



Firmware Division

Prof. Alberto Borghese
Dipartimento di Informatica
borgese@di.unimi.it

Università degli Studi di Milano

Riferimenti sul Patterson 5a ed.: 3.4, 3.5



Sommario

- **Divisione intera**
- Circuiti divisione intera
- Divisione e moltiplicazione



Moltiplicazione



128 x		
214 =	128x(4x10 ⁰)+	unità
-----	128x(1x10 ¹)+	decine
512 + x10 ⁰	128x(2x10 ²) =	centinaia
128- + x10 ¹	27392	
256-- = x10 ²		

27392		

Prodotto come **somma dei prodotti parziali**. Ognuno dei prodotti parziali aggiunge il **moltiplicando pesato con il peso della cifra analizzata del moltiplicatore** x un numero di volte pari alla cifra analizzata del moltiplicatore.

Nel caso binario le cifre sono solo 0,1 ==>



Algoritmi per la moltiplicazione



Il razionale degli algoritmi firmware della moltiplicazione è il seguente.

Moltiplicando	1 1 0 1 1 x
Moltiplicatore	1 0 1 =

Si analizzano sequenzialmente i bit del moltiplicatore e:

1) Si mette 0 nella posizione opportuna (se il bit analizzato del moltiplicatore = 0).

2) Si mette una copia del moltiplicando nella posizione opportuna (se il bit analizzato del moltiplicatore = 1).

	1 1 0 1 1 +
	0 0 0 0 0 -
	1 1 0 1 1
	1 1 0 1 1 - -
	1 0 0 0 1 1 1

Prodotto

La moltiplicazione viene effettuata come somme successive, con peso crescente, di uno tra i 2 valori: {moltiplicando, 0}



La divisione decimale



Dividendo Divisore

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ 36\text{--} \\ \text{-----} \\ 11\text{-} \\ 0\text{-} \\ \text{-----} \\ 116 \\ 108 \\ \text{-----} \\ 8 \end{array}$$

Quoziente

Resto

Dividendo = Divisore * Quoziente (quoto) + Resto

A.A. 2023-2024

5/74

<http://borghese.di.unimi.it/>



Razionale della divisione



1. Analizzo il dividendo

2. Identifico quale potenza di 10 iniziare a considerare. Questa è la **massima potenza, p**, tale per cui 10^p moltiplicata per il divisore mi dà un valore inferiore al dividendo.

In questo caso:

$$p = 1 \rightarrow 10 * 12 = 120$$

$$p = 2 \rightarrow 100 * 12 = 1200$$

$$p = 3 \rightarrow 1000 * 12 = 12000.$$

3. Determino qual è il **numero massimo di volte, q₂**, per cui il divisore x 10^p può essere moltiplicato per ottenere un numero inferiore al dividendo.

In questo caso:

$$q_2 = 1 \rightarrow 1 * 1200 = 1200$$

$$q_2 = 2 \rightarrow 2 * 1200 = 2400$$

$$q_2 = 3 \rightarrow 3 * 1200 = 3600$$

$$q_2 = 4 \rightarrow 4 * 1200 = 4800$$

NB la moltiplicazione x 10^p si sottoindende, allineando il q_2 x divisore alla cifra opportune.
In questo caso $3x12 = 36$ allineato alle centinaia.

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ 36\text{--} \\ \text{-----} \\ 11\text{-} \\ 0\text{-} \\ \text{-----} \\ 116 \\ 108 \\ \text{-----} \\ 8 \end{array}$$

i.it\



Razionale della divisione

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ 3600 \\ \text{-----} \\ 116 \\ 0000 \\ \text{-----} \\ 116 \\ 108 \\ \text{-----} \\ 8 \end{array}$$

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ 36\text{--} \quad \text{Centinaia} \\ \text{-----} \\ 11\text{-} \\ 0\text{-} \quad \text{Decine} \\ \text{-----} \\ 116 \\ 108 \quad \text{Unità} \\ \text{-----} \\ 8 \end{array}$$

4. Ho stabilito $p = 2$, $q_2 = 3 \rightarrow$ posso erodere una quantità pari a: $3 * 12 * 10^2 = 3600$

$$\begin{array}{r} 3716 - \\ 3600 = \\ \text{-----} \\ 116 \end{array} \quad \begin{array}{l} q_2 = 3 \text{ è la cifra del quoziente cioè di quanto stiamo erodendo} \\ p = 2 \rightarrow 10^2 \text{ è il peso del divisore} \\ \\ \text{Resto parziale} \end{array}$$

Procedo ricorsivamente ripetendo i passi 2-4 sul resto fino a quando non ottengo un resto inferiore al divisore.

I passi successive al primo sono leggermente diversi.



Sviluppo della divisione - I

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ 36\text{--} \\ \text{-----} \\ 11\text{-} \\ 0\text{-} \\ \text{-----} \\ 116 \\ 108 \\ \text{-----} \\ 8 \end{array}$$

1. Analizzo il dividendo

2. Considero la potenza $p = \mathbf{p-1}$. La potenza 1 in questo caso.

$$p = 1 \rightarrow 10^1 * 12 = 120$$

3. Determino qual è il **numero massimo di volte**, q_1 , per cui il divisore $\times 10^p$ può essere moltiplicato per ottenere un numero inferiore al dividendo.

In questo caso:

$$q_1 = 0 \rightarrow 0 * 120 = 0$$

$$q_1 = 1 \rightarrow 1 * 120 = 120$$

4. Ho stabilito $p = 1$, $q_1 = 0 \rightarrow$ posso erodere una quantità pari a: $0 * 12 * 10^1 = 0$

$$\begin{array}{r} 116 - \\ 0 - = \\ \text{-----} \\ 116 \end{array} \quad \begin{array}{l} q_1 = 0 \text{ è la cifra del quoziente cioè di quanto stiamo erodendo} \\ p = 1 \rightarrow 10^1 \text{ è il peso del divisore} \\ \\ \text{Resto parziale} \end{array}$$



Sviluppo della divisione - II

$$\begin{array}{r} \text{---} \\ 3716 : 12 = 309 \\ \text{---} \\ 36\text{--} \\ \text{-----} \\ 11\text{-} \\ 0\text{-} \\ \text{-----} \\ 116 \\ 108 \\ \text{-----} \\ 8 \end{array}$$

1. Analizzo il moltiplicatore
2. Considero la potenza $p=p-1$, **0 in questo caso.**
 $p = 0 \rightarrow 1 * 12 = 12$
3. Determino qual è il **numero massimo di volte, q_0** , per cui il divisore $\times 10^p$ può essere moltiplicato per ottenere un numero inferiore al dividendo.
In questo caso:
 $q_0 = 8 \rightarrow 8 * 12 = 96$
 $q_0 = 9 \rightarrow 9 * 12 = 108$
4. Ho stabilito $p = 0$, $q_0 = 9 \rightarrow$ posso erodere una quantità pari a: **$9 * 12 * 10^0 = 108$**

$$\begin{array}{r} 116\text{-} \\ 108\text{=} \\ \text{----} \\ 8 \end{array}$$

$q_0 = 9$ è la cifra del quoziente cioè di quanto stiamo erodendo
 $p = 0 \rightarrow 10^0$ è il peso del divisore

Resto totale



Razionale della divisione decimale

La divisione è l'operazione inversa del prodotto

$$2629 : 12 = 219 \quad (\text{resto} = 1) \quad \rightarrow \quad 219 \times 12 + 1 = 2629$$

$$[(2 \times 10^2) + (1 \times 10^1) + (9 \times 10^0)] \times 12 + 1 = 2629$$

Da cui segue che ad ogni passo della divisione erodiamo (**sottraiamo**) una quantità via via decrescente. Al primo passo sottraiamo a entrambi i membri:

$$1. \quad [(1 \times 10^1) + (9 \times 10^0)] \times 12 + 1 = 2629 - (2 \times 10^2) \times 12 = 2629 - 2400 = 229 \rightarrow \text{resto parziale.}$$

Abbiamo cioè eroso 24 centinaia al dividendo 2629.

$$2. \quad \text{Al passo successivo eroderemo al resto parziale 12 decine:}$$

$$[(9 \times 10^0)] \times 12 + 1 = 229 - (1 \times 10^1) \times 12 = 229 - 120 = 109 \rightarrow \text{resto parziale}$$

$$3. \quad \text{E al passo finale eroderemo } 12 \times 9 = 108 \text{ unità:}$$

$$[1 = 109 - (9 \times 10^0) \times 12 = 1 \rightarrow \text{resto finale.}]$$

Ad ogni passo si cerca di erodere il più possibile!



Confronto con la moltiplicazione



La divisione è l'operazione inversa del prodotto

$$2629 : 12 = 219 \text{ (resto = 1)} \quad \rightarrow \quad 219 \times 12 + 1 = 2629$$

Divisione: Ad ogni passo erodiamo (sottraiamo) un certo numero di volte il divisore moltiplicato per la massima cifra possibile (cifra del quoziente), moltiplicato anche per la potenza associata alla posizione della cifra del dividendo, che stiamo considerando.

Opera per sottrazioni successive di quantità sempre più piccole proporzionali al divisore, producendo via via i resti parziali.

Moltiplicazione: a ogni passo sommiamo il moltiplicando moltiplicato per il numero di volte indicato dalla cifra del moltiplicatore considerata e per la potenza associata alla posizione della cifra del moltiplicatore, che stiamo considerando.

Opera per somme successive di quantità sempre più grandi proporzionali al moltiplicatore, producendo via via le somme parziali.



La divisione tra numeri binari



Divisione decimale fra i numeri $a = 1.001.010$ e $b = 1.000$ $a : b = ?$

1001010 : 1000 \Rightarrow 1001 Dividendo : Divisore $74 : 8 = 9$ resto 2

1000 -

10 ← Quoziente

0000 =

101 - ← Resti parziali

0000 =

1010 -

1000 =

10 Resto

Dividendo = Quoziente x Divisore + Resto



Confronto tra divisione tra numeri binari e decimali



In DECIMALE:

Ad ogni passo devo verificare QUANTE VOLTE il resto parziale contiene il divisore. Il risultato è un numero che va da 0 a 9 = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. 0 = il divisore non è contenuto nel resto.

In BINARIO:

Ad ogni passo devo verificare SE il resto parziale contiene il divisore. Ovverosia se lo contiene 0 o 1 volta. Il risultato è un numero che può valere {0, 1}.

$\begin{array}{r} _ _ _ \\ 3716 : 12 = 309 \\ \underline{3600} \\ 116 \\ \underline{00} \\ 116 \\ \underline{108} \\ 8 \end{array}$	$\begin{array}{r} _ _ _ _ _ _ \\ 1001010 : 1000 = 1001 \\ \underline{} \\ 1010 - \\ \underline{000000} = \\ \\ 1010 - \\ \underline{000000} = \\ \\ 1010 - \\ \underline{1000} = \\ \\ 10 \end{array}$
--	--



Peculiarità della divisione



- Come rappresento la condizione “il divisore è contenuto nel resto parziale”?

Esempi:

10 (resto parziale) non contiene 1000 (divisore)

1001 (resto parziale) contiene 1000 (divisore)

Per tentativi.

Eseguo la sottrazione.

Risultato $\geq 0 \rightarrow$ il resto parziale è maggiore del divisore.

