| SCHEDA **ASE2104** | | Data: **12 Aprile 2021** |
|---|---|---|
| Cognome | Nome | |

# ESERCIZIO N°1

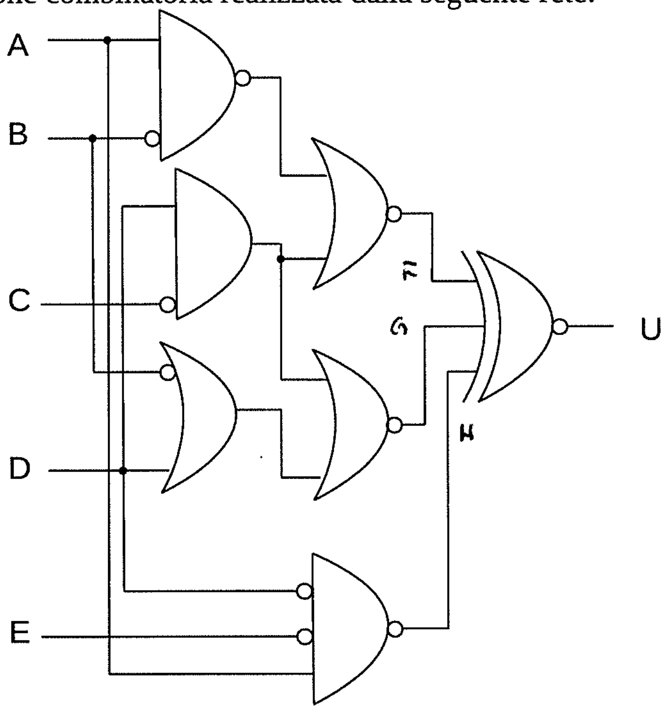
8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che esegue il prodotto scalare tra due vettori di 512 componenti, costituite da 1 valore intero con segno su 1 byte, il cui primo elemento è rispettivamente puntato da Z e X, lasciando il risultato (determinare quanti byte occorrono per la sua rappresentazione senza che si abbia overflow) a partire dalla locazione indicata da Y (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 modulo di confronto per comparatore 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 comparatore che opera in T su 2 operandi a 6 bit e restituisce in uscita il risultato in formato *one hot*, e determinarne il $T_{pd}$ massimo.

# ESERCIZIO N°4

4 punti

Determinare il grafo secondo Mealy sincronizzato di un riconoscitore per le seguenti sequenze comunque interallacciate: 00100, 101, 0111.

Determinare il numero minimo di D-FF necessari per la sintesi della rete e disegnare l'architettura (la sintesi completa delle reti non è richiesta).

## ESERCIZIO N°5

5 punti

Disegnare lo schema logico di un latch di tipo D con abilitazione E e reset prioritario (agisce anche in assenza di abilitazione) R.

## ESERCIZIO N°6

5 punti

Dati i numeri $X = -\pi/5$, $Y = -\sqrt{7} + 7/25$, $Z = -19/11$

**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 migliore rappresentazione in virgola mobile nel formato standard IEEE 754 singola precisione (binary32).

# 1

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU, che esegue il prodotto scalare tra due vettori di 512 componenti, costituite da 1 valore intero con segno su 1 byte, il cui primo elemento è rispettivamente puntato da Z e X, lasciando il risultato (determinare quanti byte occorrono per la sua rappresentazione senza che si abbia overflow) a partire dalla locazione indicata da Y (iniziando dal byte meno significativo).

