

*Jens Kofod Hansen*

# **GPS Synkroniseringsmodul**

## **Videosynkronisering vha. GPS**

Bachelorprojekt, April 2007



Diplom E - bachelorprojekt ved Ørsted – DTU  
af Jens Kofod Hansen

Afleveret d. 10-04-2007  
I samarbejde med DK-Technologies A/S



*Jens Kofod Hansen*

# **GPS Synkroniseringsmodul**

## **Video synkronisering vha. GPS**

*Bachelorprojekt, April 2007*



Diplom E - bachelorprojekt ved Ørsted – DTU  
af Jens Kofod Hansen

Afleveret d. 10-04-2007  
I samarbejde med DK-Technologies A/S



## **GPS synkroniseringsmodul, Videosynkronisering vha. GPS**

### **Rapporten er udarbejdet af:**

Jens Kofod Hansen

### **Vejleder(e):**

Thomas Holm Hansen - DK-Technologies A/S

Gøsta Thuesen, Ørsted - DTU

### **Ørsted•DTU**

Danmarks Tekniske Universitet

Ørstedes Plads

Bygning 348

2800 Kgs. Lyngby

Denmark

[www.oersted.dtu.dk](http://www.oersted.dtu.dk)

Tel: (+45) 45 25 38 00

Fax: (+45) 45 93 16 34

E-mail: [info@oersted.dtu.dk](mailto:info@oersted.dtu.dk)

---

Udgivelsesdato: 10. April, 2007

Klasse: 2 (fortrolig)

Udgave: 1. udgave

Bemærkninger: Denne rapport er indleveret som led i opfyldelse af kravene for opnåelse af titlen diplomingeniør på Danmarks Tekniske Universitet.

Rettigheder: © DK-Technologies A/S, 2007



Diplom E - bachelorprojekt ved Ørsted – DTU  
af Jens Kofod Hansen

Afleveret d. 10-04-2007  
I samarbejde med DK-Technologies A/S

## Abstract

The rapport explains a solution method for synchronisation of several video sources (called genlocking), by the use of GPS-technology. Concepts of videosynchronisation and how LTC timecode works, is explained, and gives an overview of the project. The methods are explained theoretically, as well as how it is implemented on the actual hardware module. The hardware consist of a MCU and FPGA kernel, which by the use of a GPS-chipset, can obtain high precision synchronisation. Further more, several problems which occurred during the design of the module, is solved.



## Resumé

Rapporten gennemgår en løsningsmetode til synkronisering af flere videokilder (kaldet genlocking), ved brug af GPS-teknologi. Koncepterne indenfor videosynkronisering, samt hvordan LTC tidskode fungerer forklares og giver et overblik over opgaven. Metoderne gennemgås teoretisk, og der forklares hvordan de er blevet implementeret på selve hardwaren. Hardwaren består af en MCU og FPGA kerne, der ved hjælp et GPS-chipset kan generere højpræcisionssynkronisering. Desuden gennemgås hvordan forskellige problemstillinger, der er opstået ved design af opgaven, løses.



## Indholdsfortegnelse

Abstract.....	6
Resumé.....	7
Liste over figurer.....	9
Liste over tabeller.....	10
1. Indledning.....	11
1.1 Baggrund.....	11
1.2 Problemformulering.....	12
1.3 Metode.....	12
1.4 Afgrænsning og forudsætninger.....	12
1.5 Arbejde af andre.....	12
2. Introduktion til genlock-modul.....	13
2.1 Genlock.....	13
2.2 LTC.....	14
2.3 GPS.....	15
2.4 Beskrivelse af PT5300 mainframe.....	16
3. Teori.....	18
3.1 Overordnet FPGA og MCU programstruktur.....	20
3.2 PLL-opbygning.....	23
3.3 Stabilitet af GPS oscillator.....	25
3.4 Genlock signalgenerator.....	28
3.5 LTC generator.....	30
3.6 Indstillings og opsætningsmuligheder (tidszone, sommer/vintertid, videoformat).....	34
4. Praktisk afsnit.....	35
4.1 Hardwarebeskrivelse.....	35
4.2 Software og protokoller.....	38
4.3 Software – MCU.....	43
4.4 Software – FPGA.....	46
5. Test og evaluering af GPS genlock og LTC kode.....	51
5.1 Funktionalitetstest.....	51
5.2 Kvalitetstest (Jitter/præcision af signaler, stabilitet/langtidstest).....	51
6. Videre forbedringer og optimeringsmuligheder.....	52
6.1 Montering af oven VCXO (OCXO).....	52
6.2 Montering af ekstern præcisions DAC.....	53
7. Konklusion.....	54
7.1 Ufærdige dele.....	54
7.2 Løsning.....	54
Appendiks A: Hardware design overvejelser .....	56
Appendiks B: Ordliste.....	60

## Liste over figurer

.....	14
Figur 2-1: Forskydning af NTSC frekvens.....	14
Figur 2-2: PT5300 mainframe.....	16
Figur 3-1: Diagram for overordnet dataflow.....	19
Figur 3-2: Flowdiagram for opstart system.....	21
Figur 3-3: Blokdiagram for PLL.....	23
Figur 3-4: Tællemetode til måling af 1 PPS faseforskydningen.....	25
Figur 3-5: Forskydning af 1 PPS over tid.....	26
Figur 3-6: Flowdiagram for clockkompensator.....	27
Figur 3-7: Forskydning af NTSC-periode.....	28
Figur 3-8: Opsummering af forskydningsberegning.....	29
Figur 3-9: Diagram over LTC-opbygning.....	30
Figur 3-10: Bifase-modulering af 0-bits og 1-bits, samt tider for videoformater.....	30
Figur 4-1: Oversigt over hardware-modul.....	37
Figur 4-2: Foto af hardware-modul.....	37
Figur 4-3: Clockdiagram for I2C læsning og skrivning.....	39
Figur 4-4: RS232 input flowchart.....	44
Figur 4-5: NMEA parser flowchar.....	45
Figur 4-6: FPGA opstartssekvens.....	47
Figur 4-7: Opbygning af tri-state controller.....	48
Figur 4-8: Flowdiagram for I2C-state machine.....	49
Figur 4-9: Simuleret periodeforlængelse.....	50
Figur 5-1: OCXO implementerings diagram.....	53
Figur A-1: Microstrip design software.....	57
Figur A-2: Microstrip på PCB.....	57
Figur A-3: Topologi for Sallen-Key filter.....	58
Figur A-4: Filter karakteristik for LTC-filter.....	59

## Liste over tabeller

Tabel 3-1: timing i forhold til frame nulpunkt.....	28
Tabel 3-2: Neddeling af 148.5 MHz til LTC clockfrekvenser.....	31
Tabel 3-3: Afrunding af nedclockningsfaktor til LTC-frekvenser .....	31
Tabel 3-4: Opbygning af LTC dataframe for PAL og NTSC.....	33
Tabel 4-1: Oversigt over NMEA beskeder i NavSync modulet.....	40
Tabel 4-2: Oversigt over POLYT besked.....	41
Tabel A-1: Funde LTC filter komponent-værdier.....	58

# 1. Indledning

## 1.1 Baggrund

I alle TV-produktioner, er det vigtigt at video-signaler kører synkront (kaldet genlocking, en forkortelse for generator locking). Der findes flere video-standarder, men fælles for alle er, at de viser et bestemt antal billeder pr. sekund. Skiftes der mellem video-kilder, er det naturligvis vigtigt, at den ene kilde har vist præcis ét helt billede, og at den nye kilde *skal til* at vise et nyt billede. Problemer opstår, hvis video-kilderne ikke er synkroniseret ordentligt, og der skiftes kilde efter visning af eksempelvis et halvt billede. Effekten bliver et rullende billede, kendt fra VHS-bånd, hvor flere videoklip er optaget asynkront oven i hinanden.

Det har længe været standard, at man havde én master sync-generator i et produktions-studie, der forgrenede en master-clock ud til andre sync-generatorer eller kameraer. Problemet opstår, når der indgår eksterne studier eller broadcast-vogne, i en live tv-produktion. Her er det ikke muligt at have indbyrdes synkronisering via kabler. Derfor ønskes en trådløs løsning fundet.

GPS (Global Positioning System) er en udbredt og billig teknologi. Teknologien er mest brugt til positionsbestemmelse, men viser sig også at være udmærket til timing-applikationer. På hver GPS-satellit befinder sig fire atom-ure, der er ultrastabile. Atom-ure er dyre, men ved at synkronisere et billigere quartz-krystal op til GPS-satellitens atom-ur, får man en høj præcision til en brøkdel af prisen.

I følgende rapport gennemgås en løsning, til synkronisering af video-signaler, vha. en GPS-modtager.

## **1.2 Problemformulering**

En trådløs metode til synkronisering af videosignaler (kaldet genlocking) skal implementeres. Desuden skal modulet kunne tidsstemple video-signaler vha. LTC (Linear Time Code). Modulet ønskes udviklet ved brug af GPS-teknologi, på et allerede eksisterende hardwaremodul. Rapporten gennemgår altså primært softwaremetoder, der er taget i brug ved udvikling. Følgende problemstillinger ønskes redegjort for:

- Hvordan låses en masterclock bedst muligt op til GPS-frekvensen?
- Hvordan bruges GPS-tid og frekvens til synkronisering og LTC-kodning?
- Hvordan kommunikeres der på tværs af hardware-moduler (MCU, FPGA og GPS)?
- Hvad sker der ved GPS-svigt og forstyrrelser, og hvordan afværges dette?

## **1.3 Metode**

Opgaven skal analyseres som flere moduler. På topniveau ses hele modulet med sine underkomponenter (FPGA, MCU og GPS-chip). Et flowchart opbygges for at beskrive overordnet funktionalitet, og et diagram over ønskede kommunikationsveje og protokoller beskriver hvordan modulerne skal forbindes indbyrdes. Fra topniveau nedbrydes modulet til gradvist lavere funktionsblokke, der hver skal fungere selvstændigt.

## **1.4 Afgrænsning og forudsætninger**

GPS-modulet fungerer som et indstiksmodul i en overordnet mainframe (PT5300), fremstillet af DK-Technologies A/S. For at få et endeligt brugbart produkt, kræves det at der udvikles en ny software til denne mainframe. Dette er uden for projektets omfang, og vil ikke mindske forståelsen for GPS-modulet. Dog vil der ikke være en brugergrænseflade til GPS-modulet, og opsætningen skal altså ændres direkte i kildekoden, eller via programmeringsinterfacet (JTAG-interfacet).

## **1.5 Arbejde af andre**

Projektet bygger videre på arbejde udført i praktikperioden. Praktikken omfattede bla. valg af komponenter, udlægning af PCB, montering og fejlsøgning. Eksamensperioden omfatter implementering af software. Dog er softwarestrukturen blevet overvejet ved hardwaredesign, og hardwaren vil blive beskrevet kort, for at opnå fuld forståelse for projektet.

Envidere er der benyttet en PLL (Phase-locked loop), der tidligere er blevet anvendt i DK-technologies produkter. Der er både benyttet den hardwaremæssige implementation, såvel som en VHDL-komponent, der fungerer som digital loopfilter og styring. Funktionaliteten af PLL'en vil i denne rapport blive beskrevet, men de designmæssige overvejelser vil ikke blive gennemgået.

## 2. Introduktion til genlock-modul

I følgende afsnit findes en hurtig gennemgang af koncepterne bag de implementerede elementer. Hvert koncept er beskrevet kort, og der vil blive gået mere i dybden med de enkelte funktionaliteter i afsnittet ”Teori”.

### 2.1 Genlock

Genlocksyste­mer bruges til mastersynkronisering af flere kameraer eller andre videokilder. Genlocksyste­mets hovedfunktioner består i at generere fire synkroniseringssignaler, horisontal-, vertikal-, frame- og farve-synkronisering. Alle signaler skal køre synkront med hinanden, og indbyrdes imellem flere videokilder. Skiftes der imellem to videokilder der ikke er låst op til hinanden, opstår der et ”rullende” billede, indtil synkroniseringen igen er opnået.

De fire synkroniseringssignaler er også en del af et almindeligt composite videosignal, så almindelige fjernsyn kan vise billedet korrekt. Et standardiseret videosignal, der udover de fire synkroniseringssignaler, ikke indeholder nogen information, bruges ofte til mastersynkroniseringssignal. Dette består af et helt sort billede, kaldet et ”black burst” signal. Dette signal kan enten komme fra en ekstern black burst generator, eller blive genereret internt i genlock-modulet.

Desuden bruges synkroniseringssignalerne også til at bestemme lige eller ulige linier ved interlaced billedevisning. Dvs. hvor et halvt billede reelt set kun vises ad gangen, ved skiftevis kun at vise lige eller ulige linier. Dette sker ved dobbelt framerate, så billedet fremstår stabilt.

Ved at benytte en GPS-modtager, kan man bruge modtagerens meget præcise timinginformation samt synkroniseringspulser til at implementere et genlock system, hvor videosignalerne ikke kun er låst indenfor det samme tv-studie, men også i en større bygning, til broadcast-vogne eller med udendørsstudier. Dette forhindrer en del komplikationer, der førhen har været, samtidig med at kabelføring bliver mindre.

## 2.2 LTC

LTC (Linear Timecode) er en standard defineret af Society of Motion Picture and Television Engineers (SMPTE) og benyttes til at tidsstemple og synkronisere lyd og billedsekvenser. Tidskoden indeholder framenummer og tidsstempel for den enkelte frame i sekunder, minutter og timer. Derudover indeholder en tidskode-frame øvrige data som f.eks. frame-format, fase-kompensation og nogle bits afsat til brugerdefinerede data.

Da NTSC-frameraten ikke går op i hele sekunder, opstår der en uoverensstemmelse mellem tidskoden og frame-tælleren. Et sekund i NTSC-tid er  $1/1,001$  sekund real-tid. Hvis der tælles til 30 frames på et NTSC-sekund, vil tiden være 3,6 sekunder forskudt i forhold til real-tid efter en time. På **figur 2-1** ses, hvordan 30/1,001 Hz frekvensen langsomt bliver forskudt i forhold til 30 fps. Derfor er der indført en anden tællemetode kaldet *dropframe*. Denne metode tæller ikke frame 00 og 01 med i hvert minut, bortset fra minut-tallene 00, 10, 20, 30, 40 og 50. Herved minimeres tidsforskydningen mellem NTSC-tid og real-tid til -3,6 ms.



**Figur 2-1:** Forskydning af NTSC frekvens

LTC signalet er et bifasekodet lydsignal, der ligger omkring 2 kHz afhængig af videoformatet. Bifasekodning fungerer således, at der for en høj bit er én faseovergang, hvorimod der for en lav bit ingen faseovergang er. Der skiftes samtidig fase for hver bit. En hel LTC-frame er på 80 bits, derfor ændres bitraten i forhold til frameraten (f.eks. 25 fps for PAL og ca. 30 fps for NTSC).

I LTC koden er indlagt en synkroniseringssekvens bestående af et fast bitmønster på 16 bits. De 14 midterste bits er symmetriske omkring midten, hvilket gør det muligt at detektere synkroniseringssekvensen i begge retninger. Første og sidste bit er komplementære til hinanden, hvilket gør det muligt at detektere hvilken retning koden afspilles i.

## 2.3 GPS

GPS (Global Positioning System) benyttes oftes i sammenhæng med navigation eller tracking af mobile enheder vha. satellitter. GPS kan give præcis information om position, hastighed og retning, udfra de over 50 GPS satellitter, der er i omløb om Jorden. Satellitterne er placeret således, at der hele tiden er midst fire synlige satellitter over horisonten, så en modtager kan få forbindelse, uanset hvor på Jorden den befinder sig.

En mindre åbenlys brug af GPS, er at man kan modtage præcis timing-information. GPS-satellitterne er hver især udstyret med fire atomure, der løbende holdes synkrone med jord-baserede kontrol-stationer. Atomure er dyre, men ved at købe en billig GPS-modtager kan man synkronisere GPS'ens quartz-clock op til satellitens atomur, og derved opnå en meget præcis clock.

Dette opnås ved at GPS satellitten sender en pseudo-tilfældig sekvens (*Coarse Acquisition code* eller *C/A*) på 1023 bit ved en frekvens på 1,023 MHz. Denne sekvens benyttes tildels til at identificere satellitten, der hver især har en unik sekvens, men også til at synkronisere til en bitstrøm af navigationsdata. Bitstrømmen synkroniseres ved korrellering af GPS'ens *C/A* sekvens og en tilsvarende sekvens genereret i modtageren. En hel side navigationsdata fylder 37.000 bit, og sendes ved en bitrate på 50 bps. Derved tager det 12,5 minut at sende en hel side. *C/A* og navigationsdata mixes sammen og moduleres med en bærebølge på 1575.42 MHz, som sendes som radiobølger til modtagere på Jorden.

