SCHEDA **ASE1904**	Data: **12 Aprile 2019**
Cognome	Nome

ESERCIZIO N°1

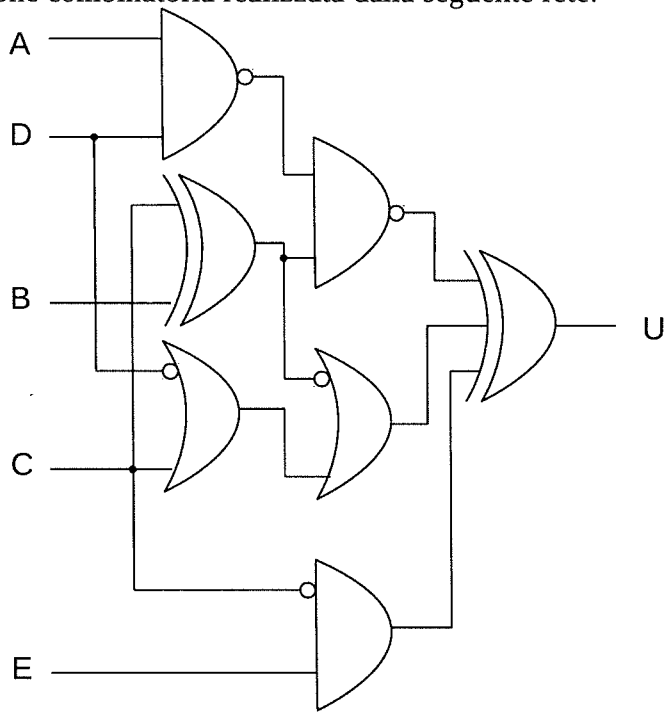
8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che esegue il prodotto scalare tra due vettori di 256 componenti, costituite da 1 valore intero con segno su 1 byte, il cui primo elemento è rispettivamente puntato da X e Y, lasciando il risultato (determinare quanti byte occorrono per la sua rappresentazione senza che si abbia overflow) a partire dalla locazione indicata da Z (iniziando dal byte meno significativo).

ESERCIZIO N°2

6 punti

Disegnare lo schema logico in forma normale a minimo numero di letterali (scegliendo la migliore tra SP e PS) della funzione combinatoria realizzata dalla seguente rete:



ESERCIZIO N°3

5 punti

a) Disegnare lo schema circuitale a porte logiche (AND, OR, NOT, XOR) di un full-adder e di un half-adder e determinarne il T_{pd} massimo in funzione del ritardo delle porte usate.

b) Usando i circuiti di cui al punto a) disegnare lo schema circuitale di un blocco sommatore/sottrattore che opera in C2 su 2 operandi a 6 bit e restituisce in uscita il risultato su 6 bit con un flag di overflow, e determinarne il T_{pd} massimo.

ESERCIZIO N°4

4 punti

Avendo a disposizione chip di memoria SRAM da 2 M x 3 (costo 0,42 €) e da 1 M x 4 (costo 0,28 €), progettare un modulo di memoria da 4 M x 16 a costo minimo.

ESERCIZIO N°5

5 punti

Disegnare lo schema logico di un comparatore digitale (in grado di fornire l'indicazione A < B, A > B oppure A = B) tra numeri a 8 bit con segno rappresentati in C1.

ESERCIZIO N°6

5 punti

Dati i numeri $X = -\pi/3$, $Y = -\sqrt{5} + 6/25$, $Z = -13/7$

a) Determinare il numero minimo di bit per rappresentarli in virgola fissa e MS con un errore assoluto minore o uguale in modulo a 10^{-2} e trovare quindi la loro rappresentazione MS.

b) Determinare la loro rappresentazione in virgola mobile nel formato standard IEEE 754 singola precisione (binary32).

c) Se si usa il microcontrollore AVR XMEGA analizzato durante il corso, quale errore si commette nel rappresentare con una rappresentazione frazionale C2 su un singolo registro i numeri di cui sopra?

