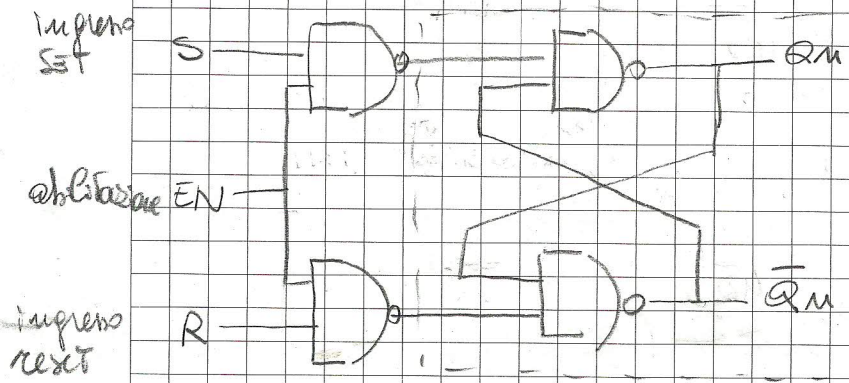


FLIP FLOP (FF) LEVEL TRIGGERED (in un caso sul livello)



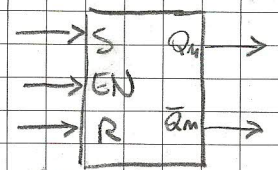
$Q^+ = Q_M$   
 $\bar{Q}^+ = \bar{Q}_M$

LATCH di forte AND con abilitazione (che ha uscite T.F.Ck) e del LATCH di forte OR

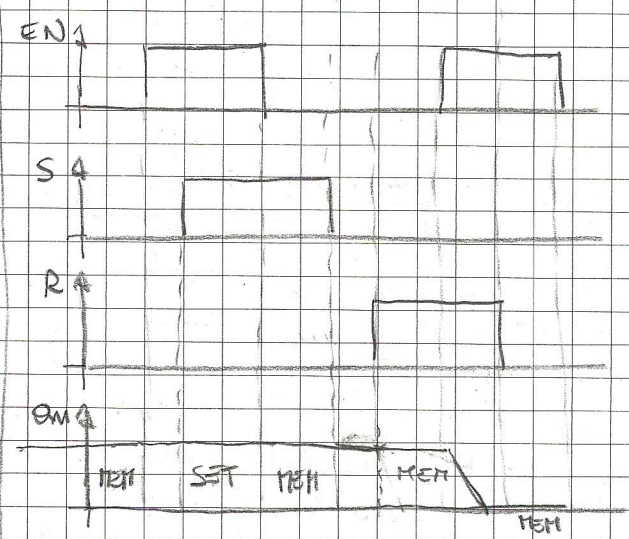
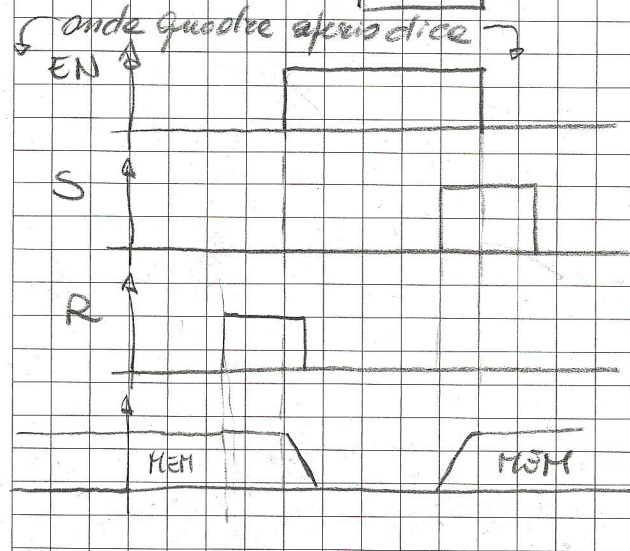
S	R	EN	Q_M
X	X	0	Q_M-1
0	0	1	Q_M-1
1	0	1	1
0	1	1	0
1	1	1	NSA

LATCH forte AND

(soluzione non predicibile)



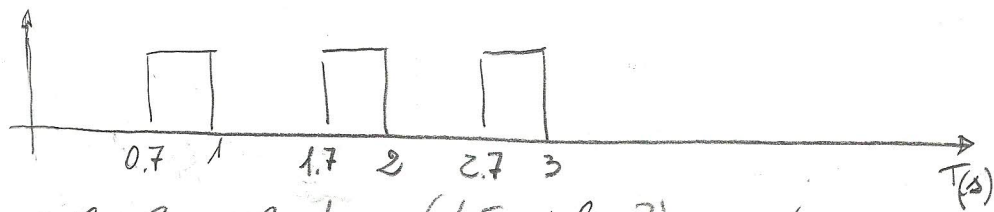
Con EN = 0 resta come risultato quello in memoria



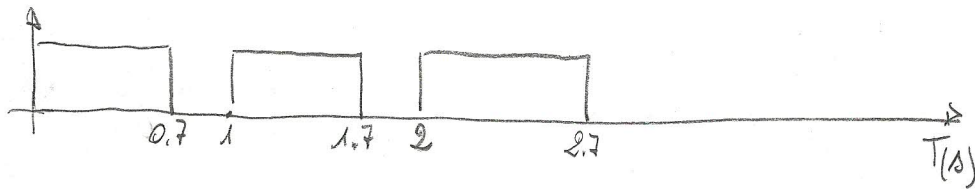
Quando EN è un'onda quadra periodica con  $T_a = T_b$  allora EN diventa tempo di clock

il Latch è replotato da un segnale singolo

il flip flop è replotato da un segnale di clock (quadra periodica con tempo alto = tempo basso)

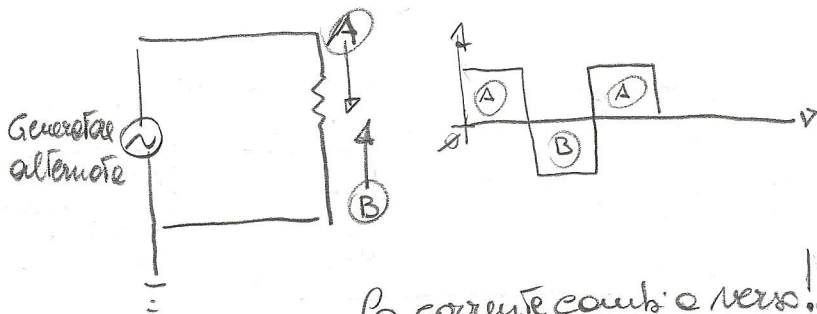
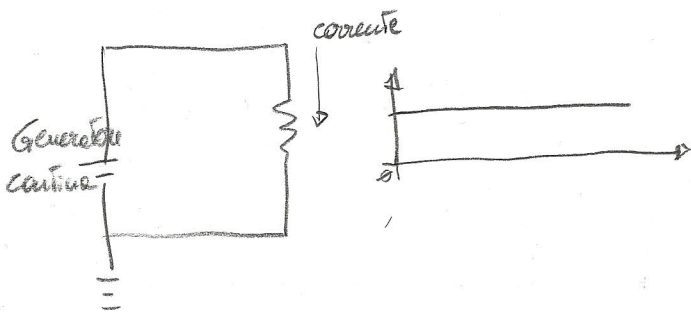
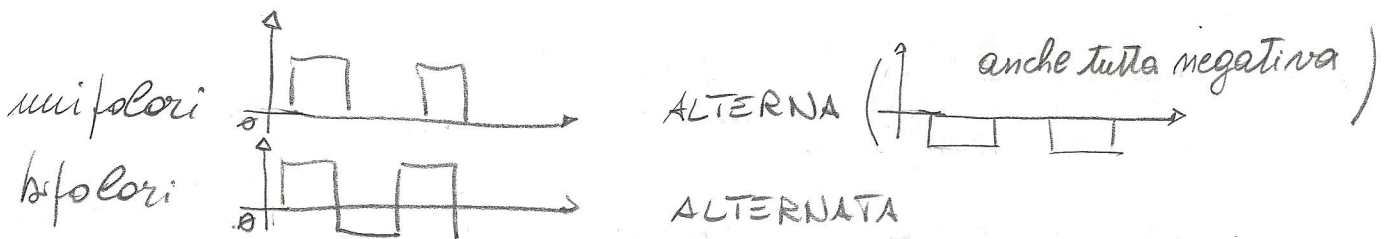


calcolare il dc (duty cycle ?) = 30%



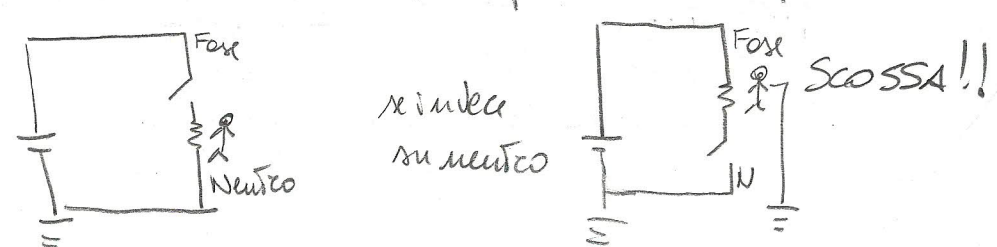
dc = 40%

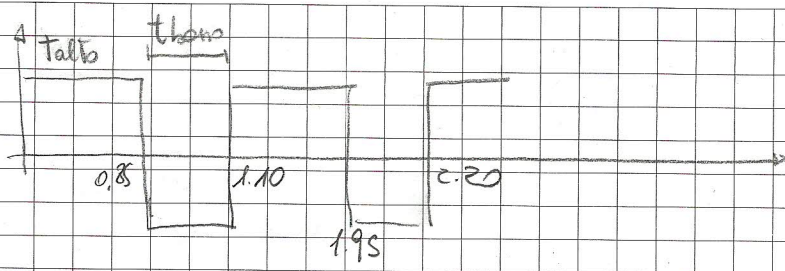
onde quadre bifase alternate (Tempo positivo = Tempo negativo)



