laboratorio di architettura degli elaboratori

circuiti logici combinatori

Daniele Radicioni

di.unito.it
Dipartimento di Informatica

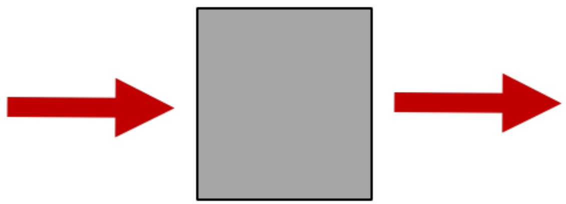
obiettivi della lezione di oggi

- Imparare a usare il simulatore online CircuitVerse
- Sperimentare i circuiti logici elementari in modo strutturato
- Risolvere esercizi simili a quelli d'esame

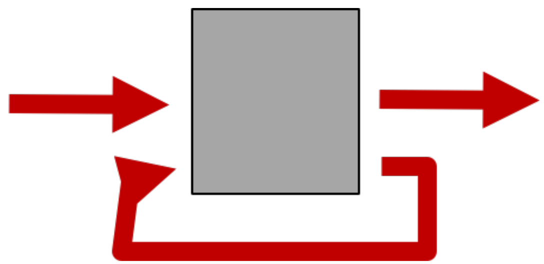
circuiti combinatori e sequenziali

- circuiti digitali utilizzano 2 valori
  - 0 (segnale compreso tra 0 e 1 volt), e
  - 1 (segnale tra 2 e 5 volt)

- i circuiti sono detti combinatori quando l'output è funzione esclusivamente dell'input; sono detti sequenziali quando l'output è funzione —oltre che dell'input— anche di uno stato.



combinatorio

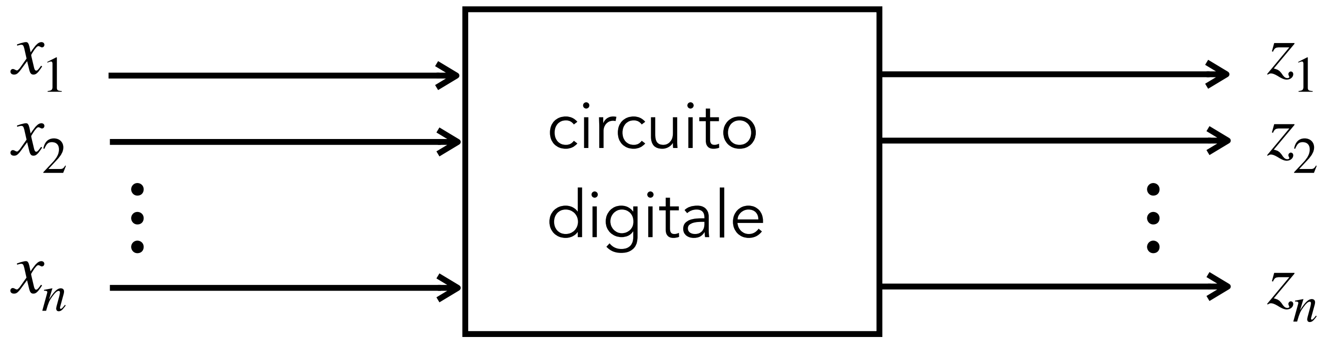


sequenziale

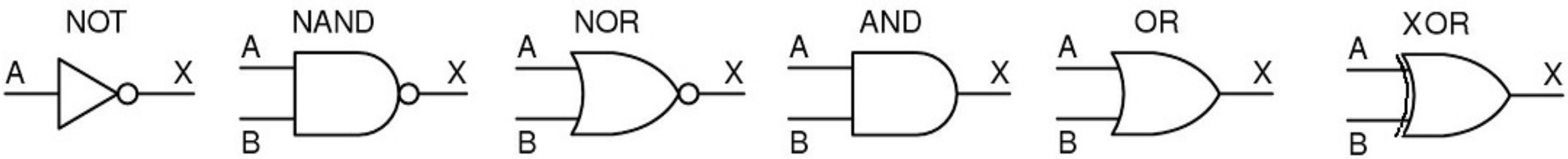
circuito combinatorio

- output è funzione esclusivamente dell'input, quindi ogni output z_i è una funzione booleana degli ingressi $x_1, x_2, \ldots, x_n$.

$$z_i = f_i\,(x_1, x_2, \ldots, x_n)$$



porte logiche



A	X
0	1
1	0

(a)

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

(b)

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

(c)

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

(d)

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

(e)

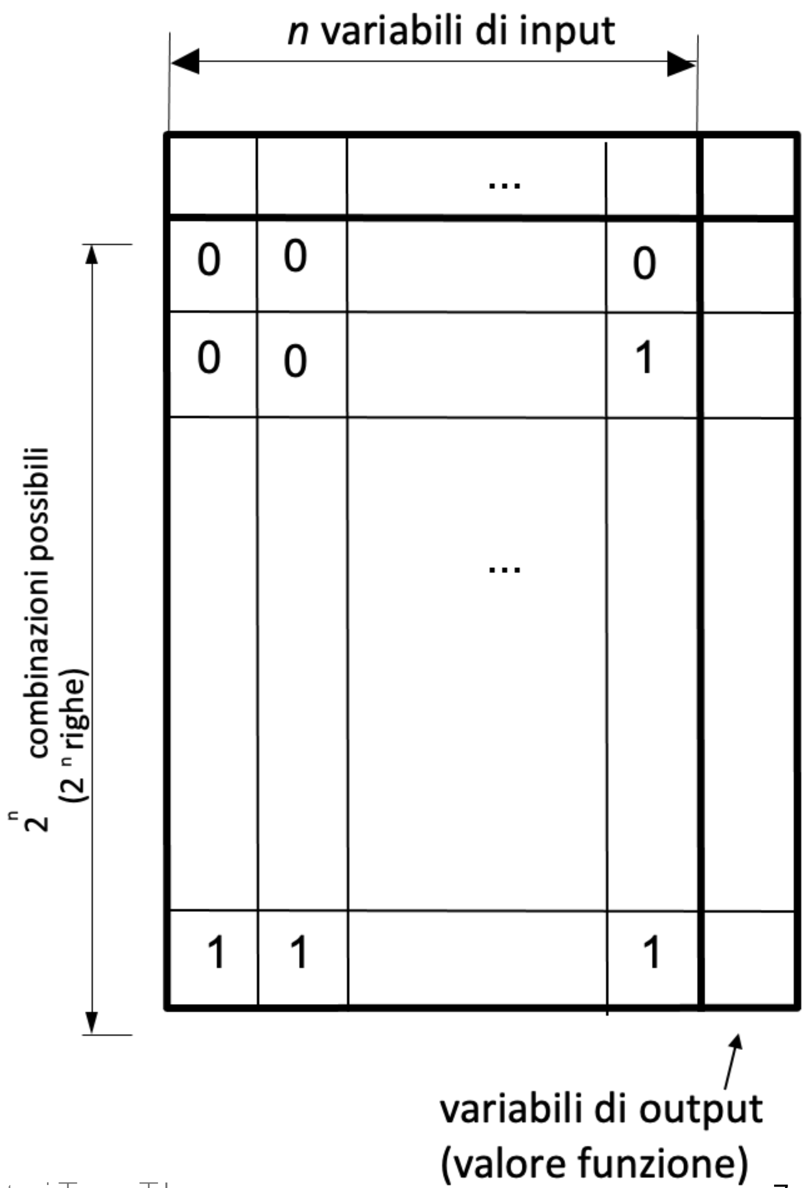
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(f)

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

algebra booleana

- Tabelle di verità: descrivono completamente il valore di una funzione Booleana attraverso tutte le combinazioni di input; n input corrispondono a 2^n combinazioni (righe)
- Forma canonica: righe ordinate per valori crescenti degli ingressi interpretando i valori delle variabili di ingresso come cifre di una codifica binaria



forme canoniche

- Formula normale disgiuntiva (FND):
  - Sommatoria (OR) di termini ciascuno dei quali è una produttoria (AND) di letterali costituiti da nomi di variabili di ingresso o da negazioni dei nomi di variabili di ingresso.
  - È minimale quando, applicando le proprietà algebriche di equivalenza non è possibile ottenere una FND equivalente contenente un numero di letterali inferiore
- Formula normale congiuntiva (FNC):
  - Concetto duale del precedente ossia è una produttoria (AND) di termini ciascuno dei quali è una sommatoria (OR) di letterali costituiti da nomi di variabili di ingresso o da negazioni di nomi di variabili di ingresso

funzione di maggioranza

- Funzione di maggioranza su tre input: restituisce 1 se la maggioranza degli input è 1; 0 altrimenti
- M (l'output) vale 1 nelle righe $\overline{A}BC; A\overline{B}C; AB\overline{C}; ABC$
- quindi la funzione M è, in forma normale disgiuntiva $M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$
  - mettendo in OR anche i casi dove M vale 0 non cambierebbe nulla

