Architettura degli Elaboratori

Esercitazione

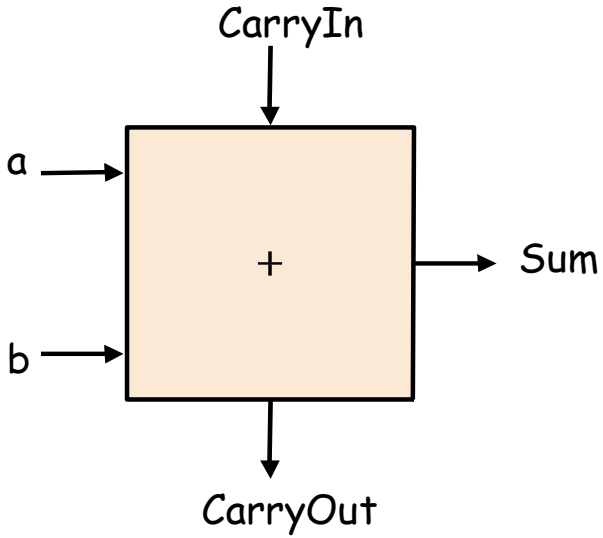
Barbara Masucci
UNIVERSITA DEGLI STUDI DI SALERNO
DIPARTIMENTO DI INFORMATICA
DIPARTIMENTO DI ECCELLENZA

Su cosa ci esercitiamo oggi?

- Addizionatore binario
- Unità Aritmetico-Logica (ALU)

Esercizio 1

- Progettare un PLA (Array Logico Programmabile) che realizzi un addizionatore ad 1 bit



Esercizio 1: Soluzione

- Ricaviamo la tavola di verità delle funzioni
  - Sum=Somma(a,b,CarryIn)
  - CarryOut= Riporto(a,b,CarryIn)

- Successivamente, esprimiamo le funzioni in forma canonica SOP

Esercizio 1: Soluzione

Tavola di verità funzione Sum

a	b	CarryIn	Sum	
0	0	0	0	
0	0	1	1	$\overline{a}\cdot\overline{b}\cdot CarryIn$
0	1	0	1	$\overline{a}\cdot b\cdot\overline{CarryIn}$
0	1	1	0	
1	0	0	1	$a\cdot\overline{b}\cdot\overline{CarryIn}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$a\cdot b\cdot CarryIn$

$$Sum = \overline{a}\cdot\overline{b}\cdot CarryIn + \overline{a}\cdot b\cdot\overline{CarryIn} + a\cdot\overline{b}\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

Esercizio 1: Soluzione

Tavola di verità funzione CarryOut

a	b	CarryIn	CarryOut	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{a}\cdot b\cdot CarryIn$
1	0	0	0	
1	0	1	1	$a\cdot\bar{b}\cdot CarryIn$
1	1	0	1	$a\cdot b\cdot\overline{CarryIn}$
1	1	1	1	$a\cdot b\cdot CarryIn$

$$CarryOut = \bar{a}\cdot b\cdot CarryIn + a\cdot\bar{b}\cdot CarryIn + a\cdot b\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

$$= \bar{a}\cdot b\cdot CarryIn + a\cdot b\cdot CarryIn + a\cdot\bar{b}\cdot CarryIn + a\cdot b\cdot CarryIn + a\cdot b\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

$$= a\cdot CarryIn + b\cdot CarryIn + a\cdot b$$

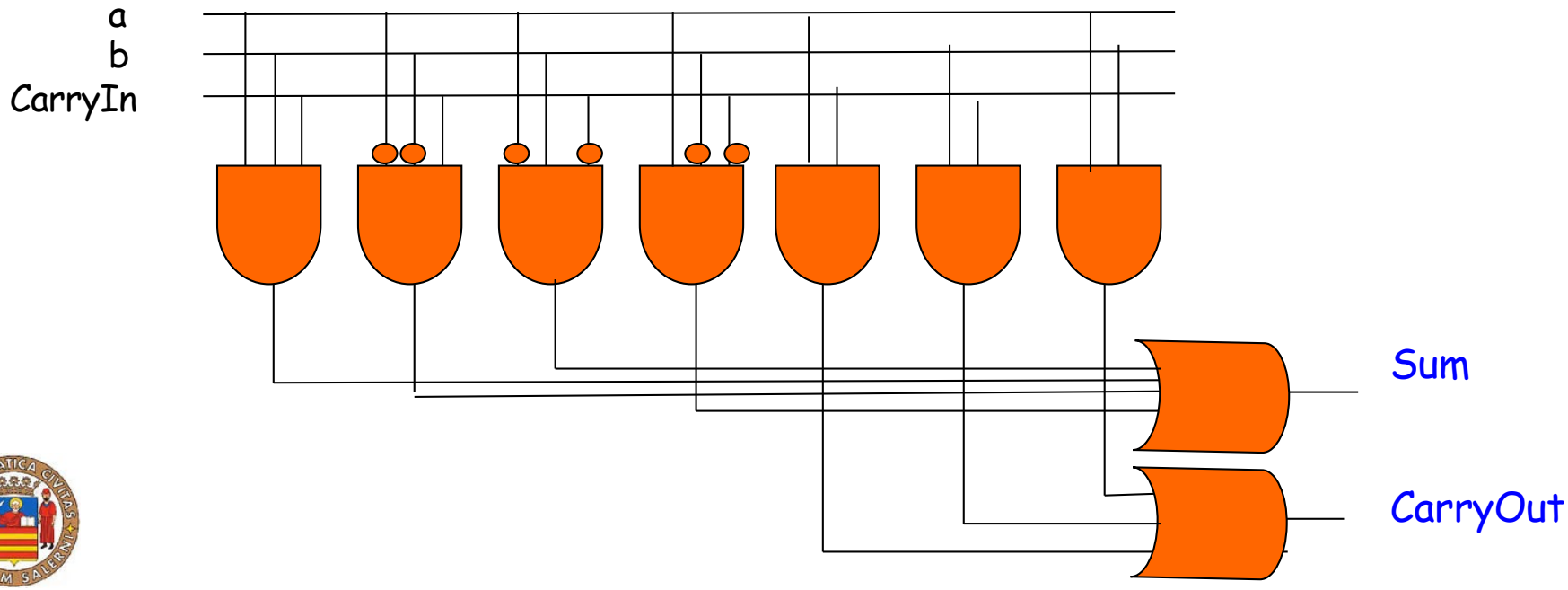
Minimizziamo con le proprietà di idempotenza e del complemento