GPS tid er forskellig fra UTC (Universal Coordinated Time) også kendt som GMT (Greenwich Mean Time), idet det ikke er tilrettet Jordens omdrejning. Derved er GPS tiden nu foran UTC tid, fordi UTC tilpasses med skudsekunder, for at være synkron med døgnet. Måden man definerer GPS-tid på, er også forskellig fra UTC tid. GPS tid opdeles i ”antal uger” og ”antal sekunder i ugen”, i stedet for timer, minutter, sekunder, dage og år. GPS tid startede kl 00:00 d. 6 januar år 1980.

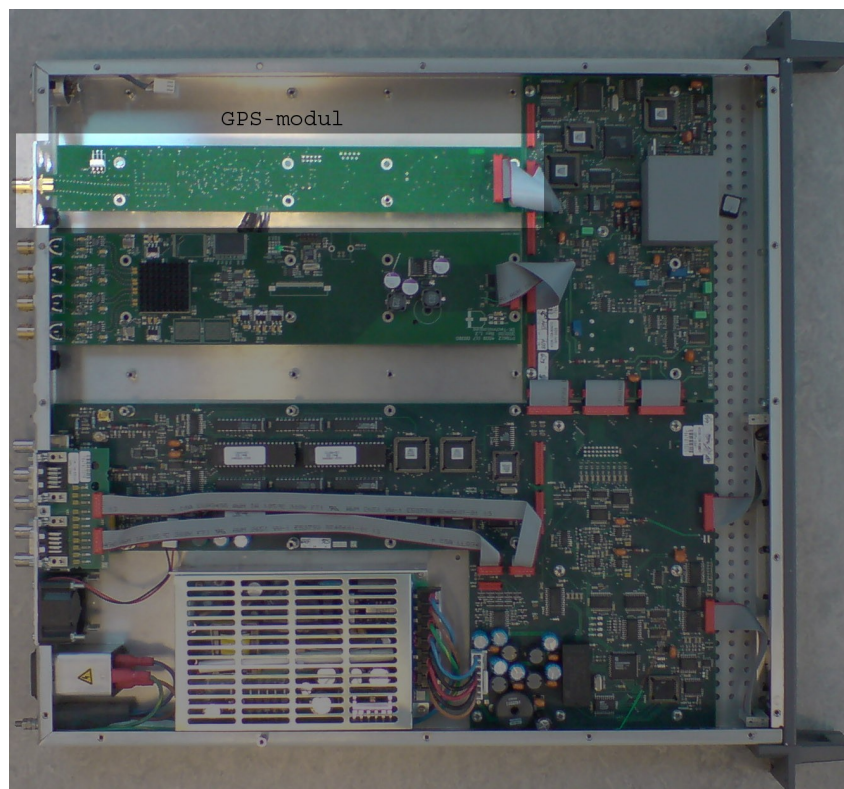
Selvom GPS tiden er meget præcis kan der opstå forstyrrelser på forskellig vis, der kan forårsage *clockdrift*. Dette kan være forstyrrelser forårsaget af ionosfæren, satellit-clocken, kvantiserings- og beregnings-fejl osv. Dog kan mange af disse fejl reduceres i de forskellige timing-optimerede GPS chipsets der er tilgængelige. Dette sker ved at modulet sættes til at antage, at positionen er stationær, og benytte diverse adaptive algoritmer til at opnå en højere præcision.

Ved at benytte en præcis frekvens fra GPS-modtageren og den absolutte GPS-tid, kan man implementere signalgeneratorer, der kan genskabe synkrone signaler uafhængigt af hinanden. Dermed er GPS en ideel mulighed for at skabe et Genlock-modul, der er uafhængig af et mastersignal (*black burst* signal).



## 2.4 Beskrivelse af PT5300 mainframe

GPS-Genlock modulet er ikke en selvstændig enhed, men en mindre del af en sync-generator. DK-Technologies sync-generator kaldes PT5300, og er modul-opbygget, så den kan opbygges til kundens behov. Der er plads til forskellige typer moduler, men de enkelte moduler kan dog ikke sidde på en vilkårlig plads. På **figur 2-2** ses et billede af PT5300, og markeringen angiver hvor GPS-modulet sidder:



**Figur 2-2:** PT5300 mainframe

PT5300 mainframen har indtil videre kun haft mulighed for at synkronisere lokale apparater, idet det har været krævet at trække kabler imellem disse og mainframen. Master-sync-kilden har typisk været en SDI-generator (Serial Digital Interface), som har genereret et black-burst signal. Et black-burst signal, er et video-signal, der kun indeholder sync-pulserne og ingen billeddata, og dermed er helt sort. Fremover vil det dog blive muligt at synkronisere frit, vha. GPS-timing, hvis bare der er satellit-forbindelse til rådighed.

På mainframen sidder en højpræcisions-krystal oscillator, indbygget i en 'ovn'. Ovnens opgave er at holde quartz krystallens temperatur og dermed også frekvens forbliver konstant. Denne frekvens låses til input-sync-signalet, hvad enten det er fra et black-burst signal eller et GPS-modul. Når først oven-oscillatoren er fastlåst, ændres den ikke ved mindre jitter fra sync-kilden. Mindre forstyrrelser fra kilden filtreres altså væk.

Mainframen indeholder desuden software, der gør det muligt at kommunikere med indstiksmodulerne. Når et indstikmodul sidder i mainframen, bliver det registreret, og en række menupunkter kommer til rådighed, alt efter hvilket modul der indsættes i mainframen. Det er altså i mainframens software, at opsætning af sommertid, tidszoner osv. skal foretages. Disse data sendes så til GPS-modulet, der benytter valgene. Mainframens software omfattes ikke i denne rapport, og vil ikke blive beskrevet nærmere.

### 3. Teori

Det primære modul i systemet vil være GPS chippen. Denne skal kunne levere tidspunkt til tidstempling vha. LTC tidskode, samt levere en præcis GPS-timet frekvens til lås af masterclock. Tidspunkt skal udlæses i to formater. Til LTC koden bruges UTC tid, der bruges til civil tidsregning. Til genlock systemet bruges GPS tid, der er en fast tidsenhed uden skudsekunder. GPS tid startede kl 00:00 d. 6. januar 1980. Ved at bruge dette tidspunkt som absolut nulpunkt for signalgeneratoren, kan synkrone genlocksignaler for flere isolerede genlock systemer opnås.

Som signalgenerator benyttes en FPGA (Field Programmable Gate Array – programmerbar logik). Synkroniseringssignaler til videoformaterne PAL, NTSC, 24 FPS skal kunne genereres. Derfor skal FPGA'en have en clock, der kan nedclockes til disse framerates. For heltalframerates (24, 25 og 30 fps), er denne clock mindre væsentlig. Men fordi NTSC frameraten ikke er et heltal (30/1.001 fps), kan der ikke benyttes en vilkårlig clockrate. En brugbar clockrate er 148,5 MHz, der ved en nedclockningsfaktor på 4954950 clockcycles giver NTSC frameraten. Derfor vælges der en VCXO (Voltage Controlled Xtal Oscillator) på 148,5 MHz.

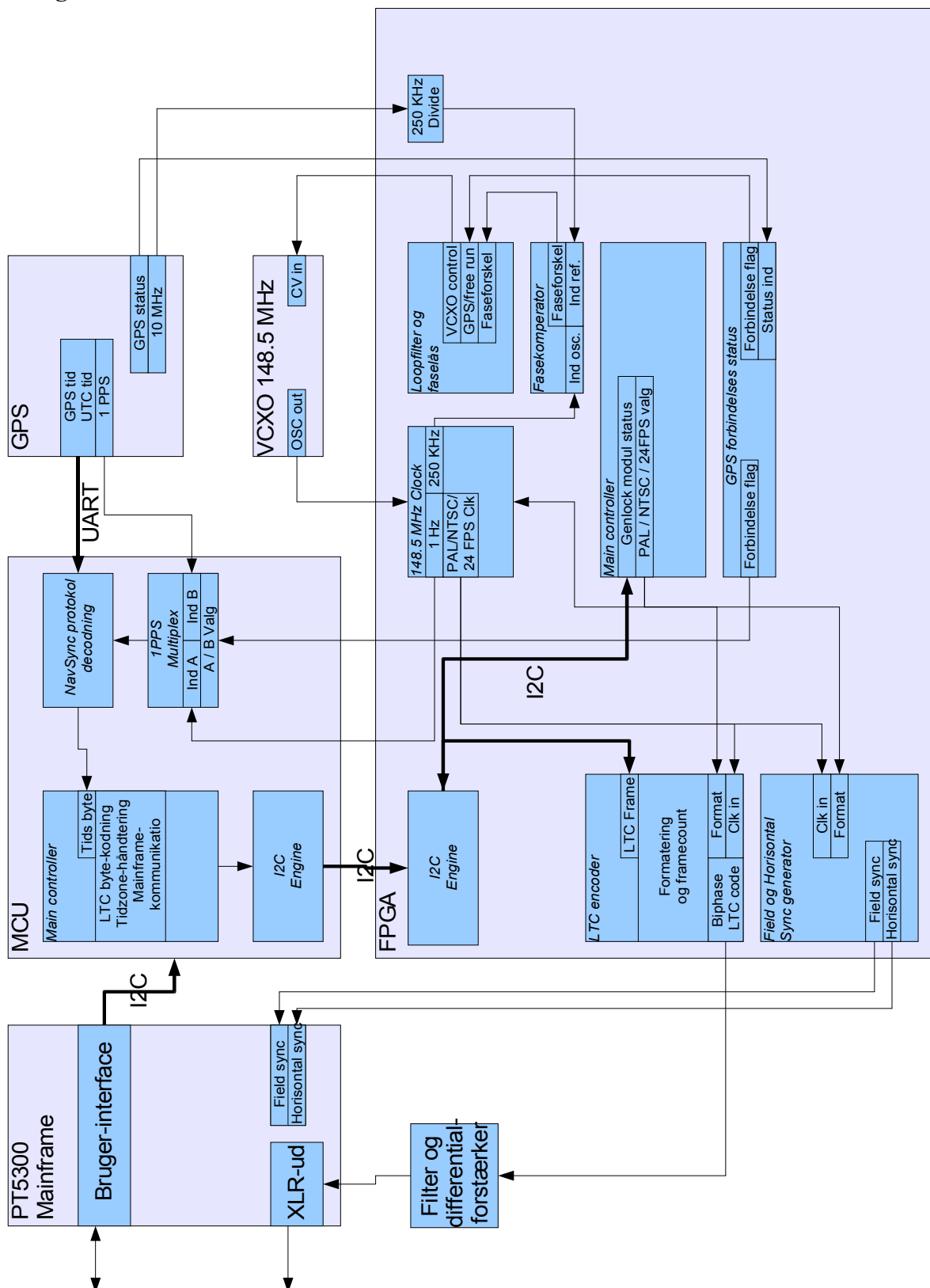
Præcisionen af FPGA'ens clock er vigtig, da den afgør kvaliteten af genlock modulet. Ved at låse clocken til GPS'ens clock, låses der i teorien til de meget præcise atomure i GPS satellitterne. Dog kan der ske forstyrrelser i selve GPS modtageren, men disse fejl kan der til dels kompenseres for i FPGA'en. Der låses vha. af en PLL (Phase-locked loop), der skal optimeres til at eliminere jitter.

Når FPGA'en kører ved en fast frekvens, skal den generere de ønskede synkroniseringspulser og LTC kode. Derfor skal GPS- og UTC-tid kendes af FPGA'en. FPGA'en har ikke direkte tilgang til GPS-modulet, da det er uhensigtsmæssigt, at læse og parse GPS-protokollen i FPGA'en. Derfor sættes en MCU (microcontroller) som mellemed til denne og diverse andre opgaver.

Interfacet imellem MCU og FPGA bliver en I2C-bus, hvor FPGA'en sættes til slave. MCU'en styrer desuden overordnet opsætning, såsom tidszone og videoformat. Der skal være direkte forbindelse imellem GPS-modtager og FPGA, der angiver om der er satellitforbindelse. FPGA'en skal kunne frakoble PLL'en og fastlåse VCXO-frekvensen, i det tilfælde, at satellitforbindelsen svigter.

MCU'en har desuden til opgave at være hovedled imellem genlock-modulet og mainframen. På mainframen sidder et brugerinterface med en række menupunkter til opsætning af de installerede moduler. MCU'en skal give mainframen besked om, at GPS-modulet er monteret. Mainframen bestemmer herudfra, hvilke opsætningsmuligheder der skal være i menuen. Disse valgmuligheder er tidszone-indstillinger, samt indstilling for sommer og vintertid.

På **figur 3-1** ses hvordan det overordnede data flow ser ud.



**Figur 3-1:** Diagram for overordnet dataflow

### 3.1 Overordnet FPGA og MCU programstruktur

FPGA'en og MCU'en skal sammensættes til en *kerne*, der fungerer som hovedmotor for GPS-modulet. Systemet skal køre stabilt og kunne håndtere uventede situationer, som f.eks. tab af satellit-forbindelse, at PLL'en går ud af lås, forskydning af clockdomæner, etc. Hovedkernen skal kunne eksekvere opstartsekvensen ved power-on, administrere data-flow, overvåge system-status og styre eksterne hardware-komponenter, såsom GPS-modul og VCXO.

Det er også denne del af systemet der sørger for at levere den præcise master-clock, som alle underkomponenter kører på. Disse underkomponenter er genlock-systemet og LTC-generatoren. Det er nødvendigt at kernen er yderst stabil, da alle dele afhænger heraf. Denne kerne sørger for at alle dele overholder de strenge realtidskrav.

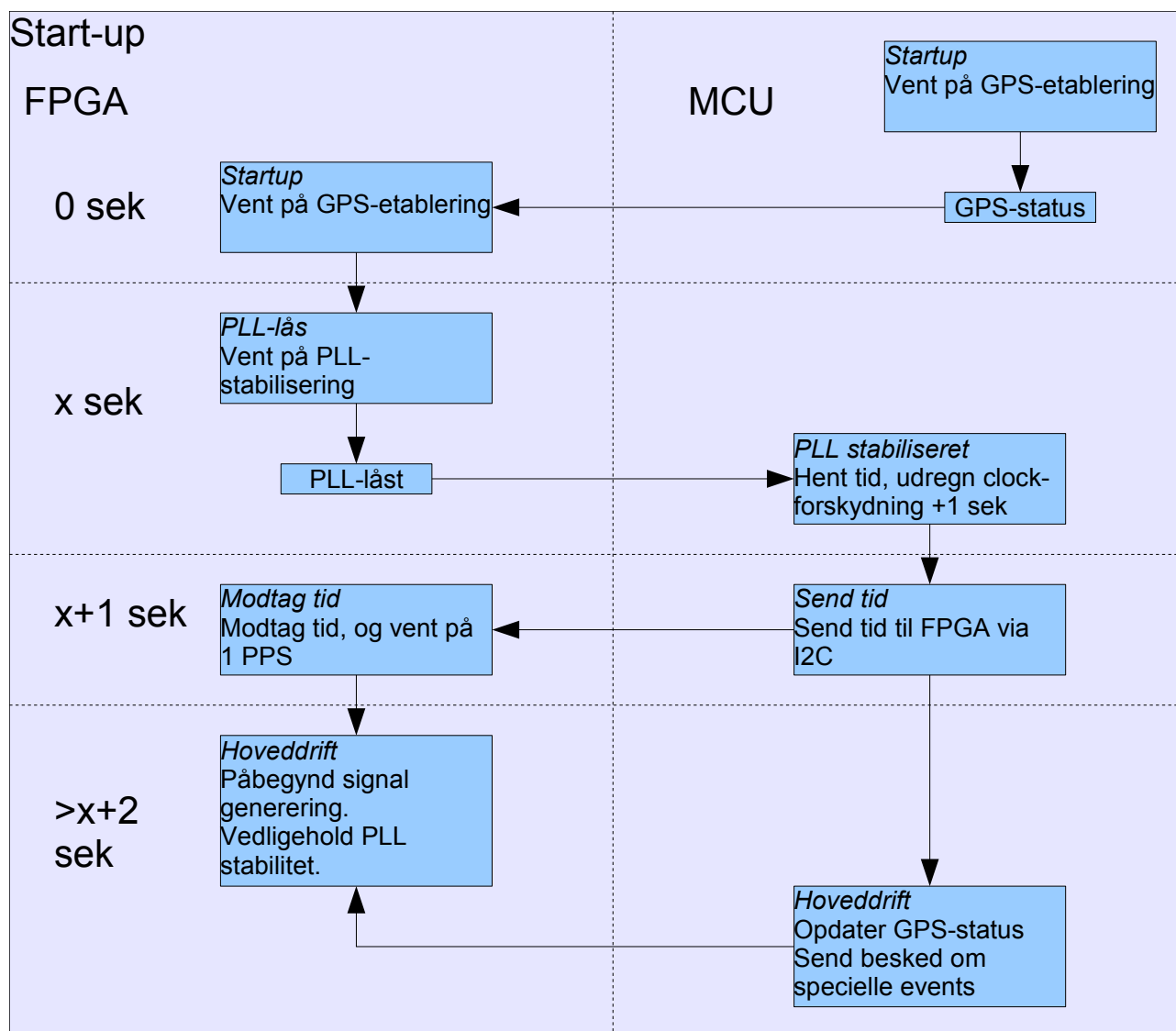
#### 3.1.1 Opstartssekvens

En vigtig funktion i kernen, er at den kan starte systemet op, så hver delkomponent bliver etableret i takt med at PLL'en bliver låst og satellit-forbindelsen bliver oprettet. Desuden skal modulet give besked til mainframen, om at det er monteret. **Figur 3-2** viser flowchart for den overordnede opstart.