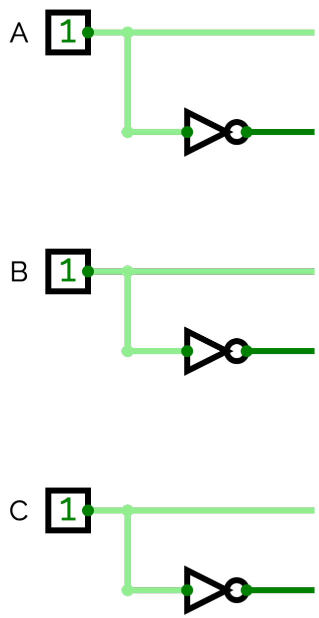
input ⟷ output

A	B	C	M
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

dalla tabella di verità alla formula algebrica

1. Scrivere la tabella di verità per la funzione
2. Disporre gli invertitori per generare il complemento di ogni input

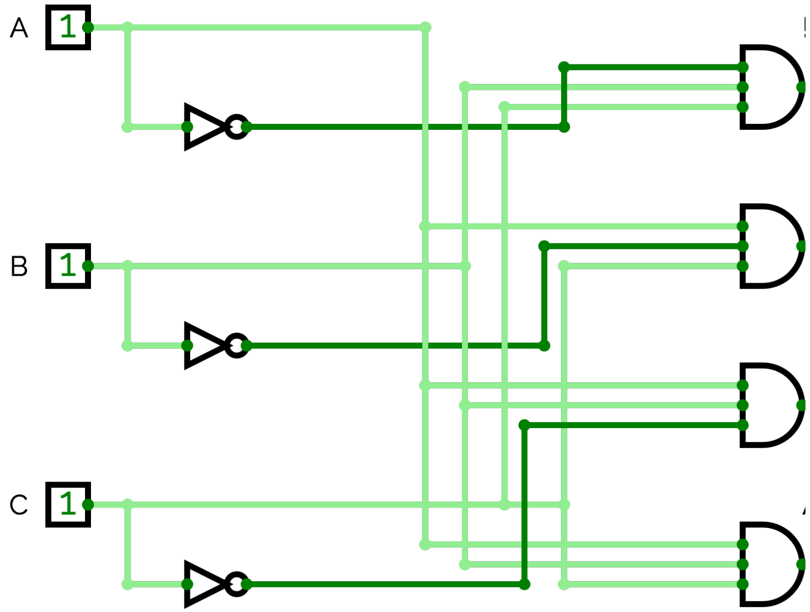
$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$



dalla tabella di verità alla formula algebrica

1. Scrivere la tabella di verità per la funzione
2. Disporre gli invertitori per generare il complemento di ogni input
3. Introdurre una porta AND per ogni termine con un 1 nella colonna dei risultati
4. Collegare le porte AND agli input appropriati

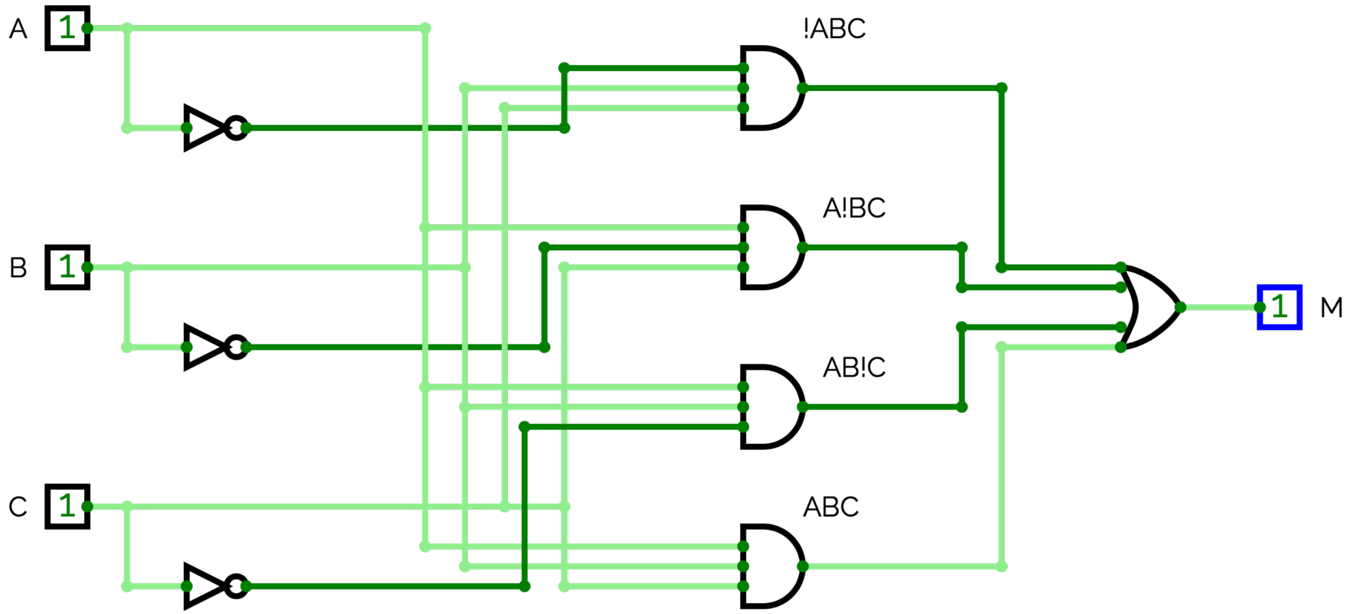
$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$



dalla tabella di verità alla formula algebrica

1. Scrivere la tabella di verità per la funzione
2. Disporre gli invertitori per generare il complemento di ogni input
3. Introdurre una porta AND per ogni termine con un 1 nella colonna dei risultati
4. Collegare le porte AND agli input appropriati
5. Inviare l'output di tutte le porte AND in una porta OR

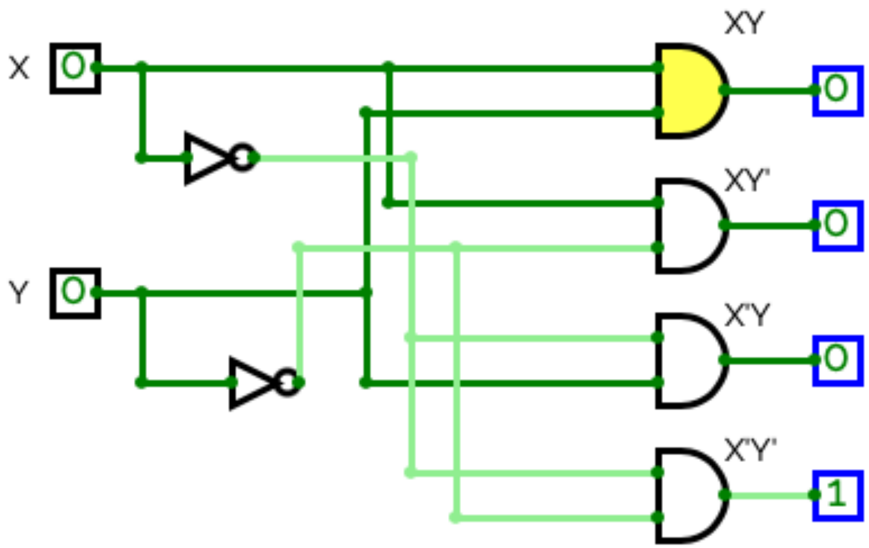
$$M = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

