

**ESERCIZIO N°1**

7 punti

Un invertitore CMOS ( $V_{Tn} = -V_{Tp} = 1\text{ V}$ ,  $K_n = -K_p = 1\text{ mA/V}^2$ ) alimentato a  $V_{CC} = 5\text{ V}$ , pilota invertitori RTL ( $R_C = 1\text{ k}\Omega$ ,  $R_B = 30\text{ k}\Omega$ ,  $h_{FE} = 100$ ), anch'essi alimentati a  $V_{CC}$ . Determinare il massimo numero di invertitori RTL che possono essere pilotati avendo la garanzia che il margine di rumore sul livello alto sia almeno pari a quello sul livello basso.

**ESERCIZIO N°2**

6 punti

Sintetizzare una rete combinatoria in forma minima (secondo il numero di letterali) somma di prodotti, a 4 ingressi e 1 uscita, in grado di riconoscere, ponendo l'uscita al valore vero, la presenza in ingresso dello 0 e dei multipli di 2, 3 e 5, espressi su 4 bit in forma binaria.

**ESERCIZIO N°3**

6 punti

Progettare una rete sequenziale sincronizzata con 1 ingresso e 1 uscita, in grado di riconoscere la presenza di almeno di 2 valori a 1 tra quelli che si sono presentati all'ingresso in corrispondenza dei tre precedenti fronti di clock.

IN: 00100010101001011100111110000

OUT: XX000000101000011110011111000 ( $t_{co}$  dopo il fronte di clock)

**ESERCIZIO N°4**

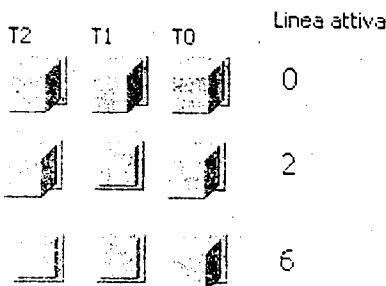
6 punti

Si hanno a disposizione tre memorie, una da  $32 \times 8$ , una da  $64 \times 8$  e una da  $512 \times 8$ . Assemblare le tre memorie nello stesso modo con cui si presentano all'interno del microcontrollore AT90S8515 i registri interni, i registri di I/O e la memoria dati estesa. Fare in modo che per indirizzi (16 b) di valore superiore a 607 nessuna delle memorie sia attiva, né in scrittura, né in lettura.

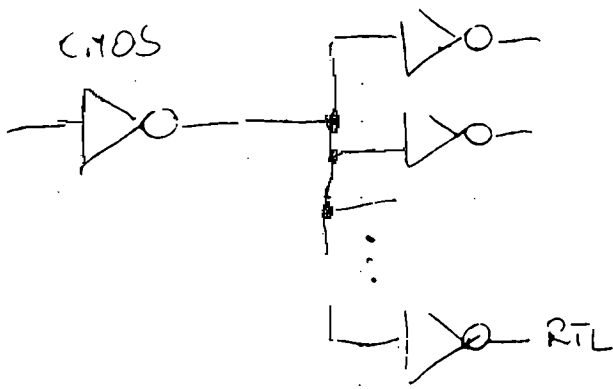
**ESERCIZIO N°5**

8 punti

Scrivere un programma per il microcontrollore AT90S8515 che, dopo avere inizializzato correttamente le porte, legga continuamente il valore di tre pulsanti T2, T1 e T0, collegati rispettivamente tra i pin PB2, PB1, PB0 e massa, e attivi la sola linea della porta A corrispondente al valore binario indicato dalla pressione dei pulsanti.



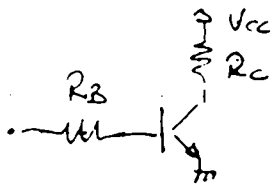
①



Sul livello di uscite basso della porta CMOS non ci sono problemi, perché gli studi e valle sono interdetti.

