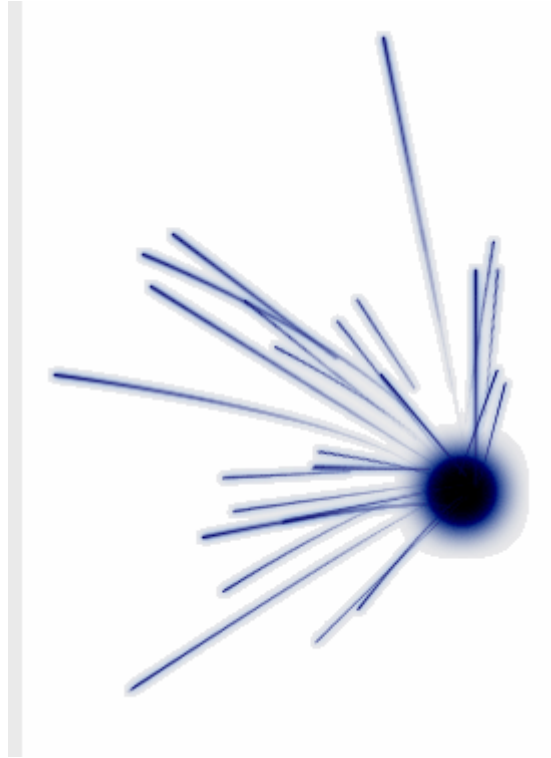




**Escola Politècnica Superior
d'Enginyeria de Manresa**

UNIVERSITAT POLITÈCNICA DE CATALUNYA



PROJECTE FINAL DE CARRERA

**ESTUDI I REALITZACIÓ PRÀCTICA D'UN SISTMA DE
SINCRONISME PER A UN RECEPTOR DE SENYALS UWB
IMPULSIUS**

NOM: ALBERT PALOMAS SERRA
BRANCA: ENGINYERIA TÈCNICA DE TELECOMUNICACIONS
ESPECIALITAT: SISTEMES ELECTRÒNICS
PROFESSOR TUTOR: PERE PALÀ SCHÖNWÄLDER
CONVOCATÒRIA: JUNY 2009
Nº REGISTRE: 1856

1. Index

2. Introducció.....	4
3. Introducció a l'UWB	6
3.1. La tecnologia UWB	7
3.2. Avantatges de la tecnologia UWB.....	8
3.3. Modulacions en UWB	8
4. Un receptor UWB a grans trets	11
4.1. Receptor de dades.....	11
4.2. Perquè síncron i no asíncron?.....	12
4.3. El problema de la sincronització quan no disposem de la senyal de sincronisme.....	13
4.4. Estructura general d'un receptor UWB síncron	13
4.5. Quina modulació emprar?	14
5. Disseny teòric del sistema de sincronisme	16
5.1. Model del sistema a grans trets	16
5.2. El llaç tancat de fase (PLL)	17
5.3. Diferents detectors de fase.....	19
5.3.1. El detector Track & Hold	20
5.3.2. Detector de fase implícit.....	21
5.3.3. El detector Early-late	24
5.4. Disseny teòric i simulació del model sobre <i>Simulink</i>	26
5.4.1. Condicions de treball de baixa freqüència.....	26
5.4.2. Generació de la senyal amb el <i>Matlab</i>	27
5.4.3. La tècnica de sincronisme 'Single- Flank'	28
5.4.5. Tècnica de sincronisme amb el detector de fase implícit	31
5.4.6. La tècnica de sincronisme 'Early- Late'	37
6. Disseny pràctic del sistema de sincronisme	42
6.1. Etapa de mostreig, el convertidor A/D.....	42
6.2. Etapa d'implementació de funcions digitals, la FPGA	45
6.3. Adaptador entre la FPGA i el VCO: el convertidor D/A	46
6.4. El mòdul VCO.....	50
6.5. Adaptador d'escala	53
6.6. Generació de la senyal de rellotge, el nou VCO.....	56
6.7. Connexió entre tots els mòduls.....	58
6.8. Test del hardware i proves.....	60
6.9. Mesura del temps de reacció del VCO	62
6.9.1. Mesura del temps de reacció de la font de tensió configurada com a VCO.....	62
6.9.2. Mesura del temps de reacció del VCO 74HCT4046.....	64
6.10. Implementació pràctica del model de sincronisme 'Single-Flank'	66
6.11. Implementació pràctica del model de sincronisme amb el detector de fase implícit.....	68
6.12. Implementació pràctica del model 'Early-Late'	69
6.13. Comparació de les tècniques de sincronisme anteriors	72
6.14. Aplicació de les tècniques a freqüències altes.....	73

7. Conclusions i línies futures	78
8. Bibliografia.....	80
Annex A: Programa per a Matlab per a generar un tren de polsos triangulars.....	81
Annex B: Programa per a Matlab per a generar un tren de polsos gaussians.....	82
Annex C: Programa per la FPGA per a provar el funcionament del hardware i implementar el detector de fase 'Track & Hold'	83
Annex D: Programa per la FPGA per a mesurar el temps de reacció del VCO.....	84
Annex E: Programa en VHDL per al sistema de sincronisme amb el detector de fase implícit.....	87
Annex F: CD amb el tots els fitxers digitals generats en la realització d'aquest projecte	89

2. Introducció

La tecnologia UWB (Ultra Wide Band) que en anglès vol dir Banda Ultra Ample, va ser una de les primeres tecnologies de ràdio descobertes i es basava en la generació de 'xispes' per tal de generar components freqüencials que es poguessin transmetre per ones electromagnètiques.

Tot i ser de les primeres, no ha estat fins fa pocs anys que se li he tret la pols i se li han començat a veure avantatges i prestacions interessants. És una tecnologia de transmissió inal·làmbrica de dades molt prometedora en l'àmbit de les PAN (Personal Area Network), per les seves característiques, com velocitat de transmissió, baix cost i baix consum entre d'altres.

Perquè en la societat actual són clau en l'enginyeria de les comunicacions els aspectes d'innovació, alta velocitat de transferència de dades, flexibilitat, integració, baix consum, dificultat d'intersecció entre d'altres, m'ha semblat interessant basar el projecte en aquesta tecnologia.

Inicialment l'objectiu del projecte era aconseguir un receptor de dades UWB, però es va veure que era un projecte massa extens i que no hi hagués hagut prou temps per a realitzar-lo. Així els objectius d'aquest projecte estan pensats per obrir camí cap a un receptor d'aquestes característiques, però sense arribar-hi.

Un senyal UWB impulsiu és un senyal format per un seguit de 'xispes' de molt poca durada, de l'ordre dels ps. Per tal de poder enviar informació amb aquestes 'xispes', cal que aquestes siguin diferents depenent de la informació que es vulgui enviar. Si ens posem a dissenyar un receptor d'aquest tipus, veurem que aquest s'ha de mirar la 'xispa' i extreure'n la informació que conté. També ens en adonarem que per poder mirar la 'xispa' cal saber quant arriba. Del procés per a arribar a saber quant arriba aquesta 'xispa' se'n diu la sincronització del receptor.

