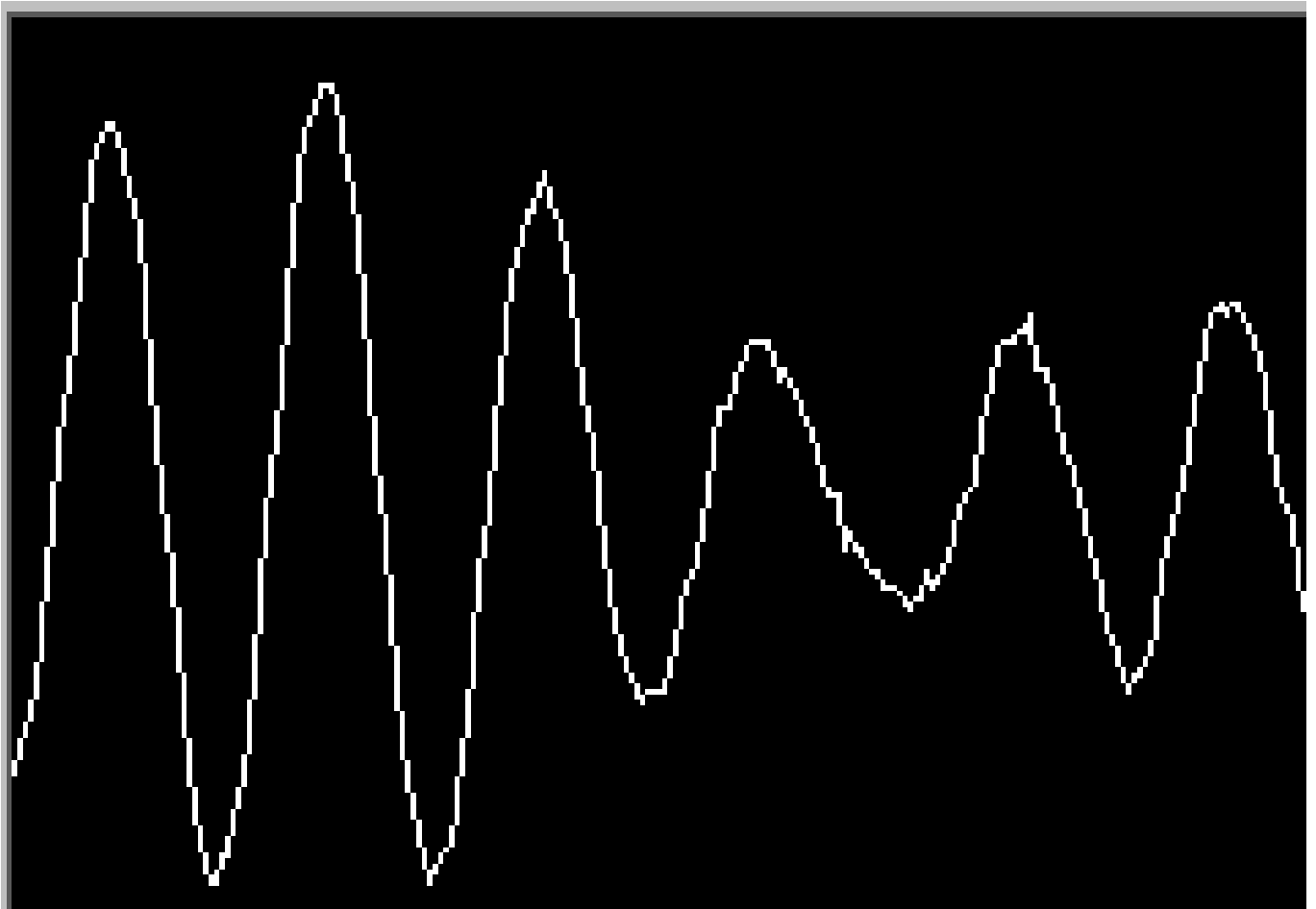




CHALMERS



Mobil Elektroencefalografi i Realtid

Examensarbete inom Data- och Informationsteknik

Oskar Eriksson
Kaixin Chen

EXAMENSARBETE

Mobil Elektroencefalografi i Realtid

Oskar Eriksson
Kaixin Chen

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET

Göteborg 2020

Mobil Elektroencefalografi i Realtid

Oskar Eriksson
Kaixin Chen

© Oskar Eriksson, Kaixin Chen, 2020

Examinator: Peter Lundin

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola / Göteborgs Universitet
412 96 Göteborg
Telefon: 031-772 1000

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Omslag:
Framtagen Elektroencefalografi signal

Institutionen för Data- och Informationsteknik
Göteborg 2020

Sammanfattning

Elektroencefalografi (EEG) har spelat stor roll för studier inom hjärntrauma och mental sjukdom. Det här projektet ämnar att skapa en mobil lösning för en EEG.

I det här projektet har en portabel EEG utvecklats på både mjukvaru och hårdvarunivå. Systemet utvecklades stegvis där varje analog del designades och simuleras i LTspice. När den analoga delen av systemet fungerade som förväntat i simuleringen, användes en kopplingsplatta för att testa det verkliga systemet. Varje delsystem testades så de uppfyllde sin förväntade funktion i systemet. Delsystemen för analog signalbehandling var enligt följande:

- Elektroder: torra elektroder
- Förstärkare: Instrumentförstärkare och icke-inverterande operationsförstärkare
- Filter: lågpass, högpass, notch och kanalfilter
- DC-offset

För att föra över den analoga signalen in i datorn för grafitning användes en A/D Converter (ADC). Ett FPGA kort användes för styrning av ADCn. Själva drivrutinerna är skrivna i VHDL (ett lågnivå programmeringsspråk). FPGA kortet tar emot den digitala signalen från ADCn och skickar den sedan vidare till en dator via en RS232 seriellt utport på FPGA kortet. EEG signalen är plottad i form av grafer på datorskärmen med hjälp av Mjukvaran LabVIEW. Hela systemet fungerar enligt förväntan och graferna liknar existerande EEG data. Men vissa delar av projektet utvecklades bara teoretiskt utan någon praktisk applikation.

Nyckelord: EEG, FPGA, ADC, LabVIEW, VHDL

Abstract

Electroencephalography (EEG) has played a big part for studies regarding brain trauma and mental illness. This project aims to create a mobile solution for an EEG.

In this project, a portable EEG have been developed regarding both hardware and software. The System was developed step by step where every analogue part was designed and simulated in LTspice. When the analogue part of the system worked according to expectations, a breadboard was used to test the physical system. Every subsystem was tested until it fulfilled the expected function in the system. The subsystems for the analogue signal processing was as follows:

- Electrodes: dry electrodes
- Amplifier: instrumental amplifier and a non-inverted operational amplifier
- Filter: low pass, high pass, notch and channel filter
- DC-offset

To transfer the analogue signal into the computer to make a graph, an A/D converter (ADC) was used. a FPGA card was used for controlling the ADC. The drivers are written in VHDL (a low-level programming language). The FPGA card receives the digital signal from the ADC and sends it to a computer via a RS232 serial output on the FPGA card. The EEG signal is plotted in the form of graphs on the computer screen with the help of LabVIEW. The entire system works as expected and the graphs resemble existing EEG data. But some parts of the project were only developed theoretically without a practical application.

Keywords: EEG, FPGA, ADC, LabVIEW, VHDL

Terminologi

- EEG: ElektroEncefaloGraf.
- ADC: Analog to digital converter. omvandlar analoga signaler till en digital motsvarighet.
- FPGA: Field-Programmable Gate Array. är en integrerad krets som används inom digitalteknik.
- PCB: Printed Circuit Board. En platta med isolerande material med mönster för elektriska ledare.
- SPI: Serial Peripheral Interface. En buss för synkron seriekommunikation.
- front to end system: Ett system där "front" styrs av användaren för att utnyttja "end" systemet som ligger från användaren.
- passband: räckvidden för vilka frekvenser som kan passera ett filter.
- RS232: Ett enkelt seriellt protokoll som används för kommunikation mellan dator och FPGA kortet.
- ON bas: Ortonormerad bas. Är ett koordinatsystem där koordinataxlarna är ortogonala (vinkelräta).
- baud: Ett mått för hur många gånger per sekund en signal ändras.
- pseudo differential ADC: En typ av ADC där två inportar används för att få en samplad digital signal. Negativ input för en pseudo differential ADC brukar kopplas till jord.
- Uart: Universal Asynchronous Receiver/Transmitter. Datorhårdvara som kan hantera olika seriella kommunikationsprotokoll.
- CMRR: Common-Mode Rejection Ratio. Mätvärde för hur bra en komponent tar bort utomstående störningar.
- FFT: Fast Fourier Transform. Tar fram amplituden för varje frekvenskomponent.

Förord

Det här projektet hade inte varit möjligt utan all hjälp vi har fått på vägen. Vi ger ett stort tack till vår handledare Sakib som har vart där varje steg på vägen.

Innehållsförteckning

Sammanfattning.....	iv
Abstract.....	v
Terminologi.....	vi
Förord.....	vii
1 Inledning	1
1.1 Bakgrund	1
1.2 Syfte	1
1.3 Mål	1
1.4 Frågeställning.....	2
1.5 Begränsningar	2
2 Metod.....	3
2.1 FPGA som utvecklingsplattform	3
3 Teori	5
3.1 Lösningar för hårdvaruproblem.....	5
3.2 A/D-omvandling.....	6
3.3 Fourieranalys och FFT	6
3.4 RS232 protokoll.....	8
3.5 Biopotentialer	8
3.6 FPGA.....	9
3.7 VHDL.....	9
3.8 Realtidssystem	9
4 Hårdvara.....	10
4.1 A/D-omvandlare: ADS-7818 [1]	10
4.2 FPGA kort: De_1 Cyclone2 [8].....	10
4.3 OP-förstärkare:TL071 [7]	10
4.4 Instrumentförstärkare :INA 118 [4]	11
4.5 RS232 till USB kabel [16]	11
4.6 Elektroder [17]	11
5 Mjukvara	12
5.1 LT spice [9]	12
5.2 Modelsim [10].....	12
5.3 CoolTerm.....	12
5.4 LabVIEW [11].....	12
5.5 Quatrus [12]	12

5.6 Kicad [13]	12
6 Genomförande	13
6.1 Från elektroder till instrumentförstärkare	13
6.2 Högpasfilter	13
6.3 Lågpasfilter	15
6.4 Bandspärrfilter	16
6.5 Kanalfilter	16
6.6 Extra förstärkningssteg	16
6.7 Offsetsteg	17
6.8 A/D-omvandlare	19
6.9 Tidsdiagram för kommunikationsprotokoll	21
6.10 VHDL och FPGA	23
6.11 Verifiering för VHDL och FPGA	23
6.12 Verifiering för seriell kommunikation mellan DE_1 och dator	26
6.13 LabVIEW	28
6.14 PCB kretsschema (skillnaden mellan PCB schema och den framtagna kretsen)...	29
7 Resultat	31
7.1 Mål som har uppnåtts	31
7.2 Mål som inte uppnåtts	31
7.3 Mättningsresultat i form av grafer från LabVIEW	31
8 Diskussion	34
8.1 Svar på frågeställningen	34
8.2 Diskussion om förbättringsalternativ för upplösning av A/D-omvandlare	34
8.3 Diskussion om verifiering och felsökning	35
8.4 Diskussion om EEG kanaler	36
8.5 Problem som tog tid att lösa under projektets gång	36
8.6 Miljö och Etik	37
8.7 Slutsats	37
Referenslista	38
Bilagor:	40

1 Inledning

1.1 Bakgrund

Hjärnan ger ifrån sig signaler hela tiden, vid tankar, rörelser och vid generella uppfattningar. De här signalerna går att analysera för att se hur olika personers hjärnor reagerar vid olika rörelser, eller upplevelser. Det gör det även möjligt att se om en persons hjärna reagerar annorlunda eller inte alls vid försök av rörelser i viss muskulatur, eller vid upplevelser som generellt sett ger stark emotionell stimulans. På så sätt är det möjligt att upptäcka vissa fysiska och psykologiska avvikelser vilket kan orsakas av skador eller sjukdomar. Med hjälp av en EEG kan sådana signaler tas upp för analys, vilket har gjort att verktyget är ett av de viktigaste vid diagnostisering av neurologiska sjukdomar[19].

Ett portabelt EEG system hade öppnat upp möjligheter att hämta data i olika miljöer utanför sjukhuset. Det hade även öppnat upp möjligheter för att hämta data vid alla tidpunkter på dygnet utan några restriktioner för vad patienten får sysselsätta sig med. Det kan öppna upp dörrar för ny data som förhoppningsvis leder till framgångar inom medicin, psykologi och biofysik. Som i sin tur kan leda till framgångar av studier inom till exempel stresshantering och optimala studieförhållanden.

1.2 Syfte

Syftet med projektet är att utveckla en portabel krets som innehåller system för att plocka upp och bearbeta hjärnaktivitet i form av EEG signaler i realtid. Att EEGn är portabel medför mindre begränsningar för användning inom diverse aktiviteter och ger även möjligheten att användas dygnet runt.

1.3 Mål

Målet för projektet är att utveckla ett mobilt EEG-system. Systemet ska inkludera följande delsystem.

1. Förstärkningssteg och filter för signalen
2. A/D-omvandlare som konverterar EEG signalen från analog till digital för att signalen ska kunna behandlas lämpligt av dator och styrmjukvaror.
3. Kontrollhårdvara i form av FPGA som styr A/D-omvandlaren och seriell kommunikation till datorn.
4. Mjukvara som konverterar seriell indata till graf för signalanalys.

5. Funktion enligt principerna för ett realtidssystem.
6. En PCB-layout för hårdvarukonstruktionen.

1.4 Frågeställning

- Finns det möjlighet att förbättra mättningsresultatet av EEG signalen som skickas från A/D omvandlaren till LabVIEW?
- Hur fungerar analys av frekvensspektrum för signaler?
- Hur många EEG kanaler kan byggas upp med hårdvarukonstruktionen?
- Finns det möjlighet att lägga till fler EEG kanaler?

1.5 Begränsningar

- Antalet elektroder som används begränsas till två stycken vilket bara ger en EEG-signal.
- Optimering för brus och gränshfrekvenser av filter genom tester av olika variationer av hårdvara kommer ej utföras.
- Projektet är ingen medicinsk enhet och ska inte användas för medicinska skäl då projektet saknar medicinska licenser.
- Tolkning av signaler kommer inte göras mer än att jämföra med befintlig EEG data.
- Projektet kommer inte ha mallar för en portabel krets då mjukvaran för PCB-layout inte fungerade som förväntat. Att kretsen är portabel innebär att det finns en färdigtillverkad prototyp som är baserad på PCB-layout. Dvs prototypen är tillräckligt kompakt och liten för att kunna vara portabel.
- Finns inga specifika systemkrav för brus.
- Kostnad ingår som en faktor för projektet och det är fördelaktigt att använda befintlig hård och mjukvara, men någon budget finns inte.

2 Metod

Arbetsgången för projektet började med en veckas planering och informationssökning. Vid sökningen gjordes även en plan för vilken ordning projektet skulle utföras i, samt att planering för kommunikation gjordes. Arbetsgången som har använts är en variation av scrum. Scrum är ett tillvägagångssätt som är väl anpassat för projekt där kraven kan vara volatila. Över projektets gång läggs kraven på en lista med ordning av prioritet som lätt kan ändras. Det gjorde att metoden var väl anpassad för projekt som utvecklades under kort tid där mycket kunde ändras, vilket passade bra för det här projektet. I projektet delades varje delmoment in i en "sprint" enligt (Figur 1) där en planering gjordes i början av sprinten och en "sprint review" gjordes i slutet för att se till att delmomentet fungerade enligt förväntan. I varje sprint gjordes "daily scrum" vilket utfördes varje dag (eller vid behov) där dagens åstadkommanden och morgondagens planering togs upp.

2 veckor	bli klar med blocksystem till adc:n i fpgan(modelsim)
2 veckor	koppla den analoga delen och göra tester
1.5 vecka	realtid dataplot i labview för att få fram en graf
1.5 vecka(+1 vecka beställningstid)	design av pcb
1.5-2 veckor	design av huvudbonad
1 vecka	flytta över analoga delen till pcb(löda)
hela tiden(under och efter varje moment)	rapportskrivning och förberedelse inför presentation

Figur 1 (tidsplanering för projektet)

Layouten för projektet vilket var en EEG gick ut på att koppla elektroder från skalpen till en krets som innehöll filter och förstärkning för signalen som sedan kopplades in i en ADC som styrdes av ett FPGA där SPI designades i VHDL kod för att få rätt signal från ADC:n. Signalen in i FPGA:n kopplades sedan vidare in till en extern dator med mjukvara för grafitning (LabVIEW).

Som filter behövdes ett högpas och ett lågpasfilter för att ta bort brus som låg på frekvenser utanför de relevanta upptagna signalerna från elektroderna. Ett notchfilter som designades för 50Hz användes även för att filtrera bort eventuella störningar från kraftnätet.

För förstärkning användes en instrumentförstärkare med ett högt Common-mode rejection ratio (CMRR) vilket reducerade bruset.

Den fysiska kretsen finns i Figur 39.

2.1 FPGA som utvecklingsplattform

Projektet har tillämpning inom sjukvård vilket kan vara tidskritiskt och handla om liv och död. Det innebär att konstruktionen måste vara ett realtidssystem för att kunna garantera en viss output efter varje klockcykel.

Utvecklingsplattform för styrning av A/D omvandlare är begränsad till följande 2 alternativ för att kunna skapa ett realtidssystem:

1. FPGA kortet
2. DSP (står för Digital Signal Processing, det är en typ av Asic som har specifik funktion för att hantera digitala signaler i realtid).

DSP är ett fungerande alternativ, men designsvårigheten av styrningsmjukvaror för DSP är mycket mer komplicerad än FPGA. Därför togs beslutet att konstruktionen baserades på FPGA på grund av familjaritet och att det fanns kort internt på Chalmers.

3 Teori

3.1 Lösningar för hårdvaruproblem

Det är nödvändigt att ta fram ett "front to end" system för att en EEG mätning ska kunna fungera. Två elektroder är kopplade till en instrumentförstärkare, där den elektriska potentialskillnaden mellan elektroderna förstärks. Sedan matas signalen till ett högpasfilter och lågpasfilter i serie för att filtrera ut bassignalen, som har en frekvens mellan 1 till 65 Hz.

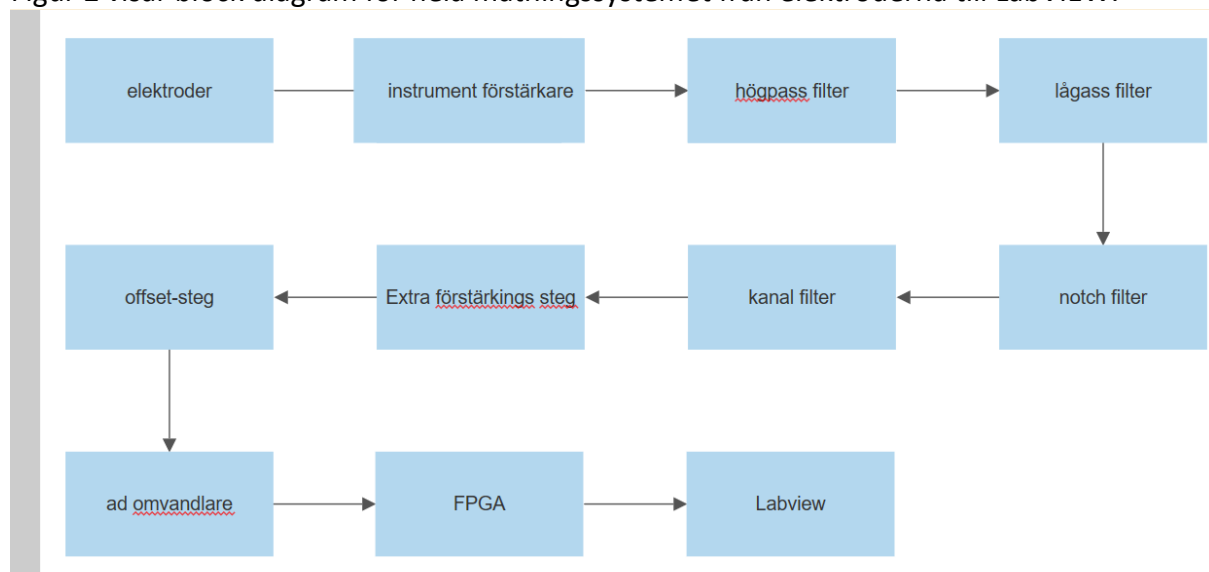
Bassignalen är kopplad till ett notchfilter som tar bort brus från elnätet (vilket är 50 Hz i Sverige). Den här signalen ska sedan passera ett kanalfilter för att ta fram olika EEG kanaler. Kanalfilter är i princip ett bandpassfilter. Passbandet (vilka frekvenser som släpps igenom) är beroende av vilken kanal som analyseras. Exempelvis har alphakanalen ett passband mellan 8 till 12 Hz.

