

SCHEDA <b>ASE2404</b>		Data: <b>12 Aprile 2024</b>
Cognome	Nome	

Il testo deve essere riconsegnato

### ESERCIZIO N°1

8 punti

Realizzare un sottoprogramma per il microcontrollore XMEGA256A3BU che nel segmento di memoria compreso tra gli indirizzi 0x2300 e 0x23EF (compresi gli estremi) sostituisca in tutte le celle in cui è contenuto un valore multiplo di 8, un nuovo valore pari a un ottavo del valore iniziale e lasci inalterate le altre celle. Nel registro R16 deve essere lasciata l'indicazione del numero di sostituzioni effettuate.

### ESERCIZIO N°2

4 punti

Un sistema di acquisizione dati prevede l'uso di un sensore di temperatura analogico, la cui tensione di uscita varia linearmente da 0 a 1,024 V quando le temperature variano da -30 a 50 °C. La tensione è acquisita da un convertitore A/D (che si può considerare ideale) unipolare a 10 bit che opera proprio nello stesso range (LSB=1 mV; prima soglia a LSB/2).

- Determinare la notazione frazionale C2 a 8 bit più conveniente per rappresentare il dato in gradi centigradi.
- Determinare le operazioni da compiere sul dato di uscita dal convertitore a 10 bit per ottenere il dato C2 a 8 bit da memorizzare (si usi l'arrotondamento classico).
- Determinare il range dell'errore di quantizzazione commesso nella rappresentazione del risultato.

### ESERCIZIO N°3

5 punti

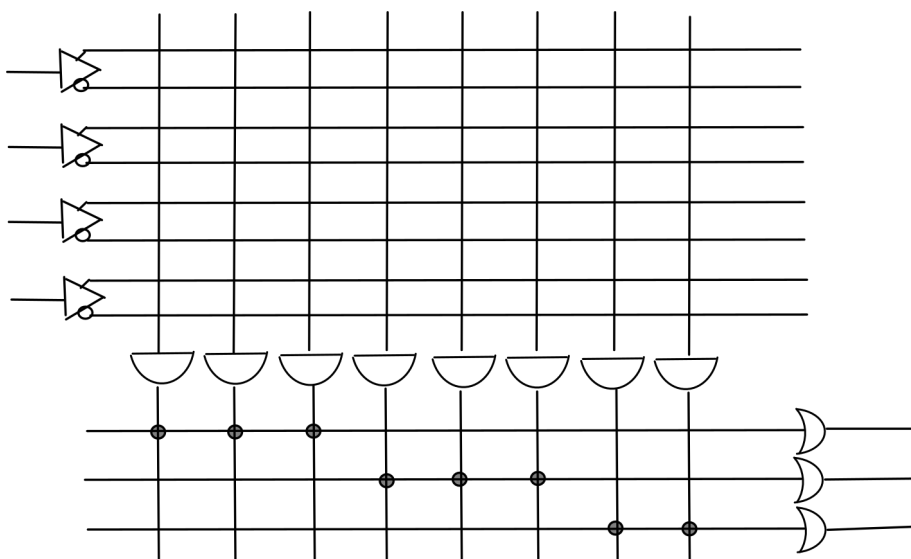
Sintetizzare in forma NAND-NAND a minimo numero di letterali la seguente funzione combinatoria a 5 variabili.

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

### ESERCIZIO N°4

6 punti

Programmare la PAL in figura in modo da realizzare la funzione XOR tra 3 variabili.



## **ESERCIZIO N°5**

5 punti

Disegnare lo schema logico di un sommatore per parole a 3 bit secondo l'approccio carry look-ahead. Si hanno a disposizione porte elementari (AND, OR, NOT, XOR).

## **ESERCIZIO N°6**

5 punti

Realizzare, usando JK-FF, una macchina sincrona secondo il modello di Mealy sincronizzato, con un ingresso e una uscita (all'accensione vale 0), e che ponga a 1 l'uscita in corrispondenza di ogni transizione in salita dell'ingresso e poi che riporti a 0 l'uscita in corrispondenza di ogni transizione in discesa dell'ingresso. Le transizioni dell'ingresso fanno riferimento a variazioni tra due fronti di clock successivi.