Per il risultato occorrono 32 bit, infatti il caso che dà il massimo valore (positivo) in uscita prevede un risultato pari esattamente a $2^9\ 2^{14} = 2^{23}$, che (per poco, infatti in C2 a 24 bit si può arrivare a $2^{23} - 1$) non è rappresentabile con 24 bit.

```
scalare:
      push R0
      push R1
      push R2
      push R16
      push R17
      push R18
      push R19
      push R20
      push R21
      push R24
      push R25
      clr R18 //registri per il risultato
      clr R19
      clr R20
      clr R21
      ldi R24,low(512)
      ldi R25,high(512)
loop:
      ld R16,Z+
      ld R17,X+
      muls R16,R17
      sbc R2,R2 //estende il prodotto con segno a 24 b
      add R18,R0
      adc R19,R1
      adc R20,R2 //estensione a 24 b
      adc R21,R2 //estensione a 32 b
      sbiw R25:R24,1
      brne loop
      ldd Y+0,R18 //scrive il risultato, non tocca Y
      ldd Y+1,R19
      ldd Y+2,R20
      ldd Y+3,R21
      subi XH,2 //ripristina X per +512, in alternativa a push e pop
      subi ZH,2 //ripristina Z per +512, in alternativa a push e pop
      pop R25
      pop R24
      pop R21
      pop R20
      pop R19
      pop R18
      pop R17
      pop R16
      pop R2
      pop R1
      pop R0
ret
```

(2) Ricaviamo la mappa della funzione partendo dai valori di F, G, H in ingresso alla XOR

$F = A \cdot \bar{B} \cdot (\bar{D}+C) = A\bar{B}\bar{D} + A\bar{B}C$ (applicando t. De Morgan)

$G = (\bar{D}+C) \cdot B\bar{D} = B\bar{D}$ (anche assorbimento)

$H = D + E + \bar{A}$

F

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 1 |
| 10 | 0 | 0 | 0 | 1 |

E=0

G

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 1 | 1 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 1 | 1 | 0 |

E=0

H

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 1 | 0 | 0 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 0 | 0 |

E=∅

Nota: F e G sono indipendenti da E e H=1 se E=1
Posso quindi trovare U:

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0* | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0* |
| 10 | 1 | 0 | 1 | 1 |

E=0

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0* |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 1 | 0 | 0 | 0 |

E=1

PS: 3 implicati essenziali

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

10 letterali

| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 1 | 1 |
| 01 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 0 |
| 10 | 1 | 0 | 1 | 1 |

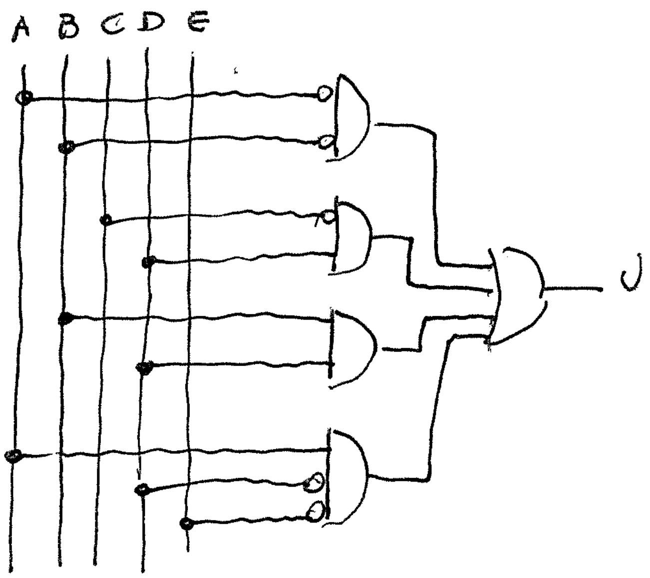
| CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 1 | 1 | 1 | 1* |
| 11 | 1 | 1 | 1* | 0 |
| 10 | 1 | 0 | 0 | 0 |