Risultato $< 0 \rightarrow$ il resto parziale è minore del divisore (non contiene il divisore)

Osserviamo che nel secondo caso abbiamo fatto in realtà una sottrazione che non avremmo dovuto effettuare.

NB il calcolatore non può sapere se il resto parziale contiene il divisore fino a quando non ha effettuato la sottrazione.



La divisione tra numeri binari



Divisione decimale fra i numeri su 7 bit: $a = 100\ 1010$ e $b = 000\ 1000$ $a : b = ?$ $74 : 8 = ?$

Nel primo passaggio allineo il divisore alla sinistra della prima cifra (MSB) del dividendo.

Allocazione di 14 bit. Sottrazione \rightarrow somma in complemento a 2:

$$\begin{array}{r}
 000\ 0000\ 100\ 1010 - \quad 000\ 0000\ 100\ 1010 + \\
 000\ 1000\ 000\ 0000 = \quad 111\ 1000\ 000\ 0000 = \\
 \hline
 111\ 1000\ 100\ 1010 \quad 111\ 1000\ 100\ 1010 \quad (= -950_{10}) \quad \text{Nuovo resto parziale} \\
 \text{provvisorio}
 \end{array}$$

Resto parziale = dividendo - divisore * 2^7
 $= 74 - 8 * 2^7 = 74 - 1024 = -950$

Ripristino il resto parziale precedente: 0000 000 100 1010



Note e strategia di implementazione



All'inizio il divisore è allineato alla sinistra del dividendo: le cifre del dividendo sono allineate agli 0 del divisore e gli 0 del dividendo sono allineati alle cifre del divisore.

Il divisore viene spostato verso dx di 1 bit ad ogni passo.

Il quoziente cresce dal bit più significativo verso il bit meno significativo. Cresce verso dx.

Ci sono $N+1$ passi di divisione, il primo darà sempre 0 e si potrebbe omettere.

Occorre quindi effettuare a ogni passo:

Resto parziale = Resto parziale - divisore * base corrente

Ripristino del resto parziale (se sottrazione < 0)

Shift quoziente a sx.

Scrittura di 1 o 0 nel registro quoziente.

Shift del divisore verso dx.

Utilizzo un unico registro per dividendo e resto (il resto si ottiene per erosione del dividendo).

Considero il primo resto parziale uguale al dividendo (inizializzazione).

Il primo passo sarà "a vuoto" perchè produrrà sempre quoziente 0.



Inizializzazione: Resto = [0 | Dividendo]
 Divisore = [Divisore | 0]
 Quoziente = 0
 k = 0

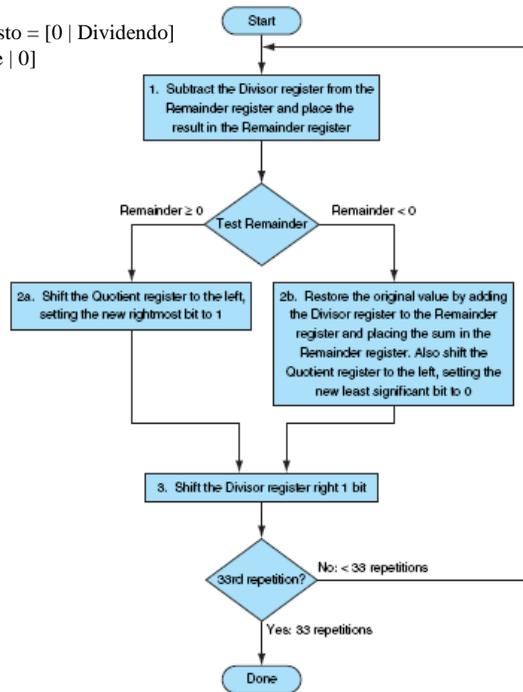


Divisione: algoritmo per 32 bit

1001010 : 1000 = 1001

```

-----
1010 -
00000 =
-----
1010 -
00000 =
-----
1010 -
1000 =
-----
10
  
```



Esempio - 1



Divisione decimale fra i numeri $a = 7$ e $b = 2$ su 4 bit. $a : b = ?$

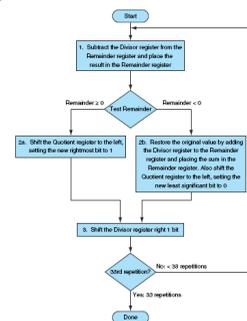
Inizializzo il divisore alla sinistra delle quattro cifre significative. La prima cifra del quoziente sarà sempre 0.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111

$$\text{Divisore} = 1x2^4x2^1 = 32$$

$$\text{Resto} = 7$$

$$7 - 32 = -25 < 0$$





Esempio - 2



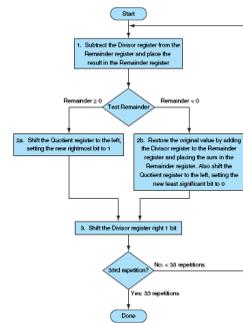
Divisione decimale fra i numeri $a = 7$ e $b = 2$ $a : b = ?$

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0100	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111

$$\text{Divisore} = 1 \times 2^3 \times 2 = 16$$

$$\text{Resto} = 7$$

$$7 - 16 = -9 < 0$$



Esempio - 3



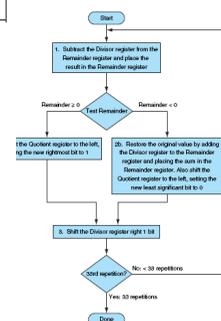
Divisione decimale fra i numeri $a = 7$ e $b = 2$ $a : b = ?$

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1: Rem = Rem - Div	0000	0000 1000	0111 1111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111

$$\text{Divisore} = 1 \times 2^2 \times 2 = 8$$

$$\text{Resto} = 7$$

$$7 - 8 = -1 < 0$$





Esempio - 4

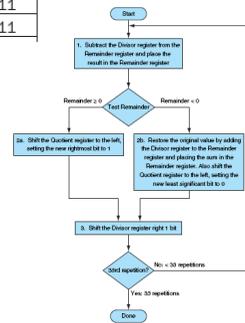


Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	1110 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	1111 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1: Rem = Rem - Div	0000	0000 1000	1111 1111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111
4	1: Rem = Rem - Div	0000	0000 0100	0000 0011
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011

$$\text{Divisore} = 1 \times 2^1 \times 2 = 4$$

$$\text{Resto} = 7$$

$$7 - 4 = +3 > 0 \text{ nuovo resto}$$



Esempio - 5

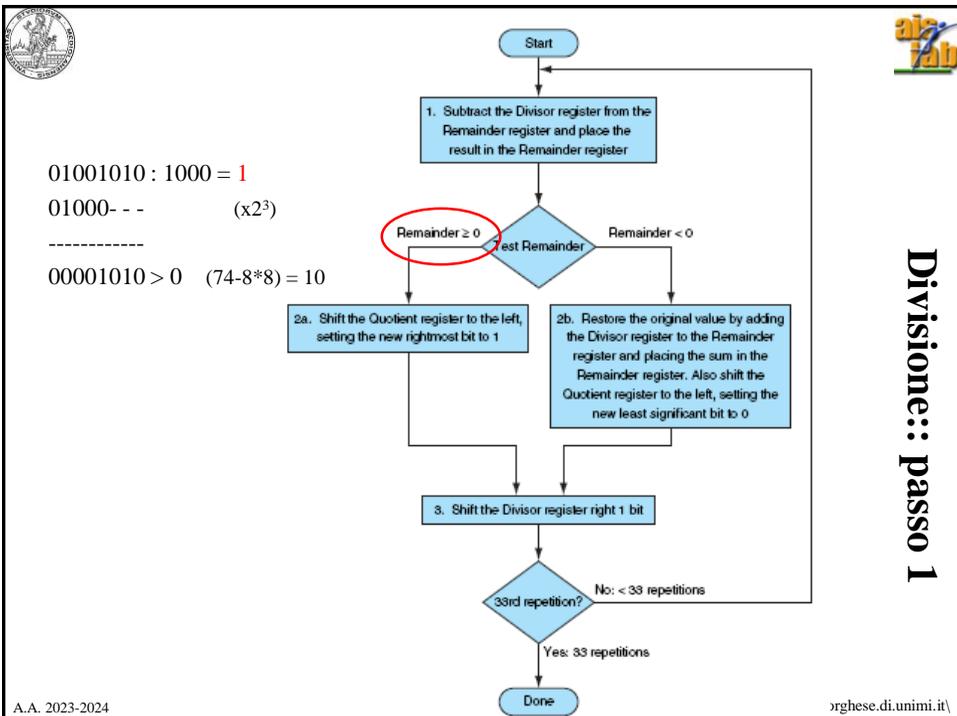
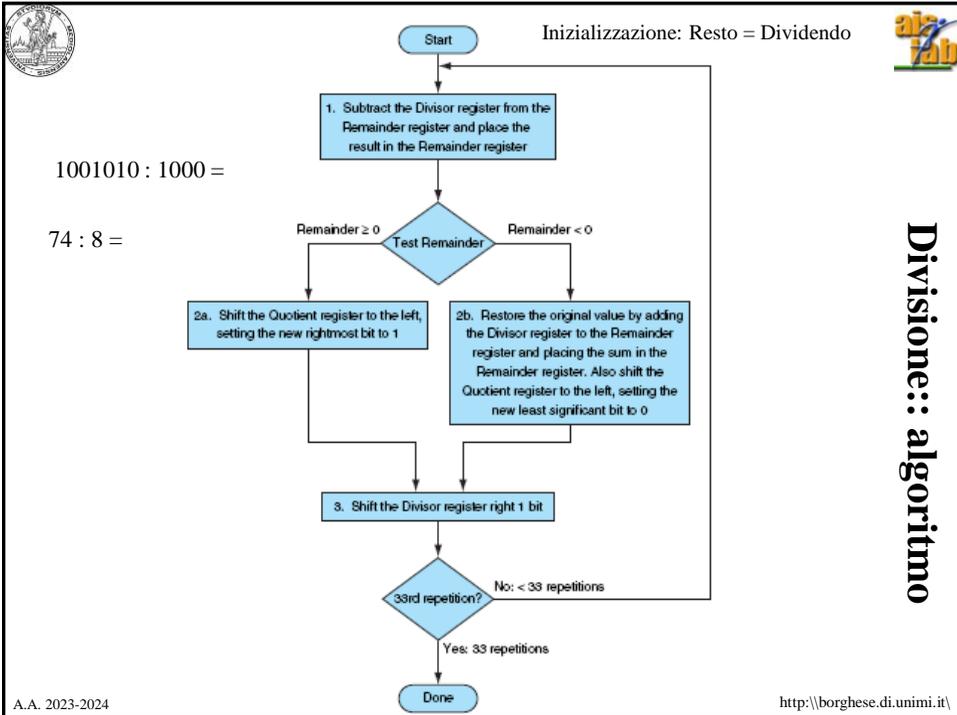


Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	1110 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	1111 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1: Rem = Rem - Div	0000	0000 1000	1111 1111
	2b: Rem < 0 ⇒ +Div, sll Q, Q0 = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111
4	1: Rem = Rem - Div	0000	0000 0100	0000 0011
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011
5	1: Rem = Rem - Div	0001	0000 0010	0000 0001
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

$$\text{Divisore} = 1 \times 2^0 \times 2 = 2$$

$$\text{Resto} = 3$$

$$3 - 2 = +1 > 0 \text{ nuovo resto} = \text{resto finale}$$





$$01001010 : 1000 = 10$$

$$01000 \quad (x2^3)$$

$$00001010 - \quad (74-8*8) = 10 > 0$$

$$- -1000 - - \quad (x2^2)$$

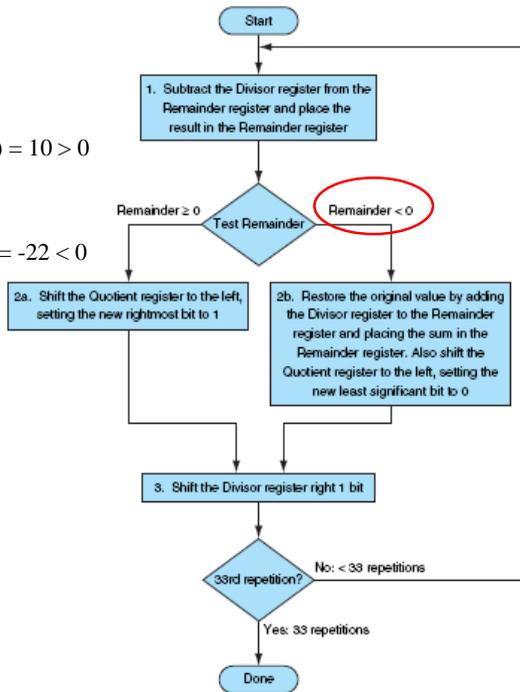
$$11101011 + \quad (10-8*4) = -22 < 0$$

$$- -1000 - -$$

$$00001010$$



Divisione:: passo 2



$$01001010 : 1000 = 100$$

$$01000 \quad (x2^3)$$

$$00001010 - \quad (74-8*8) = 10 > 0$$

$$- -1000 - - \quad (x2^2)$$

$$11101011 + \quad (10-8*4) = -22 < 0$$

$$- -1000 - - \quad (x2^2)$$

$$00001010 - \quad 10$$

$$- - -1000 - \quad (x2^1)$$

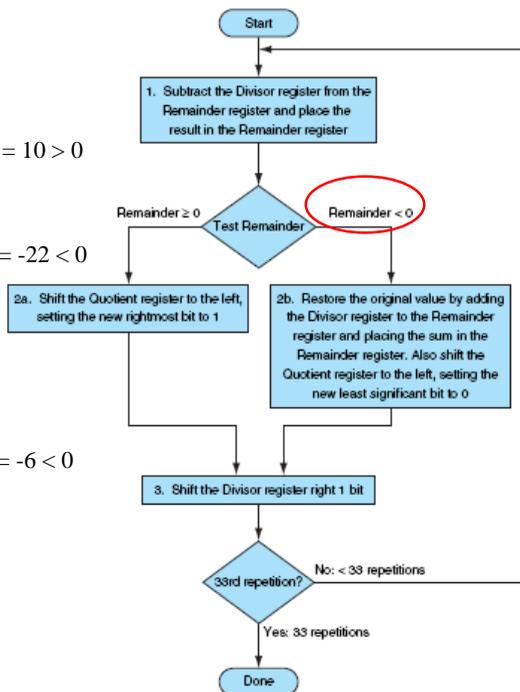
$$11110110 \quad (10-8*2) = -6 < 0$$

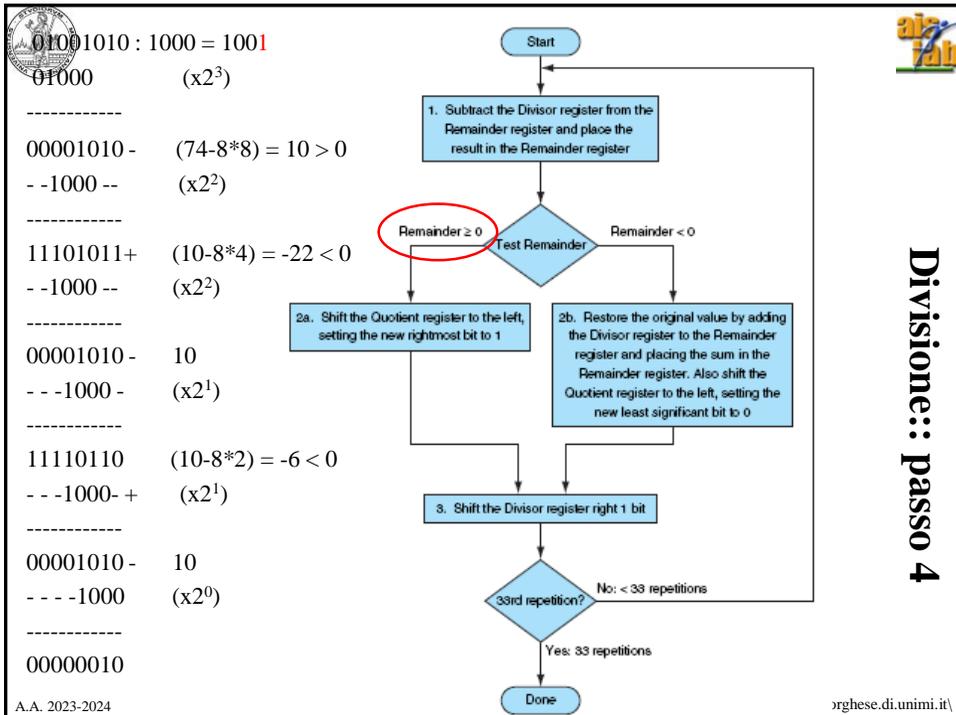
$$- - -1000 + \quad (x2^1)$$

$$00001010 \quad 10$$



Divisione:: passo 3





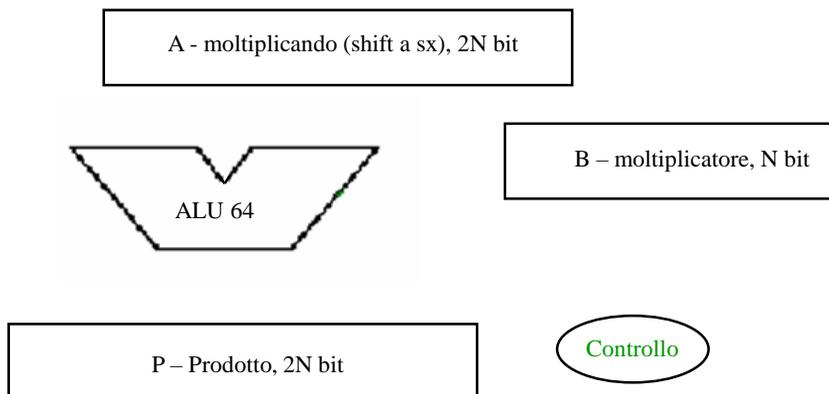
Sommaro

- Divisione intera
- **Circuiti divisione intera**
- Divisione e moltiplicazione

A.A. 2023-2024 http://\borghese.di.unimi.it



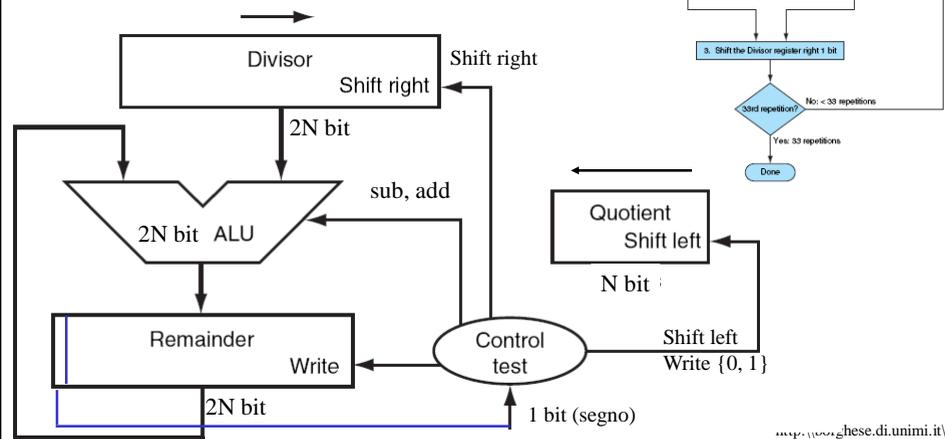
Implementazione circuitale gli stessi attori della moltiplicazione



Il circuito firmware della divisione

Inizializzazione:

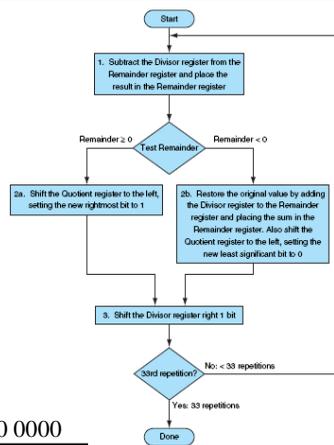
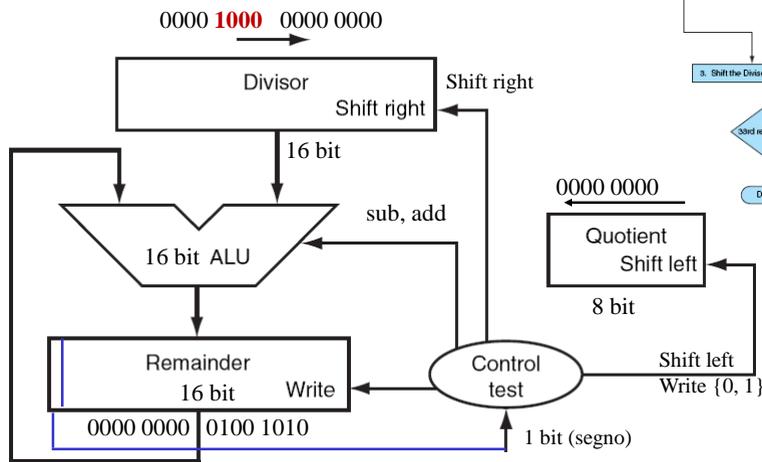
- Resto = 0 | Dividendo
- Divisore | 0





Il circuito firmware della divisione

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo $74 : 8$ su 8 bit
 -- Divisore | 0

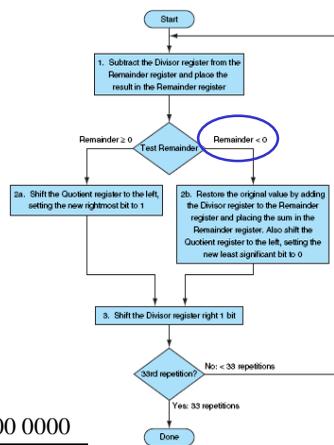
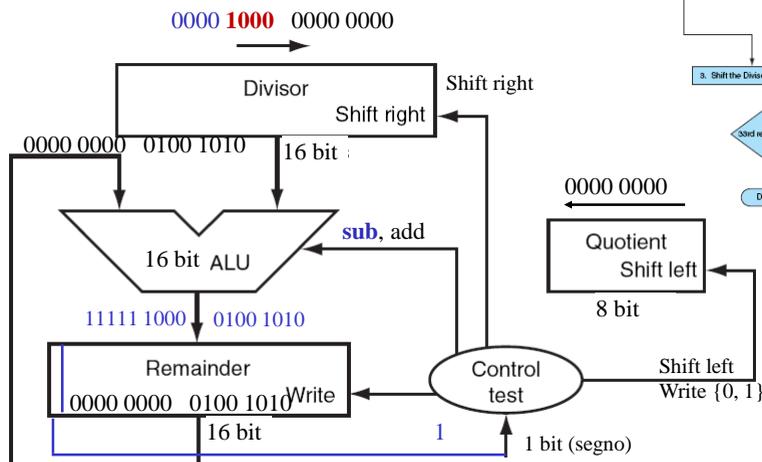