Signalen som förstärks av instrumentförstärkaren är fortfarande inte tillräckligt tydlig, (dvs amplituden är inte tillräckligt stor för att vara läsbar) därför måste ett extra förstärkningssteg läggas till efter kanalfiltret.

A/D-omvandlaren (ADS-7818) som används i projektet kan enbart hantera positiva spänningsnivåer mellan 0 till 5 Volt. Men EEG signalen innehåller både negativa och positiva spänningsnivåer. Därför måste en DC-offset läggas till innan den skickas vidare till A/D-omvandlaren.

Signalen matades sedan till FPGA kortet (De_1) via A/D-omvandlaren. FPGA kortet packar ihop den digitala signalen i form av en seriell signal som är baserad på RS-232 standarden, för att överföra den till datorn via en USB-port. I sista steget levereras signalen över till LabVIEW (Grafiskt programmeringsspråk) där signalen ritas som en graf. En FFT (Fast Fourier Transform) analys görs för att se vilken frekvens som är aktiv.

Figur 2 visar block diagram för hela mätningssystemet från elektroderna till LabVIEW.



Figur 2 (blockdiagram för hårdvarulösning)

3.2 A/D-omvandling

EEG signalen är analog, dvs den är en tidskontinuerlig signal. Den måste omvandlas till en digital signal med hjälp av en A/D-omvandlare för att datorn ska kunna tolka signalen. A/D-omvandlaren samplar signalen med en regelbunden samplingsfrekvens, det innebär att den tidskontinuerliga EEG signalen konverteras till en tidsdiskret signal [2].

A/D-omvandlaren som används i projektet (ADS-7818) samplar signalen med en samplingsfrekvens på 320 Hz. Just den här frekvensen är designad med hjälp av VHDL-kod som styr A/D-omvandlaren.

3.3 Fourieranalys och FFT

FFT (Fast Fourier Transform) används i det här projektet för att i LabVIEW ta fram amplitud av olika frekvenskomponenter av EEG signalen. Det finns ett färdigt FFT funktionsblock i LabVIEW vilket innebär att det inte finns behov att ta fram algoritmen själv.

Det är möjligt att implementera den matematiska metoden för FFT i FPGA kortet (trots att det är inte inkluderat i projektet). Det gör att FFT analys i datorn inte behövs. Genom att begränsa antalet mjukvara och hårdvara som används kan priset reduceras om masstillverkning av en produkt skulle göras. Därför är det bra att vara medveten och ha grundläggande kunskap om den matematiska metoden för FFT.

FFT (Fast Fourier Transform) är ett kraftfullt verktyg för signalanalys inom frekvensspektrum. För att förstå hur FFT fungerar måste den matematiska bakgrunden förklaras för just det här verktyget. Matematiska bakgrunden till FFT är utan tvekan den mest komplicerade teoretiska delen som inkluderas i projektet. Därför är det viktigt att gå igenom det extra noggrant.

FFT är baserat på fourier transform och diskret fouriertransform.

En tidskontinuerlig signal kan beskrivas i form av kombinationer av flera cosinus-signaler som har olika frekvenser (som är multipel med varandra) och fasförskjutningar. Det innebär att det är möjligt att välja dessa cosinus-signaler som basvektorer som bygger upp ett koordinatsystem. Just det här koordinatsystemet kan beskriva vilken signal som helst i form av linjära kombinationer av dessa basvektorer[3]. Det kan tydligt betecknas med följande matematiska formel:

$$s(t) = DC + \sum_{i=1}^{\infty} a_i \times \cos(i \times \omega_0 t + \varphi_i) \quad [3]$$

Observera att $s(t)$ står för själva signalen som önskas att beskrivas. DC är likspänningsamplitud som signalen innehåller. φ_i är fasförskjutning för basvektorer. a_i är amplitud för varje frekvenskomponent.

Men det finns ett problem med att använda cosinussignaler som basvektorer. Det är på grund av att dessa basvektorer inte är ortogonalt normerade. Det gör att det blir svårt att ta fram amplitud a_i för varje frekvenskomponent.

ON-baser (Ortogonalt Normerade baser) innebär att basvektorer är vinkelräta mot varandra (skalärprodukt av basvektorerna är alltså 0) och har en normerad längd som är 1 [3]. Det finns en stor fördel med att välja ortogonalt normerade komponenter som basvektorer i varje koordinatsystem om signalen betraktas som en vektorkomponent. Med hjälp av ON-baser är det möjligt att beskriva signalen (i vektorform) med följande matematiska formel:

$$V = \sum_{i=1}^m (V \cdot B_i) \times B_i \quad [3]$$

Observera att m är antalet komponenter som finns i vektor V . $(V \cdot B_i)$ är skalärprodukten mellan vektor V och olika basvektorer (beroende på vilken index basvektorn har). Nu går det att se att amplitudkomponenter enkelt kan tas fram mha $(V \cdot B_i)$ vid användning av en ON-bas.

$\sum_{i=1}^{\infty} a_i \times \cos(i \times \omega_0 t + \varphi_i)$ måste skrivas om till en uttryck som är baserat på ON-baser. Det kan göras enkelt med hjälp av Eulers formel:

$$\sum_{i=1}^{\infty} a_i \times \cos(i \times \omega_0 t + \varphi_i) = \sum_{i=-\infty}^{\infty} c_i \times e^{ji\omega_0 t} \quad [3]$$
(Obs observera att $c_i = 0.5 \times a_i \times e^{j\varphi_i}$)

Det kan konstateras att de nya ON basvektorerna är $e^{ji\omega_0 t}$. Men det är inte självklart att det är basvektorer som ger en ON-bas. Ett tydligt ON samband syns vid omskrivning av uttrycket mha eulers formel:

$$e^{ji\omega_0 t} = j\sin(i\omega_0 t) + \cos(i\omega_0 t)$$

Det finns en sinus och en cosinus komponent som är vinkelrät mot varandra i uttrycket. Därför finns ett ortogonalt samband.

Nu kan signalen beskrivas med hjälp av de ortogonalt normerade baserna som har valts med följande matematiska formel:

$$S(t) = \sum_{i=1}^m (S(t) \cdot e^{ji\omega_0 t}) \times e^{ji\omega_0 t}$$

$S(t) \cdot e^{ji\omega_0 t}$ är skalärprodukt mellan den önskvärda signalen och basvektorerna.

Skalärprodukten kan även betraktas som amplitud för olika frekvenskomponenter som signalen innehåller. dvs:

$$c_i = S(t) \cdot e^{ji\omega_0 t}$$

Signalvektor V och basvektor B_i kan betraktas som signaler som innehåller samma tidkomponent T . Då är det möjligt att skriva om skalärprodukten med följande integralformel enligt definition för skalärprodukt av två signaler med samma tidsparameter T :

$$c_i = \frac{1}{T} \times \int_0^T S(t) \times e^{-ji\omega_0 t} dt \quad [3]$$

Det här uttrycket är alltså definitionen av en fouriertransform.

Det finns också DFT (diskret fouriertransform) som i princip är fourieranalys i diskret form:

$$c_i = \frac{1}{T} \times \int_0^{\text{antal sampelingar}} S(nTS) \times e^{-j\omega_0 nTS} dt$$

n står för samplingsindex och TS står för samplingens periodtid (tid mellan varje sampling).

FFT är en algoritm som är baserad på DFT. FFT underlättar mängden beräkningar som DFT utför genom att separera samlingsindex i både udda och jämna delar (grupper).

FFT och DFT ger exakt samma beräkningsresultat i slutändan. Den enda skillnaden är att FFT går mycket snabbare än DFT på grund av den underlättade beräkningsprocessen.

3.4 RS232 protokoll

Kommunikation mellan PC (LabVIEW) och FPGA kortet går via en ledning som är baserad på RS232 standard. Sändaren i projektet (FPGA kortet) skickar ut data via pinnen **TD** (står för transmitted data). Mottagaren i projektet (seriell port i datorn) tar emot data via pinnen **RD** (står för receive data). Det finns också **DSR** (data set ready) och **DTR** (data terminal ready) pinnar som verifierar om respektive sändare och mottagare är redo för att ta emot data. Just den här verifieringen sker automatiskt mellan datorn och FPGA kortet. Det finns också möjlighet att lägga till en extra hårdvaruhandskakning via pinne **CTS** (clear to send) och **RTS** (ready to send). Den extra handskakningen har inte inkluderats i projektet.

Varje RS232 datapaket börjar med en startbit (övergång från logiskt 1 till 0) och en stoppbit (övergång från logiskt 0 till 1). Det finns 8 bitar binär data som ligger mellan start och stoppbitarna som hanteras av LabVIEW för att plotta signalen i form av en graf. Det finns alltså 10 bitar binär data inklusive start och stoppbit i varje datapaket.

Dataöverföringshastighet är fullständigt baserat på baud rate. Baud rate är ett koncept som beskriver hur snabb övergången är mellan varje symbol. Varje symbol i RS232 standard inkluderar bara en binär bit som kan representera antingen logiskt 1 eller 0. Det innebär att baudrate för RS232 är ekvivalent med bitrate (hur snabb övergång det är mellan varje bit).

I projektet styrs baud rate och bit rate av seriella pulser som driver A/D-omvandlaren (pulserna genereras av FPGA kortets interna kristallosillator). Baud rate för projektet har valts till 9600 Hz vilket matchar en standard baud rate inom RS232 protokollet.

3.5 Biopotentialer

En EEG läser potentialskillnader som uppstår i hjärnan och tar sig till skalpen. Enligt [14] uppstår potentialerna när ett fenomen som heter Postsynaptisk potential (PSP) uppstår. Fenomenet uppstår när hjärnan skickar elektriska signaler till en nervcell via något som heter dendriter. Vid överföring hamnar signalen både på insidan och utsidan av nervcellen. När nervcellen har tillräckligt med laddning skickar den ut en signal genom nervtrådarna. Det kallas för en aktionspotential (AP) vilket varar under mycket kortare tid än PSP och är för det mesta inom membranet och medför därför inte mycket till en EEG mätning. För att få en EEG signal behövs den sammanlagda potentialen från PSP av tusentals nervceller som aktiveras

samtidigt. PSPn i det yttre lagret av hjärnan summeras och går delvis ut genom skallen till skalpen. Där går potentialen att mäta med hjälp av elektroder.

3.6 FPGA

FPGA står för Field programmable gate array. Det är i princip ett kretskort som innehåller miljontals (eller mer) logiska grindar som är packade i form av arrayer. Det finns möjlighet att styra alla dessa logikgrindar (nästan samtidigt) inom en klockcykel. Därför finns det möjlighet att skapa ett realtidssystem mha av FPGA.

3.7 VHDL

VHDL (Very High Speed Integrated Circuit Hardware Description Language) är ett programmeringsspråk som är brett använt inom styrning av FPGA (Field programmable gate array) och ASIC (application specific integrated circuit). VHDL kodning är baserat på olika processer som går parallellt (samtidigt) med varandra. Det finns möjlighet att skapa dessa processer med hjälp av en kristalloscillator i FPGA kortet. Det innebär att systemdesignen i projektet kan bete sig som ett realtidssystem som är baserat på klockfrekvensen av kristalloscillatorn.

3.8 Realtidssystem

Ett realtidssystem innebär att det finns ett skarpt tidskrav. Systemet måste kunna garanteras att reagera inom ett visst tidsintervall.

4 Hårdvara

4.1 A/D-omvandlare: ADS-7818 [1]

ADS-7818 är en Pseudo differential A/D-omvandlare med 12 bitars upplösning. Den inkluderar även ett enkelt seriellt interface som matar ut data. Det är en stor fördel för implementering av projektet, på grund av att drivrutinerna för A/D-omvandlaren är skrivna i VHDL (very high speed hardware description language). Det är ett lågnivå (dvs nära själva hårdvaran) programmeringsspråk, vilket innebär att det tar extra lång tid för kod design och verifiering.

A/D-omvandlaren har ett enkelt seriellt interface som inte kräver skrivning och läsning av olika kommandon på registernivå vilket underlättar belastningsnivån för arbetet. Det krävs också en extern klocksignal som genereras av FPGA kortet för att driva komponenten.

4.2 FPGA kort: De_1 Cyclone2 [8]

Mätning av EEG signalen ska fungera i ett realtidsbaserat system. Det innebär att det finns ett skarpt tidskrav för läsning av signalen. Drivrutinerna för FPGA kortet som sätter igång A/D-omvandlaren och hanterar signalen skrevs med hjälp av VHDL kod. Viss binär utsignal efter varje sampling vilket matchar det skarpa tidskravet för ett realtidssystem kan förväntas.

De_1 är ett FPGA kort som innehåller de nödvändiga funktionerna för implementering av projektet.

Följande funktioner i FPGA kortet används för projektet:

1. 50 MHz intern klock oscillator: används för att generera seriell klocksignal för A/D-omvandlaren
2. GPIO-pinnar: används för att koppla till A/D-omvandlarens ingång och utgång för att överföra seriella klocksignaler och läsa den digitala utsignalen.
3. RS232 kommunikationsblock: används för att överföra de 8 mest signifikanta bitarna av den digitala signalen till PCn.
4. Switch: används för resetsignal

4.3 OP-förstärkare: TL071 [7]

TL071 är en lågbrus operationsförstärkare som är tillverkad av Texas-Instruments. Komponenten används för design av både lågpass och högpassfilter i projektet. TL071 är också använd inom konstruktionen av det extra förstärkningssteget och offset steget.

4.4 Instrumentförstärkare :INA 118 [4]

INA 118 är en differential instrumentförstärkare som är tillverkad av Texas Instruments. Komponenten har ett CMRR (Common Mode Rejection Ratio) som ligger runt 110 dB. CMRR är ett mått som mäter hur bra en komponent kan mäta en differential signal.

INA 118 användes för att förstärka differentialsignalen mellan två elektroder som är kopplade till patientens skalp.

4.5 RS232 till USB kabel [16]

FPGA kortet De_1 innehåller en utport som är baserad på RS232 protokoll. En RS232 till USB kabel användes i projektet för att kommunicera mellan datorn och FPGA kortet.

4.6 Elektroder [17]

Elektroder används i projektet för att plocka upp signaler från yttre lagret av hjärnan via skalpen. Torra elektroder används för att fånga upp elektriska potentialer som skickas vidare till resten av systemet. Fördelen med torra jämfört med våta elektroder är att ingen förberedelse av skalpen eller konduktivt gel är nödvändigt. Det gör att de går snabbt att applicera elektroderna utan några större förkunskaper. Torra elektroder har kontakt med skalpen på flera punkter, vilket gör att elektroden plockar upp generell aktivitet inom ett område.

5 Mjukvara

5.1 LT spice [9]

LT spice är ett simuleringsmjukvara för kretsdesign. Det används framförallt vid simulering av frekvenssvar för alla filter i projektet.

5.2 Modelsim [10]

Modelsim är en simuleringsmjukvara för FPGA design. Den användes vid simulering av VHDL kod, för att styra FPGA kortet.

5.3 CoolTerm

Coolterm är en enkel mjukvara som läser utdata från seriella portar i form av hexadecimala tecken. Mjukvaran användes för att verifiera den seriella portens funktion.

5.4 LabVIEW [11]

LabVIEW används för att läsa och plotta utsignalen från seriella porten. Inställning av seriell port kan konfigureras enkelt med hjälp av LabVIEWs Visa-read och Visa-Write blocker.

5.5 Quartus [12]

Quartus används för att överföra VHDL kod till själva DE_1 FPGA kortet. Det kan implementeras enkelt med hjälp av den inbyggda programmer funktionsblocket av Quartus som kopplar variabler i mjukvaran till portar på hårdvaran (FPGA kortet). Observera att FPGA kortet kopplas till Quartus via USB-Blaster port.

5.6 Kicad [13]

Kicad är en Cad mjukvara för PCB-design. Med hjälp av Kicad kan PCB layout för hårdvarukrets genereras. Den kan sedan skickas till fabrikant för tillverkning.

6 Genomförande

I det här kapitlet presenteras genomförandet av projektet.

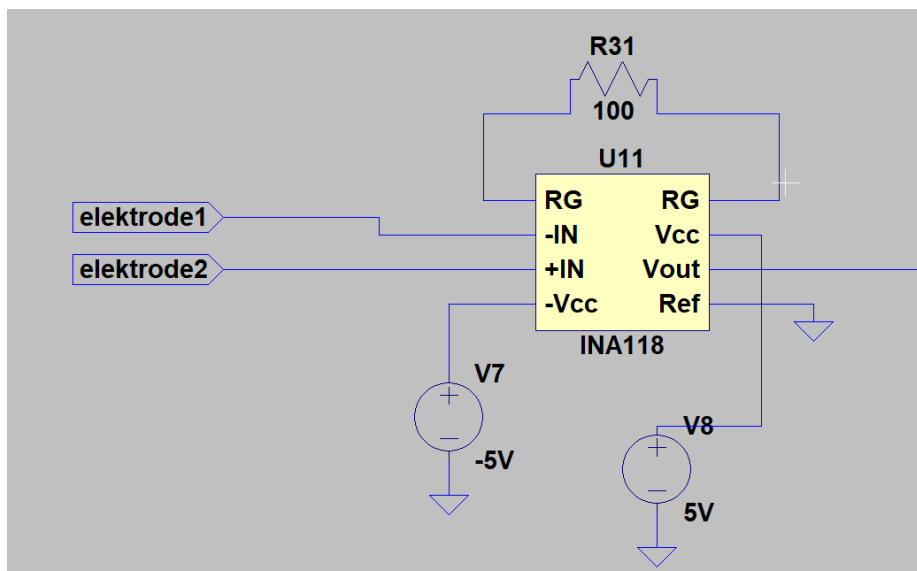
Först beskrivs hårdvarukonstruktionen för mätningssystemet med hjälp av både teoretisk bakgrund och LT-spice simuleringar (från elektroder till A/D-omvandlare). En mer detaljerad beskrivning av hårdvaran finns i kapitel **3.1 lösningar för hårdvaruproblem**.

Sedan används en A/D-omvandlare (ADS-7818). ADS-7818 styrs av FPGA kortet med hjälp av VHDL kod.

Till slut presenteras kommunikation mellan FPGA kortet och LabVIEW. Där den digitala data som representerar EEG signaler (alpha och beta kanaler) plottas i grafen. FFT analys tas även fram.

6.1 Från elektroder till instrumentförstärkare

Två elektroder är kopplade till ingångarna av INA 118. En resistor som har storleken 100 ohm är kopplad till den interna spänningsdelningskretsen för komponenten. En 100 ohm resistorn ger en förstärkning på 500 gånger enligt datablad [4]. Potentialskillnaden (som ligger på mikrovoltnivå) mellan två elektroder (som är kopplade till en EEG kanal) förstärks 500 gånger av instrumentförstärkaren, sedan skickas signalen vidare till filter för att ta ut EEGns bassignal.



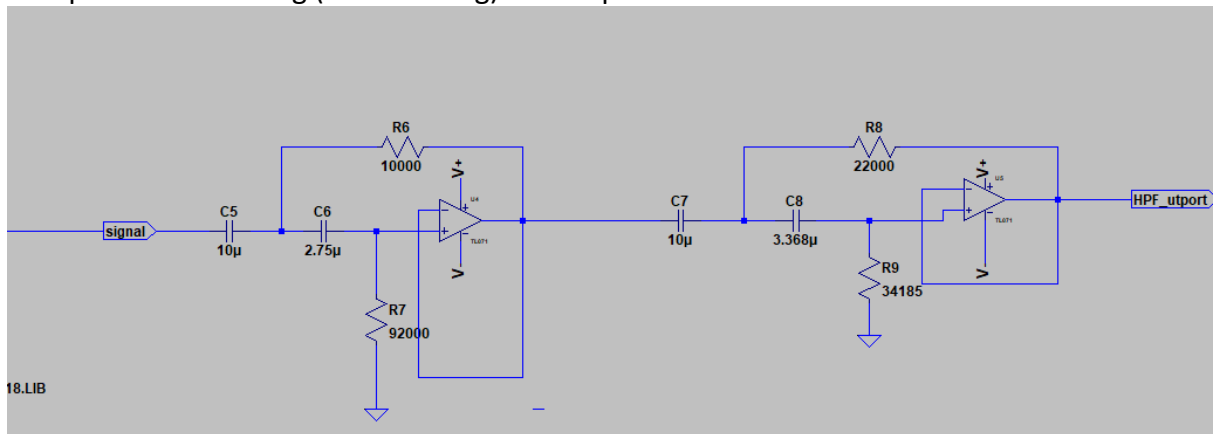
Figur 3 (från elektroder till instrumentförstärkare INA 118)

6.2 Högpassfilter

Frekvensbandet för en EEG signal ligger mellan 1 Hz och 65Hz. Det innebär att högpassfiltret måste ha en brytfrekvens som ligger runt 1 Hz.