Esercizio 1: Soluzione

- Otteniamo la rete AND-to-OR per le funzioni

$$Sum = \overline{a}\cdot\overline{b}\cdot CarryIn + \overline{a}\cdot b\cdot\overline{CarryIn} + a\cdot\overline{b}\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

$$CarryOut = a\cdot CarryIn + b\cdot CarryIn + a\cdot b$$

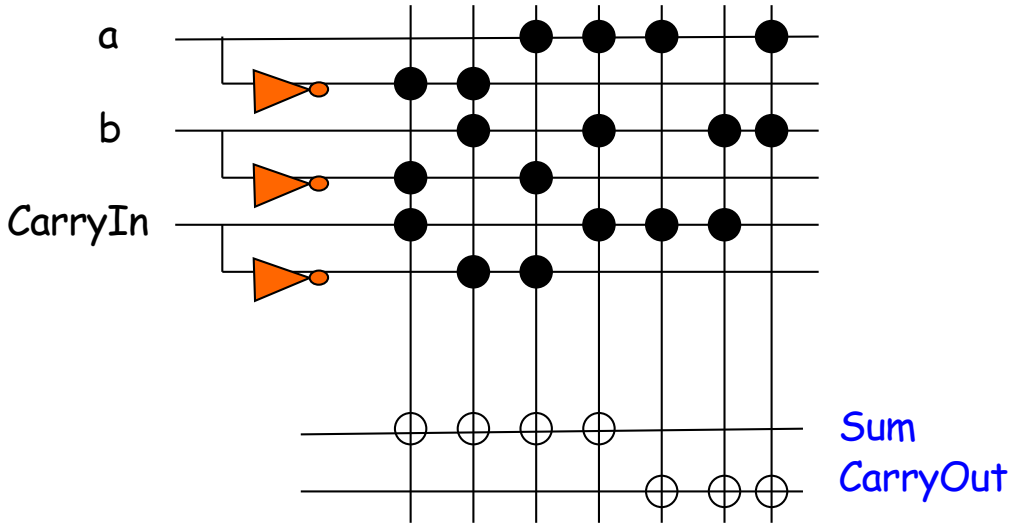


Esercizio 1: Soluzione

- Rappresentazione equivalente sotto forma di griglia

$$Sum = \overline{a}\cdot\overline{b}\cdot CarryIn + \overline{a}\cdot b\cdot\overline{CarryIn} + a\cdot\overline{b}\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

$$CarryOut = a\cdot CarryIn + b\cdot CarryIn + a\cdot b$$



Esercizio 2

- Progettare una rete combinatoria che realizzi le funzioni Sum e CarryOut usando porte AND, OR e XOR

Esercizio 2: Soluzione

$$\text{Sum} = \overline{a}\cdot\overline{b}\cdot CarryIn + \overline{a}\cdot b\cdot\overline{CarryIn} + a\cdot\overline{b}\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$
$$= (a\cdot\overline{b} + \overline{a}\cdot b)\cdot\overline{CarryIn} + (\overline{a}\cdot\overline{b} + a\cdot b)\cdot CarryIn$$
$$= (a \oplus b)\cdot\overline{CarryIn} + (a \oplus b)\cdot CarryIn$$
$$= a \oplus b \oplus CarryIn$$

a	b	CarryIn	Sum	
0	0	0	0	
0	0	1	1	$\overline{a}\cdot\overline{b}\cdot CarryIn$
0	1	0	1	$\overline{a}\cdot b\cdot\overline{CarryIn}$
0	1	1	0	
1	0	0	1	$a\cdot\overline{b}\cdot\overline{CarryIn}$
1	0	1	0	
1	1	0	0	
1	1	1	1	$a\cdot b\cdot CarryIn$

Esercizio 2: Soluzione

a	b	CarryIn	CarryOut	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\bar{a}\cdot b\cdot CarryIn$
1	0	0	0	
1	0	1	1	$a\cdot\bar{b}\cdot CarryIn$
1	1	0	1	$a\cdot b\cdot\overline{CarryIn}$
1	1	1	1	$a\cdot b\cdot CarryIn$

$$CarryOut = \bar{a}\cdot b\cdot CarryIn + a\cdot\bar{b}\cdot CarryIn + a\cdot b\cdot\overline{CarryIn} + a\cdot b\cdot CarryIn$$

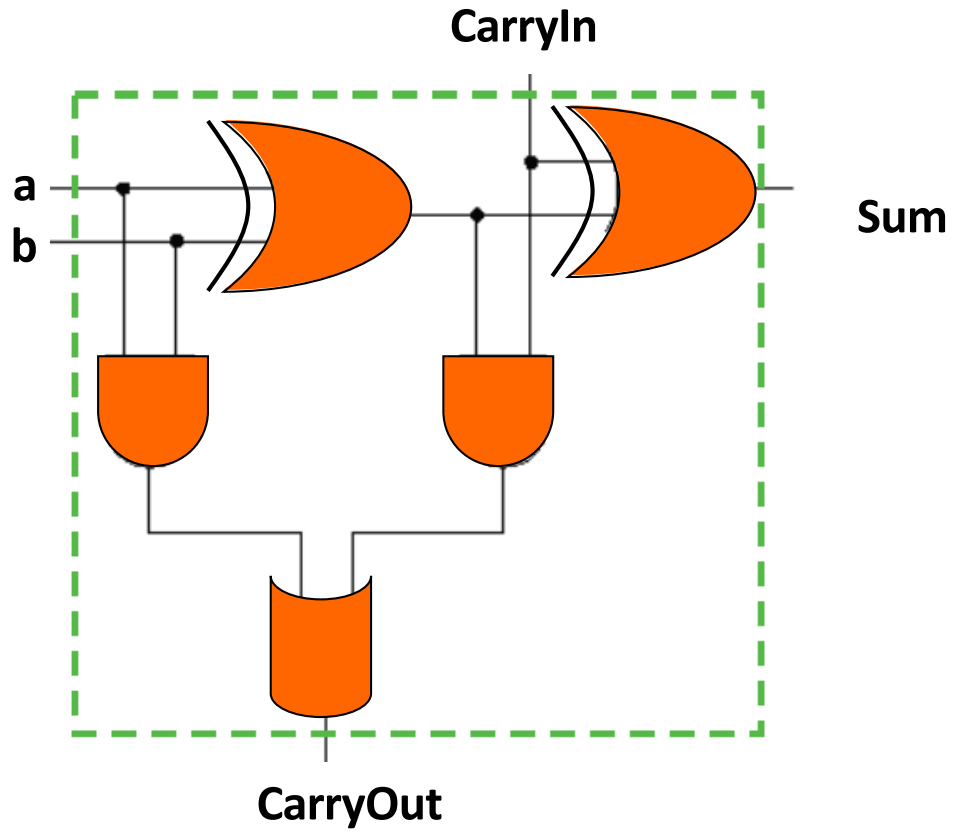
$$= (\bar{a}\cdot b + a\cdot\bar{b})\cdot CarryIn + a\cdot b\cdot(\overline{CarryIn} + CarryIn)$$

$$= (a \oplus b)\cdot CarryIn + a\cdot b$$

Esercizio 2: Soluzione

$$Sum = a \oplus b \oplus CarryIn$$

$$CarryOut = (a \oplus b) \cdot CarryIn + a \cdot b$$



Esercizio 3

- Progettare una ALU ad 1 bit che realizzi le seguenti funzioni, specificando gli opportuni segnali di controllo
  - AND
  - OR
  - NAND
  - NOR
  - Somma
  - Sottrazione

