

# Sintetizzatore di Frequenza

William Perri 4427140

20/06/2020

Insegnamento di LABORATORIO DI ELETTRONICA A.A. 2019/20

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Descrizione del Progetto</b>	<b>4</b>
<b>3</b>	<b>Realizzazione</b>	<b>15</b>
3.1	Parte su Deeds . . . . .	15
3.2	Parte su bread-board . . . . .	16
<b>4</b>	<b>Risultati</b>	<b>18</b>
<b>5</b>	<b>Conclusioni</b>	<b>19</b>
<b>6</b>	<b>Riferimenti</b>	<b>20</b>

# 1 Introduzione

Lo scopo del progetto consiste nel realizzare un sintetizzatore di frequenza basato su PLL (Phase Locked Loop). Un PLL è composto da un rivelatore di fase, un filtro passa-basso ed un VCO (Voltage Controlled Oscillator).

Il progetto verrà dapprima simulato mediante software di simulazione ed in seguito realizzato in parte su breadboard ed in parte su una FPGA (Field Programmable Gate Array) Cyclone II, programmata mediante il software Quartus II sulla base del progetto realizzato su Digital Electronic Deeds.

## 2 Descrizione del Progetto

Come già detto nel capitolo precedente, il nostro sintetizzatore si baserà su un PLL il cui schema a blocchi è il seguente:

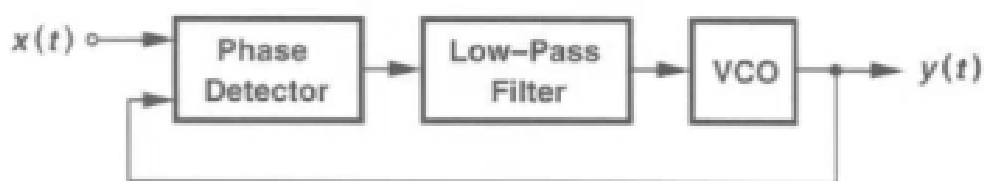
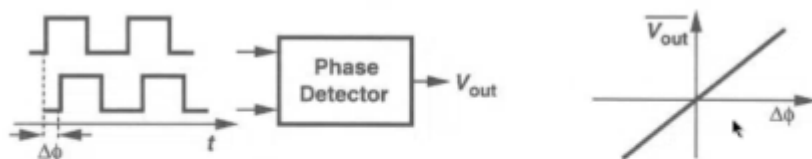


Figura 1: Schema a Blocchi PLL



$$\overline{v_{\text{out}}} = K_{PD} \Delta\phi, \quad (8.2)$$

Figura 2: Il Phase Detector lega la differenza di fase degli ingressi alla tensione d'uscita.