<http://www.giese.di.unimi.it/>



Il circuito firmware della divisione – passo 1a

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo $74 - 8 \times 2^8 = -1974 < 0$
 -- Divisore | 0



<http://www.giese.di.unimi.it/>



Il circuito firmware della divisione – passo 1b

Inizializzazione:

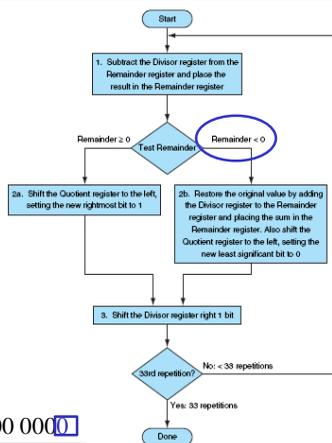
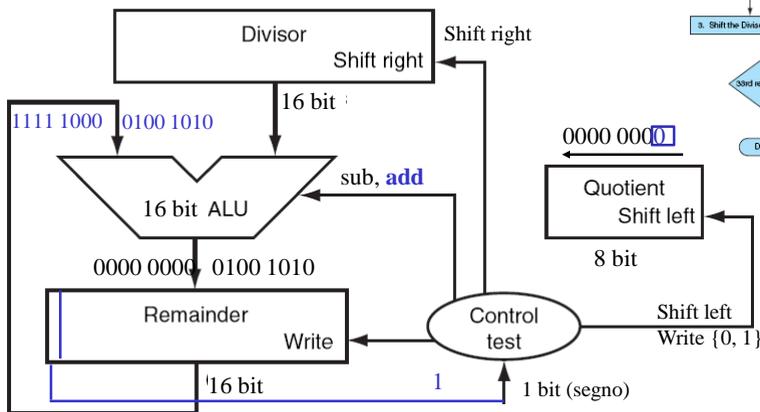
-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$-1974 + 8 \times 2^8 = +74 -- Q_k = 0$$

0000 1000 0000 0000



www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 1c

Inizializzazione:

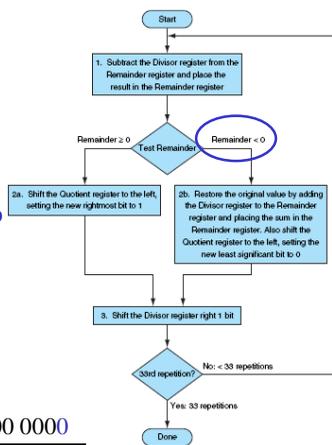
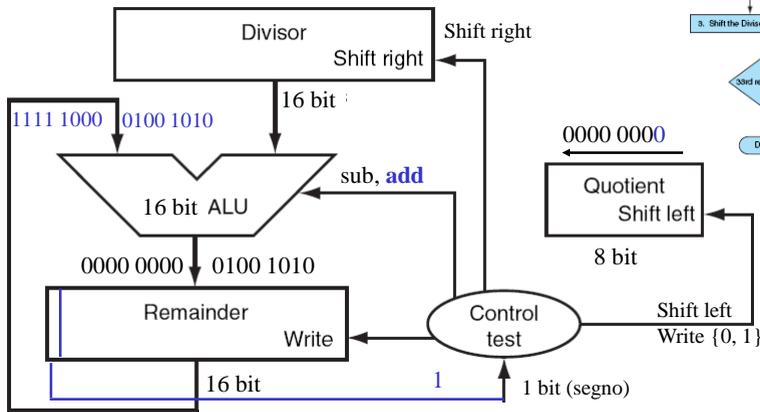
-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$\text{Divisore: } 2048_{10} \text{ ShiftDx } 1 \rightarrow 1024_{10}$$

0000 0100 0000 0000



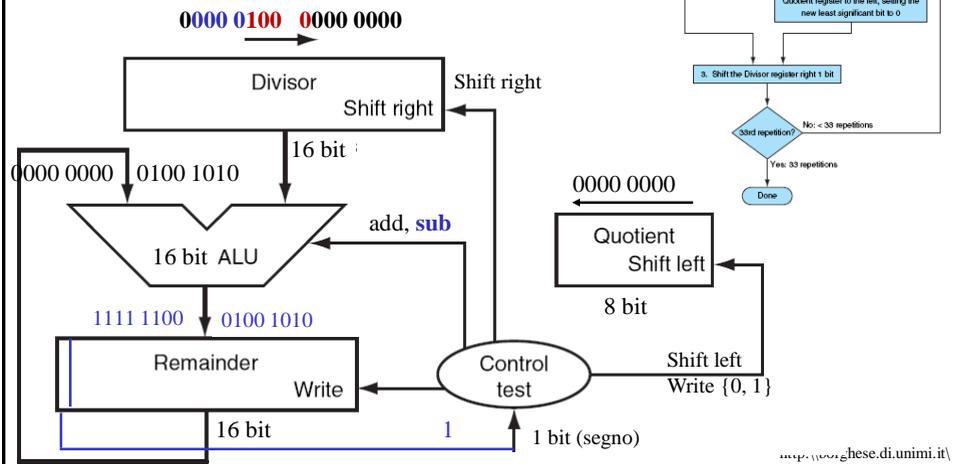
www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 2a

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

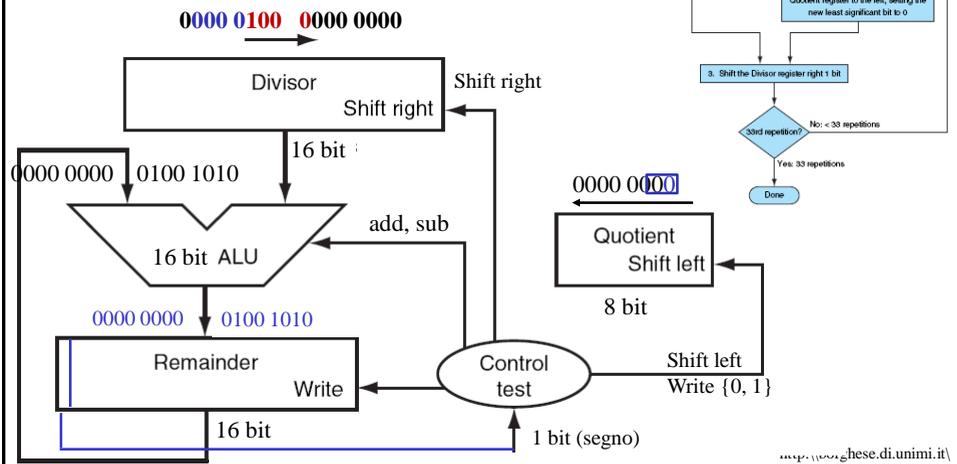
$$74 - 8 \times 2^7 = -950 < 0$$



Il circuito firmware della divisione – passo 2b

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

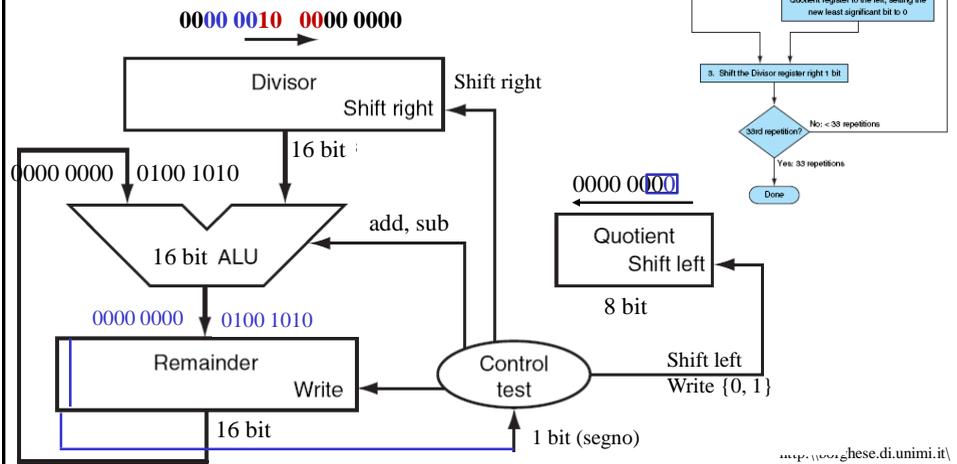
$$74 - 8 \times 2^7 = -950 < 0 \quad -- Q_k = 0$$





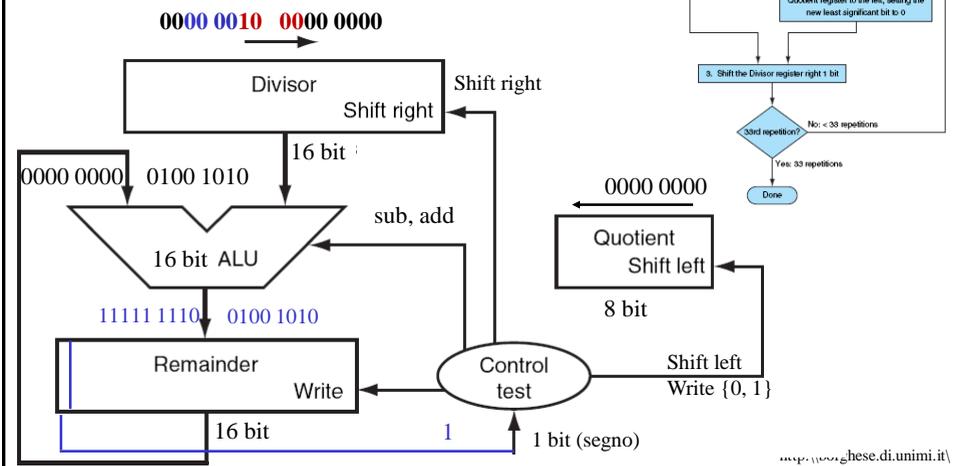
Il circuito firmware della divisione – passo 2c

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
Divisore: 1024_{10} ShiftDx 1 -> 512_{10}



Il circuito firmware della divisione – passo 3a

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^6 = -438 < 0$

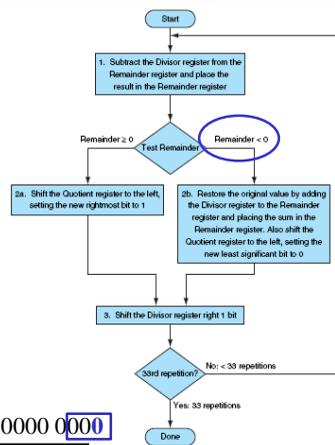
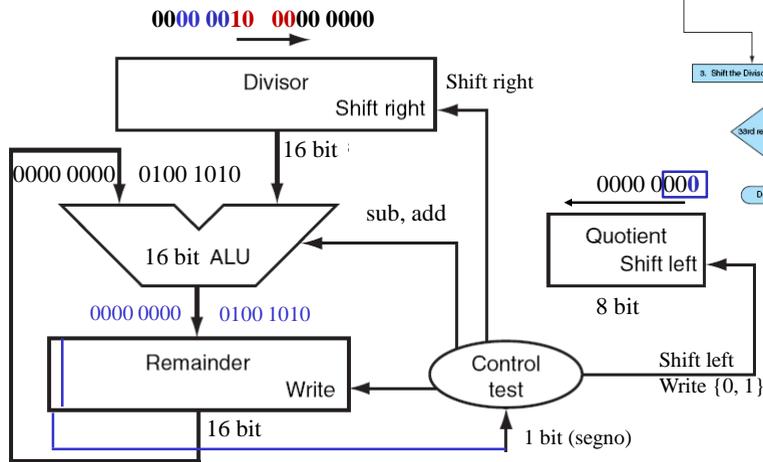