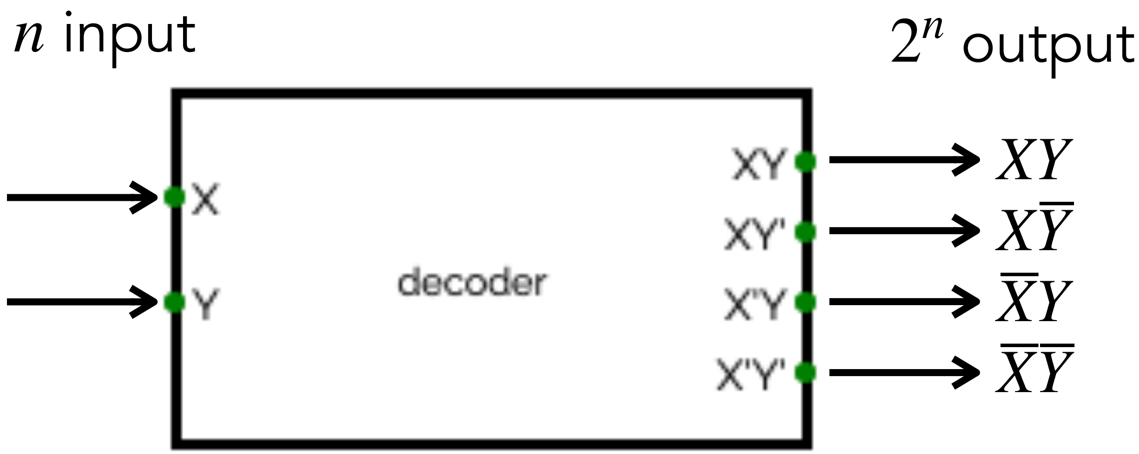


circuiti notevoli

decoder

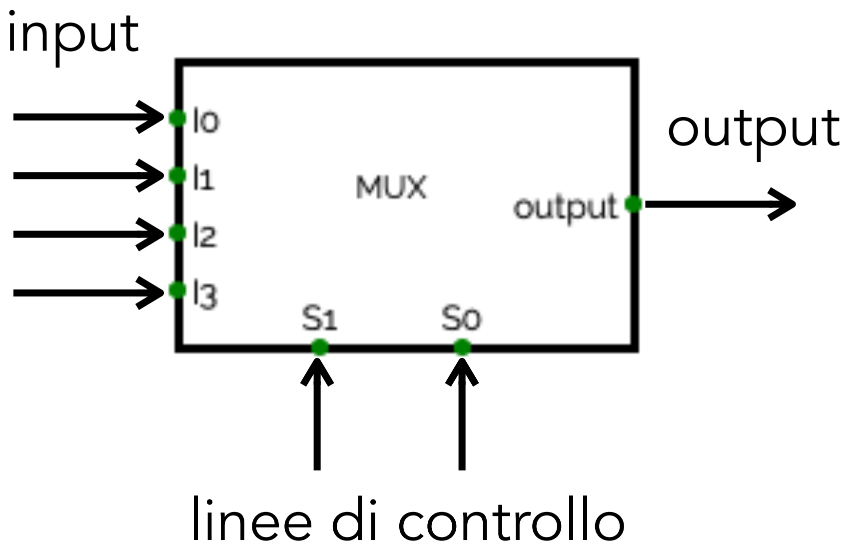
- decodifica input binario da un set di n input a 2^n output, in particolare seleziona una specifica linea di output, cioè una sola linea di output è posta a 1, mentre le altre restano a 0



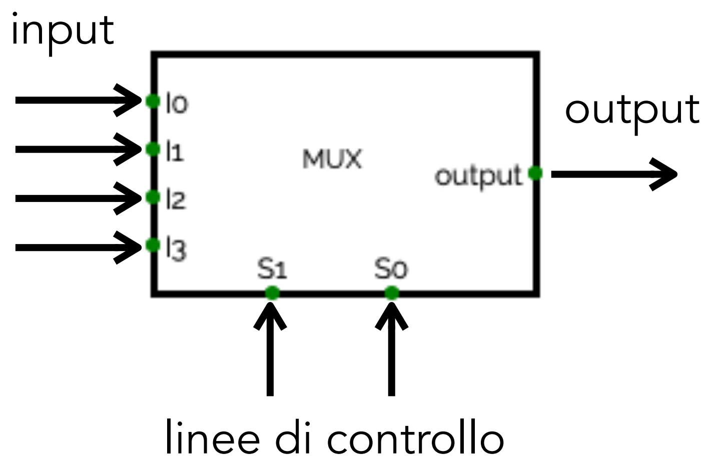


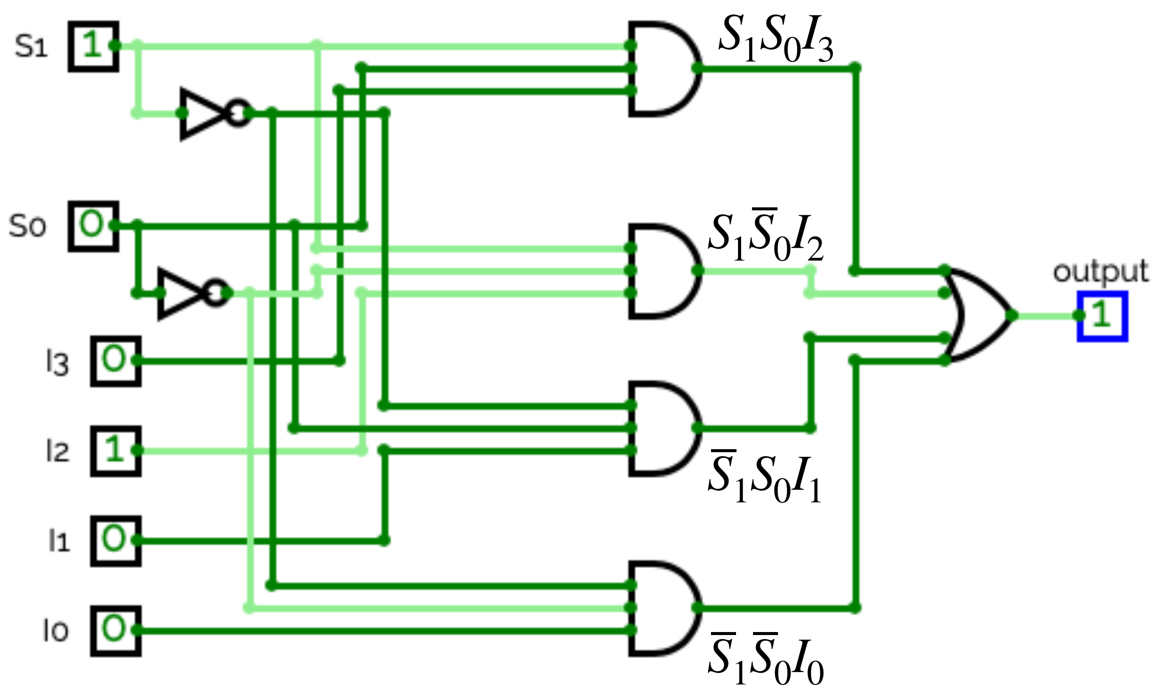
multiplexer (MUX)

- MUX circuita un segnale da una delle linee di input dirigendolo su un singolo output sulla base dell'informazione fornita dalle linee di controllo.

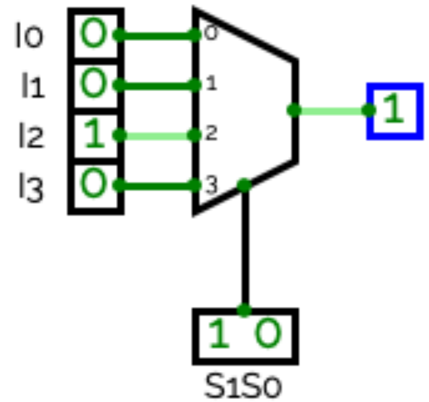


multiplexer (MUX), step into





$\Longleftrightarrow$

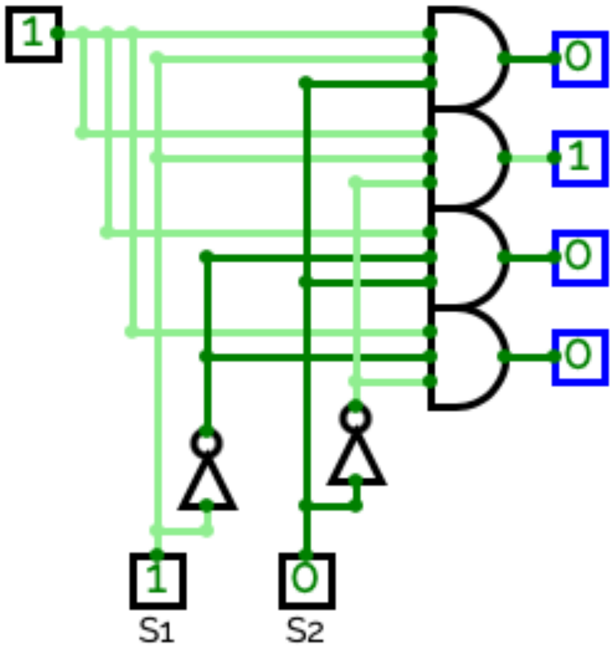


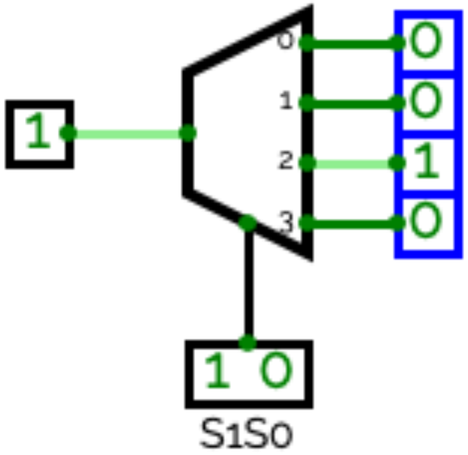
demultiplexer

- Il demultiplexer realizza la funzione inversa rispetto al multiplexer: distribuisce il segnale in input (unica linea di ingresso) su una delle uscite selezionata dagli ingressi di controllo.

demultiplexer

- data una unica linea di ingresso, il demultiplexer distribuisce il segnale su una delle uscite selezionata dagli ingressi di controllo.



