

**Modulo:** Approfondimenti sui Sistemi Aritmetici  
di un computer: tipo intero [P2\_02]

**Unità didattica:** Operazioni aritmetiche elementari [2-AT]

**Titolo:** Operazioni aritmetiche su interi e algoritmi per implementarle

Argomenti trattati:

- ✓ Tavole di definizione delle operazioni aritmetiche
- ✓ Operatori binari (Algebra di Boole)
- ✓ Algoritmi per l'addizione e la sottrazione aritmetiche
- ✓ Addizione algebrica binaria mediante operatori bitwise

**Prerequisiti richiesti:** sistemi di numerazione, aritmetica binaria

# Operazioni aritmetiche

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0(*)
2	2	3	4	5	6	7	8	9	0(*)	1(*)
3	3	4	5	6	7	8	9	0(*)	1(*)	2(*)
4	4	5	6	7	8	9	0(*)	1(*)	2(*)	3(*)
5	5	6	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)
6	6	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)
7	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)
8	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)	7(*)
9	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)	7(*)	8(*)

base 10

(\*) riporto = 1

# Esempio base 10

$$\begin{array}{r}
 1936.27 + \\
 234.56 = \\
 \hline
 2170.83
 \end{array}$$

$$\begin{array}{r}
 1936.27 + \\
 234.56 = \\
 \hline
 2170.83
 \end{array}$$

riporto = 1

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0(*)
2	2	3	4	5	6	7	8	9	0(*)	1(*)
3	3	4	5	6	7	8	9	0(*)	1(*)	2(*)
4	4	5	6	7	8	9	0(*)	1(*)	2(*)	3(*)
5	5	6	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)
6	6	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)
7	7	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)
8	8	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)	7(*)
9	9	0(*)	1(*)	2(*)	3(*)	4(*)	5(*)	6(*)	7(*)	8(*)

$$\begin{array}{r}
 1936.27 + \\
 234.56 = \\
 \hline
 2170.83
 \end{array}$$

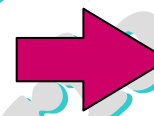
# Come eseguire l'addizione di due numeri binari?

Operando sul singolo bit!

$$\begin{array}{r} \text{op1} \quad 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ + \\ \text{op2} \quad 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ = \\ \hline 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \end{array}$$

<b>+</b>	0	1
0	0	1
1	1	0 <sup>r</sup>

<b>r</b>	0	1
0	0	0
1	0	1



$$\begin{array}{r} \text{op1} \quad 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ + \\ \text{op2} \quad 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ = \\ \hline \quad \quad \quad 1 \end{array}$$

$$\begin{array}{r} 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ + \\ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ = \\ \hline \quad \quad \quad 0^r \end{array}$$

$$\begin{array}{r} \text{op1} \quad 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ + \\ \text{op2} \quad 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ = \\ \hline \quad \quad \quad 1^r \end{array}$$

... etc ...

# Addizione aritmetica mediante operatori binari

	1	1	1	1	0	1	0	(*)
op1	0	1	0	1	1	0	1	+
op2	0	0	1	1	1	0	1	=
<hr/>								
	1	0	0	1	0	1	0	

(\*) **riporto** = 1

<b>+</b>	0	1
0	0	1
1	1	0(*)

base 2

XOR

<b>r</b>	0	1
0	0	0
1	0	1

AND

# operatori dell'Algebra Booleana

<b>A</b>	<b>NOT</b> ( $\neg A$ )
<b>0</b> (o F)	<b>1</b> (o V)
<b>1</b> (o V)	<b>0</b> (o F)

<b>A</b>	<b>B</b>	<b>OR</b> ( $A \vee B$ )
<b>0</b> (o F)	<b>0</b> (o F)	<b>0</b> (o F)
<b>0</b> (o F)	<b>1</b> (o V)	<b>1</b> (o V)
<b>1</b> (o V)	<b>0</b> (o F)	<b>1</b> (o V)
<b>1</b> (o V)	<b>1</b> (o V)	<b>1</b> (o V)

<b>A</b>	<b>B</b>	<b>AND</b> ( $A \wedge B$ )
<b>0</b> (o F)	<b>0</b> (o F)	<b>0</b> (o F)
<b>0</b> (o F)	<b>1</b> (o V)	<b>0</b> (o F)
<b>1</b> (o V)	<b>0</b> (o F)	<b>0</b> (o F)
<b>1</b> (o V)	<b>1</b> (o V)	<b>1</b> (o V)

