

Il testo deve essere riconsegnato nella cartellina. Non è ammessa la consultazione degli appunti e dei compiti precedenti. Si possono consultare i data sheet. **Non usare il colore rosso nello svolgimento.**

### ESERCIZIO N°1

5 punti

Determinare la rappresentazione nel formato binary 32 (IEEE754) dei seguenti numeri, arrotondando il valore effettivo al codice più vicino, e poi valutare l'errore assoluto con segno definito da  $(\hat{x} - x)$  usando, se l'errore non è nullo, 3 (strettamente) cifre significative

$$-160^{64}/6400^{32}$$

$$2^{-140}$$

$$-4 \log_{17}(13^{1000})$$

### ESERCIZIO N°2

8 punti

Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU, che divide per 8 (con eventuale arrotondamento al valore intero superiore) tutti gli elementi (interi su 2 byte, in C2, LSB first) di un vettore di dimensione  $n$  (contenuto in R16), posto in memoria all'indirizzo contenuto in Y. Nel caso in cui la dimensione  $n$  sia nulla, la subroutine esce senza effettuare modifiche.

### ESERCIZIO N°3

5 punti

Disegnare lo schema logico di una rete che esegue la funzione delle 5 variabili  $X_4, X_3, X_2, X_1$  e  $X_0$ , facendo riferimento alla forma normale (SP o PS) che presenta il minimo numero di letterali. La funzione, non completamente definita, è caratterizzata dalla seguente tabella di verità:

0, -, 0, -, 1, -, -, 1, 0, 1, 1, -, 1, 1, 0, 1, -, -, 0, 0, 0, -, 1, 0, -, 1, 1, -, 0, -, 1, 0.

### ESERCIZIO N°4

5 punti

Progettare un contatore sincrono  $\overline{U}/D$ , modulo 13, con abilitazione, con architettura ripple carry.

### ESERCIZIO N°5

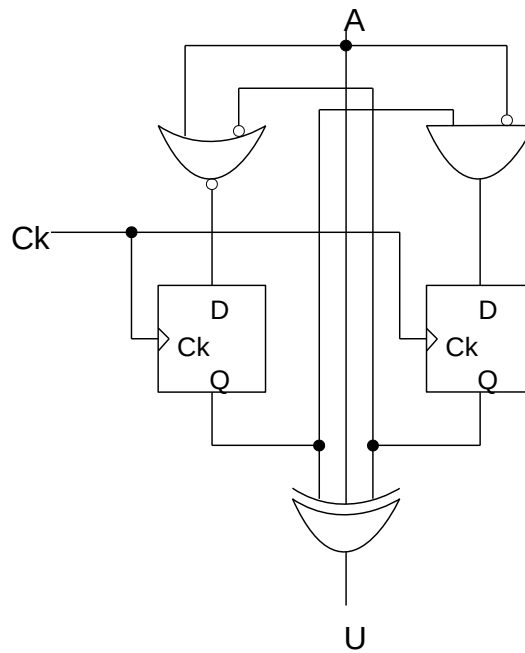
5 punti

Realizzare, usando esclusivamente multiplexer 2:1, la rete combinatoria proposta nell'esercizio 3. Evitare l'uso di multiplexer ridondanti.

# ESERCIZIO N°6

5 punti

Determinare tipologia architetturale e grafo di flusso della seguente macchina sequenziale sincrona.



1

Determinare la rappresentazione nel formato binary 32 (IEEE754) dei seguenti numeri, arrotondando il valore effettivo al codice più vicino, e poi valutare l'errore assoluto con segno definito da usando, se l'errore non è nullo, 3 (strettamente) cifre significative.

Conviene semplificare la frazione con le proprietà delle potenze. Si ha una potenza del 2. Il valore è rappresentabile esattamente come normalizzabile.

$$-\frac{160^{64}}{6400^{32}} = -\frac{2^{256}}{2^{192}} = -2^{64}$$

$$S = 0b1 \quad E = 64 + 127 = 191 = 0b10111111$$

$$T = 0 = 0b000\ 0000\ 0000\ 0000\ 0000\ 0000$$

$$\epsilon = \hat{x} - x = 0$$

Anche il secondo valore è una potenza di 2.

