

Il testo deve essere riconsegnato nella cartellina. **Non usare il colore rosso nello svolgimento.**

ESERCIZIO N°1

8 punti

Nella memoria di un microcontrollore XMEGA della famiglia AVR, a partire dall'indirizzo il cui valore è contenuto nel puntatore Z, sono contenuti 3 valori senza segno da 1 byte che rappresentano la lunghezza dei 3 lati di un triangolo.

Realizzare una subroutine che determina se i valori dei 3 segmenti possono effettivamente costituire un triangolo, lasciando in R16 il valore 0xFF; altrimenti R16 deve essere azzerato.

ESERCIZIO N°2

5 punti

Disegnare lo schema logico della forma NAND-NAND ottima di una rete combinatoria a 5 ingressi, X_4, X_3, X_2, X_1 , e X_0 caratterizzata dalla tabella di verità:

{1, 1, -, -, 1, -, 1, 0, -, 0, 1, 1, -, 1, 0, 0, 1, 0, -, 1, 1, 0, -, 1, -, 0, -, 0, -, 1}.

Indicare gli implicanti essenziali, giustificando l'affermazione.

ESERCIZIO N°3

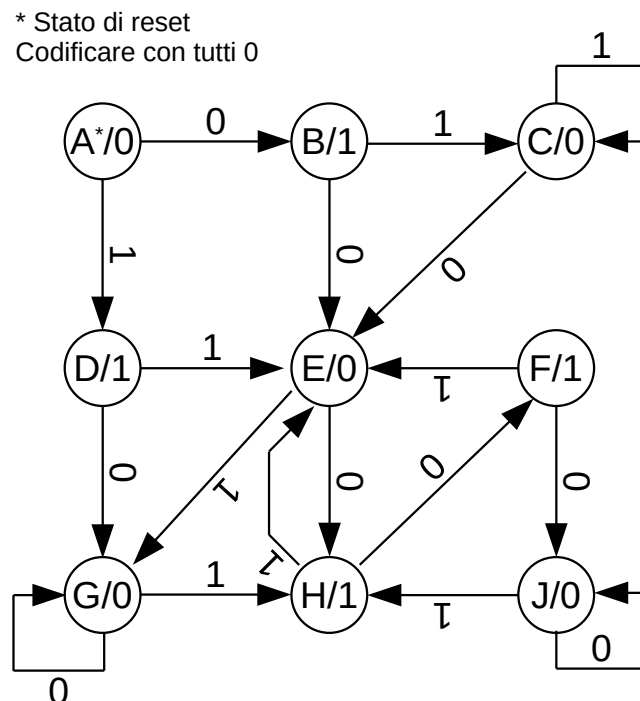
5 punti

Disegnare lo schema logico di un contatore sincrono (up/down con abilitazione) modulo 13 facendo uso di T-FF.

ESERCIZIO N°4

5 punti

Sintetizzare una macchina sequenziale sincrona di Moore, il cui funzionamento è definito dal grafo seguente, individuando e semplificando gli eventuali stati equivalenti. Si usino esclusivamente JK-FF e porte logiche elementari (AND, OR, NOT).



ESERCIZIO N°5

5 punti

Determinare la rappresentazione binary32 (IEEE 754-2008) del numero $-2 \cdot 10^{-21}$ scegliendo il codice che minimizza l'errore relativo (in modulo) e calcolare tale errore.

Determinare poi la rappresentazione binary32 anche del cubo del numero dato.