Il circuito firmware della divisione – passo 3b

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0

$$74 - 8 \times 2^6 = -438 < 0 \text{ -- } Q_k = 0$$



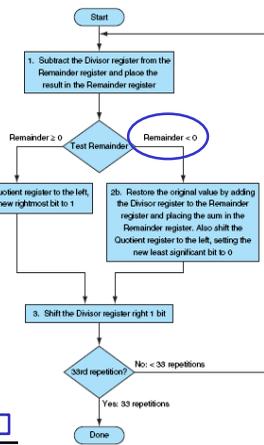
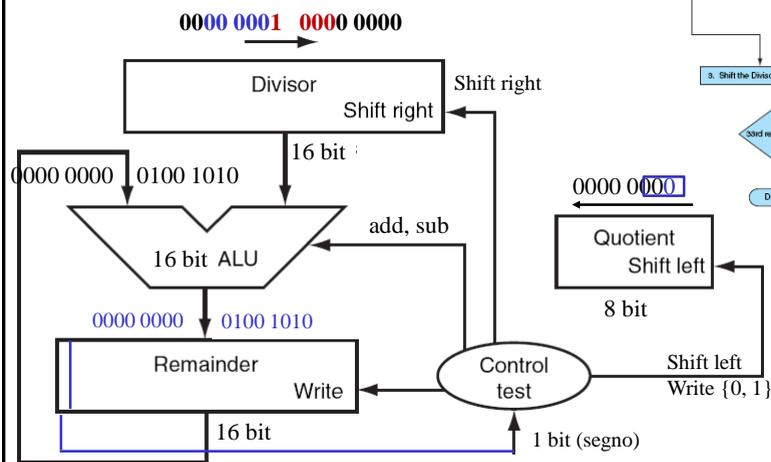
www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 3c

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0

$$\text{Divisore: } 512_{10} \text{ ShiftDx } 1 \rightarrow 256_{10}$$

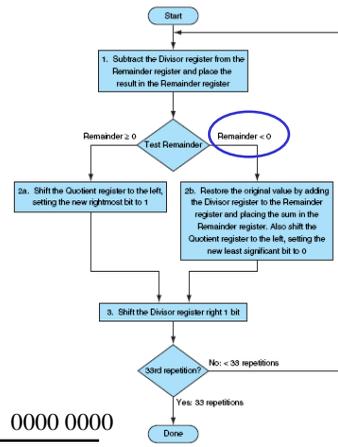
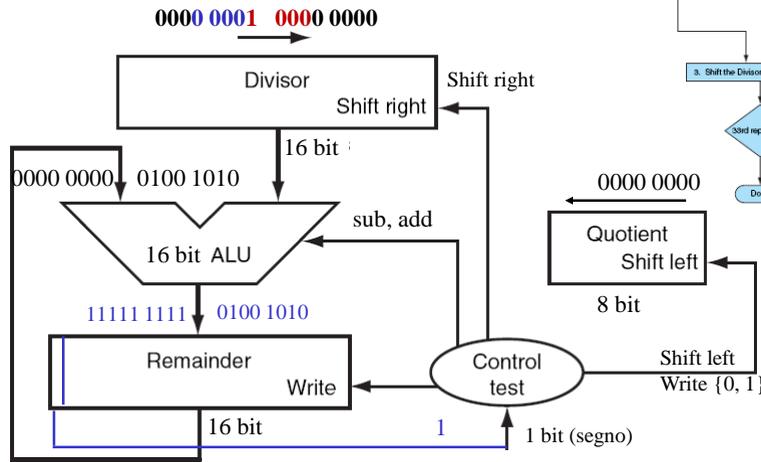


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 4a

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^5 = -182 < 0$

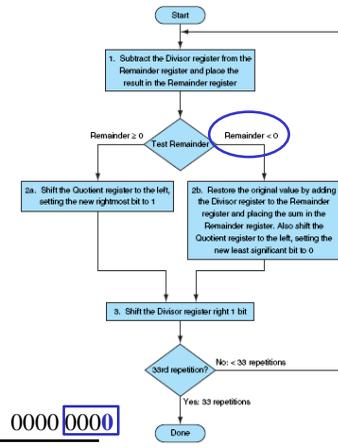
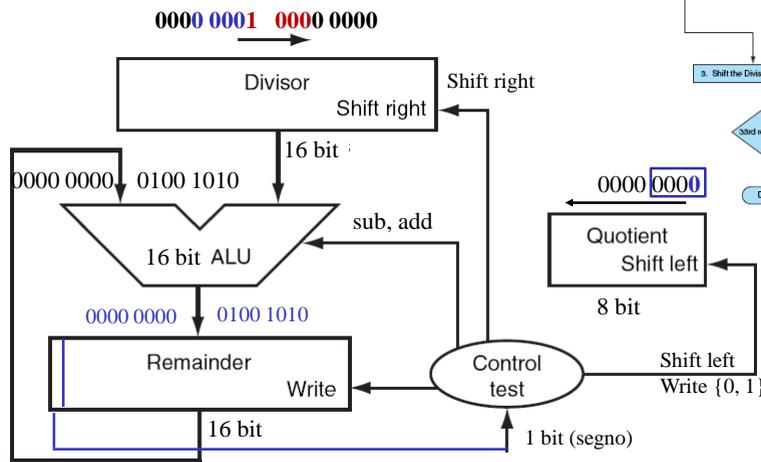


http://www.ghe.se.di.unimi.it/



Il circuito firmware della divisione – passo 4b

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^5 = -182 < 0 \text{ -- } Q_k = 0$



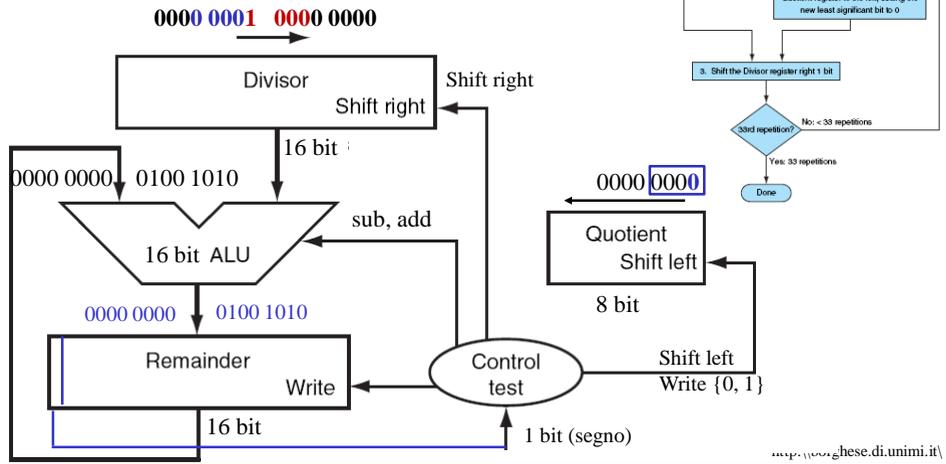
http://www.ghe.se.di.unimi.it/



Il circuito firmware della divisione – passo 4c

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

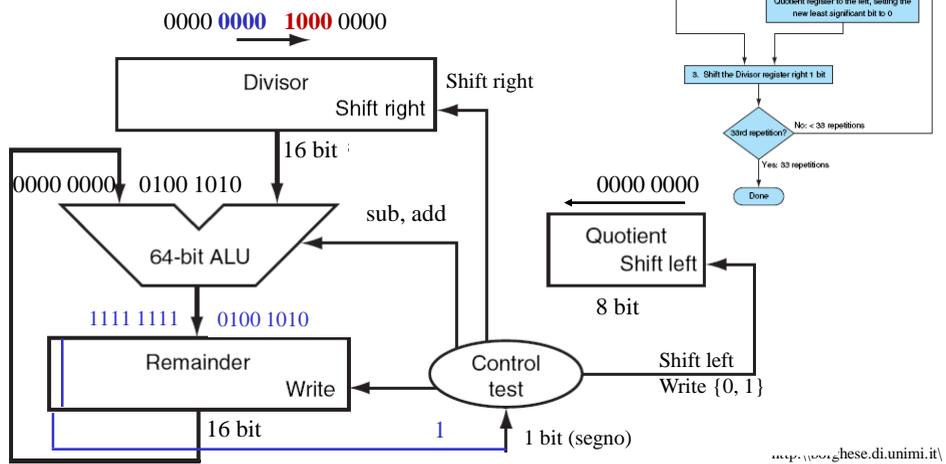
$$74 - 8 \times 2^5 = -182 < 0 \rightarrow Q_k = 0$$



Il circuito firmware della divisione – passo 5a

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

$$74 - 8 \times 2^4 = -54 < 0$$





Il circuito firmware della divisione – passo 5b

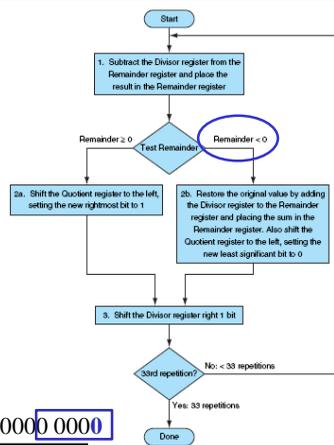
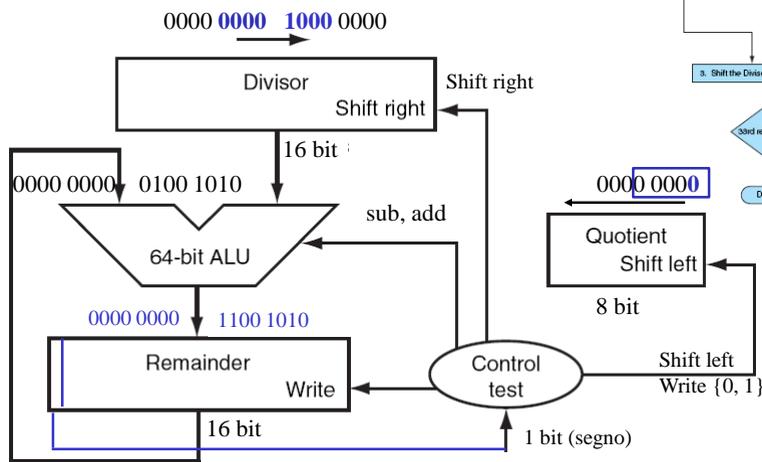
Inizializzazione:

-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$74 - 8 \times 2^4 = -54 < 0$$



www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 5c

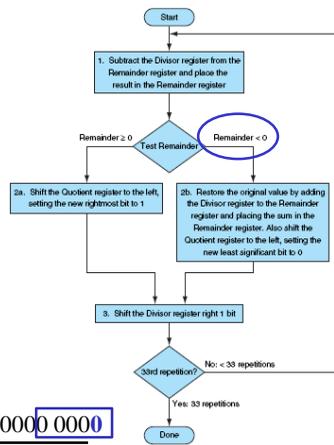
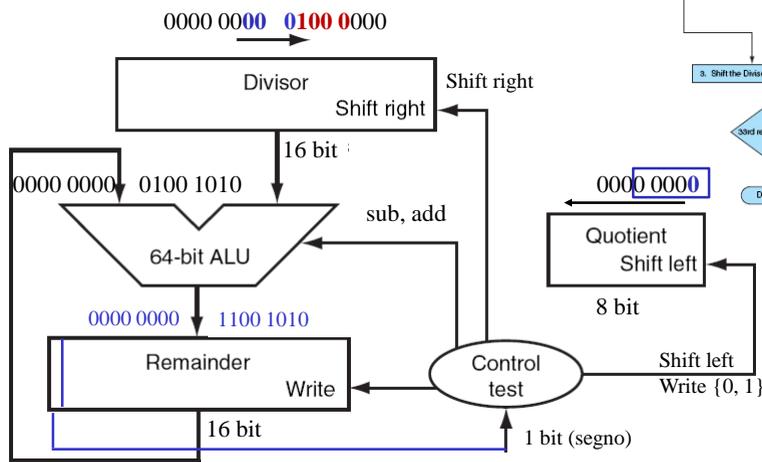
Inizializzazione:

-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$74 - 8 \times 2^4 = -54 < 0$$

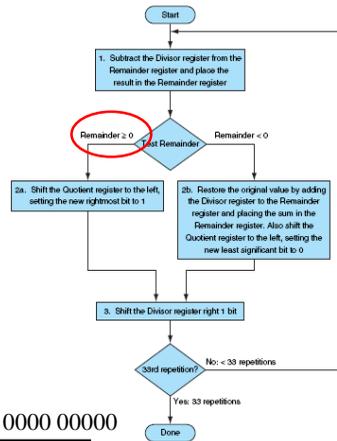
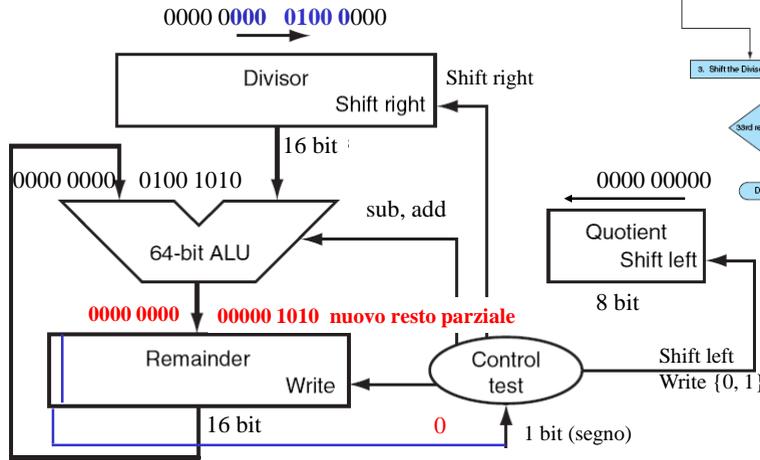


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 6a

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^3 = +10 \geq 0$

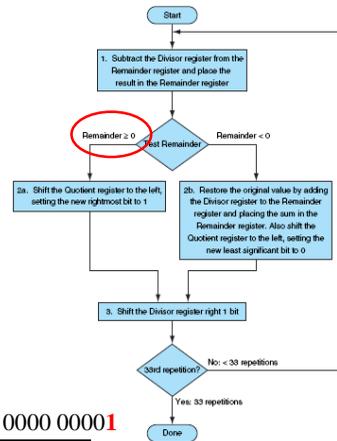
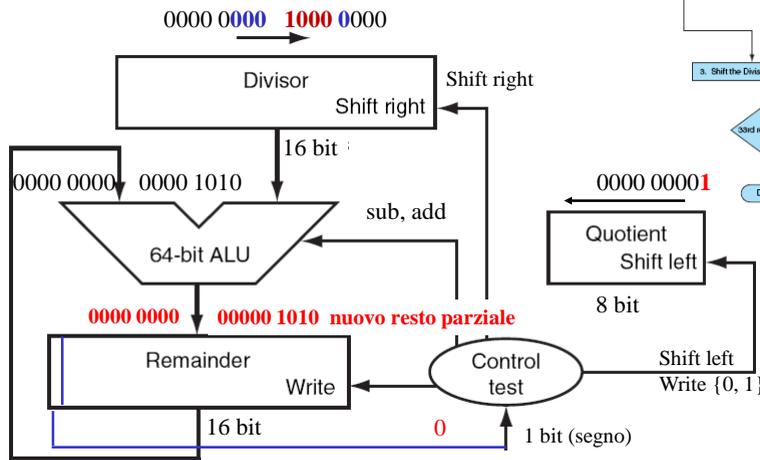


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 6b

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^3 = +10 \geq 0$

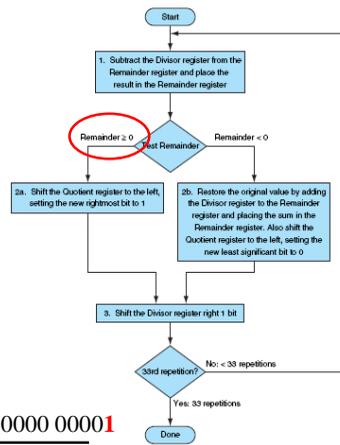
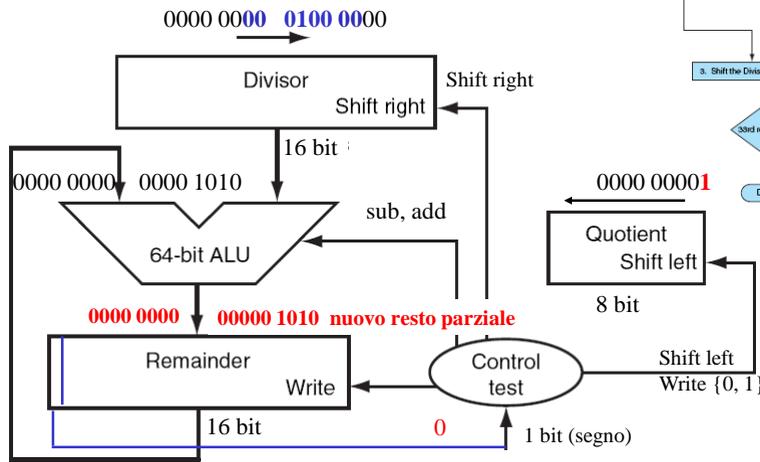


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 6c

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $74 - 8 \times 2^3 = +10 \geq 0$

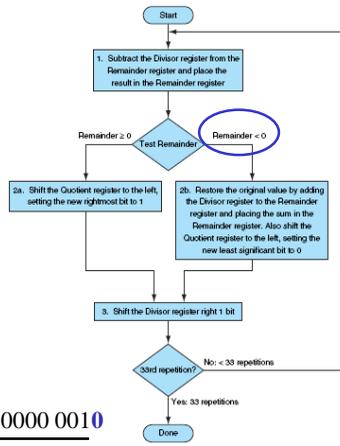
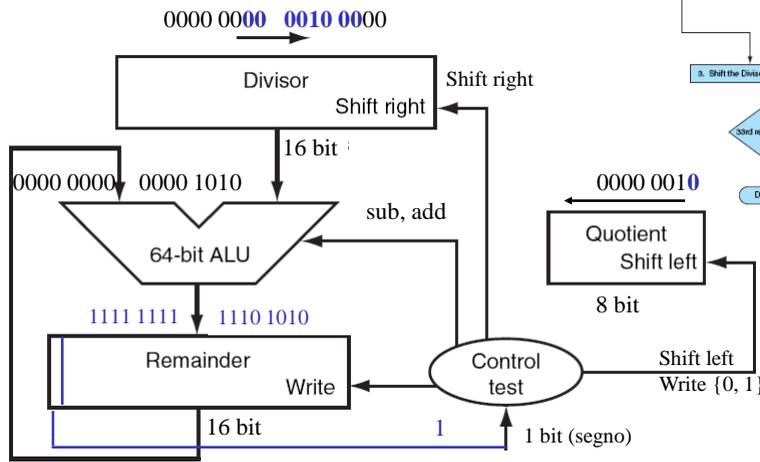


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 7a

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $10 - 8 \times 2^2 = -22 < 0$



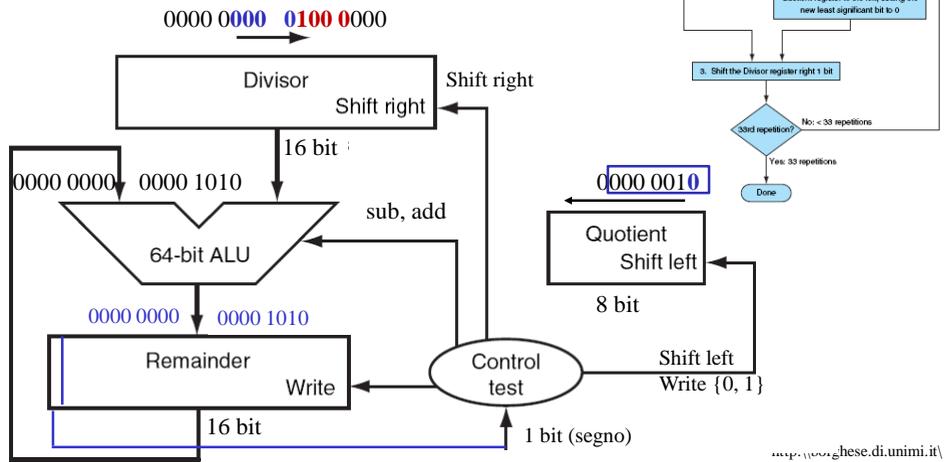
www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 7b