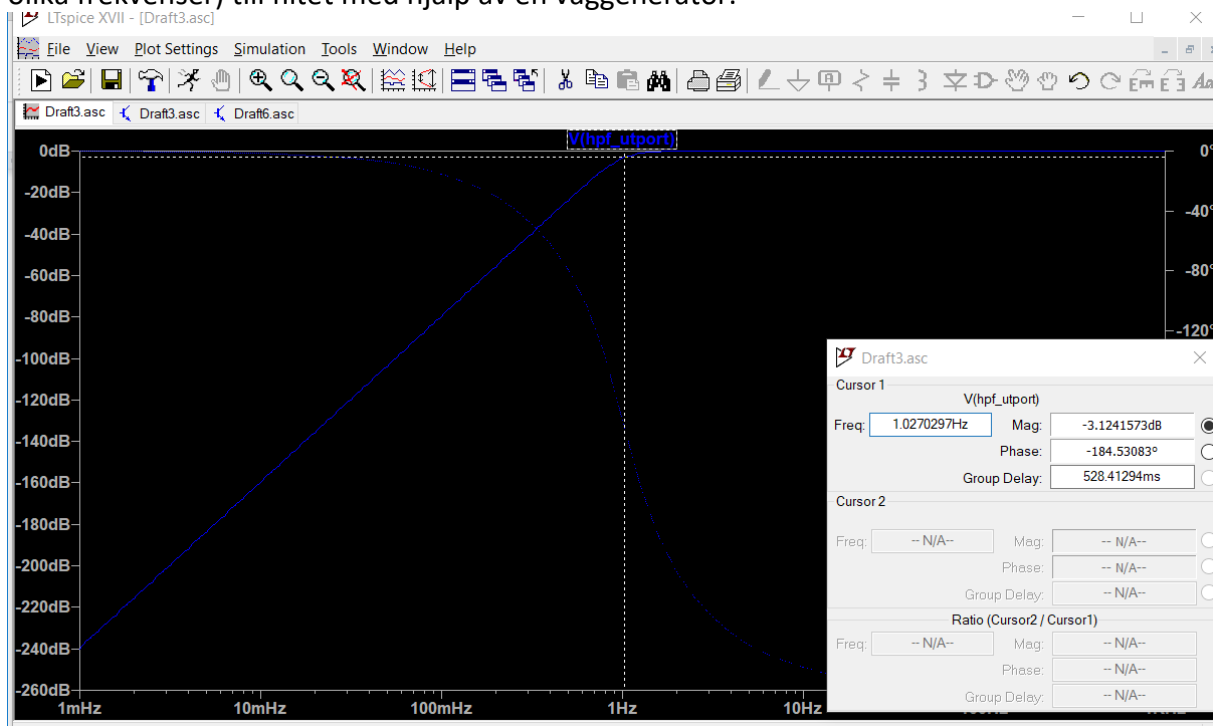
Vid projektet designades ett 4e ordningens Butterworth högpasfilter för att filtrera bort de lågfrekventa störningarna. Det utförs genom att seriekoppla två stycken andra ordningens Butterworth högpasfilter. Principen för Butterworth filter är helt enkel att skapa en undersväng och en översväng vid olika steg (ordning) i filterdesignen. Till slut jämnas översvängen ut med undersvängen, det ger ett platt passband.

Två operationsförstärkare används för att skapa Sallen key-kopplingar. Sallen key-kopplingar innebär användning av operationsförstärkare som spänningsföljare för att undvika att utimpedans i olika steg (eller ordning) i filtret påverkar varandra.



Figur 4 (4th ordning Butterworth högpasfilter)

I småsignalssimulering (Figur 5) syns en brytfrekvens som ligger runt 1 Hz. Det finns också ett hyfsat platt passband. Därför går det att tillgodosäkra att filterdesignen stämmer i simuleringen. Men det finns också behov att verifiera om filtret fungerar i verkligheten (på kopplingsplatta). Verifieringen kan implementeras enkelt genom att mata en AC signal (med olika frekvenser) till filtret med hjälp av en våggenerator.



Figur 5 (simulering för småsignalanalys för lågpasfiltret)

Tabell 1 visar mätningsresultat när en sinusvåg med olika frekvenser är matad från signalgeneratoren till högpasfiltret.

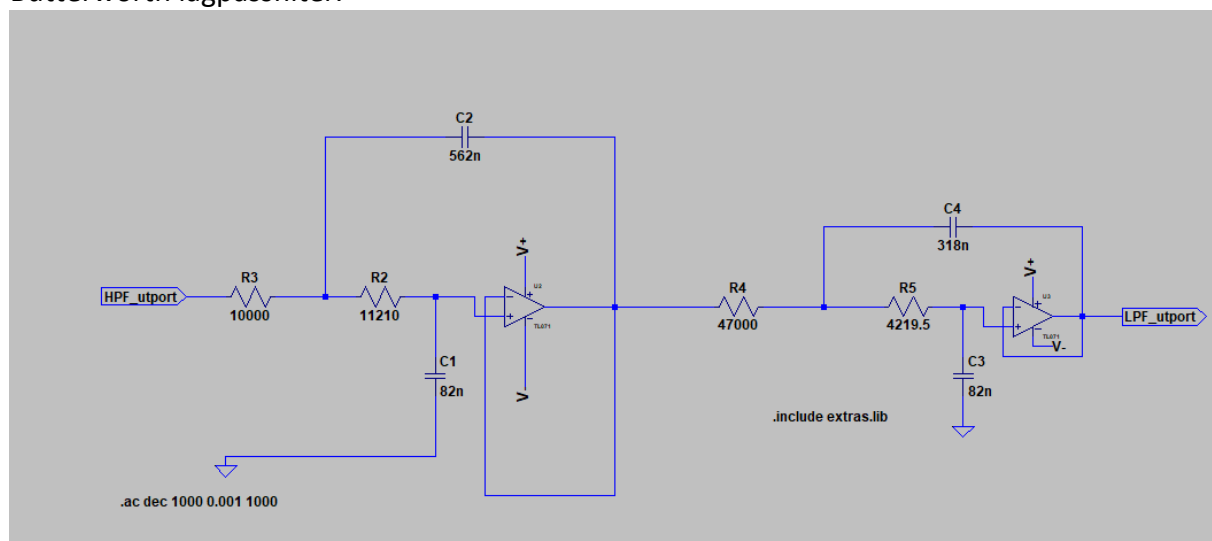
Frekvens	insignal (peak to peak)	utsignal (peak to peak)
2Hz	200 mV	200 mV
1.5Hz	200 mV	196 mV
1HZ	200 mV	124 mV
0.5HZ	200 mV	16 mV
0.1Hz	200 mV	0 mV (det går knappt att läsa signalen från oscilloskopet på grund av att den är så dämpad)

Tabell 1 (mätning för lågpasfilter)

Det syns tydligt att frekvenskomponenter som ligger nära och under 1 Hz är dämpade. Filtrets beteende ser symmetriskt ut jämfört med det simulerade resultatet. Därför kan det konstateras att filtret fungerar enligt förväntan.

6.3 Lågpasfilter

Frekvensbandet för en EEG signal ligger mellan 1 Hz och 65Hz. Det innebär att ett lågpasfilter måste konstrueras för att uppnå en brytfrekvens som ligger runt 65 Hz. Vid projektet designades ett 4e ordningens Butterworth lågpasfilter (Figur 6) för att filtrera bort högfrekventa störningar. Det innebär en seriekoppling av två stycken andra ordningens Butterworth lågpasfilter.

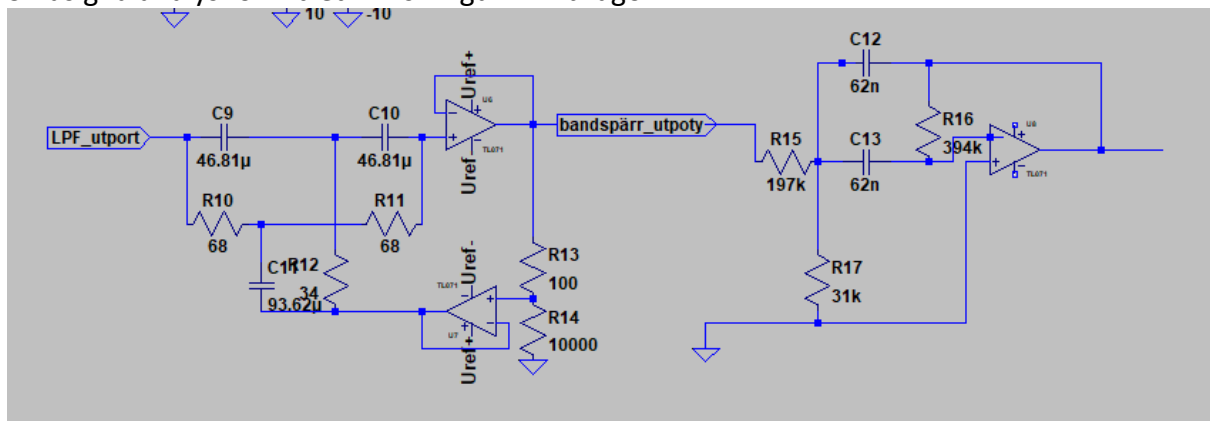


Figur 6 (4e ordningens Butterworth lågpasfilter)

Småsignalanalys och verifieringstabell för lågpåssfiltret finns i Figur 23 och Tabell 3 som finns i bilagor.

6.4 Bandspärrfilter

Ett 4e ordningens bandspärrfilter (Figur 7) är designat för att ta bort de icke önskvärda störningar som kommer från elnätet. Designen som har använts är en twin-T design för att möjliggöra anpassning av bandbredd och dämpning. Det svenska elnätet har en frekvens som ligger på 50 Hz. Det innebär att bandspärrfiltret ska kunna kvittera frekvenskomponenter som ligger runt 50 Hz. Komponenterna räknades ut med hjälp av[6]. Småsignalanalys för filtret finns i Figur 24 i bilagor.



Figur 7 (bandspärrfilter)

6.5 Kanalfilter

Det finns olika kanaler som har olika passband för EEG signalanalys. Det finns två kanaler (Alpha och Betakanalen) som är konstruerade i projektet, men bara en kan visas i taget.

Alphavågen dyker upp när patienten är avslappnad, den har ett frekvensomfång som ligger mellan 8 till 12 Hz. Ett 4e ordningens Butterworth bandpassfilter med ett passband som ligger mellan 8 till 12 Hz är designat för att ta fram alphavågen från EEG bas signalen.

Betavågen dyker upp när patienten är koncentrerad. Den har ett frekvensomfång som ligger runt 12 till 30 Hz. Ett andra ordningens Butterworth bandpassfilter med ett passband som ligger mellan 12 till 30 Hz är designat för att ta fram Betavågen från EEG bassignalen.

Kopplingsschema och frekvenssvar för filterna finns från Figur 25 till Figur 28 i bilagor.

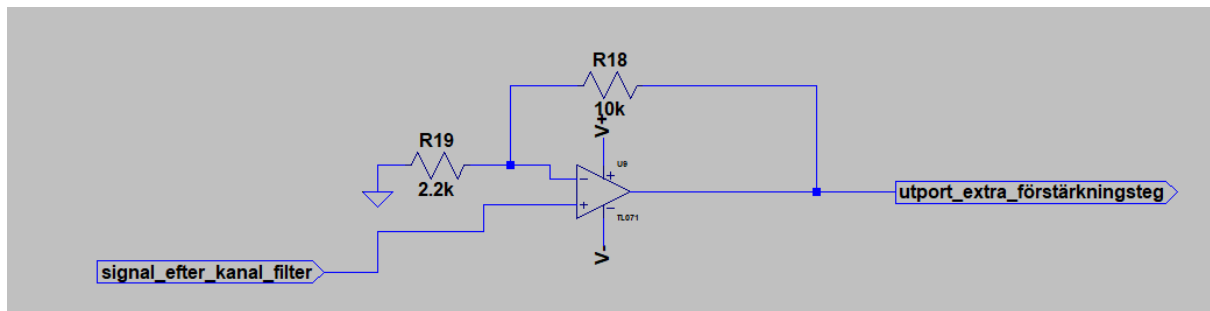
6.6 Extra förstärkningssteg

Signalen som har passerat kanalfiltret är fortfarande inte tillräckligt stark, dvs amplituden av signalen är alldeles för låg. Därför finns det behov att konstruera ett extra icke inverterande

förstärkningssteg (Figur 8) innan signalen skickas vidare. Förstärkningen har ett matematiskt samband som är kopplat till resistor R18 och R19:

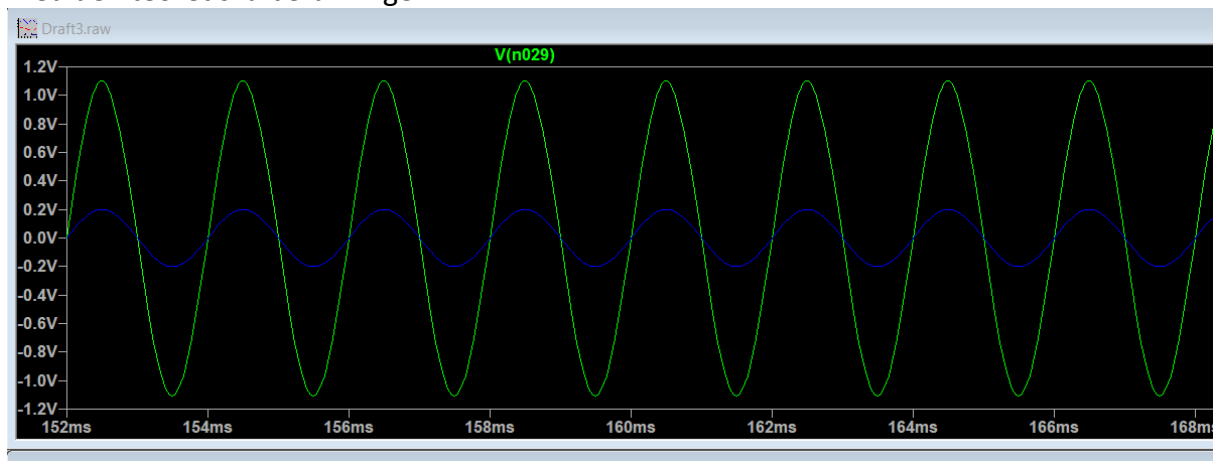
$$\text{Förstärkning} = 1 + \frac{R18}{R19}$$

I labbsal testades konstruktionen med olika resistorer, efter många olika försök konstaterades att det är mest lämpligt att använda 2.2 Kohm för R19 och 10 Kohm för R18. Den här designkonstruktionen ger en förstärkning som ligger runt 5.54 gånger. Signalens amplitud efter det extra förstärkningssteget är varken för hög eller låg för att vara hanterbar av A/D-omvandlaren.



Figur 8 (extra förstärkningssteg)

Figur 9 visar simuleringsresultat för det extra förstärkningssteget vid inmatning av en sinusvåg (som har peak to peak amplitud på 400 mV) i input. Obs att den blå vågformen representerar insignal och den gröna vågformen representerar utsignalen efter förstärkning. Förstärkning av konstruktion enligt Figur 9 ligger runt 5.45 gånger vilket stämmer överens med den teoretiska beräkningen.



Figur 9 (simulering för extra förstärkningssteg)

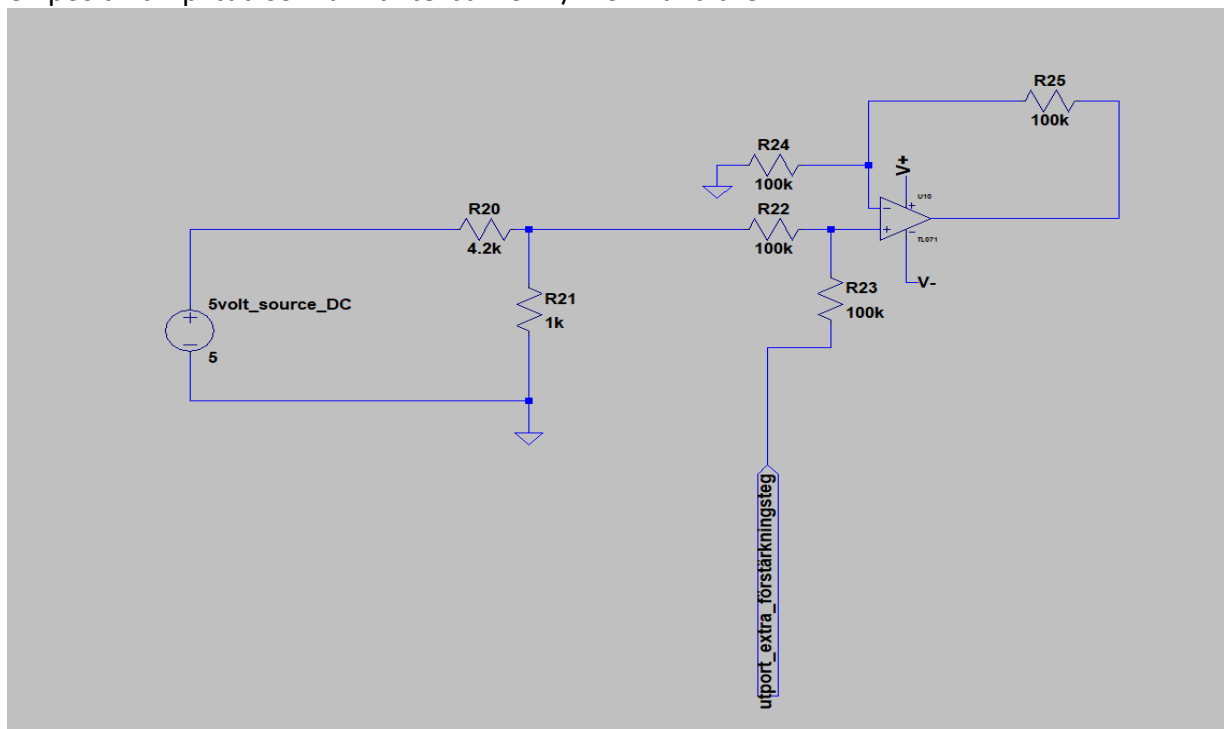
6.7 Offsetsteg

A/D-omvandlaren (ADS-7818) som används i projektet kan enbart hantera positiva amplitudnivåer som ligger mellan 0 till 5 volt. Men EEG signalen innehåller både positiva och negativa amplitudkomponenter. Därför finns det behov av att designa ett offset steg med hjälp av en icke inverterande summeringskoppling (Figur 10).

En 5V DC spänning är kopplad till en spänningsdelningskrets. Storleken av DC offseten kan beräknas med följande matematiskt samband som är baserat på spänningsdelning:

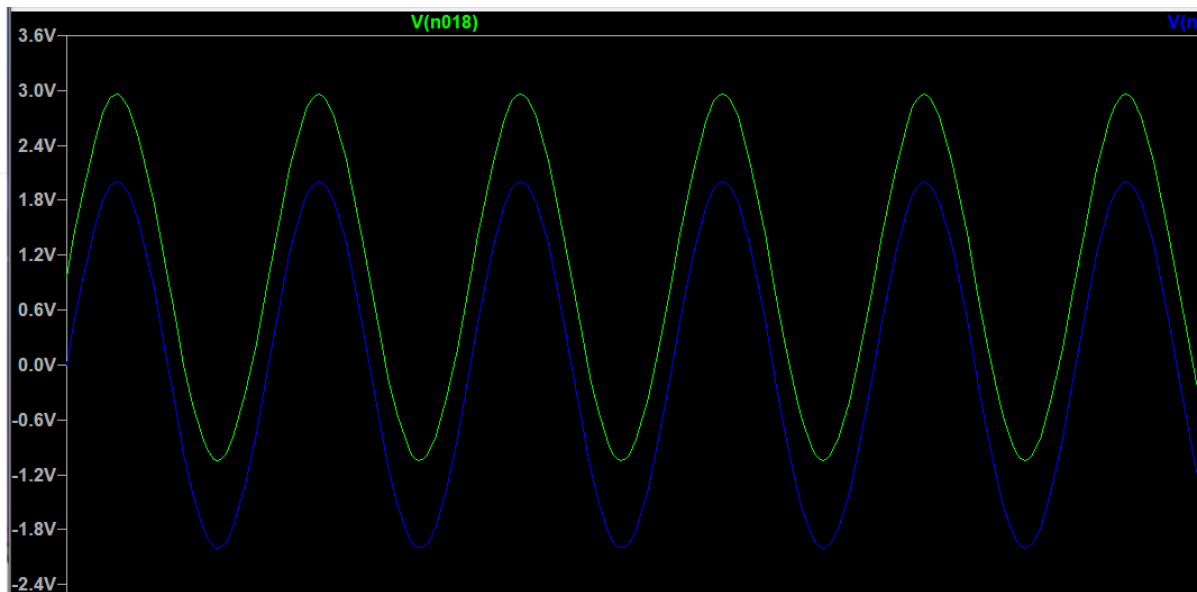
$$V_{out} = V_{in} \times \frac{R_{21}}{R_{20} + R_{21}} = 0.96 V$$

Med många försök i labsalen kan det konstateras att det är mest lämpligt att välja 4.2 kohm och 1 kohm för respektive R20 och R21. Det ger en offsetspänning på 0.96 volt. Den här DC offseten ska adderas ihop med signalen som har passerat det extra förstärkningssteget (med hjälp av en icke inverterande summeringsoperationsförstärkare). Det innebär att den negativa spänningsnivån av EEG signalen för olika kanaler ska kunna förskjutas upp och bli en positiv amplitud som är hanterbar för A/D-omvandlaren.