1

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che esegue il prodotto scalare tra due vettori di 256 componenti, costituite da 1 valore intero con segno su 1 byte, il cui primo elemento è rispettivamente puntato da X e Y, lasciando il risultato (determinare quanti byte occorrono per la sua rappresentazione senza che si abbia overflow) a partire dalla locazione indicata da Z (iniziando dal byte meno significativo).

```
/*
Il risultato richiede 3 byte in quanto il suo range è
-256*127*128..256*128*128, ovvero -2^22+2^15..2^22
*/
scalarXY:
	push R0
	push R1
	push R2
	push R16
	push R17
	push R18
	push R20
	push R21
	push R22
	clr R20 //azzera l'accumulatore
	clr R21
	clr R22
	clr R16 //256 cicli
	loop:
		ld R17,X+
		ld R18,Y+
		muls R17,R18
		sbc R2,R2 //se prodotto negativo, viene -1
		add R20,R0 //accumula il risultato esteso in segno
		adc R21,R1
		adc R22,R2
		dec R16
		brne loop
	std Z+0,R20 //salva il risultato
	std Z+1,R21
	std Z+2,R22
	dec XH //ripristina i puntatori
	dec YH
	pop R22
	pop R21
	pop R20
	pop R18
	pop R17
	pop R16
	pop R2
	pop R1
	pop R0
	ret
```

(2) Esaminiamo i 3 ingressi della porta XOR terminale (F, G, H dall'alto

$$F = \overline{\overline{AD}\cdot(B\oplus C)} = AD + (\overline{B\oplus C})$$

$$G = (\overline{B\oplus C}) + \overline{D} + C$$

$$H = E\cdot\overline{C}$$

Costruiamo le mappe delle 3 funzioni

BC \ AD	00	01	11	10
00	1	1	1	1
01	0	0	1	0
11	1	1	1	1
10	0	0	1	0

00	01	11	10	$\overline{B\oplus C}$
1	1	1	1	1
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0

