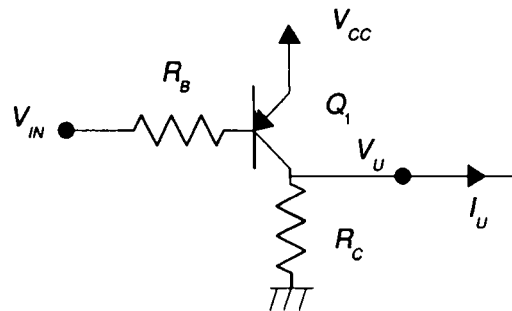


ESERCIZIO N°1

6 punti

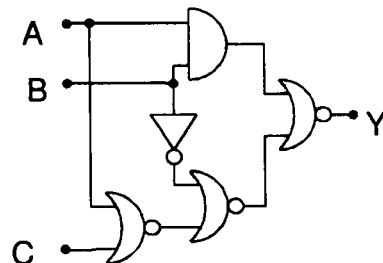
Determinare la caratteristica V_U-I_U nella porta seguente quando $V_{IN} = 0$. Per il transistor *npn* si ha $h_{FE} = 100$; $V_{EB(on)} = 0,7$ V; $V_{EC(sat)} = 0,1$ V. Inoltre $V_{CC} = 8$ V; $R_B = 10$ k Ω ; $R_C = 1$ k Ω .



ESERCIZIO N°2

6 punti

Realizzare con una porta AOI CMOS, disegnandone lo schema elettrico, la seguente rete logica descritta con generiche porte logiche.



ESERCIZIO N°3

6 punti

Sintetizzare una rete sequenziale sincronizzata con un ingresso e una uscita, secondo il modello di Moore, in grado di riconoscere la sequenza (non interallacciata) 00100. Ogni volta che la sequenza viene riconosciuta, e solo allora, la rete pone in uscita (per un solo ciclo di clock) il valore vero.

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.

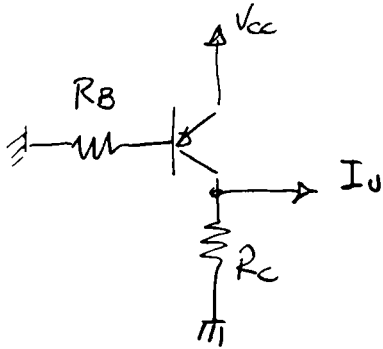
```
A:  IF J THEN D ELSE B; OP = 101111
B:  IF L THEN A ELSE C; OP = 011100
C:  IF K THEN B ELSE F; OP = 101111
D:  IF M THEN E ELSE H; OP = 010000
E:  IF M THEN G ELSE D; OP = 111101
F:  IF J THEN C ELSE A; OP = 011100
G:  IF L THEN H ELSE E; OP = 100011
H:  IF K THEN F ELSE G; OP = 010000
```

ESERCIZIO N°5

8 punti

Scrivere un sottoprogramma per il microcontrollore AVR90S8515 che cancelli un numero di locazioni consecutive pari al contenuto di R0 nella memoria dati estesa a partire dall'indirizzo contenuto nel puntatore Y.

