



# Designnotat

Tittel: Tonegenerator

Forfatter: Eirik Mathias Silnes

Versjon: 1.1

Dato: 15. desember 2023

## Innhold

<b>1</b>	<b>Problembeskrivelse</b>	<b>2</b>
<b>2</b>	<b>Prinsipiell løsning</b>	<b>3</b>
2.1	Hvit støy . . . . .	3
2.2	Aktivt filter topologi . . . . .	3
<b>3</b>	<b>Realisering</b>	<b>5</b>
3.1	Hvit støy . . . . .	5
3.2	Aktivt filter . . . . .	5
3.3	Test og resultat . . . . .	6
<b>4</b>	<b>Konklusjon</b>	<b>9</b>
	<b>Referanser</b>	<b>10</b>

---

# 1 Problembeskrivelse

I dette designnotatet skal det designes to ting som tilsammen vil utføre en oppgave. Oppgaven er å filtrere frem en spesifikk tone ut av hvit støy. De to delene som skal designes er en hvit støy generator og et aktivt filter. Hvit støy generatoren skal designes ved hjelp av en FPGA og skal generere hvit støy som skal mates inn i det aktive filteret. Det aktive filteret vil filtrere ut støyen så det kun er en spesifikk tone igjen. Denne tonen skal så kunne spilles av på en høyttaler eller PC. I figur 1 er det vist en oversikt over hvordan sluttssystemet kan se ut. Videre skal dette notatet ta for seg:

- Forestlått design og implementasjon av sluttssystemet
- Responsen, båndbredden og Q-faktoren til det aktive filteret
- Resulterende spektrum for  $V_s$  og  $V_o$