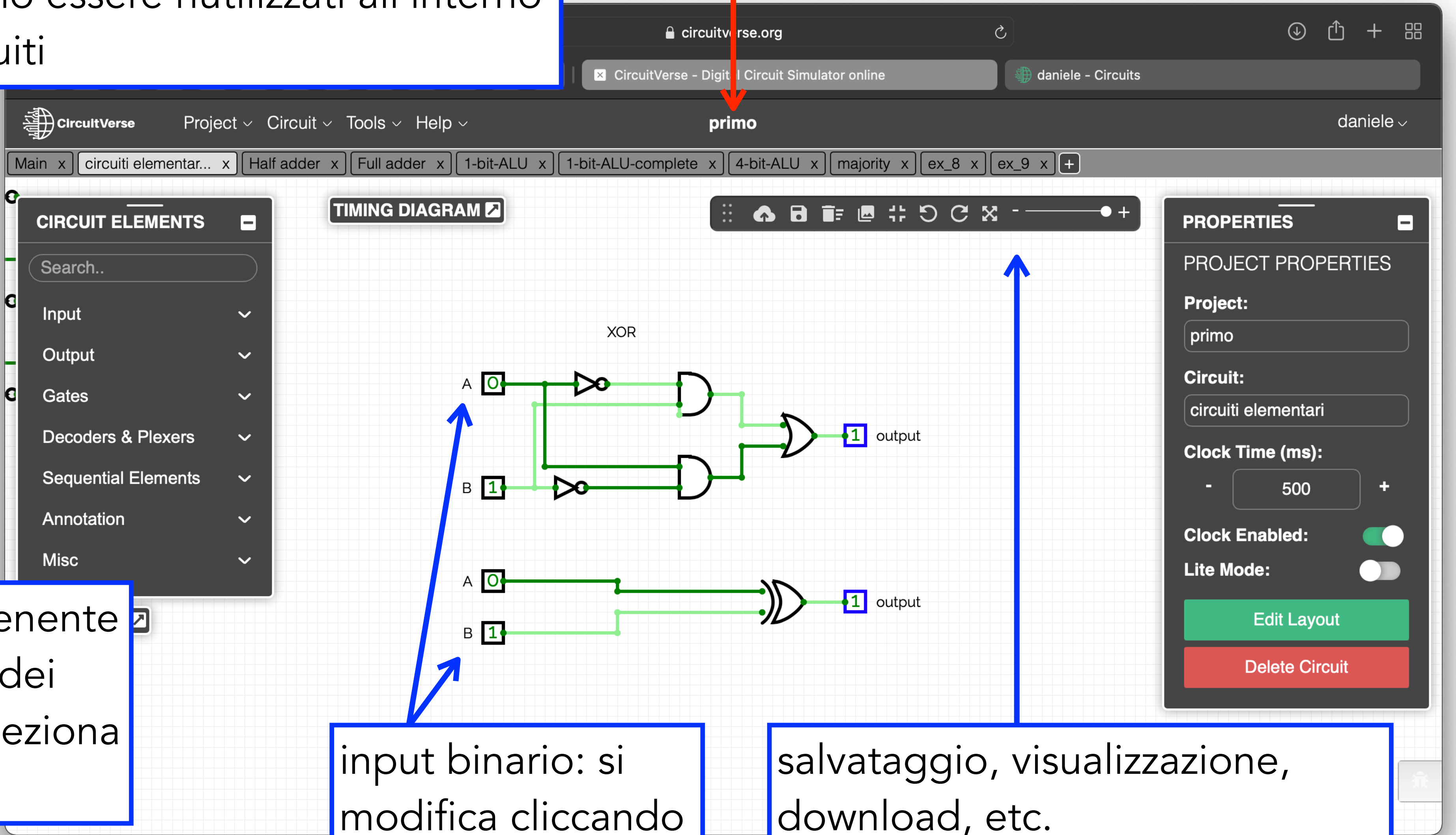


esercizi

simulatore CircuitVerse

- Cloud-based, realizzato in HTML5:
  - https://circuitverse.org
  - è necessario registrarsi con email/password per salvare i propri progetti e per accedere a quelli condivisi

un progetto è costituito da vari circuiti, che possono essere riutilizzati all'interno di altri circuiti



paletta contenente gli elementi dei circuiti: si seleziona trascinando

input binario: si modifica cliccando

salvataggio, visualizzazione, download, etc.

Esercizio 0 - esplorazione del tool

- Costruire i circuiti della slide precedente
  - Imparare a fare il «wiring» (collegare col mouse i fili), spostare, trascinare, mettere etichette
  - Modificare il suo layout, per poterlo riutilizzare come sottocircuito in futuro (ad es: spostare gli input/output lungo il bordo rettangolare)

Esercizio 1: half adder

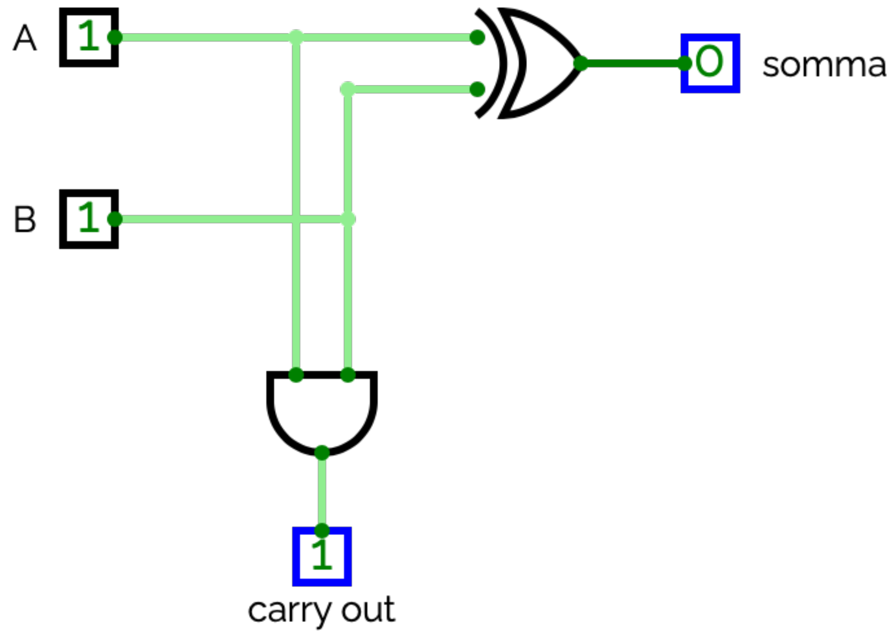
ingressi		uscite	
A	**B**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- costruire il circuito corrispondente all'half adder descritto in tabella.

Esercizio 1: half adder

ingressi		uscite	
A	**B**	**Sum**	**Carry**
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- costruire il circuito corrispondente all'half adder descritto in tabella.



Esercizio 2: full adder

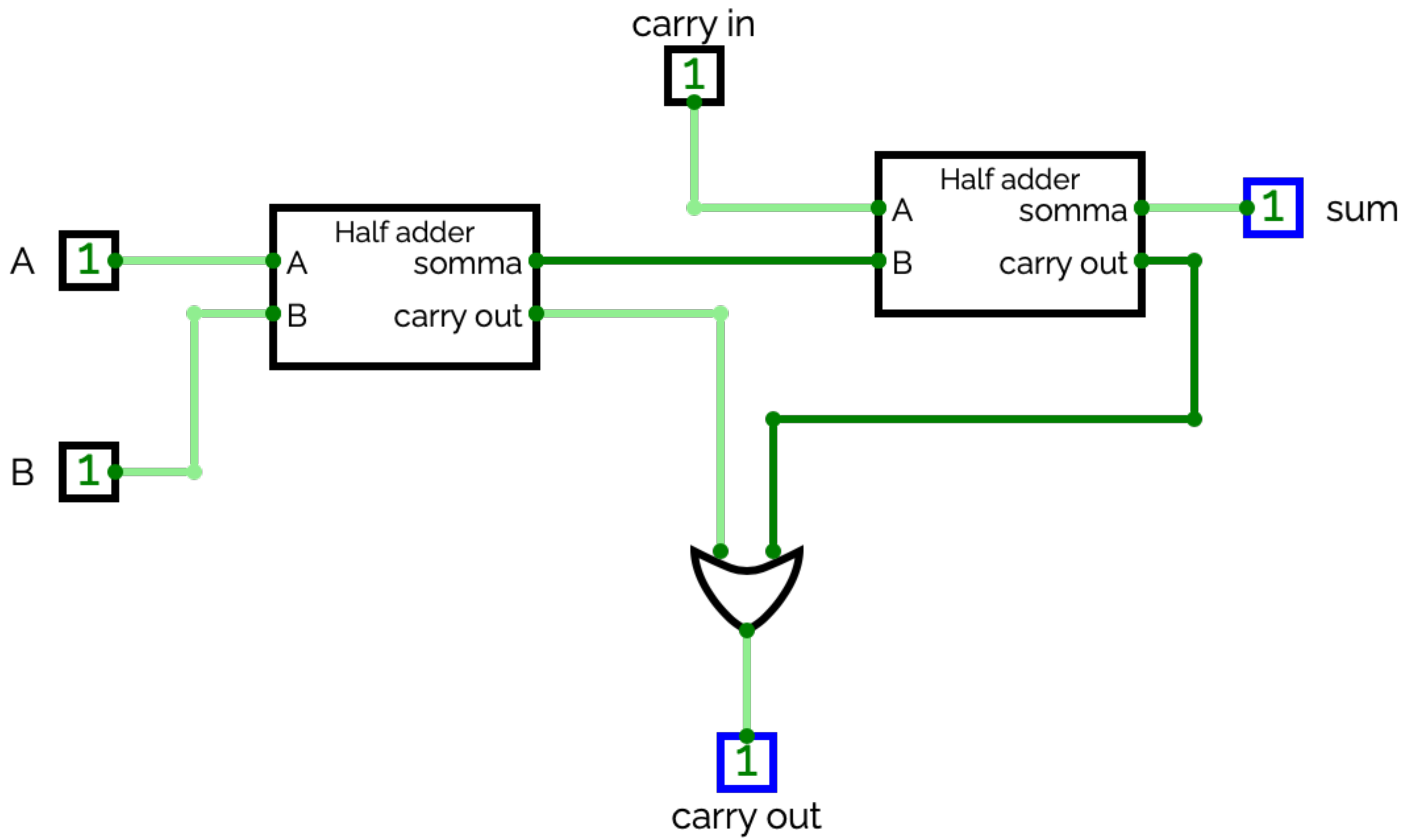
- costruzione del full adder, riutilizzando l'half adder dell'esercizio precedente.

