

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

ESERCIZIO N°1

8 punti

Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che valuta l'opposto di un numero intero a 8 byte in complemento a 2 contenuto in memoria all'indirizzo puntato da X (a partire dal byte meno significativo) e memorizza il risultato nella stessa posizione. Se (e solo se) l'operazione di cambio segno dà overflow, il valore del flag V al ritorno deve essere vero.

ESERCIZIO N°2

5 punti

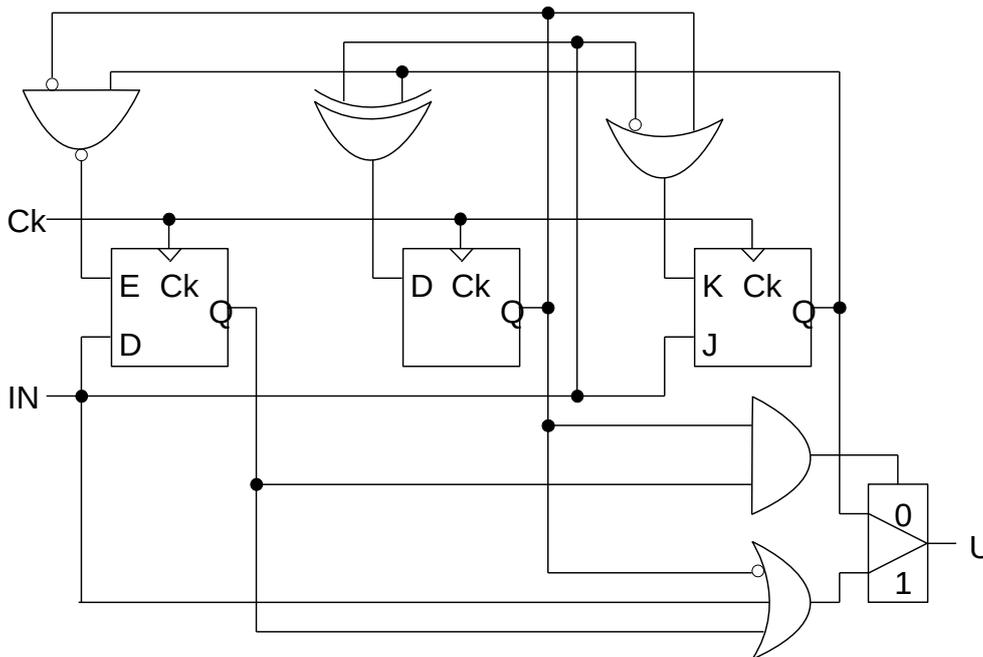
Disegnare lo schema logico in forma NOR-NOR a minimi letterali della funzione combinatoria di 5 variabili (X_4, X_3, X_2, X_1, X_0) definita dalla seguente tabella di verità.

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

ESERCIZIO N°3

5 punti

Determinare la tipologia architetturale e il grafo delle transizioni della seguente rete sequenziale sincrona.



ESERCIZIO N°4

5 punti

Disegnare un circuito a porte logiche in grado di generare i segnali di clock necessari al funzionamento di un flip-flop master-slave, graficando le uscite generate. Si ha a disposizione un'onda quadra come segnale di partenza (nel grafico si consideri per ciascuna porta elementare il ritardo di un quadretto e l'onda quadra abbia periodo di 16 quadretti).

ESERCIZIO N°5

5 punti

Esprimere come valori frazionali [1.7] in C2 con la migliore approssimazione possibile i seguenti numeri razionali: $3/7$; $-97/98$

Mostrare poi il risultato del prodotto tra i due numeri così come sarebbe ottenuto dalle operazioni FMUL, FMULS e FMULSU operanti sulle rappresentazioni trovate (nel caso non sia valida la proprietà commutativa, mostrare i 2 possibili risultati)

ESERCIZIO N°6

5 punti

Disegnare lo schema logico di un registro universale a 8 bit (più un bit aggiuntivo per il carry) in grado di implementare funzionalità simili ai comandi (di un microcontrollore con architettura AVR) indicati di seguito, sulla base del valore di un segnale di controllo a 3 bit.