Visto il suo ordine di grandezza, si può rappresentare in modo esatto come NON normalizzabile.

$$2^{-140}$$

$$S = 0b0 \quad E = 0b0$$

$$T = 2^{(-140+149)} = 2^9 = 0b000\ 0000\ 0000\ 0010\ 0000\ 0000$$

$$\epsilon = \hat{x} - x = 0$$

Il terzo valore si esprime facilmente usando le proprietà dei logaritmi. Si può rappresentare come normalizzabile.

$$-4 \log_{17}(13^{1000}) = -4000 \frac{\ln(13)}{\ln(17)} = -3621,25833248... = -2^{11} \cdot 1,76819254515...$$

$$S = 0b1 \quad E = 11 + 127 = 138 = 0b10001010$$

$$T = 6444066 = 0b110\ 0010\ 0101\ 0100\ 0010\ 0010$$

$$\epsilon = \hat{x} - x \simeq 3,17 \cdot 10^{-5}$$

```
/* Realizzare un sottoprogramma per il microcontrollore AVR XMEGA256A3BU,
che divide per 8 (con eventuale arrotondamento al valore intero superiore)
tutti gli elementi (interi su 2 byte, in C2, LSB first)
di un vettore di dimensione n (contenuto in R16),
posto in memoria all'indirizzo contenuto in Y.
Nel caso in cui la dimensione n sia nulla,
la subroutine esce senza effettuare modifiche. */
```

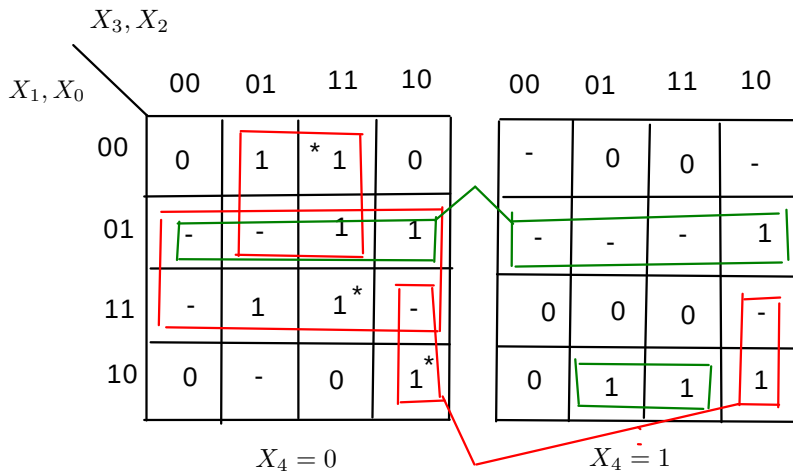
```
div8_c2:
  tst R16
  breq end_div8
  push R16
  push R24
  push R25
  push YL
  push YH
  loop:
    ld R24,Y //carica il valore da dividere senza modificare Y
    ldd R25,Y+1
    asr R25 //divide per 2 con segno, 3 volte
    ror R24
    asr R25
    ror R24
    asr R25
    ror R24
    brcc poi
    adiw R25:R24,1 //arrotonda al successivo se >=0,5
  poi:
    st Y+,R24 //memorizza e incrementa
    st Y+,R25
  dec R16
  brne loop
  pop YH
  pop YL
  pop R25
  pop R24
  pop R16
end_div8:
  ret
```

3

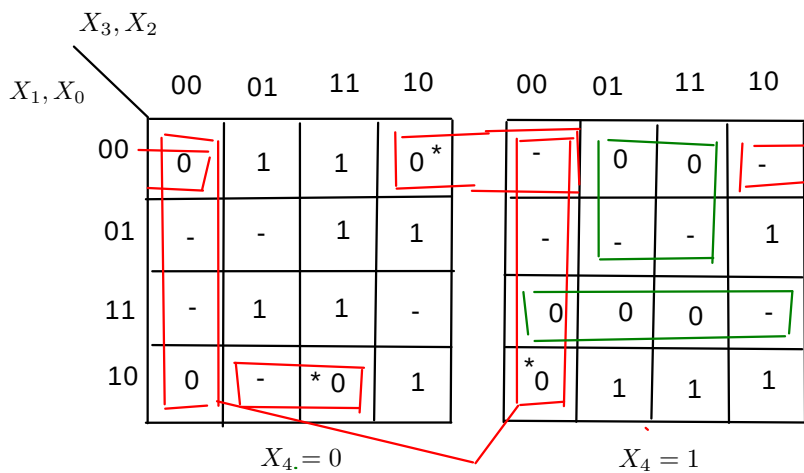
Disegnare lo schema logico di una rete che esegue la funzione delle 5 variabili  $X_4, X_3, X_2, X_1$  e  $X_0$ , facendo riferimento alla forma normale (SP o PS) che presenta il minimo numero di letterali.