BC \ AD	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	0	0	1

00	01	11	10	$\overline{B\oplus C}$
1	1	1	1	1
1	1	1	1	0
1	1	1	1	1
1	0	0	1	0

BC \ AD	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$E = \phi$

00	01	11	10
1	1	1	1
0	0	0	0
0	0	0	0
1	1	1	1

$E = 1$

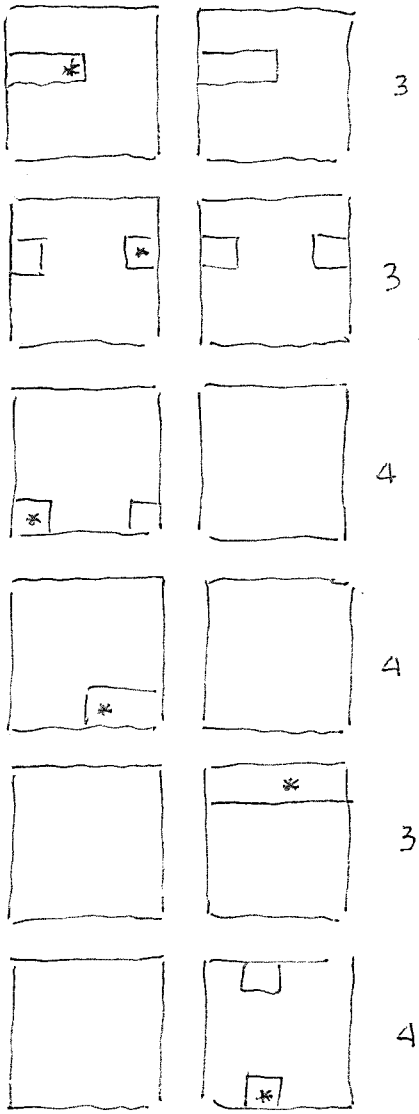
eseguo la XOR bit a bit

BC \ AD	00	01	11	10
00	0	0	0	0
01	1	1	0	1
11	0	0	0	0
10	1	0	1	1

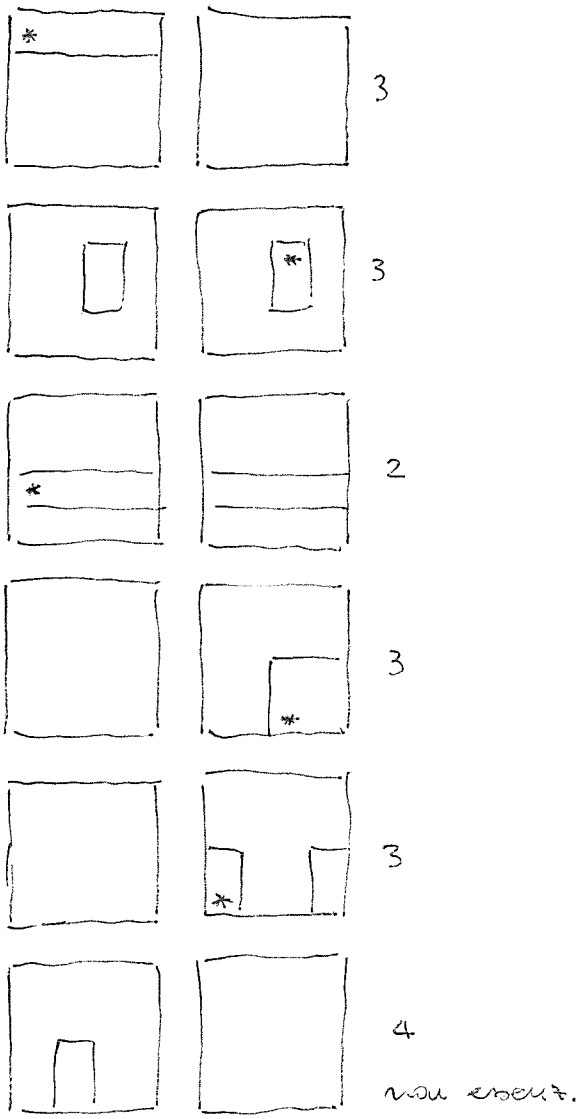
00	01	11	10
1	1	1	1
1	1	0	1
0	0	0	0
0	1	0	0

