

ESERCIZIO N°1

6 punti

Disegnare in tecnologia CMOS una rete che esegua la funzione

$$\overline{A + (B + CD)}$$

ESERCIZIO N°2

6 punti

Determinare, se possibile, una soluzione circuitale con l'uso di una resistenza che permetta di collegare 6 porte di tipo *A* all'uscita di una porta di tipo *B*. L'alimentazione è 5 V e i parametri delle due famiglie sono:

$$V_{ILA} = 2 \text{ V}; I_{ILA} = -0.1 \text{ mA}; V_{IHA} = 3 \text{ V}; I_{IHA} = 0.2 \text{ mA};$$

$$V_{OLA} = 1 \text{ V}; I_{OLA} = 2 \text{ mA}; V_{OHA} = 4 \text{ V}; I_{OHA} = -3 \text{ mA}.$$

$$V_{ILB} = 2 \text{ V}; I_{ILB} = -0.5 \text{ mA}; V_{IHB} = 3 \text{ V}; I_{IHB} = 0.1 \text{ mA};$$

$$V_{OLB} = 1 \text{ V}; I_{OLB} = 2 \text{ mA}; V_{OHB} = 4 \text{ V}; I_{OHB} = -1 \text{ mA}.$$

ESERCIZIO N°3

7 punti

Disegnare, usando se possibile flip-flop *JK*, una macchina di Moore senza ingressi che generi in uscita un'onda rettangolare con ciclo di lavoro 2/3 e periodo pari a 3 cicli di clock. La macchina deve funzionare correttamente qualunque sia il valore iniziale delle uscite *Q* dei flip-flop.

ESERCIZIO N°4

6 punti

Sintetizzare un sequenziatore, indicando il contenuto della ROM, che fornisca alla parte operativa un numero a due bit secondo la seguente legge:

- 1) se il valore corrente dell'uscita è pari e il flag *F* è vero, l'uscita deve essere uguale al valore attuale incrementato di 1 (modulo 4), se il flag *F* è falso l'uscita invece deve essere nulla.
- 2) se il valore corrente dell'uscita è dispari e il flag *G* è vero, l'uscita deve essere uguale a 3, se il flag *G* è falso l'uscita deve essere uguale al valore attuale incrementato di 2 (modulo 4).

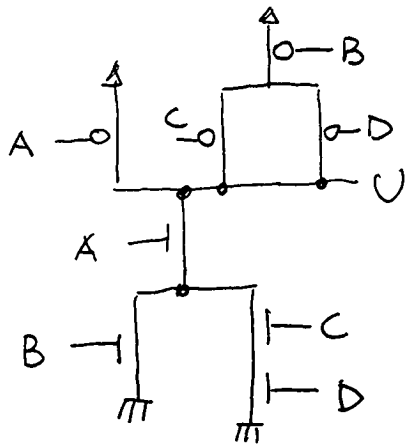
ESERCIZIO N°5

8 punti

Scrivere un programma per il microcontrollore AT90S8515 che ponga (continuamente) nella locazione di memoria puntata da *X* il numero degli ingressi della porta *B* diversi dai loro corrispondenti nella porta *D*. I valori logici in ingresso alle porte sono impostati tramite pulsanti collegati tra pin e gnd. Che valore dovrebbe contenere il registro *X* perché questo dato appaia in uscita sulla porta *A*? Inizializzare correttamente le porte *A*, *B* e *D*.

① Trasforma

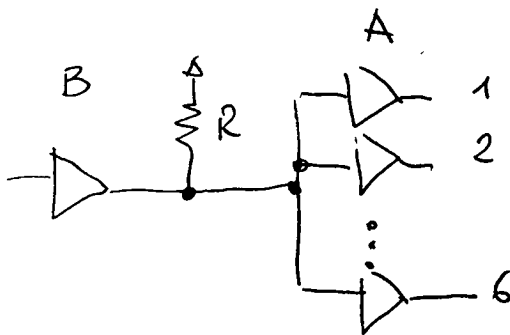
$$U = \overline{A + (B + CD)} = \overline{A \cdot (B + CD)} \quad \text{e realizzato}$$



② Condizioni sulle tensioni : OK

Condizione sulle correnti : $6 I_{IHA} > |I_{OHB}|$ (VIOLAZ.)

Si può provare a rendere disponibile più corrente sul livello alto con un pull-up.



$$2.9k \leq R \leq 5k\Omega$$

è possibile

Condizioni per R

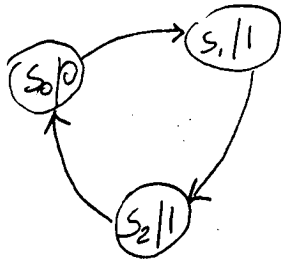
livello alto: $|I_{OHB}| + \frac{V_{CC} - V_{OH B}}{R} > 6 I_{IHA}$