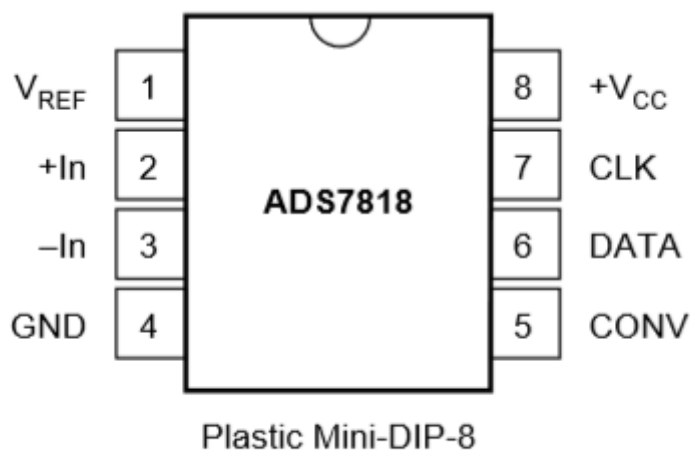
Figur 10 (Offsetsteg tillsammans med spänningsdelning)

Simuleringen i Figur 11 visar att den blå insignalen (en sinusvåg med peak to peak amplitud 4V) förskjuts upp med ungefär 0.96 V efter offsetsteget. Det stämmer överens med den teoretiska beräkningen.



Figur 11 (simulering för Offsetsteg)

6.8 A/D-omvandlare



Figur 12 (ADS-7818 som inte är kopplad till andra komponenter)[1]

ADS-7818 har 8 stycken pinnar som ska kopplas vidare. I det här kapitlet förklaras funktioner av de olika pinnarna och tidsdiagram av dataöverföring.

I projektet används den internt on-board 2.5V reference kretsen av ADS-7818. Därför är pinnen **Vref** kopplad enbart till en bypasskondensator (bypasskondensatorn kvitterar en viss del av AC störningar, **Vref** måste vara en stabil DC spänning). Kapacitansen av bypasskondensatorn är 0.1 μF enligt rekommendation från databladet[1].

Det finns ett matematiskt samband mellan **Vref** och range av den analoga inputen. Den giltigt analoga inputen för konstruktionen kan beräknas med följande formel enligt datablad:

Analog input full scale range = $V_{ref} \times 2 = 2.5 \times 2 = 5V$ [1]

Därför kan det konstateras att input range för analoga signalen är 0-5V.

Upplösning av A/D-omvandlaren kan beräknas med följande formel:

$$\text{upplösning} = \frac{\text{full scale range}}{2^{12}} = \frac{5}{4096} = 1.22 \text{ mV}$$

Men notera att den här upplösningen enbart gäller för konstruktioner som använder alla 12 databitar av ADS-7818. I projektet används bara de 8 mest signifikanta databitarna för att kunna kommunicera mellan FPGA och datorn med hjälp av rs232 protokoll (det finns en mer noggrann beskrivning av RS232 i kapitel **2.4 RS232 protokoll**).

Den verkliga upplösningen för konstruktionen måste beräknas om:

$$\text{upplösning} = \frac{\text{full scale range}}{2^8} = \frac{5}{256} = 19.53 \text{ mV}$$

ADS-7818 är en pseudo differential A/D-omvandlare. Det innebär att det finns möjlighet att separera den analoga jorden från A/D-omvandlarens jord genom att koppla in en pseudo ground till **Vin-**. Det kan kvittera DC common mode spänning. Men den här funktionen används inte i projektet på grund av att själva signal och ADS-7818 delar samma jord enligt konstruktionsdesignen.

Pinnen **Vin+** är kopplad till signalen som passerade alla filter och förstärkningssteg. Pinnen **Vin-** är parallellkopplad med jord i ADS-7818.

Det kräver en extern klocksignal för att driva konverteringsprocessen av ADS-7818.

Pinnen **CLK** är kopplad till den externa klocksignal som är genererad av De_1 FPGA kortet via GPIO pinnen A_13. Frekvensen av klocksignalen är designad för 9600 Hz för att matcha standard baudrate för RS232 protokollet (VHDL koden som genererar klockpulsen finns i kapitlen **FPGA och VHDL**).

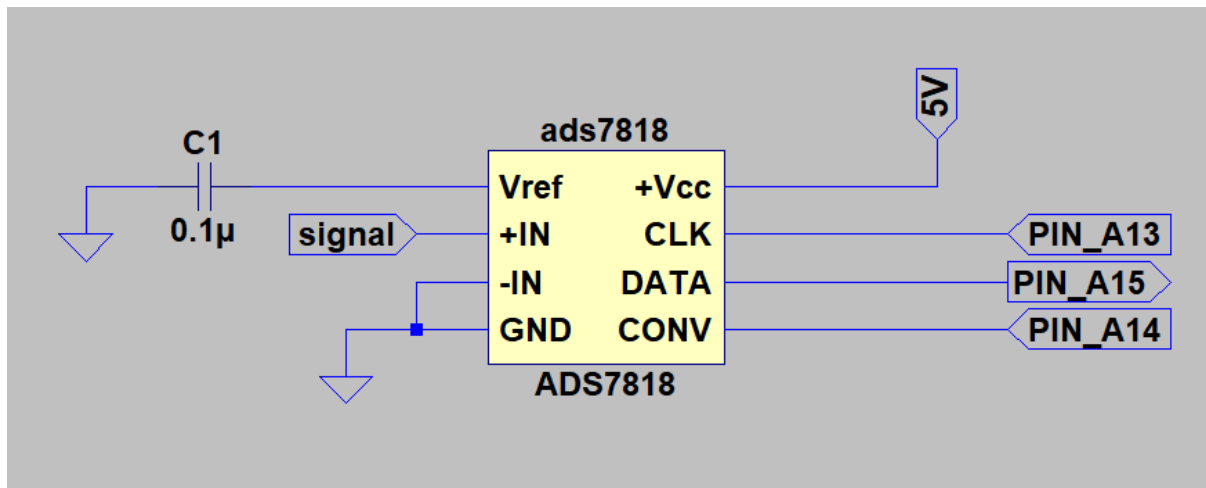
Det finns också ett matematiskt samban mellan samplingsfrekvens och klockfrekvens enligt datablad:

$$\text{Samplingsfrekvens} = \frac{\text{klockfrekvens}}{\text{antal klockpulser för varje sampling}} = \frac{9600}{30} = 320 \text{ Hz} [1]$$

Observera att antalet klockpulser för varje sampling är en parameter som kan justeras beroende på designval med hjälp av VHDL kod. I projektet används 30 klockpulser.

En konverteringssignal **CONV** som genereras av FPGA kortet behövs för att kontrollera sample/hold mode och start av dataöverföring[1]. Pinnen **CONV** är kopplad till GPIO pinnen A_14 på DE1 FPGA kortet.

Seriell data utport **DATA** är kopplad till FPGA via GPIO pinnen A_15. Den digitala signal som överförs ut genom porten innehåller alla 12 bitar data som A/D konverteras av ADS-7818.



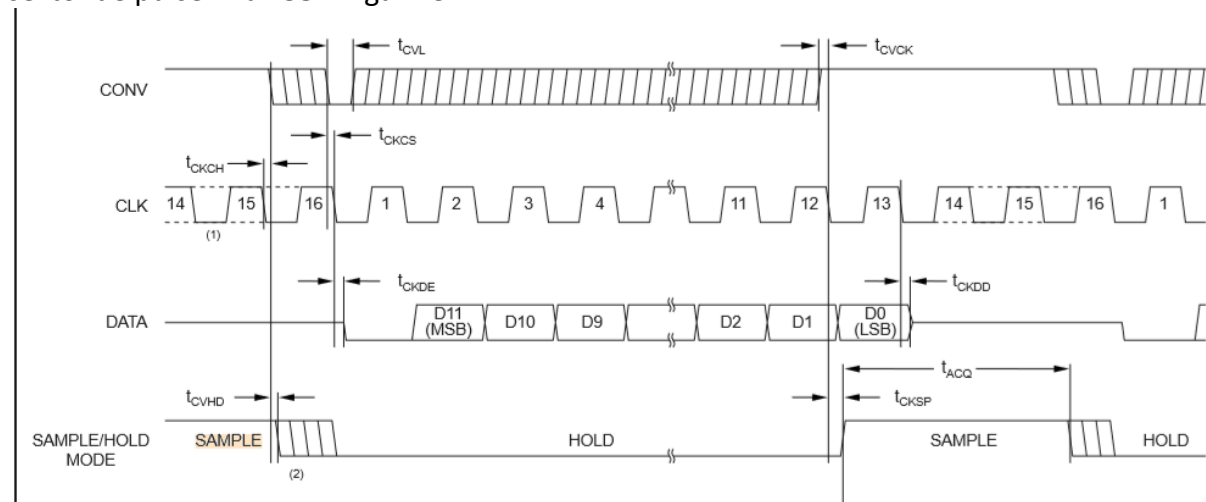
Figur 13 (ADS-7818 som är kopplad till de andra delarna. Notera att LTspice symbol för ADS-7818 inte finns tillgänglig online. En självgjord symbol för komponentens utseende är använd i LTspice för att visar hur kopplingen ser ut)

6.9 Tidsdiagram för kommunikationsprotokoll

Det finns 2 stycken olika operation modes för ADS-7818. Det första kallas för Basic konverterings mode enligt datablad[1]. Figur 14 visar tidsdiagram för basic operation mode.

ADS-7818 går från Sample-mode till Hold-mode när **CONV** pulsen går från högt till lågt vid början av första klocksignalen. Överföring av de 12 databitarna (från MSB till LSB) startas vid andra klockpulsen till den trettonde pulsen.

ADS-7818 går från Hold-mode till Sempel-mode när **CONV** puls går från lågt till högt vid början av fjortonde pulsen. Sampling börjar vid fjortonde pulsen och slutar vid slutet av den sextonde pulsen när **CONV** går ner.



Figur 14 (basic konverterings mode för ADS-7818) [1]

Egentligen finns det inget krav på att det ska ta 3 klockpulser (från puls 14 till 16) om t_{acq} (t_{acq} står för den minsta samplingstiden vilket är 350ns enligt datablad) matchas[1].

Basic konverterings mode har inte riktigt funkat för projektet i labsalen på grund av att kommunikationen mellan datorn och FPGA kortet är baserat på RS232 standarden. Sekvensen av överförda databitar måste gå från LSB (den minst signifikant biten) till MSB (den mest signifikanta biten). Annars tolkar datorn indatan på ett felaktigt sätt.

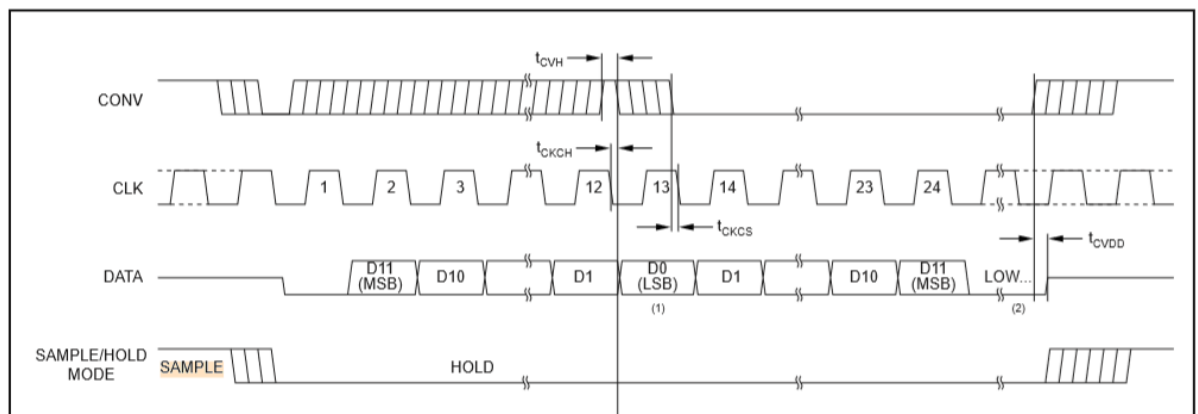
Det är nödvändigt att implementera den andra operations mode som heter LSB mode enligt datablad[1].

Tidsdiagram och synkronisering för det här operations mode är mer komplicerat än basic konverterings mode. Figur 15 visar tidsdiagrammet för LSB-first operation mode.

ADS-7818 går från Sample-mode till Hold-mode när **CONV** puls går från högt till lågt vid början av första klockpulsen. Överföring av 12 databitar (från MSB till LSB) utförs från andra till trettonde klockpulsen. Den första delen funkar exakt likadant som basic konverterings mode.

Genom att dra upp **CONV** pulsen vid tolfte pulsen och sedan dra ner den en gång till vid den trettonde klockpulsen när LSB håller på att överföras, är det möjligt att upprepa dataöverföring med omvänd sekvens (från LSB till MSB) från puls 13 till puls 24.

Det skapar möjlighet att läsa de 8 mesta signifikant bitarna från LSB till MSB med hjälp av VHDL kod. Det går att göra enkelt genom att läsa data från puls 17th till 24th (databitarna som motsvarar de 8 mest signifikanta bitarna).



Figur 15 (LSB-first operation mode för ADS-7818) [1]

VHDL kod som sätter igång operations mode visas i kapitel **6.10 VHDL och FPGA**

6.10 VHDL och FPGA

FPGA kortet DE_1 användes för att styra ADS-7818. Drivrutinen är skriven i programmeringsspråket VHDL utan användning av färdiga bibliotek. VHDL koden är baserad på 3 stycken processer som går parallellt med varandra.

Den första processen genererar den seriella klockpulsen som är kopplad till **CLK** input. Den här pulsen styr konverteringsprocessen tillsammans med **CONV** pulsen. För en mer detaljerad beskrivning kan man bläddra till Figur 15 som ligger i kapitel **6.9 Tidsdiagram för kommunikationsprotokoll**. Pulsen genereras med hjälp av den interna 50 MHz kristall klock-oscillatorn. **CLK** skapas med en process där en signal (**CLK**) flanktriggas efter 5208 pulser från den interna kristall klock-oscillatorn. Det skapar en **CLK** med $50 \text{ MHz}/5208=9600 \text{ Hz}$ vilket matchar standard baud rate för RS232 protokollet för att möjliggöra kommunikation mellan datorn och FPGA kortet. VHDL koden för just den här processen finns i Figur 29 i bilagor. Den här processen är även verifierad i hårdvaran med hjälp av ett oscilloskop.

Den andra processen genererar pulsen som är kopplad till **CONV** porten i ADS-7818. Den här pulsen styr konverteringsprocessen tillsammans med **CLK** pulsen. Genom att dra upp **CONV** pulsen vid tolfte **CLK** pulsen, sedan dra ner den en gång till vid trettonde **CLK** pulsen när LSB håller på att överföras, är det möjligt att upprepa dataöverföring med omvänd sekvens (från LSB till MSB) från puls 13 till puls 24. För mer detaljerad information om hur **Conv** pulsen funkar kan man bläddra till Figur 15 som ligger i kapitel **6.9 Tidsdiagram för kommunikationsprotokoll**. **CONV** pulsen som mättes i oscilloskop finns att se i Figur 30 i bilagor.

Den tredje processen läser signalen från den sjuttonde till tjugofjärde **CLK** pulsen. Dvs de 8 mest signifikanta bitarna tas fram. Sedan lägger processen till en start och stop bit i början och slutet av signalerna enligt RS232 standarden. För mer detaljerad information om hur **RS232** fungerar kan man bläddra tillbaka till kapitel **3.4 RS232 protokoll**. Signalerna ska tillslut passera från **Uart** till USB kabeln för att överföras till LabVIEW som är installerat i datorn.

6.11 Verifiering för VHDL och FPGA

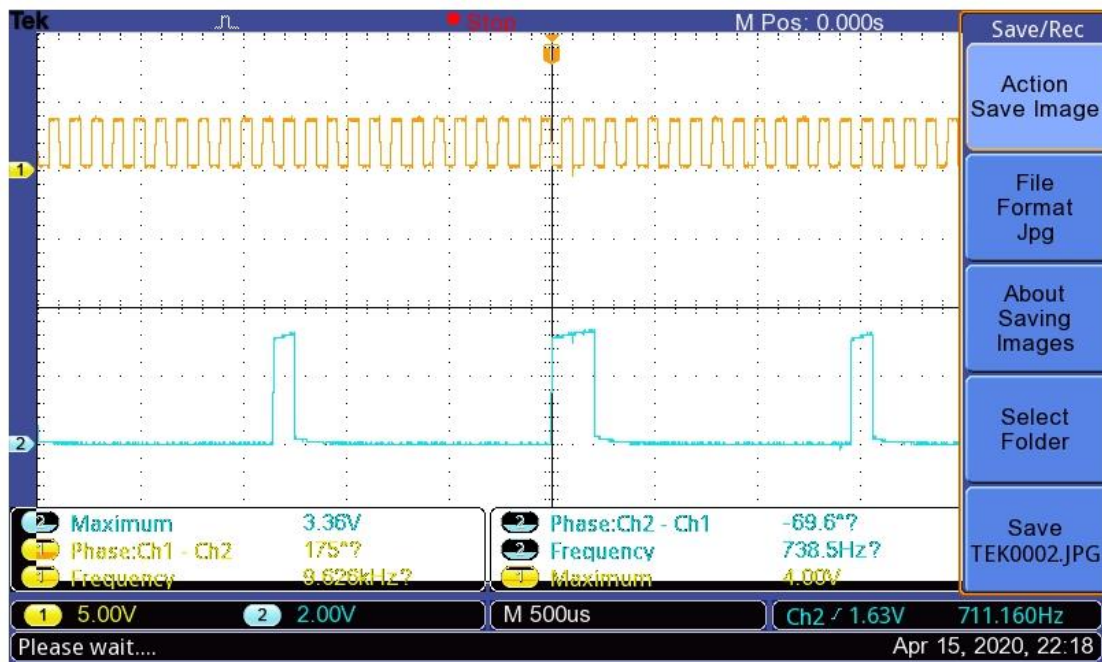
En kritisk punkt vid överföring av analoga till digitala signaler är matchning. Matchning görs mellan signalerna för **CLK** och **CONV**, vilket görs i VHDL kod för att sedan skickas från DE_1 kortet in i ADCn.

Vid design av VHDL kod är det viktigt att matcha transmissionsledningen vilket är en RS232 vilket har en baud rate på 9600 Hz. Det betyder att seriella klockan (gula grafen) måste ha samma frekvens för att systemet ska vara i fas med ledningen. Enligt (Figur 16) är frekvensen vid test på fysisk krets nära 9.6kHz vilket matchar RS232 standarden.

CONV (blå grafen) i (Figur 16) används för kontroll av framför allt sample/hold funktionen hos ADS-7818. Den breda stapeln betyder att signalen ska samplas och när signalen blir låg

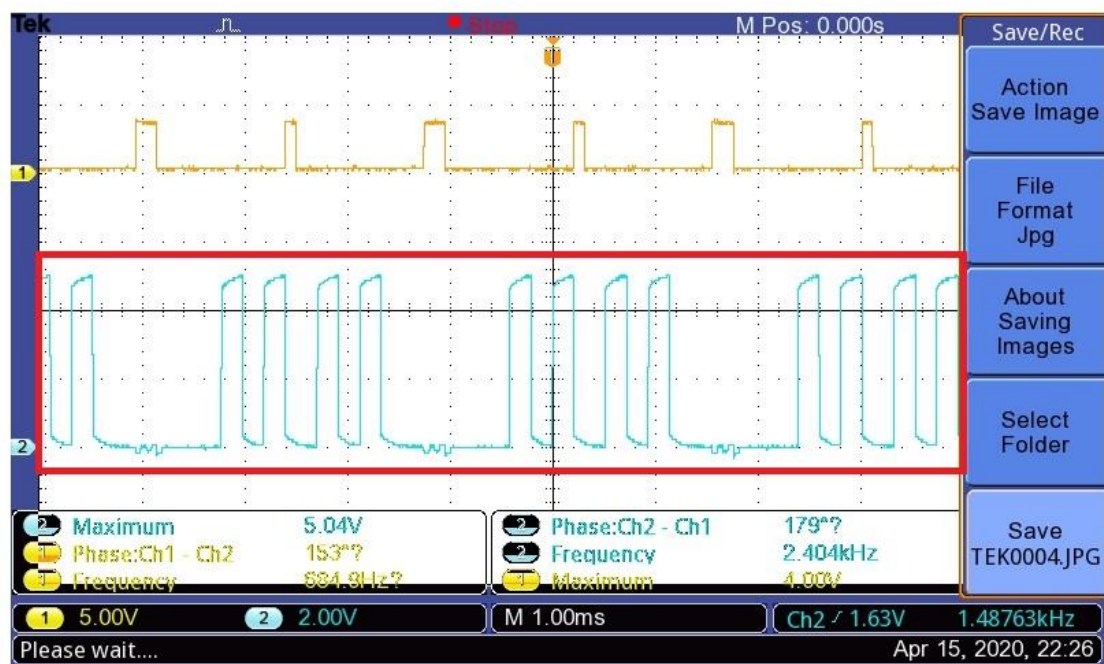
så går kretsen över till hold för att skicka över data. Den smalare stapeln används för att ändra från standarden att data skickas med MSB först till att skicka LSB först.