SP: 2 essenziali e altri 2

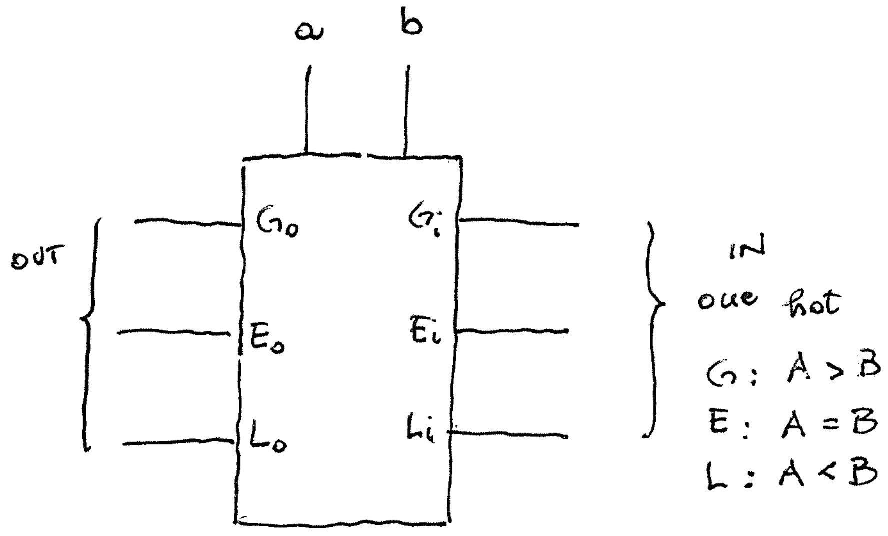
$U = \bar{A}\bar{B} + \bar{C}D + BD +$
$+ A\bar{D}\bar{E}$

9 letterali

Schema logico



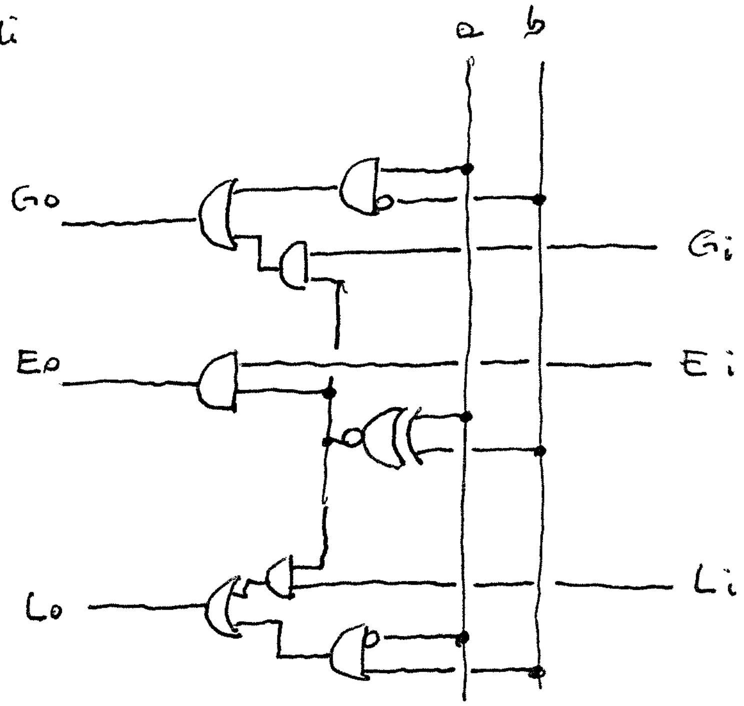
3a) Modulo di confronto per comparatore (tra binari unsigned)
LSB first




bit a=b

$$E_o = E_i \cdot (\overline{a \oplus b})$$
$$G_o = a\bar{b} + G_i (\overline{a \oplus b})$$
$$L_o = \bar{a}b + L_i (\overline{a \oplus b})$$