Solo 2 operatori sono necessari! Esempio: dalle relazioni di De Morgan si ottiene la tavola di **AND**

<b>A</b>	<b>B</b>	<b><math>\neg A</math></b>	<b><math>\neg B</math></b>	<b><math>\neg A \vee \neg B</math></b>	<b><math>\neg(\neg A \vee \neg B)</math></b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>



## operatori su singoli bit

NOT	
0	1
1	0

OR	0	1
0	0	1
1	1	1

AND	0	1
0	0	0
1	0	1

XOR	0	1
0	0	1
1	1	0

NOR	0	1
0	1	0
1	0	0

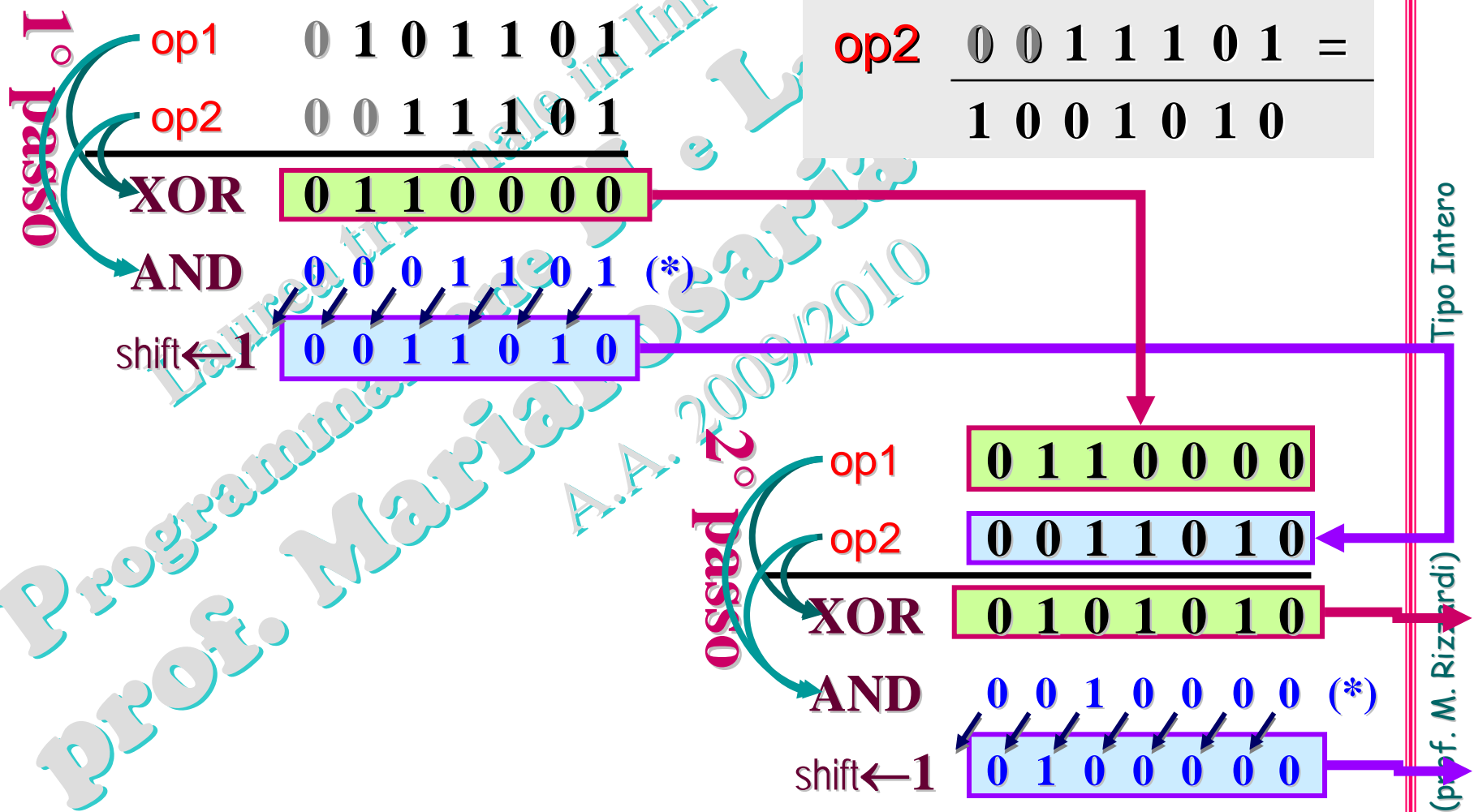
NAND	0	1
0	1	1
1	1	0



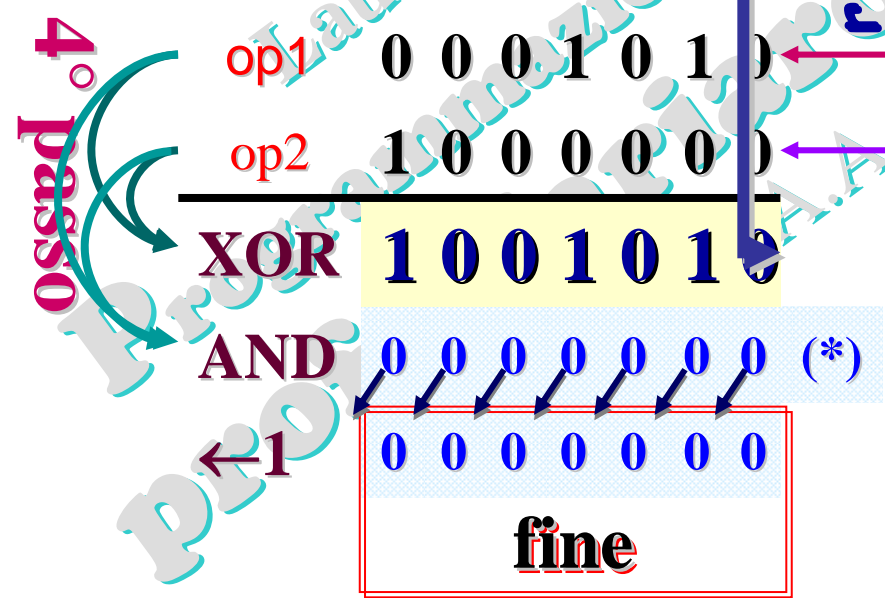
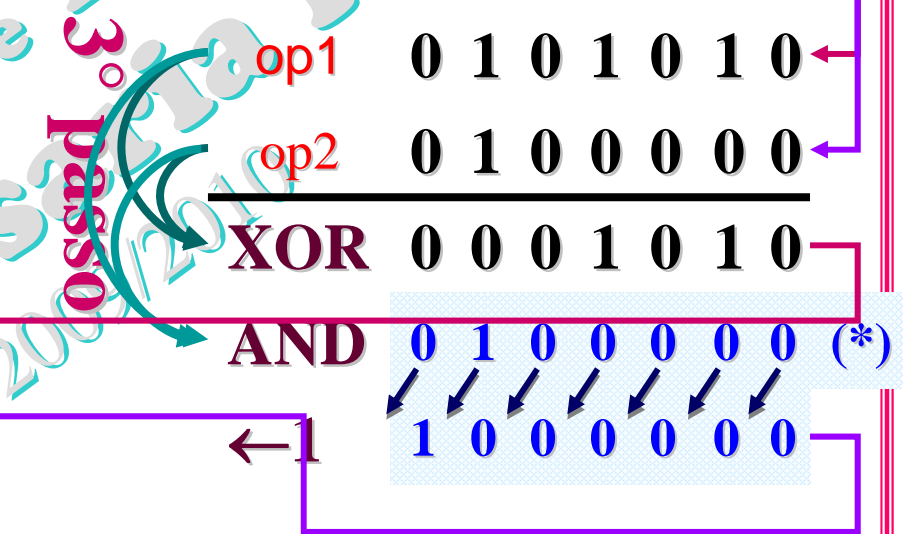
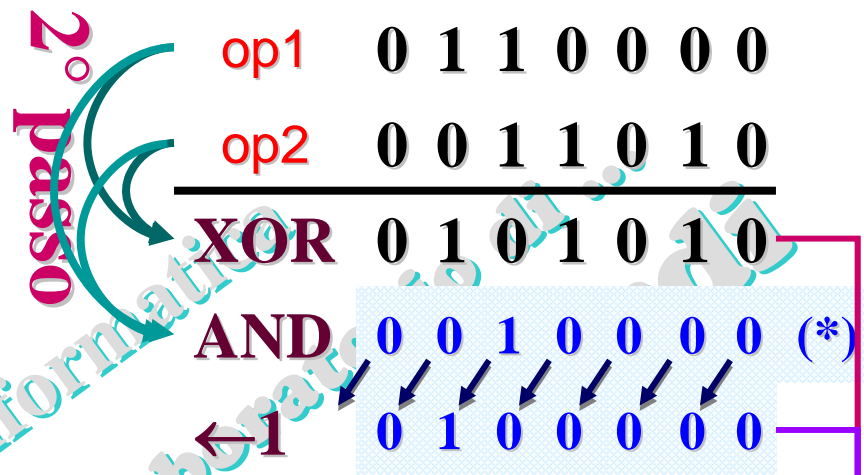
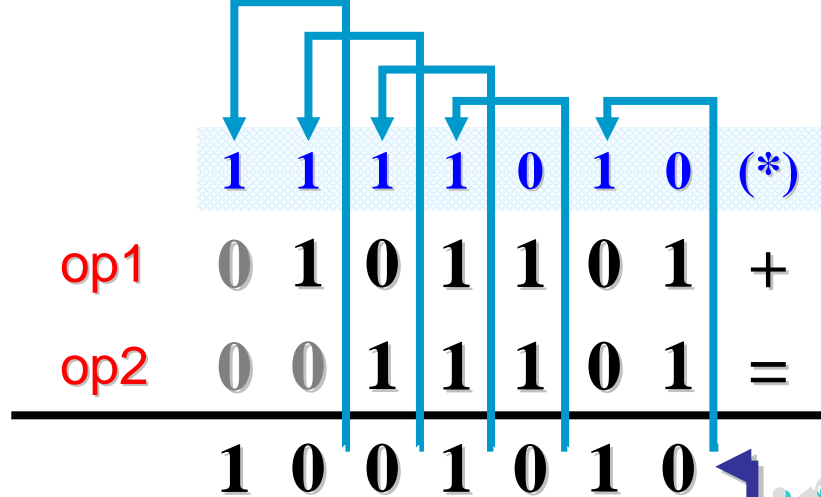
# Altro algoritmo per l'addizione di due numeri binari