ESERCIZIO N°6

5 punti

Realizzare la rete combinatoria dell'esercizio 2 facendo uso esclusivamente di mux 2:1, cercando di ridurre il numero, evitando l'uso di blocchi non necessari (duplicati, mux con ingressi identici, ecc.).

```
/*Nella memoria di un microcontrollore XMEGA della famiglia AVR, a partire
dall'indirizzo il cui valore è contenuto nel puntatore Z, sono contenuti
3 valori senza segno da 1 byte che rappresentano la lunghezza dei
3 lati di un triangolo. Realizzare una subroutine che determina se i valori dei
3 segmenti possono effettivamente costituire un triangolo,
lasciando in R16 il valore 0xFF; altrimenti R16 deve essere azzerato.*/
```

```
triangle:
    push R20
    push R21
    push R22
    push R23
    ldi R16,0xFF          //risultato se triangolo possibile
    ld R20,Z              //lato a
    ldd R21,Z+1            //lato b
    ldd R22,Z+2            //lato c
    mov R23,R20            //verifica la prima disuguaglianza triangolare
    add R23,R21
    brcs c1ok              //a+b>255>c
    cp R22,R23              //a+b>c
    brcs c1ok
    rjmp nook
c1ok:
    mov R23,R21            //verifica la seconda disuguaglianza triangolare
    add R23,R22
    brcs c2ok              //b+c>255>a
    cp R20,R23              //b+c>a
    brcs c2ok
    rjmp nook
c2ok:
    mov R23,R22            //verifica la terza disuguaglianza triangolare
    add R23,R20
    brcs c3ok              //c+a>255>b
    cp R21,R23              //c+a>b
    brcs c3ok
    nook:
    clr R16                //triangolo impossibile
c3ok:
    pop R23
    pop R22
    pop R21
    pop R20
    ret
```

Disegnare lo schema logico della forma NAND-NAND ottima di una rete combinatoria a 5 ingressi caratterizzata dalla tabella di verità:

$\{1, 1, -, -, 1, -, 1, 0, -, 0, 1, 1, -, 1, 0, 0, 1, 0, 1, 0, -, 1, 1, 0, -, 1, -, 0, -, 0, -, 1\}$.

Indicare gli implicanti essenziali, giustificando l'affermazione.

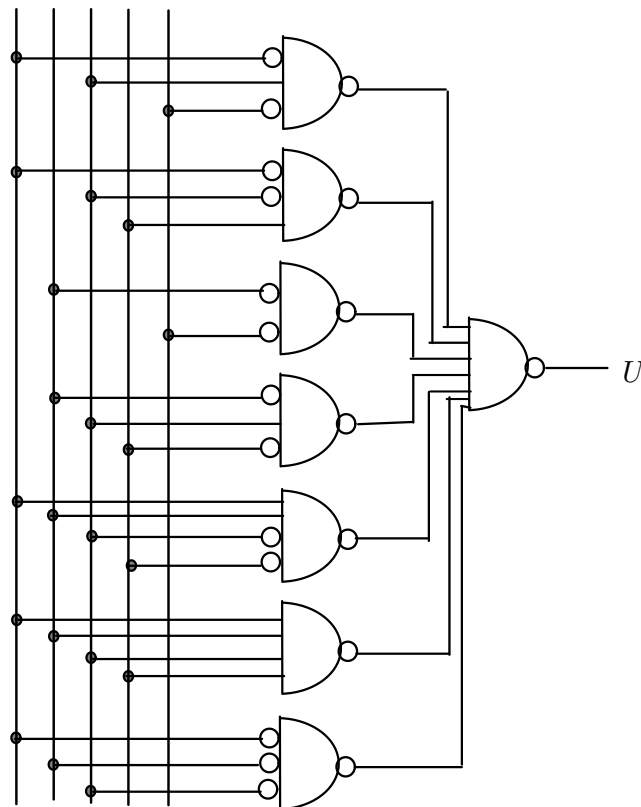
		X_3, X_2							
X_1, X_0		00	01	11	10	00	01	11	10
	00	1	1	-	-	1	-	-	-
	01	1	-	*1	0	0	*1	0	*1
	11	-	0	0	1*	0	0	1*	0
	10	-	*1	0	1	1	1	-	-

$X_4 = 0$ $X_4 = 1$

Servono 7 implicanti, di cui 6 essenziali. La soluzione ottima ha 22 letterali. Parto dalla forma SP, che poi converto in NAND-NAND con il teorema di De Morgan.

$$U = \overline{X_4}X_2\overline{X_1} + \overline{X_4}\overline{X_2}X_1 + \overline{X_3}\overline{X_0} + \overline{X_3}X_2\overline{X_1} + X_4X_3\overline{X_2}\overline{X_1} + X_4X_3X_2X_1 + \overline{X_4}\overline{X_3}\overline{X_2}$$