És precisament aquest procés el que es vol aconseguir amb aquest projecte. Concretament aconseguir estudiar, dissenyar i implementar físicament un sistema de

sincronització per a aquest tipus de receptors són les metes a aconseguir en aquest treball.

Per aconseguir això es buscaran diferents tècniques de sincronització ja existents i se'n desenvoluparan de noves. Totes elles s'estudiaran i es dissenyaran models de sistemes de sincronisme que les facin servir, els quals es simularan i es portaran a la pràctica en un circuit real. Tots aquests procediments es faran a baixa freqüència per tal d'eliminar molts dels problemes que porten les altes freqüències. Al final del projecte, s'haurà d'haver aconseguit un model de sistema de sincronisme que funcioni correctament i a baixa freqüència tant en entorns de simulació com en la realitat.

Espero que la lectura d'aquest treball sigui entenedora i de profit per no només aquells que estan interessats en el disseny de receptors d'aquest tipus o en qualsevol altre circuit de sincronització, sinó també per a tots aquells aficionats a la electrònica.

M'agradaria finalment agrair l'ajuda prestada, la participació, els consells i suggerències donades durant el desenvolupament d'aquest projecte al meu director de projecte Pere Palà, a la meva família, a la meva parella Clara i als meus amics.

3. Introducció a l'UWB

L'UWB és una tecnologia de ràdio que va ser inventada per Marconi a l'any 1897 en el seu afany per aconseguir una transmissió de ràdio a distància. Per fer això, havia de generar components freqüencials per tal que “alguna cosa” es pogués propagar per l'aire.

Investigant per aquesta línia va observar que una “xispa”, com pot ser la d'una bugia d'automòbil, generava components freqüencials, tals que es propagaven per l'aire. No és d'estranyar que Marconi transmetés “xispes” en els seus primers receptors. I en efecte, l'emissor generava xispes i el receptor no era res més que un detector magnètic.

Aquesta tecnologia va ser àmpliament utilitzada pels primers operadors Morse fins a la primera dècada del segle XX. Més tard es va abandonar, ja que no permetia transmetre gaires operadors Morse pel mateix canal (ocupava massa ample de banda) i es va substituir per transmissors basats en alternadors i oscil·ladors (de banda estreta).

No ha estat fins fa poc, que aquesta tecnologia s'ha vist prometedora com a capa física en l'àmbit de les PAN. Les seves propietats la fan ideal per aquest tipus de xarxes: permet grans velocitats de transmissió a poca distància, no interfereix a les transmissions pròximes, té un baix consum, és difícil d'interceptar i altres.

Actualment, existeixen prototipus d'alguns aparells que funcionen amb la tecnologia UWB, com per exemple *pendrives USB*.

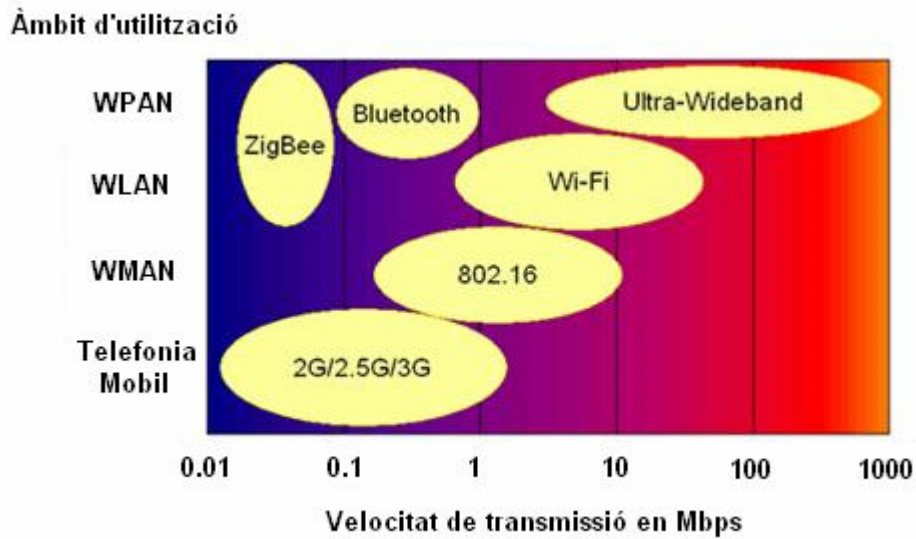


Fig. 3.1. Tecnologies inalàmbriques utilitzades segons la freqüència

3.1. La tecnologia UWB

A diferència de les tecnologies de ràdio convencionals, que s'estudien en domini freqüencial, les propietats que particularitzen la tecnologia UWB estan en el domini temporal. Com el seu nom diu, les senyals UWB tenen un espectre freqüencial molt ample, això en el domini temporal es tradueix en polsos molt estrets, de l'ordre dels ps.

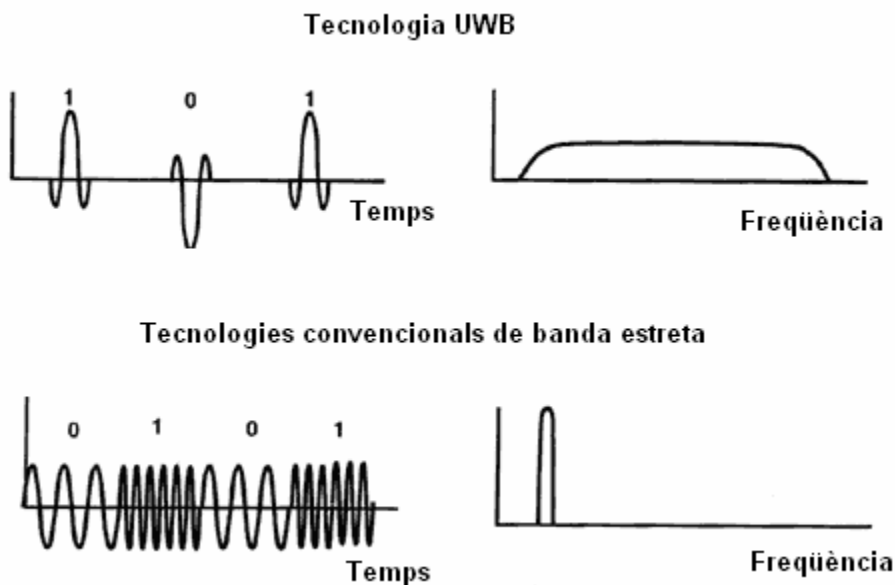


Fig. 3.2. Característiques d'un senyal UWB i d'un de banda estreta

3.2. Avantatges de la tecnologia UWB

Les senyals UWB no afecten ni es veuen afectades per a les senyals de banda estreta. Això és així perquè la potència de les senyals UWB està distribuïda en moltes freqüències i s'assembla al soroll. Aquesta propietat fa que aquestes senyals siguin difícils d'interceptar. Com que la radiació emesa no està per sobre del soroll tampoc cal pagar llicències d'emissió per a efectuar una transmissió UWB.

3.3. Modulacions en UWB

Existeixen dos tipus de modulació en UWB, una és la OFDM (*Orthogonal Frequency Division Multiplexing*) i la altra és la modulació per impulsos. Aquest projecte es centra en la modulació per impulsos i deixa completament de banda la modulació OFDM.