Esercizio 3: Soluzione

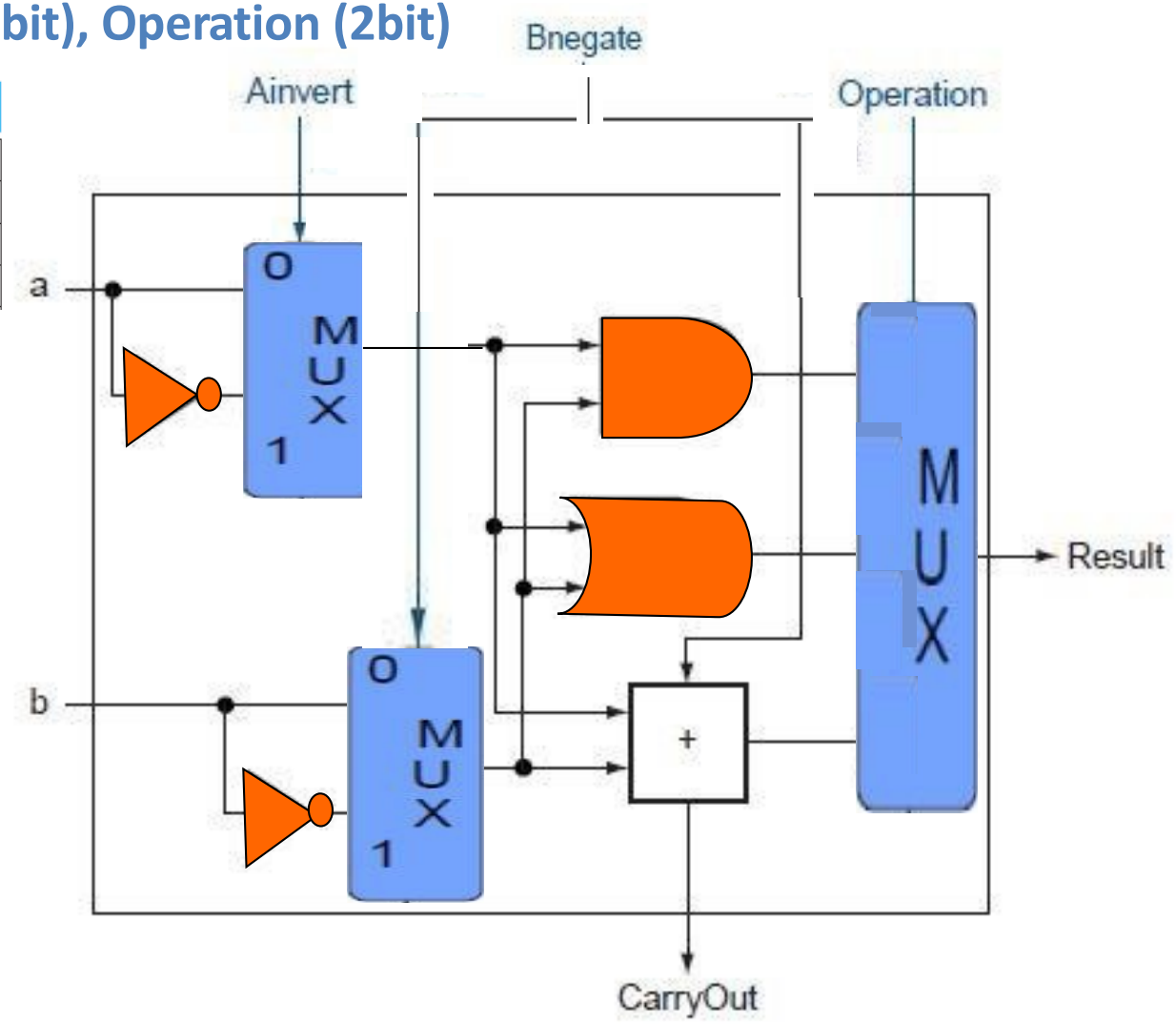
- Ricordiamo che abbiamo già costruito un'ALU ad un bit per realizzare
  - AND
  - OR
  - NOR
  - Somma
  - Sottrazione

- Come aggiungere il NAND?

Esercizio 3: Soluzione

Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
1100	NOR



Esercizio 3: Soluzione

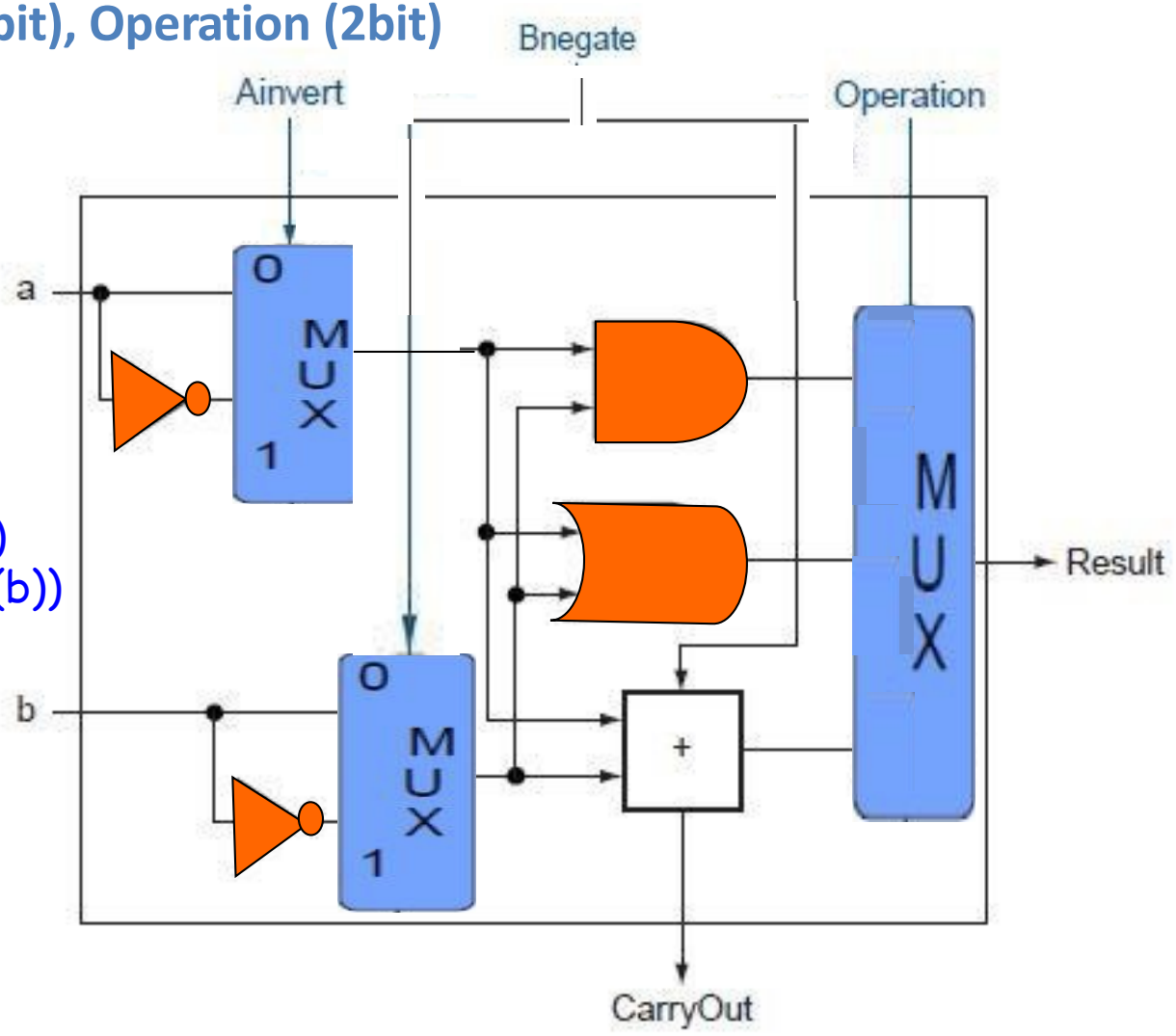
Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
1100	NOR