Un sistema di acquisizione dati prevede l'uso di un sensore di temperatura analogico, la cui tensione di uscita varia linearmente da 0 a 1,024 V quando le temperature variano da -30 a 50 °C. La tensione è acquisita da un convertitore A/D (che si può considerare ideale) unipolare a 10 bit che opera proprio nello stesso range (LSB=1 mV; prima soglia a LSB/2).

- a) Determinare la notazione frazionale C2 a 8 bit più conveniente per rappresentare il dato in gradi centigradi.
- b) Determinare le operazioni da compiere sul dato di uscita dal convertitore a 10 bit per ottenere il dato C2 a 8 bit da memorizzare (si usi l'arrotondamento classico).
- c) Determinare il range dell'errore di quantizzazione commesso nella rappresentazione del risultato.

- a) Visto il range in gradi celsius, incluso nel range [-64..63,5], conviene usare una notazione frazionale C2 [7.1]  
La risoluzione della rappresentazione è mezzo grado.
- b) Il convertitore misura la tensione del sensore in millivolt con approssimazione  $\pm 0,5$  mV (a parte l'ultimo codice che per la temperatura massima può presentare un errore di -1,5 mV)

Le operazioni da fare per calcolare il codice dalla temperatura in Celsius sono

- 1) cambio scala (80/1024)
- 2) traslazione (-30)
- 3) arrotondamento al codice di mezzo grado più vicino.

$$N = (T + 30) \frac{1024}{80} \quad \text{da cui} \quad T = \frac{80N}{1024} - 30$$

- c) Nel caso in cui si usi una legge di conversione ottimale, l'errore è massimo si ha quando il valore effettivo è vicino alle soglie tra un codice e l'altro e, in caso di arrotondamento, vale  $\pm 0,25^\circ\text{C}$  (trascurando l'errore introdotto dall'ADC, più piccolo in tutti i casi).

```
/* Realizzare un sottoprogramma per il microcontrollore XMEGA256A3BU
   che nel segmento di memoria compreso tra gli indirizzi 0x2300 e 0x23EF
   (compresi gli estremi) sostituisca in tutte le celle in cui
   è contenuto un valore multiplo di 8, un nuovo valore
   pari a un ottavo del valore iniziale e lasci inalterate le altre celle.
   Nel registro R16 deve essere lasciata l'indicazione
   del numero di sostituzioni effettuate.*/
```

```
modify:
push R18                //variabile di appoggio
push XL                 //puntatore
push XH
clr R16                //inizializza conteggio
ldi XL,low(0x2300)
ldi XH,high(0x2300)
loop:
  ld R18,X
  rcall div8           //verifica divisibilità ed esegue la divisione
  brne m1
  inc R16
  st X,R18            //scrive il risultato
  m1: adiw XH:XL,1    //incrementa puntatore
  cpi XL,low(0x23EF+1)
  brne loop
  cpi XH,high(0x23EF+1)
  brne loop
pop XH
pop XL
pop R18
ret
```