La corrente cambia verso!  
La resistenza è quindi bidirezionale (passiva)

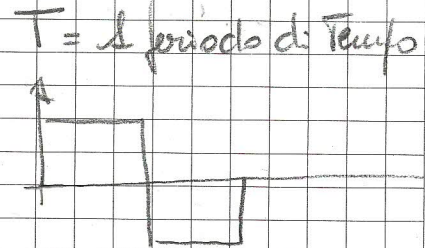
fase = filo marrone nero o grigio  
 neutro = filo azzurro che porta al negativo del generatore  
 Terra = quello/verde  
 gli interruttori vanno messi sulle fase per evitare di fendere la scorta





onde quadre  
alterate bipolare

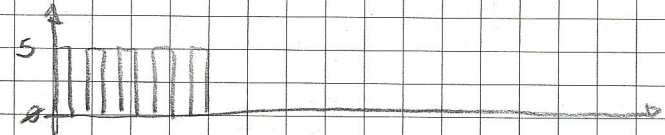
$$D.C. = \frac{t_a}{T} = \frac{0,85}{1,9} = 0,44\%$$



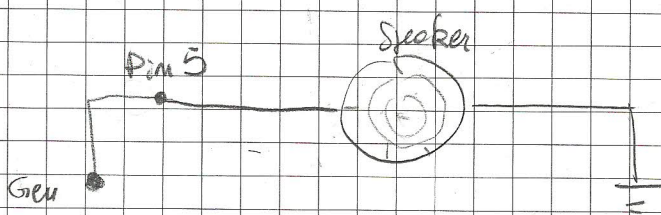
Suono



suono basse frequenze (basso)



suono alte frequenze (sopra)



Pin 5 (5, 001 pin)

Temp. ecc --

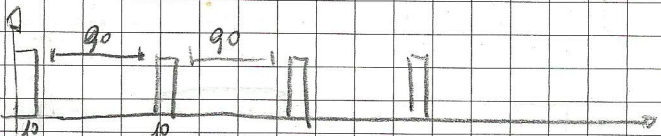
Pin 5 (HIGH)  
delay 1000 (basso)

Pin 5 (LOW)  
delay 1000 (alto 1/s)

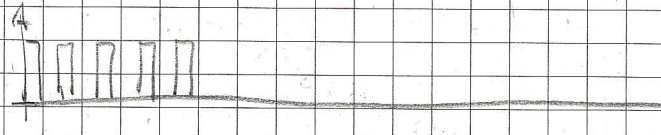
per cui una frequenza delay 1000 + delay 1000  
fa  $2^{ms}$  quindi un suono a 500 Hz

20 - 20000 Hz orecchio umano

se il delay lo fatto a  $10^{ms}$  avrò un suono in alte frequenze  
e quindi udibile all'orecchio umano a 10.000 Hz



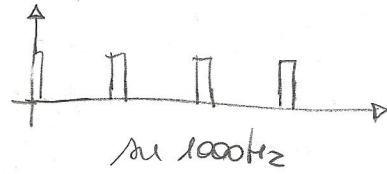
suono basse frequenze



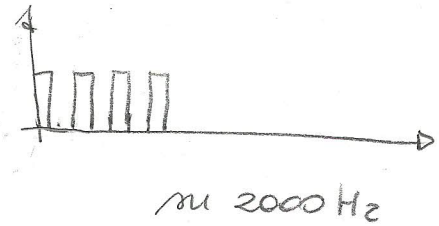
suono alte frequenze

Delay se man specific microseconds per default milliseconds per

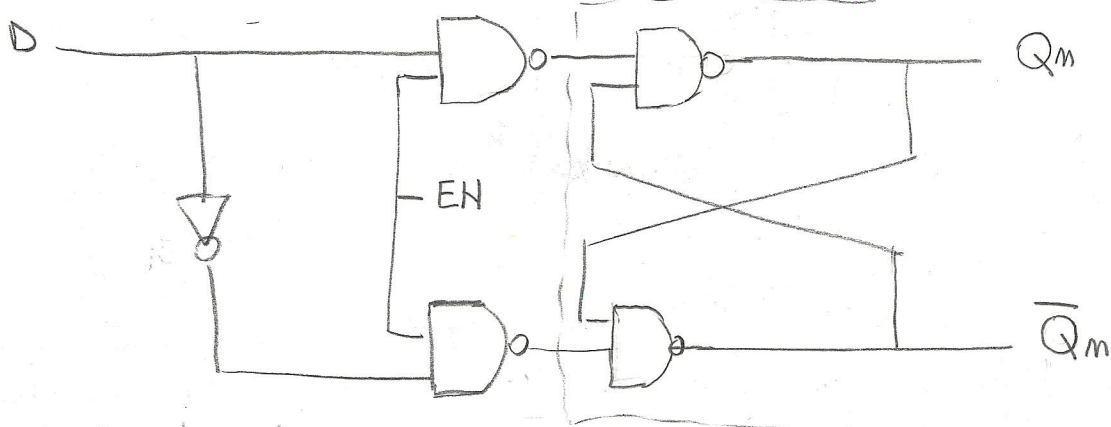
```
PinMode (5, output);
DigitalWrite (5, high);
Delay (100);
DigitalWrite (5, low);
Delay (900);
```



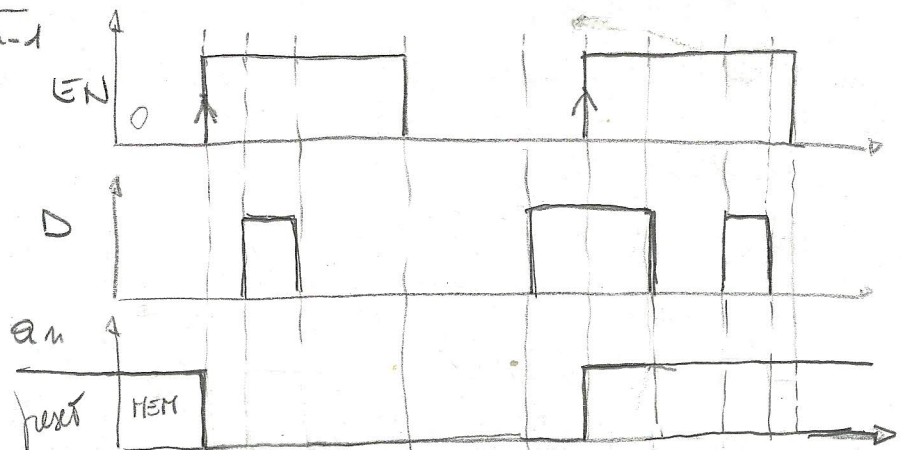
```
PinMode (5, output);
DigitalWrite (5, high);
Delay microseconds (250);
DigitalWrite (5, Low);
Delay microseconds (250);
```



$$T = \frac{1}{2000 \text{ Hz}} = 0,5 \cdot 10^{-3} \cdot 5 \cdot 10^4 = 0,5 \mu\text{s} \quad \text{periodo uguale}$$



EN	D	Qm
0	X	Qm-1
↑	0	0
↑	1	1



reset (parte da 1)  
clear (parte da 0)

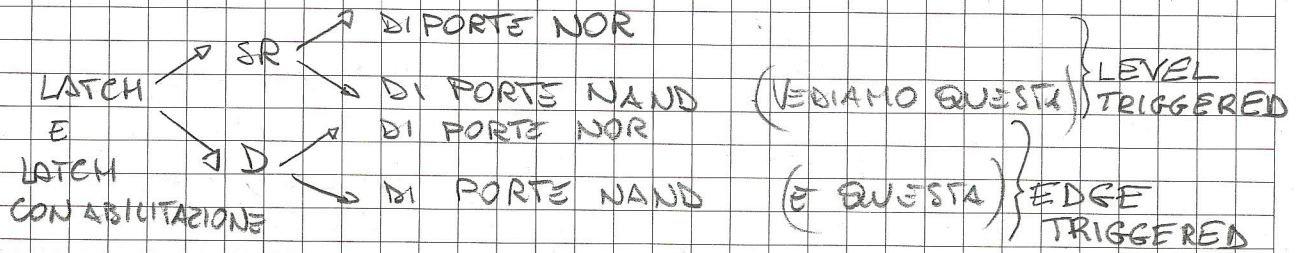
eserc: come latch D e SR con le temporizzazioni

DC. esercitazione n° 8

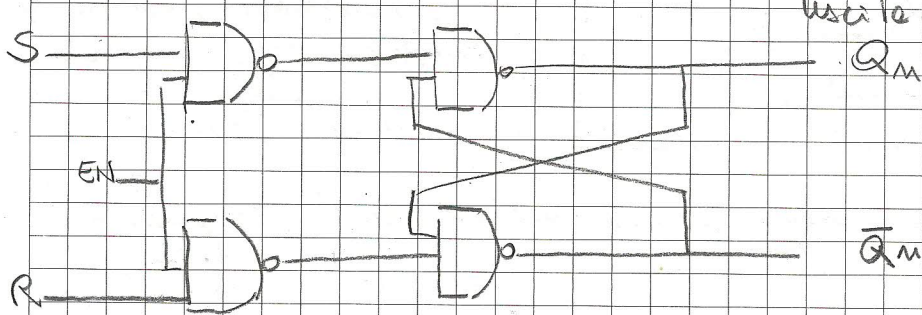
HARDWARE

ELEMENTI DI MEMORIA ELETTRONICA: LATCH →

→ LATCH CON ABILITAZIONE → FLIP FLOP



LATCH SR DI PORTE NAND CON ABILITAZIONE (EN=ENABLE)



dipende dall'uscita al tempo T dipende da EN e da T-1

EN	R	S	Q <sub>m</sub>
0	X	X	Q <sub>m-1</sub> → MEM (situazione in memoria)
1	0	1	1
1	1	0	0
1	1	1	NA (ambiguità)
1	0	0	Q <sub>m-1</sub>