$$V_{OL} = 0V \quad V_{IL} = 0.7V \quad N_{IL} = 0.7V$$

Sul livello alto, per la RTL si ha

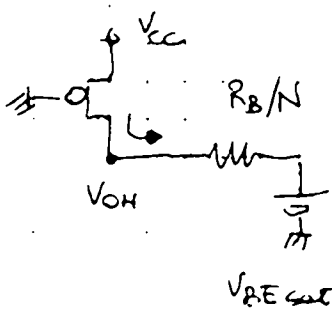


$$V_{IH} = V_{CESAT} + R_B \frac{V_{CC} - V_{CESAT}}{R_C \beta_{FE}} = 2.27V$$

Per il calcolo del fan-out si deve quindi garantire (\$N\_{IH} = N\_{IL}\$)

$$V_{OH} = V_{IH} + N_{IH} = 2.97V$$

Questo valore deve essere presente in uscita di CMOS con \$N\$ porte che assorbono corrente. Modello per l'analisi



Con \$V\_{OH} = 2.97V\$ siamo in zone TRIOBJ

Si avrebbe (al segno in figura)

$$|I_{OH}| = -\frac{\beta_F}{2} (V_{OH} + V_{CC}) (V_{CC} + V_{OH} + 2V_{BEsat}) = 6.06 mA$$

Con questa corrente si pilotano le porte a valle, purché

$$V_{BEsat} + \frac{R_B}{N} |I_{OH}| > V_{OH} \quad \text{da cui}$$

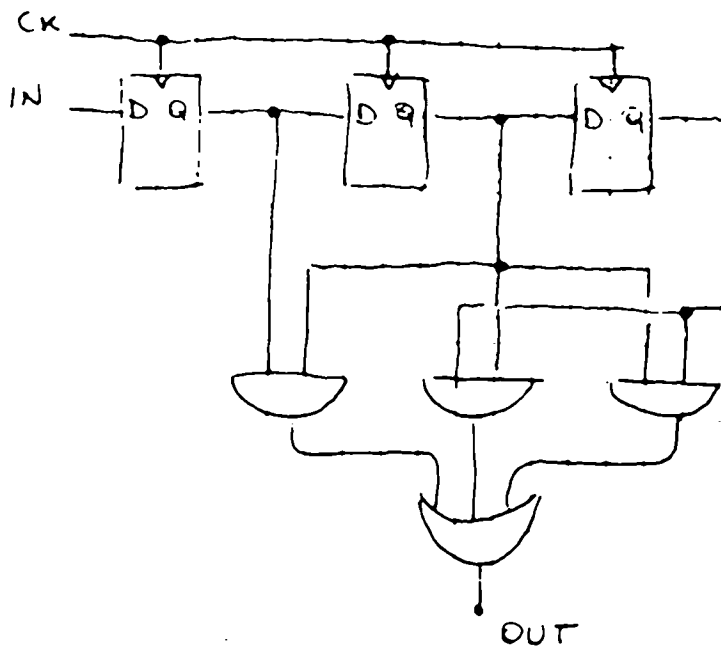
$$N < \frac{R_B |I_{OH}|}{V_{OH} - V_{BEsat}} = 83.77 \rightarrow N = 83$$

- ② La funzione si ricostruisce rapidamente.  
 Vale 1 in corrispondenza di tutte le caselle, escluso 1, 7, 11, 13

$x_1, x_0$		$x_3, x_2$			
		00	01	11	10
00	1	1	1	1	
01	0	1	0	1	
11	1	0	1	0	
10	1	1	1	1	

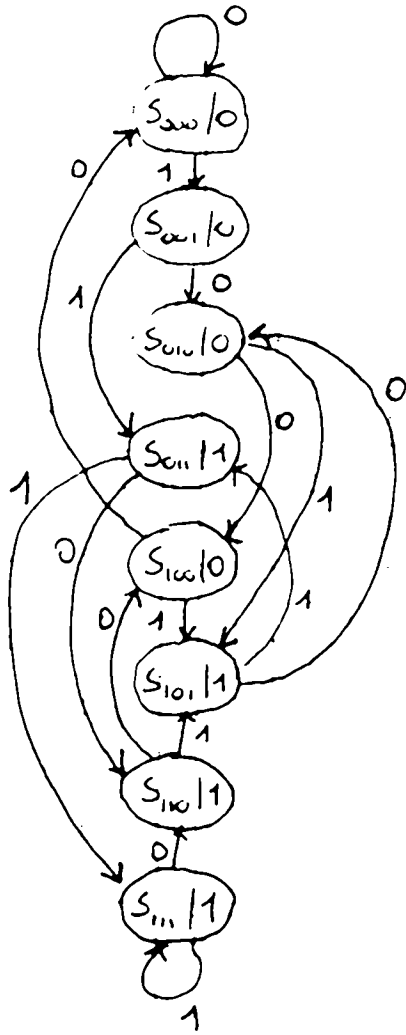
$$y = \bar{x}_0 + \bar{x}_3 x_2 \bar{x}_1 + x_3 \bar{x}_2 \bar{x}_1 + \bar{x}_3 \bar{x}_2 x_1 + x_3 x_2 x_1$$