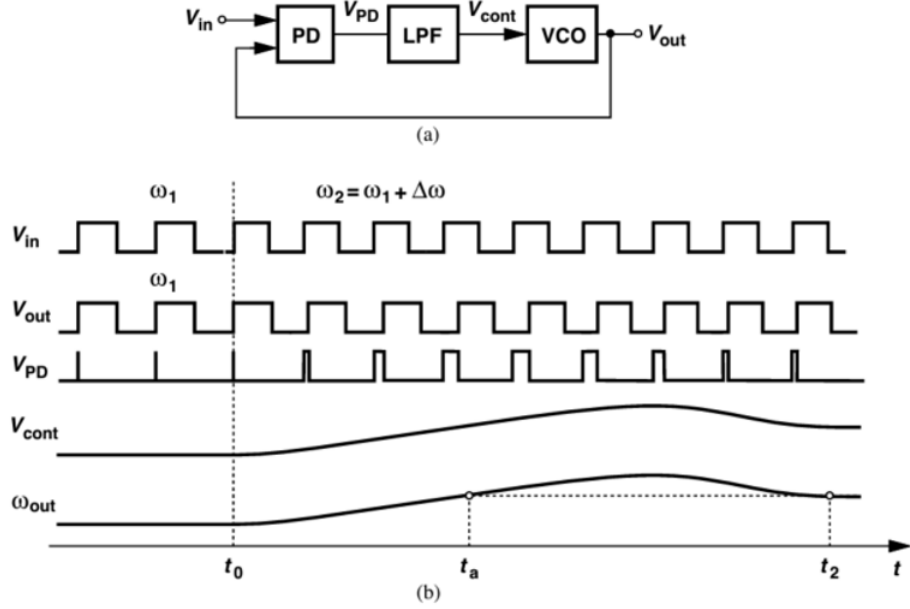


Figura 3: Risposta ad un gradino in frequenza

Analizzando la Figura 3, possiamo capire che se la tensione in uscita dal VCO è nulla (o al valore di riferimento), esso oscillerà ad una frequenza  $\omega_0$ , per modificare questa frequenza c'è necessità di cambiare la tensione di controllo  $V_{CONT}$  in uscita dal LPF, la differenza di fase tra  $V_{IN}$  e  $V_{OUT}$  permette a  $V_{PD}$  di generare un  $\Delta V$  per modificare la frequenza del VCO. Il LPF deve "pulire" la tensione di controllo dai piccoli picchi che la tensione  $V_{PD}$  avrebbe. Nell'istante  $t_0$  la frequenza del segnale d'ingresso cambia, il periodo della  $V_{in}$  diminuisce, l'impulso  $V_{PD}$  si allarga facendo aumentare  $V_{CONT}$ , all'istante  $t_a$ , a causa delle dinamiche dell'anello, la tensione  $V_{CONT}$  continua a salire, facendo aumentare conseguentemente la frequenza di oscillazione del VCO, fino a quando non si stabilizzerà a  $\omega_1$  all'istante  $t_2$ .

Andando ad esplicitare il LPF come un filtro del primo ordine, otteniamo quanto segue in Figura 4

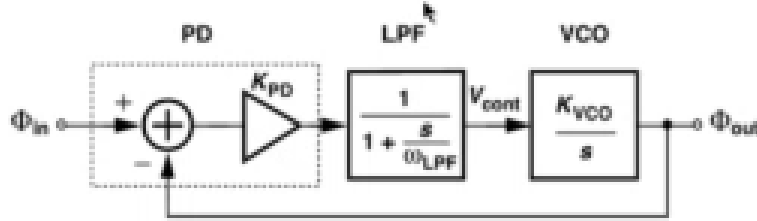


Figura 4:

Abbiamo la proporzionalità data dal PD, un polo dato dal LPF ed infine l'integratore che rappresenta il VCO. Possiamo calcolare le funzioni di trasferimento, sia ad anello aperto, sia ad anello chiuso ottenendo:

$$H_0(s) = K_{PD} \frac{1}{1 + \frac{s}{\omega_{LPF}}} \frac{K_{VCO}}{s} \text{ e } H(s) = \frac{K_{PD} K_{VCO}}{\frac{s^2}{\omega_{LPF}^2} + s + K_{PD} K_{VCO}}$$

Quanto detto fin'ora però presenta un problema: abbiamo

$\omega_{LPF} \ll \omega_{IN} - \omega_{OUT}$ , pertanto la differenza non riesce a passare nel LPF e il circuito non riesce ad agganciarsi. Per il nostro progetto quindi avremmo bisogno di un rivelatore che non tenga conto solo della fase ma anche della frequenza: PFD (Phase-Frequency Detector, Figura 5).