Sambandet mellan **CLK** och **CONV** är viktig på många sätt. En av de viktigaste funktionerna som sambandet utgör är hur systemet ändrar från att skicka MSB först till LSB först. Sample/Hold funktionen går till hold när den tjocka pelaren på **CONV** blir låg. **CLK** pulsen efter att Sample/hold funktionen har blivit låg är första pulsen för **CLK** för den här funktionen. För att ändra från att skicka MSB först till LSB först måste **CONV** bli hög igen innan slutet av **CLKs** 12e puls och sedan gå ner under **CLKs** 13e puls.



Figur 16 (gul för seriella klockan och blå för **CONV**)

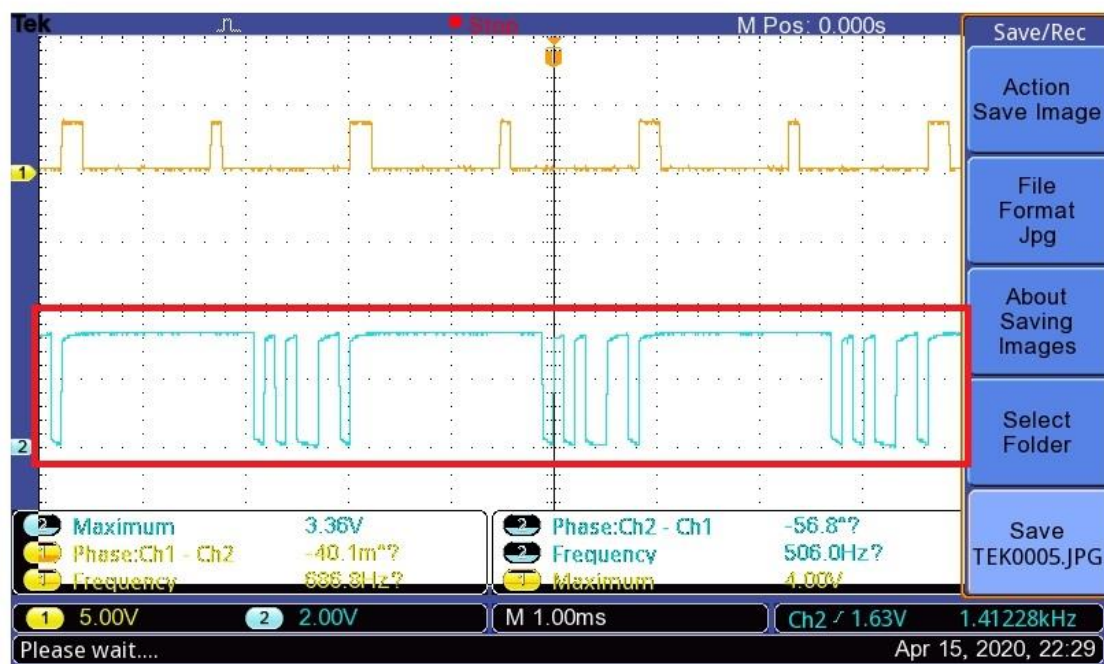
Det som är att lägga märke till i (Figur 17A) är hur den överförda datan (rödmarkerat) ser ut före och efter den smala pulsen på **CONV** signalen. Datatalet som visas i den blå grafen före slutet på den smala pelaren i den gula grafen är den samma efter den smala pelaren fast med omvänd bitsekvens. Talet har alltså ändrat ordning så MSB som var först har nu hamnat sist och LSB ligger först istället. Nu är alltså LSB först och signalen skickas in i FPGA kortet för att enbart skicka vidare signalen med LSB först.



Figur 17A (gul är **CONV** och blå är skickad data från ADS-7818)

Signalen ut från FPGA kortet ser ut som den rödmarkerade grafen i (Figur 17B). Signalen består först av en startbit vilket behövs enligt standard för transmission över RS232 följt av den digitala signalen vilket består av åtta binära bitar följt av en stoppbit.

Den blå grafen i (Figur 17B) är den samma som i (Figur 34) i bilagor bara att de höga signalerna har blivit låga och de låga har blivit höga. Det beror på att RS232 länken omvänder polariteten på signalen.



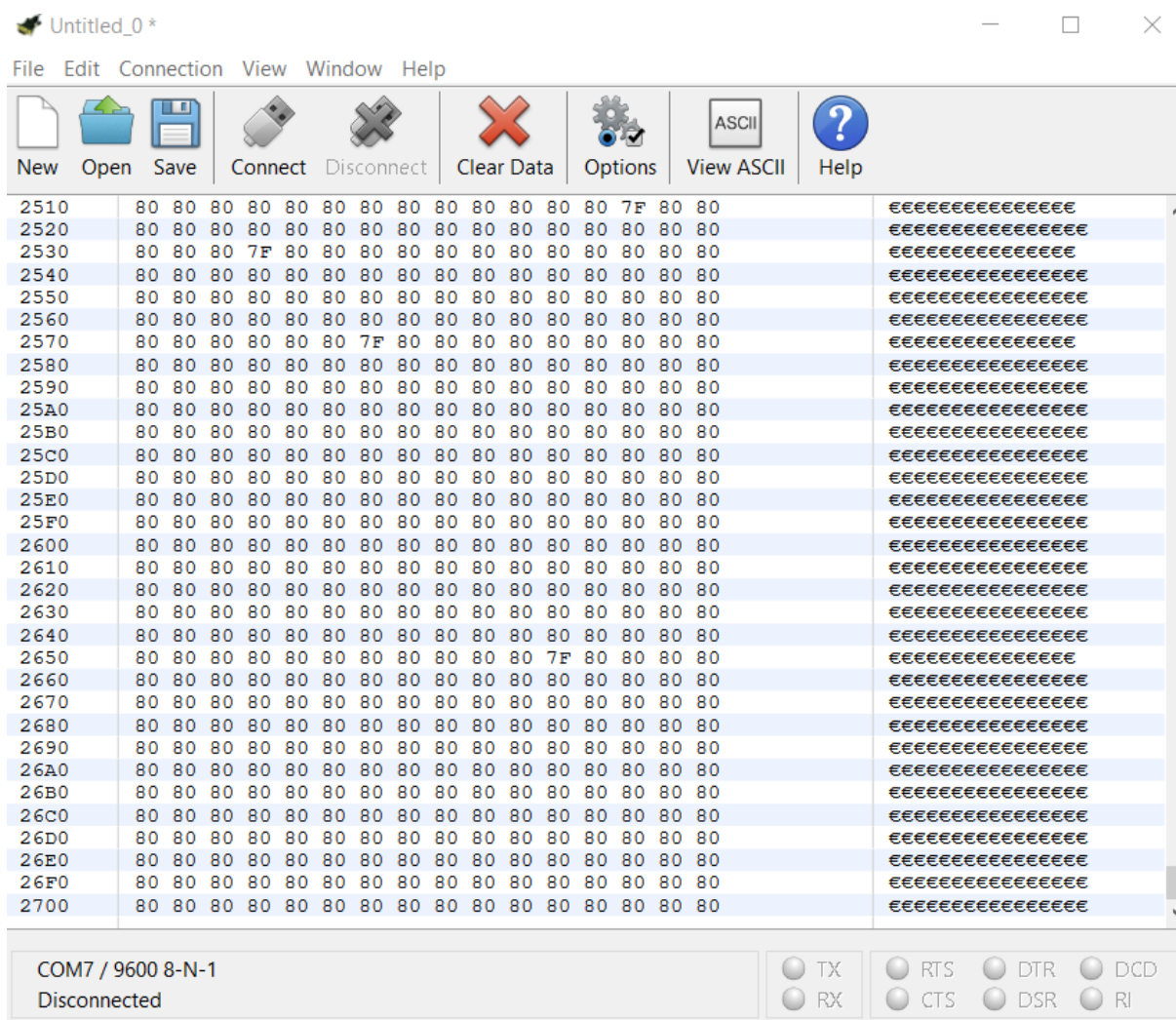
Figur 17B (Den blå grafen är den digitala signalen efter bearbetning i FPGA kortet och den gula signalen är **CONV**)

6.12 Verifiering för seriell kommunikation mellan DE_1 och dator

Det finns behov att verifiera om kommunikationen (vilket är baserat på RS232 protokoll) mellan DE_1 kortet och datorn fungerar på ett lämpligt sätt. CoolTerm är en enkel mjukvara som kan läsa ut signaler från datorns seriella port som är kopplad till FPGA kortet. De utlästa signalerna ska visas i form av hexadecimala tal. Det öppnar möjligheten att verifiera kommunikationen genom att mata in olika likspänningsnivåer som representerar olika digitala ord till ADS-7818. Sedan ska FPGA kortet hantera signalerna och packa ihop de 8 mest signifikanta bitarna i en form som är baserad på RS232 protokollet innan det skickas till datorn (via **Uart** till USB kabeln).

Om allt fungerar korrekt, ska Coolterm visa det digitala ord som matchar DC spänningen som skickades in.

Figur 18 visar hur CoolTerm tolkar insignalen från seriella porten vid inmatning av en likspänningsnivå på 2.5 V. OBS att 2.5 V motsvarar digitala ordet 80 (hexadecimal) vid användning av full scale range mellan 0 till 5V för ADS-7818 (vilket stämmer överens med konfigurationen för **Vref** pinnen av ADS-7818 i projektet). Tabell 2 visar mättningsresultat som visas i CoolTerm när olika likspänningsnivåer matas in som motsvarar olika digitala ord.



Figur 18 (CoolTerm verifieringsexempel)

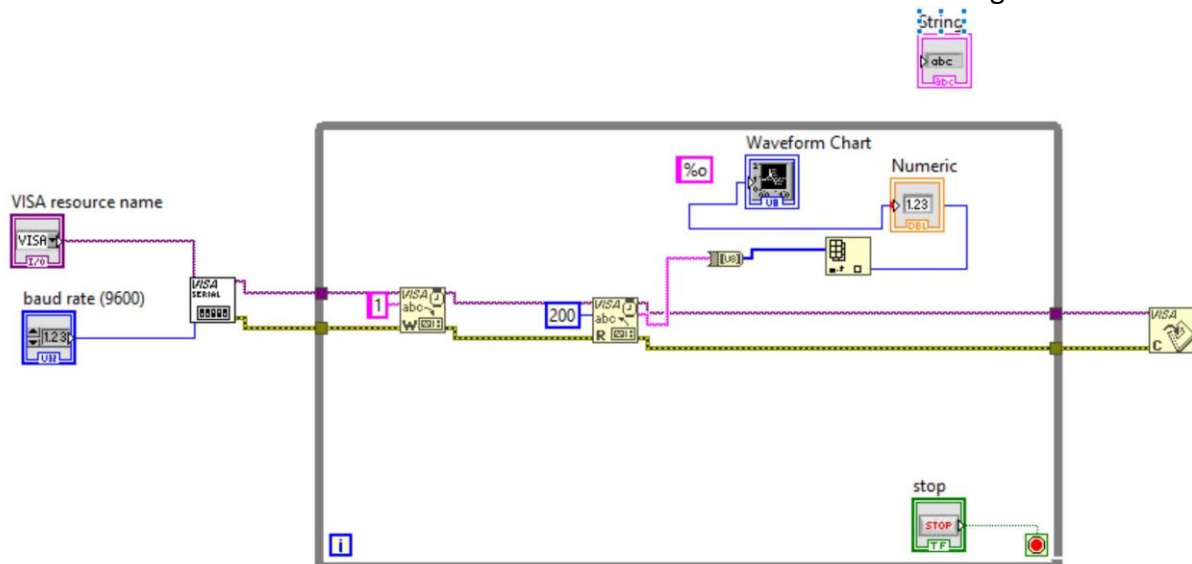
In spänning	uppmätt digital ord	teoretiskt digital ord
1V	33 (hexadecimal)	33 (hexadecimal)
2V	66 (hexadecimal)	66 (hexadecimal)
3V	99 (hexadecimal)	99 (hexadecimal)
4V	CD (hexadecimal)	CC (hexadecimal)
5V	FF (hexadecimal)	FF (hexadecimal)

Tabell 2 (tabellen visar mättnings resultat i CoolTerm)

Det går att konstatera att seriell kommunikation mellan FPGA kortet och datorn fungerar eftersom den inmatade likspänningsnivån matchar de digitala ord som matas ut av CoolTerm.

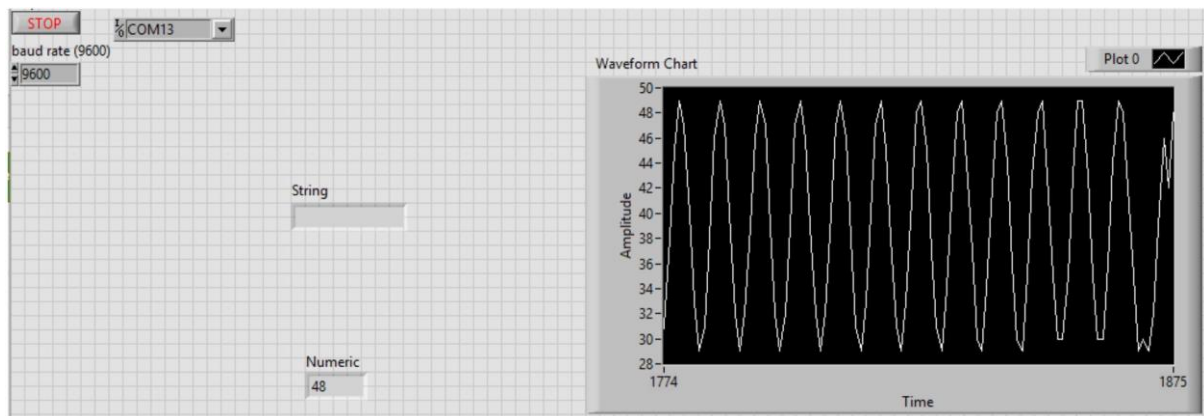
6.13 LabVIEW

Efter RS232 porten kopplas digitala signalen in i datorn och bearbetas med hjälp av LabVIEW. Enligt LabVIEW koden i (Figur 19) så skickas signalen in via en VISA som även behöver matas med standard baud rate för den använda RS232 länken, vilket i det här fallet är 9600 Hz. Efter VISA terminalen skickas signalen in i en "while loop" för att signalen alltid ska uppdateras. I while loopen skickas signalen först till en "VISA Write Function" vilket behövs för att signalen ska ändras till rätt format för att sedan läggas in i en "VISA Read Function" vilket skriver ut en "String" (rosa linjen) med värdet på signalen med 200 bytes noggrannhet. Från "VISA Read Function" skickas även VISA signalen vidare när while loopen stoppas till en "VISA Close Function" vilket behövs för att slutföra sessionen. Strängen från "VISA Read Function" skickas sen till ett "String To Byte Array" block där varje string karaktär har ett ASCII värde på 8 bitar. De här bitarna är signalens ursprungliga värde i en form som kan läsas av en grafitare. Alla karaktärerna placeras i var sitt element i en array som sedan skickas vidare till en "Array element" vilket plockar ut karaktärernas värde från ett element i taget. varje värde skickas sedan vidare som en integer till en "Numeric" vilket visar elementets värde i siffror som sedan skickas till en Waveform Chart som ritar grafen.



Figur 19 (LabVIEW program för grafitning)

Figur 20 visar grafen som är plottad i LabVIEW när en sinusvåg matas in till ADS-7818. För att göra det möjligt för kommunikation mellan RS232 länken och "DE_1" kortet behövs en portdefiniering vilket är COM13 i det här fallet. "Numeric" visar värdet på varje karaktär i stringen från "VISA Read". "Waveform chart" visar grafen som ritar signalen som skickas in via RS232 länken in i datorn.



Figur 20 (när en sinusvåg matas in till A/D-omvandlare)

6.14 PCB kretsschema (skillnaden mellan PCB schema och den framtagna kretsen)

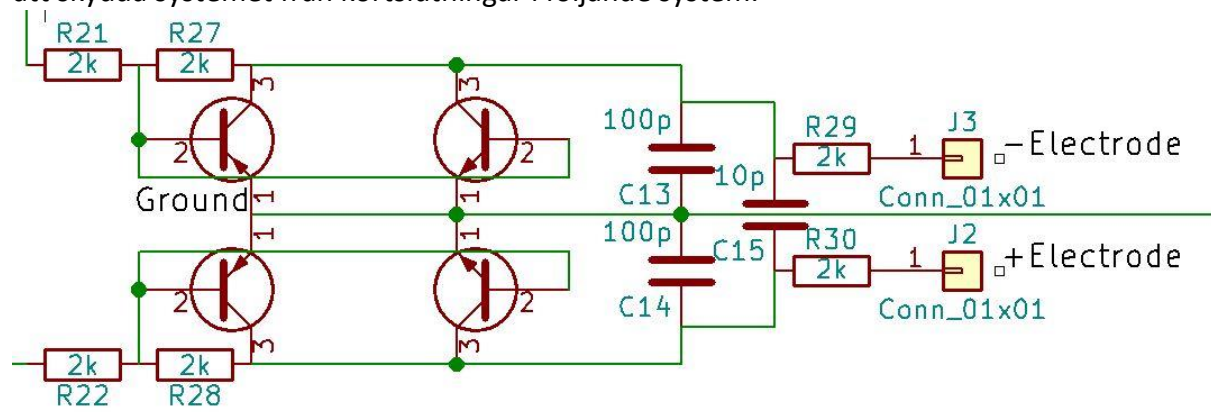
I projektet fanns det ingen beställd och testad krets för PCBn då mjukvaruproblem (Kicad krascha vid generering av footprints för layouten) gjorde att det inte gick att beställa en PCB. Det har därför inte varit möjligt att testa PCB schemat i verkligheten. Ändringar som gjordes vid konstruktion av PCB jämfört med den första kretsen var följande:

- Säkerhetskrets har lagts till i början av kretsen för att skydda kretsen och användaren vid parallell användning av ytterligare elektroniska verktyg på kroppen.
- En Right Leg Drive krets har lagts till för förbättring av CMRR.
- Kanalfiltret har tagits bort och kanalerna implementeras nu genom att byta ut komponenterna för bassignalsfiltret. Det förbättrar bruset då bassignalsfiltret inte gjorde något när ett kanalfilter var inkopplat. Det går att löda på kontakter på PCBn vilket gör att det är möjligt att välja vilken signal som önskas av alpha, beta eller bassignal genom att fritt byta komponenterna till önskat filter.
- Det extra förstärkningssteget har tagits bort då det är möjligt att ändra amplitudaxeln digitalt i LabVIEW. Upplösningen på amplitudaxeln blir lite sämre med metoden. Det är även möjligt att öka förstärkningen i instrumentförstärkaren.

Right Leg Drive kretsen finns det många varianter av. Kretsen som används i projektet är grundläggande då den inte använder kondensatorer för frekvensfaktorer. Designen för Right Leg Drive kretsen är tagen från den använda instrumentförstärkarens (ADS-7818) datablad(4).

Säkerhetskretsen har grundläggande kondensatorsäkringar för EEG signaler. Kretsen har även en transistorkoppling (Figur 40) vilket leder ström ut från kretsen om transistorernas tröskelspänning överstigs. Om för höga spänningar inträffar i kretsen kommer därför kretsen inte bli skadad då strömmarna leds ut via transistorerna till jord. Problemet går även att lösa genom att koppla Zenerdioder ut ifrån kretsen. Problemet med den lösningen är att skillnaden mellan tröskelspänningen och rimlig EEG spänning är för stor och det medför en risk för kretsen att skadas, då kretsens förstärkning är för stor för användning av

Zenerdioder. En "isolation amplifier" ska även kopplas vid slutet av den analoga kretsen för att skydda systemet från kortslutningar i följande system.



