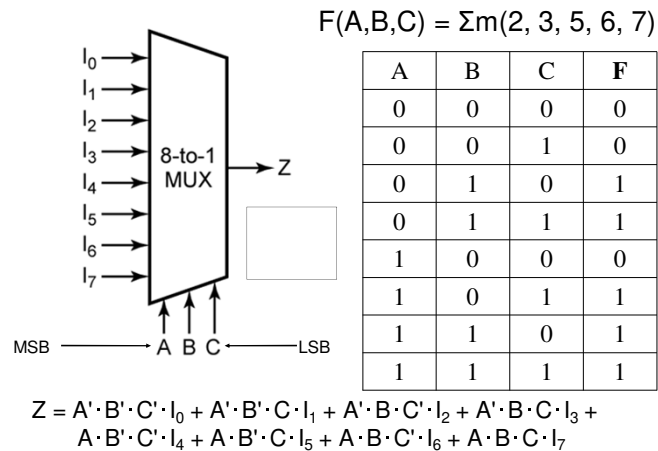
- Per ciascuna riga della tabella di verità in cui la funzione vale 1, settare a 1 il corrispondente ingresso del multiplexer
- Ciò significa che il mintermine corrispondente a tale ingresso è abilitato nelle uscite del multiplexer
- Settare a 0 i rimanenti ingressi del multiplexer

Esempio

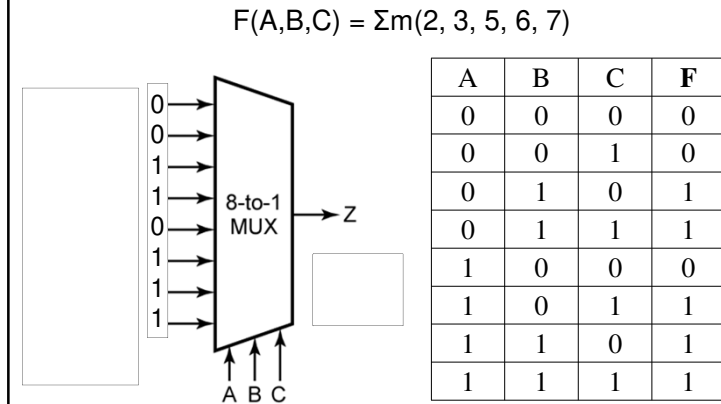
Realizziamo la seguente funzione Booleana utilizzando un multiplexer 8-to-1:

$$F(A,B,C) = \Sigma m(2, 3, 5, 6, 7)$$

Esempio (Soluzione)



Esempio (Soluzione)

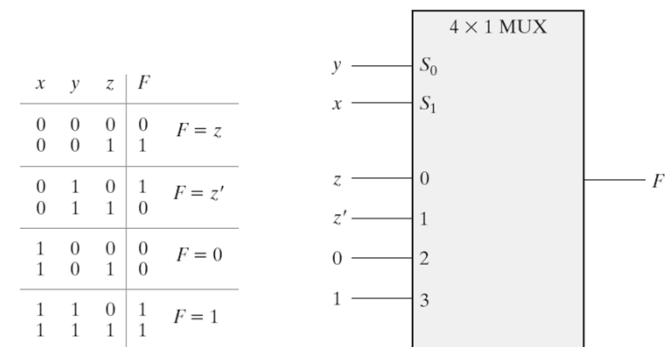


Progettazione funzioni mediante un multiplexer a n/2 ingressi

- Raggruppare le righe della tabella di verità in n/2 coppie
- Ciascuna coppia di righe corrisponde al **termine di prodotto di (m-1) variabili**
- Ciascuna **coppia di righe può essere mappata su un ingresso del multiplexer**
- Determinare la funzione logica di ciascuna coppia di righe in termini della **m-esima variabile**
- Ad esempio se la m-esima variabile è **x**, i **valori possibili sono x, x', 0, 1**