Kernen genererer et statusflag, der signalerer om både GPS-forbindelsen og PLL'en er stabiliseret. En tæller registrerer tiden, hvori systemet har været stabilt. Dette skal forhindre, at systemet starter lige så snart GPS-forbindelsen er oprettet, da forbindelsen kan være ustabil lige ved etablering, og at der kan være usikkerhed i de leverede data. Tælleren sørger for at systemet starter efter et antal sekunder med låst PLL og etableret GPS-forbindelse. Herefter giver et master reset-signal besked til alle underkomponenter, om at systemet er stabilt, og underfunktionerne kan påbegyndes.

Ved påbegyndelsen, skal en vigtig faktor beregnes, nemlig tiden fra det absolutte nulpunkt for GPS'en. Genlock systemet skal starte samme sted i perioden, uanset hvilket system der startes op. For video-systemer med hele framerates (24 og 25 FPS) er det ikke et problem, men NTSC frameraten på 30/1,001 FPS, starter ikke samme sted på et vilkårligt sekund. Hvert 1001 sekund ligger 'start' synkront med hele sekunder. Sekundet efter er starten på en NTSC frame forskudt en smule, og ud fra GPS-tidens ugenummer og tid på ugen, skal denne forskydelse beregnes.

Det samme gælder for LTC'en, hvor starten også forskydes, men ydermere forskydes UTC-tiden. Desuden forskydes tiden forskelligt for dropframe- og non-dropframe-LTC. Det vil ikke være hensigtsmæssigt, at udregne tidsforskydningen ud fra GPS-nulpunktet, da forskydningen ellers ville blive for stor. Derfor opsynkroniseres tiden på døgn-basis, eksempelvis kl 00:00 hvert døgn. Starttiden vil blive beregnet i MCU'en, da denne opgave ikke er egnet for en FPGA. Tiden skal beregnes forud, så den aktuelle tid kan være i systemet, ved påbegyndelse af signal-generering.



**Figur 3-2:** Flowdiagram for opstart system

### 3.1.2 Hoveddrift

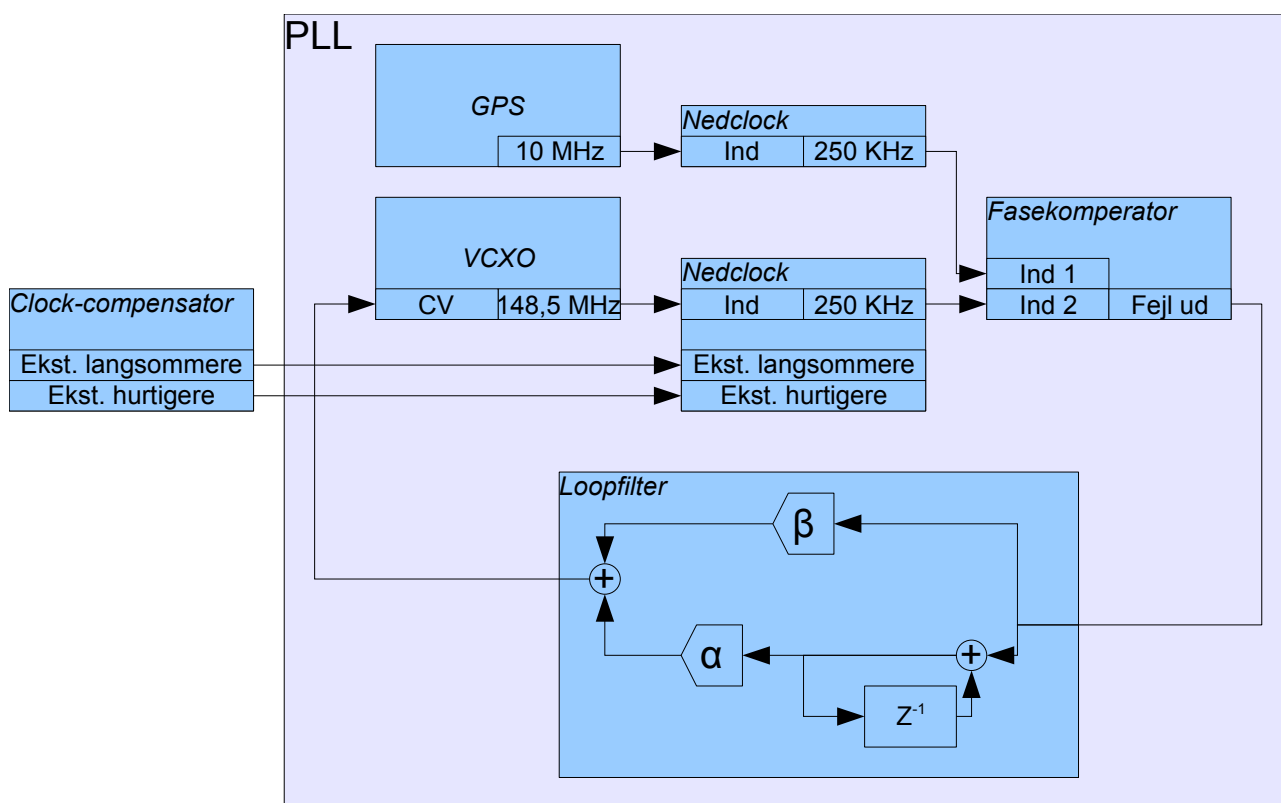
Når systemet kører, er der adskillige funktionaliteter der skal vedligeholdes. Det vigtigste er overvågning og styring af master-clocken. Clocken styres af en PLL, der kører selvstændigt som en underkomponent. Sker der en forstyrrelse i reference-clocken, eller mistes GPS-forbindelsen, er det vigtigt af PLL'en slås fra, og at hovedkernen tager over. Slås PLL'en fra, benyttes en midlet styreværdi i stedet, til at fastholde VCXO'en. Den midlede styreværdi skal ikke indeholde de seneste værdier, da fejl godt kan opstå før systemet registrerer dem. Derved ville fejlene indgå i denne værdi. Værdierne midles derfor først et antal sekunder efter de er blevet modtaget.

Når clocken kører stabilt, kan hovedfunktionerne køre uafbrudt i takt med hinanden. Genlock signalerne skal køre periodisk, og LTC-generatoren skal tælle frames og tid uafbrudt. Signal-generatorerne skal ikke påvirkes af nogle former for svigt af GPS eller i tilfælde af at PLL'en går ud af lås.

Når synkroniseringen først er blevet etableret, skal den bevares. Svinger GPS'en, skal en fast styrespænding fastholde VCXO'en. Svinger PLL'en derimod, vil alle signaler blive mere påvirkede, da det er masterclock-signalet, der forskydes. Dette skal så vidt muligt forhindres, men skulle det ske, er det hensigtsmæssigt, ikke at hard-synkronisere signalerne, ved stabilisering af GPS eller PLL, da dette kunne resultere i et stort spring i generatoren. Ved hard-synkroniseringen menes at resette synkroniseringen til GPS-tid øjeblikkeligt. Der skal så vidt muligt synkroniseres op langsomt og flydene, da dette ikke vil blive bemærket i samme grad.

### 3.2 PLL-opbygning

PLL'en vil blive baseret på et design, tidligere benyttet i andre produkter hos DK-Technologies. PLL'en vil altså bestå af en selvstændig VHDL-underkomponent, der skal kunne varetage sin egen drift i de fleste situationer. Dog skal det fra top-komponenten være muligt for at se status af låsen, samt kompensere for forstyrrelser af GPS-clocken, ved at påvirke loopet. Sidstnævnte funktionalitet er speciel for GPS-modulet, og skal altså implementeres i den eksisterende komponent.



**Figur 3-3:** Blokdiagram for PLL

PLL'en består først og fremmest af en fasedetektor. Fasedetektoren sammenligner referencen fra GPS'en med en VCXO. Da referencen er 10 MHz og VCXO'en er 148,5 MHz, må der nedskaleres til en fælles clockrate. Den højeste fællesnævner for begge clocks er 250 KHz, og der indsættes derfor to nedclocknings-blokke før fasedetektoren.

Faseforskellen vil give en fejl i det tilfælde, at frekvens eller fase er forskellig. Denne fejl sendes til feedback-loopet igennem et førsteordens led, samt direkte, begge med variable gains på henholdsvis alpha og beta. Førsteordens ledet har et højt gain, og kompenserer for DC-fejl, mens det direkte led har et lavere gain, og kompenserer for hurtige udsving.

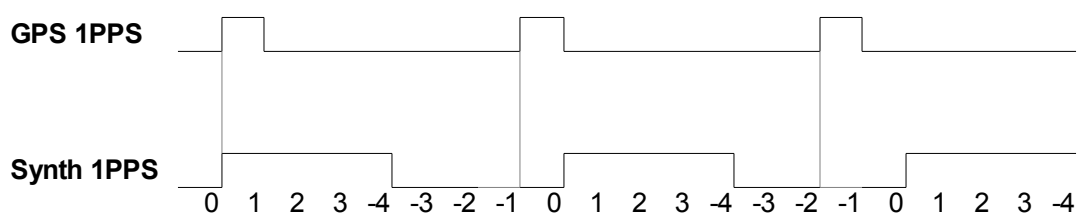


For at kunne påvirke PLL'en udefra til at få VCXO'en til at køre hurtigere eller langsommere, tilkobles to signaler. Disse skal bruges til at kompensere for fejl i referencen, der har akkumuleret sig over længere tid. Dette kan der læses mere om i afsnittet "stabilitet af GPS oscillator". En test af stabilitet og jitter i PLL'en kan findes i afsnittet "Kvalitetstest (Jitter/præcision af signaler, stabilitet/langtidstest, støj)"

### 3.3 Stabilitet af GPS oscillator

GPS-modulets 10 MHz clock og 1 PPS (1 pulse per second) skulle i det idéelle tilfælde gerne være helt synkrone. VCXO'en på 148.5 MHz, som er låst til GPS'ens 10 MHz clock, skulle derfor gerne tælle sin 148.500.000'ende periode på 1 PPS'en. Denne præcision er et kvalitetskriterie for Genlock og LTC systemet.

Det kan hurtigt undersøges, hvorvidt de forskellige clocks er synkrone. Ved at lade FPGA'en tælle en periode på 1 Hz ved 148,5 MHz og sammenligne med GPS'ens 1 PPS, kan faseforskydningen nemt findes. På **figur 3-4** ses hvordan der tælles.



**Figur 3-4:** Tællemetode til måling af 1 PPS faseforskydningen

Som det ses af eksemplet ovenfor forskydes clocken først 0, så -1 og til sidst -2 clockperioder. Der tælles fra -74.500.000 til 74.244.999 i koden, som er et sekund talt ved 148,5 MHz.

En serie af clockforskydninger udlæses fra FPGA'en, for at se hvorledes mønsteret er. Et eksempel på dette kan se på **figur 3-5**. Ved at studere sekvensen, ses det at resultatet ofte udlignes løbende. En clockcycle foran efterfølges ofte af en clockcycle bagud, og resultatet ender derved i nul. Disse forskydninger må anses som kvantiserings støj der ikke kan korrigeres for.

Det sker dog i nogle tilfælde, at en forskydning ikke får udlignet sig, og over lang tid vil disse små forskydninger akkumuleres til en uacceptabel stor forskydning, som der skal kompenseres for.

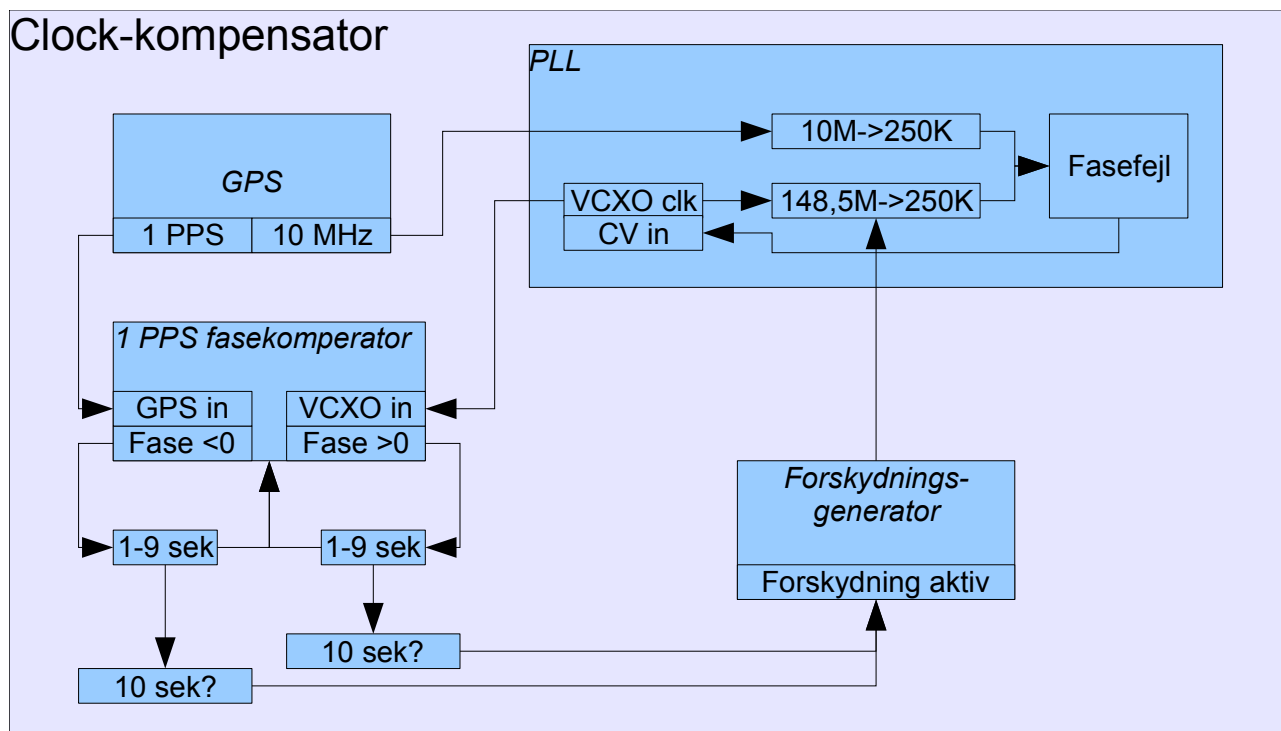
Kompenseringen skal foretages langsomt og ubemærket, så der ikke forekommer forstyrrelser i synkroniseringssignalerne. Det er derfor ikke hensigtsmæssigt at tilføje/fjerne en 148.5 MHz clockcycle, for at indhente referencen. En mere elegant løsning, er at foretage en justering via PLL'en. PLL'ens langsomme loop-filter vil gøre en kompensering mere ”transparent” og ”glidende”.

Proceduren skal kun lave indgreb hvis en forskydning ikke udlignes efter længere tid. Derfor skal en counter tælle antallet af sekunder, hvori clock'en er forskudt. Udlignes forskydningen, resettes counteren. Udlignes forskydningen derimod ikke efter et vis antal sekunder, skal et indgreb foretages i PLL'en. Herefter skal der ventes til systemet har stabiliseret sig, hvorefter forskydningen overvåges igen.

buffer[0]	00
buffer[1]	00
buffer[2]	00
buffer[3]	00
buffer[4]	00
buffer[5]	00
buffer[6]	-01
buffer[7]	01
buffer[8]	00
buffer[9]	00
buffer[10]	-01
buffer[11]	01
buffer[12]	00
buffer[13]	00
buffer[14]	-01
buffer[15]	01
buffer[16]	-01
buffer[17]	01
buffer[18]	-01
buffer[19]	01
buffer[20]	00

**Figur 3-5:**  
Forskydning af 1 PPS  
over tid

## Clock-kompensator



**Figur 3-6:** Flowdiagram for clockkompensator

På **figur 3-6** ses hvordan to countere tæller op ved hhv. for lang eller for kort periode-tid. Når eksempelvis 10 sekunder, aktiveres en forskydnings-generator, der påvirker clocken til PLL'en. Denne vil så enten forøge eller sænke hastigheden på VCXO'en, idet den tror den er ude af lås, og derved vil faseforskydningen på den genererede 1 PPS mindskes.

### 3.4 Genlock signalgenerator

Signalgeneratoren laver field og horisontal-pulser (herefter f- og h-pulser) til mainframen. Mainframen låser sig fast på disse pulser, og alle andre signaler kører herefter på denne lås. På mainframen sidder desuden en meget præcis oven-oscillator, der gennem en langsom PLL, mindsker jitter betydeligt.