La modulació per impulsos, té lloc en el domini temporal i deixa de banda qualsevol aspecte freqüencial. Modular uns impulsos vol dir donar forma a un tren de polsos en funció de les dades que es volen transmetre. Cada impuls o pols és un bit, és a dir si es volen transmetre 8 bits, caldrà enviar 8 polsos.

L'aspecte d'un senyal impulsiu sense modular és aquest:

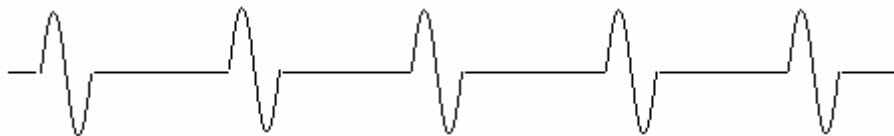


Fig. 3.3. Tren de polsos sense modular

Per modular les dades sobre els polsos es fan servir diferents tècniques, les més utilitzades són les que segueixen a continuació.

Modulació en amplitud (PAM) Pulse Amplitude Modulation

El pols té una amplitud o una altra depenent de si es vol transmetre un '1' o un '0'.

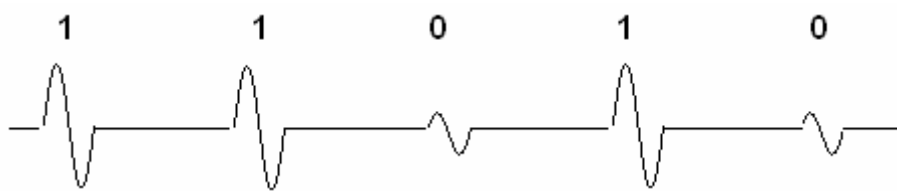


Fig. 3.4. Dades modulades en PAM

Modulació en posició del pols (PPM) Pulse Position Modulation

El pols té una posició diferent respecte la normal en funció de si es vol transmetre un '1' o un '0'.

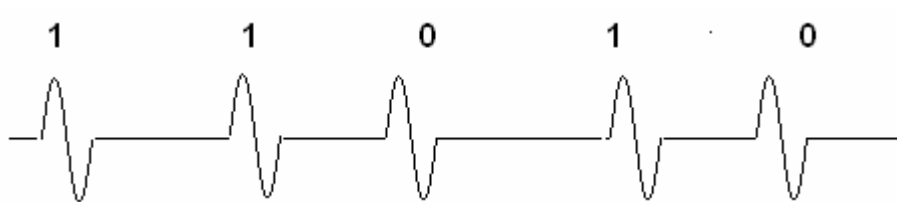


Fig. 3.5. Dades modulades en PPM

Modulació en fase (BPM o PSK) Phase Shift Keying

El pols té una fase o una altra en funció de si es vol transmetre un '1' o un '0'.

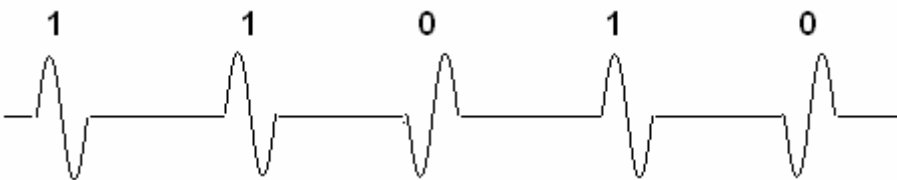


Fig. 3.6. Dades modulades en PSK

Modulació On-Off (OOK) On Off Keying

El pols es posa o no depenent de si es vol transmetre un '1' o un '0'.

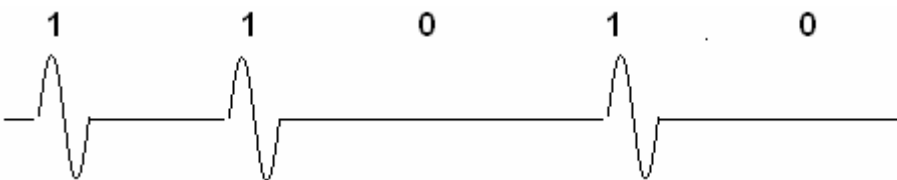


Fig. 3.7. Dades modulades en OOK

Si aquestes formes d'ona es posen en una antena, aquestes es transmeten per l'aire i es poden rebre des d'un altre punt, on un receptor les pot desmodular i recuperar les dades transmeses: en els casos anteriors "11010".

4. Un receptor UWB a grans trets

El sistema de sincronisme que es pretén estudiar i desenvolupar en aquest projecte, té com a finalitat aconseguir sincronitzar un receptor UWB de senyals impulsius. A continuació es detalla com hauria de ser aquest receptor.

4.1. Receptor de dades

El sistema de sincronització que es vol aconseguir està pensat per funcionar en un receptor de dades. Això vol dir que el receptor, a partir de la senyal rebuda, ha de obtenir les dades que ha transmès l'emissor.

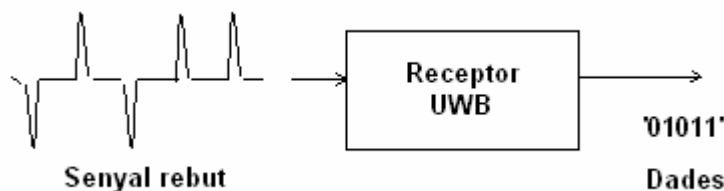


Fig. 4.1. Diagrama de blocs d'un receptor de dades

Un cop rebudes, les dades, es podrien lliurar a un dispositiu que les necessités. Seria interessant que a més a més aquest receptor pogués treballar a una gran velocitat, és a dir que pogués rebre molts bits per segon. Això el faria apte per a l'àmbit de les PAN, on cada cop es requereix més velocitat de transmissió.

Els aparells d'aquest àmbit, com els telèfons mòbils, els ordinadors portàtils i altres, són aparells que s'han d'alimentar amb bateries, per això és important que gastin poc. Així doncs aquest receptor també és dissenyarà amb aquesta condició: que tingui un baix consum.

Per altra banda, com que es tracta d'una transmissió de dades, s'haurà de decidir si es farà de manera síncrona o asíncrona.

4.2. Perquè síncron i no asíncron?

No hi ha cap estàndard que digui si les transmissions UWB han de ser síncrones o asíncrones, per tant haurem de valorar quins avantatges i quins inconvenients té cada opció per a decidir com haurà de ser el nostre receptor.

L'avantatge de les transmissions asíncrones, és que no es necessita la mateixa senyal de rellotge en el receptor que en l'emissor. I llavors perquè no ens decantem per aquesta opció? Si la transmissió de dades fos asíncrona, els polsos s'haurien de transmetre d'alguna manera tal que el receptor no se'ls hagués de mirar estrictament en una posició concreta, sinó que se'ls podria mirar en una posició indiferent. Aquesta manera de mirar-se'ls podria ser per submostreig, és a dir amb una freqüència diferent a la del emissor i per tant segur que en algun moment o altre veuria algun pols. El problema d'aquesta opció és que s'haurien de transmetre molts polsos per enviar un sol bit i per tant la velocitat de transmissió seria baixa.