La funzione, non completamente definita, è caratterizzata dalla seguente tabella di verità:

0, -, 0, -, 1, -, -, 1, 0, 1, 1, -, 1, 1, 0, 1, -, -, 0, 0, 0, -, 1, 0, -, 1, 1, -, 0, -, 1, 0.



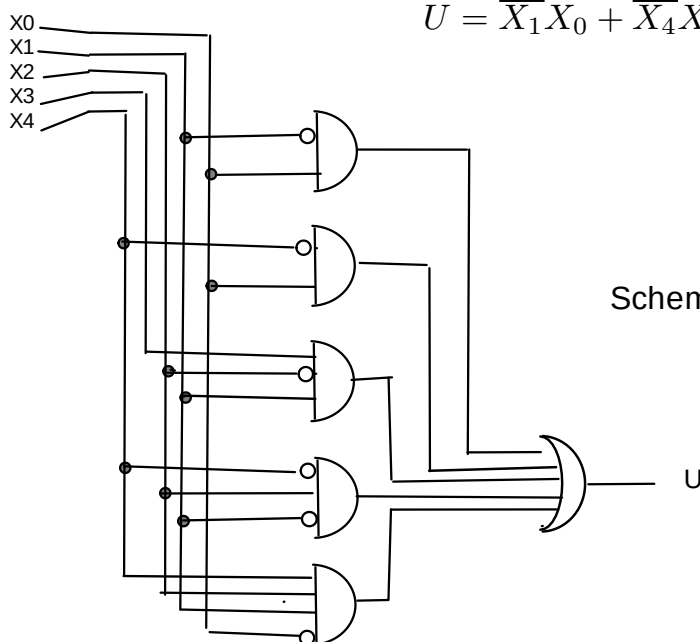
SP; rossi essenziali  
14 letterali  
(2 ordine 3, 2 ordine 2,  
1 ordine 1)



PS; rossi essenziali  
15 letterali  
(1 ordine 3, 3 ordine 2,  
1 ordine 1)

La soluzione SP è migliore.

$$U = \overline{X_1}X_0 + \overline{X_4}X_0 + X_3\overline{X_2}X_1 + \overline{X_4}X_2\overline{X_1} + X_4X_2X_1\overline{X_0}$$