nuovo input

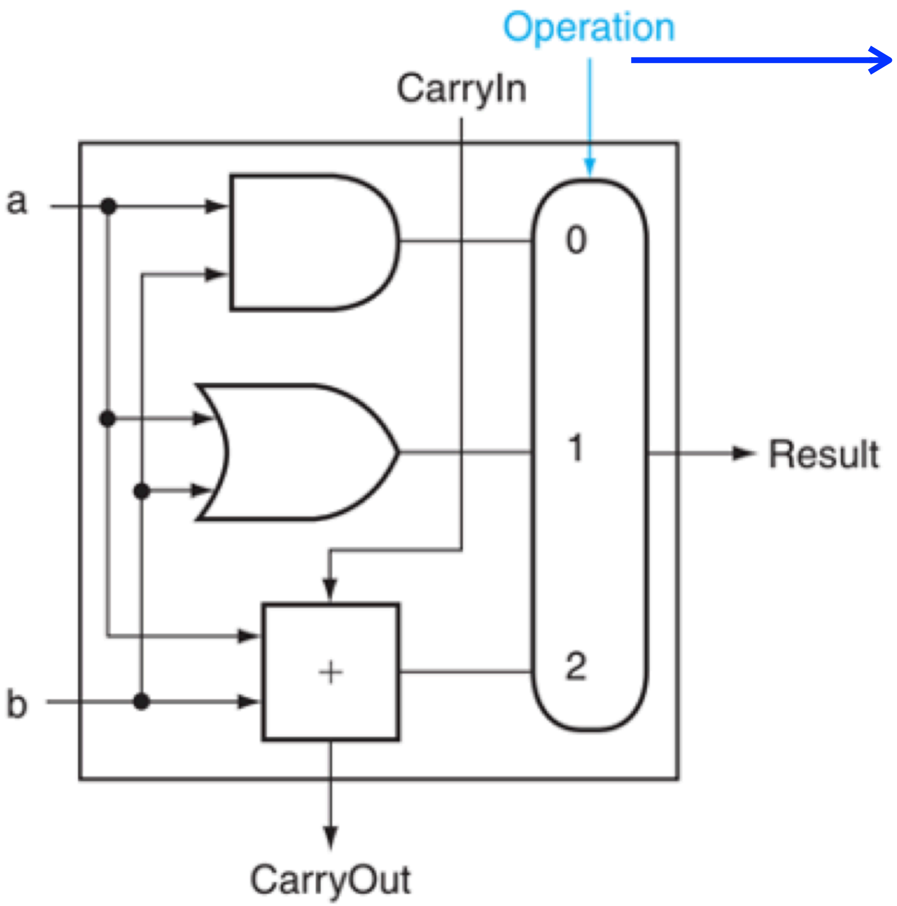
Inputs			Outputs		Comments
a	b	CarryIn	CarryOut	Sum	
0	0	0	0	0	$0 + 0 + 0 = 00_{two}$
0	0	1	0	1	$0 + 0 + 1 = 01_{two}$
0	1	0	0	1	$0 + 1 + 0 = 01_{two}$
0	1	1	1	0	$0 + 1 + 1 = 10_{two}$
1	0	0	0	1	$1 + 0 + 0 = 01_{two}$
1	0	1	1	0	$1 + 0 + 1 = 10_{two}$
1	1	0	1	0	$1 + 1 + 0 = 10_{two}$
1	1	1	1	1	$1 + 1 + 1 = 11_{two}$

FIGURE A.5.3 Input and output specification for a 1-bit adder.

Esercizio 2: full adder

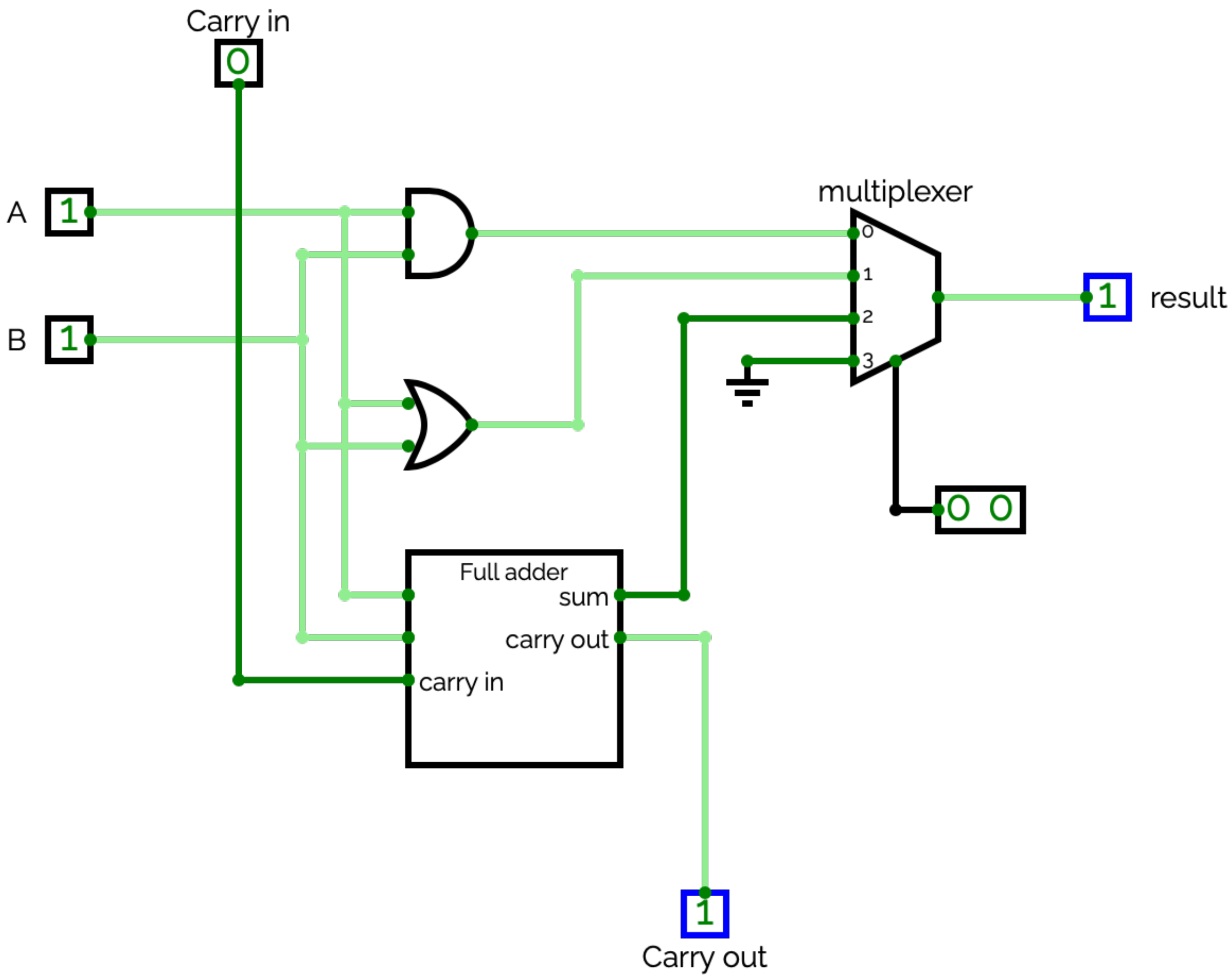


Esercizio 3: ALU a 1 bit (semplificata)