L'altra opció és que el receptor sigui síncron. D'aquesta manera el receptor s'ha de mirar el senyal estrictament en una posició concreta; en la posició que digui el senyal de sincronisme. L'inconvenient d'aquesta opció és que en el receptor es necessita la mateixa senyal de sincronisme que en l'emissor i com que aquest és inal·làmbric no disposem d'aquesta senyal directament. L'avantatge que suposa una transmissió síncrona és que es pot aconseguir molta més velocitat que en una transmissió asíncrona perquè en cada pols es transmet un bit.

Si valorem les dues opcions, la síncrona és la que s'ajusta més a les nostres necessitats. Així doncs és necessari un sistema de sincronisme.

Una transmissió síncrona en tecnologia UWB seria així: l'emissor transmetria un tren de polsos, generats diferentment si s'ha de transmetre un '0' o un '1' en funció de la modulació emprada. Els polsos es transmetrien amb una freqüència determinada, marcada per un rellotge. Per exemple, un emissor amb un rellotge de 1 Hz, transmetria un pols cada segon, conseqüentment la velocitat de transmissió de dades seria d'1bps.

El receptor s'hauria de mirar aquesta senyal un cop per segon i ho hauria de fer just sobre on hi ha el pols i no a un altre lloc, d'aquí la necessitat del sistema de sincronisme.

4.3. El problema de la sincronització quan no disposem de la senyal de sincronisme

Un receptor síncron necessita la senyal de rellotge amb que l'emissor ha generat les dades per poder-les rebre correctament. Si la transmissió és inal·làmbrica, no hi ha cap manera d'enviar la senyal de rellotge de l'emissor al receptor i per tant s'haurà de buscar alguna manera d'obtenir aquesta senyal des del receptor.

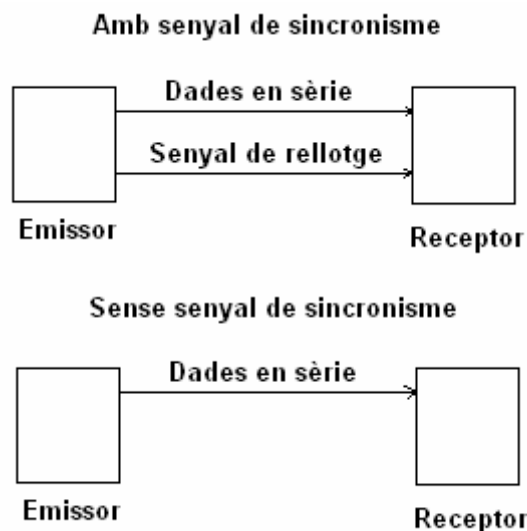


Fig. 4.2. Diferència entre una transmissió amb la senyal de rellotge i una transmissió sense aquesta senyal

4.4. Estructura general d'un receptor UWB síncron

Un receptor síncron, s'ha de mirar cada pols que rep i decidir si és un '1' o un '0'. Però quan se'l ha de mirar? Això li diu la senyal de sincronisme, que per anar bé, hauria de ser la mateixa que la de l'emissor. És a dir, el receptor s'ha de mirar la senyal just en el moment que arriba el pols, ni més tard ni més aviat.

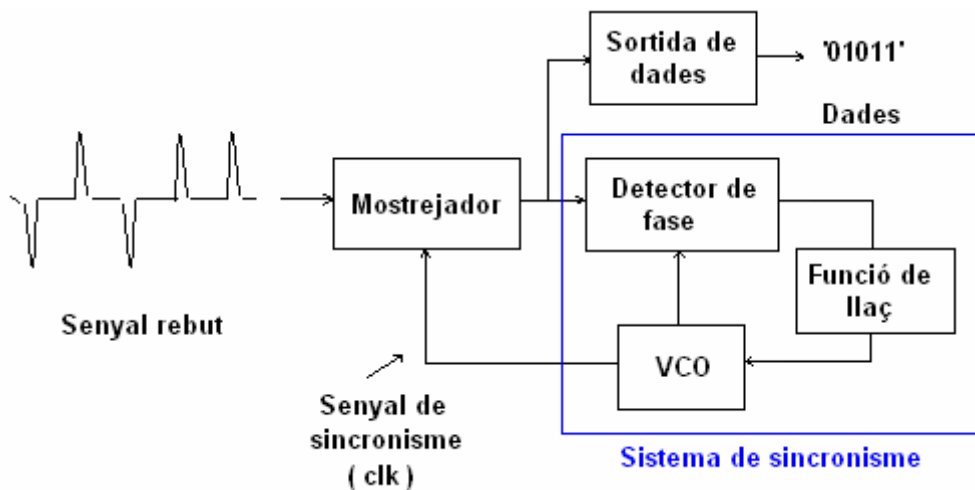


Fig. 4.3. Diagrama de blocs d'un receptor UWB síncron

Algú podria pensar que si la freqüència amb que l'emissor envia els polsos és coneguda, només cal posar un rellotge de la mateixa freqüència en el receptor. Això no funcionaria perquè es podria encertar la freqüència però seria molt difícil encertar la fase. A més si s'encertés la fase, una minúscula variació en un dels dos rellotges faria que el receptor es deixés de mirar els polsos allà on toca. Per tant s'haurà de buscar alguna manera perquè la senyal de rellotge i la senyal rebuda sempre estiguin en fase.

Per obtenir la senyal de rellotge adequada, el receptor ho haurà de fer a partir de la senyal rebuda. En general, per obtenir-la es fa amb un llaç tancat de fase, en anglès PLL (Phase Locked Loop). Més endavant s'entra en detall amb això del PLL.

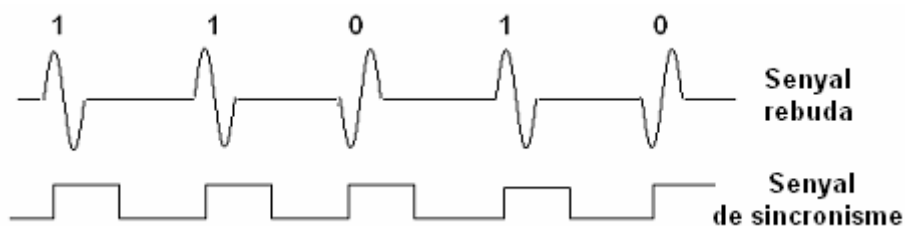


Fig. 4.4. Senyal rebut i senyal de sincronisme sincronitzats

4.5. Quina modulació emprar?

És important triar una modulació adequada, ja que aquesta serà determinant en el disseny del sistema de sincronisme.

Si estudiem els avantatges i els inconvenients de cada modulació ens en adonarem del següent: Si fem servir una modulació PAM o OOK, el receptor pot confondre una deficiència de la qualitat del senyal amb un pols més petit, però tenim l'avantatge de que tots els polsos ens arriben separats en el mateix interval i serà més fàcil obtenir la senyal de sincronisme. Si fem servir una modulació PPM els polsos '1' i els polsos '0' ens arriben en intervals diferents i serà molt difícil trobar la senyal de sincronisme.