Poiché

NAND(a,b) = NOT(AND(a,b))
= OR(NOT(a),NOT(b))

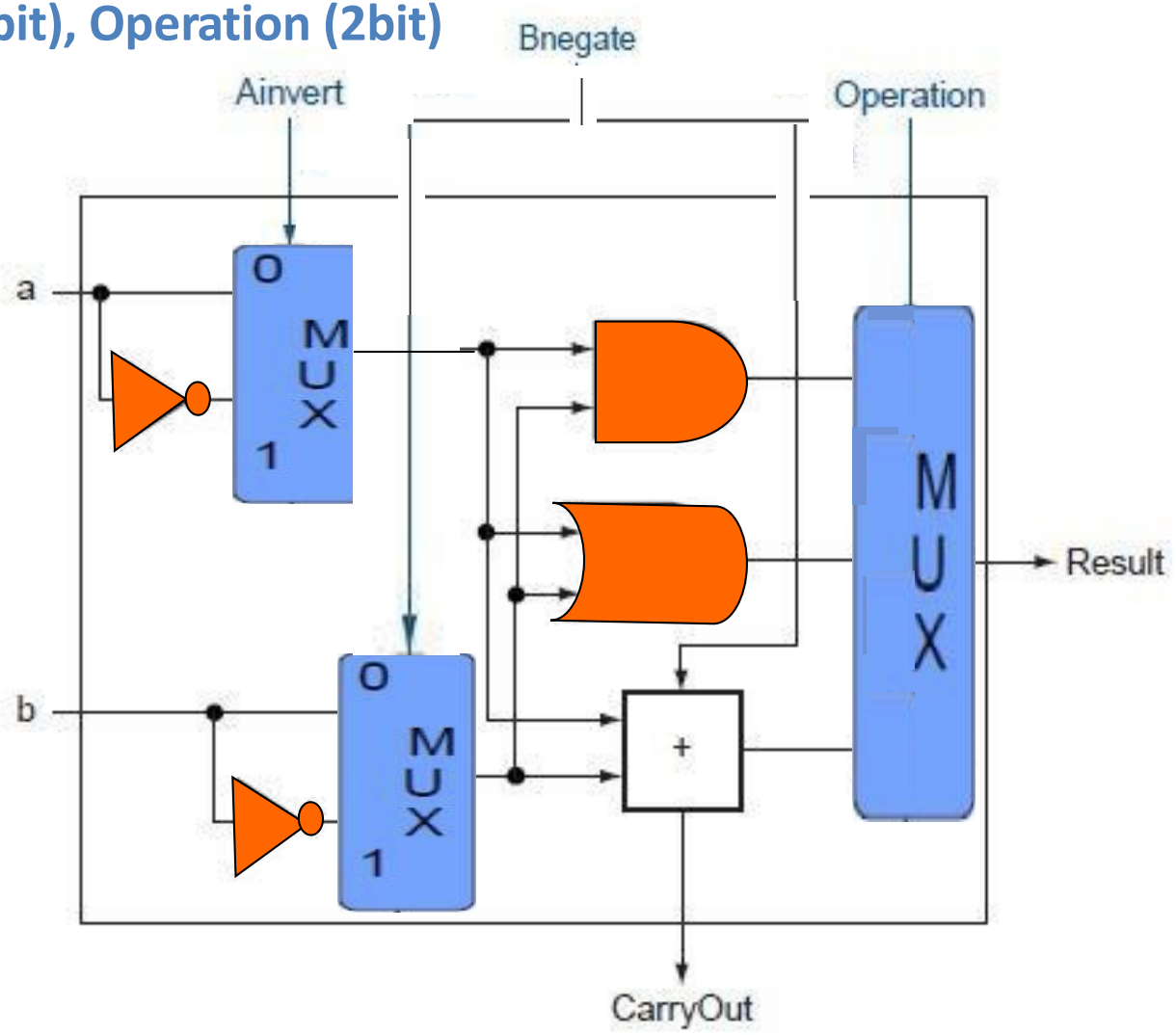
basta aggiungere
una riga alla tabella



Esercizio 3: Soluzione

Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
1100	NOR
1101	NAND

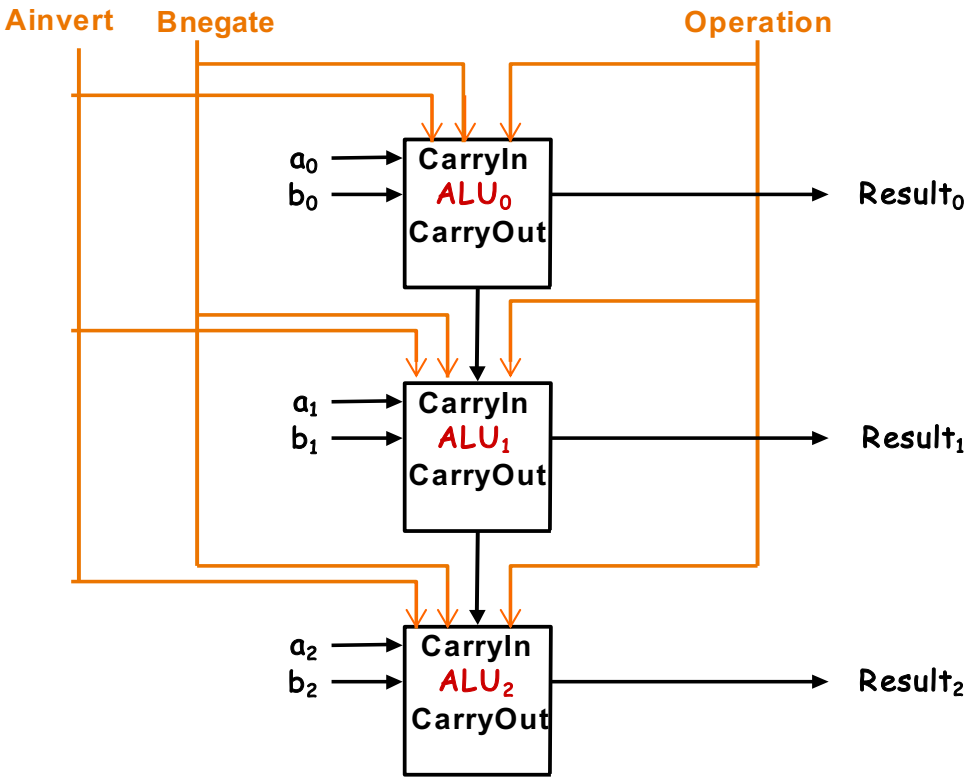


Esercizio 4

- Progettare una ALU a 3 bit che realizzi le seguenti funzioni, specificando gli opportuni segnali di controllo
  - AND
  - OR
  - NAND
  - NOR
  - Somma
  - Sottrazione

Esercizio 4: Soluzione

- Input:
  - $a=a_2a_1a_0$
  - $b=b_2b_1b_0$
- Output:
  - Result $=Result_2Result_1Result_0$