- segnale di controllo codificato su 2 bit, che indica l'operazione:
  - 00: AND logico
  - 01: OR logico
  - 10: somma aritmetica completa
  - 11: non usato

Esercizio 3: ALU a 1 bit (semplificata)

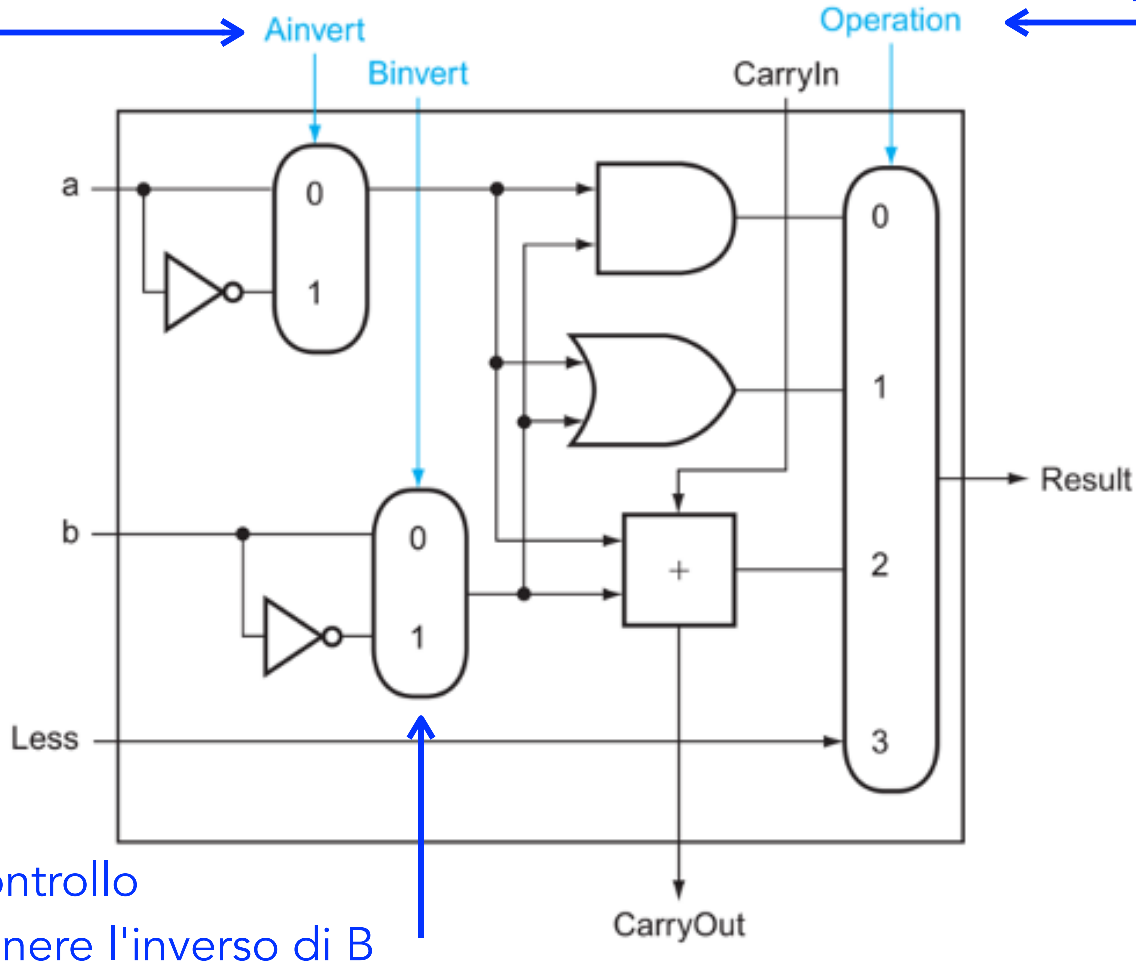


di.unito.it
Dipartimento di Informatica

Esercizio 4: ALU a 1 bit (completa)

bit di controllo
per ottenere l'inverso di A

multiplexer

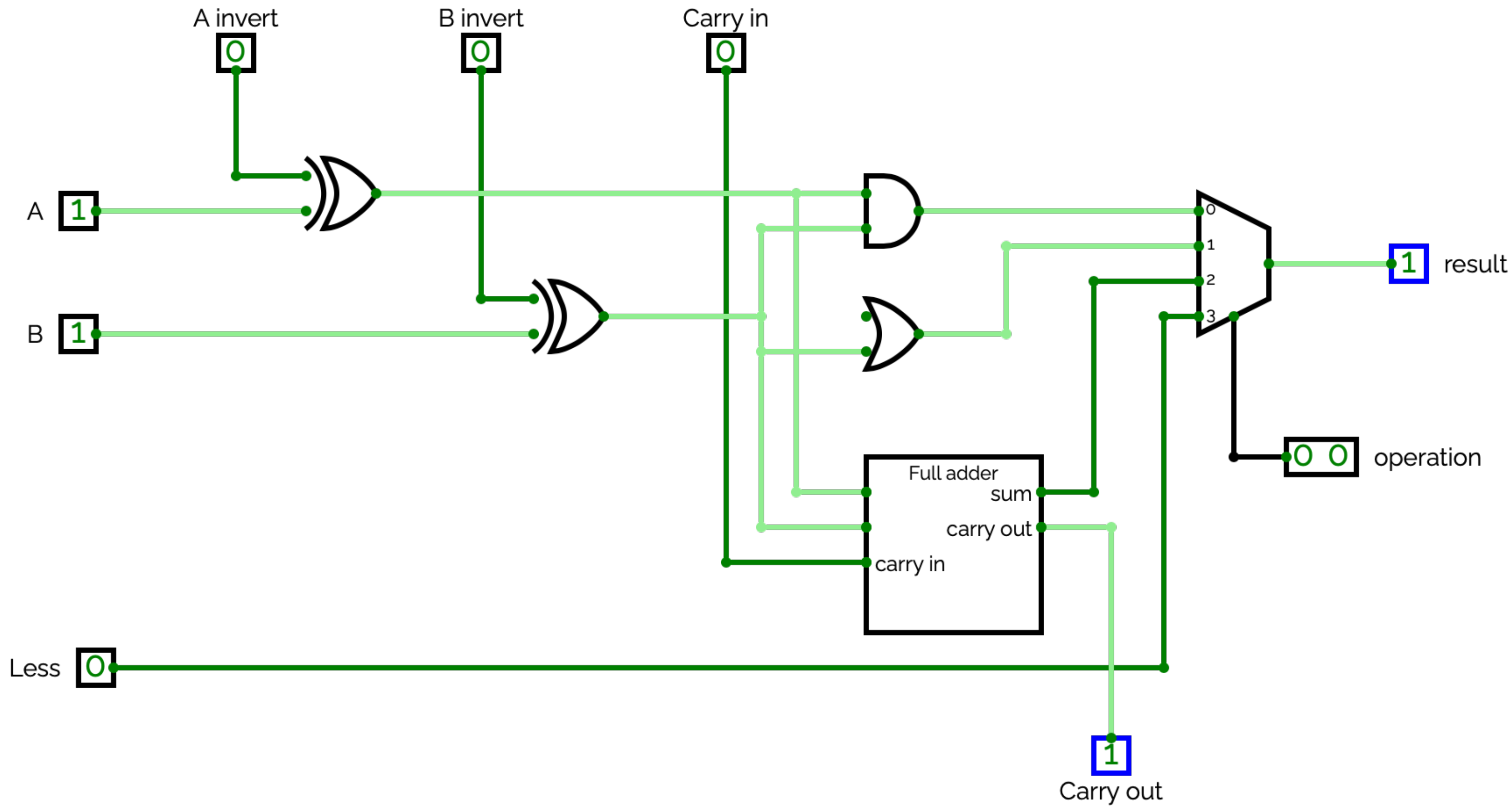


bit ulteriore di input
collegato al caso 11 del
multiplexer

bit di controllo
per ottenere l'inverso di B

dı.unıto.it
DIPARTIMENTO
DI INFORMATICA

Esercizio 4: ALU a 1 bit (completa)