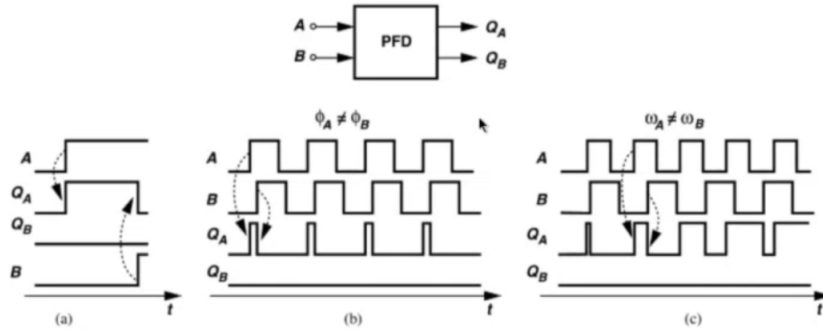


Figura 5: Schema di un PFD

Se il fronte di salita del segnale  $A$  arriva prima del fronte del segnale  $B$ , il segnale d'uscita  $Q_A$  va a 1 fino a quando non arriva il fronte di  $B$ , quindi la larghezza dell'impulso di  $Q_A$  è proporzionale alla differenza di fase tra i due segnali. Al contrario ovviamente se  $B$  fosse in anticipo rispetto ad  $A$ , accadrebbe la medesima cosa però sul segnale  $Q_B$ . Questo dispositivo è stato implementato come si può vedere in Figura 6.

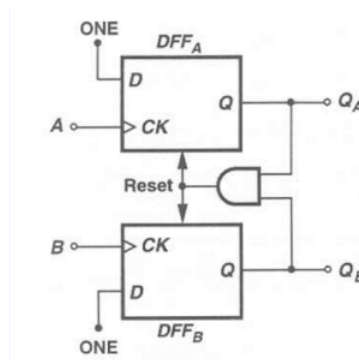


Figura 6: Realizzazione di un PFD

A questo punto andiamo a realizzare un pompa di carico da aggiungere in cascata al PFD

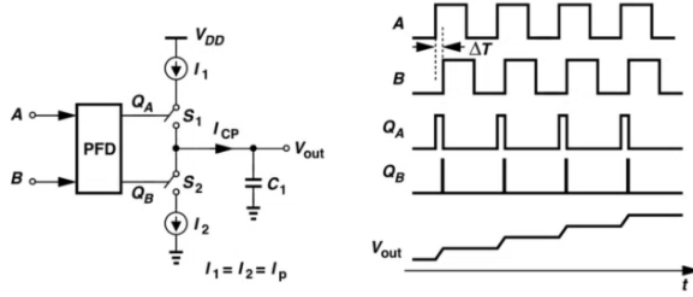


Figura 7: Rappresentazione simbolica della Charge Pump

Se  $Q_A$  è a 1, chiude lo switch  $S_1$ , facendo caricare il condensatore  $C_1$ , facendo aumentare la frequenza del VCO. L'uscita  $Q_B$  invece fa scaricare lievemente il condensatore, in quanto gli impulsi di  $Q_B$  sono molto più stretti di quelli di  $Q_A$ , il condensatore  $C_1$  sostituisce anche il LPF, generando però un problema, la nuova funzione di trasferimento è  $H(S) = \frac{I_P K_{VCO}}{2\pi C_1 S^2 + I_P K_{VCO}}$

Questa presenta due poli immaginari nell'origine, rendendo così il sistema instabile. A questo punto per ovviare a questo problema si deve aggiungere uno zero in zero alla funzione di trasferimento, aggiungendo un resistore prima del condensatore. Il fattore di smorzamento diventa  $>0$ , il margine di fase, che in precedenza era nullo, ora è aumentato. Questo non è l'unico aspetto positivo di questa realizzazione, perché anche all'aumentare del guadagno, vediamo che



il margine di fase aumenta ulteriormente di conseguenza (Figura 8), migliorando la stabilità.