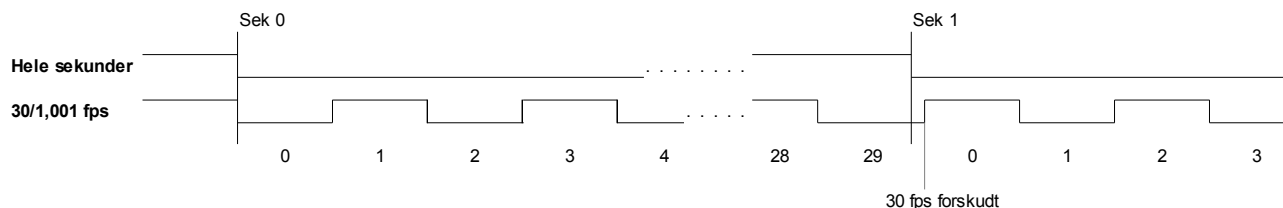
Pulserne er ikke standard-signaler, men interne signaler der benyttes af DK-Technologies mainframe. Tiderne er målt ved at indsætte et standard genlock-modul i PT5300 mainframen og måle med et oscilloskop. Duty-cycles og offsets i forhold til hinanden er altså ikke taget efter en standard, men specielt defineret til denne mainframe. Dog er frekvenserne stadig i overensstemmelse med et givet videoformat, f.eks. 50 Hz for PAL fieldpulser. I **tabel 3-1** kan ses hvilke periode-tider der benyttes til de forskellige video-formater. Offset er hvornår første *opadgående* flanke kommer i forhold til frame-nulpunktet. Værdierne er afrundet til én decimal.

**Tabel 3-1:** timing i forhold til frame nulpunkt

Videoforformat	NTSC	PAL	24 FPS
H-offset	0,3 us	0,4 us	0,4 us
H-lavtid	43,7 us	52,4 us	54,6 us
H-højtid	9,7 us	11,6 us	12,1 us
F-offset	59,8 us	71,7 us	74,7 us
F-lavtid	43,8 us	52,5 us	54,7 us
F-højtid	39947,5 us	39947,5 us	41612 us

#### 3.4.1 Cyklus

Udregning af cyklusen for signalgeneratoren, gælder kun for NTSC-frekvensen, som det vil blive vist. Problemet består i, at frekvensen ikke er et heltal såsom PAL (25 fps) eller 24 fps, men derimod 30/1,001 fps. Derved opstår den situation, at periodestarten ligger forskudt i forhold til hele sekunder, som det ses på **figur 3-7**.



**Figur 3-7:** Forskydning af NTSC-periode

24 og 25 fps starter præcist på hvert hele sekund. 30/1,001 FPS starter kun synkront med sekunderne hvert 1001. sekund. Derfor skal der et offset på, der angiver forskydningen for NTSC-signalet. Offsettet skal findes ud fra tidens nulpunkt, som er d. 6. januar 1980 kl 00:00. Dvs. at for hver vilkårlig generator der igangsættes, anses den for at være startet præcist på førnævnte tidspunkt, og kører derfor synkront med alle andre generatorer.

Offsettet er forholdsvis simpelt at finde. Udfra GPS tiden, kan man beregne hvor mange sekunder siden det absolutte nulpunkt, der er gået. Da forskydningen er nul, hvert 1001. sekund, beregnes modulus 1001 af sekunder siden nulpunktet, for at finde hvor i 1001-sekundsperioden forskydningen er. For at mindske størrelsen af værdierne benyttes også det faktum, at forskydningen er nul hver 143. uge. Herefter kan forskydningen i sekunder omregnes til en forskydning i clock-cycles ved 148,5 Mhz. En opsummering af udregningen ses på **figur 3-8**.

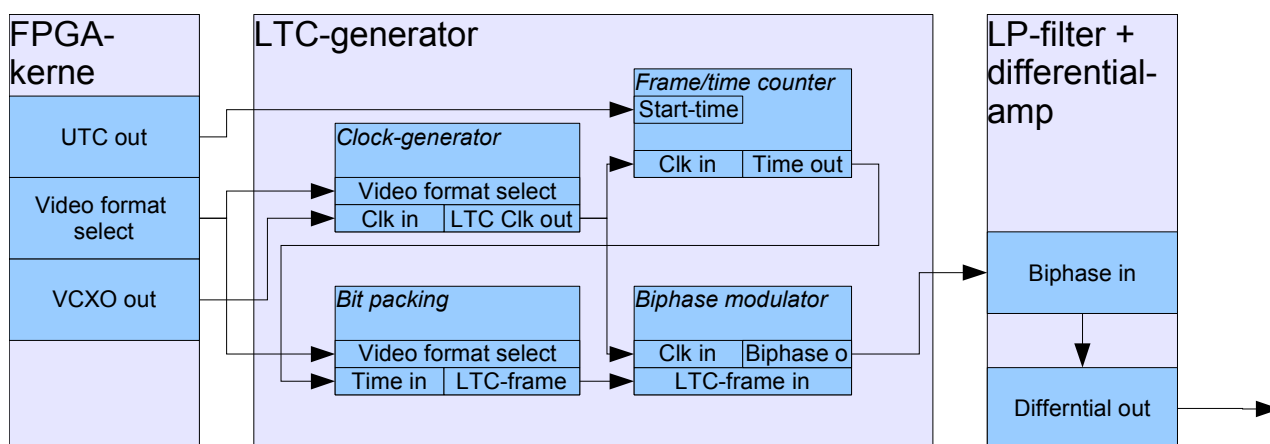
Antal uger <b>MOD</b> 143	Ugers forskydning
Ugers forskydning <b>X</b> Sek. pr. uge + sek i aktuel uge	Sek. forskydning
Sek. forskydning <b>MOD</b> 1001	Periodeforskydning
Periodeforskydning <b>X</b> 4450549	Clockcycle forskydning
148.500.000 – (clockcycleforskydning <b>MOD</b> 148.000.000)	Reel forskydning

**Figur 3-8:** Opsummering af forskydningsberegning

## 3.5 LTC generator

### 3.5.1 Oversigt

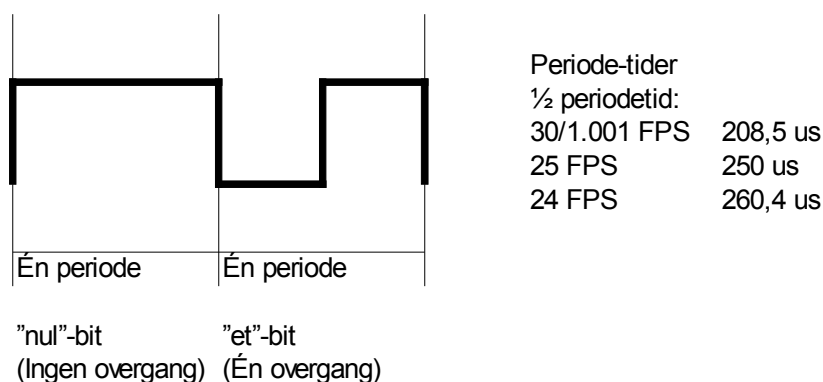
Input består af masterclocken, samt tidsinformation, både som UTC og GPS tid. Output består af tids-informationen kodet som et bifasemoduleret signal i de respektive videosystemers framerate. Ændring af de forskellige videoformater skal kunne foretages fra mainframens interface. Opbygning ses på **figur 3-9**.



**Figur 3-9:** Diagram over LTC-opbygning

### 3.5.2 Bifasemodulering

Bifasemodulering er en metode til at overføre binære værdier via én signalvej. Metoden bygger på, at der for hver bitperiode er en faseovergang. Disse faseovergange ligger med et fastlagt interval, indenfor en vis tolerance (SMPTE normen angiver tolerancen for LTC til 1% afvigelse på periodetiderne). Sendes et "nul"-bit, er der kun denne ene faseovergang pr. bit, men sendes et "et"-bit, indsættes der yderligere én faseovergang ved den halve periodetid.



**Figur 3-10:** Bifase-modulering af 0-bits og 1-bits, samt tider for videoformater.

### 3.5.3 Framerates

LTC-generatoren skal internt generere de forskellige videosystemers clockrates ud fra systemets 148,5 MHz masterclock. Der er 80 bits per billedframe. Dette multipliceret med frameraten giver bitraten, dvs. henholdsvis 1920, 2000 og 2400/1,001 bps for 24, 25 og 30/1,001 FPS. Dog skal clocken være 2 gange bitraten, for at kunne gengive et ”et”-bit, jf. skemaet ovenfor.

Bitclocken skal genereres fra en neddeling af masterclocken. Neddelingsfaktoren findes ved at dividere masterclockraten med den dobbelte bitrate. Neddelingsfaktoren for de respektive videosystemer ses udregnet i **tabel 3-2**.

**Tabel 3-2:** Neddeling af 148.5 MHz til LTC clockfrekvenser

Framerate	2 x bitraten	neddelingsfaktor
24 FPS	3840	77343,75
25 FPS	4000	74250
30/1,001 FPS	4800/1,001	30968,4375

Da ikke alle clockrates går op i masterclocken, må der afrundes i nedclockningen, og indsættes længere eller kortere perioder med et vist interval. Derved opnås den ønskede nedclockningsfaktor, set over længere tid. Et eksempel herpå kunne være følgende sekvens "2 2 3 2 2 3 2 2 3", hvor der kan anses at være en neddelingsfaktor på  $2 \frac{1}{3}$ .

Et andet kriterie er, at der ikke er for stor forskel mellem den korte og den lange periodetid. En for stor forskel vil resultere i for meget jitter på det modulerede signal. Kravet er mindre end 1% afvigelse på periodetiderne, jf. SPTME normen.

**Tabel 3-3:** Afrunding af nedclockningsfaktor til LTC-frekvenser

Framerate	Kort neddeling	Lang neddeling	Forhold
24 FPS	38671	38672	1/7
25 FPS	37125	0	1
30/1,001 FPS	30968	30969	9/7

### 3.5.4 Dropframe og tidsforskydning

Dropframe er en metode, hvormed man kan kompensere for tidsforskydningen der opstår, når man tæller ved 30 FPS, når den reelle framerate er ca 29,97 FPS. Dvs. at tidstælleren kommer bagud. Efter en time vil tælleren være 107,89 frames bagud, hvilket svarer til 3,6 sekunder. Dette svarer til 1 ms pr.sekund.

Ved at skippe to frames for hvert minut, bortset fra minuttallene 00, 10, 20, 30, 40 og 50, kommer



tælleren forud med præcis 108 frames. Dette mindsker tidsforskydningen til -3,67 ms efter en time. For at forskydningen ikke skal akkumuleres over længere tid, resettes tiden hvert døgn kl. 00:00. Dvs. at den maksimale forskydning bliver 86,4 millisekunder efter 24 timer.

For at simplificere udregningen af forskydningen, ses der bort fra, at der ikke springes frames over i hele ti'er-minuttal. I stedet tages et gennemsnit for, hvor meget tiden forskydes pr. sekund. Dvs. at der ved hvert sekund lægges  $33 \frac{1}{3}$  ns til tiden.

### 3.5.5 cyklus

cyklusen for LTC-generering stemmer overens med cyklusen for syncpulsgenerering. Dvs. at for 24 FPS og PAL skal der ikke udregnes en forskydning, men for NTSC forskydes sekundstart i overensstemmelse med genlockforskydningen. Se evt. afsnittet om genlock, side 25.

Ydermere skal der udregnes en forskydning for tiden ved NTSC, hvad enten det er for dropframe eller nondropframe. Som ved periodeforskydningen, skal tidsforskydningen udregnes ét sekund forud, så det rigtige tidspunkt kan nå FPGA'en før signalgeneratoren startes.

Derfor skal der udregnes antal sekunder fra midnat plus ét sekund. Dette multipliceret med sekundforskydningen for non-dropframe (-1 ms pr. sekund) eller dropframe ( $+33 \frac{1}{3}$  ns pr. sekund).

### 3.5.6 Bitpacking

Alle data skal pakkes ned i en "frame" på 80 bits. Data ligger lagret lidt forskelligt i de forskellige video-formater, men forskellen ligger kun i de bruger-definerede bits, der i dette projekt ikke benyttes. Derfor kan disse forskelle ignoreres.

Bits'ene er grupperet i binære grupper af 4-bits. Ved siden af disse grupper, kommer enkelte bits, der indeholder information, som angivet i tabellen nedenfor. Bits'ene er pakket i big endian format, hvilket vil sige, at LSB kommer først, og MSB sidst. Dette er omvendt alle data, der behandles i FPGA'en, og derfor skal al tidsinformation konverteres fra little endian til big endian.

I **tabel 3-4** ses hvordan en LTC dataframe er opbygget.

**Tabel 3-4:** Opbygning af LTC dataframe for PAL og NTSC

Bit nr:	Betydning
0-3	Frame enere
4-7	Binær gruppe 1
8-9	Frame tiere
10-11	Dropframe flag + colorframe flag
12-15	Binær gruppe 2
16-19	Sekund enere
20-23	Binær gruppe 3
24-26	Sekund tiere
27	Fase korrektion bit (PAL)/Binær gruppe setting (NTSC)
28-31	Binær gruppe 4
32-35	Minut enere
36-39	Binær gruppe 5
40-42	Minut tiere
43	Binær gruppe settings bit 0
44-47	Binær gruppe 6
48-51	Timer enere
42-45	Binær gruppe 7
46-47	Timer tiere
48-49	Binær gruppe setting bit 1+fase korrektion bit (PAL)/Binær gruppe settings bit 1+2 (NTSC)
50-53	Binær gruppe 8
54-79	Sync word

### **3.6 Indstillings og opsætningsmuligheder (tidszone, sommer/vintertid, videoformat)**

Fra mainframen kan vælges en række opsætningsmuligheder. Det er uden for rapportens afgrænsning at beskrive mainframens software, men det vil blive beskrevet, hvilke data MCU'en skal modtage, og hvordan indstillingerne bliver fulgt.

#### **3.6.1 Tidszone-indstilling**

Tidszonerne skal kunne indstilles fra UTC (GMT) tid +/- 30-mins spring. Der findes steder på jorden, hvor tidszonen er en halv time forskudt, og dette skal naturligvis også understøttes. Der skal altså sendes en værdi fra 0-46 fra mainframen, hvilket svarer til -11½ til + 11½ timers forskydning. 23 svarer til GMT, altså nul timers forskydning.

Når den aktuelle tid modtages ved opstart, skal tidszone-forskydningen lægges til, eller trækkes fra. Dette sker i UTC-tiden. Overskrides antallet af sekunder i ugen, skal ugenummeret ændres. Ugenummeret og tiden på ugen er den eneste måde at holde øje med skift til vintertid/sommertid, da datoen ikke direkte angiver dette.

#### **3.6.2 Sommertid/vintertids-indstilling**

Sommer-/vinter-tidsskift ligger forskelligt rundt omkring i Verden, men ikke alle lande benytter det. Derfor skal det kunne slås helt fra, eller til, hvor man samtidig sætter en dag i ugen, hvilken månede det skal være, samt tidspunkt på døgnet, hvor der skiftes frem/tilbage. Ved rapportens deadline er der dog ikke nået at blive designet en metode, til at implementere dette.

#### **3.6.3 Videoformat**

Video-formatet kan bestemmes imellem de tre valgmuligheder 24, 25 og 30 FPS. Desuden skal det bestemmes, hvorvidt LTC-generatoren skal benytte dropframes eller ej, ved valg af 30/1,001 FPS.

## 4. Praktisk afsnit

### 4.1 Hardwarebeskrivelse

Hardwaren består af et indstiksmodul til en mainframe. Dette indstiksmodul indeholder alle de nødvendige delkomponenter til at modtage GPS-signaler, og derudfra generere synkroniseringspulser og LTC-tidskode. I afsnittet beskrives hardwarens interface imellem de øvrige delkomponenter.

#### 4.1.1 FPGA

Interface til FPGA'en består af følgende:

- I2C-forbindelse til modtagelse af GPS-tid, UTC-tid og modtagelse af genlock og UTC indstillinger, samt læsning af status
- Output til VCXO kontrol
- Output til LTC tidskode og genlocksignaler
- Output til MCU - 1 PPS, GPS status, PLL låse-status og GPS-forbindelsesstatus
- Input fra GPS - 10 MHz clock, 1 PPS og GPS-forbindelsesstatus
- Diverse debugging LEDs og DIP-switches, samt genereal purpose forbindelser til MCU

FPGA'en er en Spartan 3E xc3s100E med 66 brugerdefinerede I/O-pins, fordelt på fire banker. Til FPGA'en er tilknyttet en 1 MBIT PROM til programlagring.

Styringen af VCXO'en sker direkte fra FPGA'en, dvs. en pin laver et pulse density moduleret (PDM) signal, der føres igennem et RC-led for at skabe et variabelt DC-niveau.

Derudover skal FPGA'en varetage tidskritiske opgaver, som ikke er mulige at udføre i en MCU. Dette består af generering af genlock-signaler og LTC-tidskode.