X_4, X_3, X_2, X_1, X_0



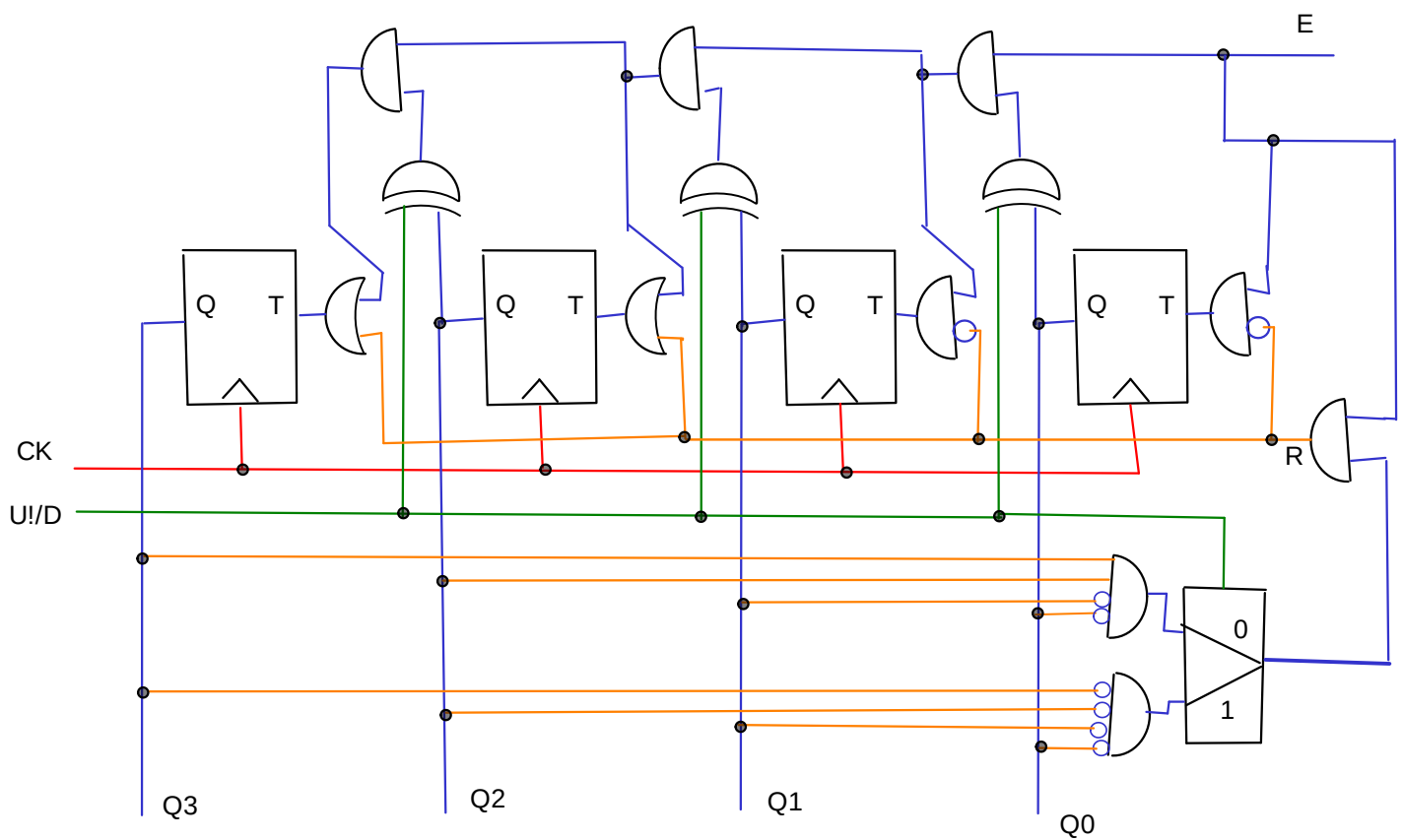
3

Disegnare lo schema logico di un contatore sincrono (up/down con abilitazione) modulo 13 facendo uso di T-FF.

Esamino le sequenze di conteggio:
up down

....	
1010	0010
1011	0001
1100	0000
(FFBB)	(FFBB)
0000	1100
....	

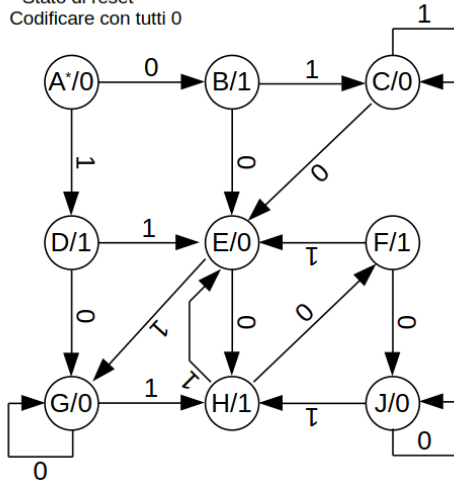
Lo stesso operatore FFBB va bene per entrambe le direzioni.