X = qualsiasi valore

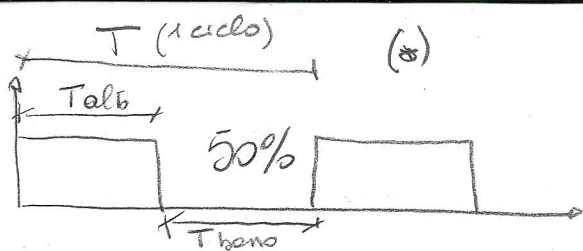
DIAGRAMMA DI TEMPORIZZAZIONE



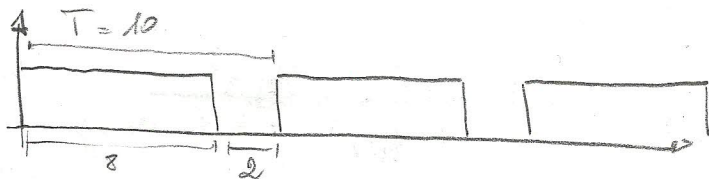
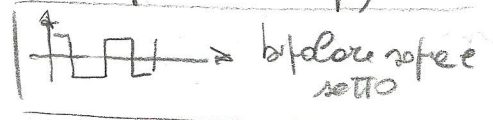
si vuole la forma di quale quadre di tipo (impulsiva) aperiodico

condizione iniziale Q<sub>m-1</sub> quando non c'è corrente cioè valore di uscita

- appena si RTI: EDS Electronic Data Systems - Auselda AED Group - Accenture - Enel APE accende il flip flop
- se è un LATCH è normale

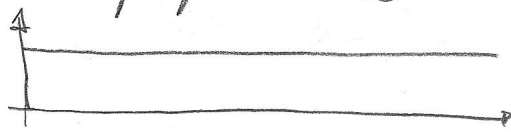


onde quadre mi colore (tutti a sopra o tutti a sotto la rife dei Temp)



dc = 80%

Nota dc = 100% è la continua e la frequenza = 0



- (\*) •  $T_a = T_b$
- DUTY CYCLE è la frazione percentuale in cui il segnale è alto per l'intero periodo

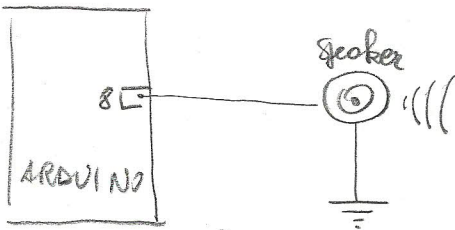
$$d.c. = \frac{T_a}{T_a + T_b}$$

nel nostro caso dato che  $T_a = T_b$

$$d.c. = \frac{T_a}{2T_a} = \frac{1}{2} = 0,5$$

quindi 50%

Se non c'è il periodo non si può calcolare la frequenza perché  $f = \frac{1}{T}$



Segnale  $f = 500 \text{ Hz}$  (La diopson)

Pin Mode (8, OUTPUT)

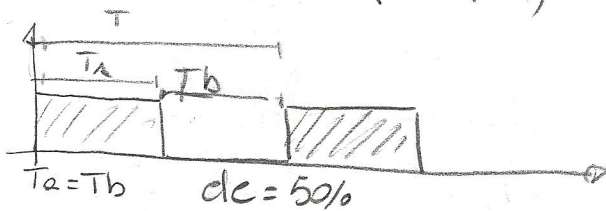
DigitalWrite (8, HIGH)

Delay (1)

DigitalWrite (8, LOW)

Delay (1)

il segnale 500 Hz dc 50% in uscita



$$T = \frac{1}{500} = 0,2 \cdot 10^{-2} = 2 \cdot 10^{-3} \text{ s} = 2 \text{ ms}$$

$T = 2 \text{ ms}$  (millesimo di secondo)



dc = 30%  $T_a = 3$   $T_b = 4$   $f = 1000 \text{ Hz}$

Periodo 1 ms (millesimo secondo)

$T_a = 0,3 \text{ ms}$   $T_b = 0,7 \text{ ms}$

$$T = \frac{1}{f} = \frac{1}{1000} = 10^{-3} \text{ s}$$

Pin Mode (8, OUTPUT)

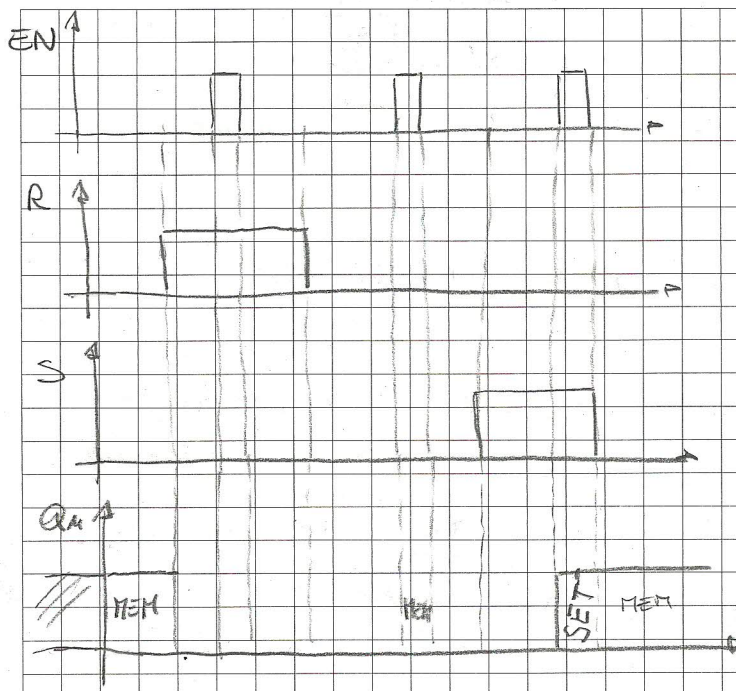
DigitalWrite (8, HIGH)

Delay microseconds (300)

DigitalWrite (8, LOW)

Delay microseconds (700)

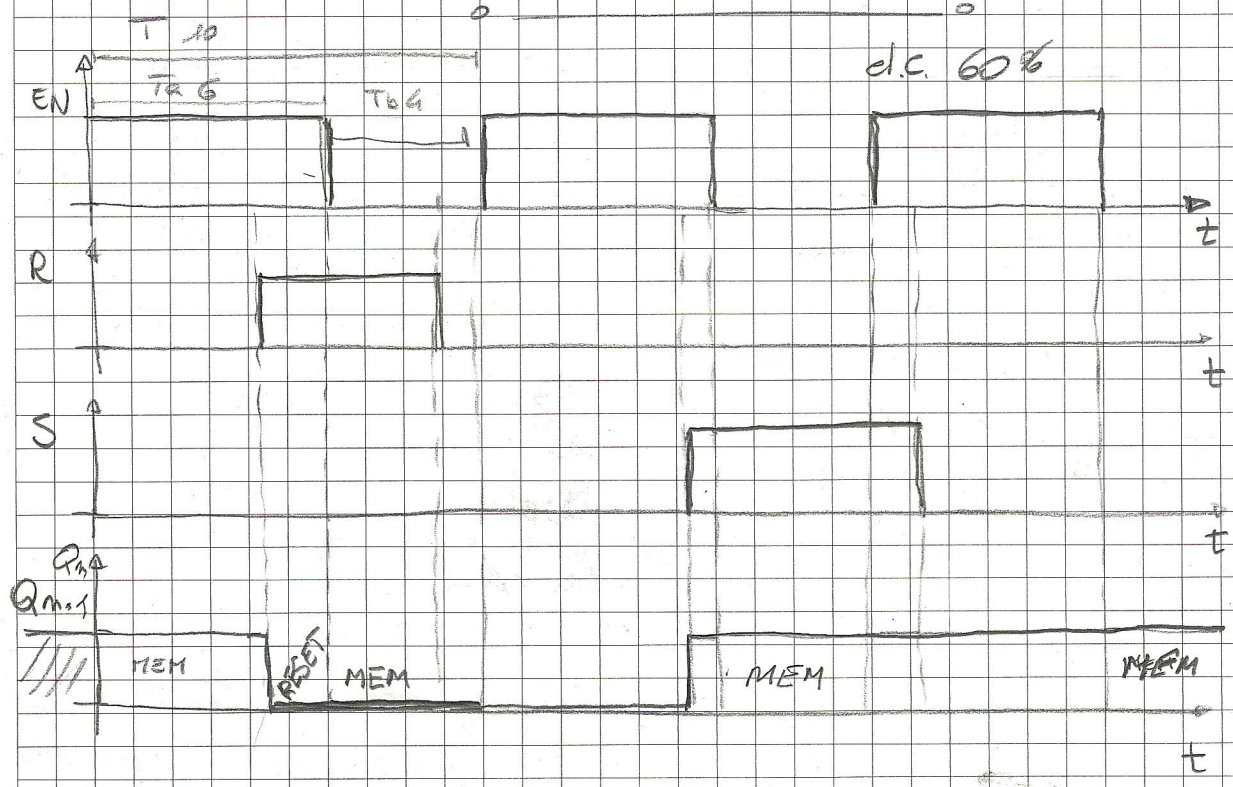
con esce un segnale di 1000 Hz con d.c. 30%