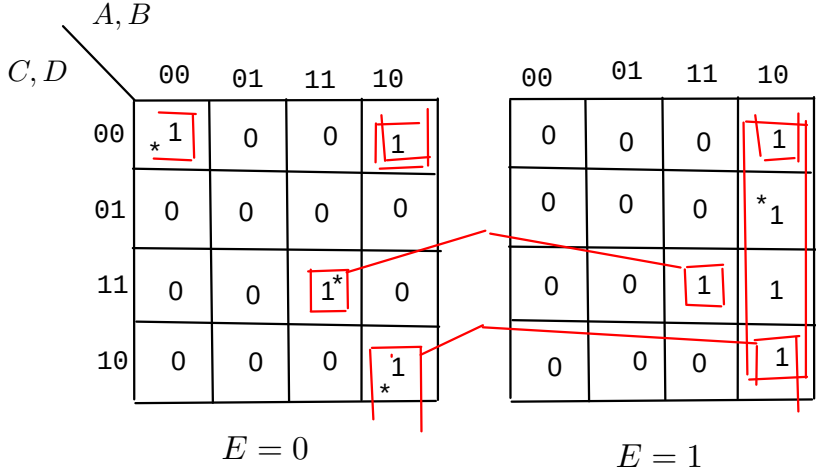
```
/*Verifica divisibilità per 8 (in Z) del valore in R18 ed eventualmente
   esegue la divisione, lasciando il risultato in R18 stesso
```

```
div8:
andi R18,0x0b111
brne nok8             //non multiplo di 8, esce con Z falso
lsr R18
lsr R18
lsr R18               //ha diviso 3 volte per 2
sez                  //esce con Z vero
nok8: ret
```

Sintetizzare in forma NAND-NAND a minimo numero di letterali la seguente funzione combinatoria a 5 variabili.

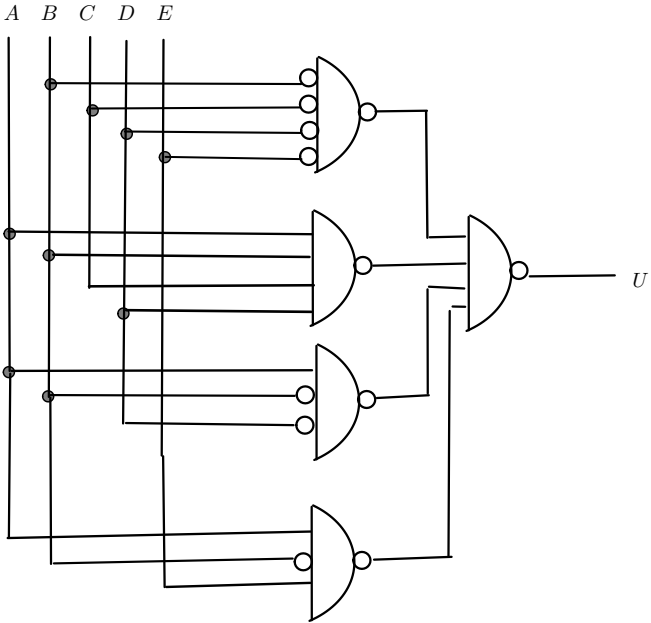
$$U = (A + \overline{C})(\overline{B} + D)(A + C + \overline{E})(\overline{D} + E + B)(\overline{B} + C + \overline{D})$$

Colloco gli implicati in mappa e poi eseguo la sintesi cercando implicanti essenziali e principali che coprano la funzione.