Quindi

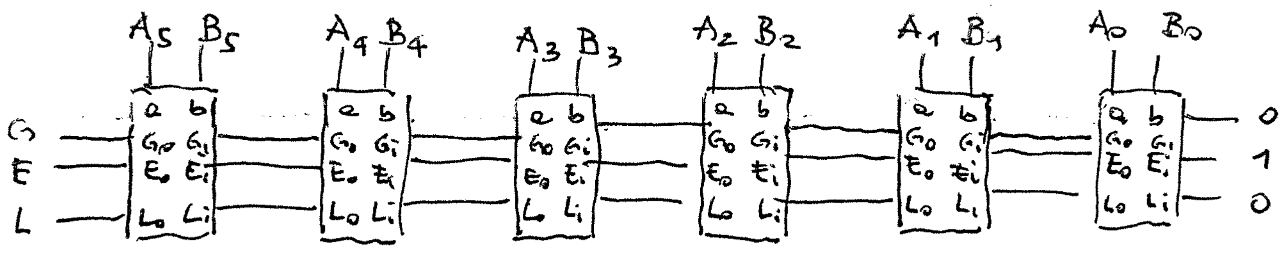



$Tpd(G_o) = T_{XOR} + T_{inv} + T_{AND} + T_{OR}$ $(T_{XOR} > T_{NOT})$

$Tpd(L_o) = Tpd(G_o)$

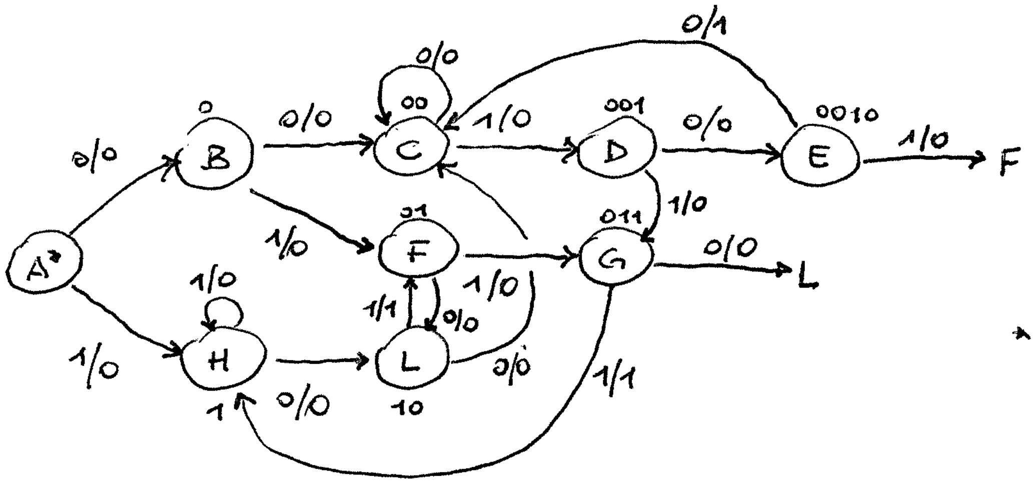
$Tpd(E_o) = T_{XOR} + T_{inv} + T_{AND}$

(3b) il comparatore in T funziona esattamente come il comparatore binario (la codifica T mantiene lo stesso ordinamento dei rappresentati)



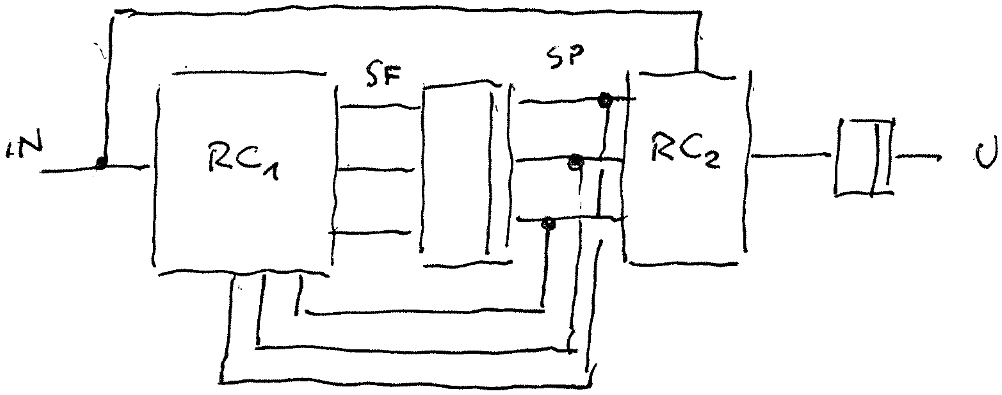

Tempo di ritardo del percorso critico: 6 Tpd ($G_o$)