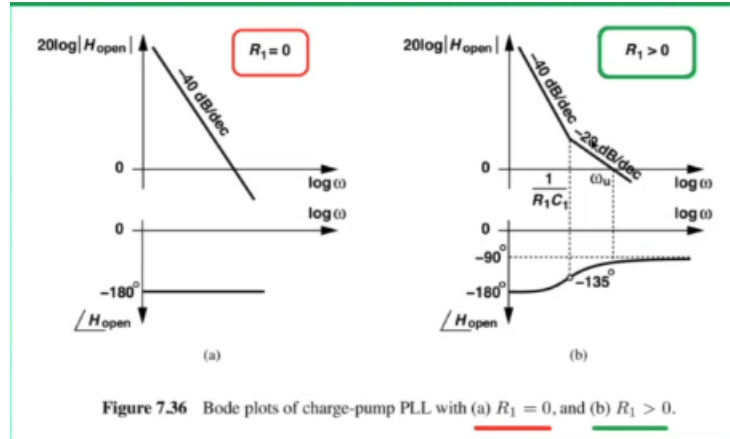


Figura 8: Il margine di fase aumenta

Nella Figura 7 possiamo vedere che sarebbero necessari due generatori di corrente(ideali). Ovviamente non possiamo avere dei generatori ideali, quindi approssimeremo i due generatori come due generatori reali, quindi con una  $r_p$  in parallelo al generatore, che possiamo trasformarlo in un equivalente Thevenin quindi con la  $r_p$  in serie. A questo punto possiamo utilizzare due buffer Tristate, uno con ingresso a 1, l'altro con l'ingresso a 0. Collegando le due uscite  $Q_A$  e  $Q_B$  del PFD agli input enable dei due buffer, avremo una approssimazione dei nostri due generatori (Figura 9). In serie alle due uscite dei buffer andremo poi ad inserire due resistenze  $R_{P1}$  e  $R_{P2}$  che sarebbero quelle dei generatori reali.

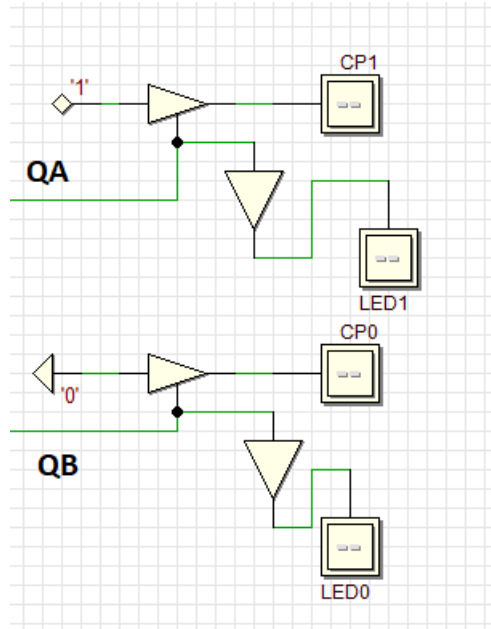


Figura 9: Realizzazione su Deeds dei due generatori di corrente

Il segnale  $CP1$  e  $CP2$  sono le uscite effettive della pompa di carico che andranno sul LPF, mentre le due uscite  $LED0$  e  $LED1$  sono state inserite per vedere i led della FPGA quando il nostro sistema sta modificando la frequenza di oscillazione. Ci sono infinite soluzioni per dimensionare i componenti necessari in questo blocco del nostro sistema, in questo caso  $R_{P1}$  e  $R_{P2}$  sono state scelte pari a  $27k\Omega$  quindi le correnti che scorrono nei due rami sono  $I_1 = I_2 = \frac{V_{DD} - V_{CONT}}{R_{P1}}$  essendo  $V_{CONT} = V_{DD}/2$   $I_1 = I_2 = 60\mu A$ . A questo punto dovremmo scegliere una  $R_1$  (quella del LPF) di valore significativamente più bassa delle due appena scelte, pertanto scegliamo  $R_1 = \frac{R_{P1}}{10} = 2.7k\Omega$ . Per dimensionare  $C_1$  andremo a prendere la formula del fattore di smorzamento  $\zeta$  che vedremo in seguito, dopo aver aggiunto il divisore per M, e lo

porremo uguale a 0.8/1 che ci porterà alla stabilità del sistema.

A questo punto possiamo inserire un divisore di frequenza.