Cerco le sintesi ottime

ST



PS



TUTTI ESSENZIALI

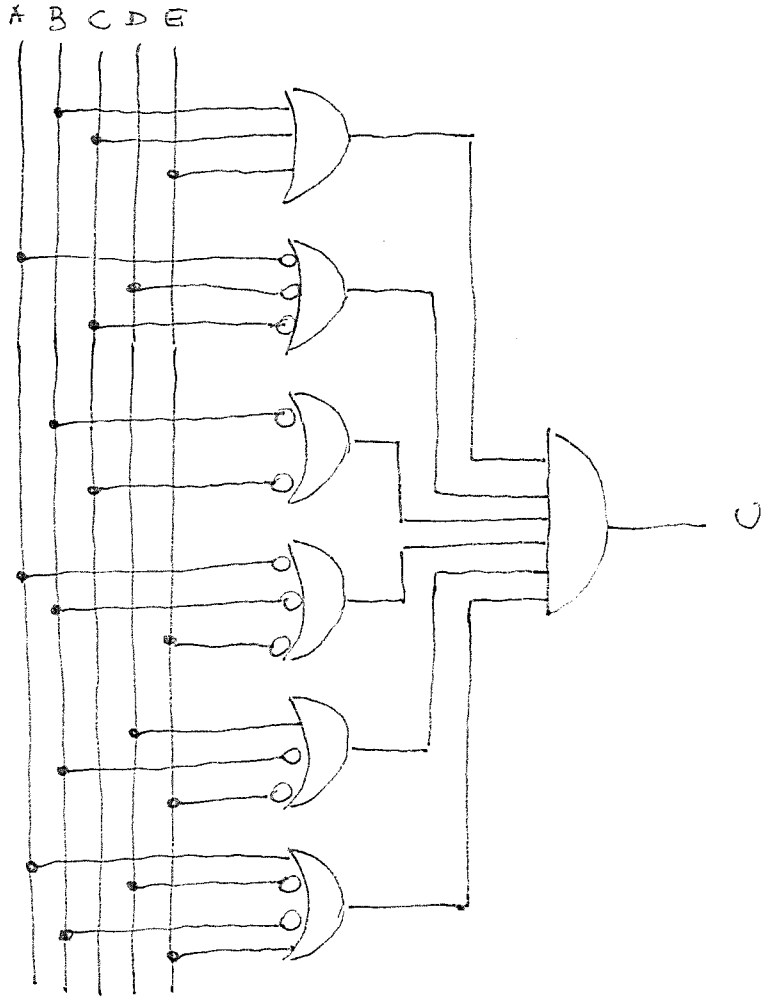
21 esterali

18 esterali

Conviene la sintesi PS_ Espressione in forma normale

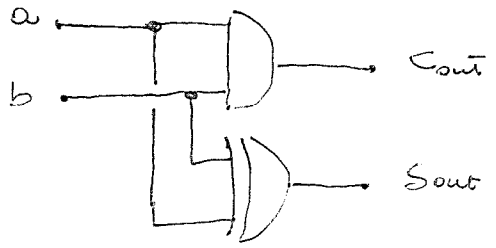
$$U = (B+C+E)\cdot(\bar{A}+\bar{D}+\bar{C})(\bar{B}+\bar{C})(\bar{A}+\bar{B}+\bar{E})(D+\bar{B}+\bar{E})\cdot$$
$$\cdot(A+\bar{D}+\bar{B}+E)$$

Schema logica



(3)

HALF ADDER

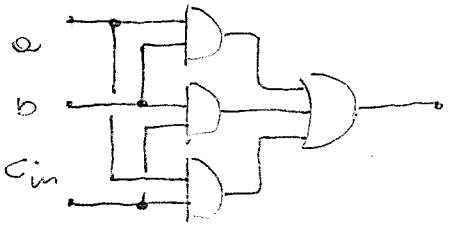


Ritardi

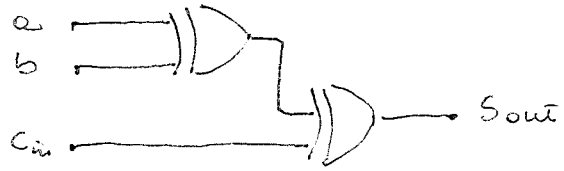
(t_{AND})

(t_{XOR})