4

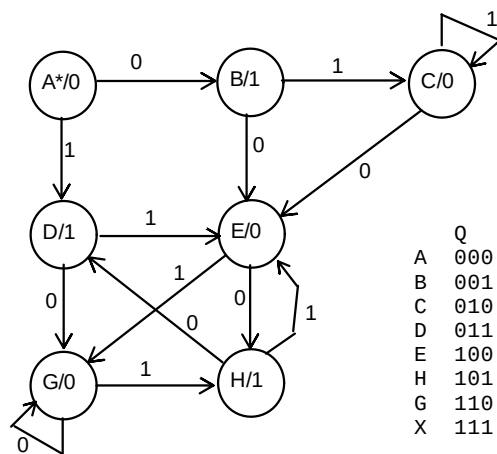
Sintetizzare una macchina sequenziale sincrona di Moore, il cui funzionamento è definito dal grafo seguente, individuando e semplificando gli eventuali stati equivalenti. Si usino esclusivamente JK-FF e porte logiche elementari (AND, OR, NOT).

* Stato di reset
Codificare con tutti 0



Come prima cosa osserviamo che gli stati G e J come pure D e F sono equivalenti, avendo la stessa evoluzione.

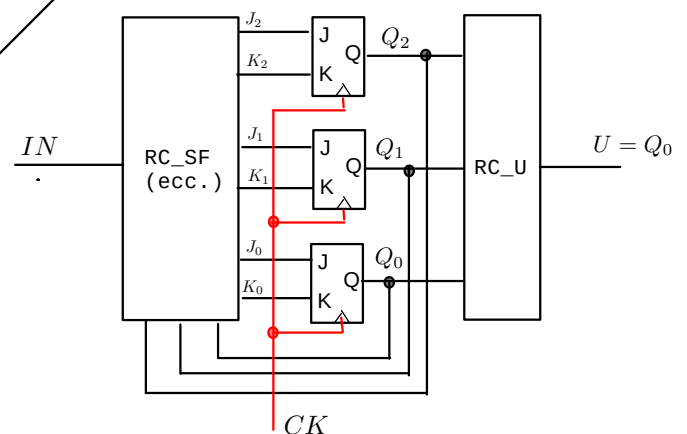
Posso ridisegnare il grafo semplificato, in cui sono sufficienti solo 3 bit di memoria non trasparente (JK-FF).



	Q ₂	Q ₁	Q ₀	U
A	0	0	0	0
B	0	0	1	1
C	0	1	0	0
D	0	1	1	1
E	1	0	0	0
H	1	0	1	1
G	1	1	0	0
X	1	1	1	1

Architettura

Con la codifica scelta, l'uscita coincide con una delle variabili di stato.



Mappa delle transizioni

Q_2, IN		Q_1, Q_0			
		00	01	11	10
00	00	001	100	110	100
01	01	011	010	100	010
11	11	110	100	---	101
10	10	101	011	---	110

Q	Q ⁺	J	K
0	0	0	-
0	1	1	-
1	0	-	1
1	1	-	0

Mappe di eccitazione

0	1	1	1
0	0	1	0
-	-	-	-
-	-	-	-

-	-	-	-
-	-	-	-
0	0	-	0
0	1	-	0

$$J_2 = \overline{IN}Q_0 + \overline{IN}Q_1 + Q_1Q_0$$

$$K_2 = \overline{IN}Q_0$$

0	0	-	-
1	1	-	-
1	0	-	-
0	1	-	-

-	-	0	1
-	-	1	0
-	-	-	1
-	-	-	0

$$J_1 = \overline{IN}Q_2 + \overline{IN}Q_0 + \overline{IN}Q_2Q_0$$

$$K_1 = \overline{IN}Q_2 + \overline{IN}Q_0 + \overline{IN}Q_2Q_0$$

1	-	-	0
1	-	-	0
0	-	1	-
1	-	-	0

-	1	1	-
-	1	1	-
-	1	-	-
-	0	-	-

$$J_0 = \overline{Q_2}Q_1 + \overline{IN}Q_1 + \overline{IN}Q_2Q_1$$

$$K_0 = \overline{Q_2} + \overline{IN}$$

Determinare la rappresentazione binary32 (IEEE 754-2008) del numero dato sotto, scegliendo il codice che minimizza l'errore relativo (in modulo) e calcolare tale errore. Determinare poi la rappresentazione binary32 anche del cubo del numero dato.