d.c. 10%

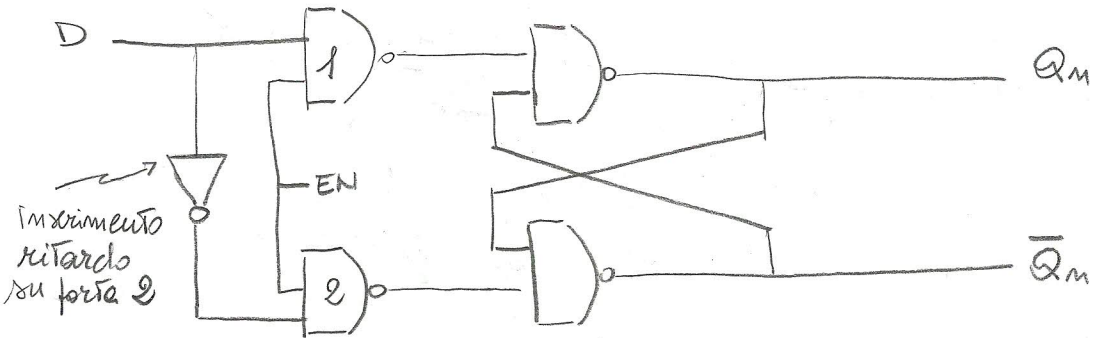
maximale de  
20Hz  $\approx$  19500Hz

EN	R	S	Q <sub>n</sub>
0	X	X	Q <sub>n-1</sub>
1	0	0	Q <sub>n-1</sub>
1	0	1	1
1	1	0	0
1	1	1	NA (amponale)



d.c. 60%

# LATCH D con enable di forte NAND

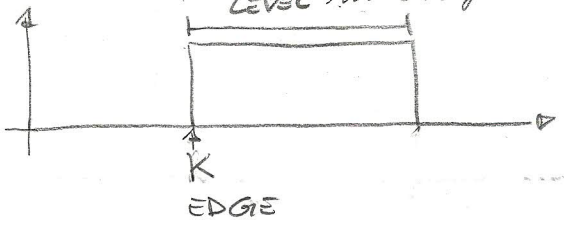


La differenza consiste nell'esistenza del ritardo

Il LATCH SR è di tipo LEVEL TRIGGERED  
 Il LATCH D è di tipo EDGE TRIGGERED

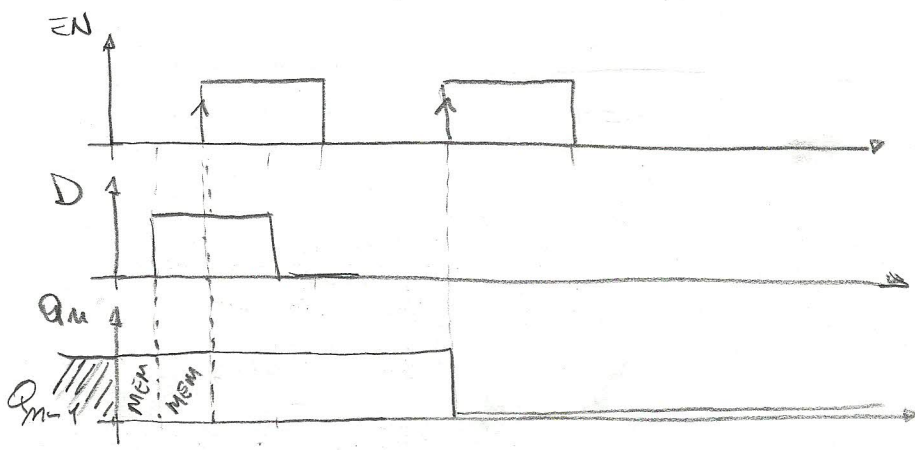
TRIGGERED = immerso

La differenza è che  
 LEVEL tutto il periodo può immercare



L'immerso può avvenire SOLO nel momento di salita ossia nel momento K

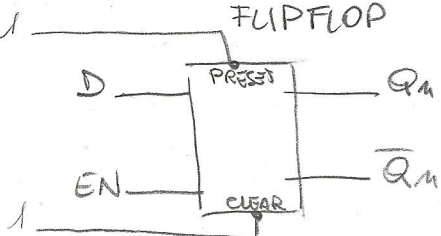
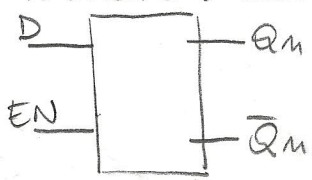
EN	D	Qm
0	X	Qm-1
1	0	0
1	1	1



Il flip flop è così più semplice dovendo considerare meno variabili

Che differenza c'è fra LATCH D e FLIP FLOP D è che nel secondo non si può lasciare casuale la condizione all'accensione

il LATCH D si schematizza

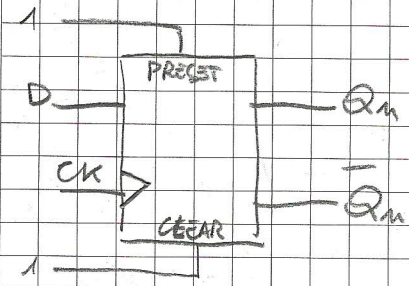


Il CLEAR a 1 impone pulizia delle memorie (forse a 0)

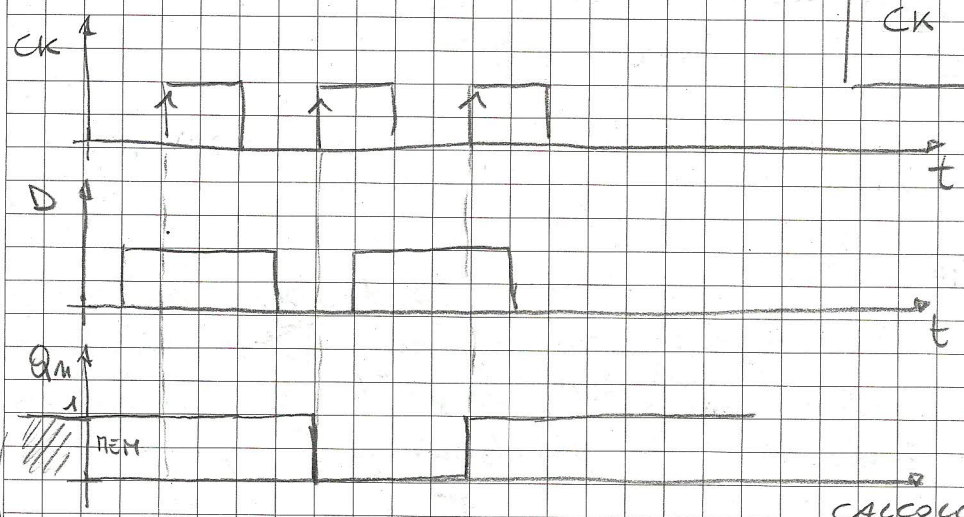
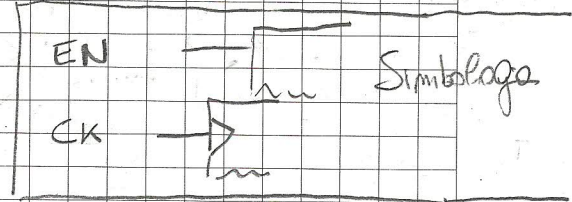
Il PRESET a 1 impone set a 1 delle memorie



nel flip flop si ha così un'onda quadrata con  $T_a = T_b$   
 onde EN di fronte a frequenza di: CLOCK (CK)



quindi d.c. 50% onde il segnale  
 di abilitazione diventa:

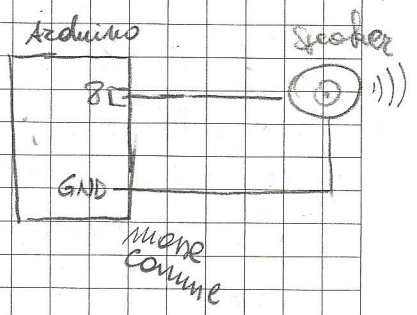


CK	D	Q <sub>n</sub>
0	x	Q <sub>n-1</sub>
↓	0	0
↑	1	1

PRESET = 1 d.c. 60% f 2000 Hz

CALCOLO PERIODO  
 $T = \frac{1}{f} = \frac{1}{2000} = 5 \cdot 10^{-4} \text{ s}$

- Pin Mode (8, OUTPUT)
- Digital Write (8, HIGH)
- Delay microseconds (300)
- Digital Write (8, LOW)
- Delay microseconds (300)



$T \cdot 60\% \cdot \text{Delay}$   
 $5 \cdot 10^{-4} \cdot 6 \cdot 10^{-2} = 3 \cdot 10^{-4} = 3 \cdot 10^{-4} \cdot 10^6 = 3 \cdot 10^2 = 300 \text{ ms}$  (microseconds)