### 4.1.2 MCU

Interface til MCU består af følgende:

- RS232-forbindelse til GPS-modul (Opsætning af GPS og modtagelse af GPS data)
- I2C-forbindelse til mainframe (registrering og opsætning af modul, samt udlæsning af status)
- I2C-forbindelse til FPGA, som beskrevet i afsnittet *FPGA*

MCU'en er en C8051 microprocessor model F320 fra firmaet Silicon Laboratories, med en hardware UART, et I2C-interface samt tre 8-bits bidirektionelle porte.

MCU'en er hovedprocessor for styring af alle delkomponenter, og er mellemlid for mainframe, GPS-modul og FPGA. MCU'en håndterer opsætningsmuligheder, tidsindstillinger og kommunikation.

### 4.1.3 GPS

Interface til GPS:

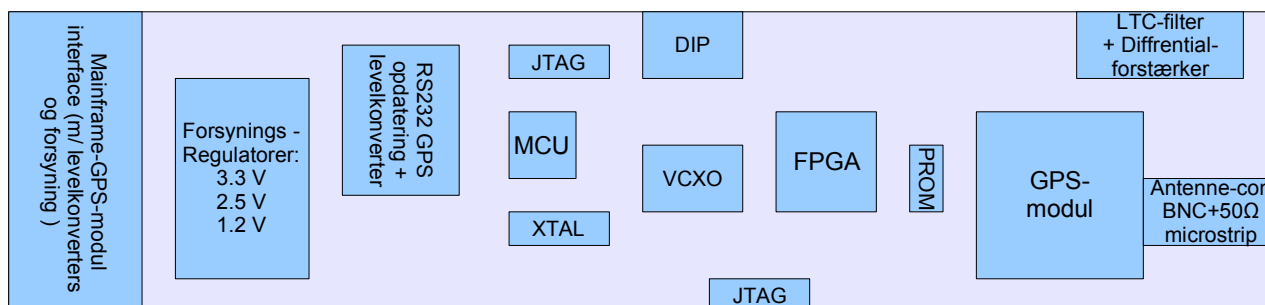
- RS232-forbindelse (sender/modtager ASCII data kodet i NMEA protokol)
- Satellit-status
- Tids-signaler (10 MHz og 1 PPS)
- Antenneindgang med phantom-power (3.3 V – impedanstilpasset 50 ohm)

GPS chipset'et er et NavSync TIM modul, der er specielt optimeret til timing-brug. Modulet kan sættes op til at kun at videresende ønsket data, dvs. at alt andet end tids-information og GPS-forbindelsesinfo slås fra. Endvidere leveres et 10 MHz og et 1 PPS signal, som er den reference der skal synkroniseres op til.

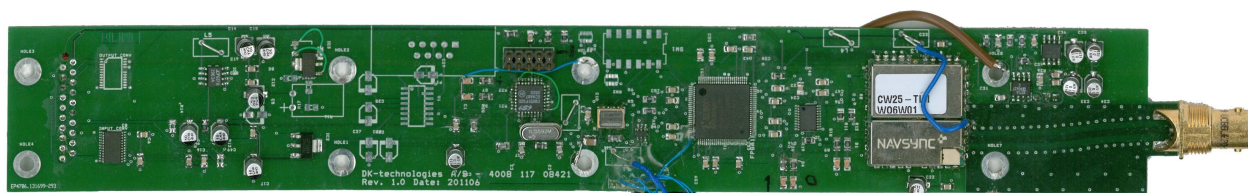
### 4.1.4 LTC

LTC-delen består af et 2. ordens lavpasfilter og en differential-forstærker. Indgangen til filteret kommer direkte fra en pin på FPGA'en og er altså logik-niveauer 0V-3.3V. Filteret begrænser frekvensindholdet i firkantpulserne til 15 KHz, og differentialforstærkeren giver et balanceret output.

## 4.1.5 Oversigt



**Figur 4-1:** Oversigt over hardware-modul



**Figur 4-2:** Foto af hardware-modul

## 4.2 Software og protokoller

### 4.2.1 Protokoller

Der benyttes tre forskellige protokoller i modulet, nemlig I2C og RS232, som er hardware protokoller til seriel transmission, og NMEA protokollen, som er en softwareprotokol til GPS-data. I2C vil blive implementeret i software, men vil også blive benyttet via en hardware-implementeret I2C-bus i MCU'en. RS232 er kun benyttet ved hardware-implementeringen i MCU'en og GPS'en, og vil ikke blive beskrevet yderligere.

### 4.2.2 I2C

I2C er en 2-wire serial-bus-standard udviklet af Philips. Standarden understøtter flere enheder på én bus, ved at benytte unikke adresser for hver enhed. Transmissioner kan håndteres på fire forskellige måder, slave-transmit, slave-recieve, master-transmit og master-recieve, men ikke alle enheder kan benytte alle metoder.

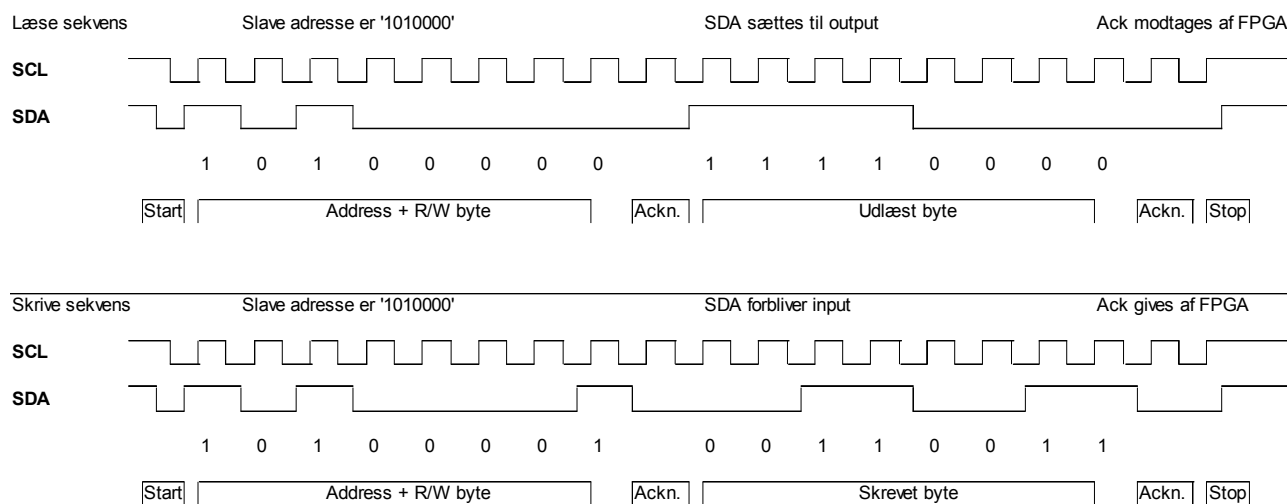
Der benyttes to linier til kommunikation kaldet SDA og SCL. SDA er data-linien og SCL er clock-linien. Begge linier er open-drain, og trækkes normalt højt via pull-up modstande. Benytter to separate enheder bussen på samme tid, vil begge enheder prøve at kommunikere indtil den ene taber forbindelsen, ved at en linie trækkes lav, mens enheden transmitterer et højt signal. Den udkonkurrerede enhed vil stoppe transmissionen, mens den anden enhed vil færdiggøre sin transmission.

En transmission påbegyndes ved et start-signal, og ender med et stop-signal. Et startsignal består af en nedadgående flanke på SDA, mens SCL er høj, og et stop-signal består af en opadgående flanke på SDA, mens SCL er høj. Disse to situationer er unikke, og kan ikke forekomme under en transmission, da SDA her ikke må ændres under en høj SCL.

Genererer masteren en start-betingelse, begynder slaven at lytte på forbindelsen. Masteren skal herefter sende en adresse til den respektive enhed, samt én bit, der angiver, om der ønskes at læses fra eller sendes til enheden. Adressen er syv bit lang, og sammen med læs/skriv-bit'en skal der altså sendes otte bit. Masteren sætter de enkelte bits på SDA, og giver en puls på SCL, så slaven indlæser bit'en. Der sendes i rækkefølgen MSB-LSB. For at bekræfte at slaven har modtaget data korrekt, holdes SDA lav, og der clockes én gang på SCL. Dette kaldes en acknowledge-bit.

Alt efter om der vælges at skrive eller læse, sættes SDA til input eller output fra FPGA'ens side, og der kan nu sendes eller modtages en byte. Metoden er den samme som når adresse-byten skal sendes. Masteren styrer clocken, og læser/skriver bitvist. Herefter skal der igen foretages et

acknowledge. Om det er masteren eller slaven der skal give et acknowledge, afhænger af om der læses eller skrives, respektivt. Herefter skal der gives en stop-betingelse, hvilket sætter slaven i vente-mode, indtil der igen skal kommunikeres.



**Figur 4-3:** Clockdiagram for I2C læsning og skrivning



### 4.2.3 NavSync ASCII protokol (NMEA)

NavSync GPS-modulet sender og modtager via en RS232-forbindelse. Modulet benytter protokollen NMEA (National Marine Electronics Association), som ofte er implementeret i GPS-moduler. Da det kun er timinginformation der skal bruges i genlock-modulet, vil kun disse beskeder blive beskrevet i detaljer. Desuden beskrives beskeder til opsætning af GPS-modulet.

NMEA protokollen er ASCII-formaterede beskeder, og kan altså direkte aflæses ved udskrivning til skærm eller printer, i modsætning til binære beskeder. I de første kommercielle GPS-moduler var det da også kun NMEA beskederne man fik udskrevet, til brug ved søfartsnavigation. For hver navigations-besked er defineret en ASCII-streng bestående af en header eller en identifikation, startende med et '\$', om hvilke data der følger. Herefter kommer de forskellige data, adskilt af kommaer. Til sidst afsluttes med en checksum, bestående af en 8-bits sum af alle ASCII-værdier mellem '\$' og selve checksummen.

Ulempen ved en ASCII-formatering er naturligvis at det skal parses, for at få en binær værdi, og at de forskellige NMEA beskeder kan indeholde forskellige typer data, så en standardparser er kompliceret at lave. Der benyttes eksempelvis bogstaverne 'n', 's', 'e' og 'w' nogle beskeder, for at angive retninger, mens værdier både kan være angivet som integers eller floats.

**Tabel 4-1:** Oversigt over NMEA beskeder i NavSync modulet

Forkortet	NMEA besked
GLL	GPGLL – Geographic position – Latitude longitude
RMC	GPRMC – Recommended Minimum Specific GNSS Sentence
VTG	GPVTG – Course Over Ground Speed
GGA	GPGGA - GPS Fix Data
GSA	GPGSA - GNSS DOPS and Active Satellites
GSV	GPGSV - GNSS Satellites in View
GRS	GPGRS - GNSS Range Residuals
GST	GPGST - GNSS Pseudorange Errors Statistics
PLT	POLYT - Time
PLP	POLYP - Position (Lat, Long)
PLU	POLYU - UTM position
PLG	POLYG - Local Grid position
PLS	POLYS - Satellite data
PLH	POLYH - HDS Time Information
PLI	POLYI - Additional Information

## 4.2.4 POLYT, timing information

Den besked, der bruges til at udlæse tid, er *POLYT*. Poly-beskeder er ikke-standardbeskeder, der følger NMEA protokollen, og kan indeholde chipset-specifikke informationer, som f.eks. timing-information, kvalitet af data og status for modulet.

POLYT udlæses som følgende:

*\$POLYT,hhmmss.ss,ddmmyy, UTC\_TOW ,week, GPS\_TOW ,Clk\_B , Clk\_D ,PG,cs*

hvor de enkelte data forstås som:

**Tabel 4-2:** Oversigt over POLYT besked

Navn	Beskrivelse
\$POLYT	NMEA-besked header
hhmmss.ss	UTC tid i timer, minutter, sekunder.brøkdels sekunder
ddmmyy	Dato i dag.måned.år
UTC_TOW	UTC tid på ugen i sekunder
week	GPS ugenummer
GPS_TOW	GPS tid på ugen
Clk_B	Clock bias (nanosekunder)
Clk_D	Clock drift (nanosekunder/sekund)
PG	1 PPS Granularitet (nanosekunder)
cs	Checksum

En enkel besked kunne altså se således ud:

*\$POLYT,123456.00,250299,123456.00,0978,123456.00,123456,123.456,28,cs*

## 4.2.5 POLYS, information om GPS-forbindelse

POLYS-beskeden indeholder en del information om satellite-data, som er irrelevant for projektet. Det vigtige er den sidste information i beskeden, der angiver om data er gyldigt. Et 'A' angiver at data er validt, mens et 'N' angiver at data er ugyldigt. Denne information er vigtig, for at kunne fastholde masterclocken ved svigt af GPS-signal.

#### 4.2.6 Opsætning af serielforbindelse

Ved opsætning af GPS-modulet sendes en streng med en NMEA besked. For at ændre en opsætning, sendes en besked med headeren \$PRTHS|R|Q, hvor 'S' er SET, 'Q' er QUERY og 'R' er REQUEST. Herefter sendes en header for hvilken indstilling, der skal ændres. Mulighederne er UxCM for UART konfiguration og UxOP for valg af NMEA-outputs. X står for hvilken port (1-3) der skal opsættes. Port 1 bliver brugt til navigations-data, mens port 2-3 kun bruges til debugging og sammenkædning af flere GPS moduler (også kaldet assisted GPS, hvor en hjælpestation assisterer en GPS modtager med dårligt signal).

For at opsætte en port sendes altså:

*\$PRTHS, UxCM,transmit\_baud, recieve\_baud,parity,stopbits*

Et eksempel kunne være:

*\$PRTHS,U1CM,57600,57600,N,1*

Den nuværende version af GPS-modulet kan kun sende 8-bit, en stopbit og ingen paritets-check.

#### 4.2.7 Valg af besked-output

For at ændre output-opsætning sendes en streng formateret på følgende måde:

*\$PRTHS,UxOP,(liste af beskeder, samt interval)*

Et eksempel kunne være:

*\$PRTHS,U1OP,ALL=0,PLT=1*

Her angives, at ingen beskeder skal sendes (ALL=0) bortset fra POLYT, som skal sendes hvert sekund. PLT er forkortelsen for POLYT jf. ovenstående tabel. ALL betyder alle beskeder. Ønskes en besked kun sendt f.eks. hvert tredje sekund sættes pågældne besked =3.

## 4.3 Software – MCU

### 4.3.1 Opstartssekvens

Ligesom FPGA'en, har MCU'en en række opgaver at udføre ved power-on. Det første er registrering til mainframen. Mainframen sender forespørgsler til alle monterings-pladser, om der er monteret et modul, og i så fald hvilket. Dette sker, ved at alle moduler har et unikt ID-nummer. Modtager et modul sit eget ID-nummer gives svar tilbage, og bekræfter at det er monteret. De forskellige pladser i mainframen har forskellige funktionaliteter, og GPS-modulet kan altså ikke monteres et vilkårligt sted. Pladsen hvor GPS-modulet er monteret er forbeholdt sync-moduler, og har et specielt interface, der kan modtage field og horisontal-pulser. Kommunikations-protokollen på denne plads er I2C. Metoden til at give svar tilbage, er ved at give en 'acknowledge', ved modtagelse af korrekt ID-nummer. 'Acknowledge' beskrives nærmere i afsnittet om I2C-protokollen.

Den næste opgave MCU'en har, er at opsætte GPS-chippen. Først og fremmest sættes, hvilke informationer der ønskes sendt fra GPS-modulet. Dette er timing-information, samt forbindelses-status, som beskrevet i afsnittet om NMEA-protokollen. Herefter sættes den frekvens, der skal benyttes til PLL'en. Reference-frekvensen der benyttes i PLL'en er 250 KHz, men frekvensen ud af GPS-modulet blev sat til 10 MHz, og siden nedskaleret til 250 KHz i FPGA'en, da det viste sig at være mere præcist, end hvis GPS-chippen blev sat til at levere 250 KHz direkte.

Når GPS'en er klar, ventes der til forbindelsen er etableret, og GPS-tiden kan læses ud. Dette tidspunkt skal bruges til at beregne, hvor i sync-genereringspulserne og LTC-koden der skal startes. Forskydningen beregnes og sendes til FPGA'en. Der skal beregnes ét sekund forud, da FPGA'en først skal modtage data, og være klar til at der gives tilladelse til at påbegynde signal-generering. Dette gives ved næste sekundpuls, hvilket vil sige det tidspunkt, der blev udregnet for.

### 4.3.2 Hoveddrift

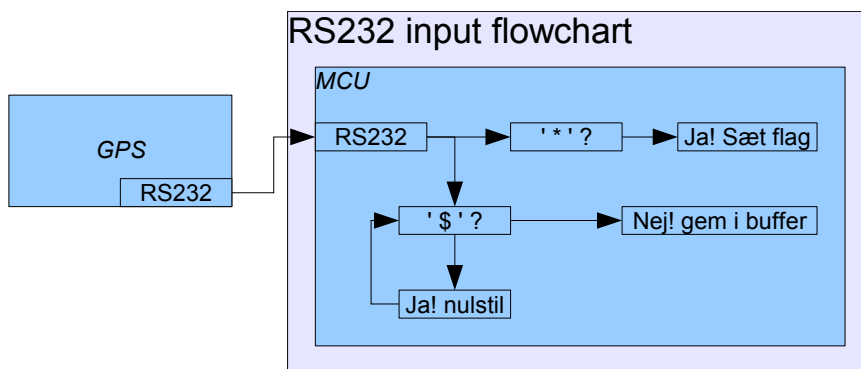
Når GPS-chippen er initialiseret, tidsdata er overført til FPGA'en og der er givet klarsignal, er MCU'ens opgave at overvåge input fra brugeren via mainframen. Disse input består af tidszoneopsætning, sommer/vintertidsopsætning og videoformatvalg. Ændres nogle af disse parametre resettes hele systemet, og en ny opstartsekvens initialiseres med den nye opsætning.

Foruden at modtage input fra brugeren, skal MCU'en hele tiden overvåge om der skal skiftes til eller fra sommertid. Dette gøres som beskrevet i afsnittet om "sommertid/vintertids-indstilling". Forskellen på denne ændring er, at hele systemet ikke skal resettes, men at tidspunktet skal skifte, uden at påvirke andre dele af systemet. Dvs. det er kun LTC-generatoren der ændrer output.

### 4.3.3 NMEA-protokol parser

Når GPS-modulet sender data, sker det via en RS232-forbindelse til MCU'en. MCU'ens seriellport er interrupt-drevet og indlæser automatisk til en buffer. Ved modtagelse af ASCII-værdien for '\$', resettes buffer-index til nul. '\$' betyder start på en NMEA-besked. Herefter læses hver karakter ind løbende i bufferen, indtil ASCII-værdien for '\*' modtages. '\*' angiver at hele beskeden er modtaget, og efterfølgende kommer checksummen (der dog ikke benyttes da RS232 linien er kort og regnes for pålidelig).

Nu ligger en hel NMEA-besked i bufferen. Et flag sættes højt, som tegn til MCU'en at bufferen skal kopieres og parses. Kopieringen til en ny buffer, er for at en ny NMEA-besked kan indlæses i bufferen, mens den gamle besked bliver parset. På **figur 4-4** ses et flowchart for RS232 indlæsning.



**Figur 4-4:** RS232 input flowchart

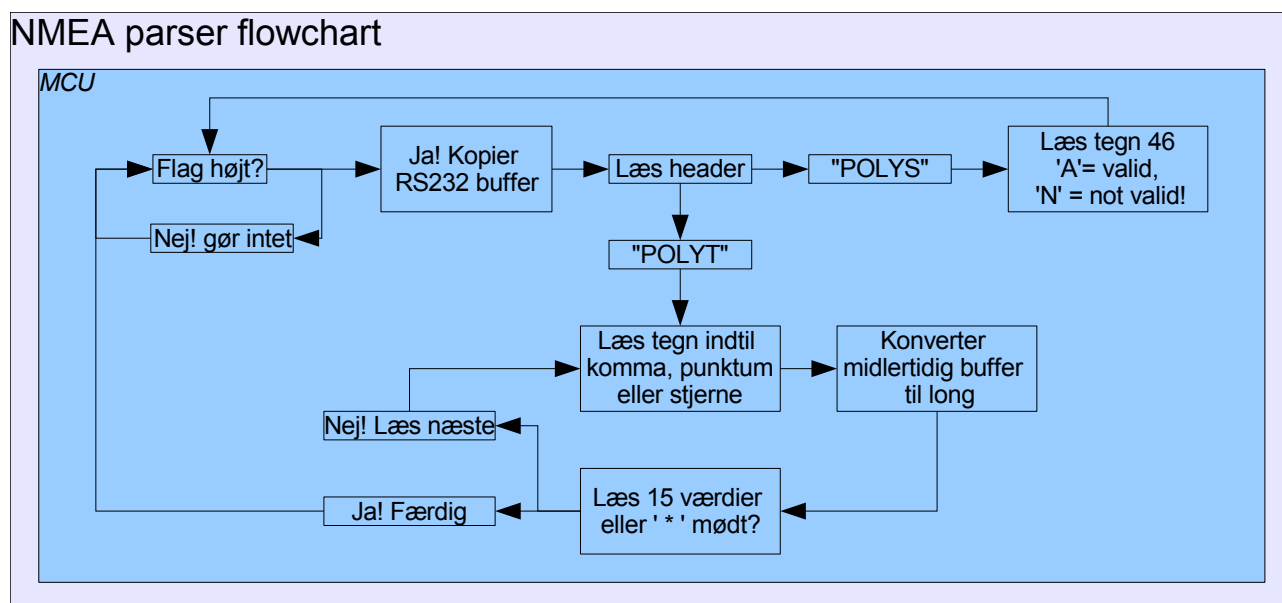
Parseren er ikke en standard NMEA-parser, forstået på den måde, at den kun kan fortolke POLYS og POLYT beskeder. Hver NMEA-besked er forskellig i længde og data-type opbygning, så hver besked skal parses forskelligt. I dette projekt er det overflødigt, da kun de to førnævnte beskeder benyttes.

Det først der skal kontrolleres er ID-headeren. En simpel streng-sammenligning af de fem første karakterer efter '\$', kan hurtigt afgøre om det er en POLYS eller POLYT besked, og herefter starte parsing af den respektive besked.

POLYS beskeder indeholder data om GPS-forbindelsen. Det meste information er overflødigt for projektets vedkommende, det eneste der behøves at udlæses er hvorvidt der er forbindelse til det nødvendige antal satelliter. Dette kan undersøges ved at læse den sidste værdi i beskeden (se bilag for NMEA beskeder). Et 'A' her angiver, at forbindelsen er etableret, og timing-data der udlæses er validt. Et 'N' angiver ugyldigt data. Parsing af denne type beskeder kræver altså kun en simpel sammenligning af en karakter, der fast ligger på plads 46 i strengen.

POLYT beskeder kræver en del mere af parseren. Efter ID-headeren følger en mængde information der skal bruges til projektet. Derfor vil alle informationer i denne beskedtype blive udlæst. Metoden er simpel. Hver information er adskilt af et komma eller punktum. Beskeden afsluttes med '\*'. Efter headeren læses alle karakterer til en midlertidig buffer. Når et komma, punktum eller '\*', stoppes indlæsningen til den midlertidige buffer. Den midlertidige buffer indeholder nu en hel information, der kan konverteres til en binær værdi. Denne værdi gemmes i en buffer, og de næste informationer indlæses på samme måde.

Nu ligger alle timing-informationer binært i et array, men de er stadig ikke helt færdig-parsede. UTC-tiden ligger ikke gemt som timer, minutter og sekunder, men i én værdi, der eksempelvis kunne se således ud "123506". Denne værdi skal deles ud i tre separate værdier. Dette gøres nemt, ved at tage modulus 100 til værdien, for at udlæse sekunder. Herefter divideres med 100, og værdien "1235" fås. Igen tages modulus 100, for at udlæse minutter. Til sidst divideres igen med 100, og timeantallet "12" fås.



**Figur 4-5:** NMEA parser flowchar

**Figur 4-5** skal ses i sammenhæng med **figur 4-4** på side 41. "Flag højt" hentyder til det flag, der sættes når en hel NMEA-besked er indlæst.

## 4.4 Software – FPGA

FPGA'ens hovedkerne er opbygget som en statemachine for selve driften af modulfunktioner, såsom start-up sekvens, stabil drift og ustabil drift. Derudover er der nogle underkomponenter, der skal være selvstyrende. Dette indbefatter I2C-kommunikation, clock-kompensering og PLL.

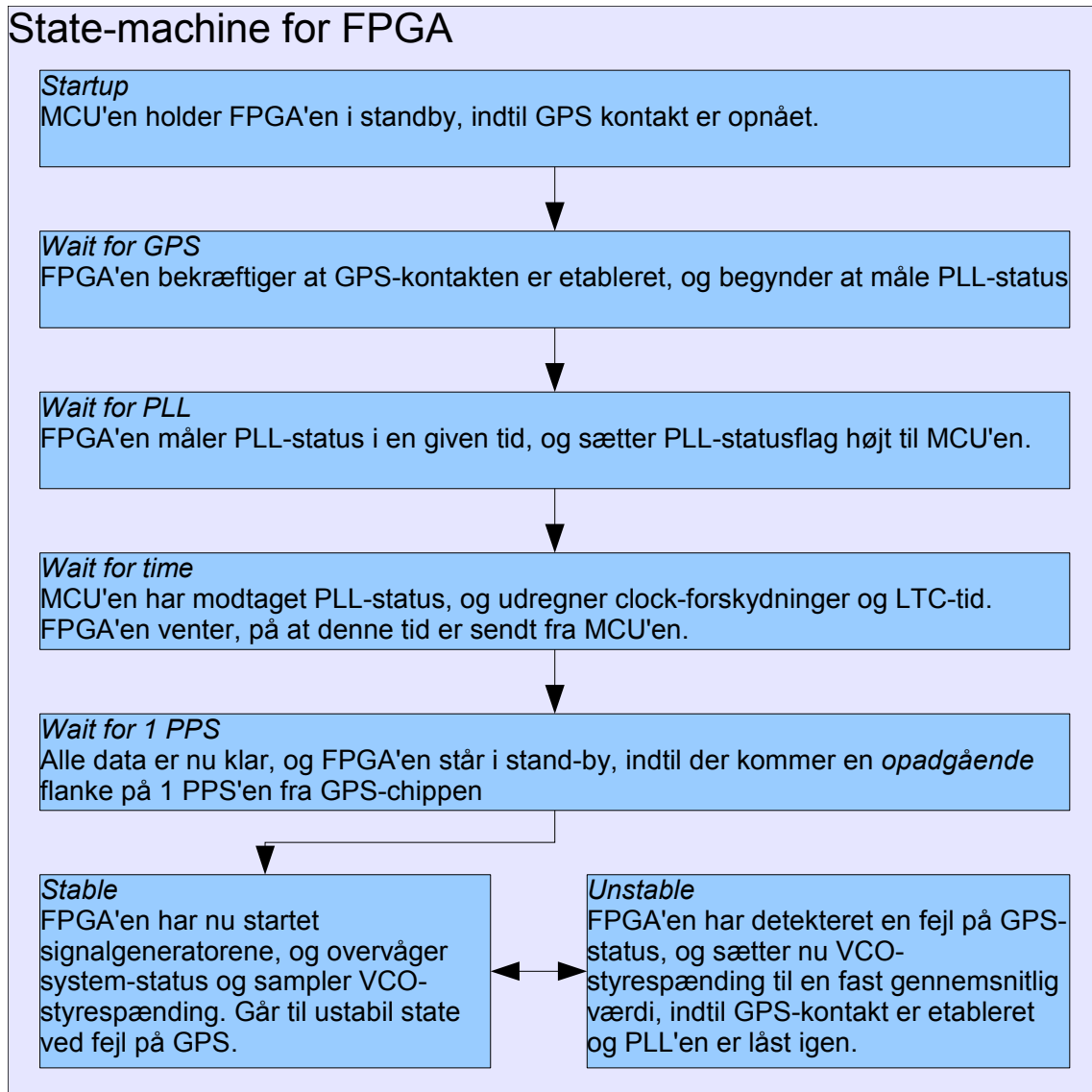
### 4.4.1 Hovedkerne

Hovedkernen styrer al drift af signalgeneratorer og håndtering af de situationer der kan forekomme ved drift. Hovedkernen bliver holdt ude af drift, ved et masterreset signal, styret af MCU'en. Først når MCU'en er bootet op og klar, gives der signal til FPGA'ens hoved state-machine, om at starte op. Opstarten følger diagrammet, der kan ses i afsnittet ”Overordnet MCU og FPGA programstruktur”. På **figur 4-6** ses mere specifikt, hvad der sker i hver state ved opstartssekvens og herefter ved hoveddrift.

Ved siden af hoved state-machinen, varetager FPGA'en også andre opgaver, der er uafhængige af de forskellige states. En af disse opgaver er I2C-kommunikation med MCU'en. Ved start-up resettes alle timing-værdier, og der skal sendes nye værdier fra MCU'en. Disse værdier skal alle sendes i en specifik rækkefølge, for at FPGA'en tolker dem korrekt. Herefter kan der kun sendes værdier for nyt klokkeslæt, indtil næste master-reset. Klokkeslættet skal kunne sendes, for at ændre sommertid/vintertid.

Herudover kommer PLL'ens drift, som hovedsageligt er selvstyrende. Der skal dog kunne skiftes mellem fast styrespænding til VCO'en, eller en selvregulerende styrespænding via faseloop'et. Dette skift foretages ved skift af states mellem stabil og ustabil. PLL'en beskrives nærmere i afsnittet ”PLL”.

## State-machine for FPGA



Figur 4-6: FPGA opstartssekvens



## 4.4.2 I2C-bus

I2C-bussen består af en underkomponent, med de fornødne ind/ud-porte:

Ind:

- SCL (master clock)
- SDA (data linie)
- Sendbyte (udgående byte)
- Fast adresse (7 bits)

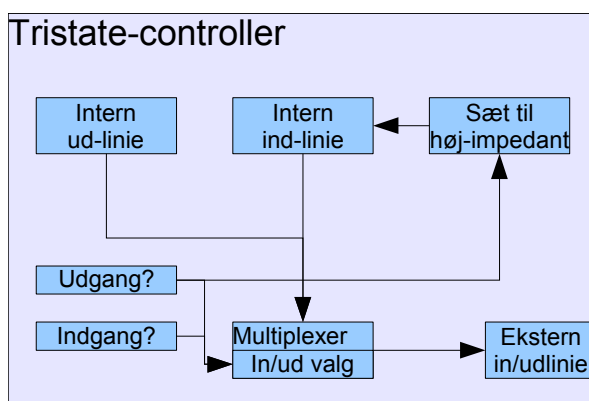
Ud:

- SDA (data linie)
- Recievedbyte (modtaget byte)

Dertil kommer master-clock og reset indgange.

Ved at opbygge I2C-bussen som en underkomponent, kan der hurtigt laves flere busser, med hver deres unikke adresse. På denne måde kan der eksempelvis laves én bus til at modtage tids-information og én bus til at modtage setup-information. På denne måde kan indgående data nemt sorteres. Dog kræver det en del mere logikplads i FPGA'en. På **figur 4-8** ses et flowchart for I2C-motoren.

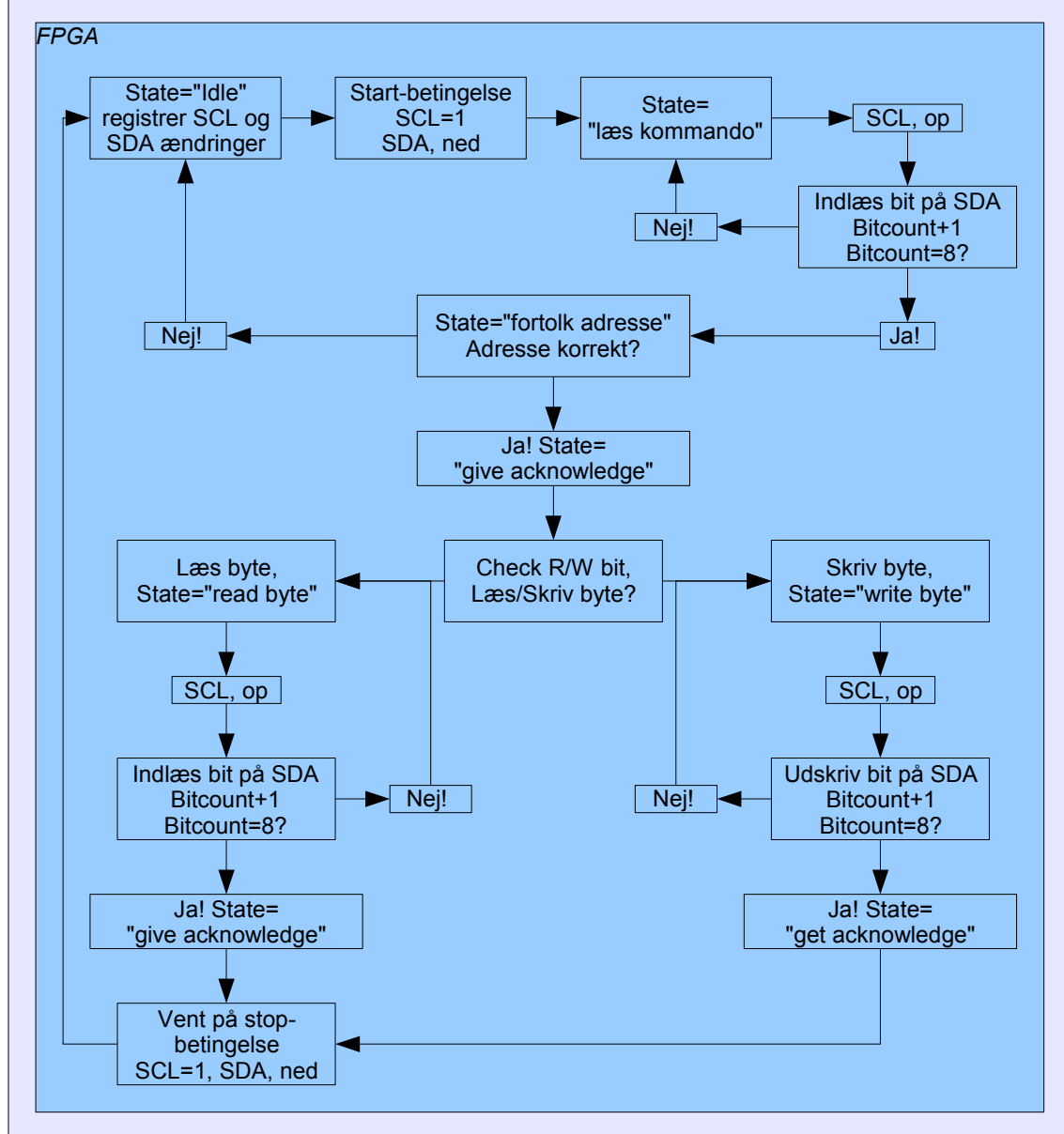
SDA-linien skal kunne fungere som en bidirektionel port. Dvs. der skal kunne læses og skrives på SDA-pin'en. I2C dikterer at begge linier er open-drain, med en pull-up modstand. Dette betyder at linien normalt er høj, og aktivt skal trækkes lav. For at implementere dette i VHDL, kræves det at man laver en tristate-bus.