Schema logico

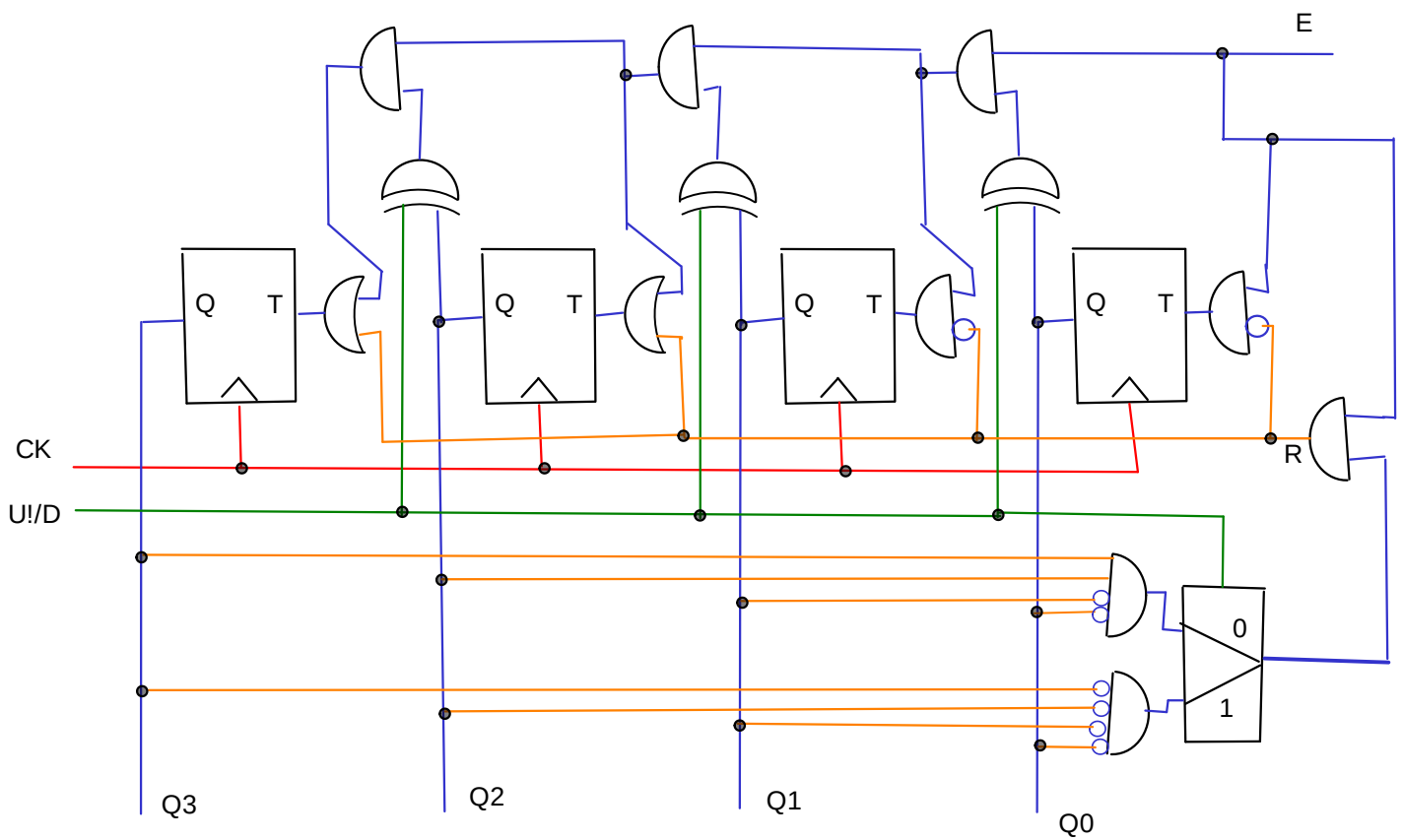
4

Progettare un contatore sincrono U!/D, modulo 13, con abilitazione, con architettura ripple carry.

Esamino le sequenze di conteggio:

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

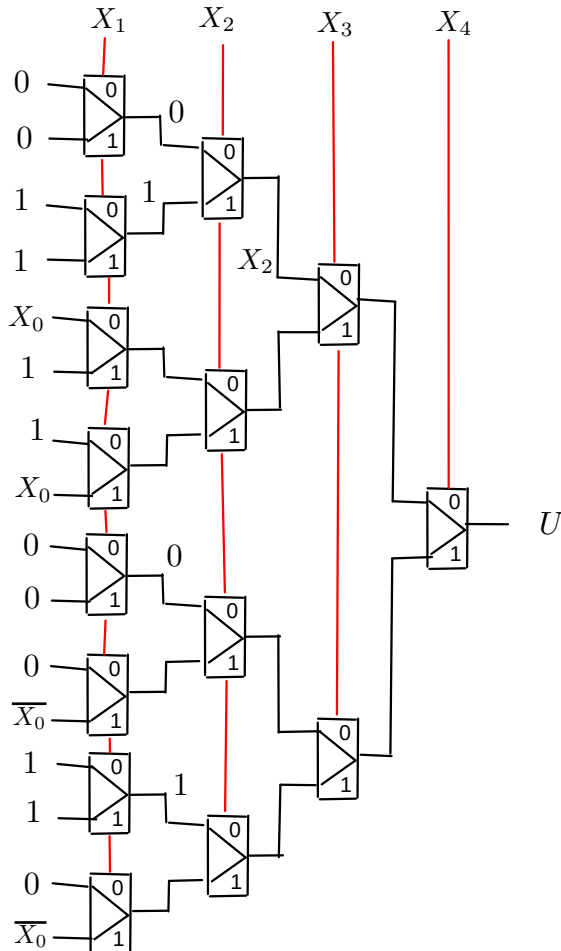
Lo stesso operatore FFBB va bene per entrambe le direzioni.