Figur 40 (säkerhetskretsen för PCB-design)

7 Resultat

7.1 Mål som har uppnåtts

1. Alla filter i hårdvarukonstruktionen fungerade som förväntat. Högpas och lågpasfilter i första steget filtrerade fram EEGns bassignal. Bandspärrfiltret dämpade störningar från elnätet. Bandpassfiltret tog fram olika EEG kanaler (Alpha och Betakanaler). Instrumentförstärkaren fungerade korrekt. Tillsammans med det extra förstärkningssteget tog den fram en läsbar EEG signal.
2. A/D-omvandlaren (ADS-7818) konverterade de samplade EEG signalerna till digitala signaler korrekt. Digitala signalerna som matades ut var ordnade från LSB till MSB. Vilket var anpassat till RS232 standard.
3. FPGA kortet DE_1 styrde ADS-7818 med hjälp av 2 pulser (**CLK** och **CONV**) som genererades med hjälp av VHDL kod. DE_1 förberedde även de digitala signalerna för överföring med hjälp av RS232 standarden med en baud rate som var 9600 Hz. Dvs, kommunikationen mellan datorn (LabVIEW) och FPGA kortet fungerade.
4. LabVIEW tog emot signalerna från DE_1 kortet och plottade de samplade EEG signalerna i en graf.
5. Konstruktionen fungerade inom ramarna för ett realtidsbaserat system. FPGA kortet och VHDL drivrutiner designades så den förväntade responsen förvärvades vid varje klockpuls.

7.2 Mål som inte uppnåtts

1. PCB-layouten är inte testad i verkligheten. Men ett kretsschema för PCB-designen togs fram. Den finns att se i Figur 37 i bilagor.

7.3 Mätningresultat i form av grafer från LabVIEW

Graferna som visas i det här kapitlet kommer från EEG tester av systemet som utfördes på olika personer.

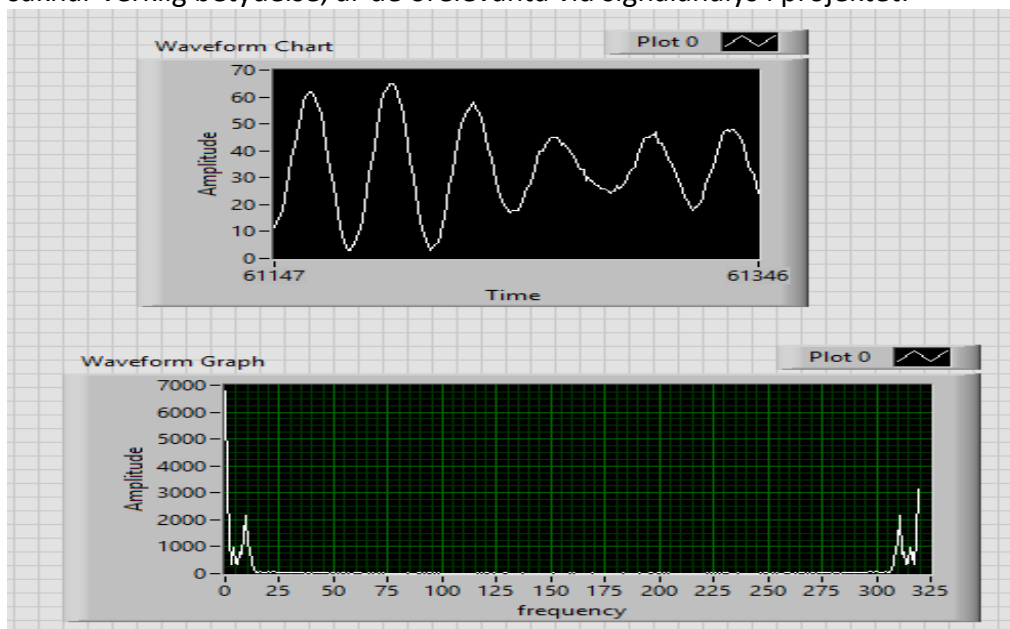
Det finns två stycken EEG kanaler (Alpha och Beta) som har realiserats i projektet. Alphavågen hade ett passband som gick mellan 8 och 12 Hz. Beta vågen hade ett passband som gick mellan 12 till 30 Hz. I det här kapitlet presenteras EEG vågorna som plottades i LabVIEW tillsammans med FFT analysen.

Figur 21 visar alphavågen vid användning av kanalfiltret som designades för alphavågen. Filtret fick alltså ett passband som låg mellan 8 till 12 Hz. Den plottade vågformen i waveform-chart fick ett symmetriskt utseende jämfört med befintlig data för en standard alphavåg. Därför konstaterades att EEG mätningssystemet fungerade korrekt. I waveform graph syns FFT analysen för den uppmätta alphavågen. Där kan amplituder för olika frekvenskomponenter analyseras. För mer detaljerad information om hur fourieranalys och FFT fungerar matematiskt finns ytterligare information i kapitel **2.3 fourieranalys och FFT**.

I frekvenskomponenten 0 Hz i FFTn fanns det en amplitudnivå. Det var alltså en DC spänning vilket motsvarade DC offseten som genererades i offsetsteget. För mer detaljerad information om hur offset-steget fungerar finns ytterligare information i kapitel **5.5 offset steg**.

Ett passband som gick mellan 8 till 12 Hz syntes tydligt i frekvensspektrat som visas av FFT waveform-graph. Dvs amplituder för frekvenskomponenter som hamnar utanför passbandet var dämpade. Det stämde överens med det teoretiska beteendet för kanalfiltret (för alphakanalen). För mer detaljerad information om hur kanalfilter fungerar finns ytterligare information i kapitel **5.4 kanalfilter**.

På högersidan av FFT waveform-graph (frekvenskomponenter från 300 Hz till 325) fanns amplitudkomponenter för spegelfrekvenser. Observera att amplituder för spegelfrekvenserna inte fanns i den verkliga signalen. Det är bara nånting som är matematiskt oundvikligt vid användning av fourieranalys och FFT. Pga att spegelfrekvenser saknar verklig betydelse, är de orelevanta vid signalanalys i projektet.



Figur 21 (Figuren visar plottad alphavåg i LabVIEW tillsammans med FFT analys)

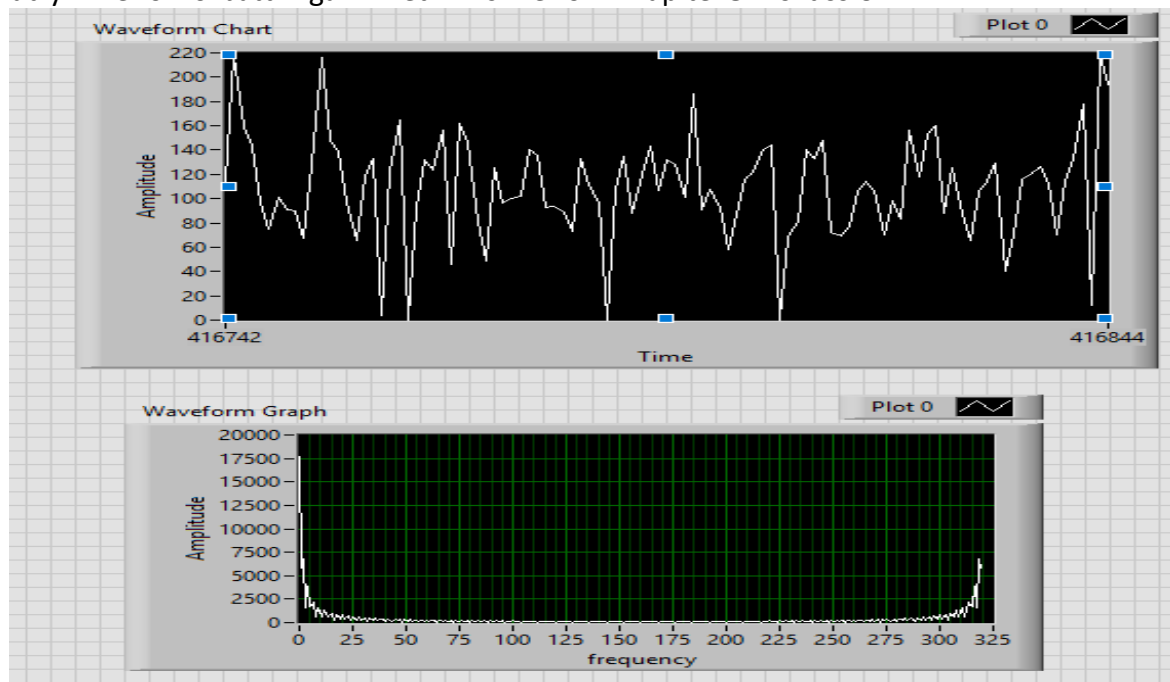
Figur 22 visar betavågen vid användning av kanalfiltret som designades för betavågen. Filtret hade ett passband som låg mellan 12 till 30 Hz. Den plottade vågformen i waveform-chart

fick ett symmetriskt utseende jämfört med befintlig data för en standard betavåg. I waveform graph syns FFT analysen för den uppmätta beta vågen.

I frekvenskomponenten 0 Hz i FFTn fanns det en amplitudnivå som kommer från offset steget.

Ett passband som gick mellan 12 till 30 Hz syntes tydligt i frekvensspektrat som visas av FFT waveform-graph (passbandet syntes inte lika bra som alphavågen, det kan beror på att amplitudkomponenterna är mer utspädda i olika frekvenskomponenter inom 12 till 30 Hz). Spegelfrekvenser som saknade verklig betydelse syntes även här mellan 270 till 325 Hz.

Mätsystemet tog fram signaler från elektroder till LabVIEW på ett lämpligt sätt. Kretsen filtrerade och förstärkte signalen korrekt. Efter offsetsteget passerade signaler till A/D-omvandlaren som var kopplad till FPGA kortet. Till slut packade FPGA kortet DE_1 de samplade signalerna till datapaket som baserades på RS232 standarden och skickades vidare till datorn. Där plottades signalen i både tid och frekvensspektrum. Utifrån resultaten går det att konstatera att EEG front to end systemet fungerar enligt förväntan. Men det finns utrymme för förbättringar vilket finns mer om i Kapitel **8 Diskussion**.



Figur 22 (Figuren visar plottad Betavåg i LabVIEW tillsammans med FFT analys)

8 Diskussion

8.1 Svar på frågeställningen

1. Finns det möjlighet att förbättra mättningsresultat av EEG signal som skickas från A/D-omvandlaren till LabVIEW?

Det finns möjlighet att öka upplösning genom att optimera mjukvaran för att göra det möjligt att utnyttja fler bitar från ADS-7818ns potential. Att byta ADS-7818 mot en mer avancerad A/D-omvandlare är också en bra lösning. Mer detaljerad information finns i kapitel **8.2 diskussion om förbättrings alternativen för upplösning av A/D-omvandlare**.

2. Hur fungerar analys av frekvensspektrum för signaler?

Analys av frekvensspektrum är baserat på FFT (fast fourier transform) i LabVIEW. Mer detaljerad information om hur fouriertransform fungerar finns i kapitel **3.3 fourieranalys och FFT**.

3. Hur många EEG kanaler kan byggas upp med hårdvarukonstruktionen?

Befintliga kanaler som brukar användas vid mätning av EEG är gamma, beta, alpha, theta och delta. Alla dessa kanalerna går att bygga genom att ändra komponenterna för kanalfiltret för att ändra passbandet. Det går inte att läsa alla kanalerna parallellt då det bara finns en inport i ADS-7818.

4. Finns det möjlighet att lägga till fler EEG kanaler?

Det finns möjlighet att konstruera mer kanaler genom att göra en PCB-layout som kan ta emot mer komponenter i sitt begränsade utrymme jämfört med kopplingsplattan. Det finns även möjlighet att läsa flera kanaler samtidigt genom att byta till en A/D-omvandlare med fler inportar. Det finns mer information om EEG kanaler i kapitlen **8.4 Diskussion om EEG kanaler**.

8.2 Diskussion om förbättringsalternativ för upplösning av A/D-omvandlare

Upplösningen för EEG signaler som plottas i LabVIEW har 8 bitars upplösning. Den är begränsad av RS232 standard eftersom det är ett protokoll som är designat för maximalt 8 bitars dataöverföring (det blir 10 bitar inklusive start och stopbit som finns i början och slutet på varje datapaket). Upplösningen för ADS-7818 är egentligen 12 bitar, det innebär att projektet inte riktigt har utnyttjat 100 procent av potentialen för ADS-7818. Det är ett problem som kan lösas med en mer optimerad mjukvarulösning (VHDL kod).

Exempelvis är det möjligt att skriva en VHDL process som delar upp de samplade signalerna (med 12 bitars upplösning) i två RS232 datapaket. Varje datapaket kan bära 6 bitar data tillsammans med 2 onödiga bitar (start och stopbit ska också läggas till). Sedan skickas datan

vidare till LabVIEW, till slut kan LabVIEW packa ihop datapaket och plotta i grafen med 12 bitars upplösning. Det finns faktiskt ett starkt incitament att pröva på just den här metoden som utnyttjar 100 procent av prestandan som ADS-7818 har. Men det fanns tyvärr inte tillräckligt med tid för design och verifiering av mjukvaror som driver just den här metoden. Hela VHDL koden måste skrivas och testas om för att det ska fungera.

Designsvårighet för själva LabVIEW funktionsblock ökar också exponentiellt. Anledning är att LabVIEW måste hantera två datapaket på rad för varje sampling. Där måste 2 onödiga databitar tas bort för att även slå ihop datapaket till ett 12 bitars digitalt ord.

Vid önskan att öka upplösning ytterligare ett steg, måste en mer avancerad och kraftfull A/D-omvandlare användas. VHDL är ett lågnivå programmeringsspråk som är hårdvarunära. En mer avancerad A/D-omvandlare brukar kräva mycket mer komplicerade mjukvarudrivrutiner som är oerhört svåra att designa med VHDL kod. Det innebär att det är fördelaktigt att byta till ett programmeringsspråk med högre nivå (lättare för programmeraren att förstå) som är arduino [15] baserat. Där färdiga bibliotek kan användas för att styra A/D-omvandlaren istället för att skriva egna drivrutiner. Men den här lösningen innebär att allt behöver byggas om från noll, på grund av att nuvarande lösning är fullständigt baserad på FPGA och VHDL. Det finns tyvärr inte tillräckligt tid för konstruktionen.

8.3 Diskussion om verifiering och felsökning

Verifiering av olika blocksystem är en viktig del för projektet. För just hårdvaror är felsökning och verifiering inte så komplicerat. På grund av att de flesta delar av hårdvarukonstruktionen är simulerade i förväg, går det att identifiera fel och problem genom att implementera mätning i olika delar av hårdvaran (från elektroder till A/D-omvandlare).

Men verifiering för mjukvaror (VHDL kod som styr A/D-omvandlaren) och seriell kommunikation är betydligt mer tidskrävande.

Verifiering av VHDL koden kräver förståelse inom både mjukvaru och hårdvarudesign. Det finns ett starkt behov att verifiera att alla pulser som går mellan FPGA kortet och ADS-7818 beter sig lämpligt (**CONV**, **CLK** och **Data** pulserna). Den absolut svåraste delen är att felsöka och verifiera **Data** pulsen. På grund av att det är LSB-first mode som implementerades i projektet, finns det ett skarpt krav att **CONV** pulsen måstes dras upp eller ner i en viss timing för att omvända sekvensen av **Data** pulsen från MSB först till LSB först. För mer detaljerad information om hur LSB-first mode fungerar finns ytterligare information i kapitel **6.9 tidsdiagram och kommunikationsprotokoll**.

Verifiering för seriell kommunikation som går mellan datorn och FPGA kortet testades med hjälp av LabVIEW från början. Genom att mata in olika likspänningsnivåer är det möjligt att verifiera om utsignalen matchar de teoretiskt förväntade digitala orden eller inte. En mer noggrann beskrivning för den här verifieringsmetoden finns i kapitel **6.12 Verifiering för seriell kommunikation mellan DE_1 och dator**. Men det gick inte så bra på grund av att LabVIEW tolkar indata som datatypen char som default inställning. Därför måste

datatypkonvertering (från char till int) implementeras för att kunna tolka datan i form av decimaler. Det gör att det är en väldigt svår process som används för att använda LabVIEW för att verifiera. Därför användes CoolTerm i projektet för verifiering (där kan datan tolkas direkt i form av hexadecimala tal).

8.4 Diskussion om EEG kanaler

I konstruktionen finns det bara två EEG kanaler som är designade (Alpha och Betakanaler). Det finns möjlighet att skapa mer kanaler genom att designa fler bandpassfilter som är designade efter de andra kanalerna.

Det finns en nackdel med nuvarande design, det är bara möjligt att visa en EEG kanal i taget. Dvs det går inte att visa flera kanaler samtidigt i form av grafer. Det kan vara nödvändigt att visa både alpha och betakanalen samtidigt för att se om patienten är aktivt tänkande eller i mentalt viloläge.

Det finns möjlighet att skapa ett EEG mätsystem som kan hantera mer kanaler på en gång genom att lägga till en spegelbild av nuvarande hårdvarukonstruktion och koppla den till ett annat kanalfilter. Men det kommer ta oerhört mycket plats och är inte riktigt någonting som är implementerbart på kopplingsplattor som har begränsat utrymme. Problemet kan möjligtvis lösas genom att göra en PCB layout som kan inkludera mer komponenter.

Det finns också en till alternativ lösning som är baserad på transistor switch. Genom att koppla transistor switch till GPIO pinnar av FPGA kortet kan olika kanaler justeras i grafer som är plottade i LabVIEW. Pinnarna som är kopplade till transistorerna konfigureras som utportar och styrs direkt av VHDL kod. Den här lösningen kräver inte många extra komponenter, därför är det genomförbart i testkretsen som är konstruerad på kopplingsplattan (om det fanns mer tid för projektet).

8.5 Problem som tog tid att lösa under projektets gång

1. Det tog oförväntat lång tid att sätta igång kommunikationen mellan ADS-7818 och FPGA kortet. Alla pulser måste fungera perfekt för att A/D-omvandling ska fungera på LSB-first mode. Felsökning måste göras stegvis med hjälp av mätningssinstrument (oscilloskop).
2. Att felsöka kommunikation mellan datorn och FPGA kortet, på grund av ett problem som orsakats av felaktig baudrate. Det rättades till genom att skriva om VHDL processen som genererade **CLK** pulsen. OBS att det är **CLK** pulsen som styr frekvenser av baud-rate.
3. LabVIEW tolkar insignalen i form av datatypen char som standard. Men för att plotta EEG signaler i grafen behövs data i form av datatypen double eller integer. Det tog

lite extra tid för att lista ut att konvertering av insignalen från char till integer kunde göras med hjälp av funktionsblocket **string to array**.

8.6 Miljö och Etik

En EEG samlar signaler som kan presentera djupt personliga och känsliga uppgifter som mentala skador och sjukdomar. Databaser för att säkra den här typen av information måste vara säkra mot cyberattacker och hacking. En portabel lättanvänd modell av den här typen av teknik kan användas på personer som inte är medvetna om vilken signifikans datan har.

Portabla EEG maskiner kan enkelt marknadsföras till privatpersoner som ett sätt att förbättra hälsan. En privatperson som inte har kunskap om hur datan tolkas kan lätt använda datan fel eller dela datan på nätet för andra att tolka den. En privatperson behöver inte nödvändigtvis respektera att datan som erhålls är känslig och ansvaret som medförs med informationen.

Vissa kritiska material och kemikalier används i stora mängder för att producera mikrochip och elektriska komponenter. Enligt **DHL** används 1.6 Kg petroleum, 32 Kg vatten, 0.7 Kg elementära gaser och 72 g kemikalier för att tillverka ett mikrochip på 2 g [18]. Alltså hade massproduktion av EEG maskiner haft en dålig påverkan på miljön. De elektriska komponenterna är bara en del av EEG maskinen. De andra delarna (exempelvis FPGA kortet) har även de sina kritiska material och krävande tillverkningsprocesser.

8.7 Slutsats

En fungerande prototyp av front to end EEG maskin med 8 bitars upplösning är konstruerad i projektet. Två EEG kanaler (alpha och beta) är inkluderat i systemet.

Men det finns fortfarande ett viktigt mål (PCB-layout) som inte är uppnått. En färdig PCB schematic är designad. Men PCB-layout togs inte fram på grund av att **Kicad** alltid kraschade vid generering av footprint. Det fanns inte tillräckligt med tid att felsöka och lösa problemet.

Det finns också stort utrymme att förbättra prestanda av EEG mätsystemet i form av bättre upplösning och flera EEG kanaler.

Projektet har uppnått funktion av en EEG maskin. Men det är inte en perfekt front to end lösning för EEG mätning. Till exempel krävs manuell omkoppling på kopplingsplattan för att ändra mellan kanalerna. Det hade eventuellt varit möjligt att lösa enligt transistorlösningen som diskuteras i **8.4 Diskussion om EEG kanaler**.