① Nelle condizioni eseguite



Con $I_O = 0$ il BJT è in saturazione

$$V_B = V_{CC} - V_{EB\text{sat}} = 7,2\text{V}$$

$$I_B = \frac{V_B}{R_B} = 0,72\text{mA (uscite)}$$

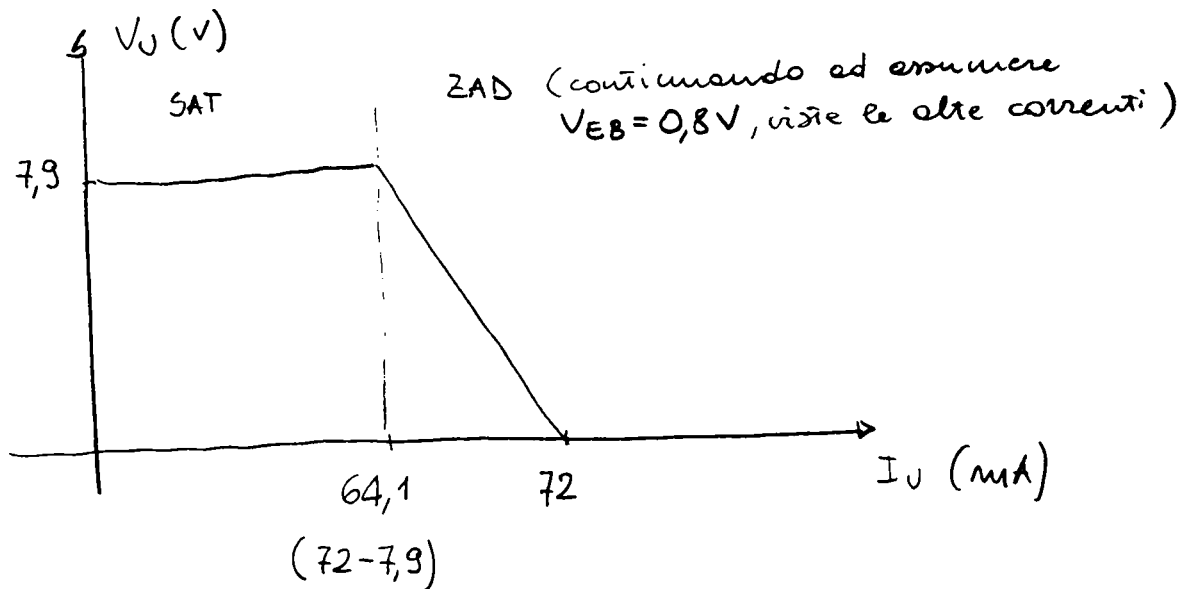
$$V_C = V_{CC} - V_{EC\text{sat}} = 7,9\text{V}$$

$$I_C = \frac{V_C}{R_C} = 7,9\text{mA} \ll h_{FE} I_B = 72\text{mA} \quad (\text{in sat})$$

Quindi la condizione di saturazione si mantiene fino a che la $I_C < 72\text{mA}$ con $I_C = \frac{V_{CC} - I_O}{R_C}$.

Poi, al crescere di I_O , si passa in zona attiva diretta

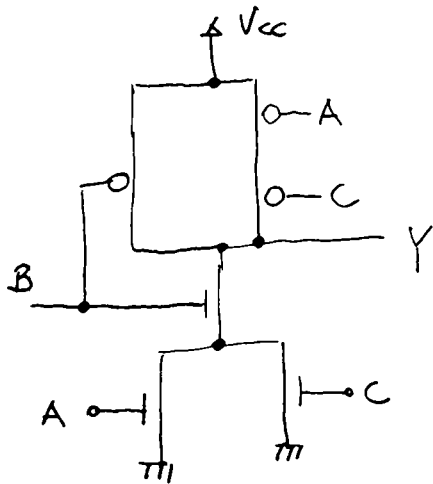
Quindi



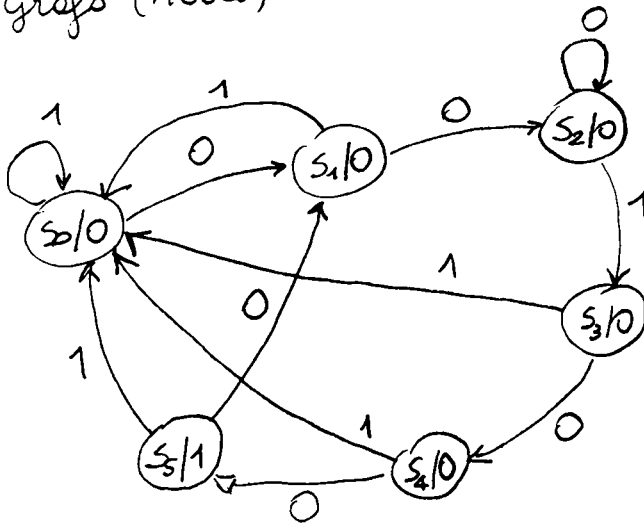
②
$$Y = \overline{AB + \overline{B} + \overline{A+C}}$$