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

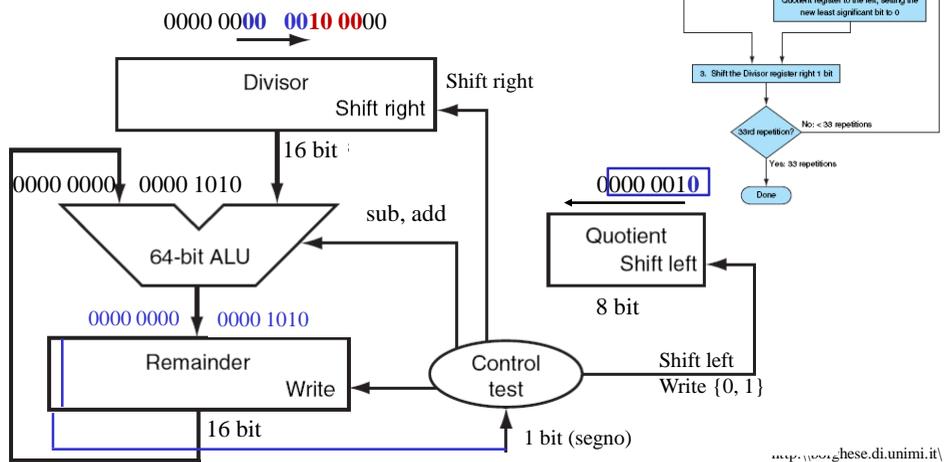
$$10 - 8 \times 2^2 = -22 < 0$$



Il circuito firmware della divisione – passo 7c

Inizializzazione: 1001010 : 1000 =
 -- Resto = 0 | Dividendo
 -- Divisore | 0

$$10 - 8 \times 2^2 = -22 < 0$$





Il circuito firmware della divisione – passo 8a

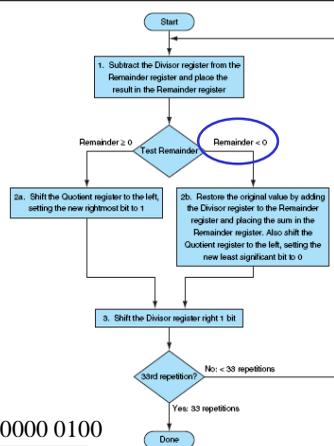
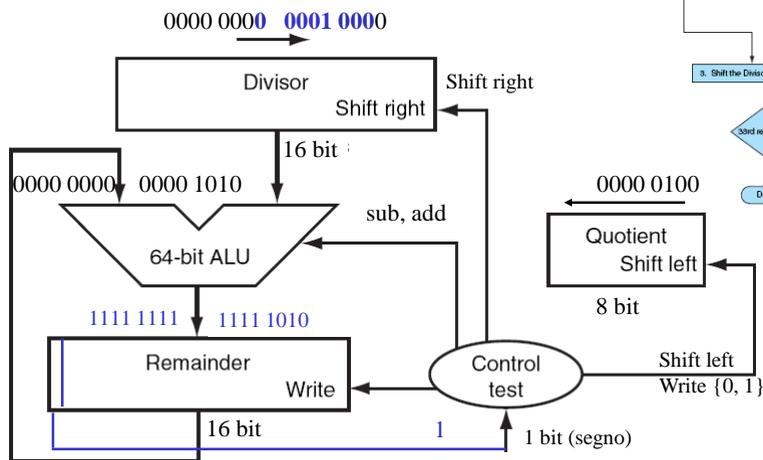
Inizializzazione:

-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$10 - 8 \times 2^1 = -6 < 0$$



www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 8b

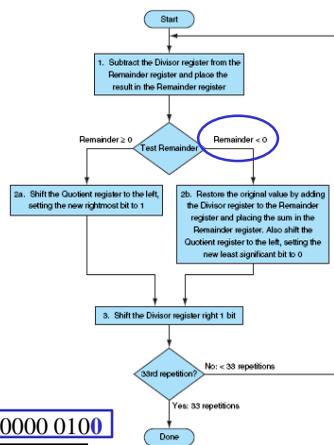
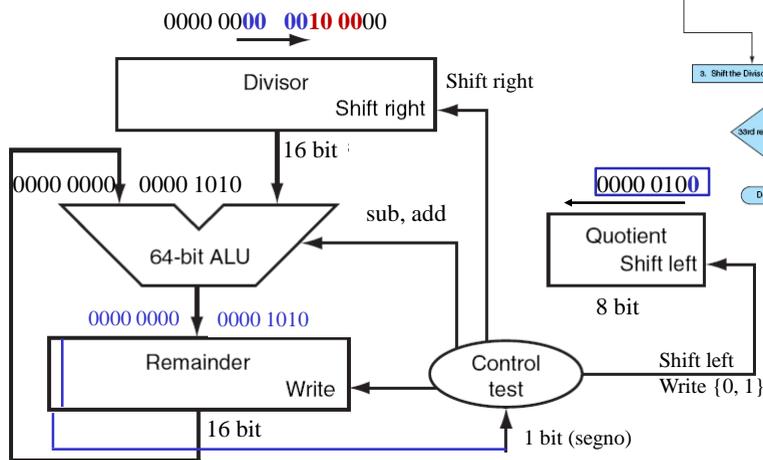
Inizializzazione:

-- Resto = 0 | Dividendo

-- Divisore | 0

$$1001010 : 1000 =$$

$$10 - 8 \times 2^1 = -6 < 0$$

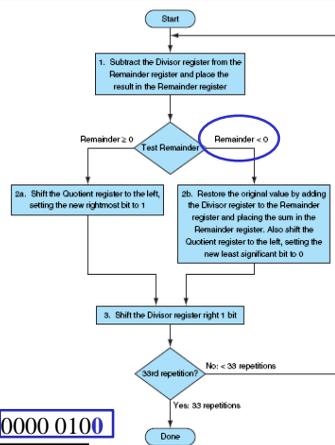
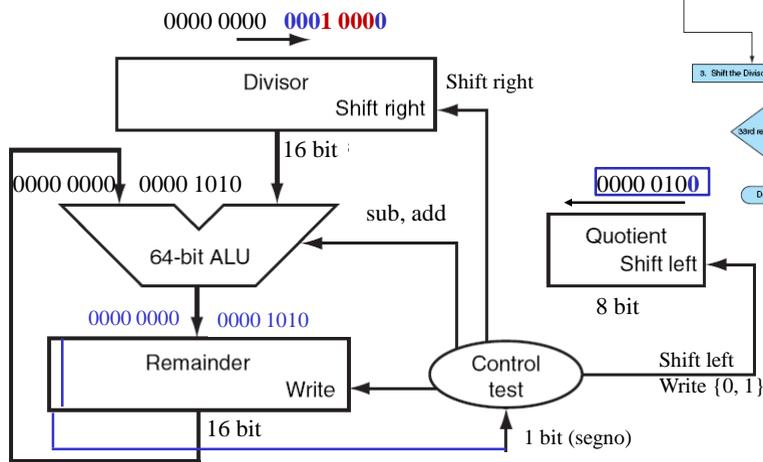


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 8c

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $10 - 8 \times 2^1 = -6 < 0$

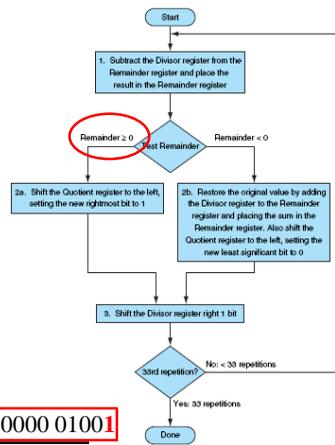
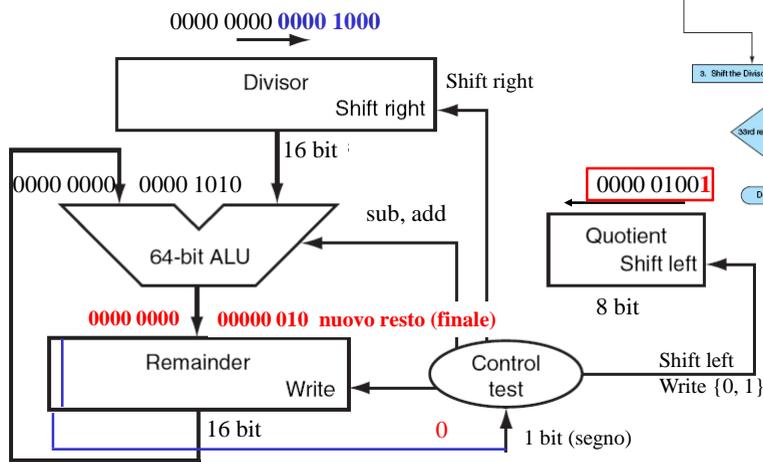


www.ghe.se.di.unimi.it



Il circuito firmware della divisione – passo 9

Inizializzazione: $1001010 : 1000 =$
 -- Resto = 0 | Dividendo
 -- Divisore | 0
 $10 - 8 \times 2^0 = +2 \geq 0$



www.ghe.se.di.unimi.it



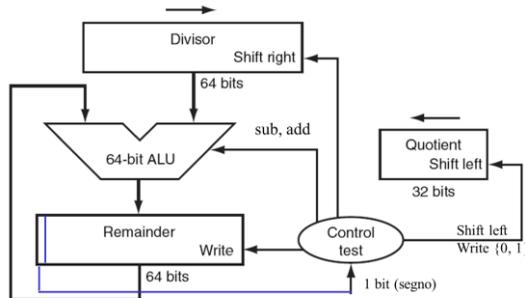
Come ottimizzare il circuito della divisione



Il divisore si sposta verso dx di un bit ad ogni passo e viene sottratto al resto parziale. Otteniamo lo stesso risultato se **spostiamo il resto parziale a sx di un bit ad ogni passo.**

Inizializziamo il resto come $RESTO = 0 \mid DIVIDENDO$.

Ad ogni passo sposto il dividendo alias resto parziale di una posizione a sx ed inserisco un bit del quozie



A.A. 2023-2024

<http://borghese.di.unimi.it/>



Esempio - 1



Divisione decimale fra i numeri $a = 7$ e $b = 2$ su 4 bit. $a : b = ?$

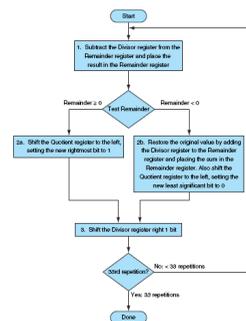
Inizializzo il divisore alla sinistra delle quattro cifre significative. La prima cifra del quoziente sarà sempre 0.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: $Rem = Rem - Div$	0000	0010 0000	0110 0111
	2b: $Rem < 0 \Rightarrow +Div, sll Q, Q_0 = 0$	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111

$$Divisore = 1 \times 2^4 \times 2^1 = 32$$

$$Resto = 7$$

$$7 - 32 = -25 < 0$$



A.A. 2023-2024

58/74



Esempio - 2



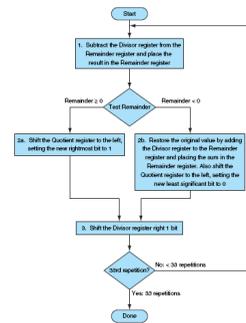
Divisione decimale fra i numeri $a = 7$ e $b = 2$ $a : b = ?$

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q ₀ = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q ₀ = 0	0100	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111

$$\text{Divisore} = 1 \times 2^3 \times 2 = 16$$

$$\text{Resto} = 7$$

$$7 - 16 = -9 < 0$$



Effetto dello spostamento relativo - I



Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q ₀ = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111

Iterazione	Passo	Quoziente	Divisore	Dividendo/Resto
0	Valori iniziali	0000	0010	0 0000 0111
1	Resto = Resto - div	0000	0010	1 1110 0111
	Resto < 0 \rightarrow Resto += div;	0000	0010	0 0000 0111
	Resto << 1 - Q ₀ = 0	0000	0010	0 0000 1110

$$0111 : 10 \Rightarrow$$

$$\begin{array}{r} 0111 - \\ 0010 = \end{array}$$



Effetto dello spostamento relativo - II



Iteration	Step	Quotient	Divisor	Remainder
		0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q ₀ = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111

Iterazione	Passo	Quoziente	Divisore	Dividendo/Resto
		0000	0010	0 0000 1110
2	Resto = Resto - div	0000	0010	1 1110 1110
	Resto < 0 \rightarrow Resto += div;	0000	0010	0 0000 1110
	Resto << 1 - Q ₀ =0	0000	0010	0 0001 1100

$$0111 : 10 \Rightarrow \begin{array}{r} 0111 - \\ 0010 = \end{array}$$



Effetto dello spostamento relativo - III



Iteration	Step	Quotient	Divisor	Remainder
		0000	0000 1000	0000 0111
3	1: Rem = Rem - Div	0000	0000 1000	0111 1111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q ₀ = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111

Iterazione	Passo	Quoziente	Divisore	Dividendo/Resto
		0000	0010	0 0001 1100
3	Resto = Resto - div	0000	0010	1 1111 1100
	Resto < 0 \rightarrow Resto += div;	0000	0010	0 0001 1100
	Resto << 1 - Q ₀ =0	0000	0010	0 0011 1000

$$0111 : 10 \Rightarrow \begin{array}{r} 0111 - \\ 0010 = \end{array}$$



Effetto dello spostamento relativo - IV



Iteration	Step	Quotient	Divisor	Remainder
4	1: Rem = Rem - Div	0000	0000 0100	0000 0111
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011

Iterazione	Passo	Quoziente	Divisore	Dividendo/Resto
		0000	0010	0 0011 1000
4	Resto = Resto - div	0000	0010	0 0001 1000
	Resto > 0	0000	0010	0 0001 1000
	Resto << 1 - Q ₀ =1	0001	0010	0 0011 0000

$$0111 : 10 \Rightarrow \begin{array}{r} 0111 - \\ 0010 = \end{array}$$



Effetto dello spostamento relativo - V



Iteration	Step	Quotient	Divisor	Remainder
5	1: Rem = Rem - Div	0001	0000 0010	0000 0011
	2a: Rem ≥ 0 ⇒ sll Q, Q0 = 1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

Iterazione	Passo	Quoziente	Divisore	Dividendo/Resto
		0001	0010	0 0011 0000
5	Resto = Resto - div	0001	0010	0 0001 0000
	Resto > 0	0000	0010	0 0001 0000
	Resto << 1 - Q ₀ =1	0011	0010	0 0010 0000

$$0111 : 10 \Rightarrow \begin{array}{r} 0011 - \\ 0010 = \end{array}$$

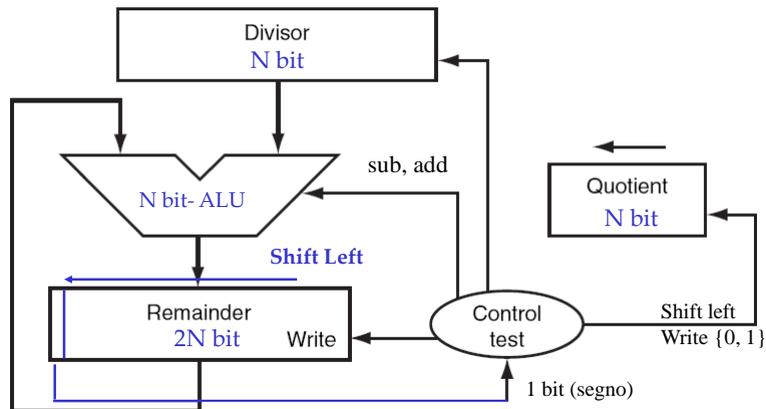
Il resto finale va fatto scorrere a dx di 1 posizione



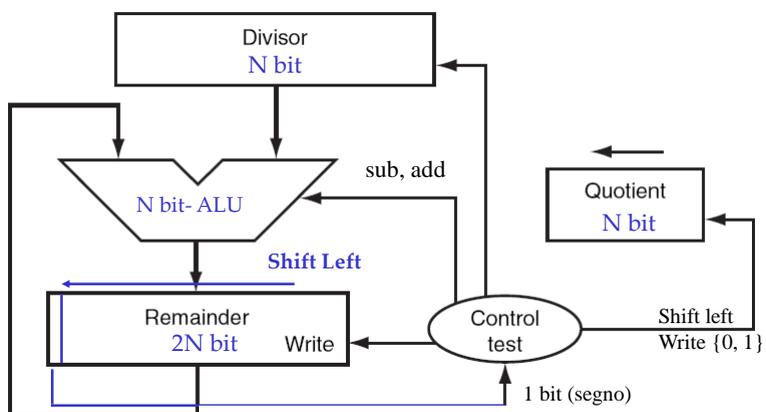
Il circuito firmware con un'ottimizzazione



Inizializzazione: Resto = 0 | Dividendo



Razionale per un'ulteriore ottimizzazione



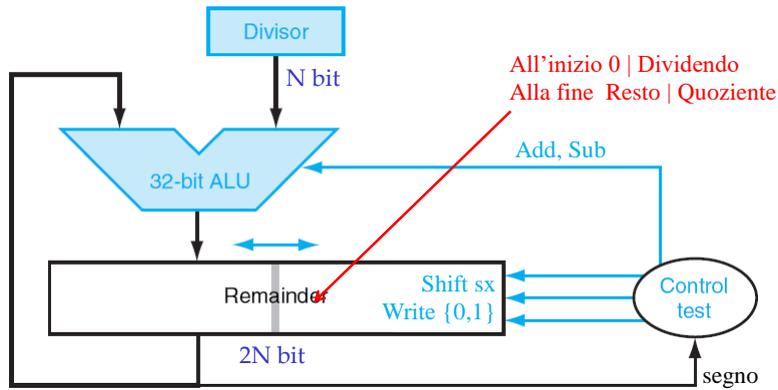
Il quoziente viene riempito un bit alla volta da dx a sx
Il dividendo viene spostato da dx a sx di una posizione



Il circuito firmware ottimizzato della divisione



Inizializzazione: Resto = 0 | Dividendo

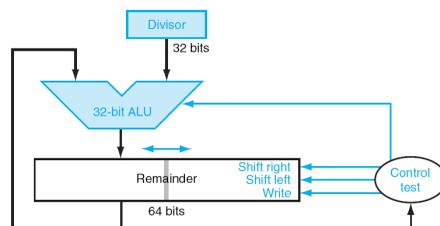


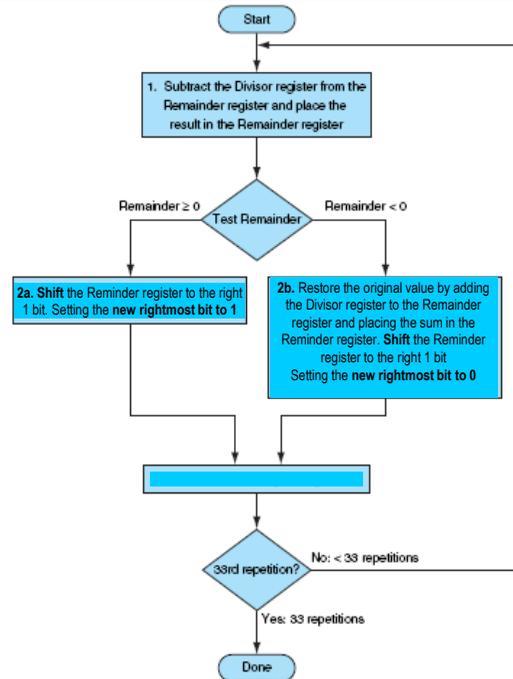
Effetto sul circuito



Iteration	Step	Quotient	Divisor	Remainder
		0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 ⇒ +Div, sll Q, Q ₀ = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111

Iterazione	Passo	Divisore	Dividendo/Resto/Quoziente
		0010	0 0000 1110
2	Resto = Resto - div	0010	1 1110 1110
	Resto < 0 -> Resto += div;	0010	0 0000 1110
	Resto << 1 - Q ₀ = 0	0010	0 0001 1100





Divisione:: algoritmo

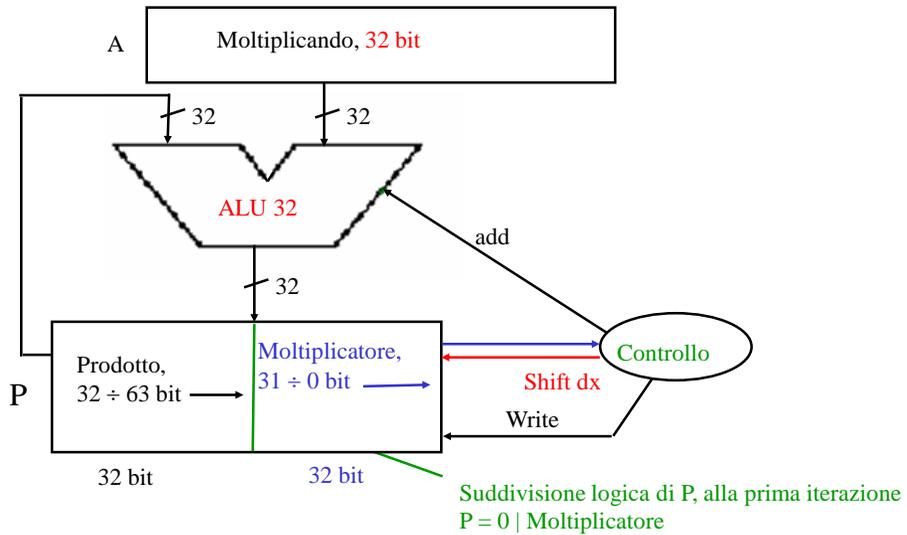


Sommario

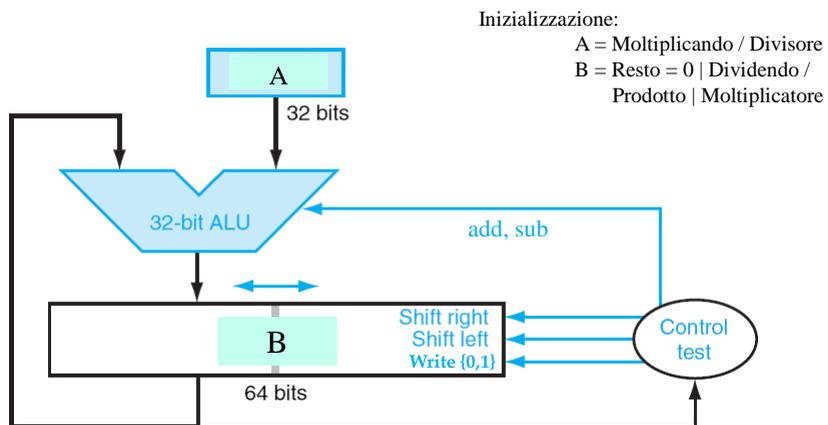
- Divisione intera
- Circuiti divisione intera
- **Divisione e moltiplicazione**



Circuito ottimizzato della moltiplicazione (su 32 bit)



Un unico circuito per moltiplicazione e divisione





Segno divisione e moltiplicazione



Moltiplicazione e divisione di numeri positivi -> viene aggiunto il segno alla fine.

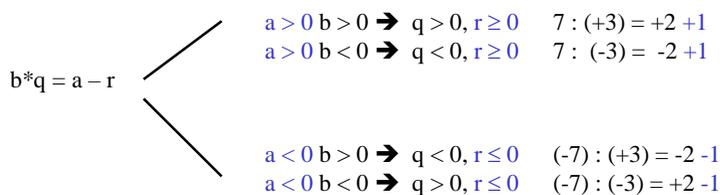
Moltiplicazione: XOR dei bit di segno di moltiplicando e moltiplicatore

Se XOR positivo -> prodotto negativo

Divisione: $a : b = q + r$

XOR dei bit di segno di dividendo e divisore. Se XOR = 1 -> quoziente negativo.

E il resto?



Il dividendo e il resto hanno sempre lo stesso segno



Sommario



- Divisione intera
- Circuiti divisione intera
- Divisione e moltiplicazione