Referenslista

- [1] "12-bit highspeed low power sampling analog-to-digital converter", 2000, [Online]. Tillgänglig: <https://www.ti.com/lit/ds/sbas078/sbas078.pdf>. Hämtad: 2020-04-14
- [2] L. Bengtsson, "AD-omvandling", Elektriska mätsystem och mät metoder, Lund: Studentlitteratur AB, pp.154-155.
- [3] L. Bengtsson, "Signalanalys i frekvensrummet", Elektriska mätsystem och mät metoder, Lund: Studentlitteratur AB, pp.300-325.
- [4] "INA118 precision low power instrumentation amplifier", 2000, [Online]. Tillgänglig: www.ti.com/lit/ds/symlink/ina118.pdf?ts=1589038273544. Hämtad: 2020-04-14
- [5] Tatum, William O. (2014). *Handbook of EEG interpretation*. Demos Medical Publishing. pp. 155–190.
- [6] 2020-05-27. [Online]. Available: <https://www.electronics-tutorials.ws/filter/band-stop-filter.html>
- [7] "TL07xx Low-Noise JFET-Input Operational Amplifiers", 2017, [Online]. Tillgänglig: <http://www.ti.com/lit/ds/slos080n/slos080n.pdf?ts=1590432427454>. Hämtad: 2020-05-27
- [8] "DE1 Development and Education Board", 2012, [Online]. Tillgänglig: https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-4904342209-de1-usermanual.pdf. Hämtad: 2020-05-27
- [9] 2020. URL: <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html> (hämtad 2020-05-27)
- [10] 2020. URL: <https://www.mentor.com/products/fv/modelsim/> (hämtad 2020-05-27)
- [11] 2020. URL: <https://www.ni.com/sv-se/shop/LabVIEW.html> (hämtad 2020-05-27)
- [12] 2020. URL: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html> (hämtad 2020-05-27)
- [13] 2020. URL: <https://kicad-pcb.org/> (hämtad 2020-05-27)
- [14] S. Siuly, et al., "Generation Organism of EEG Signals in the Brain," in *EEG Signal Analysis And Classification*, Y. Zhang, Melbourne, Victoria, Australia: Springer, 2016, pp. 7-11.
- [15] 2020. URL: <https://www.arduino.cc/> (hämtad 2020-05-29)

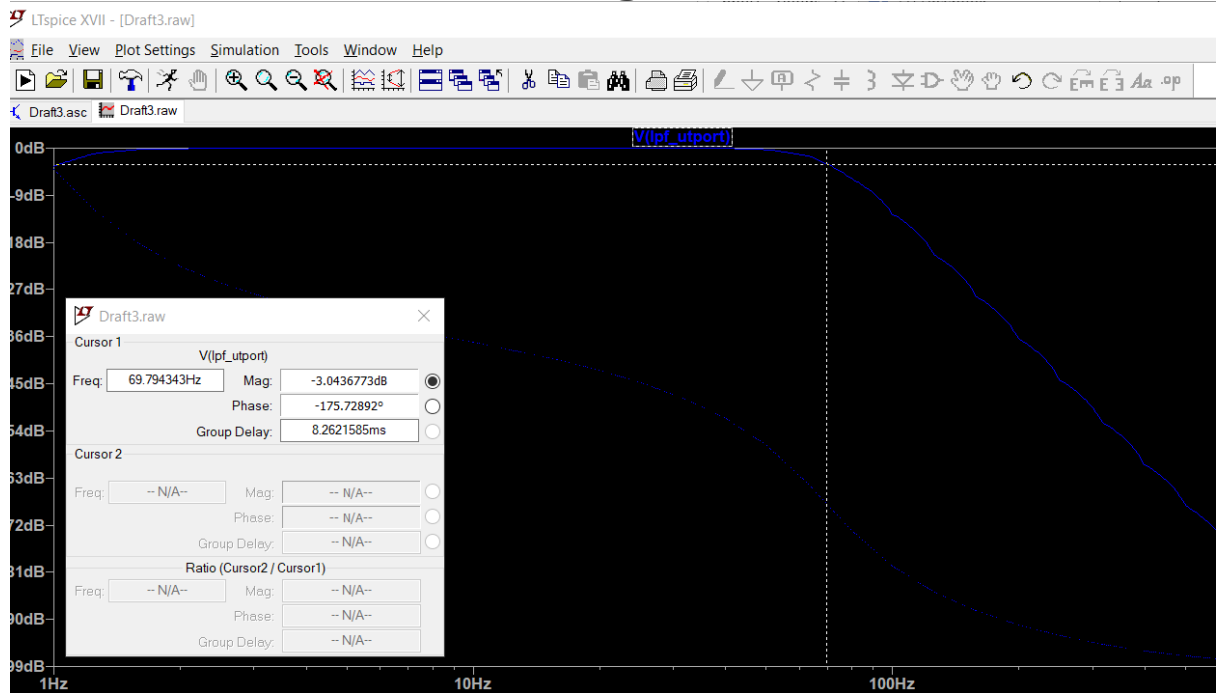
[16] 2020. URL: <https://www.electrokit.com/produkt/usb-rs232-omvandlare-for-pc-mac-linux/> (hämtad 2020-05-29)

[17] 2020. URL: <https://shop.openbci.com/products/5-mm-spike-electrode-pack-of-30?variant=8120433606670> (hämtad 2020-05-29)

[18] DHL Logistics of Things. "Addressing sustainability in the semiconductor industry", 2019. URL: https://lot.dhl.com/addressing-sustainability-in-the-semiconductor-industry/?fbclid=IwAR2K2cAuN3YheK48ZKw4veiducCh30g4ylx67TWvxqrBbp2B7DmSXM01g_U

[19] S. Siuly, et al., "What Is EEG?" in *EEG Signal Analysis And Classification*, Y. Zhang, Melbourne, Victoria, Australia: Springer, 2016, pp. 3-7.

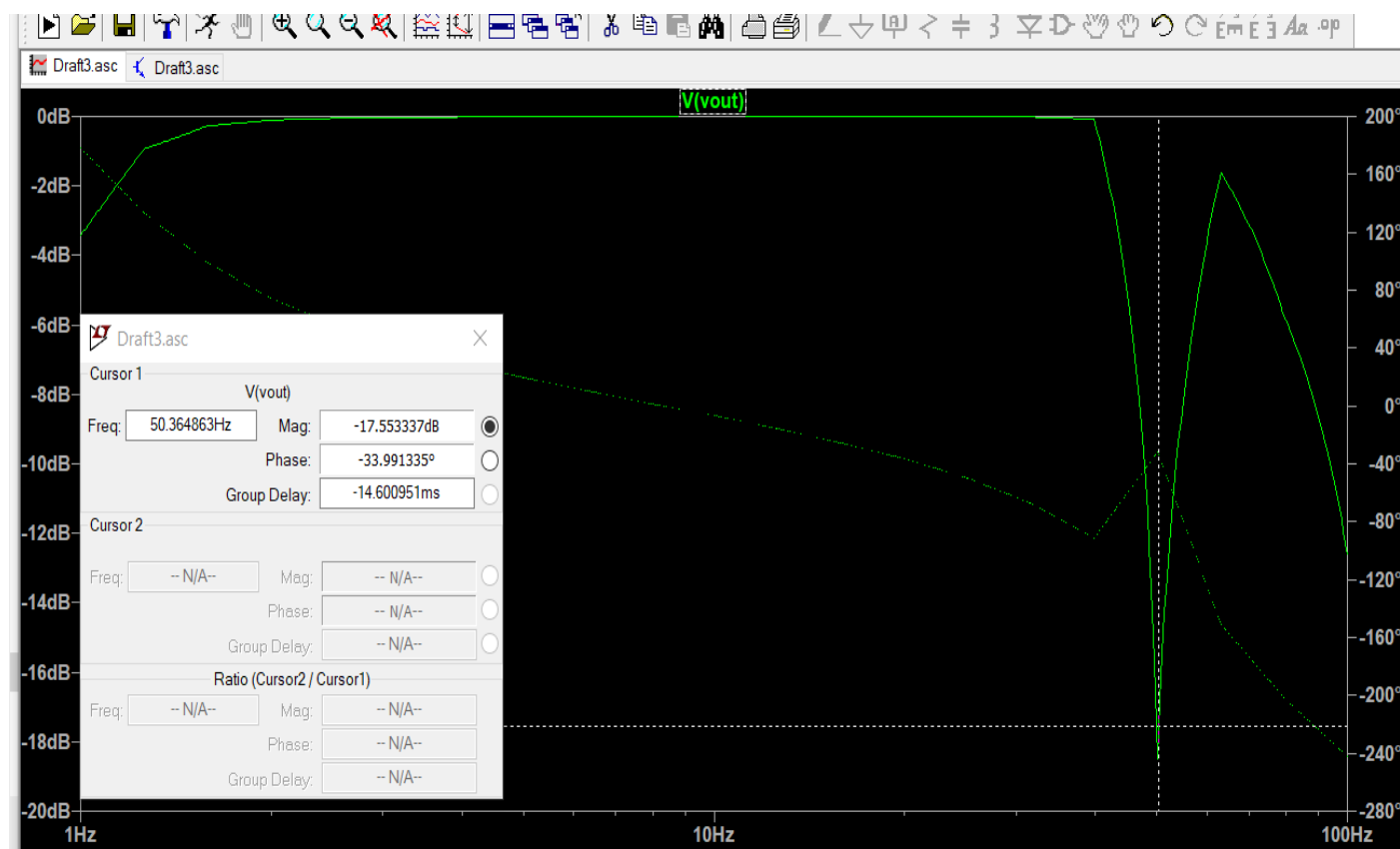
Bilagor:



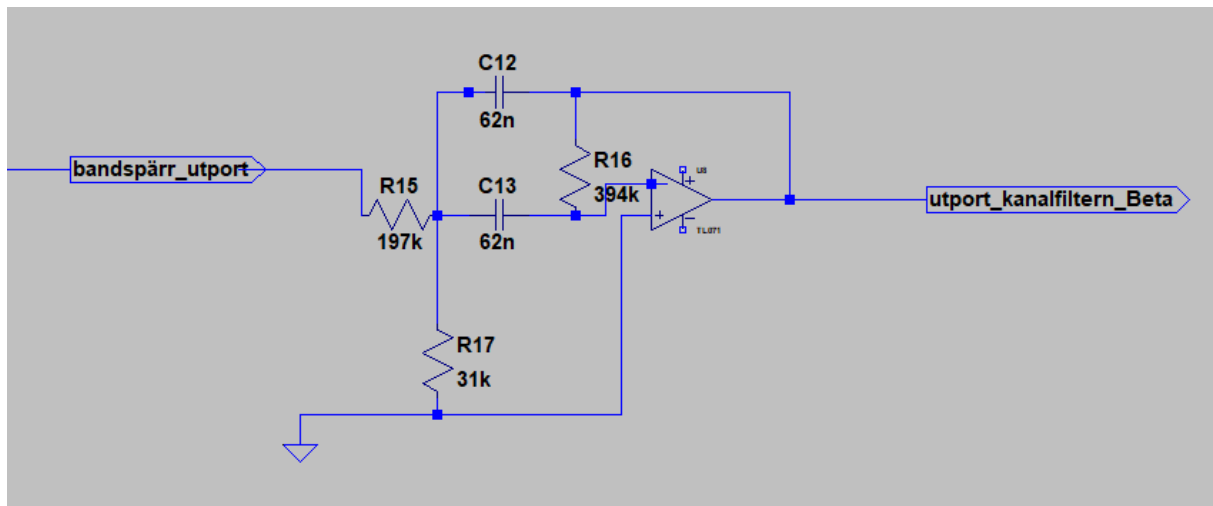
Figur 23 (småsignalsimulering för 4e ordningens lågpassfilter)

frekvens	insignal (peak to peak)	utsignal(peak to peak)
20Hz	200 mV	200 mV
40Hz	200 mV	200 mV
60Hz	200 mV	172 mV
80Hz	200 mV	110 mV
100Hz	200 mV	50 mV

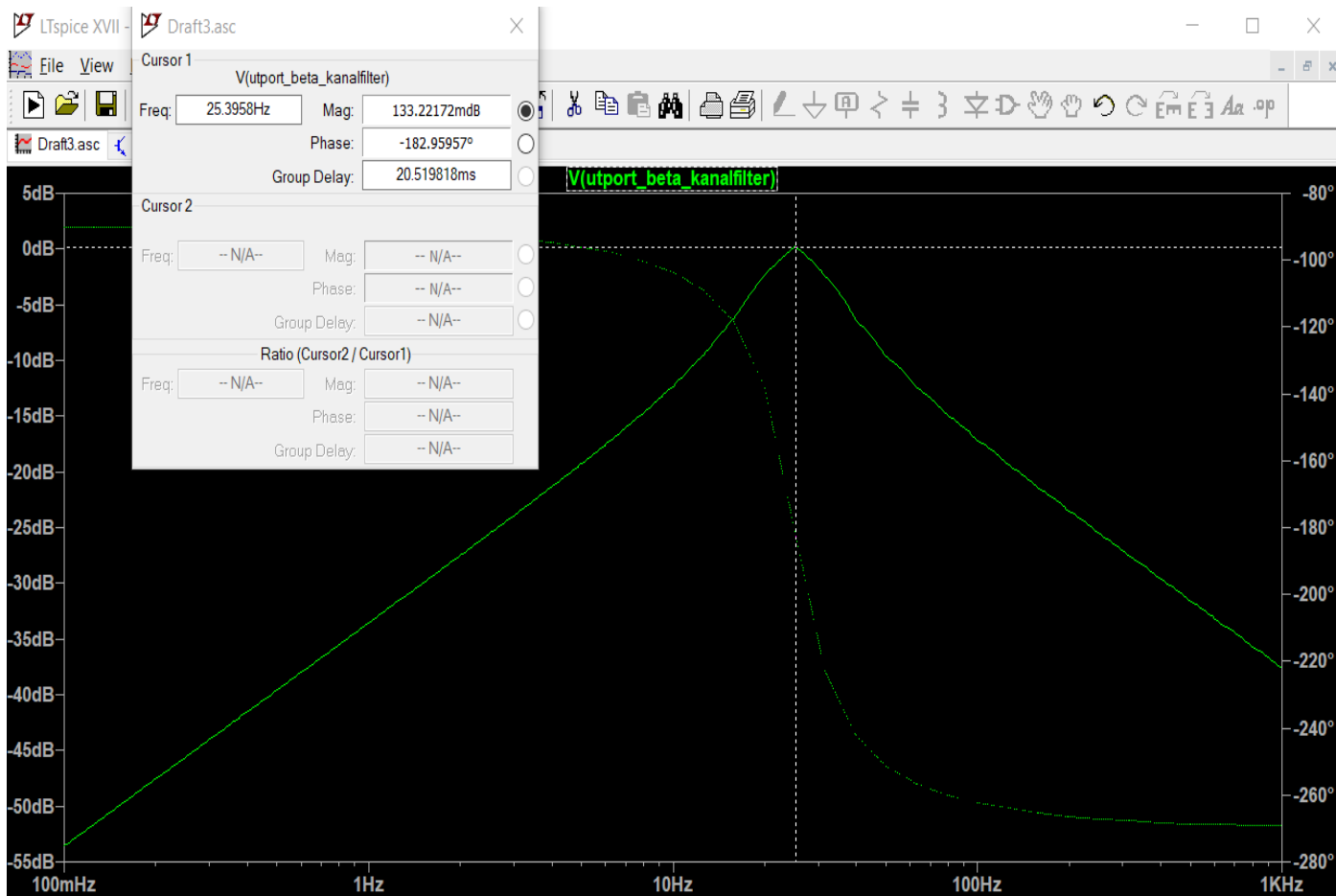
Tabell 3 (verifiering för 4e ordningens Butterworth lågpassfilter)



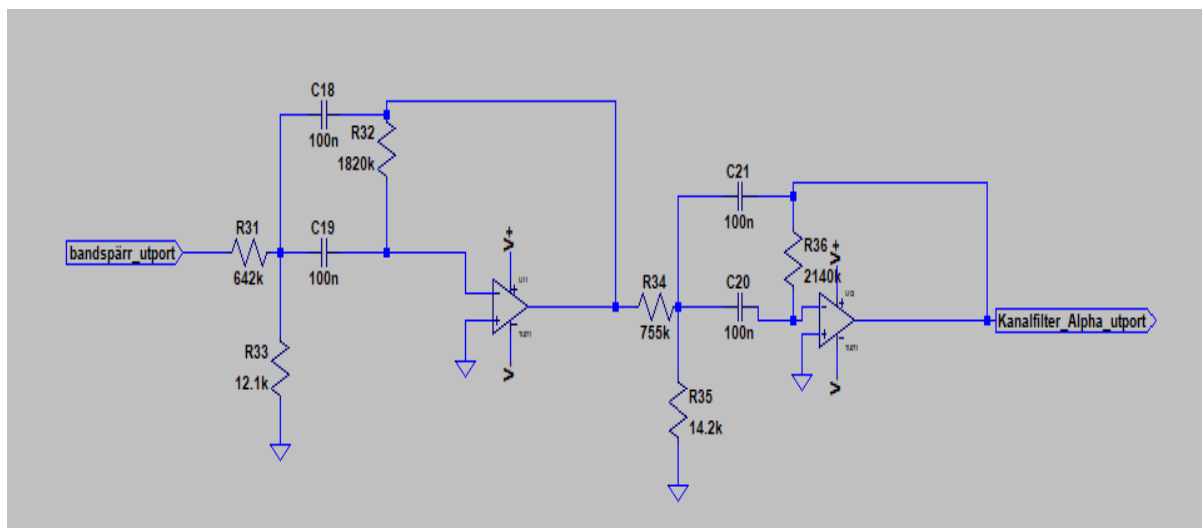
Figur 24 (småsignalanalys för bandspärrfiltret som dämpar ner frekvenskomponenten 50 Hz)



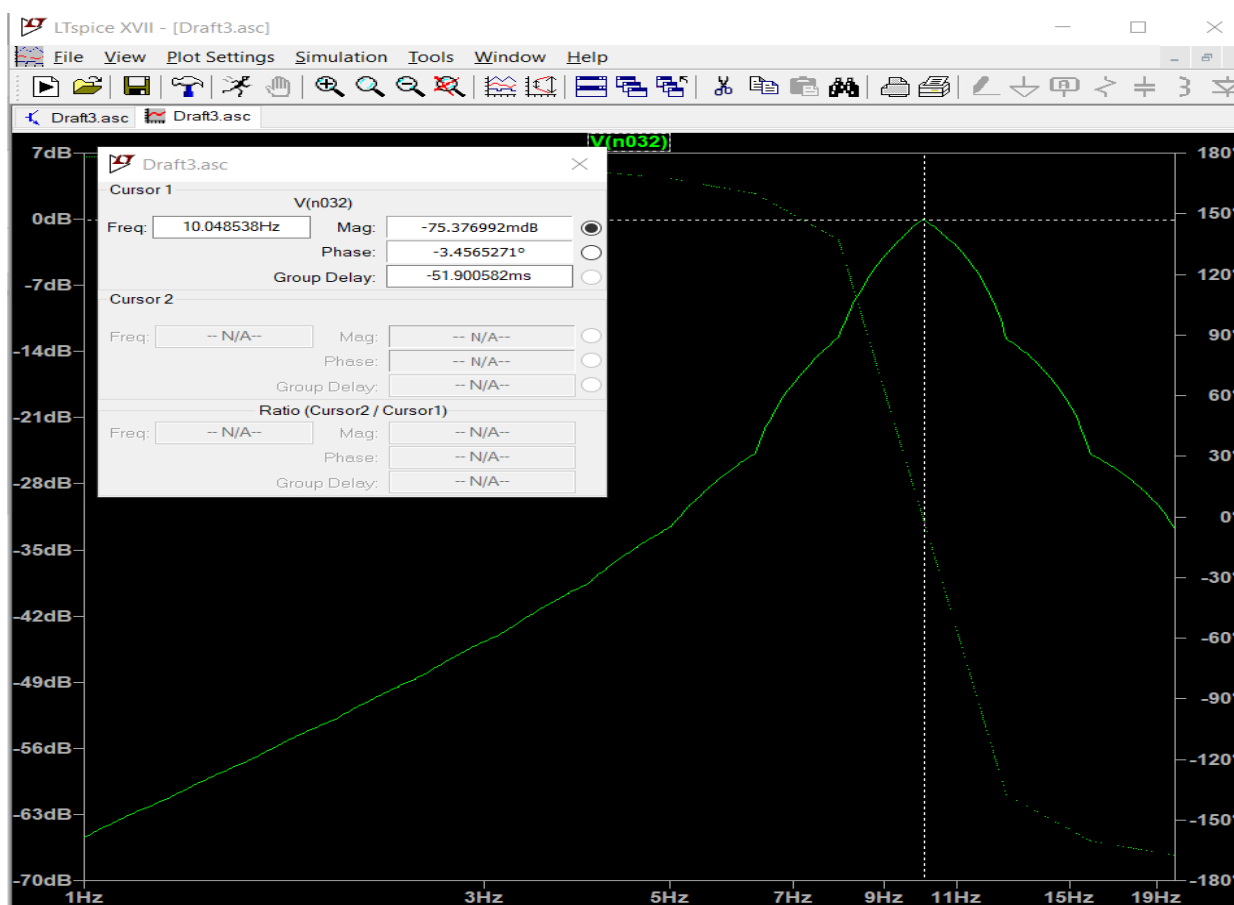
Figur 25 (koppling för 2a ordningens betakanalfilter)