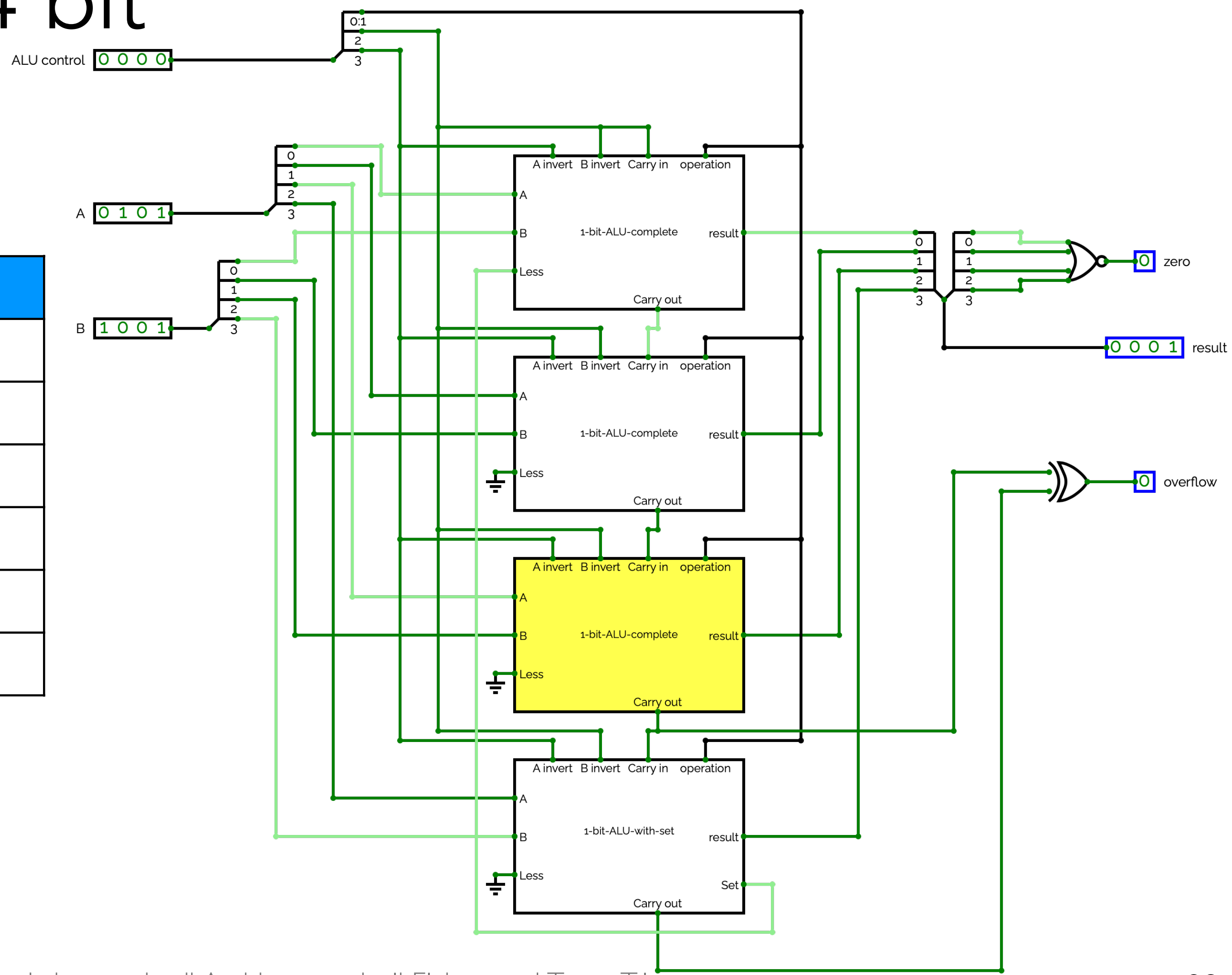
Esercizio 5: ALU a 4 bit

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set less than
1100	NOR

- Mettendo in cascata 4 ALU a 1 bit come la precedente, pensare a come costruire una ALU completa a 4 bit in grado di fare le seguenti operazioni su ingressi A e B a 4 bit:
  - AND
  - OR
  - add
  - subtract
  - set less than (restituisce 1 se A<B; 0 altrimenti)
  - NOR
- aggiungere anche l'uscita zero (che vale 1 se il risultato è zero) e l'uscita che segnala overflow.

Esercizio 5: ALU a 4 bit

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set less than
1100	NOR



di.unito.it DIPARTIMENTO DI INFORMATICA

Esercizio 6:

ALU control	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set less than
1100	NOR

1. Dato A = 1101, estrarre i suoi 2 bit meno significa
2. Calcolare i 4 possibili risultati di X OR Y
3. Dati A = 5, B = 3, fare la somma A + B
4. Dati A = 5, B = -3, fare la somma A + B
5. Dati A = -5, B = 3, fare la sottrazione A - B
6. Con A = 3, B = 5, stabilire con «set less than» se A < B
7. Con A = 3, B = -5, stabilire con «set less than» se A < B. Cosa succede?
8. Con A = -5, B = 5, stabilire se A < B. Cosa succede?
9. Per generici A e B, calcolare se A=B
10. Calcolare i 4 possibili risultati di X NOR Y

Esercizio 7: semplificazione di espressione

- dimostrare che

$$\overline{AB + \bar{A}C} = A\bar{B} + \bar{A}\bar{C}$$

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

Esercizio 7: semplificazione di espressione

- dimostrare che

$$\overline{AB + \overline{A}C} = A\overline{B} + \overline{A}\,\overline{C}$$

- applichiamo De Morgan

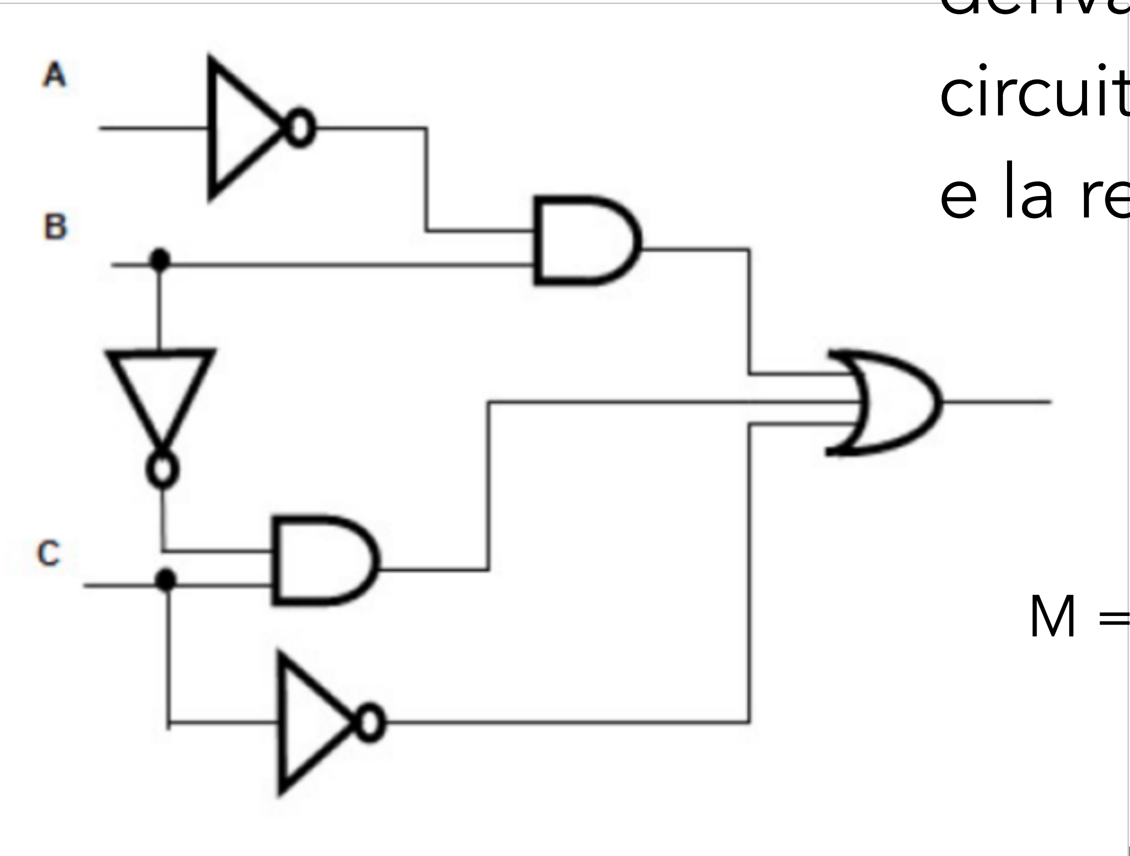
$$(\overline{A} + \overline{B})(A + \overline{C}) = \overline{A}A + \overline{A}\,\overline{C} + A\overline{B} + \overline{B}\,\overline{C} =$$

$$A\overline{B} + \overline{A}\,\overline{C} + \overline{B}\,\overline{C} = A\overline{B} + \overline{A}\,\overline{C}$$

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\overline{A} = 0$	$A + \overline{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A}\,\overline{B}$