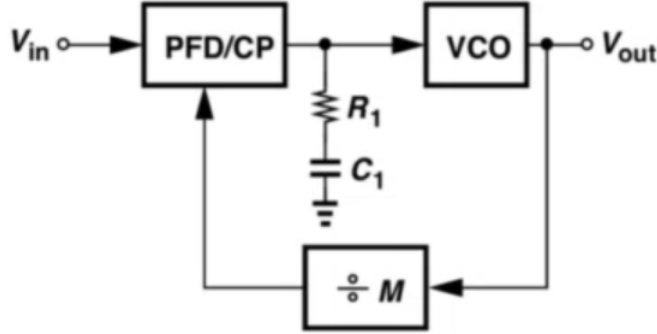


Figura 10: PLL con pompa di carico e divisore per M

Il  $K_{VCO}$  sarà noto in quanto parametro di costruzione. Con l'inserimento del divisore per M abbiamo  $\omega_n = \sqrt{\frac{I_p K_{VCO}}{2\pi C_1 M}}$  e  $\zeta = \frac{R_1}{2} \sqrt{\frac{I_p K_{VCO} C_1}{2\pi M}}$  a questo punto possiamo andare a dimensionare  $C_1$  sapendo che  $\zeta \approx 0.8 \div 1$ ,  $K_{VCO} \approx 800$  per  $C_{EXT} = 120pF$ ,  $I_P = 60\mu A$  e  $M = 100$ . Ponendo  $\zeta \approx 0.94$  otteniamo una capacità  $C_1 = 1\mu F$

Per il nostro progetto dovremo tenere conto delle richieste che sono:

$$f_{OUT} = 1MHz$$

$$f_{REF} = 10kHz$$

$$M = 100(80 \div 150).$$

Il VCO che utilizzeremo sarà il 74LS624, sono presenti due alimentazioni separate, una per la parte logica ed una per la parte di oscil-

lazione, perché il costruttore ci dice che per frequenze  $>10\text{MHz}$  è consigliato l'utilizzo di alimentazioni separate, in modo tale che i rumori provenienti dalla parte logica non influiscano sulla frequenza di oscillazione, infatti se la tensione di alimentazione avesse delle lievi oscillazioni la frequenza di lavoro cambierebbe. Tuttavia, nel nostro caso, utilizzeremo una singola alimentazione.

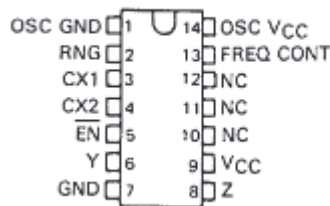


Figura 11: Pinout del VCO

Come si diceva prima, si vedono infatti i pin delle due alimentazioni separate (1-14, 7-9), il piedino di enable, nel nostro caso, lo lasciamo sempre collegato a GND per farlo lavorare ogni volta che il VCO è alimentato. Il piedino 13 è quello di ingresso della tensione di controllo che proviene da un buffer, il motivo per cui è stato inserito un buffer tra il LPF e il VCO è che dal piedino di input ci sono delle resistenze che potrebbero costituire un carico non trascurabile, pertanto inserendo un buffer realizzato con un amplificatore operazionale andremo ad ovviare a questo problema.

I pin 3 e 4 sono per una capacità esterna che noi abbiamo preso  $C_{EXT} = 120\text{pF}$ .

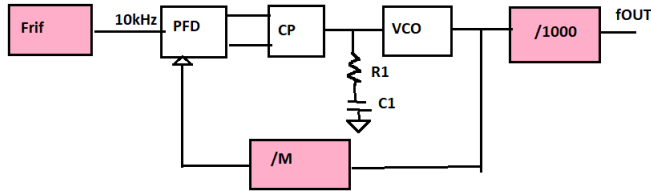


Figura 12: Schema a blocchi dei dispositivi da realizzare.