- ③ Soluzione "ad hoc"



Ogni porta AND riconosce la presenza di una coppia di 1 in una certa posizione

Gruppo di flusso, per la sintesi strutturata (two)  
 Possiamo considerare uno stato per ogni tema di valori arrivati



IN	0	1	OUT
S0	S0	S1	0
S1	S2	S3	0
S2	S4	S5	0
S3	S6	S7	1
S4	S5	S0	0
S5	S2	S3	1
S6	S4	S5	1
S7	S6	S7	1

Codifica degli stati       $OUT = q_2$

	$q_2$	$q_1$	$q_0$
S0	0	0	0
S1	0	0	1
S2	0	1	1
S3	1	1	1
S4	0	1	0
S5	1	0	0
S6	1	1	0
S7	1	0	1

$d_2$

IN, $q_2$	00	01	11	10
00, $q_0$	0	0	1	0
01, $q_0$	0	1	1	1
11, $q_0$	0	1	1	1
10, $q_0$	1	0	1	0

$d_1$

IN, $q_2$	00	01	11	10
00, $q_0$	0	1	1	0
01, $q_0$	1	1	0	1
11, $q_0$	1	1	0	0
10, $q_0$	0	1	0	0

$d_0$

IN, $q_2$	00	01	11	10
00, $q_0$	0	1	1	1
01, $q_0$	1	0	1	1
11, $q_0$	0	0	1	0
10, $q_0$	0	0	0	0

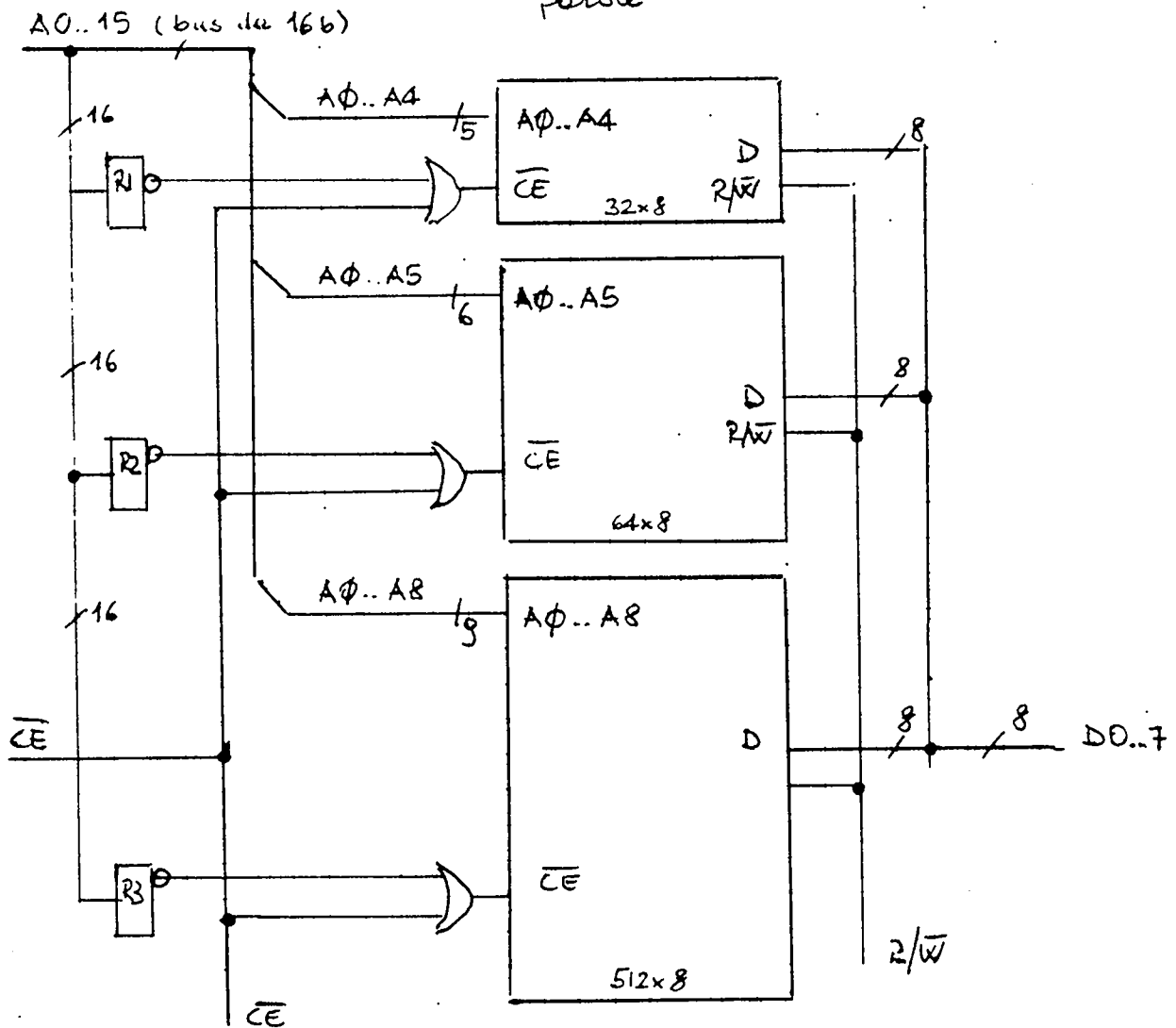
$$d_2 = IN q_2 + q_2 q_0 + IN q_0 + \overline{IN} \overline{q_2} \overline{q_1} \overline{q_0}$$