Si fem servir una modulació BPM o PSK, els polsos ens arriben en intervals iguals i amb amplitud constant, llavors no tindríem el problema de la PAM, en que es podrien confondre polsos més petits amb una baixada de la intensitat del senyal. Però cal tenir en compte que en aquesta modulació, els polsos corresponents a '1' i a '0' arriben al receptor en signes contraris. Per exemple els polsos corresponents a un '1', ens arriben en signe positiu (cap per amunt) i els polsos corresponents a un '0' arriben en signe negatiu (cap per avall). Si fem el mòdul d'aquest senyal, tenim un tren de polsos, cosa que facilitaria en gran mesura la sincronització.

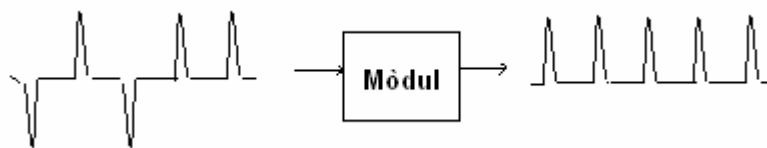


Fig. 4.5. Esquema de la funció mòdul

La modulació que presenta més avantatges és la BPM (Bi-Phase Modulation), i per tant farem servir aquesta per a dissenyar el sistema de sincronisme.

5. Disseny teòric del sistema de sincronisme

En aquest apartat es pretén aconseguir el disseny teòric d'un sistema de sincronisme que pugi fer sincronitzar un receptor com el que s'ha vist en els apartats anteriors. Per aconseguir això s'estudien i diferents tècniques de sincronització. Concretament se'n estudien dues de conegudes i una que no ho és, que és completament experimental i fruit d'aquest projecte. També es porten a terme simulacions teòriques, que son comparades i examinades per tal de detectar problemes i trobar millores.

5.1. Model del sistema a grans trets

Com es veu en l'apartat de *estructura general d'un receptor UWB síncron*, el sistema de sincronisme consta d'una etapa d'adquisició (el mostrejador), una etapa de comparació de fase (el detector de fase), i una etapa de generació de rellotge (el VCO).

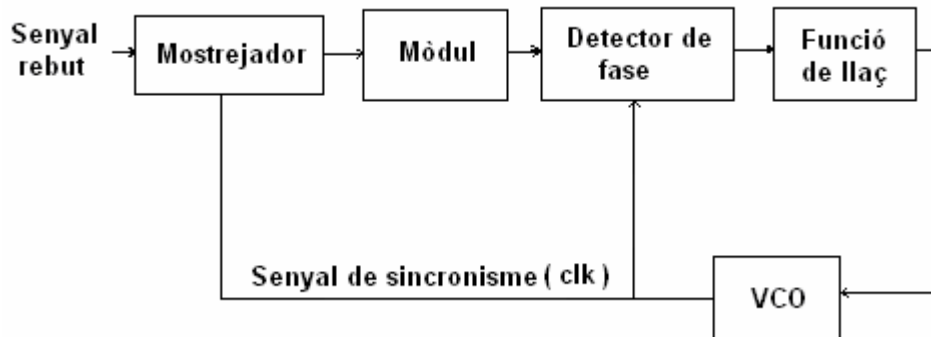


Fig. 5.1. Diagrama de blocs d'un receptor síncron

El mostrejador se'n encarrega d'obtenir el valor del senyal rebut cada cop que li diu la senyal de sincronisme. Fins que no arribi el següent cicle de clk, aquest haurà de donar a la sortida l'últim valor mostrejat.

L'etapa de comparació de fase, serveix per saber si la senyal de rellotge i la senyal d'entrada estan en fase. Dit d'una altra manera, aquesta etapa se n'ha d'adonar si el convertidor AD està prenent mostres sobre els polsos o ho està fent en un altre lloc.

L'etapa de generació de rellotge (VCO) ha de generar la senyal de sincronisme que servirà de rellotge a tots els elements síncrons del receptor. Com que la freqüència s'ha de poder controlar, s'ha fet servir un VCO en lloc d'un oscil·lador fix. D'aquest esquema se'n diu llaç tancat de fase.

5.2. El llaç tancat de fase (PLL)

El llaç tancat de fase, en anglès phase locked loop, ha d'aconseguir que la fase de la senyal rebuda sigui la mateixa que la de la senyal de sincronisme. Quan passa això es diu que el PLL està enganxat. En aquest estat, el mostrejador sempre prendrà mostres sobre els polsos. Per tant la clau del sincronisme està en aquest llaç.

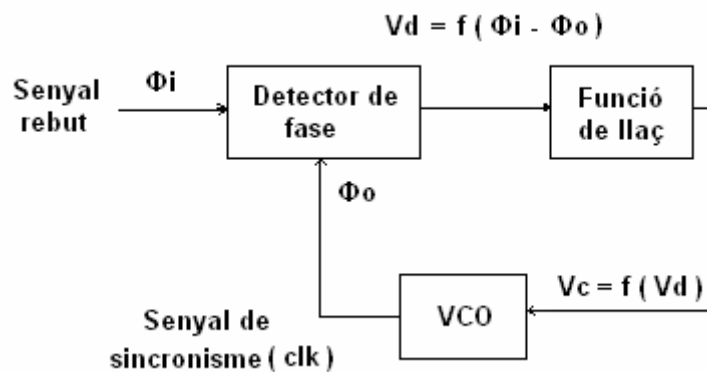


Fig. 5.2. Esquema d'un PLL

El PLL està format bàsicament per un detector de fase, una funció de llaç i un VCO.

El detector de fase dona una tensió V_d que és funció de l'error de fase entre la senyal de sincronisme i la senyal rebuda. L'error de fase es coneix com la diferència entre la fase del senyal rebut i la fase del senyal de sincronisme.

$$\Phi_e = \Phi_i - \Phi_o$$

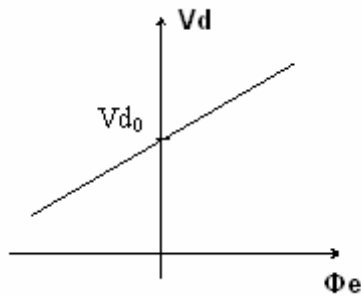


Fig. 5.3. Característica d'un detector de fase ideal

La funció de llaç s'encarrega de proporcionar la tensió V_c , que és funció de la tensió V_d . El comportament del PLL variarà depenent d'aquesta funció. Per entendre el concepte de PLL es pot deprecier aquesta funció i suposar que $V_c = V_d$.

El VCO genera un senyal de rellotge periòdic, la freqüència del qual és proporcional a la tensió V_c i la fase és la integral d'aquesta freqüència.

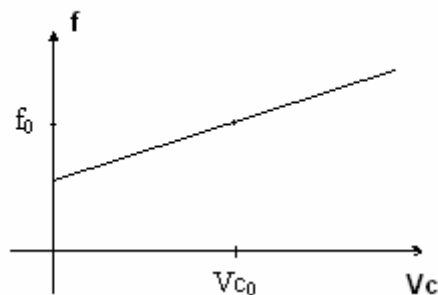


Fig. 5.4. Característica d'un VCO ideal

En general el comportament d'un PLL ideal quan està enganxat és aquest:

Si la senyal d'entrada i la senyal generada pel VCO estan en fase, el detector de fase dóna una tensió V_{d0} que fa que el VCO generi una freqüència f_0 . A aquesta freqüència l'error de fase és 0. Si en algun moment, per qualsevol causa, la fase de la senyal d'entrada canvia, la tensió generada pel detector de fase també ho farà. Això farà que el VCO actuï per tal de que l'error de fase segueixi sent 0. D'aquesta manera la fase del senyal de sincronisme anirà 'seguint' la fase del senyal rebut.