0:ASR, 1:LSL, 2:ROR, 3:LSR, 4:ROL, 5:LDI, 6:CLR, 7:COM

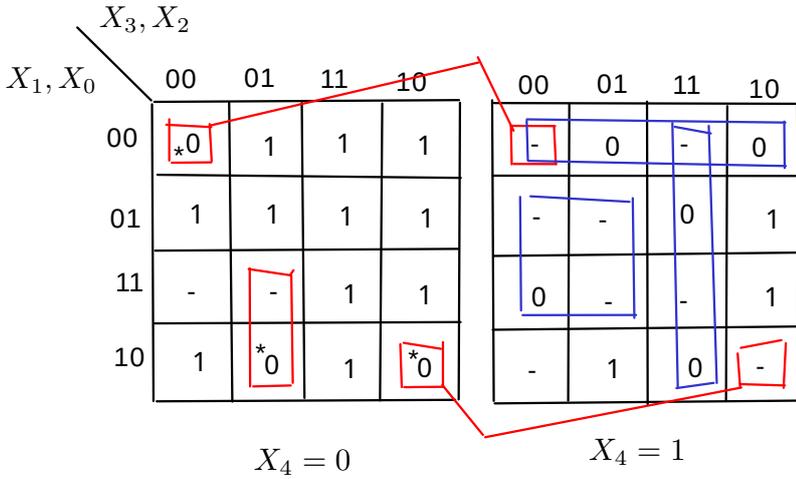
```
/*Realizzare una subroutine per il microcontrollore AVR XMEGA256A3BU che valuta
l'opposto di un numero intero a 8 byte in complemento a 2 contenuto in memoria
all'indirizzo puntato da X (a partire dal byte meno significativo)
e memorizza il risultato nella stessa posizione.
Se (e solo se) l'operazione di cambio segno dà overflow,
il valore del flag V al ritorno deve essere vero.*/
```

```
neg8_X:
  push R16
  push R17
  push R18
  ldi R16,8
  clc
  loop:
    ldi R18,0
    ld R17,X
    sbc R18,R17          //esegue l'opposto
    st X+,R18
    dec R16              //lascia inalterato C
    brne loop
  sbiw XH:XL,8          //ripristina X
  eor R17,R18           //confronta segni MSByte e pone V=0
  sbrs R17,7            //se segni diversi V resta 0
  sev                   //setta V
  pop R18
  pop R17
  pop R16
  ret
```

2

Disegnare lo schema logico in forma NOR-NOR a minimi letterali della funzione combinatoria di 5 variabili definita dalla seguente tabella di verità.

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



Parto da sintesi ottima PS

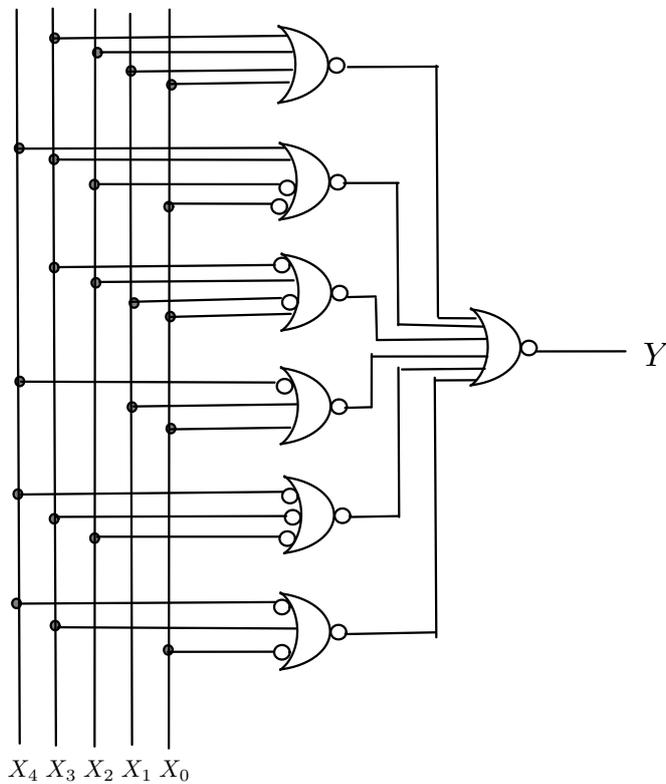
6 implicati di cui
3 essenziali
(prima riga dell'espressione)

21 letterali

$$Y = (X_3 + X_2 + X_1 + X_0)(X_4 + X_3 + \bar{X}_2 + \bar{X}_0)(\bar{X}_3 + X_2 + \bar{X}_1 + X_0) \cdot$$

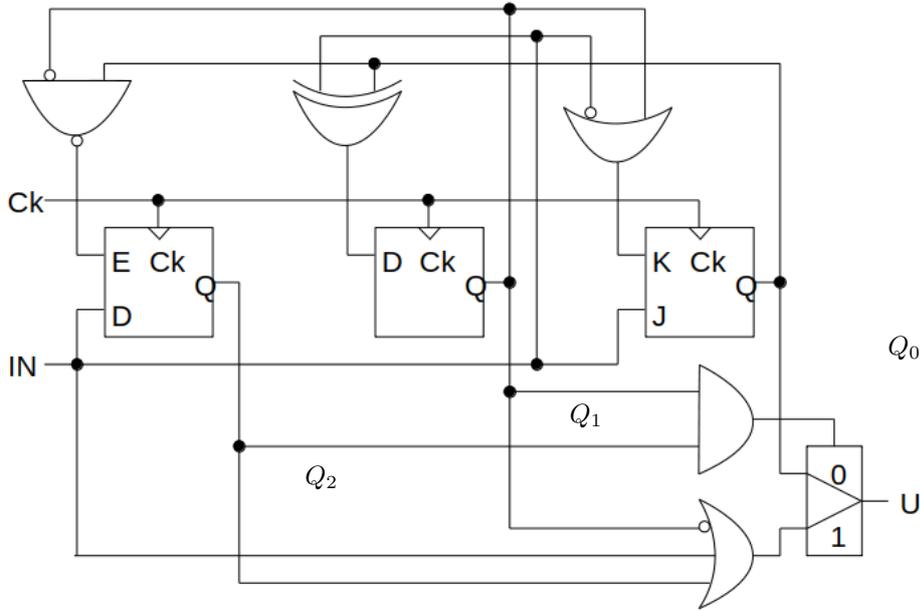
$$(\bar{X}_4 + X_1 + X_0)(\bar{X}_4 + \bar{X}_3 + \bar{X}_2)(\bar{X}_4 + X_3 + \bar{X}_0)$$

Schema NOR-NOR



3

Determinare la tipologia architetturale e il grafo delle transizioni della seguente rete sequenziale sincrona.



$$U = (\overline{Q_2} + \overline{Q_1})Q_0 + Q_2Q_1(Q_2 + \overline{Q_1} + IN) = \overline{Q_2}Q_0 + \overline{Q_1}Q_0 + Q_2Q_1$$

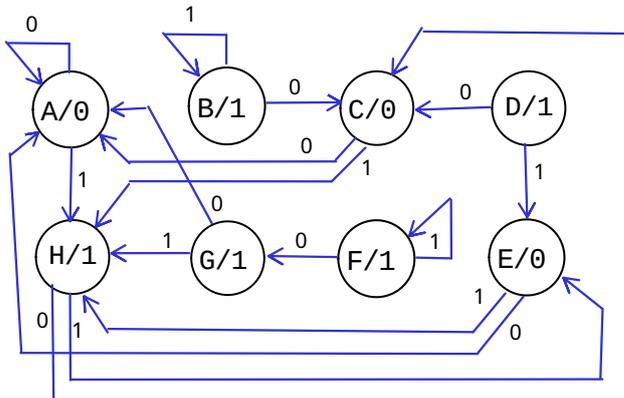
L'analisi della funzione di uscita evidenzia che l'uscita U dipende solo dallo stato presente. Quindi si tratta di una macchina di MOORE.

	Q2	Q1	Q0	IN	D2	E2	Q2+	D1=Q1+	J0	K0	Q0+	SF
A/0	0	0	0	0	0	1	0	0	0	1	0	A
				1	1	1	1	1	1	0	1	H
B/1	0	0	1	0	0	0	0	1	0	1	0	C
				1	1	0	0	0	1	0	1	B
C/0	0	1	0	0	0	1	0	0	0	1	0	A
				1	1	1	1	1	1	1	1	H
D/1	0	1	1	0	0	1	0	1	0	1	0	C
				1	1	1	1	0	1	1	0	E
E/0	1	0	0	0	0	1	0	0	0	1	0	A
				1	1	1	1	1	1	0	1	H
F/1	1	0	1	0	0	0	1	1	0	1	0	G
				1	1	0	1	0	1	0	1	F
G/1	1	1	0	0	0	1	0	0	0	1	0	A
				1	1	1	1	1	1	1	1	H
H/1	1	1	1	0	0	1	0	1	0	1	0	C
				1	1	1	1	0	1	1	0	E

