

Il testo va riconsegnato

ESERCIZIO N°1

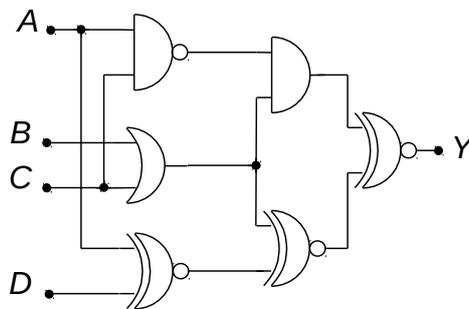
8 punti

Realizzare una subroutine per un microcontrollore della famiglia XMEGA AVR che modifica le celle di memoria nell'intervallo di indirizzi da 0x2222 a 0x2333, compresi gli estremi: ogni valore maggiore di 127 viene sostituito con un valore pari a 1/8 del precedente (divisione intera), mentre ogni valore minore o uguale a 127 viene sostituito con un valore pari al doppio del precedente.

ESERCIZIO N°2

6 punti

Determinare l'espressione logica in forma PS minima della rete seguente.



ESERCIZIO N°3

6 punti

Progettare, facendo uso di un flip-flop di tipo JK, un nuovo tipo di flip-flop sincrono con tre ingressi L, M e N caratterizzato dal seguente comportamento:

- il segnale L è un reset, attivo basso, con priorità maggiore degli altri due segnali;
- il segnale M, attivo basso, porta l'uscita a 1 e ha priorità maggiore di N;
- il segnale N, attivo basso, commuta l'uscita.

Se nessun segnale è attivo, l'uscita mantiene il valore precedente.

ESERCIZIO N°4

7 punti

Disegnare lo schema logico di un sequenziatore con contatore sincrono (dotato della possibilità di caricamento parallelo) che implementi microcodice specificato nel seguito. Si usi una ROM delle minime dimensioni possibili.

- A: IF J THEN D ELSE B; OP = 101111
- B: IF L THEN A ELSE C; OP = 011100
- C: IF K THEN B ELSE F; OP = 101111
- D: IF M THEN E ELSE H; OP = 010000
- E: IF M THEN G ELSE D; OP = 111111
- F: IF J THEN C ELSE A; OP = 011100
- G: IF L THEN H ELSE E; OP = 100011
- H: IF K THEN F ELSE G; OP = 010000

ESERCIZIO N°5

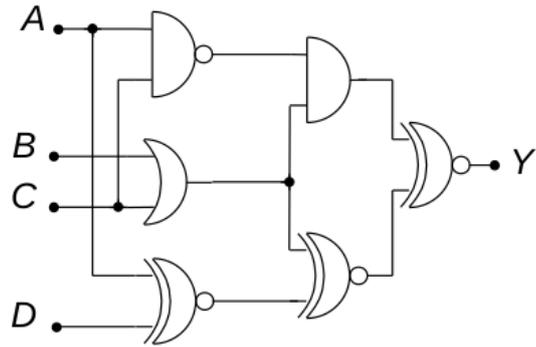
6 punti

Determinare la tensione di ingresso per cui un invertitore CMOS a vuoto assorbe la massima corrente dall'alimentazione e quale è questo valore di corrente.

Si ha $V_{DD} = 3\text{ V}$, $V_{Tn} = 0,6\text{ V}$; $V_{Tp} = -0,5\text{ V}$ e $k_n = -k_p = 4\text{ mA/V}^2$.

```
Modif:
push R16
push XL
push XH
ldi XL,low(0x2222)
ldi XH,high(0x2222)
loop:
ld R16,X
cpi R16,128
brcc div8 //maggiore strettamente di 127
lsl R16 //raddoppia
rjmp endloop
div8:
lsr R16 //divide 3 volte per 2
lsr R16
lsr R16
endloop:
st X+,R16 //salva e avanza
cpi XL,low(0x2333+1)
brne loop
cpi XH,high(0x2333+1)
brne loop
pop XH
pop XL
pop R16
ret
```

2



L'espressione della funzione da realizzare è la seguente:
 (usando le proprietà della XOR e in particolare

- la XOR è associativa
- la possibilità di riportare la negazione in uscita su uno degli ingressi
- la proprietà distributiva della AND sulla XOR
- la XOR con 1 equivale al complemento)

$$Y = (AC)'(B+C) \text{ xor } (B+C) \text{ xor } A \text{ xor } D' =$$

$$= (B+C)[(AC)' \text{ xor } 1] \text{ xor } A \text{ xor } D' = (B+C)AC \text{ xor } A \text{ xor } D' = AC \text{ xor } A \text{ xor } D' =$$

$$= AC' \text{ xor } D' = AC'D + A'D' + CD'$$

Riportando in mappa

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

$$Y = (A'+C+D)(A+D')(C'+D')$$

Possiamo descrivere il funzionamento del nuovo Flip-Flop con la seguente tabella (limitata qui alle righe corrispondenti al fronte in salita del clock).