FULL ADDER

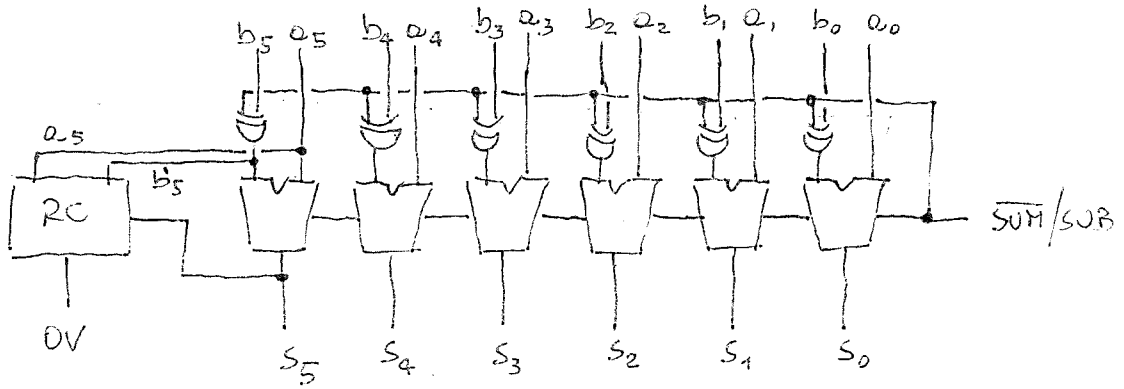


$(t_{AND} + t_{OR3}) = (t_{carry})$

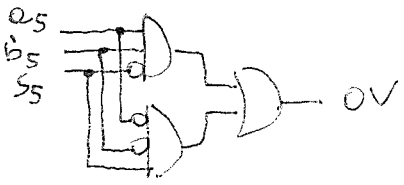


$(2\,t_{XOR}) = (t_{sum})$

SOMMA-SOTTRATTORE a 6 bit (con full-adder a rete per OV) C2



Si ha overflow quando il risultato ha segno diverso dai due operandi CONCORDI



$t_{S5} = 5\,t_{carry} + t_{sum}$

$t_{OV} = t_{S5} + t_{NOT} + t_{AND3} + t_{OR}$

(4) Confronto sul costo per bit (o per Mb)