$$x = -2 \cdot 10^{-21}$$

Procedura: il dato è negativo, quindi il bit del segno (1) si ricava immediatamente; per l'esponente, si valuta la parte intera inferiore del logaritmo in base 2 e si trova il valore (-69) da rappresentare in traslazione, sommando 127 (58); il risultato deve essere maggiore di 0, altrimenti si sconfina nei sottonormali, e minore di 255, altrimenti si usa la rappresentazione di infinito; Infine alla parte rimanente (compresa tra 1 e "quasi" 2) si toglie 1, si moltiplica per 2^{23} e si arrotonda (1514912; 0x171DA0).

Procedura per il cubo: il cubo è un numero negativo con un modulo molto piccolo, molto minore dell'LSB dei sottonormali; quindi si rappresenta con -0.

$$x = -2 \cdot 10^{-21} \simeq -(2^{-69})(1 + 1514912 \cdot 2^{-23})$$

$$S=1 \quad E=-69+127=58=0x3A \quad T=1514912=0x171DA0$$

$$1 \ 001111010 \ 001 \ 0111 \ 0001 \ 1101 \ 1010 \ 0000$$

$$\epsilon_x = \left| \frac{\hat{x} - x}{x} \right| \simeq 3,17 \cdot 10^{-8}$$

$$x^3 = -8 \cdot 10^{-63} \quad 8 \cdot 10^{-63} \ll 2^{-149}$$

$$S=1 \quad E=0 \quad T=0$$

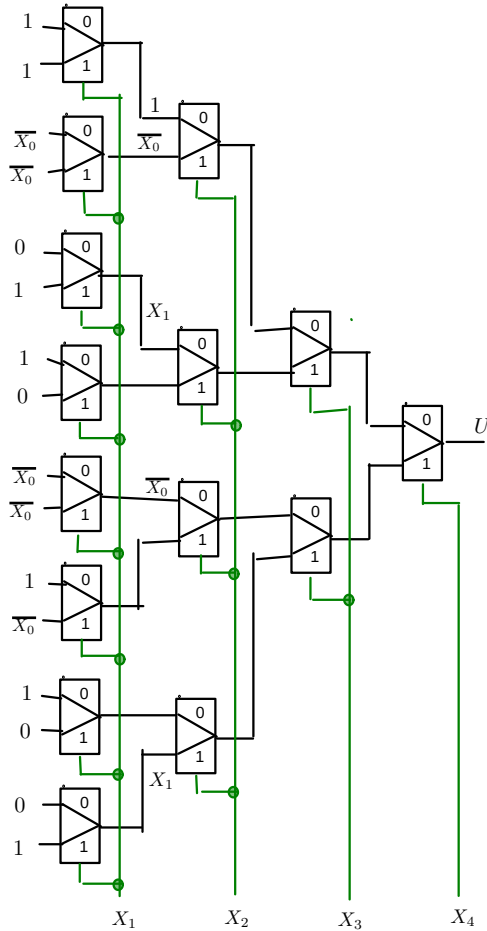
$$1 \ 000000000 \ 000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000$$

6

Realizzare la rete combinatoria dell'esercizio 2 facendo uso esclusivamente di mux 2:1, cercando di ridurre il numero, evitando l'uso di blocchi non necessari (duplicati, mux con ingressi identici, ecc.).

$\{1, 1, -, -, 1, -, 1, 0, -, 0, 1, 1, -, 1, 0, 0, 1, 0, 1, 0, -, 1, 1, 0, -, 1, -, 0, -, 0, -, 1\}$

$\{1-1, 1-1, 1-0, 1-0, 0-0, 1-1, 1-1, 0-0, 1-0, 1-0, 1-1, 1-0, 1-1, 0-0, 0-0, 1-1\}$



Dopo semplificazione:

