



**UNIMORE**  
UNIVERSITÀ DEGLI STUDI DI  
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,  
Informatiche e Matematiche

# Esercitazione 1: Introduzione a Logisim

## Architettura dei calcolatori [MN1-1143]

*Corso di Laurea in Ingegneria Informatica*  
(D.M.270/04) [16-215]  
Anno accademico 2020/2021

**Dott. Gianluca Brilli**  
[gianluca.brilli@unimore.it](mailto:gianluca.brilli@unimore.it)  
**Prof. Marko Bertogna**  
[marko.bertogna@unimore.it](mailto:marko.bertogna@unimore.it)

*È vietata la copia e la riproduzione dei contenuti e immagini in qualsiasi forma.*

*È inoltre vietata la redistribuzione e la pubblicazione dei contenuti e immagini non autorizzata espressamente dall'autore o dall'Università di Modena e Reggio Emilia.*

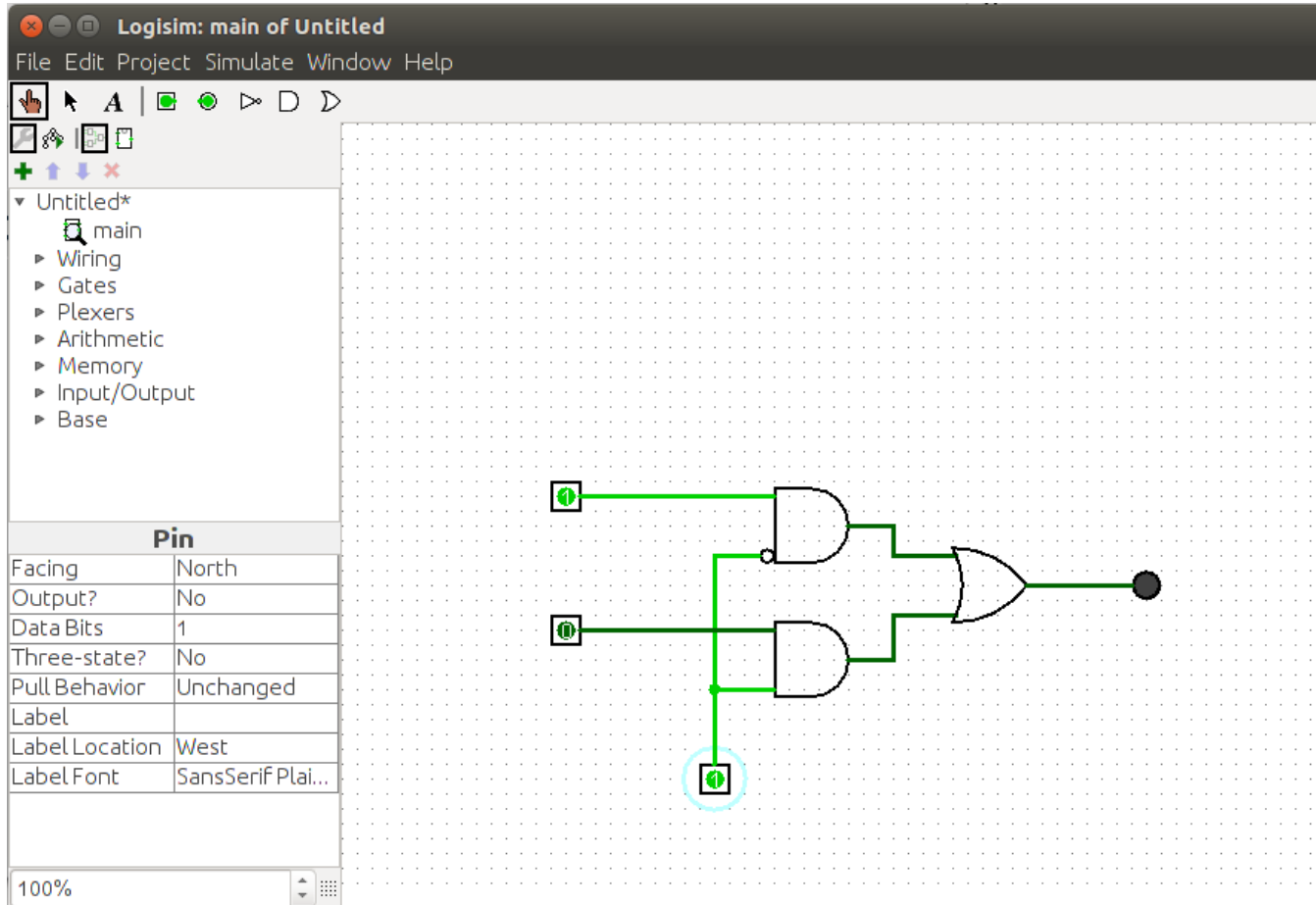
# Ambiente di Lavoro (1)

› **Logisim**: Simulatore di Circuiti Logici.

› disponibile il pacchetto jar da <https://sourceforge.net/projects/circuit/> , ma anche il pacchetto deb nelle maggiori distribuzioni GNU/Linux derivate da Debian.

› *\$ sudo apt-get install logisim*

# Ambiente di Lavoro (2)



# Porte Elementari (1)

1. Cliccare sulla porta AND. Un'ombra di una porta AND ora segue il cursore. Cliccare sulla basetta del circuito principale per posarlo.
2. Cliccare il pulsante del piedino di ingresso ("Input Pin") e piazzarne due da qualche parte a sinistra della porta AND.
3. Cliccare il pulsante del piedino di uscita ("Output Pin") e posizionarne uno da qualche parte a destra della porta AND.

# Porte Elementari (2)

1. Cliccare ora il pulsante di selezione "Select tool" e connettere i due piedini di input con la porta AND. Per posizionare un filo, basta cliccare e trascinare dal punto sorgente a quello destinazione.

2. È possibile ridurre il numero di ingressi della porta cliccandola con lo strumento di selezione e modificandone le proprietà nel pannello situato in basso a sinistra della finestra. Tale impostazione può essere applicata anche prima che il componente sia stato posato.

# Porte Elementari (3)

1. Infine, cliccare lo strumento di manipolazione ("Poke") e smanettare con i piedini di ingresso osservando gli effetti. Controllare che il comportamento della porta corrisponda con le aspettative.

# Creazione di Sottocircuiti (1)

- › Creazione di reti logiche in maniera “modulare”.
- › Creare un nuovo schema: Menu -> File -> New.
- › Creare un nuovo sotto-circuito: Menu -> Project -> Add Circuit. Lo si chiami NAND.
- › creare un circuito NAND con due piedini di ingresso a sinistra e uno di uscita a destra. Per farlo, usare solo porte AND, OR o NOT



# Creazione di Sottocircuiti (2)

› Si ritorni al circuito principale doppio-cliccando "main" nel prospetto sinistro dei circuiti. Si clicchi ora (una sola volta) l'elemento NAND posizionarlo nel circuito principale. Lo si collaudi aggiungendovi piedini di ingresso e di uscita.

# Creazione di Sottocircuiti (3) - Soluzione

The screenshot displays a circuit simulation software interface. The main workspace shows a circuit diagram on a grid background. The circuit consists of two input pins labeled 'A' and 'B', connected to a NAND gate. The output of the NAND gate is connected to an inverter, which is then connected to an output pin labeled 'OUT'. The software interface includes a menu bar (File, Edit, Project, Simulate, Window, Help), a toolbar with various icons, and a component library on the left. The component library is expanded to show a list of components under the 'Wiring' category, including Splitter, Pin, Probe, Tunnel, Pull Resistor, Clock, Constant, Power, Ground, Transistor, and Transmission Gate. Below the component library, a table displays the properties of the selected component, a NAND gate.

Circuit: NAND	
Circuit Name	NAND
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12

# Esercizio 01

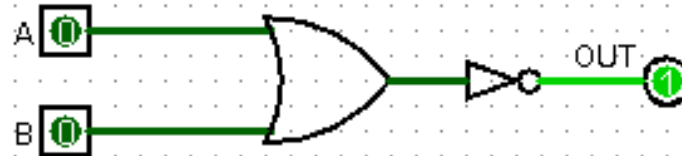
›Costruiamo e testiamo in logisim un sottocircuito per:

›NOR

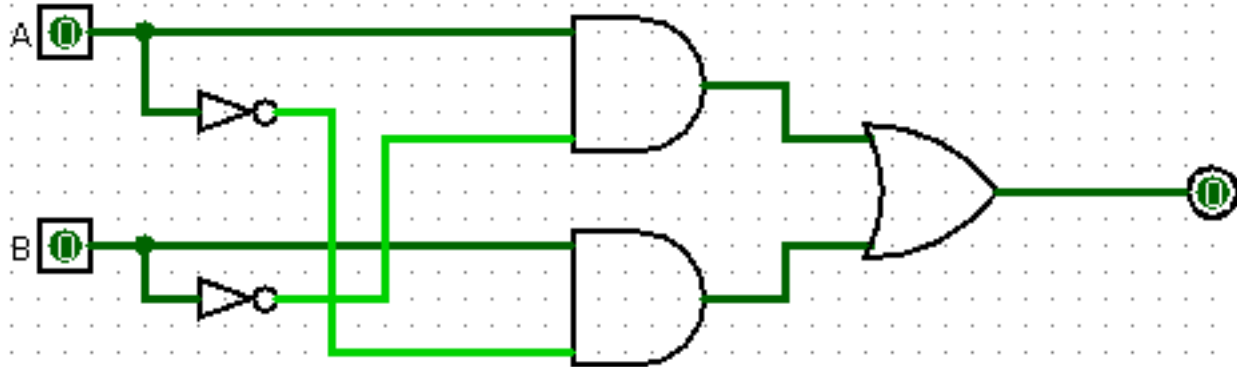
›XOR

# Esercizio 01 - Soluzione

>NOR

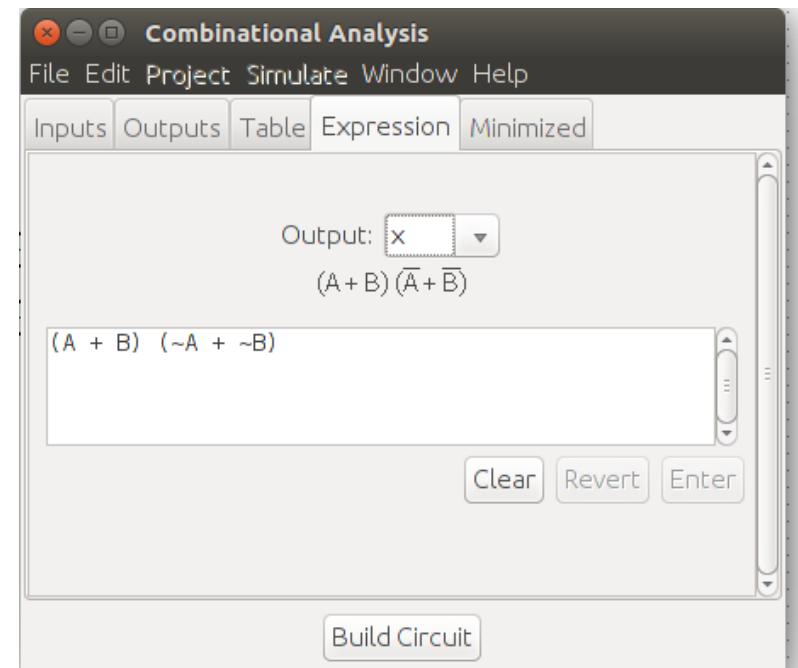
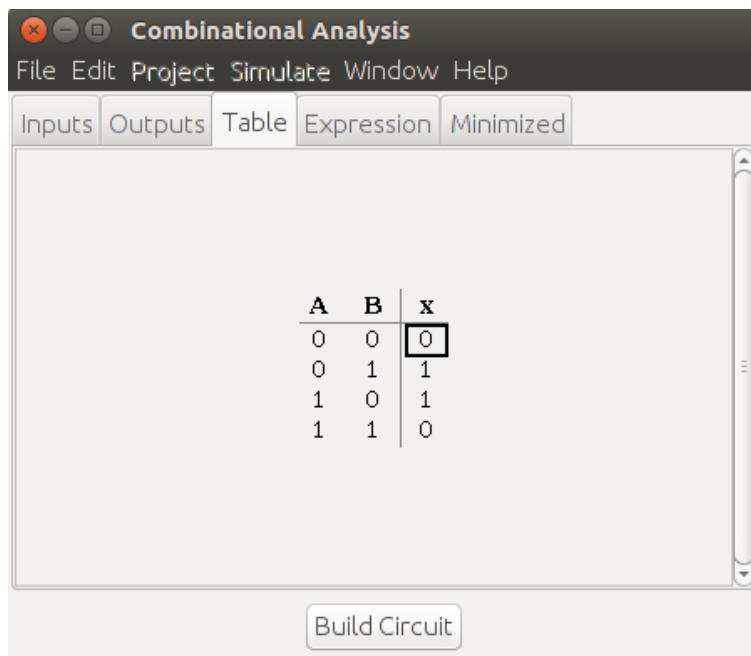


>XOR



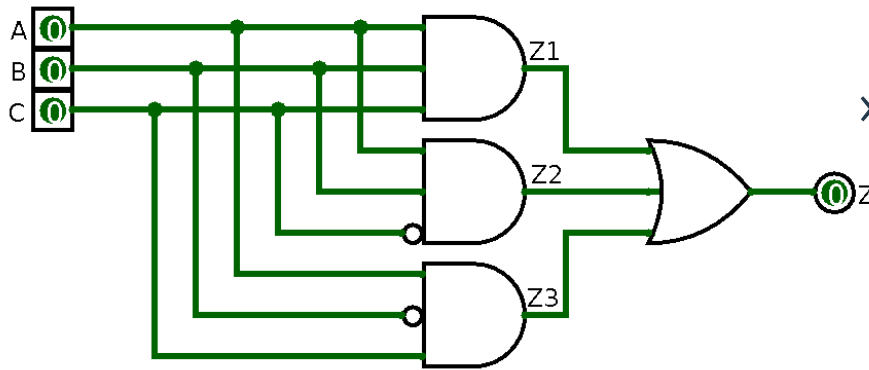
# Tabelle di Verità e Funzioni Logiche

› Nel prospetto dei circuiti cliccare destro sul sotto-circuito NAND ed analizzare il circuito selezionando "Analyze circuit".

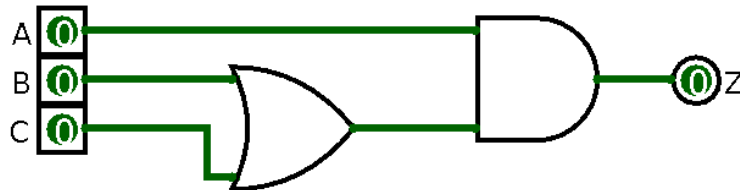


# Esercizio per Casa

› Testiamo in Logisim il seguente circuito:



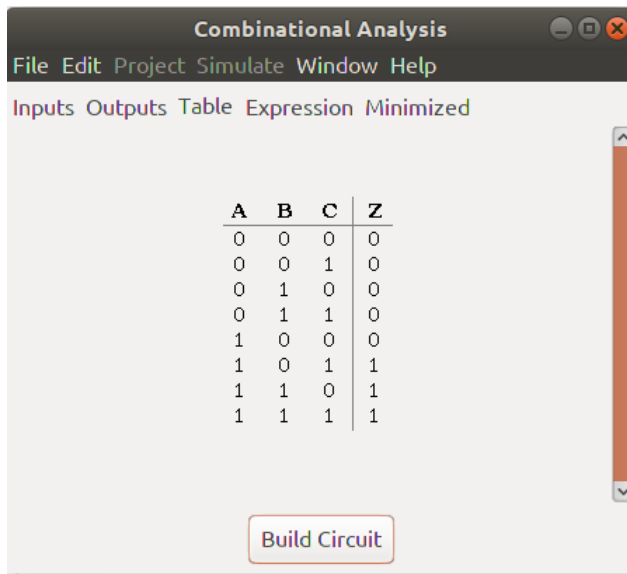
› Versione non minima



› Versione minimizzata applicando alcuni teoremi.

# Esercizio - Soluzione

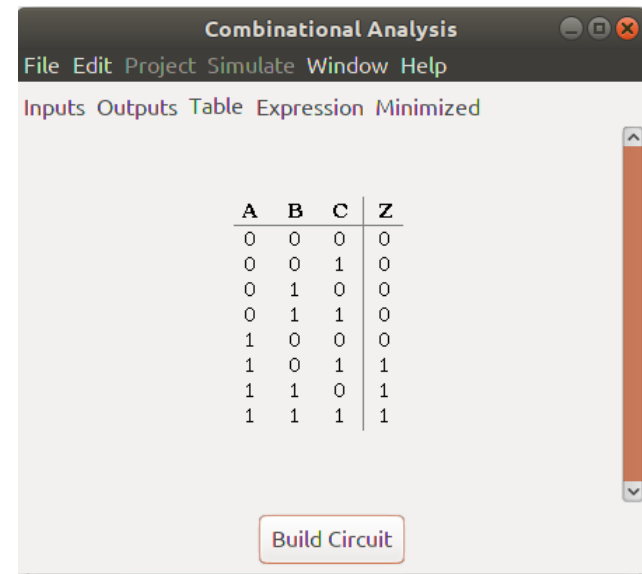
›Confrontiamo le tabelle di verità, per mezzo del tool di generazione automatica di Logisim:



The screenshot shows the 'Combinational Analysis' window in Logisim. The window title is 'Combinational Analysis' and it has a menu bar with 'File', 'Edit', 'Project', 'Simulate', 'Window', and 'Help'. Below the menu bar, there are tabs for 'Inputs', 'Outputs', 'Table', 'Expression', and 'Minimized'. The 'Table' tab is selected, displaying a truth table with columns A, B, C, and Z. The table contains 8 rows of data. At the bottom of the window, there is a 'Build Circuit' button.

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

›Circuito di partenza



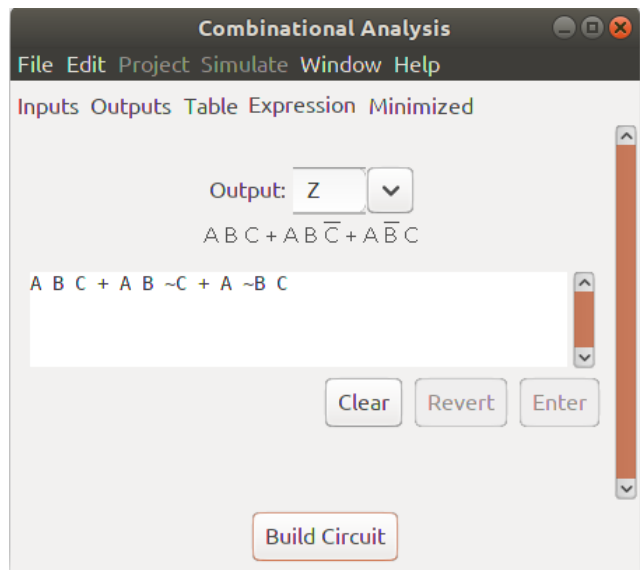
The screenshot shows the 'Combinational Analysis' window in Logisim, identical to the previous one. The 'Table' tab is selected, displaying a truth table with columns A, B, C, and Z. The table contains 8 rows of data. At the bottom of the window, there is a 'Build Circuit' button.

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

›Versione minimizzata

# Esercizio - Soluzione

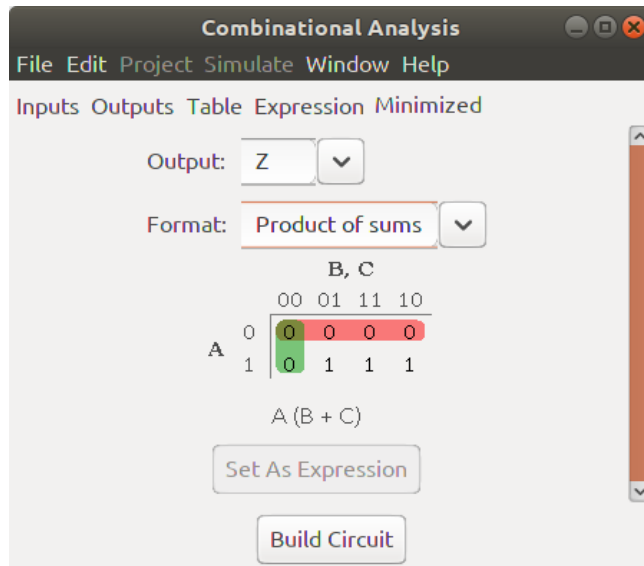
›Relativamente al primo circuito ( la versione non minimizzata ), aprendo la voce Analyze Circuit notiamo la sua espressione logica:





# Esercizio - Soluzione

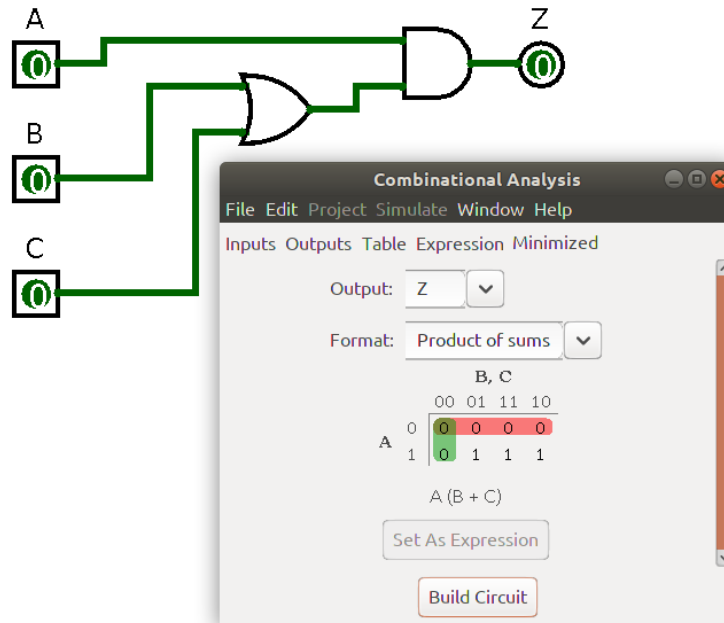
›Cliccando in basso su “Minimized” possiamo automaticamente ottenere la versione minimizzata che ci siamo calcolati tramite i teoremi dell’algebra di Boole.



$$\text{›}Z = A ( B + C )$$

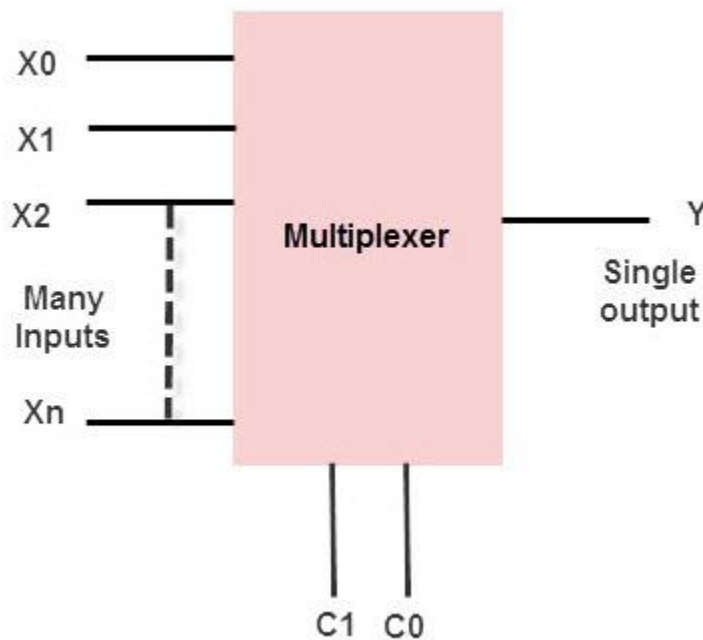
# Esercizio - Soluzione

›Cliccando infine su “Build Circuit” possiamo generare il circuito logico della versione minimizzata in automatico.



# Il Multiplexer

› Progettiamo un componente digitale che funga da selettore: il **Multiplexer**.



›  $n$  ingressi denotati con  $\{X_0, X_1, \dots, X_{n-1}\}$ ;

›  $s = \log_2(n)$  selettori, denotati con  $\{c_0, c_1, \dots, c_{s-1}\}$ ;

› un singolo ingresso, denotato con  $X$ .

# Esercizio 03 – Parte A

›Realizzare i seguenti due circuiti:

›MUX a **2 ingressi** di ampiezza **1 bit**;

›MUX a **4 ingressi** di ampiezza **1 bit**.