5.3. Diferents detectors de fase

Com s'ha vist en l'apartat anterior, el detector de fase dóna una tensió que és funció del desfàs entre la senyal de rellotge o sincronisme i la senyal rebuda. El desfàs entre les dues senyals es coneix com error de fase ' Φ_e ' i és la diferència entre la fase de la senyal d'entrada ' Φ_i ' i la fase de la senyal de rellotge ' Φ_o '.

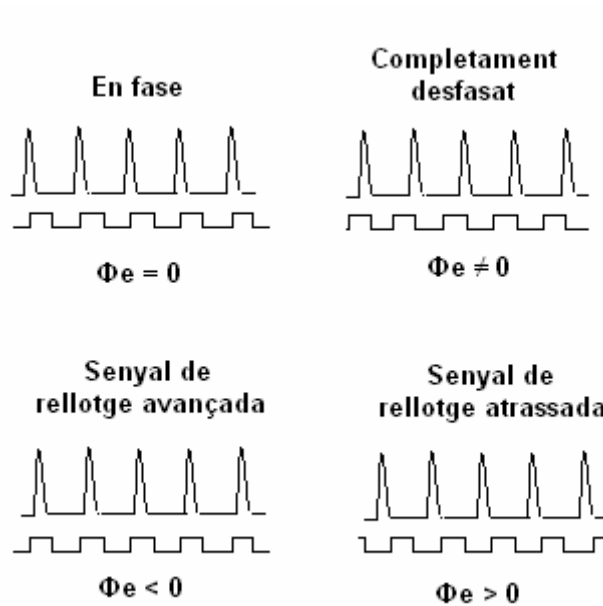


Fig. 5.5. Diferents estats de la senyal rebuda respecte la senyal de sincronisme

Per poder mostrejar el pols allà on toca, l'error de fase entre la senyal rebuda i la senyal de sincronisme ha de ser 0. Perquè el PLL pugui enganxar amb error de fase 0, cal que la característica del detector de fase tingui pendent positiu quant l'error de fase és 0. És a dir, si la tensió V_d val 1V quant l'error de fase és 0; la tensió generada per errors de fase més grans que 0 ha de ser més gran que 1V i la tensió generada per errors de fase més petits que 0 ha de ser més petita.

En la majoria dels detectors de fase quan l'error de fase és 0, la tensió V_d és màxima i el pendent és 0. Això fa que no puguin enganxar amb error de fase 0, la qual cosa comportaria que el nostre receptor no mostrejaria els polsos exactament allà on toca.

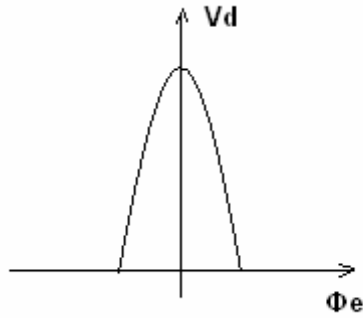


Fig. 5.6. Característica d'un detector de fase corrent

El detector de fase ideal seria el que en el punt d'error de fase = 0, la característica tingués pendent positiu.. Hi ha molts tipus de detectors de fase com per exemple mescladors, portes lògiques o altres, però tots tenen el mateix problema; que no poden enganxar amb error de fase 0. Per tant seria bo trobar un detector que no ens donés aquest problema. A continuació s'estudien tres tipus de detectors de fase.

5.3.1. El detector Track & Hold

Track & Hold en anglès vol dir seguir i mantenir. Com el seu nom indica, el que fa aquest detector de fase és mostrejar el senyal rebut a la freqüència del senyal de rellotge i mantenir la mostra a la seva sortida fins a la pròxima. Si els senyals de rellotge i rebut estan en fase, les mostres tindran el valor màxim perquè aquestes es prendran al mig del pols, que és quan aquest pren el màxim valor. En el moment en que les mostres es comencin a prendre en un punt una mica allunyat del centre del pols, el valor d'aquestes anirà disminuint.

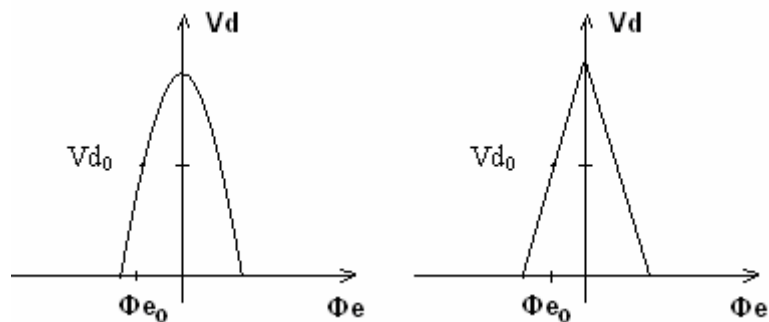


Fig. 5.7. Diferents característiques per a un detector de fase. El primer cas per polsos arrodonits i el segon cas per polsos triangulars

Així doncs un detector de fase 'track & hold' és un simple mostrejador. Aquest detector té un gran avantatge que és la seva simplicitat. Però té un inconvenient i és que no permet que el PLL enganxi amb error de fase 0. Això és degut a que en el punt de error de fase 0, el pendent de la característica no és positiu sinó que es nul.

Si la funció de llaç té un guany positiu, el PLL amb aquest detector enganxarà a la part de la característica que té pendent positiu, sent Vd_0 la que es mostra a la *Fig.5.7*.

En aquesta part, l'error de fase entre la senyal rebuda i la senyal de sincronisme és negatiu, això vol dir que no es prendran les mostres exactament sobre els polsos sinó una mica abans.

Una altra cosa a tenir en compte és que la característica del detector depèn del nivell i de la forma de la senyal d'entrada, cosa que no es massa adequada perquè la senyal pot arribar més atenuada o amb rebots. Així doncs aquest no és massa bon detector

5.3.2. Detector de fase implícit

El problema anterior es podria solucionar si fóssim capaços de trobar un sistema que tingués el següent comportament: si l'error de fase és més gran que 0, fer reaccionar el VCO perquè augmentés la freqüència i si l'error de fase fos més petit que 0, fer reaccionar el VCO perquè la disminuís.

Però hi ha un petit problema i es que el mostrejador només ens dona informació sobre el nivell de la senyal i no la fase d'aquesta, per tant no la coneixem directament. Però si que podem fer la següent hipòtesi:

Suposem que inicialment estem en fase, és a dir que estem mostrejant sobre els polsos i que llavors, per qualsevol motiu, una de les dos fases canvia. En aquest moment les mostres tindran un valor inferior a les mostres anteriors degut a que no estarem mostrejant sobre el pols exactament sinó sobre un dels seus flancs, ja sigui de pujada o de baixada. Això ens dona informació de que ens estem desfasant però no hi ha manera de saber si això està passant perquè la senyal de rellotge va avançada o va atraçada.