$$R < \frac{V_{CC} - V_{OH B}}{6 I_{IHA} - |I_{OHB}|} = 5k\Omega$$

livello basso: $I_{OLB} > |6 I_{ILA}| + \frac{V_{CC} - V_{OL B}}{R}$

$$R > \frac{V_{CC} - V_{OL B}}{I_{OLB} - |6 I_{ILA}|} = 2.9k\Omega$$

③ Partiamo dal grafo



Codifica

$s_0 : 00$
 $s_1 : 01$
 $s_2 : 10$

per lo stato rimanente ($s_3 = 11$), si richiede che NON sia stabile

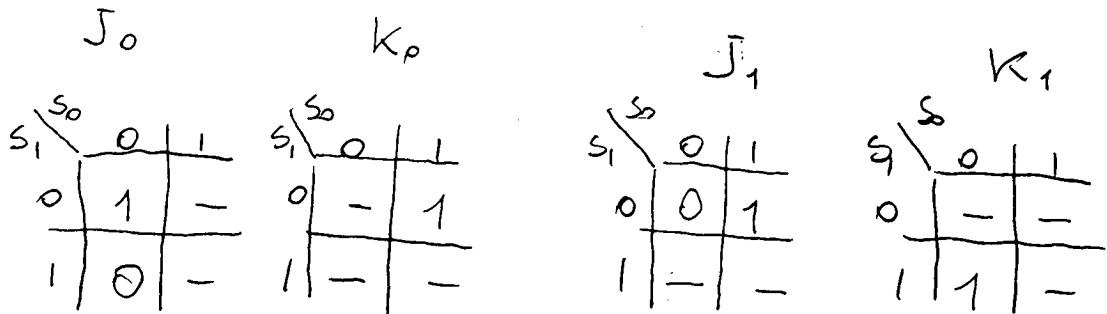
Tabella di flusso

att.	fut.	att
00	01	0
01	10	1
10	00	1
11	≠11	-

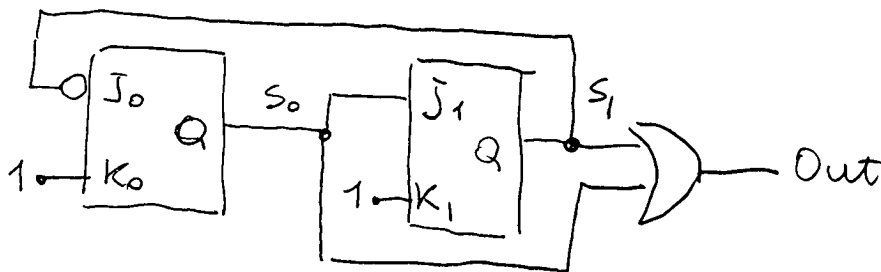
Tabella transizioni JK

TRANS	J	K
0 → 0	0	0
0 → 1	1	-
1 → 0	-	1
1 → 1	-	0

rete



$J_0 = \bar{s}_1$ $K_0 = 1$ $J_1 = s_0$ $K_1 = 1$ $Out = s_0 + s_1$



lo stato 11 va in 00

④ Sequenziatore a 4 stati -

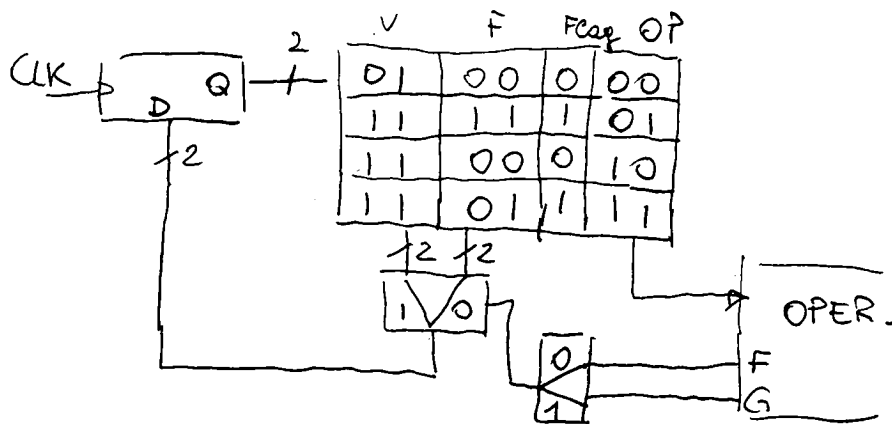
Descrizione e microcodice

S0 : IF F S1 ELSE S0 ; OP=0
 S1 : IF G S3 ELSE S3 ; OP=1
 S2 : IF F S3 ELSE S0 ; OP=2
 S3 : IF G S3 ELSE S1 ; OP=3

Flag F → 0
 G → 1

Stati S0 → 00
 S1 → 01
 S2 → 10
 S3 → 11

Architettura con ROM



⑤ Programma

```
.include "8515DEF.INC"
main:  clr  r16          ; utili per l'inizializzazione
       ser  r17
       out  DDRB, R16  ; porte B e D ingressi con pull-up
       out  DDRD, R16
       out  PORTB, R17
       out  PORTD, R17
       out  DDRA, R17 ; porta A uscita
ciclo: clr  R17
       ldi  R18, 8     ; contatore di ciclo
       in  R19, PINB  ; acquisisce i tasti
       in  R20, PIND
       eor  R19, R20  ; trova i tasti diversi
loop:  lsr  R19        ; guarda il bit meno significativo
       adc  R17, R16,  ; incrementa R17 se carry è 1
       dec  R18       ; decrementa il contatore
       brne loop     ; ripete 8 volte
       st  X, R17    ; memorizza (se X=0x003B R16 va in PORTA)
       rjmp ciclo
```