Figur 26 (småsignalanalys för betakanalfilteret)



Figur 27 (kopplingsschema för alphakanalfiltret)



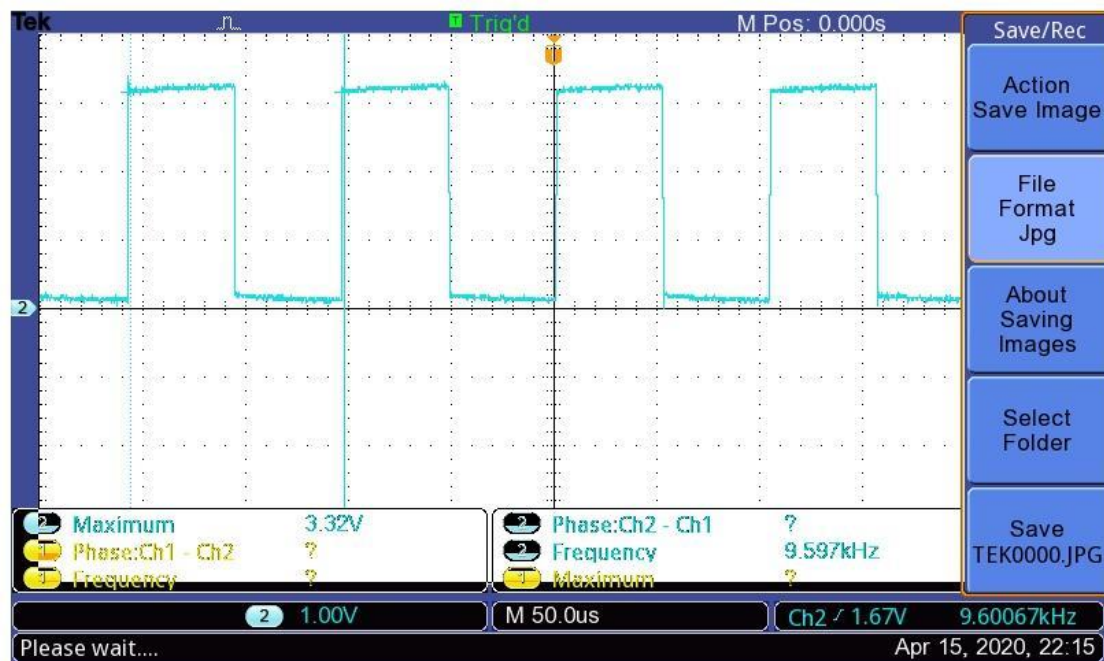
Figur 28 (småsignalanalys för alphakanalfiltret)

```

process (clk,reset) --process som generera sclk till adc
begin
if (reset='0') then
sclk_counter<=0;
clk_counter <=0;
sclk_flag <='0';
elsif(rising_edge(clk)) then
sclk_flag<='1';
if(clk_counter=(clk_to_sclk/2-1)) then
clk_counter <= clk_counter+1;
sclk_B<='1';
sclk <='1';
elsif(clk_counter=(clk_to_sclk-1)) then
clk_counter<=0;
sclk_B<='0';
sclk <='0';
if((sclk_counter>26) or (sclk_counter=26)) then
sclk_counter<=0;
else
sclk_counter<=sclk_counter+1;
end if;
else
clk_counter <= clk_counter+1;
sclk_flag<='0';
end if;
end if;
end process;

```

Figur 29 (VHDL process som genererar CLK pulser med frekvensen 9600 Hz)



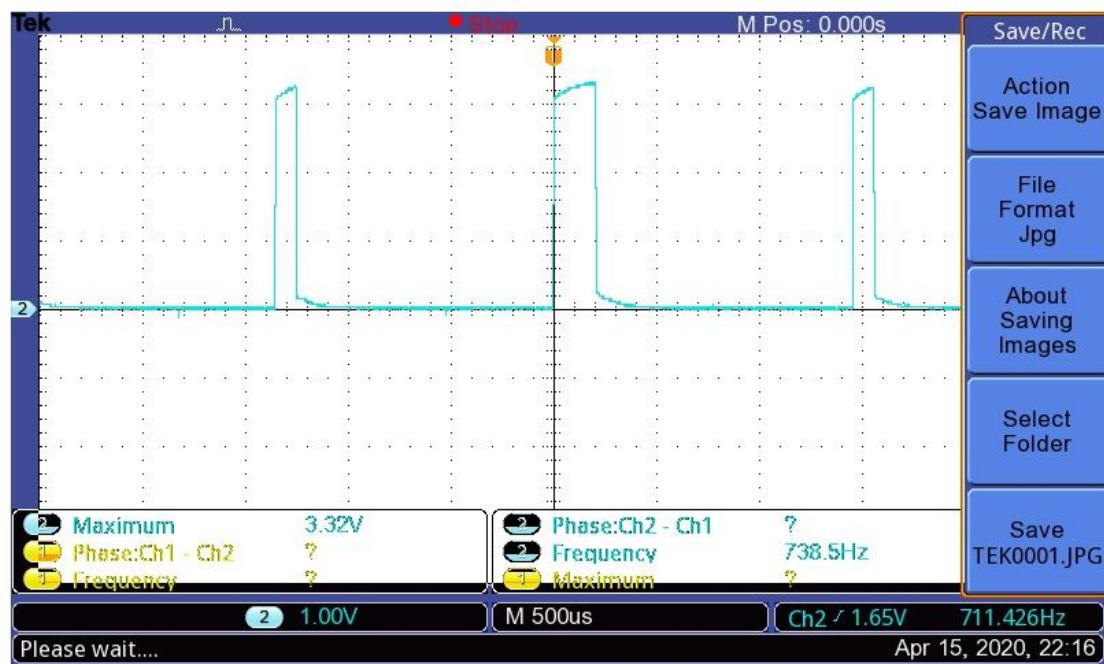
Figur 30 (Den uppmätta seriella pulsen i oscilloskop)

```

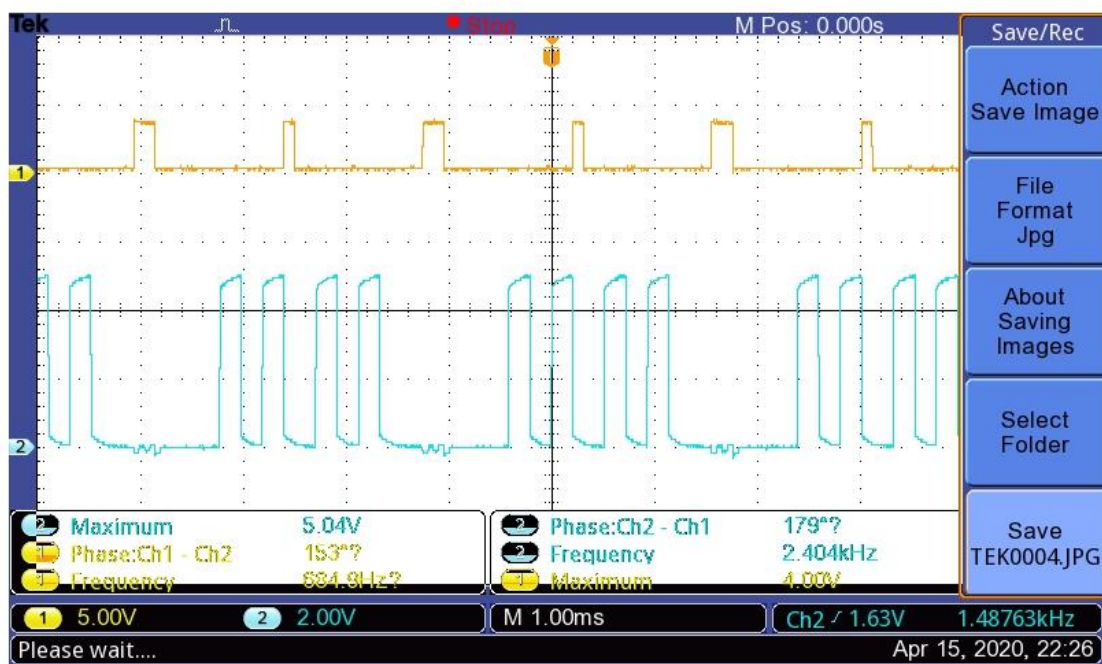
process(clk,reset) --process som generera conv puls
begin
if(reset='0') then
convert_start_B <='1';
elsif(rising_edge(clk)) then
if (sclk_counter < 11) then
convert_start_B<='0'; --lågt conv vid första 12 puls(när bit 12 till bit 1 MSB till LSB transmittas)
elsif(sclk_counter=11) then
convert_start_B<='1'; --conv puls högt vid puls 13th för att invertera bit sekvens, dvs LSB mode aktiveras
elsif((sclk_counter> 11) and (sclk_counter < 25)) then
convert_start_B <='0'; --conv puls blir låg efter 13th pulse som det står i datablad för LSB mode
else
convert_start_B <='1';
end if;
end if;
end process;

```

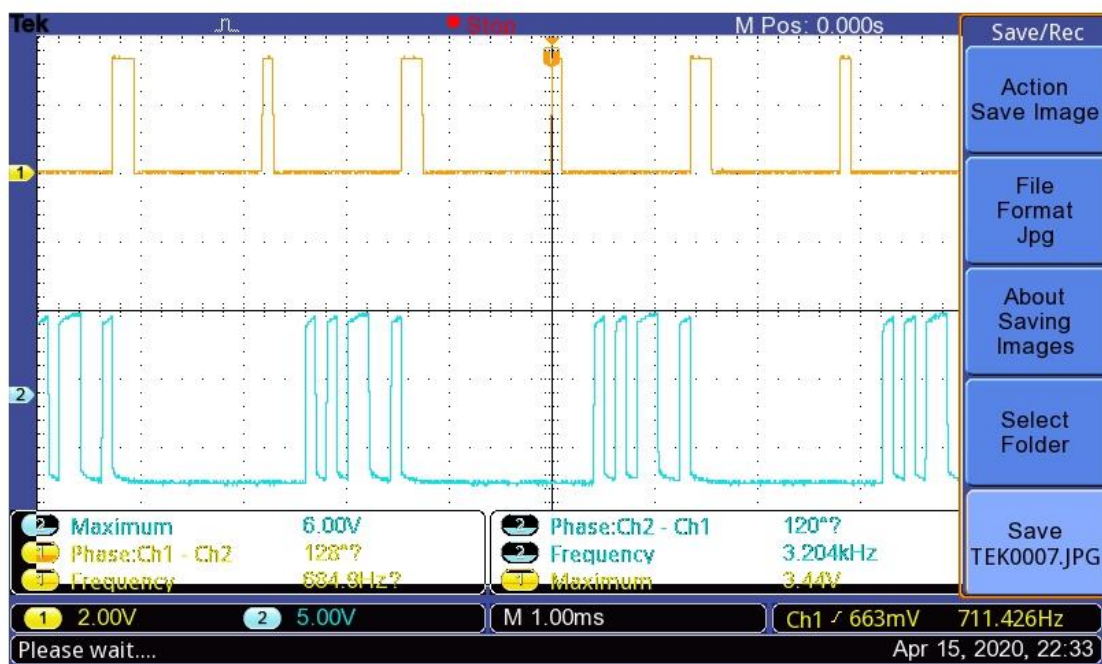
Figur 31 (VHDL process som genererar **CONV** pulser)



Figur 32 (Den uppmätta **CONV** pulsen i oscilloskop)



Figur 33 (gul är **CONV** och blå är skickad data)



Figur 34 (Den blå grafen är den digitala signalen efter RS232 länken och den gula grafen är CONV)


```

generic(clk_to_sclk : integer :=5208); -- intern clk av de_10 är 5Mhz, man får 9600 Hz om man klocka det 5208 gånger
port(
  clk : in std_logic; -- klock pulse av de_1,50 Mhz
  reset: in std_logic; -- reset kopplad till en switch i de_10
  Din: in std_logic; --seriellt in data som kommer från adc
  convert_start: out std_logic; -- conv signal
  sclk: out std_logic; -- seriellt klock pulse som ska skickas till adcr
  Dout: out std_logic; --seriellt utport som ska skicka data vidare till labview mha rs232 seriellt port
end ads;

```

Figur 35 (konfiguration av in och utportar)

```

process (clk, reset) --process som hantera buffert
begin
  if(reset='0') then
    convert_start<='0';
    Dout<='0';
    Din_B<='0';
  elsif (rising_edge(clk)) then
    convert_start<= convert_start_B;
    Din_B<=Din;
    Dout<=Dout_B;
  else
    end if;
  end process;
end arc;

```

Figur 35 (process som hanterar buffert)

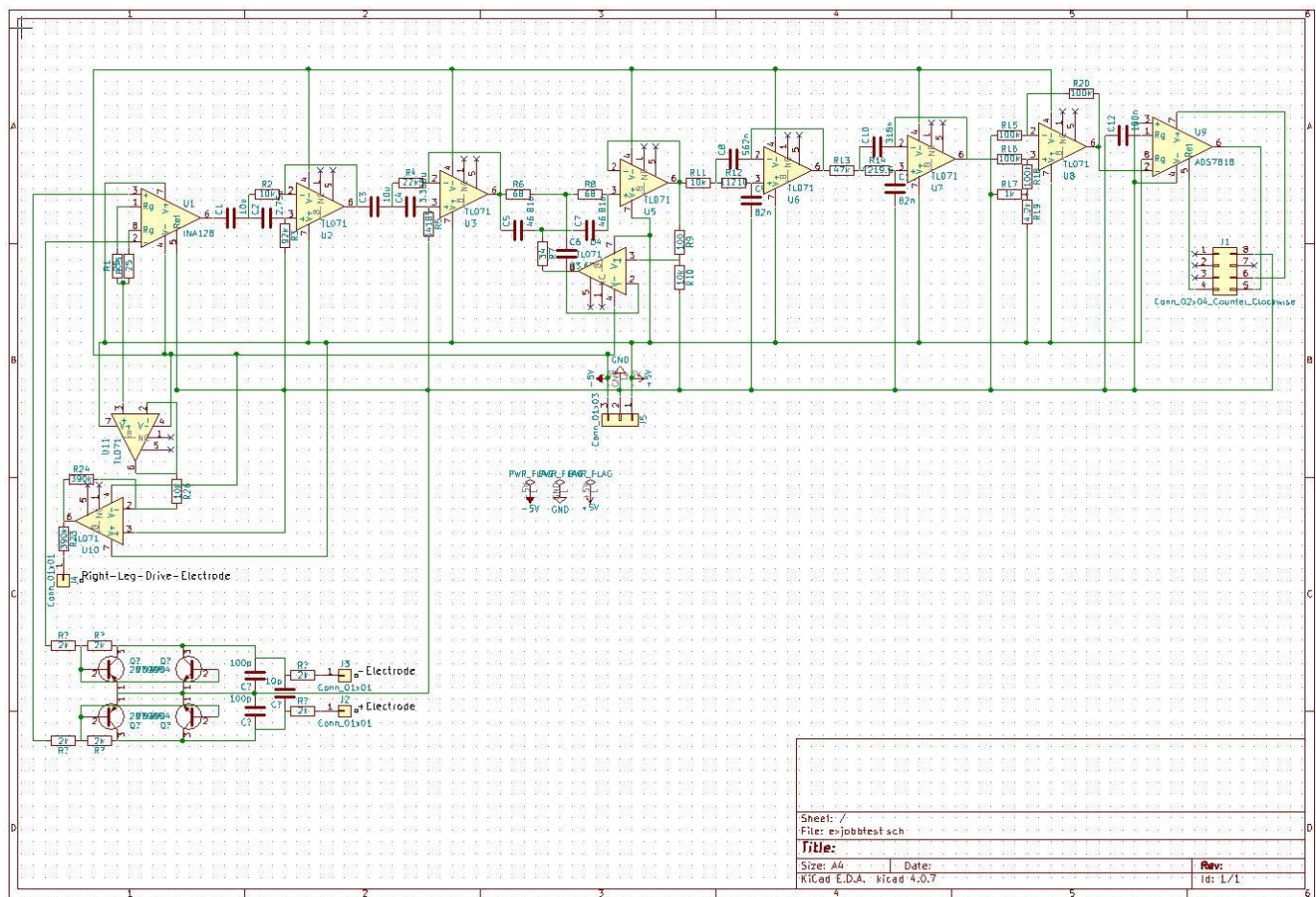
```

process(clk,reset) --process som transmittera 8 mest significant bitar och lägg till start och stop bit
begin
  if(reset='0') then
    Dout_B<='0';
  elsif (rising_edge(clk)) then
    if ((sclk_counter>16) and (sclk_counter<25)) then --Det mest 8 signifikanta bitar transmitteras
      Dout_B<=Din_B;

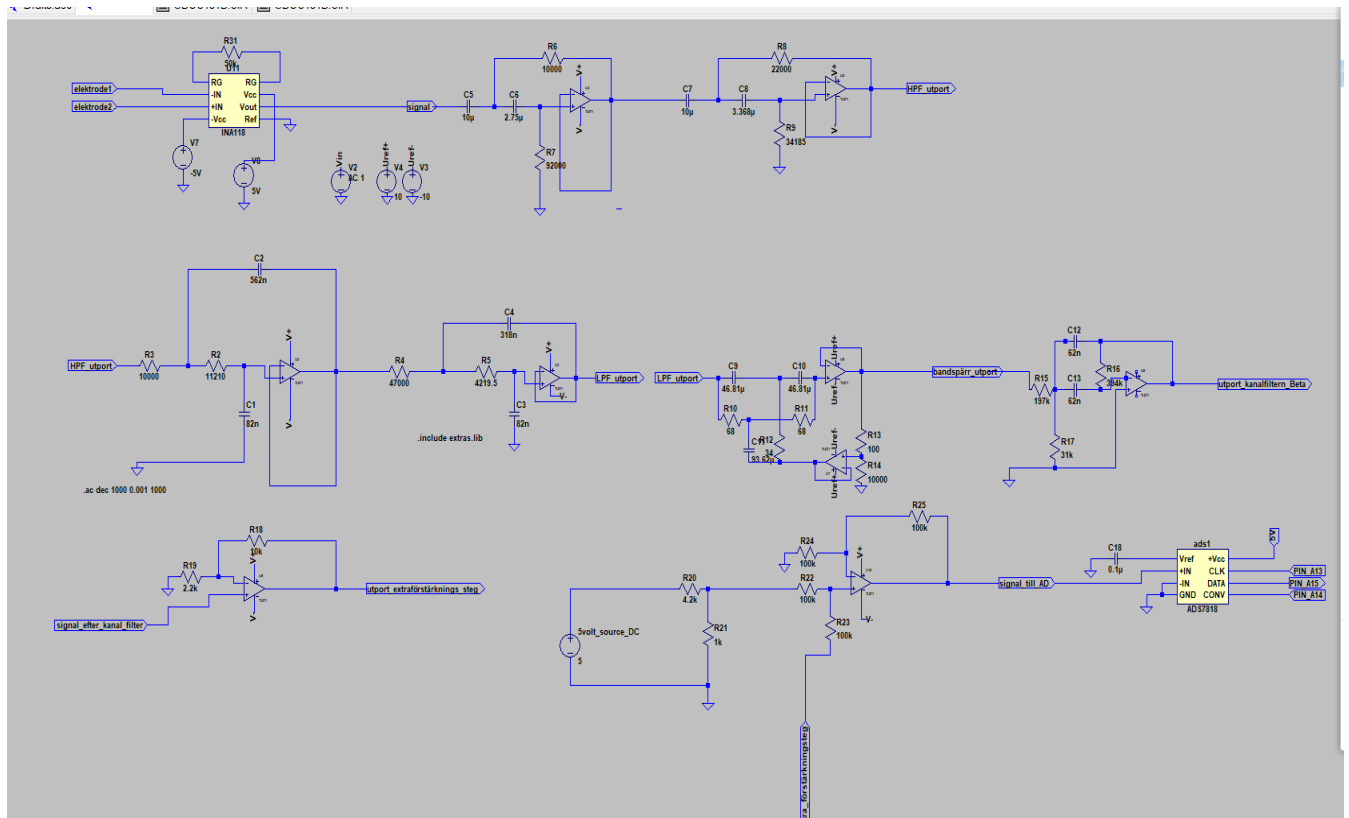
    elsif (sclk_counter<16) then
      Dout_B<='1'; --idle state för rs232 länken
    elsif (sclk_counter=16) then
      Dout_B<='0'; --lägg till startbit för transmitterad signal
    elsif (sclk_counter=25) then
      Dout_B<='0'; --lägg till stop bit
    elsif (sclk_counter>25) then
      Dout_B<='1'; --idel state för rs232 länken
    end if;
  end if;
end process;

```

Figur 36 (Processen som gör signalen redo för transporterering över RS232 länken)



Figur 37 (PCB schema)



Figur 38 (schema för testkrets)



Figur 39 (från vänster till höger: DE_1, kopplingsplatta (ADC, analog krets med filter och förstärkning), elektroder som appliceras på skalpen)