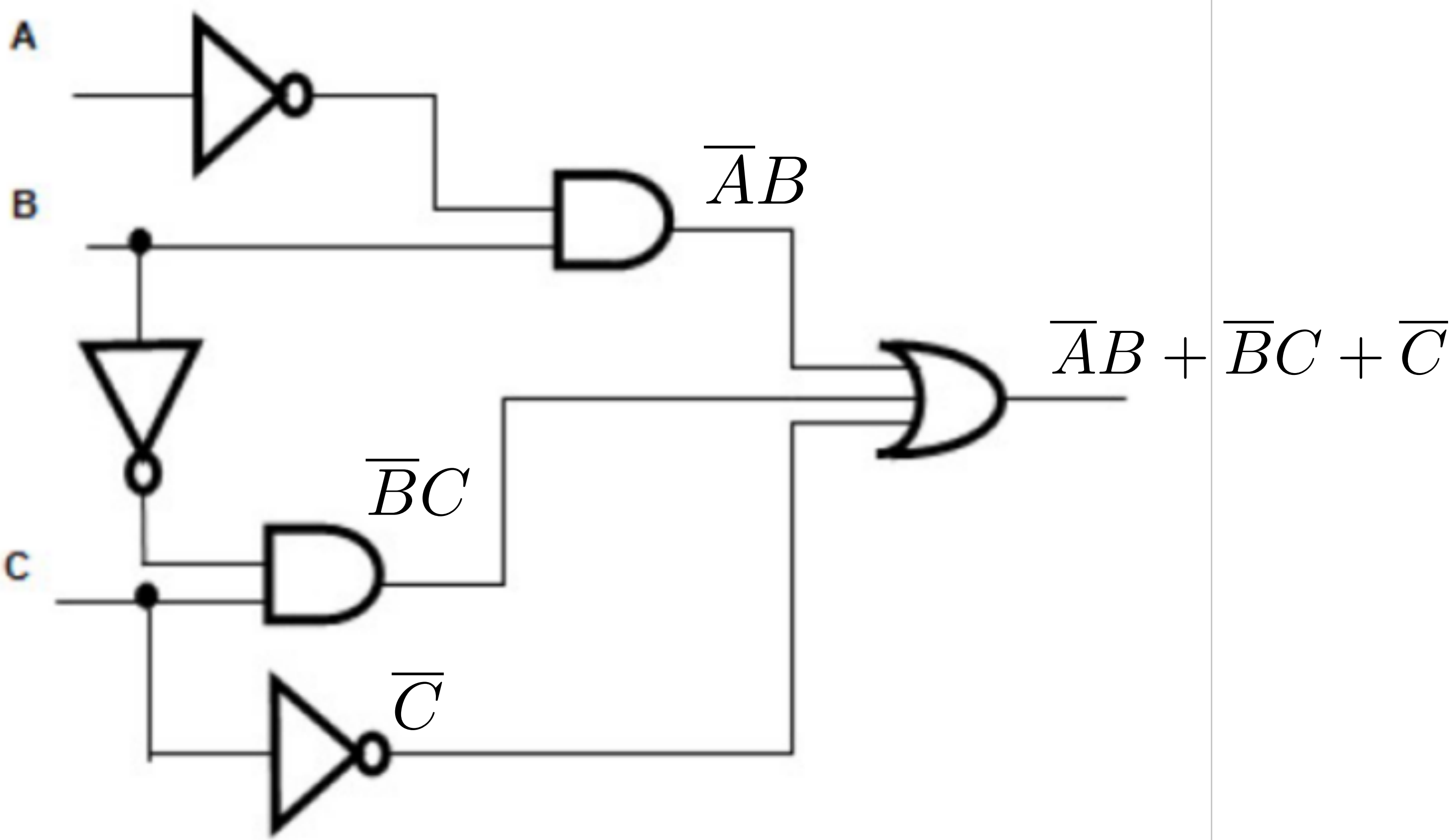
Esercizio 8: analisi di circuito

- derivare la tabella di verità dal circuito rappresentato in figura e la relativa funzione.



M =

Esercizio 8: analisi di circuito



esercizio 9

- costruire la tabella di verità e un circuito combinatorio a due entrate I_1e I_0 e quattro uscite O_3, O_2, O_1 e O_0 tale che dato in input un numero X codificato in binario puro su due bit (I_1, I_0) produca in output il suo quadrato $Y = X^2$ codificato sempre i binario puro su quattro bit $(O_3; O_2; O_1; O_0)$.

esercizio 9

- costruire la tabella di verità e un circuito combinatorio a due entrate I_1e I_0 e quattro uscite O_3, O_2, O_1 e O_0 tale che dato in input un numero X codificato in binario puro su due bit (I_1, I_0) produca in output il suo quadrato $Y = X^2$ codificato sempre i binario puro su quattro bit $(O_3; O_2; O_1; O_0)$.

X	I_1	I_0	O_3	O_2	O_1	O_0	Y
0	0	0					0
1	0	1					1
2	1	0					4
3	1	1					9

esercizio 9

- costruire la tabella di verità e un circuito combinatorio a due entrate I_1e I_0 e quattro uscite O_3, O_2, O_1 e O_0 tale che dato in input un numero X codificato in binario puro su due bit (I_1, I_0) produca in output il suo quadrato $Y = X^2$ codificato sempre i binario puro su quattro bit $(O_3; O_2; O_1; O_0)$.

X	I_1	I_0	O_3	O_2	O_1	O_0	Y
0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	1
2	1	0	0	1	0	0	4
3	1	1	1	0	0	1	9

altri esercizi

da espressione booleana a circuito (1)

- disegnare un circuito corrispondente alla seguente espressione: $(A+B)(B+C)$

da espressione booleana a circuito (2)

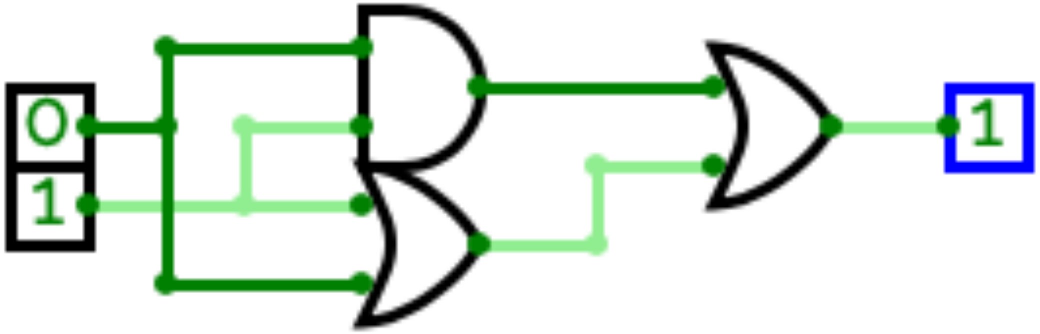
- disegnare un circuito corrispondente alla seguente espressione: $(AB + C)D$

da espressione booleana a circuito (3)

- disegnare un circuito corrispondente alla seguente espressione: $\overline{A}B + (\overline{B + C})$

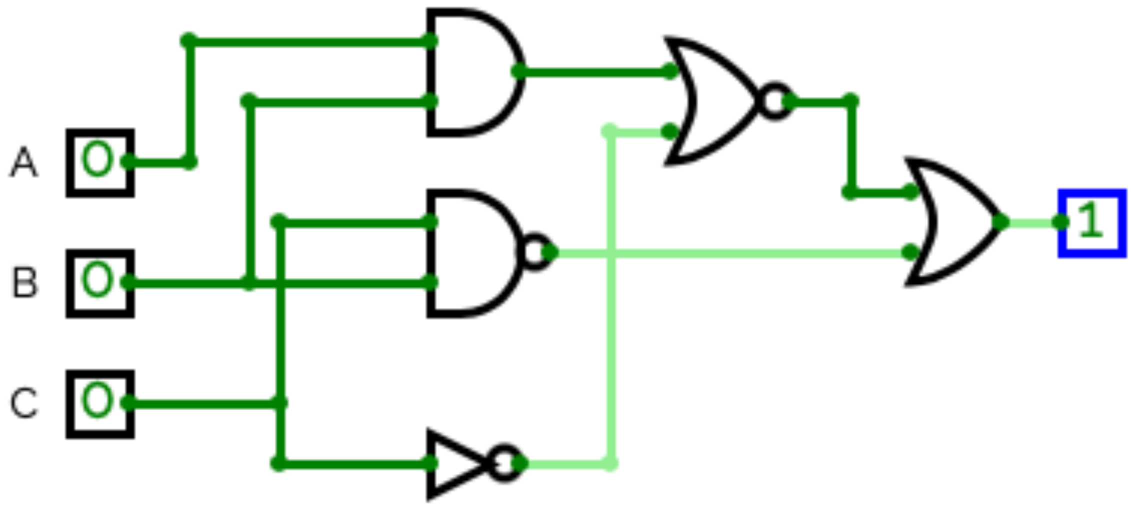
da circuito a tabella (1)

- costruire la tabella di verità corrispondente al circuito.



da circuito a tabella (2)

- costruire la tabella di verità corrispondente al circuito.



somma di numeri a 2 cifre (binarie)

- costruire su CircuitVerse il circuito visualizzato nella slide successiva utilizzando 2 full adders (realizzati nell'esercizio 2) per sommare due numeri a 2 cifre binarie
  - verificare il risultato per la somma delle coppie
    - A={10}; B={01}
    - A={10}; B={11}

materiale della lezione

https://circuitverse.org/simulator/2205101642_architettura_b_t1