Il divisore per  $M$ , realizzato sull'FPGA, presenterà uno dip switch, da 8 interruttori, per selezionare il numero  $M$ . Il DIP switch verrà collegato a due array di resistenze, quando uno singolo switch sarà ON, l'ingresso sulla FPGA verrà visto come basso, perché collegato a massa (attraverso le due resistenze), mentre, se lo switch è su OFF, l'ingresso della FPGA vede il valore logico alto (3.3V).

A questo punto l'uscita del VCO avrà un'uscita massima  $>4V$  e dobbiamo portarlo in un range tale da poterlo usare come ingresso al divisore, quindi alla FPGA, per cui dobbiamo utilizzare un Level Shifter per portarlo a 3.3V.

Lo scopo di questo blocco è quindi quello di traslare il livello logico 1 dal valore che ci dava il VCO ( $>4V$ ) a 3.3V, mediante l'utilizzo di un transistor NPN 2N2222 (o similare), la resistenza di pull-up è stata scelta da  $1.5k\Omega$  e quella in serie all'ingresso è stata presa da  $15k\Omega$ .

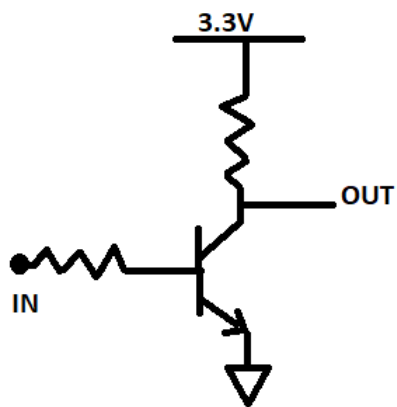


Figura 13: Level Shifter

Lo schema a blocchi completo di ogni componente è quindi il seguente:

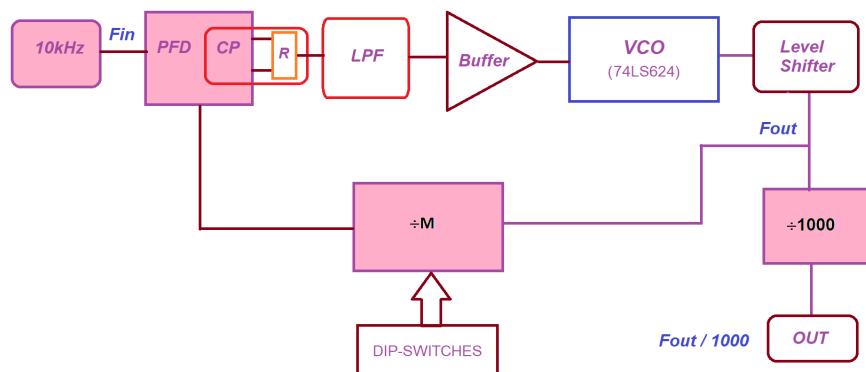


Figura 14: Schema a blocchi completo

### 3 Realizzazione

La realizzazione pratica è quindi suddivisa in due step, il primo su Deeds ed il secondo su breadboard.

#### 3.1 Parte su Deeds

La prima cosa fatta per realizzare il sintetizzatore è stata partire dall'implementare sul Digital Electronics Deeds tutte le componenti che dovranno poi essere messe sulla FPGA Cyclone II.