- Concateniamo tre copie dell'ALU costruita
  - ALU_0, ALU_1, ALU_2

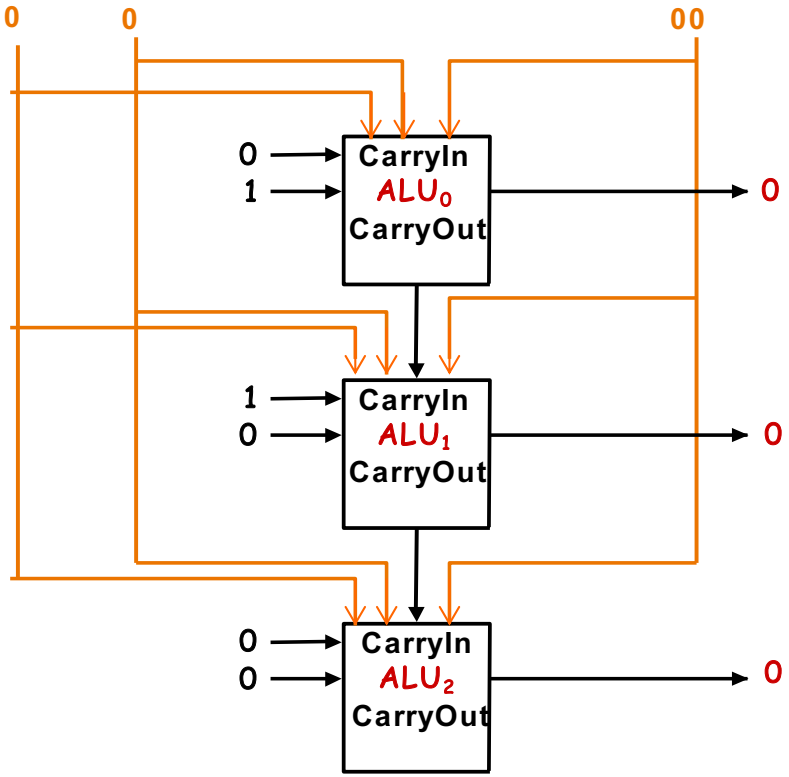


Esercizio 4.a

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione AND (010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.a: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=AND(a,b)=000

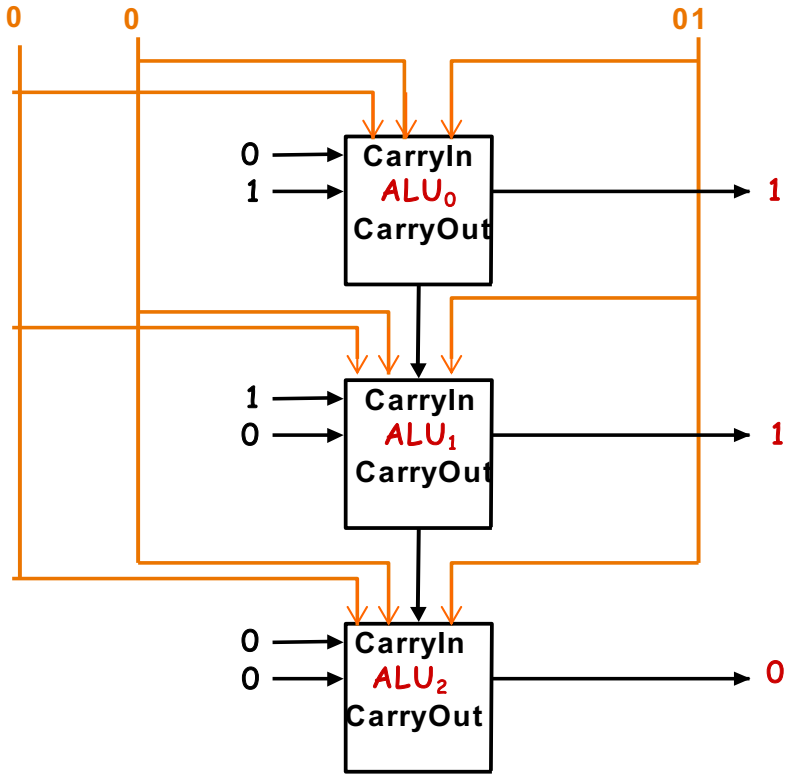


Esercizio 4.b

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione OR (010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.b: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=OR(a,b)=011