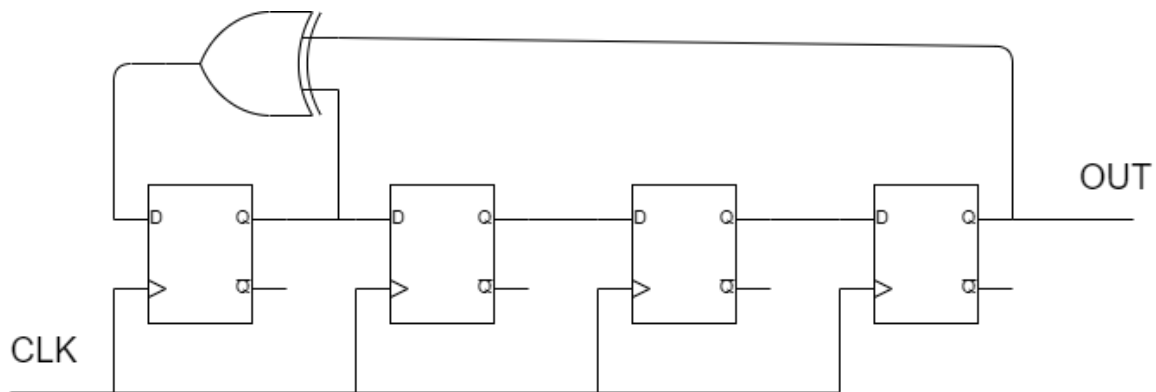


**Figur 1:** Sluttssystemet kan se ut som dette

## 2 Prinsipiell løsning

### 2.1 Hvit støy

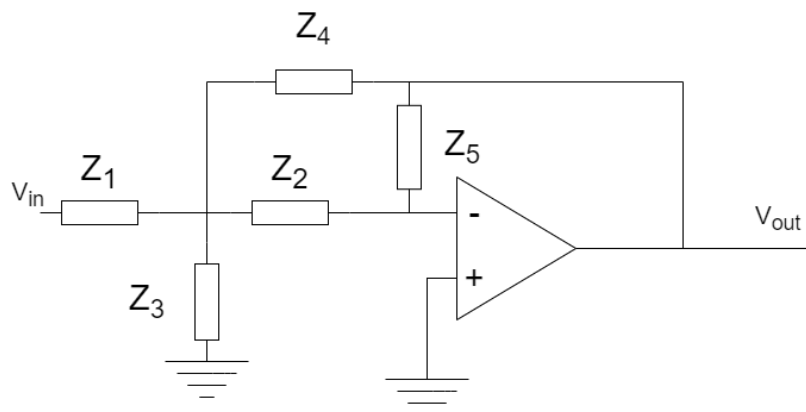
Hvit støy er et signal som er helt flatt på spektrums analysen og har en konstant effekt over alle frekvenser. For å kunne generere hvit støy trenger man en pseudo tilfeldig algoritme som kan tilfeldig velge hvilken frekvens som sendes ut av FPGAen. Det kan løses ved å implementere LFSR (Linear Feedback Shift Register) som er en algoritme som genererer en pseudo tilfeldig sekvens av bits. LFSR er en algoritme som er lett å implementere i hardware og er derfor mye brukt i FPGAer. LFSR er en algoritme som bruker XOR operasjoner for å generere en pseudo tilfeldig sekvens av bits ved en høy frekvens. Ettersom bitsene kommer ut i en tilfeldig rekkefølge vil frekvenskomponentene sin styrke i signalet være jevnt fordelt. LFSR implementeres ved å sette mange D-vipper etterhverandre og koble noen utganger i en XOR som vist figur 2.[4]



Figur 2: LFSR

### 2.2 Aktivt filter topologi

Aktive filtre benytter seg av kondensatorer og operasjonsforsterkere for å filtrere bort uønskede frekvenser. Utifra hvilken kretstopologi man velger får man forskjellige type filter og i dette designet skal vi se på Delyiannis-Friend topologien for å lage et andreordens båndpass filter. Delyiannis - Friend topologien er kan konfigureres ved å endre impedansen til hver av komponentene  $Z_1...Z_5$  til filteret som vist i figur 3[5].



**Figur 3:** Delyiannis - Friend-topologi

Verdiene til de ulike komponentene kan bestemmes ut i fra følgende ligninger [5]:

$$f_0 = \frac{1}{2\pi C \sqrt{(R_1 || R_2) R_3}} \quad (1)$$

$$H_0 = \frac{R_3}{2R_1} \quad (2)$$

$$Q = \frac{f_0}{B} = \frac{1}{2} \sqrt{\frac{R_3}{R_1 || R_2}} \quad (3)$$

$$R_1 = \frac{R_3}{2H_0} \quad (4)$$

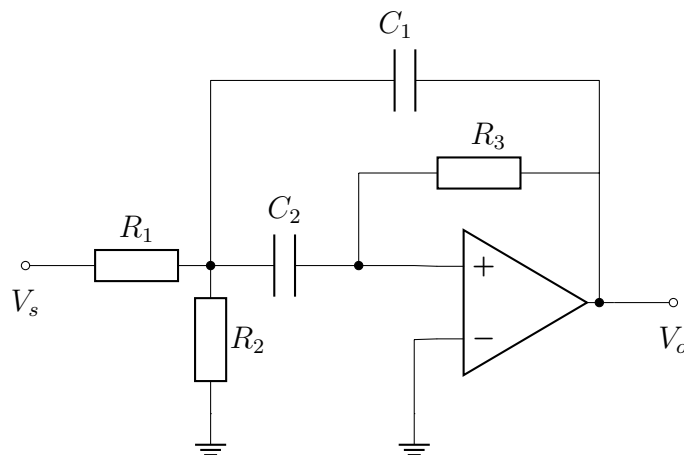
$$R_2 = \frac{R_3}{4Q^2 - 2H_0} \quad (5)$$

$$R_3 = \frac{Q}{\pi f_0 C} \quad (6)$$

En naturlig fremgangsmøte for å bruke ligningene er følgende: [3]

1. Velg en  $Q$ -faktor.
2. Velg amplituderesponsen  $H_0$  for senterfrekvensen. Ønsker man forsterkning kan man eksperimentere med  $H_0$  større enn 1.
3. Velg kondensatorstørrelse  $C$ .
4. Beregn motstanden  $R_3$ , deretter motstandene  $R_2$  og  $R_1$ .