$$\overline{Y} = AB + B \cdot (A+C) = AB + BC$$

Realizzazione CMOS



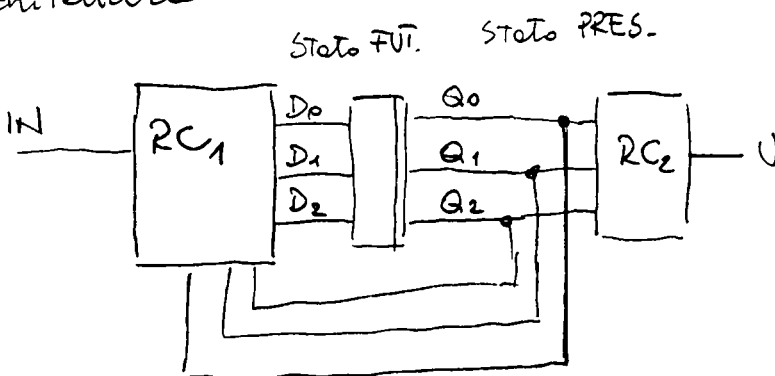
③ Grafo (Moore)



Codifica e rete per U

	$Q_2 Q_1 Q_0$	U
S_0	000	0
S_1	001	0
S_2	011	0
S_3	010	0
S_4	110	0
S_5	111	1

Architettura



Rete per U, RC_2

$$U = Q_2 Q_1 Q_0$$

Rete per lo stato futuro D_2, D_1, D_0

Q_1, Q_0		IN, Q_2			
		00	01	11	10
00	00	001	x	x	000
01	01	011	x	x	000
11	11	011	001	000	010
10	10	110	111	000	000

D_2

Q_1, Q_0		IN, Q_2			
		00	01	11	10
00	00	0	x	x	0
01	01	0	x	x	0
11	11	0	0	0	0
10	10	1	1	0	0

D_1

Q_1, Q_0		IN, Q_2			
		00	01	11	10
00	00	0	x	x	0
01	01	1	x	x	0
11	11	1	0	0	1
10	10	1	1	0	0

D_0

Q_1, Q_0		IN, Q_2			
		00	01	11	10
00	00	1	x	x	0
01	01	1	x	x	0
11	11	1	1	0	0
10	10	0	1	0	0

$$D_2 = \overline{IN} Q_1 \overline{Q_0} ; \quad D_1 = \overline{IN} Q_1 Q_0 + \overline{IN} \overline{Q_2} Q_0 + \overline{Q_2} Q_1 Q_0 ; \quad D_0 = \overline{IN} \overline{Q_1} + \overline{IN} Q_0 + \overline{IN} Q_2$$

⑤ Sotto programma

```

sub:   PUSH R0           ; salva i registri usati
       PUSH R16
       PUSH R28         ; puntatore Y
       PUSH R29
       CLR R16          ; azzerare un registro
loop:  ST Y+, R16       ; salva la cella puntata da Y
       DEC R0
       BRNE loop       ; ripete R0 volte
       POP R29         ; ripristina
       POP R28
       POP R16
       POP R0
       RET
    
```

④ Sequenza ciclica A, D, E, G, H, F, C, B, A... (flag vero)

Codifica stati e riordino codice

Stato	Codice	Flag	
A	000	J	00
D	001	K	01
E	010	L	10
G	011	M	11
H	100		
F	101		
C	110		
B	111		

Un'altra cosa
 "comprimere" anche il
 codice operativo, perché
 le bit non indipendenti.

Ne bastano 3

$$OP_5 = OP_0$$

$$OP_4 = \overline{OP_1}$$

$$OP_3 = OP_2$$

Schema logico