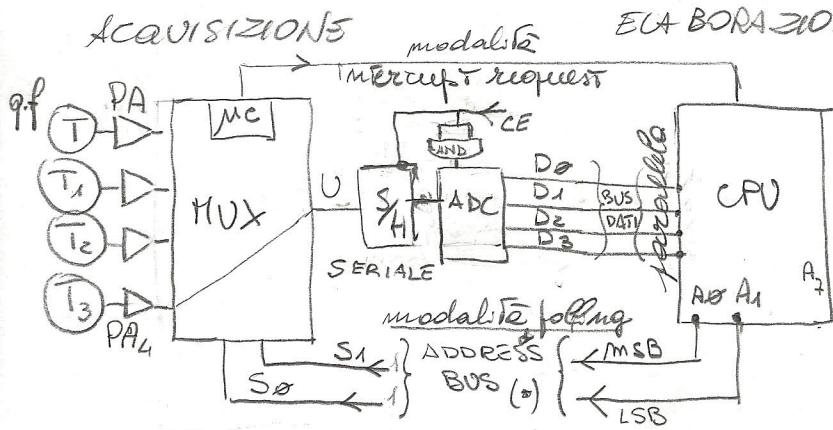
1 secondo  
 1000 millisecondi  
 1 000 000 microsecondi

# LAYOUT SISTEMA PROGRAMMABILE

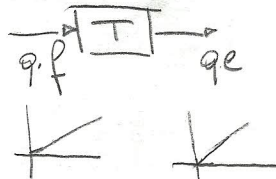
ACQUISIZIONE

ELABORAZIONE

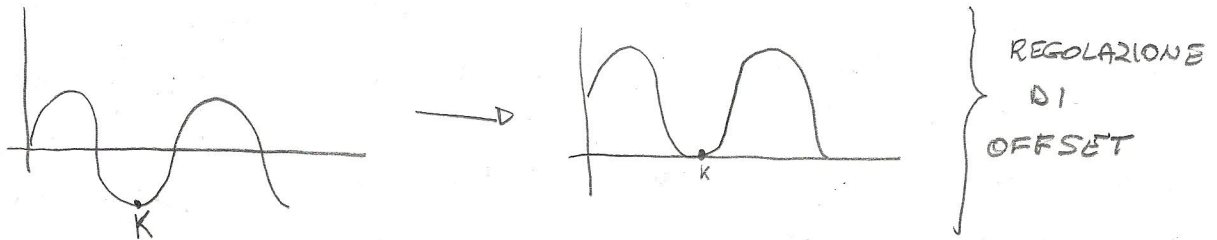
DISTRIBUZIONE



$T_x =$  Trasduttori



$PA_x =$  STADIO di PREAMPLIFICAZIONE o SISTEMI di condizionamento del segnale (rappresentato graficamente con  $\triangleright$ ) in questo caso si parla di regolazione dell' OFFSET (come si fa trascinare il segnale per farlo tutto nel quadrante positivo).



(\*) con address bus si seleziona il segnale del trasduttore corrispondente che verrà indirizzato in uscita (es  $S_0=1$   $S_1=1$  il  $T_3$  sarà inviato in uscita)

MC = microcontroller oie  
piccolo computer all'im  
oie

CPU

da  $A_2$  a  $A_8$  indirizzi periferiche / da  $A_9$  a  $A_{15}$  indirizzi di memoria

MSB = most significant bit } Gli indirizzi sono in ordine di significato  
LSB = less significant bit }  $A_2$  + significativo  $\rightarrow$   $A_8$  meno significativo  
esistono due modalità

1- Modalità di Polling  $\rightarrow$  la CPU interroga la periferica ad intervalli regolari (tramite proprie routine residenti in memoria)

2- Modalità Interrupt request  $\rightarrow$  si usano linee dedicate che collegano le CPU e che chiede un'interruzione alla CPU perché si dedichi alle richieste effettuate dalla periferica.

INTRA = indirizzo linee bus (bus) tra CPU e periferica (es. richieste nuovi fogli da stampare)

parametri delle CPU

- velocità (frequenze di clock) *espressa in*
- n° bit del Bus indirizzi
- n° bit del Bus dati

MC	PC
16 MHz	2 GHz
8 bit	64 bit
8 bit	64 bit

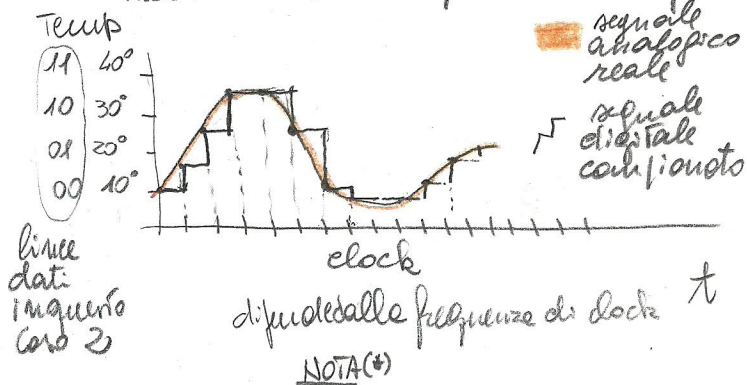
SAMPLE HOLD = componente segnale attese

ADC = analog to digital converter

Dx = dati

NOTA(\*) notare la discretizzazione sia sull'asse dei Tempi (con intervallo  $T_{clk}$ ) che nell'ampiezza (verticale) in 2 bit

Ricostruzione Temporale del segnale



L'aderenza delle ricostruzioni dipende quindi del rapporto tra digitale ed analogico perché si hanno le "rilevazioni" a istanti/momenti e non in continuo e anche in base al numero delle linee di dati forniti ad esempio con due si possono codificare solo 4 Temperature.

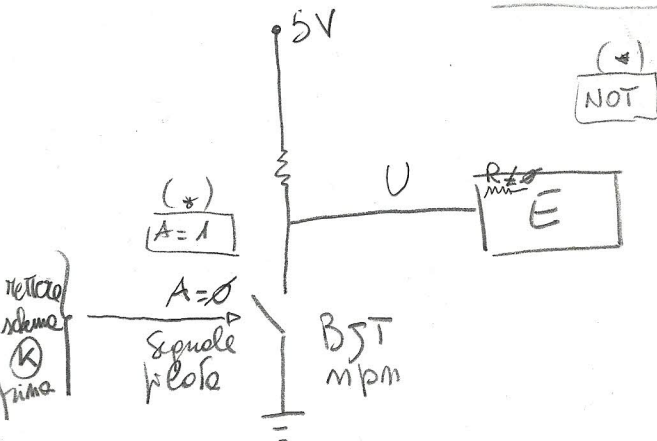
Esercizio

Schematizzo con "Interruttori comandati elettricamente" composti da transistor BJT n-p-n e resistenze in PULL-UP il seguente circuito combinatorio



PULL-DOWN la corrente fluisce se l'interruttore è chiuso  
 la corrente fluisce se l'interruttore è aperto

CIRCUITO CON RESISTENZE DI PULL-UP

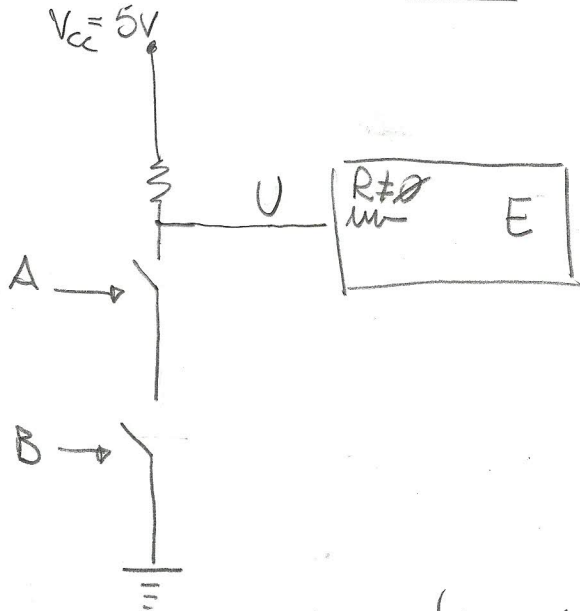


BJT p-np	BJT n-pn (*)
A=0 int chiuso	A=0 int aperto
A=1 int aperto	A=1 int chiuso

A	U
0	1
1	0

Porta and in Pullup

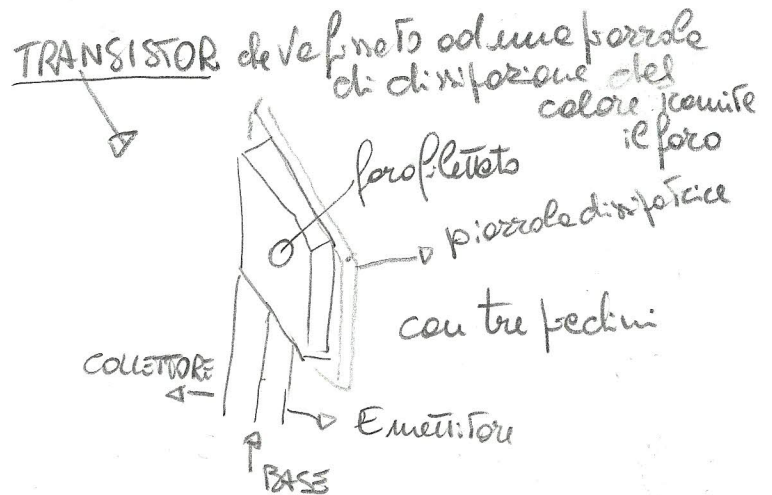
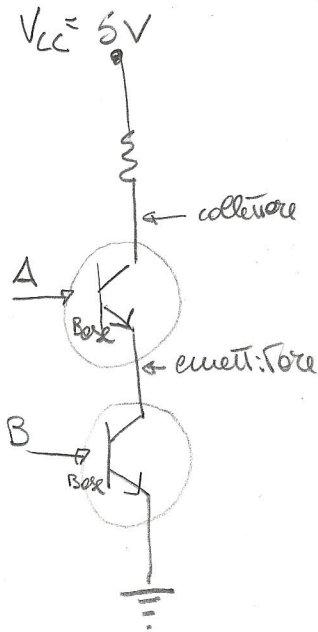
(schema (K) che precede quello di pag 6)