Esempio

$$F(x,y,z) = \Sigma m(1, 2, 6, 7)$$

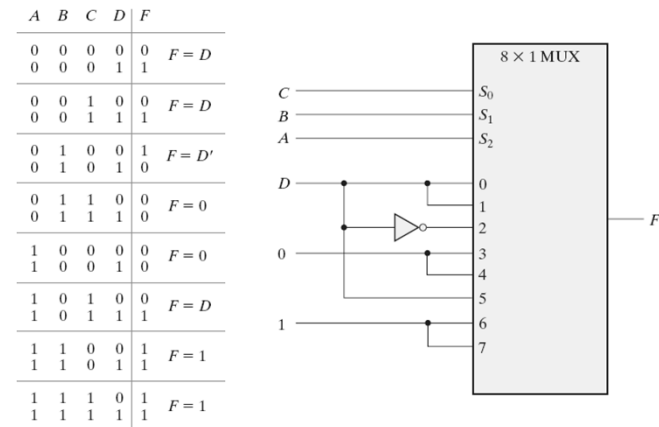


Esempio

$$F(A,B,C,D) = \Sigma m(1,3,4,11,12-15)$$

Esempio

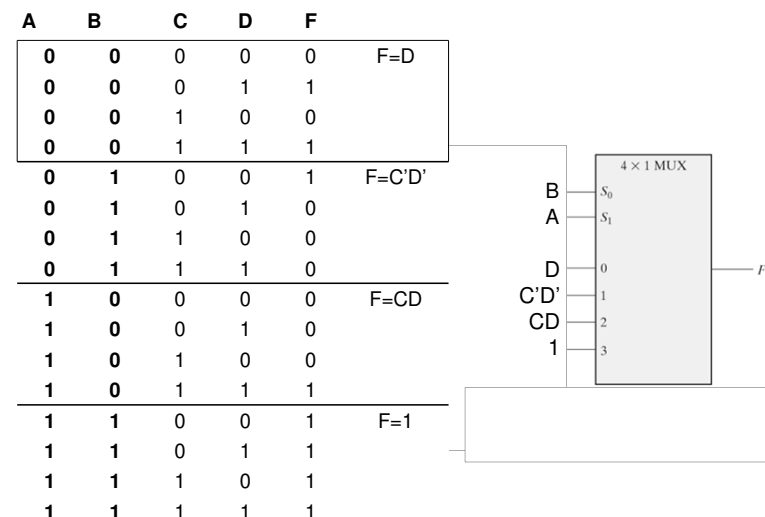
$$F(A,B,C,D) = \Sigma m(1,3,4,11,12-15)$$



Funzione logica usando un n/4 multiplexer

- Analogamente a quanto fatto prima, possiamo utilizzare un **multiplexer a n/4 ingressi per realizzare una funzione di m-2 variabili**
- In questo caso:
 - creiamo **gruppi da 4 righe nella tabella di verità**
 - utilizziamo le **ultime 2 variabili per descrivere la funzione**

Funzione mediante n/4 multiplexer

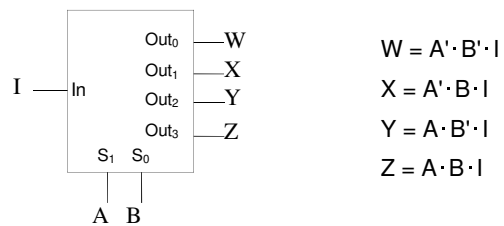


Demultiplexer

Demultiplexer

- Funzione opposta rispetto al multiplexer
- Effettua l'istridamento (routing) di un ingresso VERSO uno dei possibili output
- Un demultiplexer è caratterizzato da
 - **N** input di controllo
 - **1** ingresso dati
 - **2^N** output