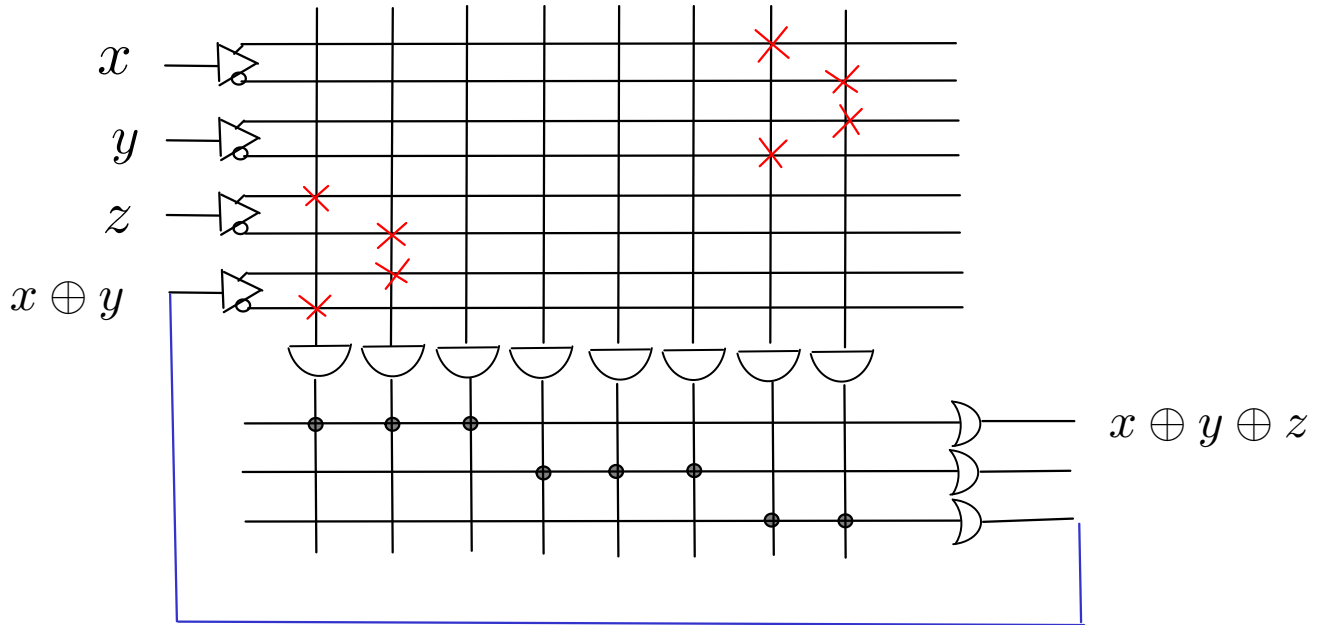
$$U = \overline{B} \overline{C} \overline{D} \overline{E} + ABCD + A\overline{B} \overline{D} + A\overline{B} E$$

La funzione viene ricoperta da 4 implicanti, tutti essenziali. Forma NAND-NAND:



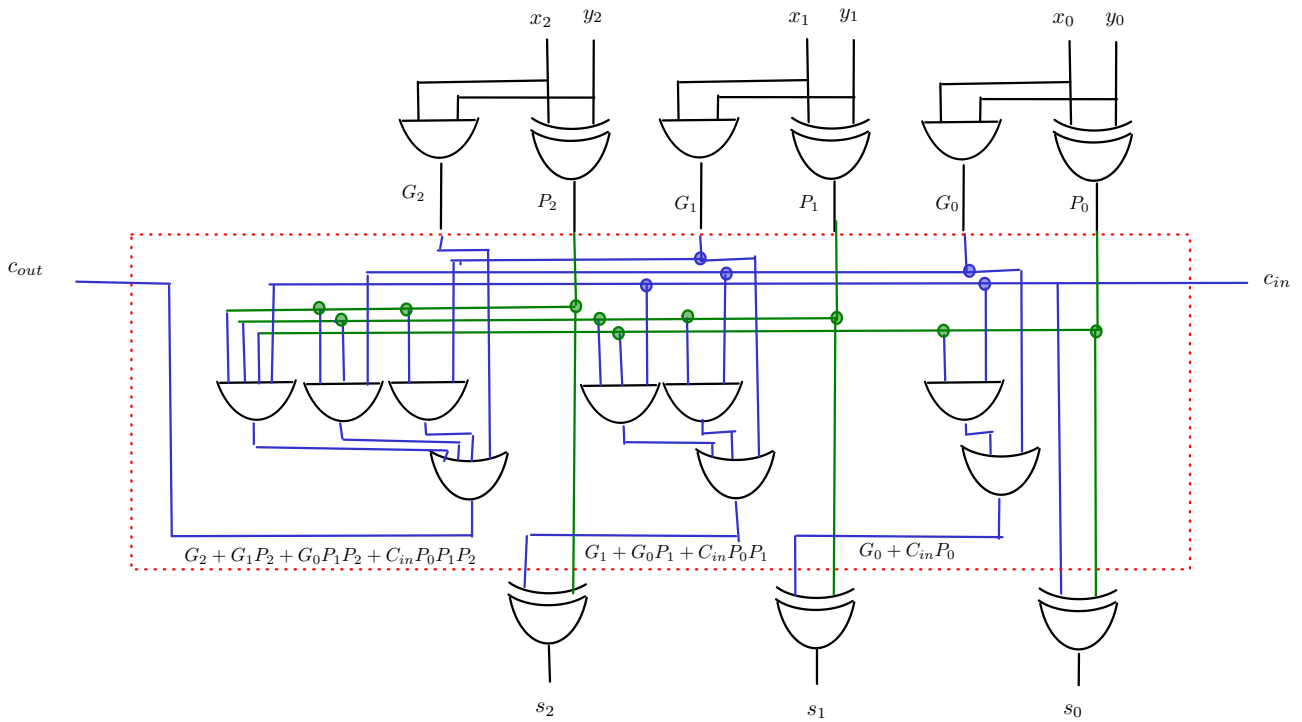
4

Programmare la PAL in figura in modo da realizzare la funzione XOR tra 3 variabili.



5

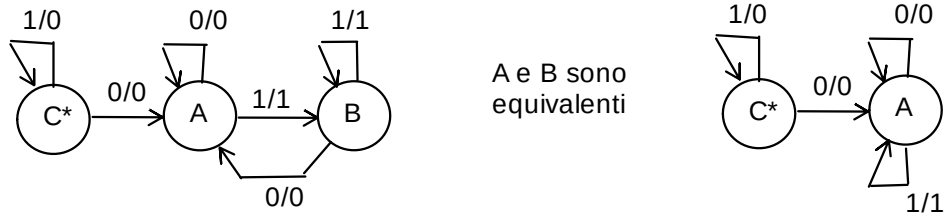
Disegnare lo schema logico di un sommatore per parole a 3 bit secondo l'approccio carry look-ahead. Si hanno a disposizione porte elementari (AND, OR, NOT, XOR).



6

Realizzare, usando JK-FF, una macchina sincrona secondo il modello di Mealy sincronizzato, con un ingresso e una uscita (all'accensione vale 0), e che ponga a 1 l'uscita in corrispondenza di ogni transizione in salita dell'ingresso e poi che riporti a 0 l'uscita in corrispondenza di ogni transizione in discesa dell'ingresso.  
Le transizioni dell'ingresso fanno riferimento a variazioni tra due fronti di clock successivi.

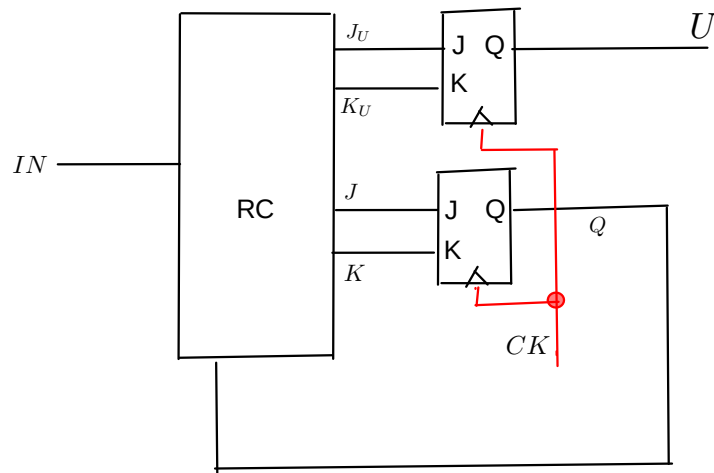
Grafo della macchina (considerando lo stato di accensione C)



Codifica Q

C: 0

A: 1



Architettura

Mapa di transizione

		IN	
		0	1
Q	0	1/0	0/0
	1	1/0	1/1

$$\begin{bmatrix} 1 & 0 \\ - & - \end{bmatrix} \quad J = \overline{IN}$$

$$\begin{bmatrix} - & - \\ 0 & 0 \end{bmatrix} \quad K = 0$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad J_U = IN\overline{Q}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad k_U = \overline{IN} + \overline{Q}$$