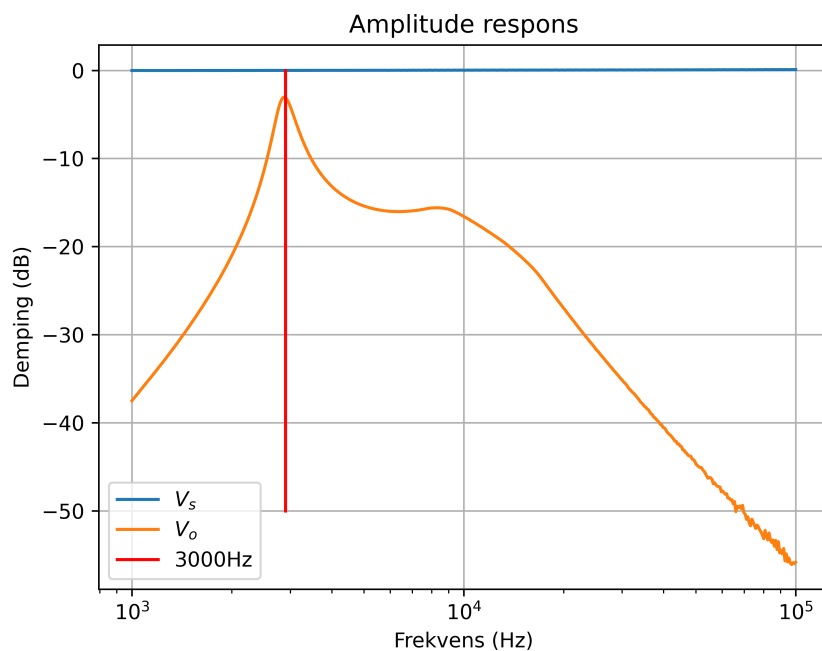




**Figur 5:** Delyiannis-Friend båndpassfilter.

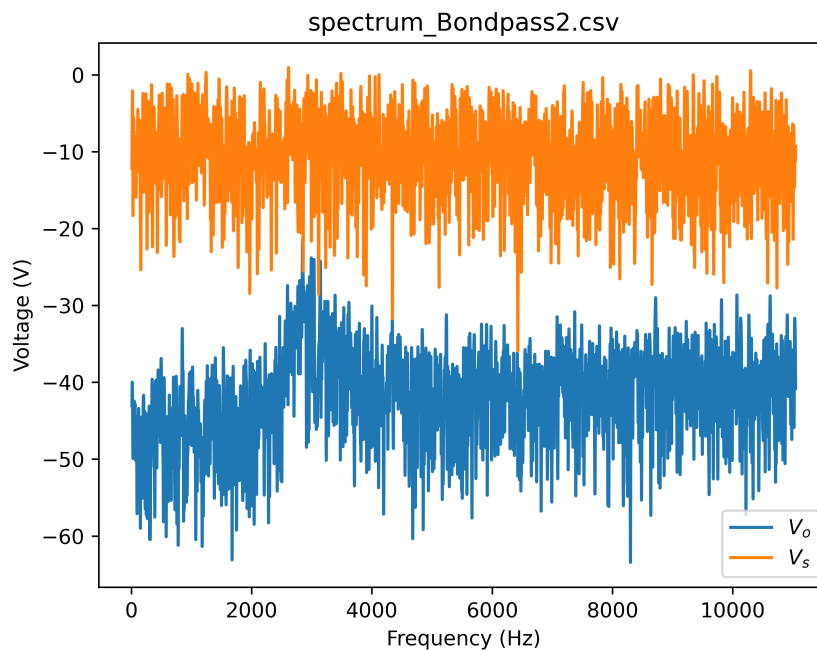
### 3.3 Test og resultat

For å teste om filteret fungerte ble det først gjort en network analyse på filteret i seg selv for å finne amplituderesponsen til systemet. Dette ble gjort ved å benytte et signalgenerator og en oscilloskop. Signalet som ble generert var en sinusbølge med frekvenser fra 0 til 10kHz. Testresultatet kan ses i Figur 6. Filteret hadde knekkfrekvensene 2943Hz og 3071Hz og ut ifra det kan vi se at båndbredden ble litt bredere enn det som ble beregnet tidligere med en målt båndbredde på 128Hz.