		NAND	AND
A	B	$\overline{A \cdot B}$	$A \cdot B$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

de cui si verifica che a noi basta avere il circuito dello schema (K) eliminando quello della pag 6

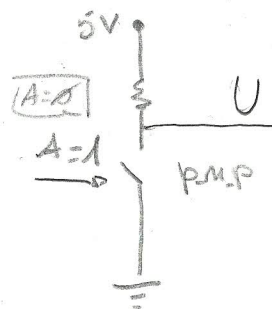
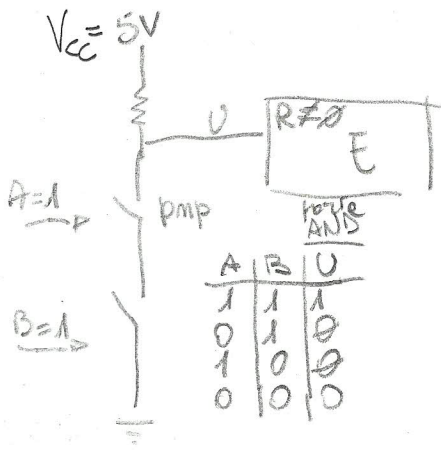
realizzazione  
BJT  
m-p-m  
K



Stesso esercizio ma con transistor di tipo

BJT p-m-p

Circuito box in Pull-up con BJT pmp (Tensione di alimentazione fissa 5V)



A	B	U
1	1	1
0	1	0
1	0	0
0	0	0

con Tabella Verità  
A=1 | int aperto  
A=0 | int chiuso

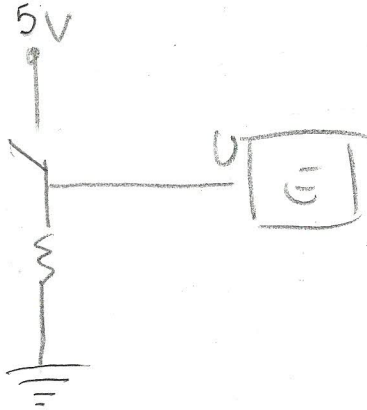
A	U
1	1
0	0

è quindi solo un circuito di ritardo

forchì dovremmo avere l'uscita delle porte AND negate

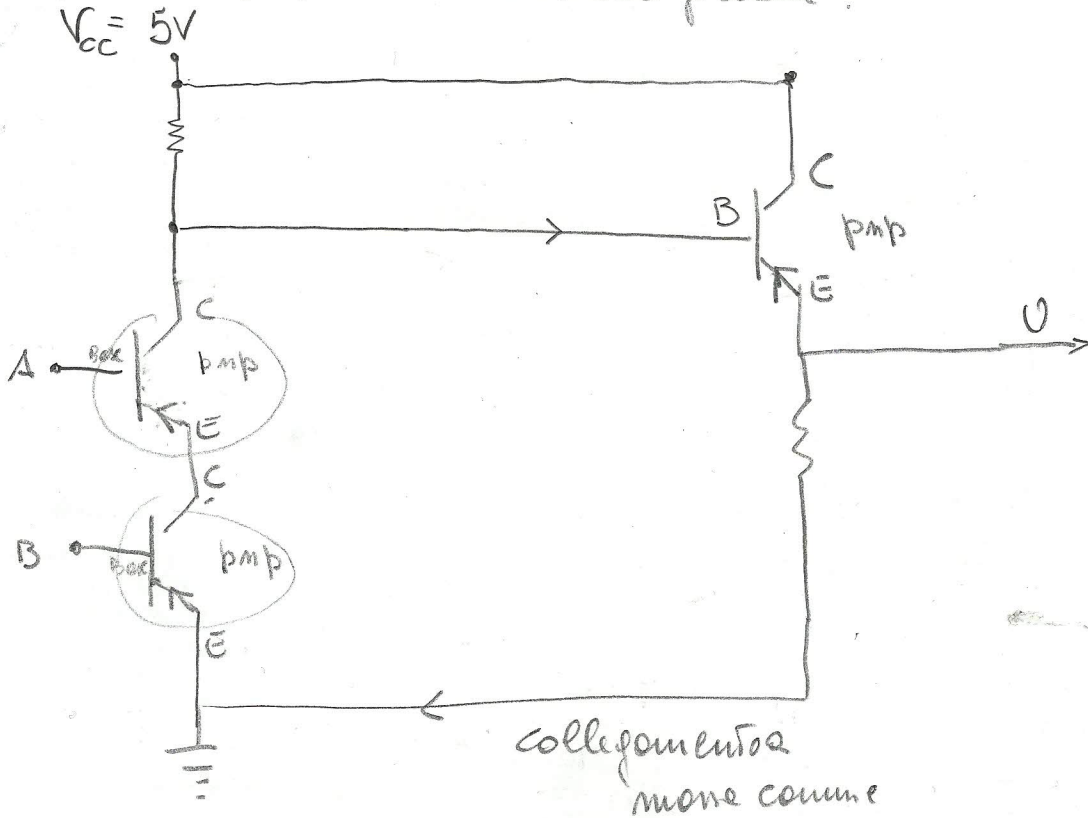


allora per negarla dobbiamo farla equiva. de:



A	U
1	0
0	1

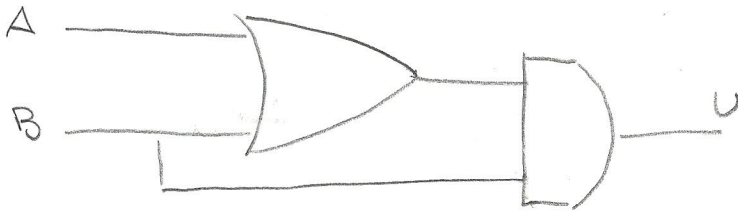
quindi avremo il circuito finale:



ESERCIZI CASA

da fare entrambi

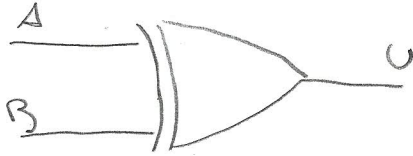
①



con:

BJT N-P-N in FULL-DOWN

②



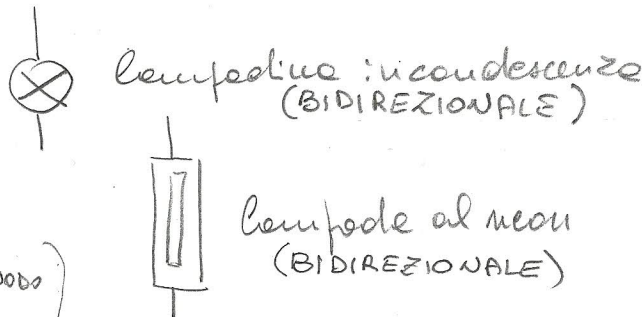
Esercizio

Si vogliono accendere 3 LED Rossi Alimentati in Tensione continua fissa 5V. Sapendo che la corrente nominale (di lavoro) del led di colore rosso è pari a 15mA schematizzare il circuito di accensione di 3 led e di menzionare l'eventuale R da inserire nel circuito

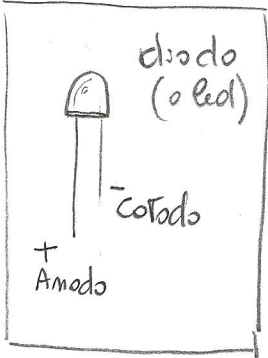
BLW 20-25mA

\* Consideriamo trascurabile la Resistenza del LED

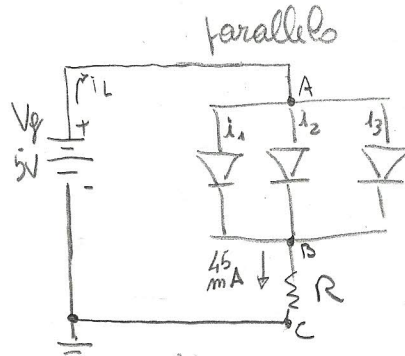
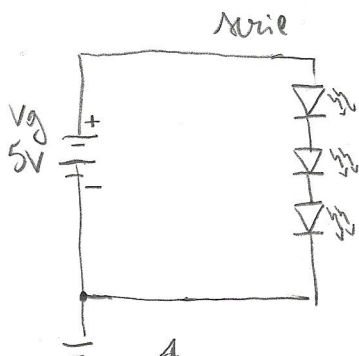
schematizzazioni:



non c'è indicazione + - perché bidirezionale.