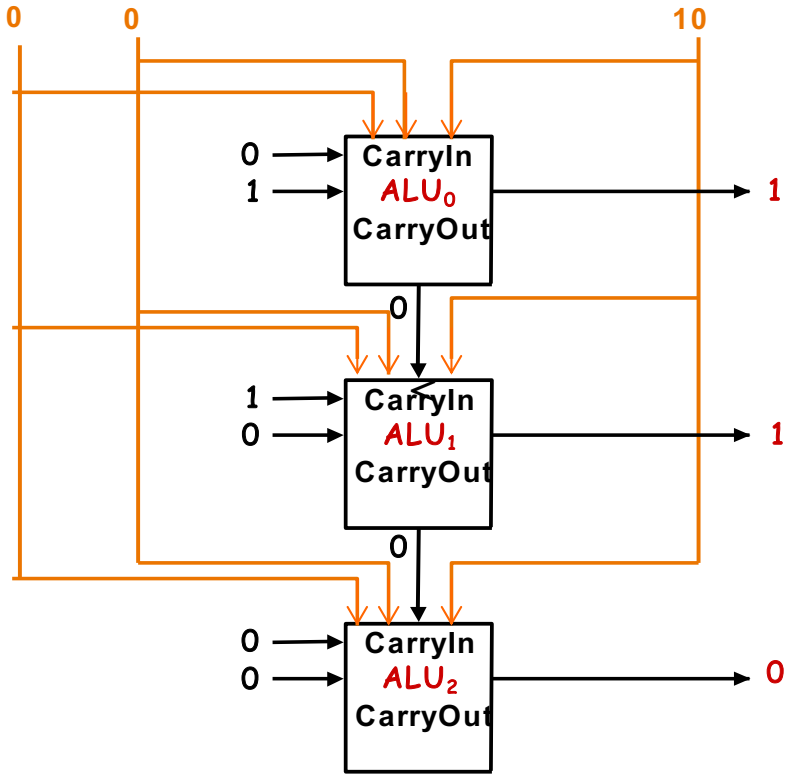


Esercizio 4.c

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione ADD(010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.c: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=ADD(a,b)=011

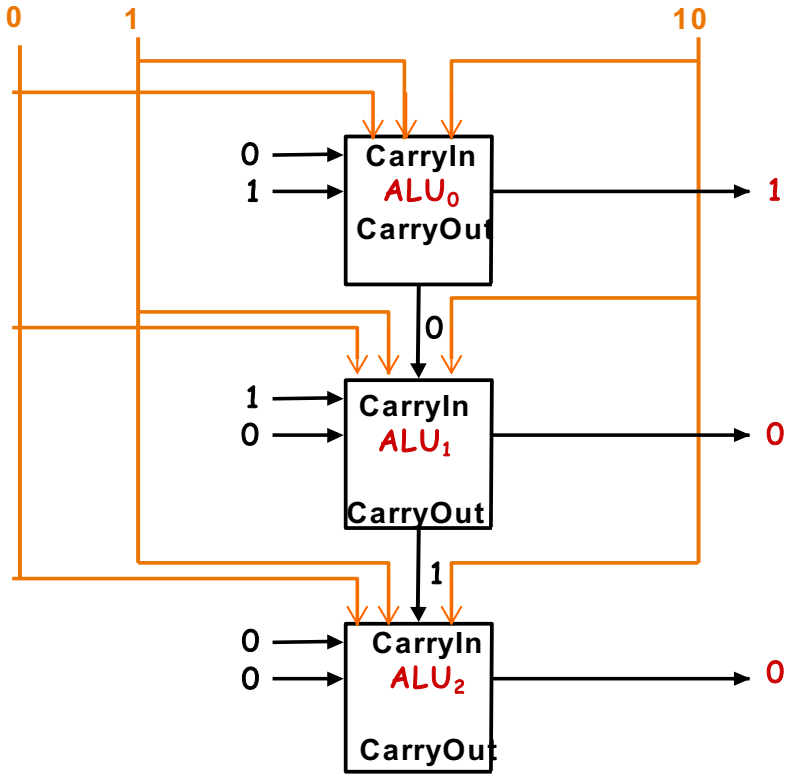


Esercizio 4.d

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione SUB(010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.d: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=SUB(a,b)=001
    - Infatti -b=110+1=111 e a+(-b)=010+111=001

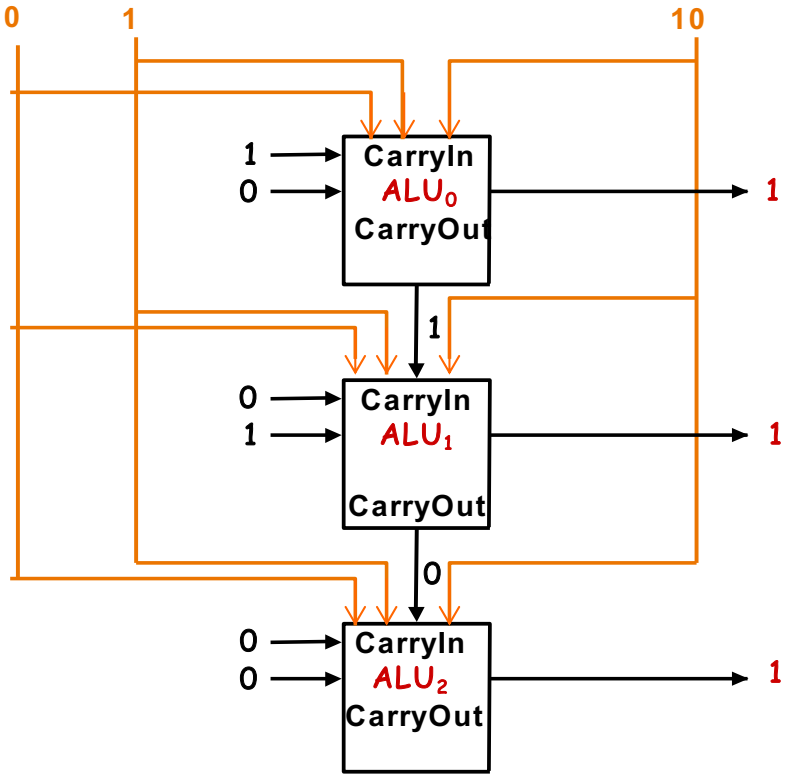


Esercizio 4.e

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione SUB(001,010), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.e: Soluzione

- Input:
  - a=001
  - b=010
- Output:
  - Result=SUB(a,b)=111
    - Infatti -b=101+1=110 e a+(-b)=001+110=111

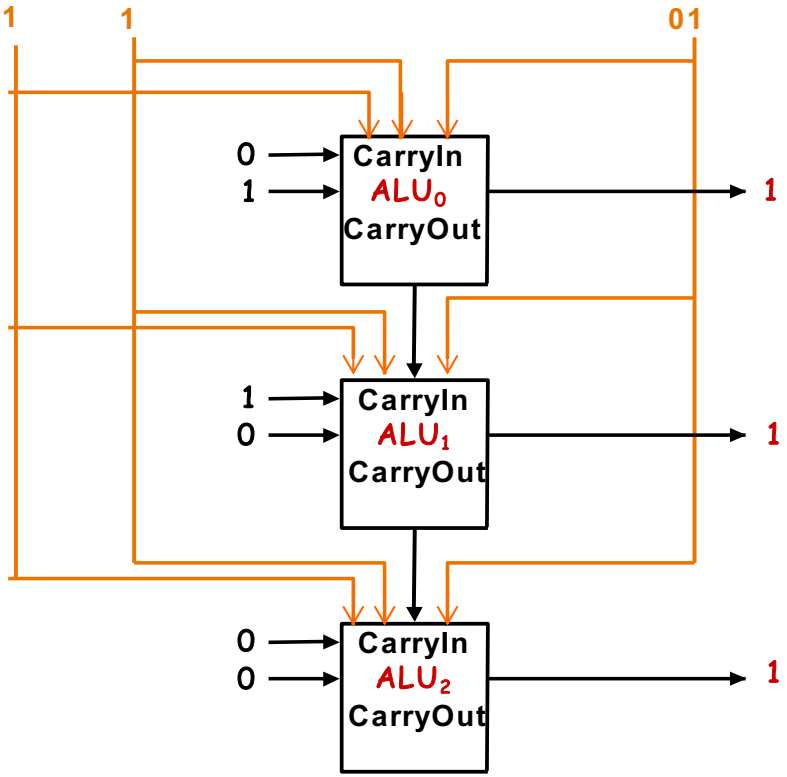


Esercizio 4.f

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione NAND (010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.f: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=NAND(a,b)=111