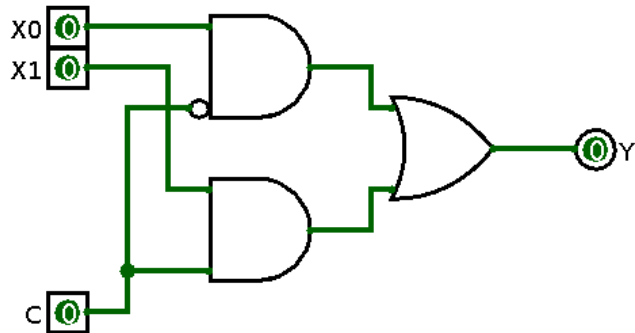
# Esercizio 03 – Parte B

›Sfruttando il MUX a 2 ingressi di ampiezza 1 bit, realizzato in precedenza, progettare il seguente circuito:

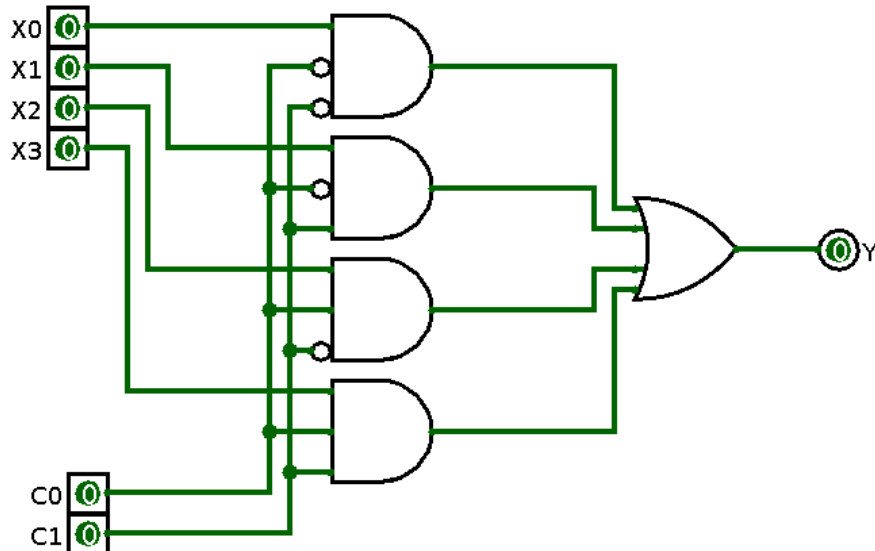
›MUX a **2 ingressi** di ampiezza **2 bit**;

›Utilizzare il componente **splitter** di Logisim e pin di ingresso/uscita a 2 bit.

# Esercizio 03 – Soluzione



> Multiplexer a 2 ingressi



> Multiplexer a 4 ingressi

# Esercizio 04

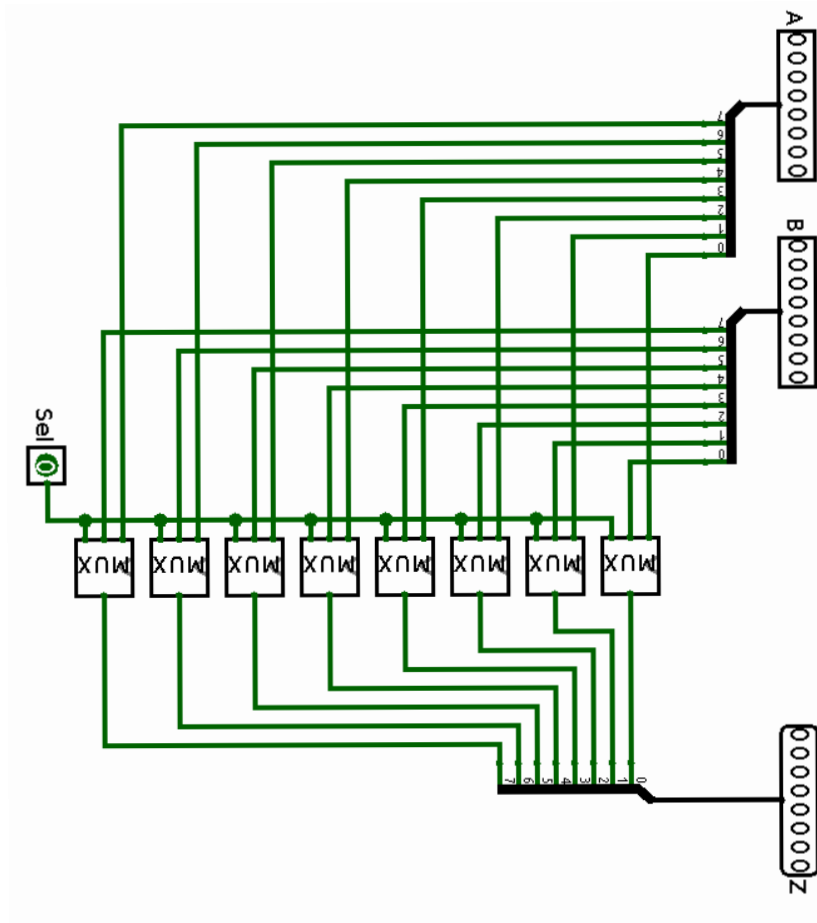
›Tramite la logica che abbiamo imparato all'esercizio precedente, realizzare i seguenti due circuiti:

›MUX a **2 ingressi** di ampiezza **8 bit**;

›MUX a **4 ingressi** di ampiezza **8 bit**;

›Salvarli infine in due sottocircuiti distinti, in quanto li utilizzeremo in seguito.

# Esercizio 04 - Soluzione



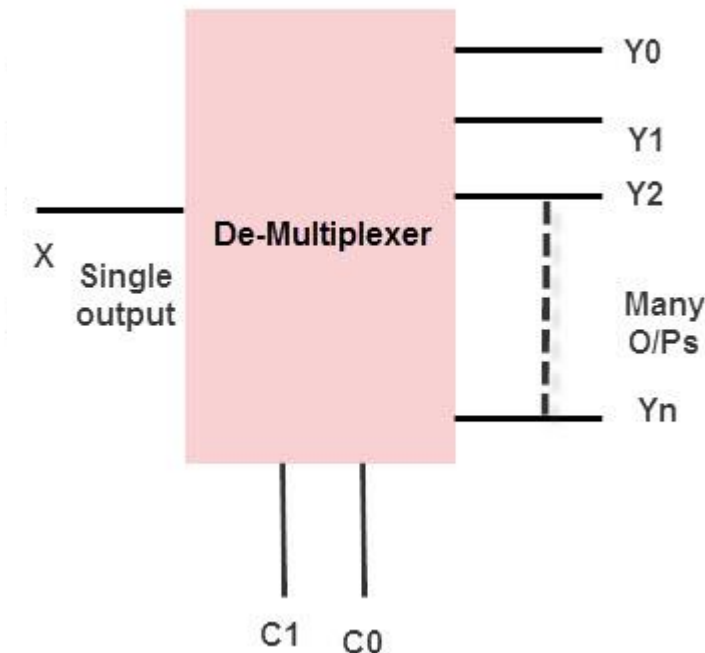
>Versione 2 ingressi a 8 bit;

>In questo caso i blocchi sono 8 MUX a 2 ingressi a 1 bit.



# Il Demultiplexer

› Progettiamo un componente digitale che faccia l'operazione inversa del MUX: il **Demultiplexer**.

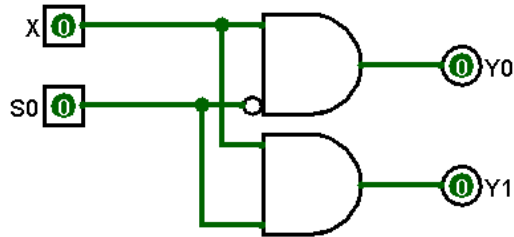


›  $n$  uscite denotate con  $\{Y_0, Y_1, \dots, Y_{n-1}\}$ ;

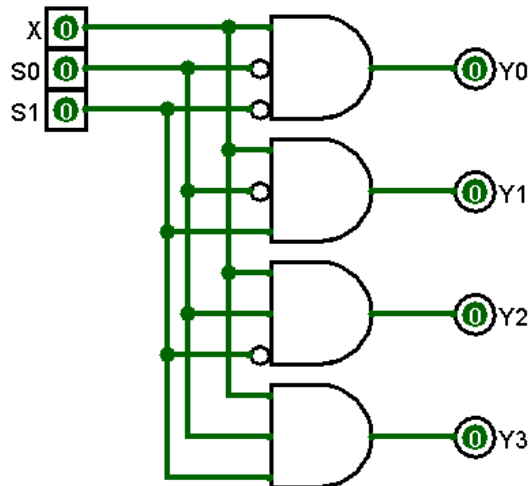
›  $s = \log_2(n)$  selettori, denotati con  $\{c_0, c_1, \dots, c_{s-1}\}$ ;

› un singolo ingresso, denotata con  $X$ .

# Esercizio 05 – Soluzione



›Demultiplexer a 2 uscite



›Demultiplexer a 4 uscite