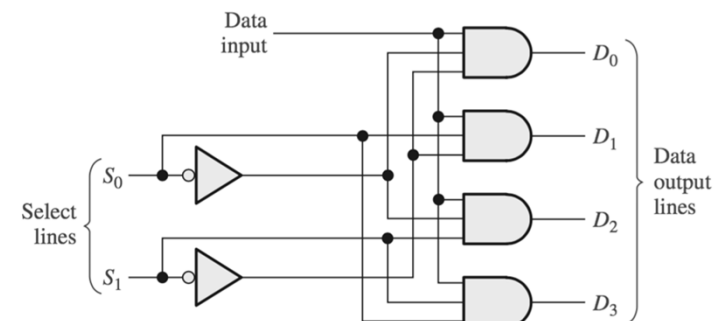
Demultiplexer



$$\begin{aligned}
 W &= A' \cdot B' \cdot I \\
 X &= A' \cdot B \cdot I \\
 Y &= A \cdot B' \cdot I \\
 Z &= A \cdot B \cdot I
 \end{aligned}$$

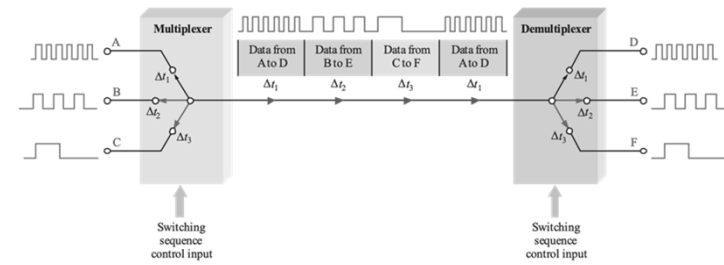
A	B	W	X	Y	Z
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Implementazione



Applicazione....

Selettore di dati

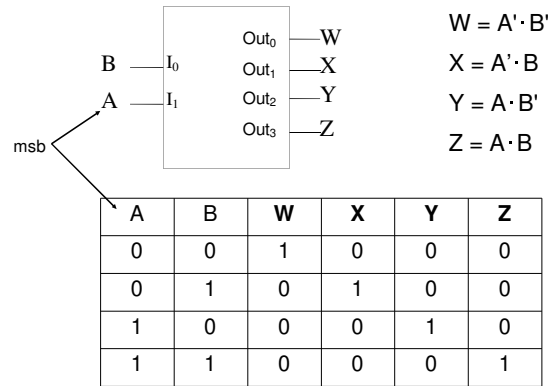


Decoder

Decoder

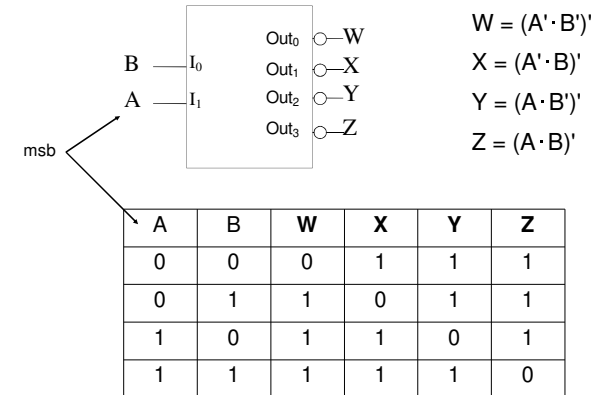
- Un decoder ha
 - **N** ingressi
 - **2^N** uscite
- Produce un valore **1 su uno dei 2^N output decodificando il valore binario dei bit d'ingresso**
- Esempio: se l'ingresso è 100 il bit di uscita alto sarà il quarto bit (100 in binario è 4)

Decoder



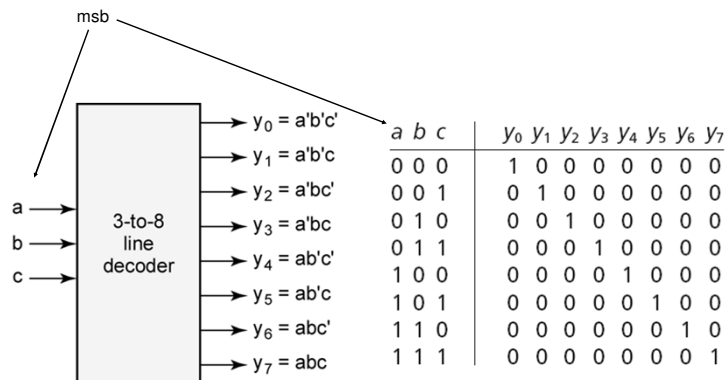
Gli ingressi pilotano valori **alti** delle uscite W,X,Y,Z