Llavors es pot suposar una de les dos coses, per exemple que la senyal de rellotge està avançada respecte la senyal d'entrada. Llavors li diem al VCO que la atraci i ja tornem a estar en fase. Però, i si la suposició és errònia? Si el senyal hagués estat atraçat i l'haguéssim atraçat més, llavors el rellotge estaria el doble de desfasat que al inici. Però si passa això, podríem avançar el rellotge el doble del que l'havíem atraçat i ja tornaríem a estar en fase. La única manera d'aconseguir aquest comportament és amb algun tipus de diagrama de flux o màquina d'estats.

Aquesta és la màquina d'estats que s'ha dissenyat per a complir aquesta funció:

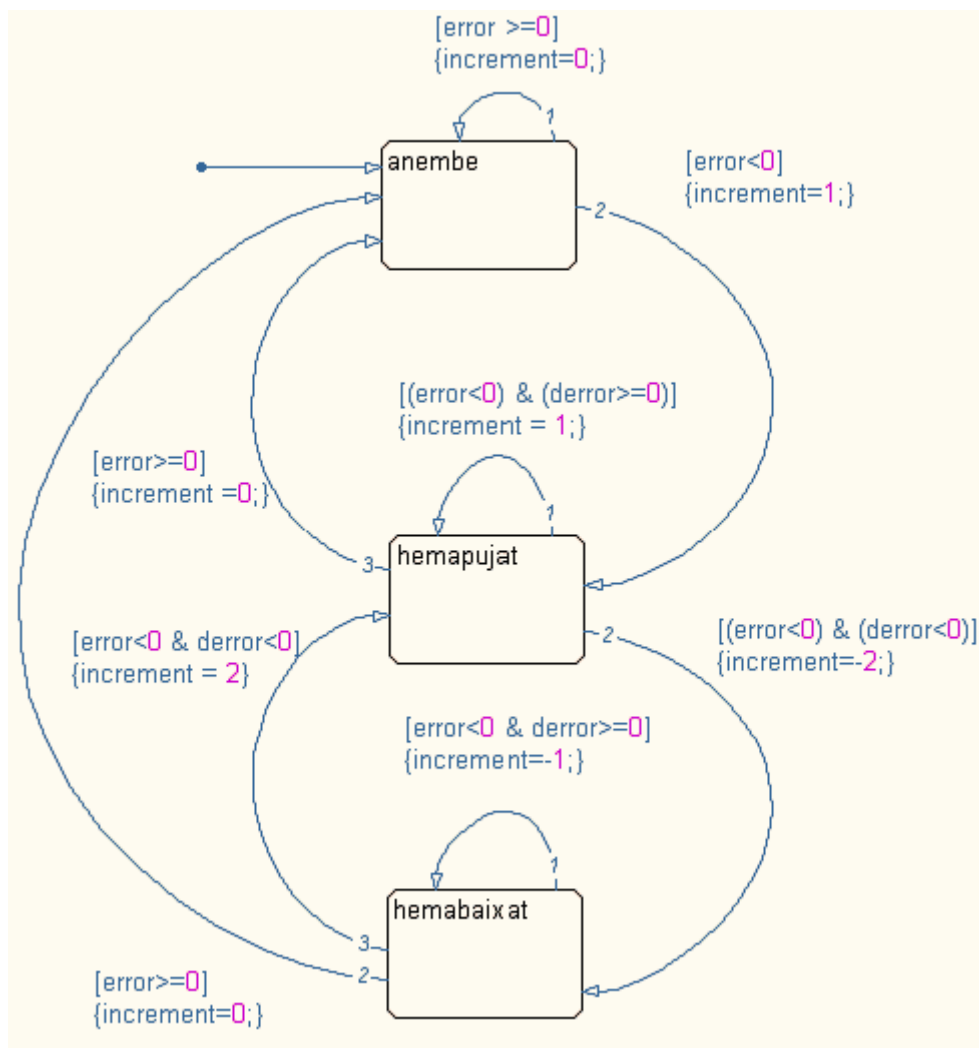


Fig. 5.8. Màquina d'estats que implementa el detector de fase implícit

La màquina d'estats és de tipus Mealy i s'ha dissenyat amb el stateflow de *Simulink*. La sintaxis és aquesta: les sentències entre `[]` són condicions de salt entre estats i les sentències entre `{ }` són accions que es fan en la transició d'estats.

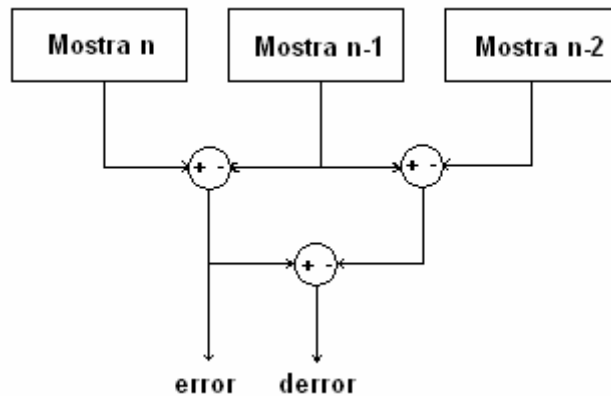


Fig. 5.9. Diagrama de blocs de com s'obtenen les variables error i derror

La variable *error* es calcula com la diferència entre la mostra actual i la mostra anterior i és negativa quant la mostra actual té un valor inferior a la anterior. Això informa a la màquina d'estats que les mostres cada cop s'estan prenent més lluny del centre del pols.

La variable *derror* informa a la màquina d'estats de si el canvi en la freqüència ha estat encertat o no. El valor d'aquesta variable es calcula restant el valor de la variable error després de la correcció amb el valor anterior a la correcció. Aquesta variable és positiva quant s'ha fet un canvi en la freqüència i s'ha encertat i és negativa quant no s'ha encertat.

La variable *increment* és el canvi que es fa en la tensió de control del VCO.

Un exemple de funcionament seria el següent:

Suposem que inicialment la freqüència de la senyal rebuda i la freqüència de la senyal de rellotge son semblants però no iguals. Les mostres es prendrien durant estones sobre els polsos i durant estones fora dels polsos. En el moment en que les mostres s'estan prenent en el centre dels polsos i es comencen a prendre desplaçades del centre, la màquina, que estava a l'estat anembé, apujarà la freqüència i passarà a l'estat hemapujat. En aquest estat comprovarà si el fet d'apujar la freqüència ha estat un bon canvi o no. En cas de que hagi estat un bon canvi, tornarà a fer el mateix un i altre cop fins que les mostres actuals tinguin un valor més gran que les mostres anteriors. Però en cas de que augmentar la freqüència hagi estat un mal canvi, llavors, la màquina passarà a l'estat hem abaixat, abaixant 2 cops la freqüència. En aquest punt segur que s'ha

encertat la decisió perquè si inicialment només hi havia dos opcions i una hem vist que no era, doncs és l'altra. D'aquesta manera, la màquina sempre obliga a la senyal de rellotge a estar amb un error de fase pràcticament nul respecte la senyal rebuda.