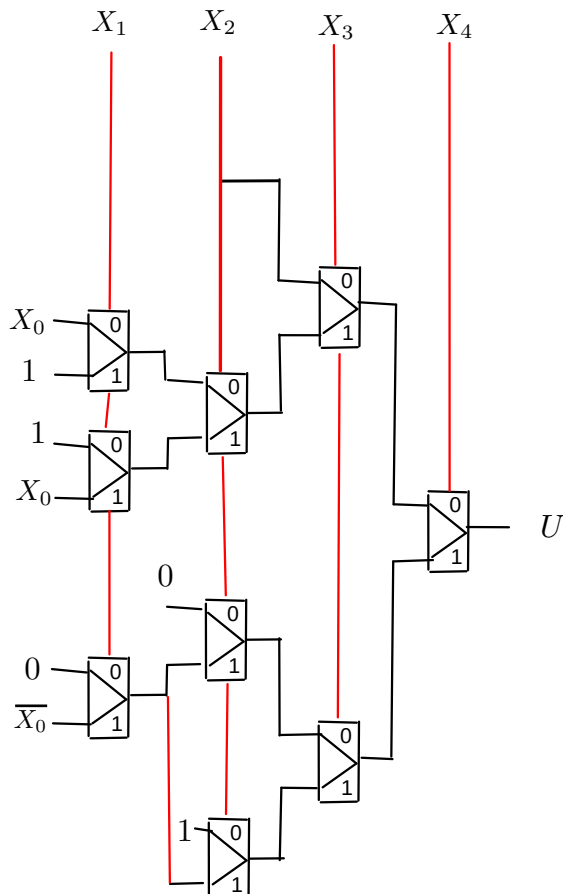
5

Realizzare, usando esclusivamente multiplexer 2:1, la rete combinatoria proposta nell'esercizio 3.  
Evitare l'uso di multiplexer ridondanti.

0, -, 0, -, 1, -, -, 1, 0, 1, 1, -, 1, 1, 0, 1, -, -, 0, 0, 0, -, 1, 0, -, 1, 1, -, 0, -, 1, 0.



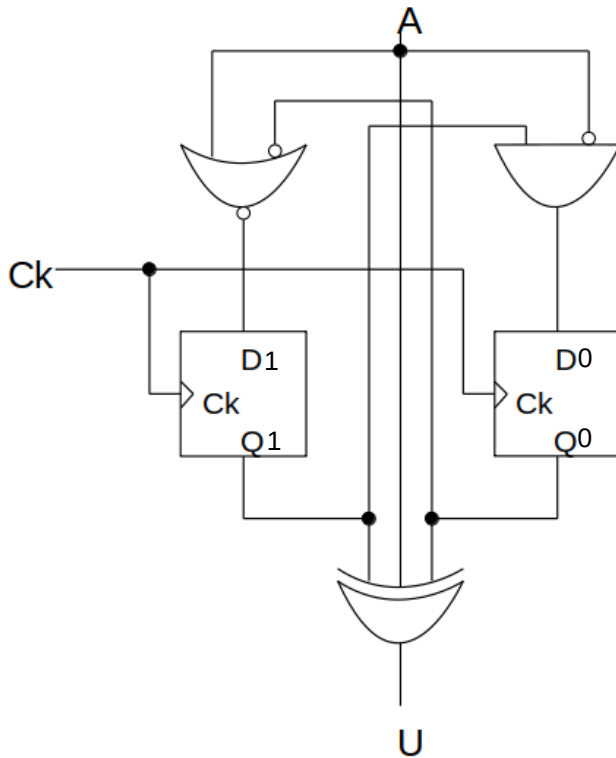
Sfruttando i don't care, posso arrivare a configurazioni semplificate



Eliminando i mux ridondanti ottengo una ulteriore semplificazione.

6

Determinare tipologia architetturale e grafo di flusso della seguente macchina sequenziale sincrona.



Questa è una macchina di Mealy, in quanto l'uscita dipende dallo stato (l'uscita dei D-FF) e dall'ingresso A

$$U = A \oplus Q_1 \oplus Q_0$$

$$D_1 = Q_1^+ = Q_0 \bar{A}$$

$$D_0 = Q_0^+ = Q_1 \bar{A}$$

St	Q1	Q0	A	U	Q1+	Q0+	St+
E	0	0	0	0	0	0	E
			1	1	0	0	E
B	0	1	0	1	1	0	C
			1	0	0	0	E
C	1	0	0	1	0	1	B
			1	0	0	0	E
D	1	1	0	0	1	1	D
			1	1	0	0	E