**Figur 4-7:** Opbygning af tri-state controller

En intern ind-linie og ud-linie routes til samme IO-pin på FPGA'en. For at undgå at konflikter, styrer et flag, om porten er input eller output. Flaget sætter ind-linien til højimpedant i det tilfælde, at der skal læses på linien.

## I2C flowchart

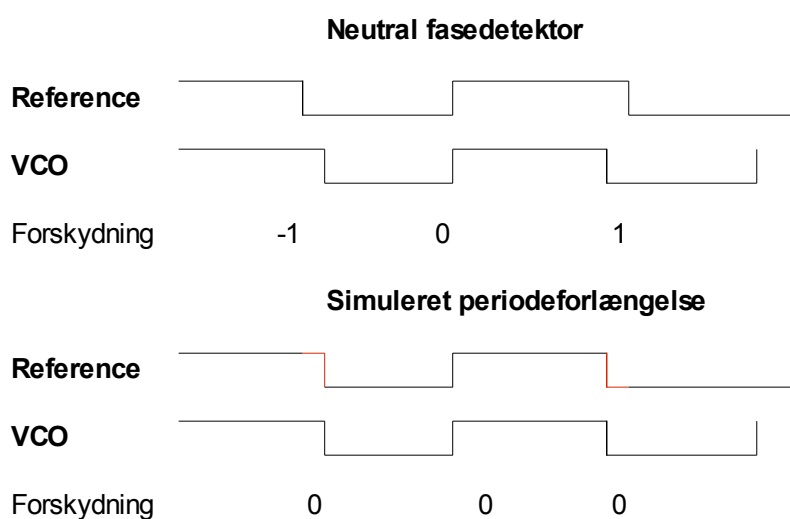
**Figur 4-8:** Flowdiagram for I2C-statemachine

Hvad der ikke fremgår af flowchartet er, at der kører en timer uafbrudt. Opfyldes en betingelse, så flowet følges, resettes timeren. Går der imidlertid for lang tid, før en betingelse opfyldes, bliver der time-out, og der springes til "idle" staten. Time-out er ifølge Philips normen 25 ms.

### 4.4.3 PLL

PLL'en er en underkomponent, der er benyttet i andre DK-Technologies produkter. Komponenten er altså ikke designet i dette projekt, kun modificeret til at kunne udføre de ekstra funktioner der er påkrævet. Derfor vil selve implementeringsmetoden kun blive beskrevet overfladisk. For et diagram over PLL'en, se **figur 3-3**.

En fasedetektor måler forskellen mellem reference- og VCO-frekvensen. Denne fejl justerer VCO'en ind, til den er i fase med referencen. Ved start kan de to frekvenser være langt fra hinanden frekvensmæssigt, og fasefejlen vil veksle hurtigt mellem at være positiv og negativ. En overgang fra negativ til positiv ses på **figur 4-9** øverst. Derfor vil regulatoren tilsvarende prøve at skrue op og ned for frekvensen tilsvarende. Dette giver en ueffektiv stabilisering og langsomt indsvingningsforløb. Derfor er fasedetektoren lavet, så den lige før (og *kun* lige før) et faseskift fra positiv til negativ forskydning eller omvendt, simulerer en længere periode i referencen, så faseskiftet udebliver. Dette ses nederst på **figur 4-9**. Derved vil styrespændingen ikke veksle hurtigt fra mellem at skrue ned og op skiftesvis, men holde ved at skrue ned eller op i længere tid. Dette giver en mere stabil regulering. Dette tilsvarende en frekvensdetektor, der først tilpasser frekvensen, hvorefter fasedetektoren tilpasser fasen.



**Figur 4-9:** Simuleret periodeforlængelse

Modifikationerne til GPS-modulet består i, at kunne påvirke PLL'en eksternt, til at køre hurtigere eller langsommere. Dette sker ved, at påvirke PLL'en, så den, ligesom ved fasedetektoren, simulerer en kortere eller længere periode i referencen, og derved skruer op eller ned.

Desuden skal styrespændingen midles over en periode, så denne værdi kan benyttes som fast styrespændingen ved GPS-svigt. Den midlede værdi skal være fra tider 10 sekunder før spændingen fastholdes, da nye værdier kan indeholde fejldata, der er målt før et GPS-svigt er detekteret.

## 5. Test og evaluering af GPS genlock og LTC kode

For at teste produktet, skal funktionaliteten afprøves, såvel som kvaliteten af det genererede signal. Dog kan produktet ikke testes som et endeligt produkt, da ikke alle funktionaliteter er fuldt implementerede.

### 5.1 Funktionalitetstest

Funktionalitetstesten indebefatter først og fremmest, at modulet kan generere LTC og genlock signaler. For at teste dette, indsættes modulet i mainramen. Modulet vil ved korrekt opstart, automatisk påbegynde generering af begge signaler. Hvis genlocksignalerne genereres korrekt, vil dette blive indikeret på mainramens frontpanel. Hvis signalet er direkte fejlagtigt, vil mainramen gå ud af genlocken, og en rød fejl-LED vil lyse. Er signalet derimod korrekt, men af dårlig kvalitet, vil meddelelsen ”unstable signal at board 3” blive vist på frontpanelet. GPS-modulet kunne ved denne test levere korrekt signal til mainramen uden warnings eller fejl. Ved at påtvinge jitter i PLL'en, mistede mainramen kortvarigt genlock, men blev hurtigt stabiliseret.

LTC-generatoren kunne testes, ved at tilkoble udgangssignalet fra modulet, til en LTC-dekoder. Dekoderen der blev benyttet var en MSD 600 (Master Stereo Display) fra DK-Technologies selv. LTC-koden blev bifasemoduleret, sendt til MSD'en og tiden blev vist korrekt på displayet. Ved projektets afslutning var der dog lidt fejltælling på frame-tælleren.

En sidste funktionalitetstest var opstart af modulet. Startsekvensen skulle eksekveres korrekt, for at modulet virkede. Dvs. at FPGA'ens statemachine og MCU'en skulle arbejde korrekt sammen, og give de rigtige signaler, for at fortsætte indtil start af hoveddrift. Ved at benytte MCU'ens debugværktøj, kunne denne opstart følges skridt for skridt, og blev verificeret til at forløbe korrekt.

### 5.2 Kvalitetstest (*Jitter/præcision af signaler, stabilitet/langtidstest*)

En af kvalitetstestene skulle være foretaget med et Tektronics Video-oscilloskop, men ved projektets afslutning var modulet ikke klar nok til at få foretaget denne test. Testen skulle måle den totale jitter fra GPS-chip igennem modulet og til det endelige udgangssignal fra mainramen. Dog kunne mængden af jitter anses for at være acceptabel lille, da mainramen ikke kom med advarsler, som beskrevet i forrige afsnit.

Ved benytte clock-kompensatoren beskrevet i afsnittet ”stabilitet af GPS-clock” blev fejl, akkumuleret over længere tid, elimineret. Denne type fejl vil der derfor kunne ses bort fra. Var clock-kompensatoren derimod slået fra, ville genlock-signalet stadig være af så god kvalitet, at mainramen ville kunne være låst. Over længere tid ville genlock og LTC perioden dog forskydes, hvilket ville bevirke, at flere adskilte GPS-genlocksyste mer ville være ude af sync.

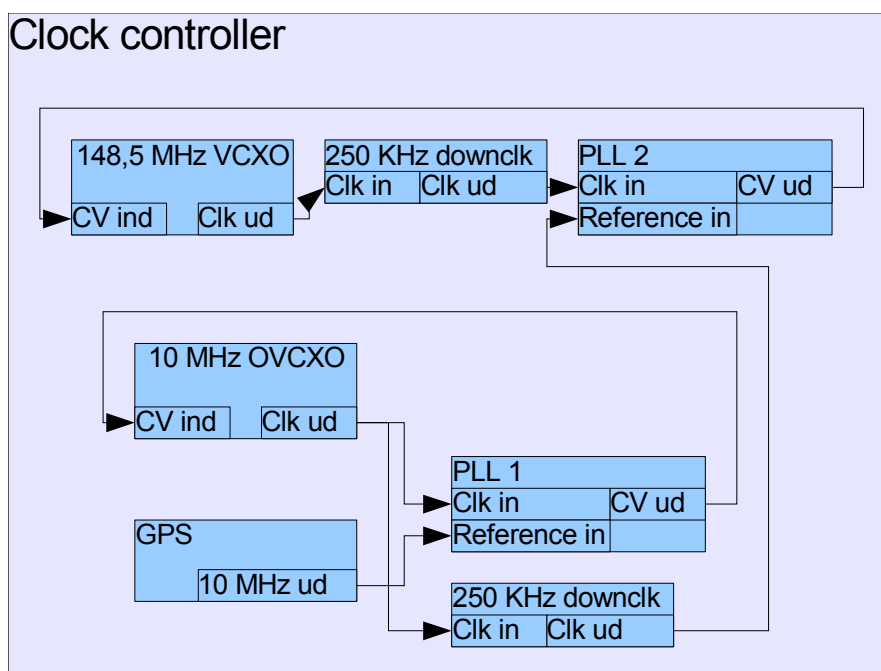
## 6. Videre forbedringer og optimeringsmuligheder

Det nuværende modul virker som det skal, men der kan foretages yderligere forbedringer, som det ikke har været muligt at nå i praktikforløbet. I dette afsnit findes eksempler på nogle videreudviklinger og forbedringer, der vil kunne øge kvaliteten af det endelige produkt.

### 6.1 Montering af oven VCXO (OCXO)

Mistes forbindelsen til GPS-satellitterne i længere tid, er det vigtigt at bevare masterclocken stabil. Dette er muligt i den nuværende hardware, men forbedringer kan godt laves, så clocken påvirkes endnu mindre. Den nuværende VCXO på 148,5 MHz har en forholdsvis høj påvirkningsfaktor på 90 PPM/Volt i hele styringsspændingsområdet. Dette gør den istand til hurtigt at kunne låses til en reference-clock, men gør den også mere støjfølsom, da et lille udsving i styrespændingen, tilsvarende et større udsving i frekvensen.

Ved at montere en oven-oscillator (OCXO), som en ekstra oscillator på modulet, kan der opnås større præcision. OCXO'en kan fås i en standardstørrelse på 10 MHz, tilsvarende GPS'ens frekvens. OCXO'en låses fast til GPS'en gennem en PLL, og 148,5 MHz VCXO'en låses til OCXO'en. OCXO'ens PLL skal udglatte jitter fra GPS'en, ved at den er låst igennem et langsomt loopfilter. Ved at benytte en OCXO med meget lav påvirkningsfaktor (f.eks. 2 PPM/Volt), fås en mindre støjfølsom, og dermed jitterfri masterclock, end GPS'en kan levere. Desuden sørger ovenen for, at stabiliteten holdes over længere tid, ved eventuelle GPS-svigt.



**Figur 5-1:** OCXO implementerings diagram

## 6.2 Montering af ekstern præcisions DAC

Den nuværende DAC er en 1-bit sigma-delta modulator med output direkte fra en pin på FPGA'en. I det system er der mange støjkluder, der kan forringe masterclockens performance.

Forsyningsspændingen kommer fra en regulator, der trækker en del andre komponenter, end FPGA'en, og regulatoren er placeret langt fra FPGA'en. Det ideelle ville være en DAC, der havde sin egen lokale regulator, der skulle være meget stabil, så referencespændingen blev så optimal som muligt.

En anden forbedring ville være en anden DAC-type. Sigma-delta modulatoren virker ved at et højfrekvent puls-tæthedssignal filtreres igennem et RC-led, til en DC-spænding. Dette er selvsagt ikke den mindst støjfri måde at lave en DC-spænding på. En mere optimal type DAC, ville være et resistor-netværk, med vægtede modstande til en summator. Denne type DAC ville kunne modtage en værdi til styrespænding, påtvinge værdien til VCXO'en og fastholde den i et ønsket tidsrum. Herefter behøver FPGA'en end ikke kommunikere med DAC'en. Hvis denne type DAC sad lige ved VCXO'en, med sin egen lokale regulator, ville man få en langt mere støjfri løsning, end i den nuværende implementation.

## 7. Konklusion

GPS-modulet blev færdigudviklet til et niveau, hvor det var muligt for mainframen at låse til de synkroniseringspulser, der blev sendt fra GPS-modulet. LTC-tidskoden kunne også fortolkes af en LTC-dekoder. Kildekoden til MCU og FPGA findes i bilag.

### 7.1 Ufærdige dele

Der blev i projektet arbejdet på at få så færdigt et produkt som muligt. I problemformuleringen er en del dog afgrænset, til et realistisk omfang for projektet. Andre dele vil blive implementeret efterfølgende.

Dog var der en del punkter, der gerne skulle være blevet implementeret, men som ikke nåede med inden afleveringsdato på rapporten. Disse omfatter sommer/vintertids-indstilling, som kun er blevet beskrevet som en ønsket feature, men som ikke nåede at blive designet. Desuden er kildekoden til MCU og FPGA stadig på udviklingsstadiet, og mangler derfor en del funktionaliteter, for at fungere som et færdiggjort produkt. Dette inkludere bl.a.:

- Midling af DAC værdi til VCXO.
- Endelig kommunikation til opsætning af videoformat på FPGA.

### 7.2 Løsning

Følgende kommer gennemgang af de problemstillinger der blev opstillet ved projektstart.

#### Hvordan låses en masterclock bedst muligt op mod GPS-frekvensen?

GPS-chipsettet kan levere en variable clockfrekvens (op til 10 MHz), som er synkron med GPS-satelliternes atomure. Ved at benytte en PLL, kan en hurtigere masterclock låses til denne GPS-clock. Ved at gøre dette på hvert modul, vil alle moduler være låst til samme GPS-clock, og dermed indbyrdes synkron, uden at have fysisk forbindelse. En hurtigere clock benyttes, fordi 10 MHz er for langsomt, at benytte som masterclock til generering af genlocksignaler og LTC kode, og her bruges en VCXO på 148,5 MHz..

#### Hvordan bruges GPS-tid og frekvens til synkronisering og LTC-kodning?

Som nævnt i forrige spørgsmål, låses en hurtigere masterclock til GPS'ens 10 MHz clock. Ydermere leverer GPS-chippen et én puls per sekund signal, der kan benyttes til at tilpasse signalgeneratorene til GPS-tid. GPS tid adskiller sig fra UTC tid, ved at der måles i antal uger, og sekunder i ugen. Denne tid startede d. 6. januar 1980 kl 00:00. Dette tidspunkt anses som værende det absolutte tidsnulpunkt, og ved at bruge dette tidspunkt som start for alle signalgeneratorer, vil en vilkårlig generator altid levere samme signal, som en anden generator.

### **Hvordan kommunikeres der på tværs af hardware-moduler (MCU, FPGA og GPS)?**

Den benyttede GPS-chip har fastlagt sin protokol, idet der kun er RS232 til rådighed. MCU'en har allerede en hardware UART til rådighed, og derfor benyttes denne til at kommunikere med GPS-chippen. GPS-chippen sender og modtager kun data ved NMEA-protokollen, som er en ASCII-protokol, udviklet til maritim GPS-navigering. Derfor skal hver besked parses til binær data.

Mainframen har også fastlagt sin protokol, idet den benytter I2C, og derfor forbindes MCU'ens hardware I2C-bus til mainframen. Hardware-bussen benyttes, da denne er interrupt-drevet og derfor kan modtage data på vilkårlige tidspunkter, uanset hvor i programmet den er. En software I2C-bus programmeres til at skulle kommunikere med FPGA'en. Denne skal kunne fungere således, at MCU'en er master, og FPGA'en slave, og det er derfor ikke nødvendigt med en interrupt-drevet bus på MCU'en, da denne bestemmer hvornår der læses eller skrives.