4

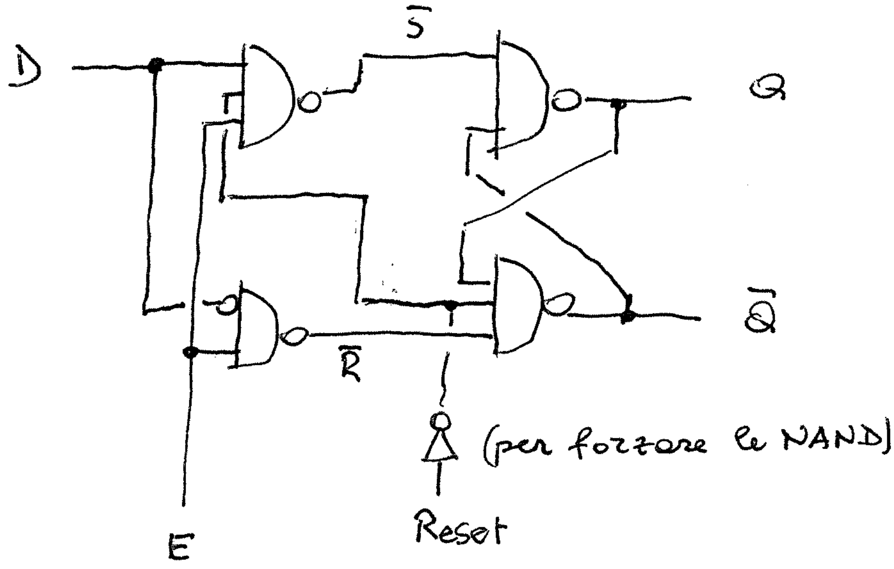



* stato iniziale

La macchina ha 9 stati, senza stati equivalenti. A regime in A e in B non si torna più. Scegliendo C come stato iniziale e accettando un possibile errore nei primi cicli, il riconoscitore si può realizzare con (3 + 1) D-FF



5 Si ha, con porte NAND



(6)

In generale, per ottenere con un frazionale un errore assoluto minore di 1/100 servono 6 bit per la parte frazionale (ARROTONDAMENTO)

I valori proposti sono indicativamente (uso tutta la precisione della calcolatrice)

$x = -0,6283\ldots \qquad y = -2,365\ldots \qquad z = -1,727\ldots$

Per rappresentare correttamente la parte intera occorrono (2+1) bit, compreso il segno.

Esaminiamo quindi una rappresentazione [3.6] MS

| | | |
|---|---|---|
| X: | 100.101000 | $\hat{x} - x = 0,00332$ |
| Y: | 110.010111 | $\hat{y} - y = 0,00638$ |
| Z: | 101.101111 | $\hat{z} - z = -0,00710$ |

L'errore soddisfa i requisiti, come previsto.
Anche una [3.5] soddisferebbe i requisiti di errore, ma si tratterebbe di un caso particolare dovuto alla scelta dei valori specifici.

| | | |
|---|---|---|
| X: | 100.10100 | 0,00332 |
| Y: | 110.01100 | -0,00925 |
| Z: | 101.10111 | 0,00852 |

Per la rappresentazione in binary 32, osserviamo che tutti e 3 sono normalizzabili

$$x = (-1)^S \cdot 2^{(E-127)} \left(1 + T\,2^{-23}\right)$$ da cui

| | S | E | T |
|---|---|---|---|
| X | 1 | 01111110 | 01000001101100101111100 |
| Y | 1 | 10000000 | 00101110110100001111000 |
| Z | 1 | 01111111 | 10111010001011101000110 |

$$\left.\begin{aligned} \varepsilon_x &= 2,78\ 10^{-8} \\ \varepsilon_y &= -1,88\ 10^{-8} \\ \varepsilon_z &= 1,25\ 10^{-8} \end{aligned}\right\}$$ per verifica, non richiesto