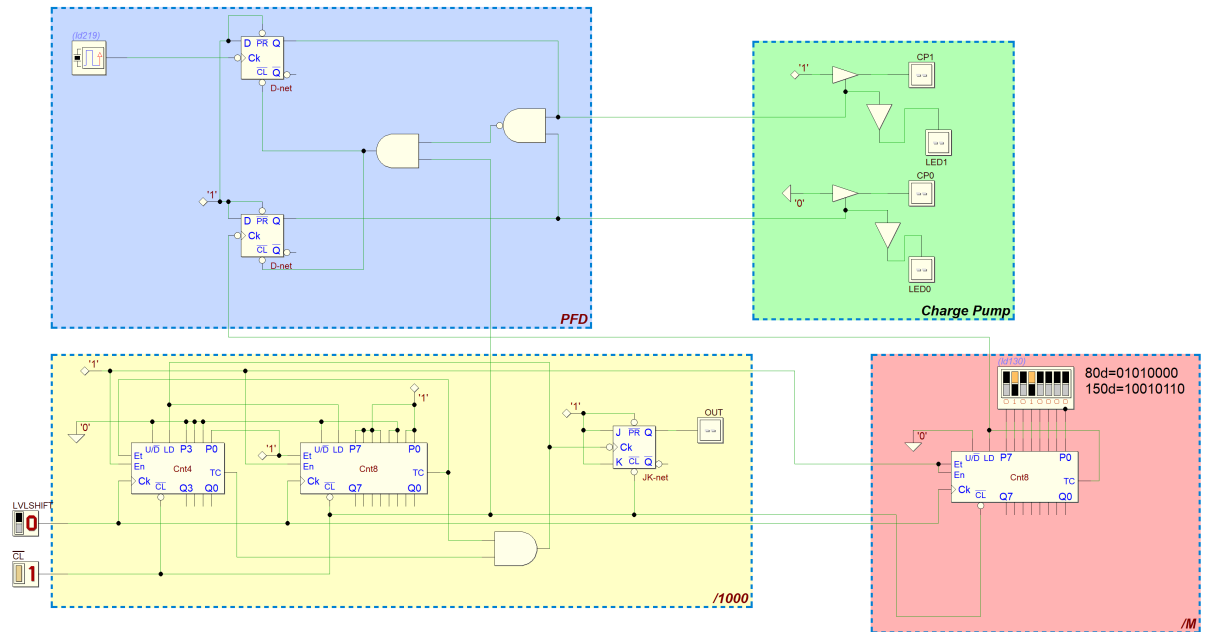


Figura 15: Schema realizzato su Deeds

In azzurro c'è il rilevatore di frequenza e fase, le cui uscite vanno alla pompa di carica (in verde).

In Giallo e in rosso possiamo vedere i due divisori, realizzati con due contatori. A sinistra c'è il divisore per mille, realizzato caricando il numero 000111110011 ovvero 499, siccome conta fino a zero, impiegano 500 cicli di clock per arrivare a zero, il flip-flop JK nella configurazione in figura ha il compito di dividere ulteriormente a metà la frequenza, ottenendo così il desiderato divisore per mille.

Con lo stesso principio, in rosso, abbiamo il divisore per M variabile, il numero M viene scelto mediante gli switch in alto. L'ingresso di entrambi i divisori è ovviamente l'uscita del level shifter realizzato su breadboard come spiegato nel capitolo precedente.

L'uscita finale è l'uscita Q del Flip Flop JK e il clock in alto a sinistra è la frequenza di riferimento.

Mediante il software Quartus II, il progetto è stato esportato sulla FPGA, assegnando ad ogni Ingresso/uscita utile un suo pin da collegare poi alla breadboard.

### 3.2 Parte su bread-board

Per il nostro progetto c'era bisogno di interfacciare la parte software con quella hardware, come detto prima sono stati assegnati i pin della FPGA nel modo seguente:



FPGA Board: Signal Assignment Summary			
Deeds View Board View			
Input/Output Connections → Board Resources:			
In/Out Name	VHDL Name	Board Resource	Aux. Info
<b>Inputs:</b>			
ICL	inCL	Push-Button: Key0	Low (if pressed)
Id130.0	iID0130(0)	Header 0: P1 [IO_48]	
Id130.1	iID0130(1)	Header 0: P1 [IO_52]	
Id130.2	iID0130(2)	Header 0: P1 [IO_55]	
Id130.3	iID0130(3)	Header 0: P1 [IO_58]	
Id130.4	iID0130(4)	Header 0: P1 [IO_60]	
Id130.5	iID0130(5)	Header 0: P1 [IO_64]	
Id130.6	iID0130(6)	Header 0: P1 [IO_67]	
Id130.7	iID0130(7)	Header 0: P1 [IO_70]	
Id219	iID0219	Clock: 10 KHz	
LVLSHIFT	iLVLSHIFT	Header 2: P3 [IO_142]	
<b>Outputs:</b>			
OUT	oOUT	Header 3: P4 [IO_32]	
CP1	oCP1	Header 0: P1 [IO_41]	
CP0	oCP0	Header 0: P1 [IO_40]	
LED0	oLED0	LED (Red): LED0	
LED1	oLED1	LED (Red): LED1	