Decoder con uscite negate

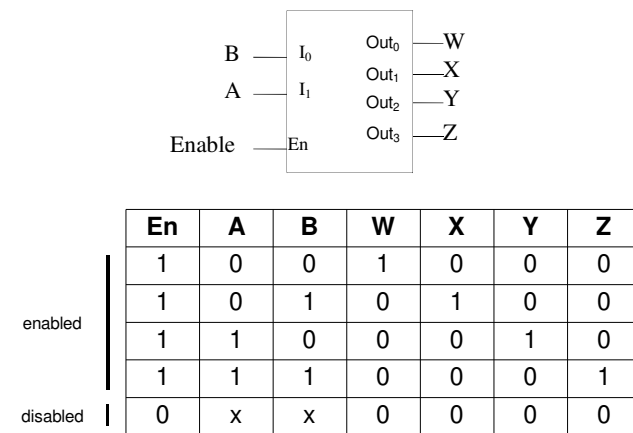


Gli ingressi pilotano valori **bassi** delle uscite W,X,Y,Z

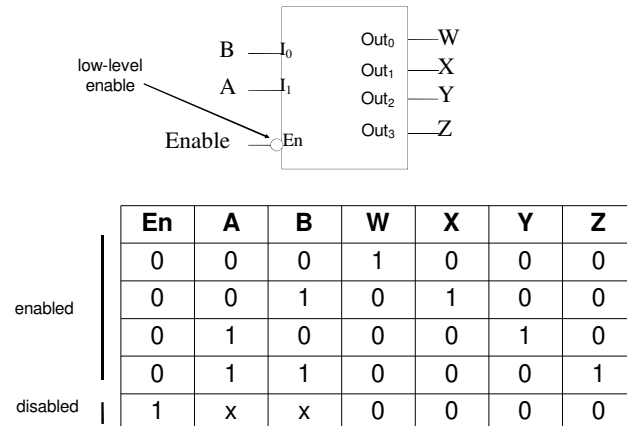
Decoder a 8 uscite



Decoder con bit Enable

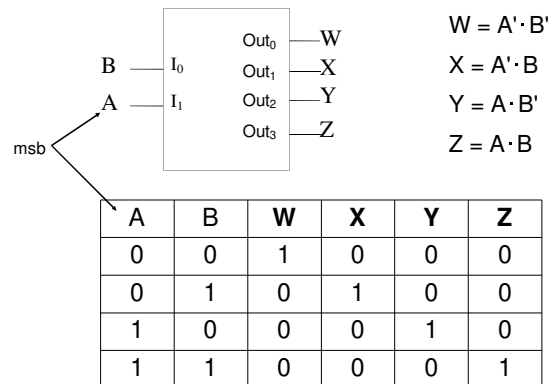


Decoder con bit Enable basso



Progettazione di circuiti logici mediante decoder

Torniamo al decoder...



Notiamo che...

Le 2^N uscite del decoder corrispondono ai **mintermini** delle **N** variabili in ingresso

Come procediamo?