(\*) mettiamo in serie un 470Ω perché così si vede poco ma durano moltissime ore di vita



$i_1 = i_2 = i_3 = 15 \text{ mA}$   
 $i_L = 45 \text{ mA}$

$V_{AB} = V_D = 1,2 \text{ V}$   
 $R_D \approx 0$  caratteristiche diodo

$V_{BC} = 3,8 \text{ V}$

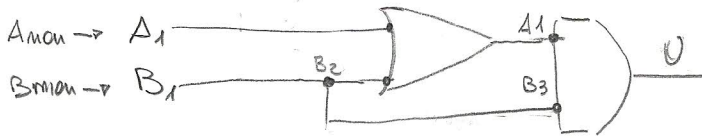
$R = \frac{V}{I_L} = \frac{3,8 \text{ V}}{45 \cdot 10^{-3} \text{ A}} = \frac{38 \cdot 10^{-1} \cdot 10^3}{45} = 0,95 \cdot 10^2 = 95 \Omega$

in successione non si accendono più quindi MEGLIO QUESTO

avendo R da 50Ω e da 100Ω c'è meglio 100Ω che così si accende meno ma non si brucia

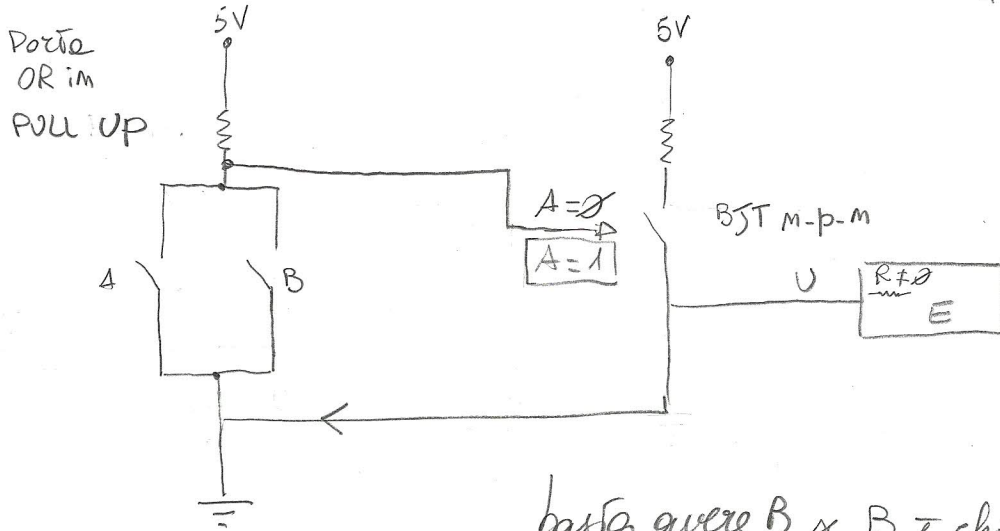
Esercizio case n° 1

BJT n-p-n in PULL DOWN



A	B	U
0	0	0
0	1	1
1	0	0
1	1	1

← semplificando

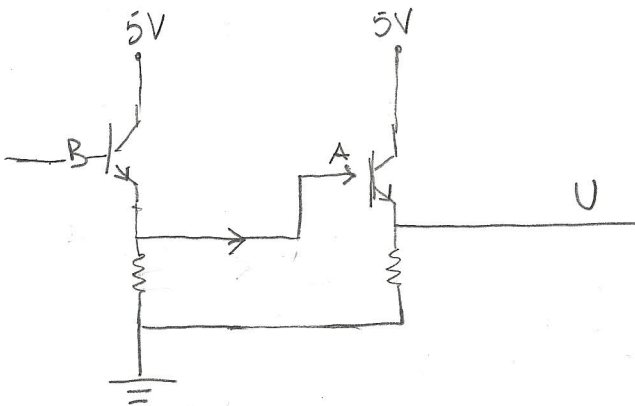


A	U
0	0
1	1

basta avere B x B  $\bar{x}$  chiuso  $U=1$  fase corrente  
 " " " aperto  $U=0$  non fase corrente

PULL DOWN

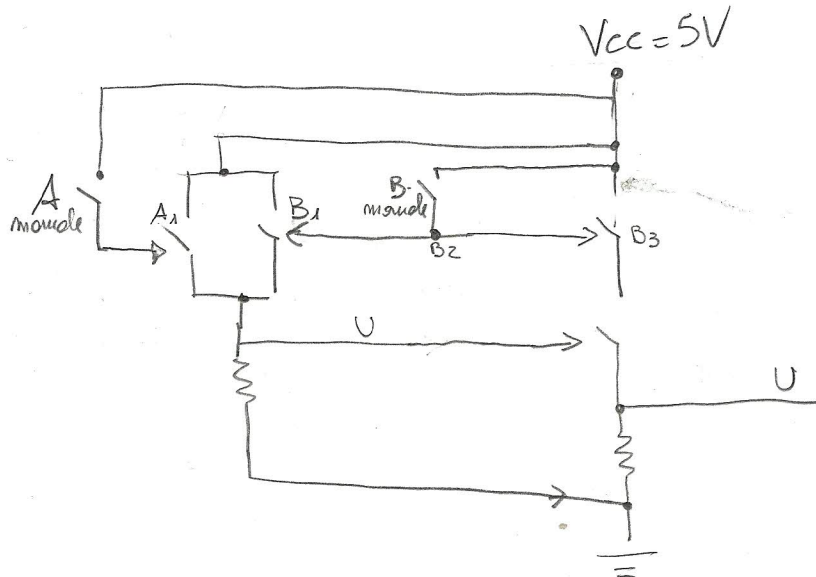
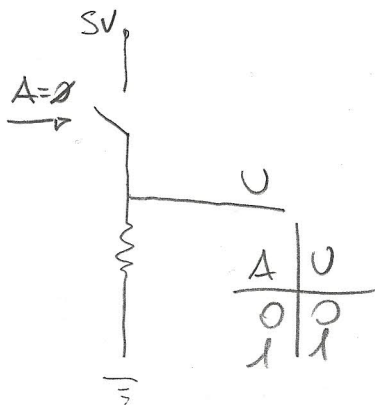
- 1 Interruttore
- 2 wate
- 3 Resistenza



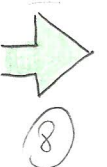
NON SEMPLIFICANDO

per fare una porta OR

collegamento elementare PULL DOWN

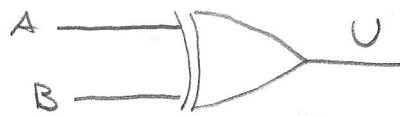


Togliendo le parti inutili  
 continua pag. successive

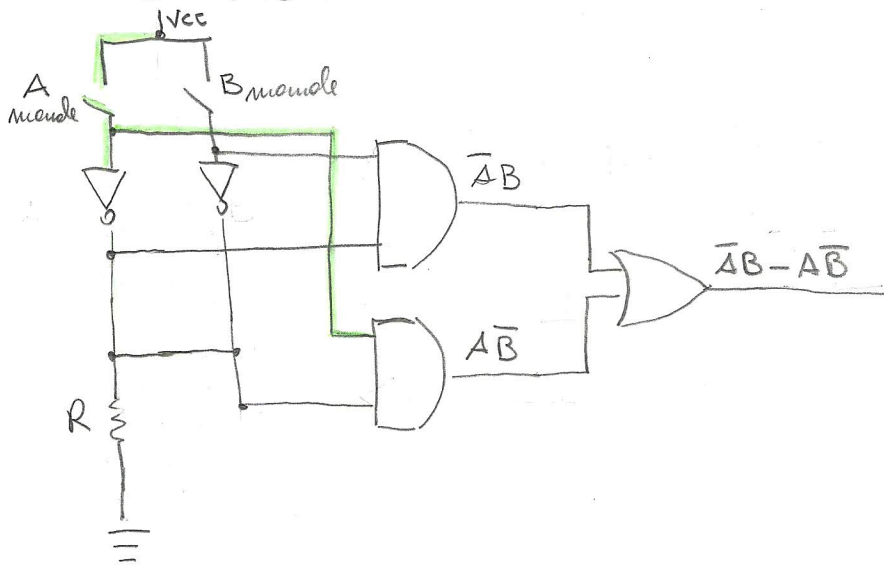


ESERCIZIO CASE M2

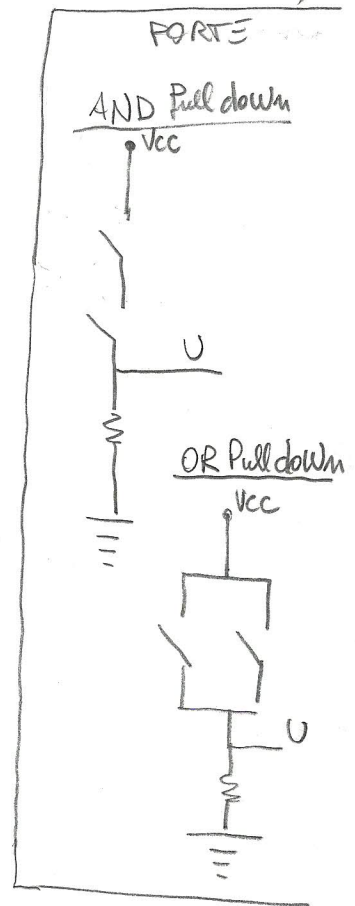
PORTA XOR



$A \oplus B = \overline{A}B + A\overline{B}$

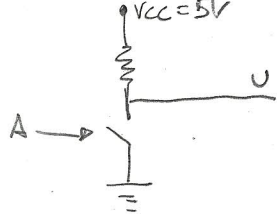


BJT n-p-n in PULL DOWN (NOTA BENE)