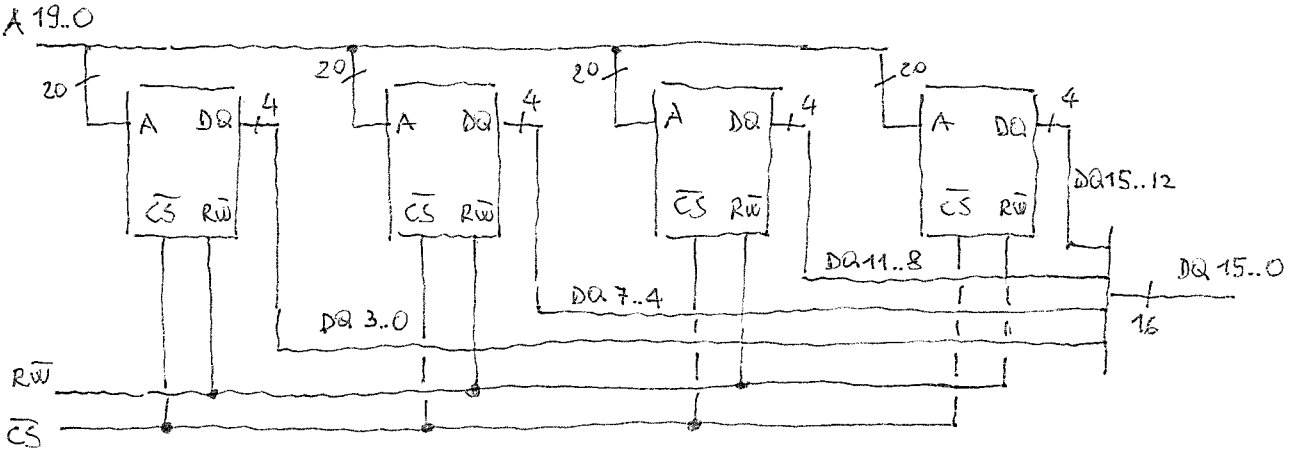
A: 2M×3 $0,42/6 = 7 ¢/Mb$

B: 1M×4 $0,28/4 = 7 ¢/Mb$

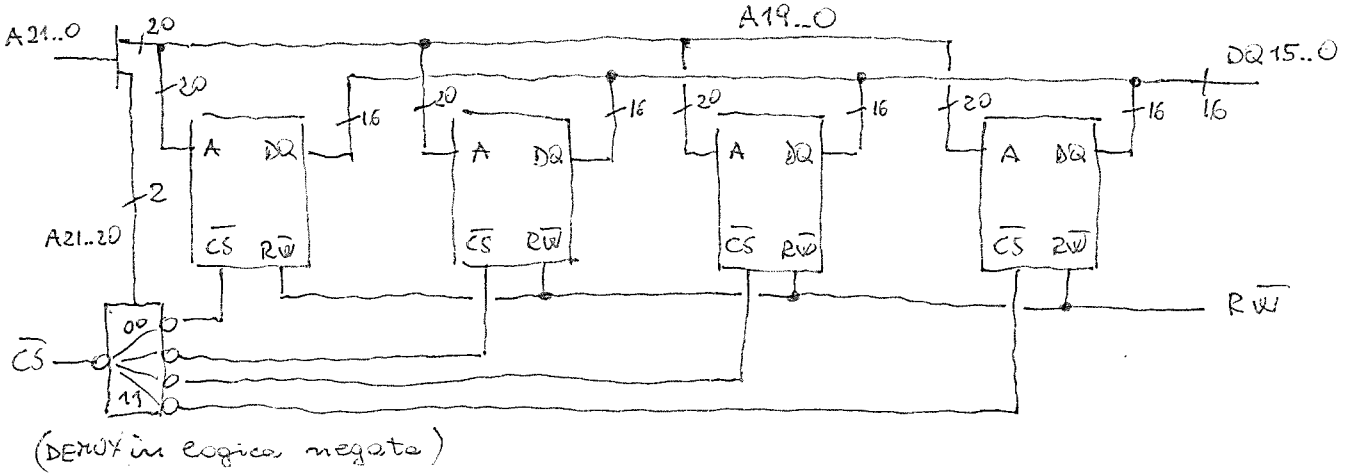
} le costo unitario è uguale
Conviene (a pari costo)
SOLUZIONI SENZA SPRECHI

Può convenire, per ridurre le parti in inventario, usare solo chip B. Si ha un montaggio più regolare e modulare.

Aumento dimensione parola 4 (1M×4) → 1M×16



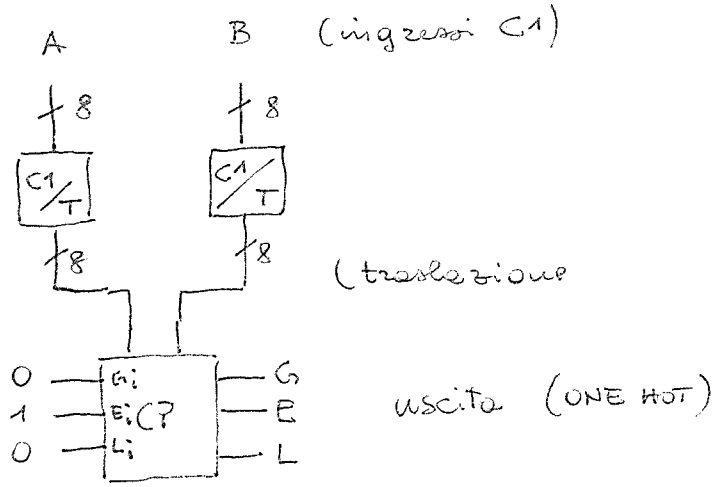
Aumento del numero di parole 4 (1M×16) → 4M×16



(DEMUX in logica negata)

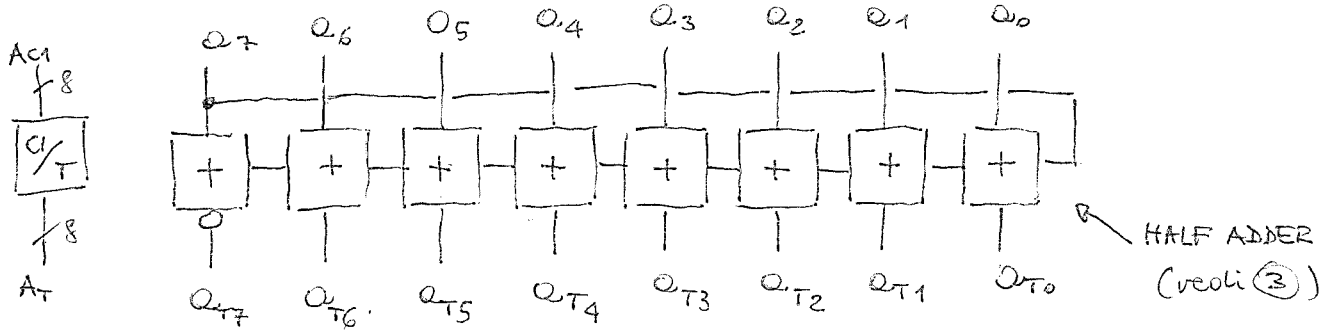
In totale 16 chip; costo 448 €

(5) Per risolvere il problema dato conviene passare in TRASLAZIONE e usare un normale comparatore binario