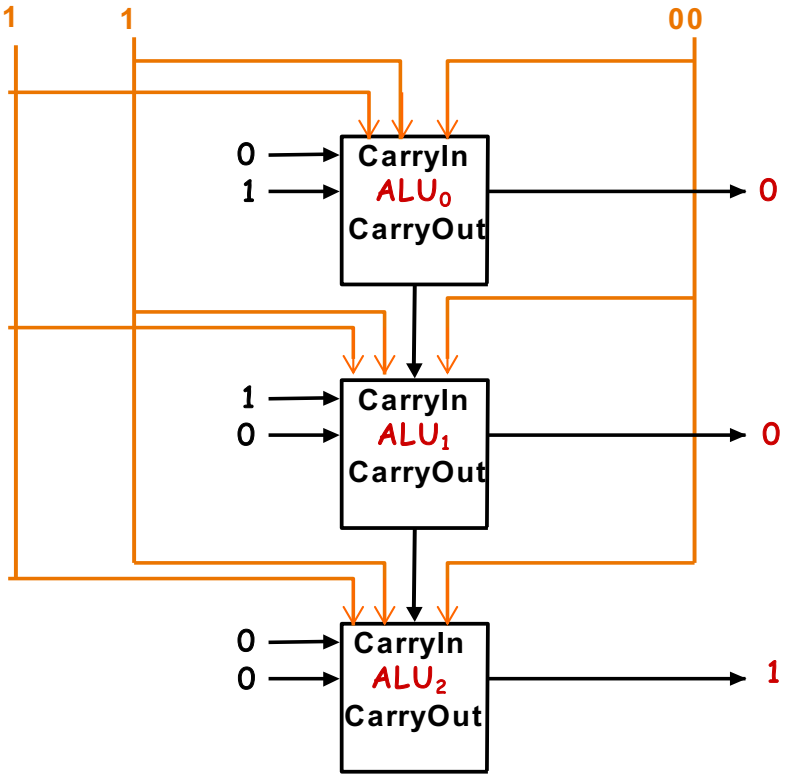


Esercizio 4.g

- Sullo schema progettato all'Esercizio 4, mostrare il risultato dell'operazione NOR (010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.g: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=NOR(a,b)=100



Esercizio 4.h

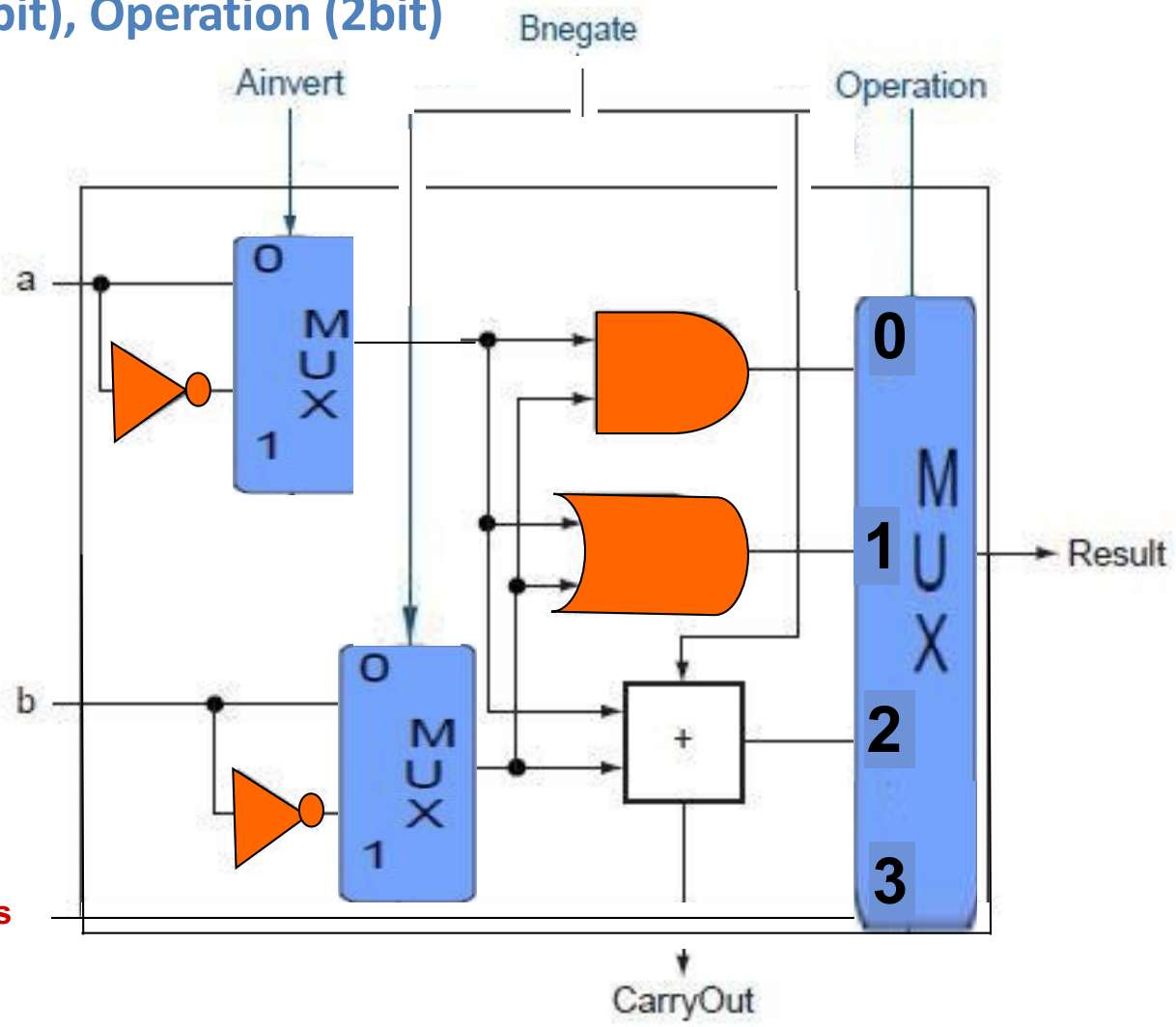
- Modificare l'ALU a 3 bit progettata, in modo da supportare anche l'operazione SLT, specificando l'opportuno segnali di controllo