Ydermere benyttes direkte forbindelser til vigtige signaler, hvorpå der skal reageres hurtigt. Dette er GPS-forbindelsesstatus fra MCU til FPGA, og PLL-status fra FPGA til MCU.

### **Hvad sker der ved GPS-svigt og forstyrrelser, og hvordan afværges dette?**

Idet at kontakten til GPS-satellitterne mistes, skal systemet forblive stabilt. MCU'en aflæser hele tiden status fra GPS-chippen, og i det tilfælde at GPS-chippen fejler, skal der sendes signal til FPGA'en, om at fastholde VCXO'en. Dette sker ved at give en fast styrespænding. Styrespændingen er en midlet værdi af de forgående styrespændinger. Dog er de sidste ti sekunders styrespændinger ikke indeholdt i denne midlede værdi, da disse allerede kan være fejlbehæftede, da der ofte går et stykke tid, fra at GPS-chippen fejler, til at den sender besked ud herom, og inden MCU'en giver beskeden videre til FPGA'en.

Ved mindre forstyrrelser i GPS-signalet opstår der ikke deciderede fejl, men mindre forstyrrelser i 10 MHz clocken kan forekomme løbende. Da alle genererede signaler skabes fra denne clock, vil forstyrrelserne over længere tid akkumuleres til en større fejl. Denne akkumulering vil oftest udligne sig selv over længere tid. Dette sker dog ikke altid, og derfor skal der implementeres et modul, der sørger for at disse fejl udlignes. Dette kan gøres ved at sammenligne en 1 PPS genereret ud fra masterclocken med GPS'ens 1 PPS. Forskydelser der ikke selv udlignes efter længere tid, udlignes manuelt, ved at foretage små justeringer i PLL'en



## Appendiks A: Hardware design overvejelser

I afsnittet er kort beskrevet hvilke overvejelser der er gjort ved opbygning af hardwaren i modulet. Projektet skal ses som en videreførelse af forgående praktikforløb, hvor modulets hardware blev udviklet. Hardwaren vil derfor ikke blive beskrevet i detaljer, men kort blive gennemgået, for at få fuld forståelse for projektet.

### Valg af enheder og komponenter

GPS-chippen der benyttes, kræves at være stabil og kunne levere en yderst præcis clock og relevant timing-data. Ydermere er størrelse, pris, tilgængelighed og dokumentation også vigtige faktorer. Sidstnævnte parametre er vigtige i en produktion, da chipsene skal være tilgængelige fremover, i det krævede styktal.

Der blev undersøgt, hvilke muligheder der var på markedet, og valget faldt på moduler fra uBlox eller fra NavSync. Der blev foretaget nogle test med de to moduler, for at finde den bedste løsning til modulet. Modulet fra uBlox var det bedst dokumenterede, og virkede generelt mere pålideligt, men kunne ikke levere mere end et 10 Hz output til synkronisering. Denne lave frekvens ville ikke være optimal at låse PLL'en med, da den vil være for langsom og følsom overfor støj.

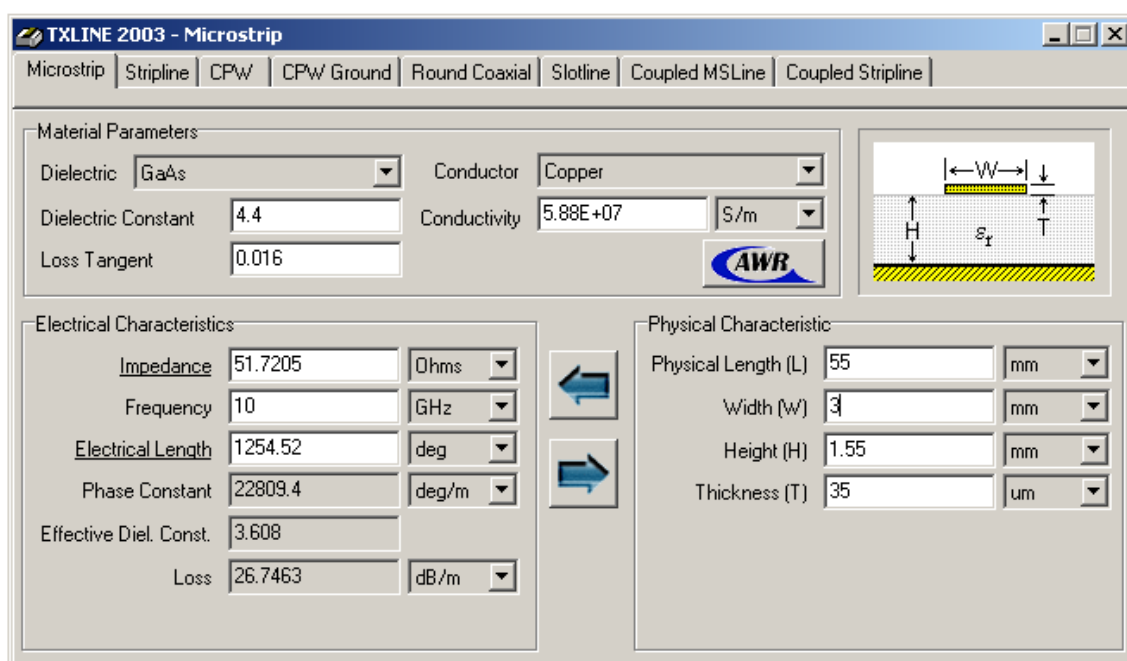
Modulet fra NavSync kunne derimod levere et 10 MHz signal, hvilket ville give en bedre performance i PLL'en. Dette modul kunne dog ikke sende data binært, men kun via en ASCII-protokol (NMEA). Desuden var ikke alle besked-typer implementeret endnu, og chippen var ikke nær så veldokumenteret som uBlox-chippen. Dog var alle informationer, der var krævet af projektet til rådighed, og da prisen var lavere end uBlox-chippen, blev NavSync-chippen valgt.

FPGA'en der blev valgt, skulle ikke opfylde de store krav. Projektet kunne anses for at være et mindre projekt, og der blev derfor fundet en FPGA fra Xilinx, Spartan 3E xc3s100E, med 1 MBit kapacitet og 66 I/O-porte, som var tilstrækkeligt.

Der blev valgt en MCU fra Silicon Laboratories til at håndtere kommunikation fra GPS-modul, mainframe og til FPGA'en. MCU'en er en C8051 processor, med én hardware UART, som kunne benyttes til kommunikation med GPS-modul og én hardware I2C-bus, som kunne benyttes til kommunikation med PT5300 mainframen. Kommunikationen med FPGA'en består af en I2C bus implementeret i software. MCU'en har desuden tre IO-porte på hver 8 bits bredde.

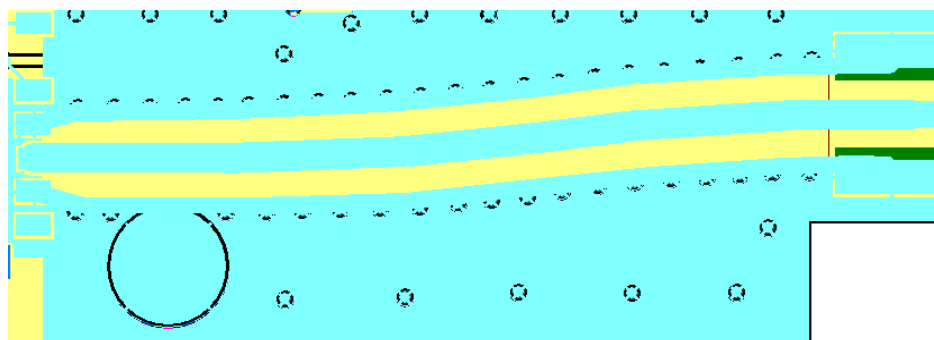
## Design af micro-strip til antenneforbindelse

Antenne-signalvejen er impedans-tilpasset til 50 ohm. Dette er kravet fra GPS-chipsettes side, og outputtet fra den aktive GPS-antenne. Det er vigtigt at signalvejen er så støjfri som muligt. Derfor er micro-strippen omgivet af et skærmende jord-lag. Dette jord-lag vil virke kapacitivt, og derfor er bredden af micro-strippen, samt afstand til jord væsentligt. Ved at benytte et lille stykke software, kan impedansen af en sådan microstrip hurtigt beregnes. Afstanden fra strippen til jordlaget bestemmes af PCB-tykkelsen. Der benyttes et standard FR4 4-lags print, med en tykkelse på 1.55 mm. Disse data indtastes i softwaren, og impedancen udregnes. Ved at optimere parametrene for micro-strippen opnås nemt en impedans på 50 ohm.



**Figur A-1:** Microstrip design software

På **figur A-2** ses hvordan den endelige microstrip ser ud på PCB'et.

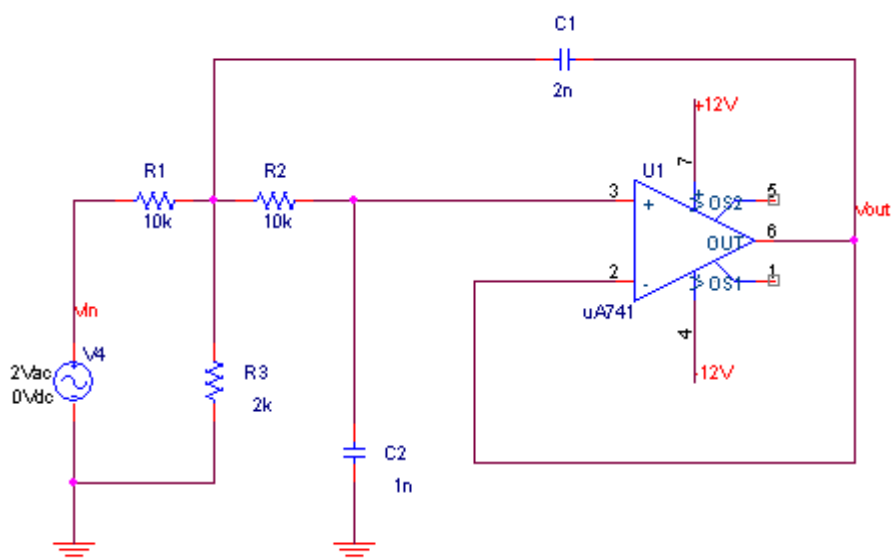


**Figur A-2:** Microstrip på PCB

Microstrippen er den brede bane. Længden og bredden er beregnet til 55 mm og 3 mm henholdsvis, hvilket giver 51,7 ohm.

## LTC-filter design

Filteret er defineret til et 2. ordens lavpasfilter ifølge SMPTE normen. Frekvenskarakteristikken ligger i det hørbare område fra 20Hz – 15KHz, således at signalet kan lagres på audio-medier. Filteret designes som et aktivt Sallen-Key filter, som det ses af **figur A-3**.



**Figur A-3:** Topologi for Sallen-Key filter

Metoden for at beregne knæfrekvensen er som følger:

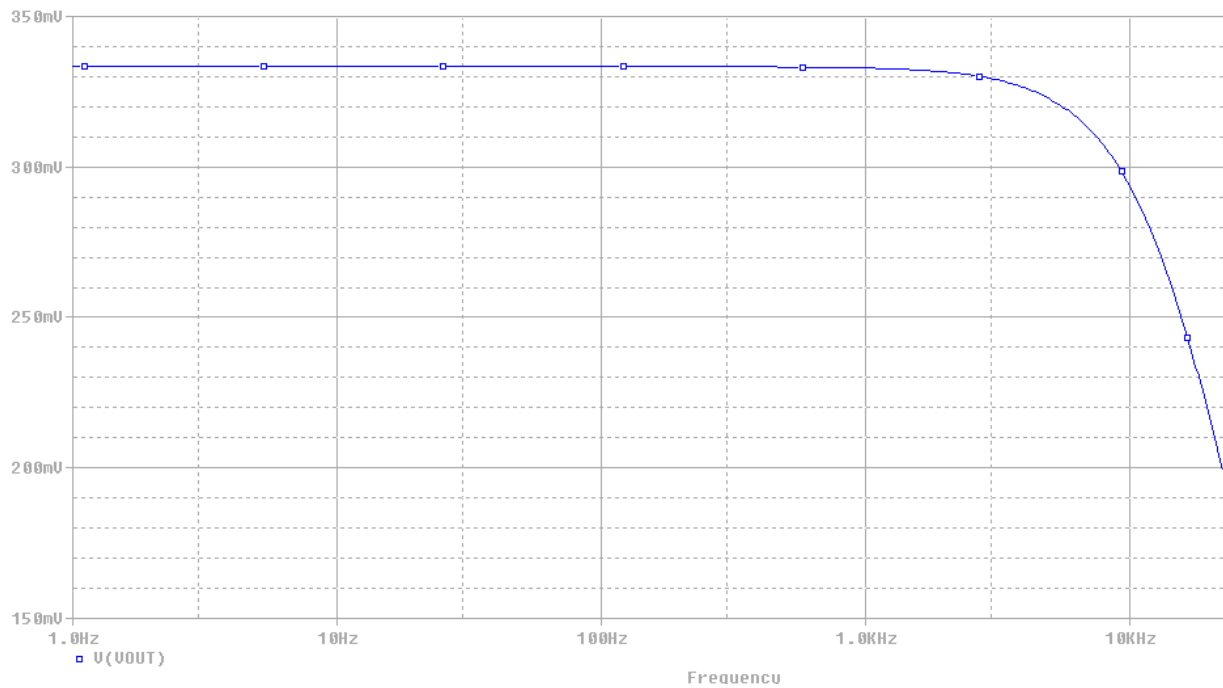
- Sæt  $C1 = \text{vilkaarlig størrelse}$ , sæt  $C2 = \frac{1}{2} \times C1$
- Sæt  $R1 = R2 = 0.707 / (2 \times \pi \times F_c \times C2)$

**Tabel A-1:** Funde LTC filter komponent-værdier

Komponent	C1	C2	R1	R2
Værdi	2 nF	1 nF	10K	10K

Dette giver en knæfrekvens på  $11252,58 \text{ Hz}$ .

Da der efterfølgende benyttes en differential-forstærker med et gain på 6 dB, skal indgangssignalet dæmpes. Det ønskes at output-spændning ligger på 1Vpp. Derfor indsættes en ekstra modstand R3, som dæmper i indgangen. Da det bifasemodulerede output fra FPGA'en ligger ved spændinger fra 0-3.3 V, skal R3 være 2K ohm, for at det endelige output overholder SMPTE-normen på 1 Vpp.

**Figur A-4:** Filter karakteristik for LTC-filter

## Appendiks B: Ordliste

**Blackburst:** SD (Standard Definition) signal, der udover sync-pulser ikke indeholder billeddata, og derved fremstår som et sort billede.

**Dropframe:** For at forhindre tidsforskydelse, når der tælles frames med NTSC-sekunder (1,001 faktiske sekunder), skippes et vist antal frames. Denne metode kaldes *dropframe*.

**FPGA:** Field Programmable Gate Array. Programmerbar logik.

**Fps:** Frames per sekund.

**Genlock:** Opsynkronisering af flere videokilder.

**GPS:** Global Positioning System. Positionering vha. satelliter. Kan også bruges til præcise timingapplikationer.

**GPS-tid:** GPS-tid starter d. 6 januar 1980, kl 00:00. Herfra tælles i uger, og sekunder i ugen.

**I2C:** Seriel databus udviklet af Philips. Benytter to wires, én clock og én datawire.

**Jitter:** Støj i en frekvens. Små ændringer i periodetiden. Kan også opstå pga. kvantisering af en clock.

**LTC:** Linear Time Code. Udviklet af SMPTE, til tidsstempling af videosignaler. Indeholder frametæller og tid, og består af et audiospor med bifasemoduleret data.

**MCU:** Micro Controller Unit. Lille processor til styring af elektroniske kredse.

**NavSync:** Irsk firma, der udvikler det benyttede GPS-chipset.

**NMEA:** National Marine Electronics Association. Forbund der har defineret protokollen der benyttes af NavSync modulet.

**NTSC:** National Television Systems Committee. Hovedsagelig amerikansk TV-standard. Benytter 525 linier pr. billede, ved 30/1,001 fps.

**PAL:** Phase Alternating Line. Hovedsagelig europærisk TV-standard. Benytter 625 linier pr. billede ved 25 fps.

**PLL:** Phase Locked Loop. Metode hvorpå én clock kan låses til en anden.

**PT5300:** DK-Technologies synkroniserings mainframe, hvori GPS-modulet skal monteres.

**SMPTE:** Society of Motion Picture and Television Engineers. Har udviklet LTC-standarden

**UTC:** Coordinated Universal Time. Også kendt som GMT (Greenwich mean time). Civil tidsregning, der indeholder skudsekunder og skuddage, for at tilpasse 24 timer til Jordens rotation.

**VCXO/OCXO:** Voltage Controlled XTal Oscillator. Spændingsstyret krystaloscillator. OCXO er en spændingsstyret krystaloscillator indbygget i en varmeovn, der derved gør temperaturstabilitet til en uvæsentlig faktor.