Figura 16: Assegnazione dei pin

Il clear è stato messo sul push button della FPGA, i PIN dal 48 al 70 sono stati usati per collegare i dip switches alla FPGA, e l'uscita del level shifter (che andrà ai due divisori) è stato assegnato al pin 142. Su ogni componente sono stati messi dei condensatori di bypass sulle alimentazioni per ridurre al minimo le oscillazioni che la tensione di alimentazione potrebbe avere. Lo schema del circuito realizzato su breadboard è il seguente:

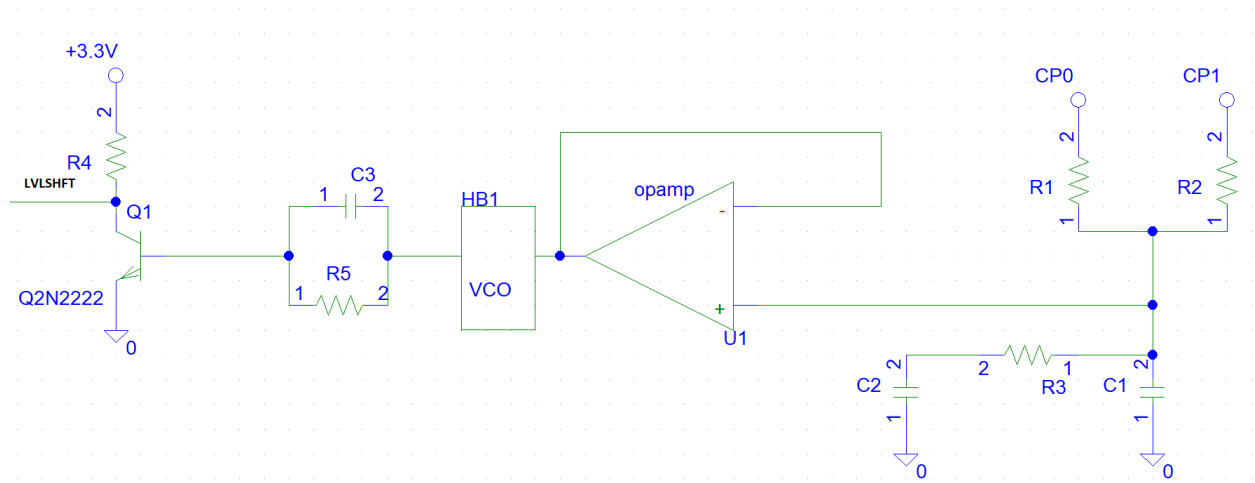


Figura 17: Schema circuitale realizzato

## 4 Risultati

Per visualizzare l'onda d'uscita si è utilizzato il software SoundScope che, interfacciandosi con la scheda audio del computer, mostra la forma d'onda in ingresso sul canale microfono. Di seguito vengono riportati diverse uscite al variare di M

## 5 Conclusioni

Per concludere, si è prima realizzata la parte su FPGA e verificato il suo corretto sviluppo mediante test con segnali noti, solo in seguito è stata realizzata la parte su bread-board dopo aver opportunamente dimensionato i componenti(vedi capitolo 2). Si è utilizzato il software SoundScope per visualizzare a schermo l'onda d'uscita, la frequenza e l'analisi spettrale.

## 6 Riferimenti

Contiene sia la bibliografia (libri, articoli, manuali) sia i siti web eventualmente impiegati durante il progetto