$$d_1 = \overline{IN} q_2 + \overline{IN} q_0 + q_2 \overline{q_1} \overline{q_0} + \overline{q_2} \overline{q_1} q_0$$

$$d_0 = IN \overline{q_1} + q_2 \overline{q_1} \overline{q_0} + \overline{q_2} \overline{q_1} q_0 + IN q_2 q_0$$

①

Estensione del NUMERO di parole



R1 riconosce gli indirizzi tra 0 e 31  
 R2 tra 32 e 95  
 R3 tra 96 e 607

APPROFONDIMENTO

Per progettare le R basta vedere come sono fatti gli indirizzi che attivano la relativa uscita.

A <sub>15</sub> A <sub>14</sub> A <sub>13</sub> A <sub>12</sub>	n°	funzione logica
000000000000xxxx	32	$R_1 = \overline{A_{15}} \cdot \overline{A_{14}} \dots \overline{A_5}$
000000000001xxxx	32	$R_2 = \overline{A_6} \cdot \overline{A_{14}} \dots \overline{A_5} + \overline{A_{15}} \cdot \overline{A_{14}} \dots \overline{A_6} \cdot \overline{A_5}$
000000000010xxxx	32	
000000000011xxxx	32	$R_3 = \overline{A_{15}} \cdot \overline{A_{14}} \dots \overline{A_7} + \overline{A_{15}} \dots \overline{A_7} \cdot A_6 \cdot A_5$
0000000001xxxxxx	128	
000000001xxxxxxx	256	
00000001000xxxxx	64	
000000010010xxxx	32	

⑤ Occorre praticamente realizzare la funzione di un decoder 3:8

```
LDI R16, 0b00000000
OUT DDRB, R16 ; porta B in ingresso
LDI R16, 0b11111111
OUT PORTB, R16 ; attiva pull-up
LDI R16, 0b11111111
OUT DDRA, R16 ; porta A in uscita
```

```
loop: IN R16, PINB ; legge i tasti
      COM R16 ; testo presunto uguale 1
      ANDI R16, 0b00000111 ; solo i tre tasti da considerare
      INC R16 ; se 1 è attiva la linea 0, ca.
      LDI R17, 0b00000001 ; linea 0 attiva
      DEC R16 ; il dato è pronto per l'out
      BREQ fine ; attiva la linea successiva
      LSL R17
      RJMP R1
```

```
fine: OUT PORTA, R17
      RJMP loop
```