Esercizio 4.h: Soluzione

Ainvert (1 bit), Bnegate (1 bit), Operation (2bit)

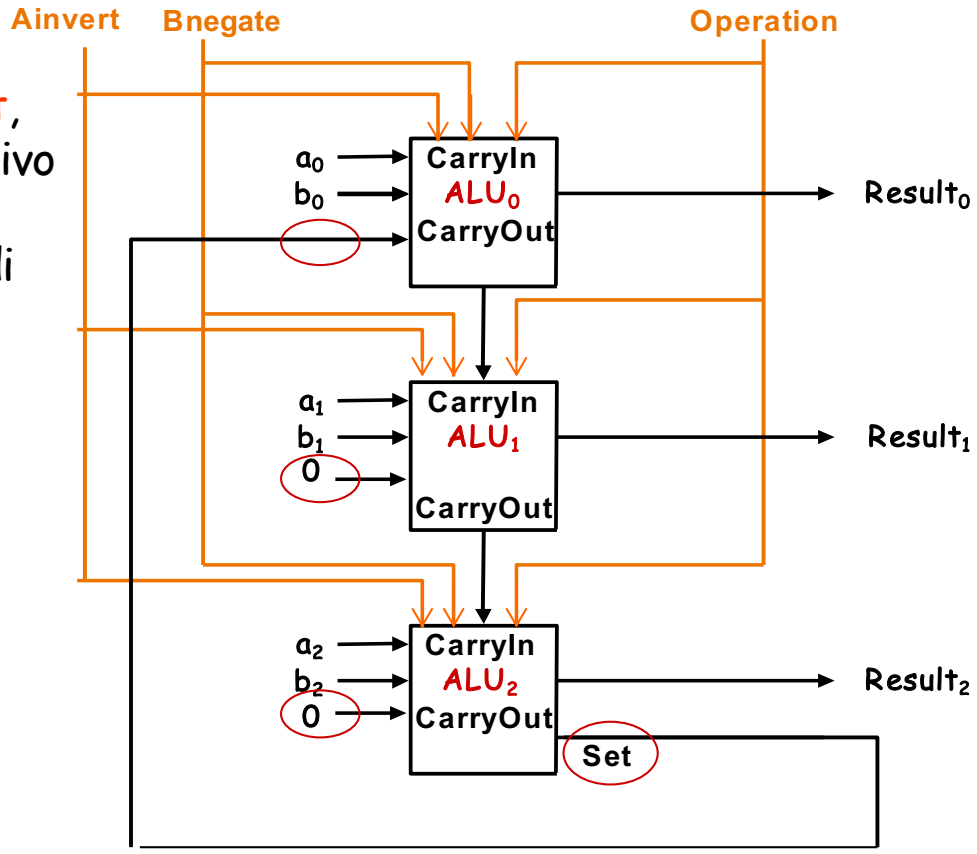
ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
1100	NOR
1101	NAND
0111	SLT

Aggiungiamo un nuovo ingresso all'ALU: Less e una riga alla tabella dei segnali di controllo



Esercizio 4.h: Soluzione

- L'ALU_2 ha un output aggiuntivo: Set, che coincide con il bit più significativo del risultato della sottrazione
- Set viene ridiretto sull'input Less di ALU_0
- Gli input Less delle varie ALU (eccetto la prima) sono posti a 0
- Bnegate è posto a 1 per sottrarre
- Operation è posta a 11 per far passare in output l'ultimo bit in ingresso al multiplexer 4:1

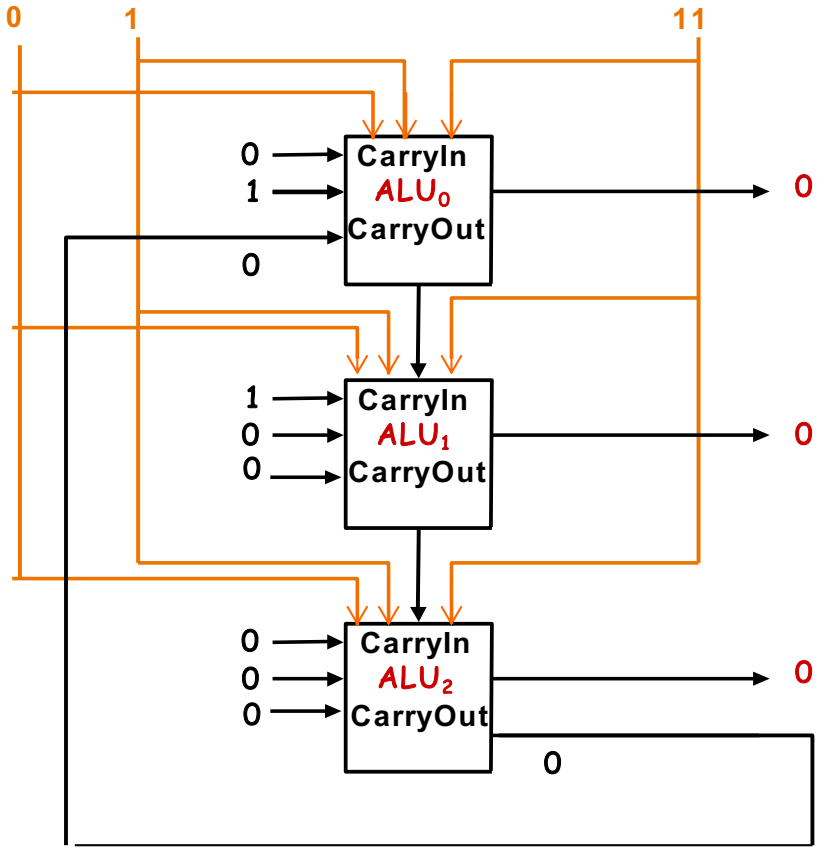


Esercizio 4.i

- Sullo schema progettato all'Esercizio 4.h, mostrare il risultato dell'operazione SLT (010,001), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.i: Soluzione

- Input:
  - a=010
  - b=001
- Output:
  - Result=SLT(a,b)=000
    - Infatti, poiché a-b=001, Set=0



Esercizio 4.j

- Sullo schema progettato all'Esercizio 4.h, mostrare il risultato dell'operazione SLT (001,010), indicando il valore
  - dei bit trasportati su ogni filo del circuito
  - dei segnali di controllo

Esercizio 4.j: Soluzione

- Input:
  - a=001
  - b=010
- Output:
  - Result=SLT(a,b)=001
    - Infatti, poiché a-b=111, Set=1