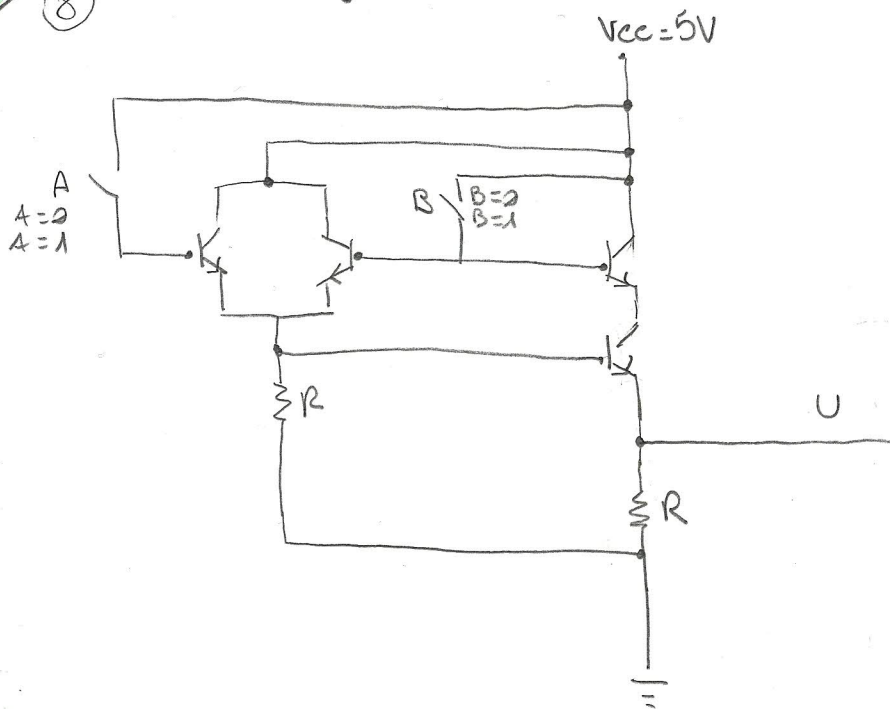


$V_{cc}$  = Tensione collettore collettore omie di Tipo BJT

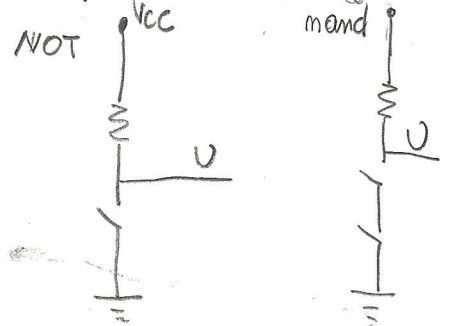
PORTA NOT semplice



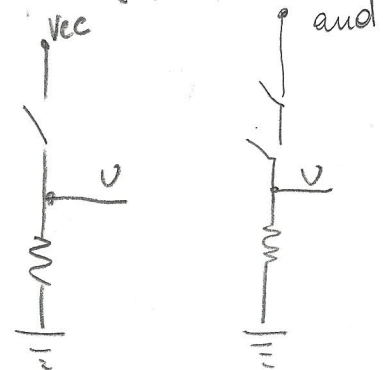
continua da pag 8



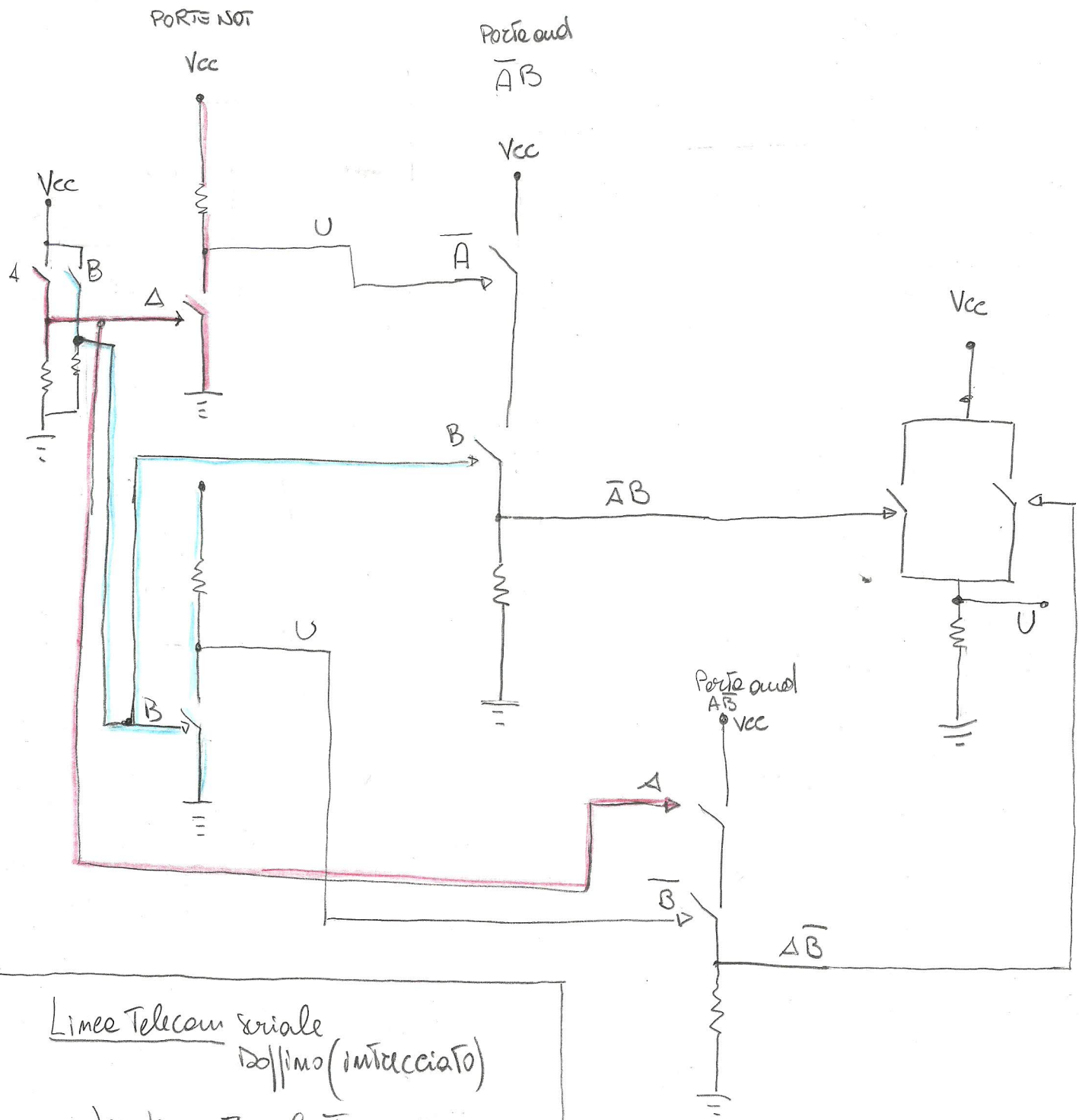
Pull up (o Pull down negato)



Pull down (o Pull up negato)







Linee Telecom seriali  
Doppino (intrecciato)

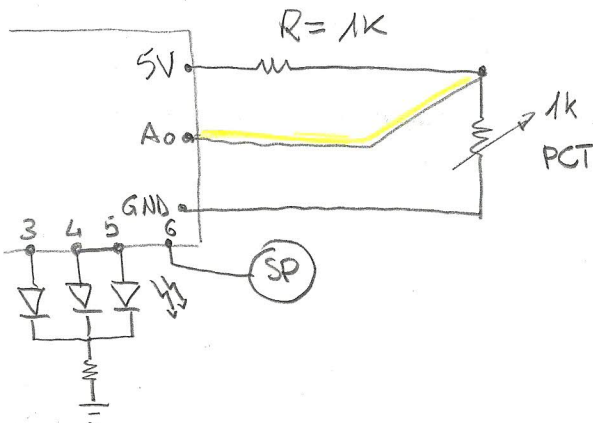
per permettere la trasmissione  
si devono seguire procedure  
codificate o sia dei  
protocolli

frequenze dai 100 ai 3400 Hz  
per comunicare

La differenza 440 Hz

# Esercizio Arduino

Temperatura	led verde	900
umidità	led giallo	650
	led rosso	400



Programma

```
void setup() {
```

```
  int rumore = AnalogRead(A0);
```

```
  Serial.begin(9600);
```

```
  pinMode(3, OUTPUT); // verde
```

```
  pinMode(4, OUTPUT); // giallo
```

```
  pinMode(5, OUTPUT); // rosso
```

```
  void loop() {
```

```
    Serial.println(A0);
```

```
    delay(100);
```

```
    if (rumore > 650)
```

```
      { digitalWrite(3, HIGH);
```

```
        digitalWrite(4, LOW);
```

```
        digitalWrite(5, LOW);
```

```
    } if (rumore > 400)
```

```
      { digitalWrite(3, LOW);
```

```
        digitalWrite(4, LOW);
```

```
        digitalWrite(5, HIGH);
```

```
    } if (rumore > 400 and rumore < 650)
```

```
      { digitalWrite(3, LOW);
```

```
        digitalWrite(4, HIGH);
```

```
        digitalWrite(5, LOW);
```

```
    } if (rumore < 400)
```

```
      { digitalWrite(3, LOW);
```

```
        digitalWrite(4, LOW);
```

```
        digitalWrite(5, LOW);
```

```
  }
}
```

Scrivere il programma:

per far suonare lo speaker  $f = 2\text{kHz}$  d.c. = 40%

per accendere 3 led e dimensionare le R per

$$I_M = 15\text{mA}$$

onda per Speaker



dc 40%

$$f = 2\text{kHz}$$

$$T = \frac{1}{f} = 0.5 \cdot 10^{-3}\text{s} =$$

$$= 500 \cdot 10^{-6}\text{ms}$$

microsecondi:

$$40\%T = \text{tempo alto} =$$

$$= 200\text{ms} = t_a$$

$$60\%T = \text{tempo basso} =$$

$$= 300\text{ms} = t_b$$

```
  digitalWrite(6, HIGH);
  delayMicroseconds(200);
  digitalWrite(6, LOW);
  delayMicroseconds(300);
}
```

Per fare variare la frequenza del suono dello speaker in proporzione al colore (e quindi al led) fanno invece la sequenza ② nel ciclo di qui led mettendo invece di 200 un valore più alto o più basso in frequenza oppure mettere Delay (rumore/1000);

(ATTENZIONE: con PCT > Temperature > frequenza)