$$D_2 = IN; \quad E_2 = Q_1 + \overline{Q_0}$$

$$D_1 = IN \oplus Q_0$$

$$J_0 = IN; \quad K_0 = \overline{IN} + Q_1$$



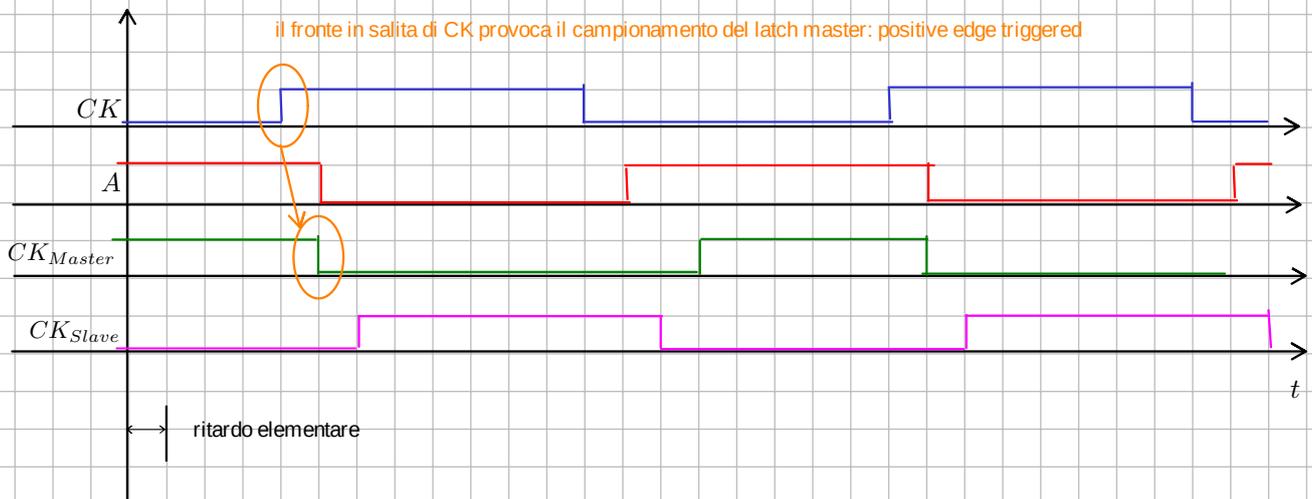
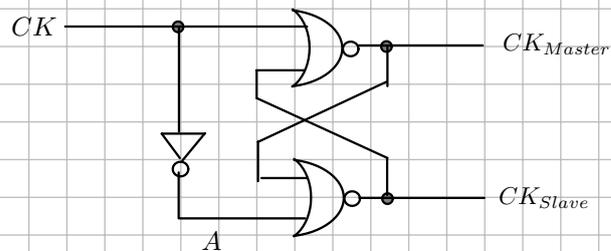
Grafo delle transizioni

4

Disegnare un circuito a porte logiche in grado di generare i segnali di clock necessari al funzionamento di un flip-flop master-slave, graficando le uscite generate.

Si ha a disposizione un'onda quadra come segnale di partenza

(nel grafico si consideri per ciascuna porta elementare il ritardo di un quadretto e l'onda quadra abbia periodo di 16 quadretti).



Esprimere come valori frazionali [1.7] in C2,
 con la migliore approssimazione possibile i seguenti numeri razionali: $3/7$; $-97/98$
 Mostrare poi il risultato del prodotto tra i due numeri
 così come sarebbe ottenuto dalle operazioni FMUL, FMULS e FMULSU
 operanti sulle rappresentazioni trovate
 (nel caso non sia valida la proprietà commutativa, mostrare i 2 possibili risultati)

$$A = 3/7 \simeq 2^{-7} \cdot 54,8571$$

Si approssima al valore superiore: 0.0110111

$$B = -97/98 \simeq -1 + 2^{-7} \cdot 1,3061$$

Si approssima al valore inferiore: 1.0000001

Consideriamo R16=A e R17=B

Esaminiamo FMUL R16,R17

Non c'è alcun motivo per cui non sia rispettata la proprietà commutativa.

Il risultato sarà in R1:R0 e si ottiene eseguendo il prodotto e lo scorrimento di 1 a sx

$$R1:R0 = 0x376E = 00110111 01101110$$

Esaminiamo ora FMULS R16,R17

Anche qui non c'è alcun motivo per cui non sia rispettata la proprietà commutativa.

Il risultato sarà in R1:R0 e si ottiene eseguendo il prodotto con segno e lo scorrimento di 1 a sx

$$R1:R0 = 0xC96E = 11001001 01101110$$

Esaminiamo ora FMULSU R16,R17

Qui invece non può in genere essere rispettata la proprietà commutativa.

Il risultato sarà in R1:R0 e si ottiene eseguendo il prodotto considerando il segno di R16

e poi lo scorrimento di 1 a sx

Visto che R16 è positivo, si avrà lo stesso risultati di FMUL

$$R1:R0 = 0x376E = 00110111 01101110$$

Infine FMULSU R17,R16 darà lo stesso risultato della FMULS

$$R1:R0 = 0xC96E = 11001001 01101110$$

Nota: il datasheet descrive il comportamento delle moltiplicazioni frazionali in modo non completamente determinato. In particolare non è chiaro se lo shift a sx finale riguarda solo R1 o la coppia completa. Il simulatore MPLAB mostra R1 ottenuto come atteso dallo shift (e nel bit 0 di R1 finisce il bit 7 di R0 prima dello scorrimento), mentre R0 assume un valore non coerente con lo shift. Ai fini della correzione del compito è stato considerato solo il valore di R1.