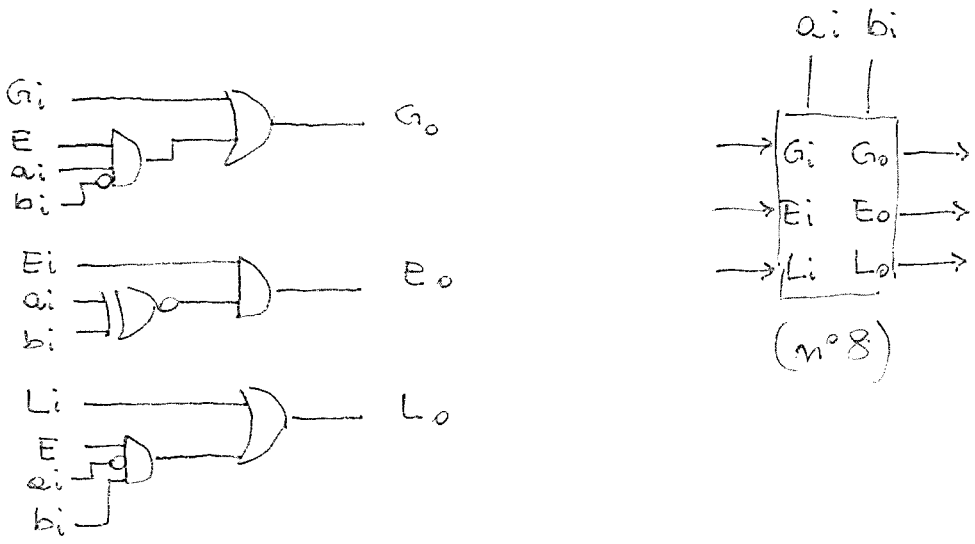
comparatore (MSb first)

Schemi logici: convertitore



- Se negativo SOMMA 1 per passare a C2
- inverte il bit PIÙ SIGNIFICATIVO per passare a T

comparatore MODULARE: schema del modulo relativo al singolo bit



(6) Per avere errore assoluto sicuramente minore di 10^{-2} occorrono PER LA PARTE FRAZIONALE

6 cifre con arrotondamento (molt per 64 e arrotondo)
7 cifre con troncamento (molt per 128 e tronco)

	arr	tronc
$\lvert -\pi/3 \rvert =$	1.000011	1.0000110
$\lvert -\sqrt{5}+6/25 \rvert =$	10.000000	1.1111111
$\lvert -13/7 \rvert =$	1.110111	1.1101101

Interessante osservare che se si sceglie l'ARROTONDAMENTO, occorre un bit in più per la parte intera (e poi uno per segno)

a) Rappresentazione completa con TRONCAMENTO MS

x	11.0000110
y	11.1111111
z	11.1101101

c) Passo in C1 (inverto bit) e poi in C2 (sommo 1), arrotondando su 8 bit

$\hat{x}$	10.1111001	10.1111010	10.111101
$\hat{y}$	10.0000000	10.0000001	10.000001
$\hat{z}$	10.0010010	10.0010011	10.001010
	(C1)	(C2)	(C2) arr.

Valuto l'errore assoluto (uso 4 cifre significative)

$\hat{x}-x = 0,0003226$

$\hat{y}-y = 0,01169$ $(>0,01)$

$\hat{z}-z = 0,01339$ $(>0,01)$

b) Rappresentazione binary 32 IEEE754 (2008)

$$(-1)^{S} \cdot 2^{E-127} \left(1 + T \cdot 2^{-23}\right)$$

	S	E	T
X	1	01111111	000.0110.0000.1010.1001.0010
Y	1	01111111	111.1111.0111.1111.0010.1000
Z	1	01111111	110.1101 0110.0110.1101.1011