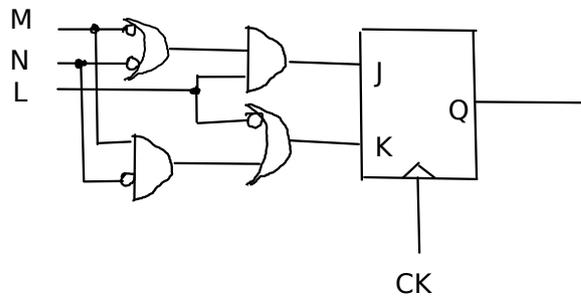
| L | M | N | Q+ |
|---|---|---|----|
| 0 | - | - | 0 |
| 1 | 0 | - | 1 |
| 1 | 1 | 0 | Q' |
| 1 | 1 | 1 | Q |

Possiamo individuare i valori di J e K relativi a ogni combinazione di ingressi

| L \ M,N | | M,N | | | |
|---------|---|-----|----|----|----|
| | | 00 | 01 | 11 | 10 |
| L | 0 | 01 | 01 | 01 | 01 |
| | 1 | 10 | 10 | 00 | 11 |

JK

$$J = L(M' + N')$$

$$K = L' + MN'$$


Individuo la sequenza ciclica e completa (caso flag vero)
A, D, E, G, H, F, C, B, A, etc

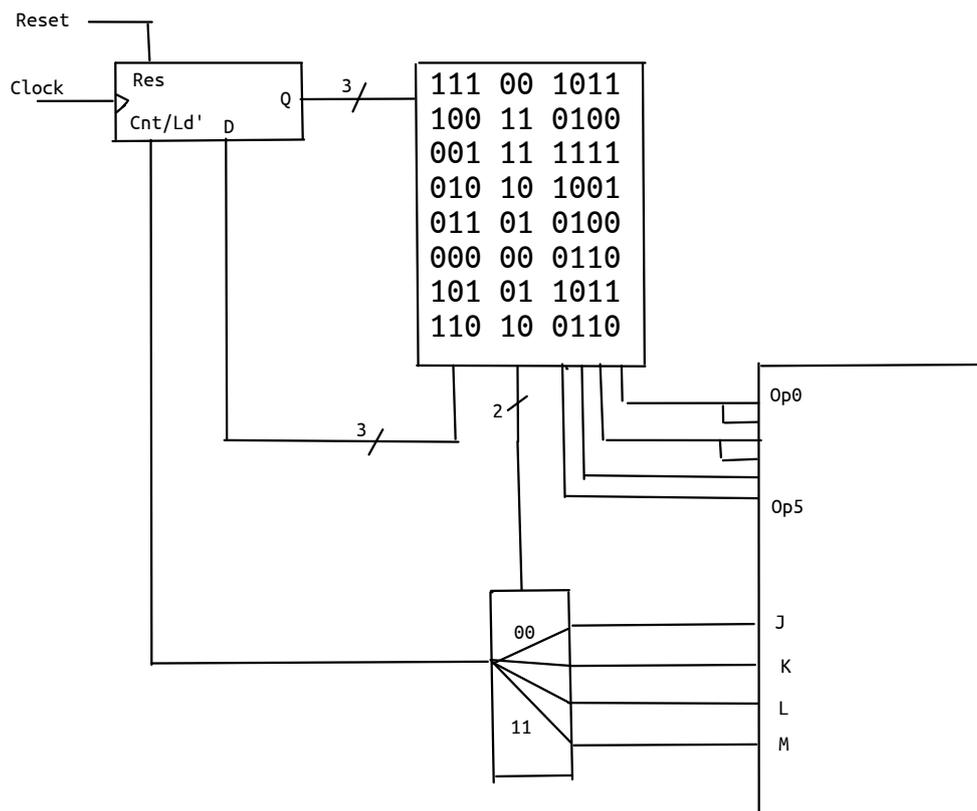
Codifico gli stati in accordo all'ordine di sequenza e riscrivo il ucodice;
codifico i flag: J(00), K(01), L(10), M(11)

```

A(000): IF J(00) THEN D(001) ELSE B(111); OP = 101111
D(001): IF M(11) THEN E(010) ELSE H(100); OP = 010000
E(010): IF M(11) THEN G(011) ELSE D(001); OP = 111111
G(011): IF L(10) THEN H(100) ELSE E(010); OP = 100011
H(100): IF K(01) THEN F(101) ELSE G(011); OP = 010000
F(101): IF J(00) THEN C(110) ELSE A(000); OP = 011100
C(110): IF K(01) THEN B(111) ELSE F(101); OP = 101111
B(111): IF L(10) THEN A(000) ELSE C(110); OP = 011100

```

Ora si può disegnare l'architettura a contatore, con il contenuto della ROM.
Si può anche osservare che i 4 bit meno significativi dell'OpCode sono uguali a coppie (quindi 2 colonne si possono omettere).



5

Tensione di ingresso di invertitore CMOS per la massima corrente da VDD
Si ha per nMOS e pMOS entrambi saturi (nel caso in cui uno dei due sia triodo, una variazione della $|V_{GS}|$ di quello saturo porterà a un aumento della I_{DD}).

Trovo la tensione impostando la condizione di eguaglianza
 $I_{DSn} = -I_{DSp}$

$$k_n/2 (V_{IN} - V_{Tn})^2 = -k_p/2 (V_{IN} - V_{DD} - V_{Tp})^2$$

faccio la radice di ambo i membri

$$\sqrt{-k_n/k_p} (V_{IN} - V_{Tn}) = \pm (V_{IN} - V_{DD} - V_{Tp})$$

sostituendo (è accettabile solo la soluzione con il -)

$$V_{IN} = (V_{DD} + V_{Tp} + V_{Tn})/2 = 1,55 \text{ V}$$

Per questa tensione $I_{DD} = 1,805 \text{ mA}$