- Vediamo come utilizzare un decoder a n uscite per realizzare una funzione con n mintermini
- Ciascun mintermine può essere mappato a un output del decoder
- Per ciascuna riga della tabella di verità in cui l'output è 1, sommare (ovvero porre in OR) gli output corrispondenti del decoder
 - Ovvero, ciascun output corrisponde a un mintermine, e il loro OR è l'espansione in mintermini della funzione
- I restanti output restano non connessi

Esempio

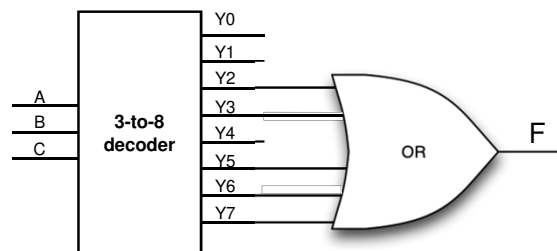
Realizzare la funzione

$$F(A,B,C) = \Sigma m(2, 3, 5, 6, 7)$$

utilizzando un 3-to-8 decoder

Soluzione

$$F(A,B,C) = \Sigma m(2, 3, 5, 6, 7)$$



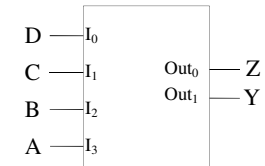
Encoder

Encoder

- Un encoder ha:
 - 2^N ingressi
 - N uscite
- Produce in **output il valore binario corrispondente all'input attivo**
- Quindi ha una **funzione duale rispetto al decoder**

Encoder

Un solo ingresso alla volta può essere posto a 1



A	B	C	D	Y	Z
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Buffer a tre stati (tri-state buffers)

Definizione

- Un tristate buffer può produrre 3 valori
 - 1 (HIGH)
 - 0 (LOW)
 - **Z, Alta impedenza, ovvero circuito aperto**

Se il controllo è a 0, si ha alta impedenza

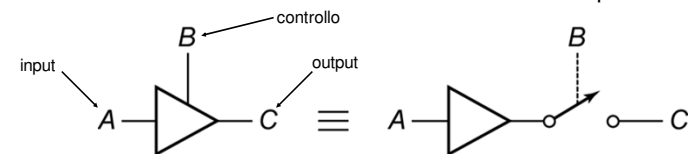
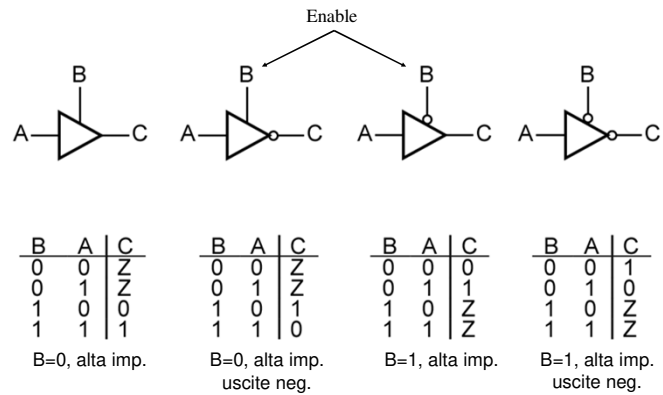
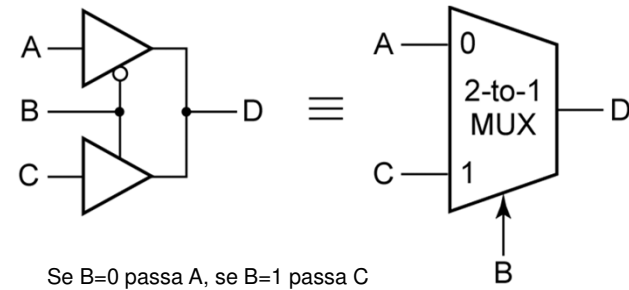


Tabella di verità



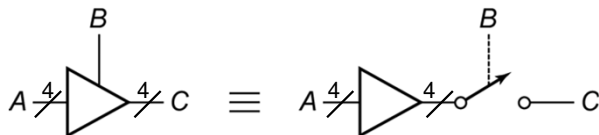
Applicazioni

Multiplexer realizzato con un tri-state buffer



Tri-state buffer a più bit

Simile al multiplexer con bus di controllo



Esempio

Adder con uno degli addendi selezionabile tra 4 ingressi (A,B,C,D)