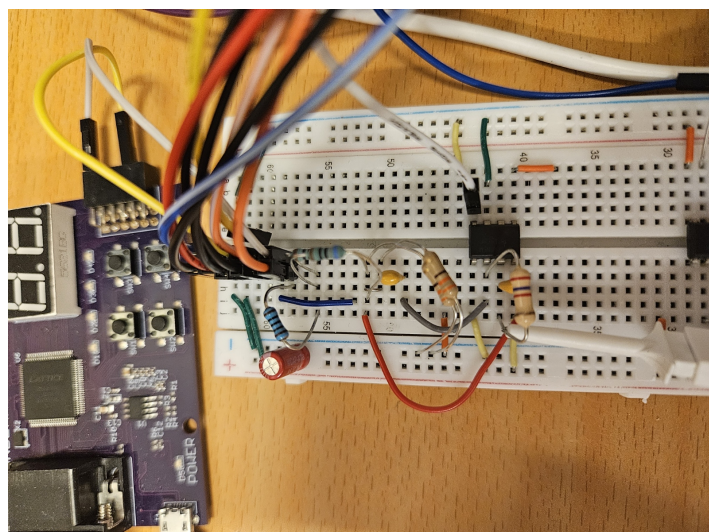
**Figur 6:** Test av filteret

Deretter ble systemet i sin helhet testet ved å koble systemene sammen. I Figur 7 så kan man se signalet  $V_s$  som er signalet som kommer ut av FPGAen og  $V_o$  som er signalet som kommer ut av filteret. Det er tydelig at filteret fungerer som det skal og at det filtrerer bort frekvenser som ikke er i nærheten av senterfrekvensen. Amplituderresponsen til filteret slipper mest gjennom frekvenser rundt 3kHz som forventet men med en veldig redusert amplitude. Den er fremdeles hørbar uten å trenge å forsterkes hvis den spilles av direkte fra WaveForms [6, Digilent Inc.]. Det var også hørbart med støy i bakrunnen som forventet, for å unngå dette kunne man brukt et høyere ordens filter og forsterket signalet etter filteret.



**Figur 7:** Test av systemet

Et bilde som viser den oppkoblede kretsen kan ses i Figur 8.



**Figur 8:** Oppkobling av krets



## 4 Konklusjon

I dette notatet ble det fremstilt et system for å generere hvit støy og isolere ut en tone fra det. For å generere hvit støy ble det brukt en LFSR fremstilt på en FPGA. Videre ble et Delyiannis-Friend filter lagd som isolerte ut en frekvens på 3000 Hz. Videre når man lyttet til signalet hørte man en spesifikk tone som var tydeligere enn resten av støyen, man kunne brukt et høyere ordens filter og forsterket signalet etter filteret for bedre ytelse.

## Referanser

- [1] L. Lundheim, *Designprosjekt 8*, Institutt for elektronisk systemdesign NTNU 2023.
- [2] P. Horowitz, W. Hill, *The Art of Electronics*, Cambridge University Press, 3. utgave, 2016.
- [3] Carl Richard Sten Fosse og Lars Lundheim, *Design av en digital støygenerator*, Institutt for elektronisk systemdesign NTNU, 2023.
- [4] Max Maxfield *Tutorial: Linear Feedback Shift Registers (LFRSs) - Part 1*, <https://www.eetimes.com/tutorial-linear-feedback-shift-registers-lfsrs-part-1/> 2023.
- [5] TLT-8016 *Active Filter and Tuned Circuits*, 2023.
- [6] Digilent Inc. *WaveForms Reference Manual*, <https://digilent.com/reference/software/waveforms/waveforms-3/start>, 2023.