## Addizione aritmetica mediante operatori bitwise del C

op1	0	1	0	1	1	0	1	+
op2	0	0	1	1	1	0	1	=
<hr/>								
	1	0	0	1	0	1	0	







**bitAND**  
+  
**leftSHIFT**

*op1*

*op2*

$$\begin{array}{r}
 \text{(*)} \quad 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \\
 \text{op1} \quad 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ + \\
 \text{op2} \quad 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ = \\
 \hline
 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0
 \end{array}$$

**bitXOR**

(\*) **riporto = 1**

<b>+</b>	0	1
0	0	1
1	1	0(*)

**base 2**

## Idea algoritmo *addizione binaria*

```

rip:=1;
while rip>0
    sum:=bitXOR(op1,op2);
    rip:=bitAND(op1,op2);
    rip:=leftSHIFT(rip,1);
    op1:=sum; op2:=rip;
endwhile
    
```

## Analogamente per la sottrazione di due numeri binari

$$\begin{array}{r}
 \text{op1} \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad - \\
 \text{op2} \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad = \\
 \hline
 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0
 \end{array}$$

Diagram showing the subtraction of two 6-bit binary numbers. The result is 001100. A blue box above the result shows the borrow propagation: -1 -1 (\*). Arrows indicate the borrow path from the 6th bit to the 5th bit, and from the 5th bit to the 4th bit.

(\*) prestito = -1

-	0	1
0	0	1(*)
1	1	0

base 2

XOR

p	0	1
0	0	1
1	0	0

?

? prestito = -1

p	0	1
0	0	1
1	0	0

**NOT**

0

1

1

0

**NOT**(op2)

**OR**

1

0

0

1

0

1

1

1

**OR**(op1, **NOT**(op2))

**NOT**

0

1

0

0

**NOT**(**OR**(op1, **NOT**(op2)))

... oppure anche ....

**? prestito = -1**

<b>p</b>	<b>0</b>	<b>1</b>
<b>0</b>	0	1
<b>1</b>	0	0

<b>NOT</b>	
<b>0</b>	1
<b>1</b>	0

**NOT (op1)**

<b>AND</b>	<b>0</b>	<b>1</b>
<b>1</b>	0	1
<b>0</b>	0	0

**AND (NOT (op1) , op2)**

# Esercizi

1

Scrivere una *function* C per eseguire l'addizione aritmetica (Versione 1) in base 2 mediante gli operatori bitwise.

**Idea algoritmo addizione binaria**

```
rip:=1;
while rip>0
    sum:=bitXOR(op1,op2);
    rip:=bitAND(op1,op2);
    rip:=leftSHIFT(rip,1);
    op1:=sum; op2:=rip;
endwhile
```

2

Scrivere una *function* C per eseguire la sottrazione aritmetica(\*) in base 2 mediante gli operatori bitwise.

(\*) cioè primo operando maggiore del secondo