La característica d'aquest detector no es pot especificar amb una funció matemàtica ni amb una gràfica ja que el funcionament es basa en un flux d'operacions. Però la lògica diu que ha de permetre enganxar amb error de fase molt pròxim a 0 tot i que mai ho acabarà sent exactament. Aquesta característica no depèn dels nivells ni de la forma de la senyal d'entrada. Això és un gran avantatge ja que si la senyal es veu atenuada, el detector seguirà funcionant igualment. A més permet enganxar amb error de fase molt pròxim a 0 tot i que mai ho serà exactament. Per altra banda és un detector bastant complex ja que s'ha d'implementar una màquina d'estats força complexa.

Aquest tipus de detector no existeix com a tal, al menys no n'he pogut trobar cap referència. És en aquest projecte que es dissenya des de zero i s'experimenta amb un detector d'aquest tipus sense tenir-ne cap referència prèvia.

5.3.3. El detector Early-late

Una altra manera de resoldre el problema podria ser implementant un detector de fase early-late. Un detector de fase early-late, que en anglès vol dir dora-tard, és un detector que, a diferència del anterior, dóna informació exacta de l'error de fase entre els dos senyals. El nom de early-late explica el seu funcionament: el mostrejador pren mostres sobre els polsos però no ho fa exactament quan ho diu la senyal de rellotge sinó que en uns polsos ho fa una mica abans i en uns altres una mica després. La diferència entre les mostres preses abans (early) i les preses més tard (late), és la informació de fase.

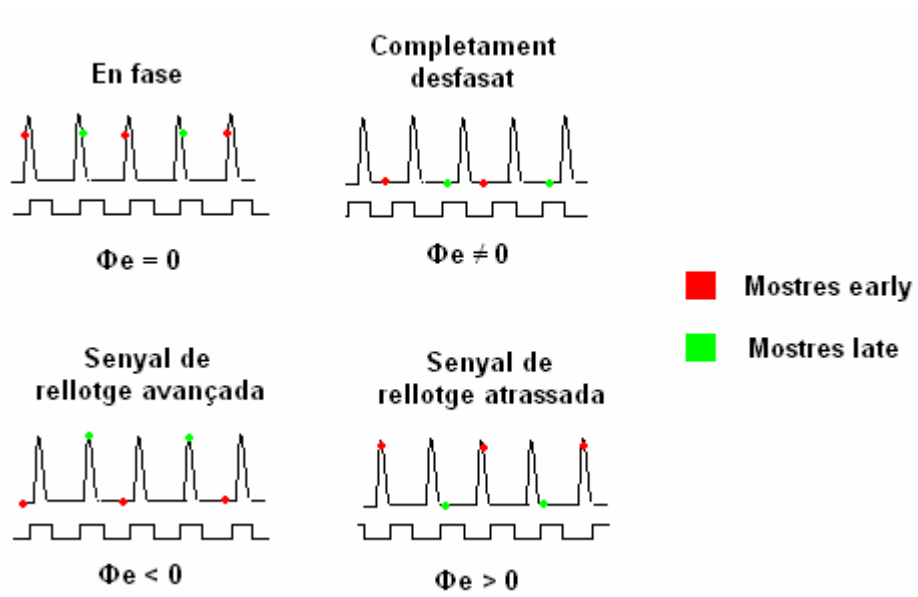


Fig.5.10. Diferents errors de fase per les senyal rebuda i de rellotge

La diferència entre una mostra early i una mostra late és proporcional a l'error de fase.

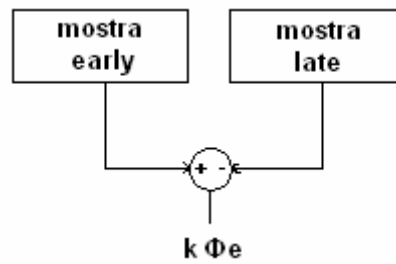


Fig. 5.11 Càlcul de l'error de fase

La característica d'aquest detector és la següent:

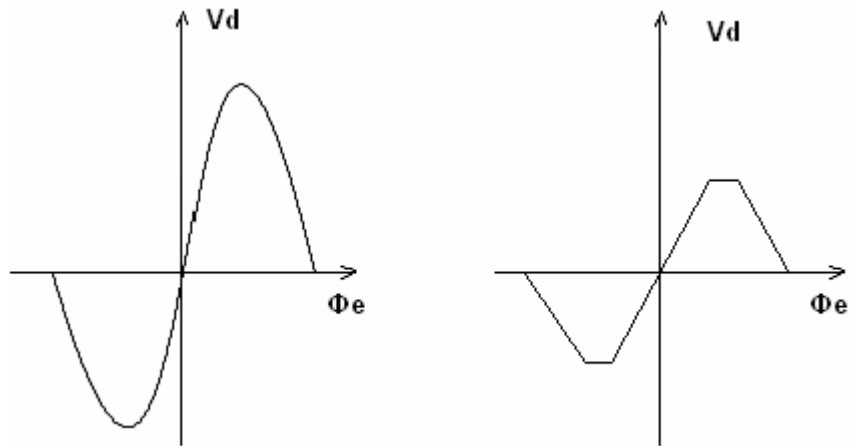


Fig. 5.12. Diferents característiques del detector de fase per a senyals diferents. En el primer cas per a polsos arrodonits i en el segon cas per a polsos trinagulars

L'avantatge d'aquest detector és que permet al PLL enganxar amb error de fase 0 degut a que la seva característica té pendent positiu quant l'error de fase val 0. A més és més fàcil d'implementar que l'implícit i per tant es presenta com un bon detector de fase.

Un inconvenient que es veurà més endavant és que per entrar en la zona de pendent positiu s'ha de fer amb l'ajuda d'un altre detector auxiliar.

5.4. Disseny teòric i simulació del model sobre *Simulink*

Ara ja tenim 3 detectors de fase que sembla que puguin enganxar amb error de fase 0. Abans de posar-nos amb la pràctica, ens hem volgut assegurar que el model de receptor que hem dissenyat funciona. És important simular un model abans d'intentar implementar-lo físicament si es pot fer. Això permet detectar errors i perfeccionar el model entre altres coses abans d'implementar-lo. En aquest cas per simular el comportament del receptor s'ha fet en l'entorn *Simulink*.

5.4.1. Condicions de treball de baixa freqüència

Per poder simular el comportament del receptor amb el *Simulink*, primer caldrà definir les característiques del senyal transmès, és a dir, quina serà la amplada dels polsos que el receptor haurà de rebre, el període de separació entre aquests polsos, els nivells i altres característiques.

Com que el que volem és aconseguir la sincronització del receptor, la freqüència del senyal d'entrada no és crítica. Teòricament hauria d'anar bé en totes les freqüències.

Per motius pràctics, s'ha triat una freqüència baixa, concretament de 100 KHz. És a dir que en el primer prototip, els polsos arribaran cada:

$$\frac{1}{100\text{KHz}} = 10\mu\text{s}$$

L'amplada dels polsos, s'ha triat de 10 vegades menys que el període de repetició, així els polsos duraran 1us. A més s'ha fet l'estudi amb polsos triangulars perfectes. Això facilita moltes coses pel fet de que els pendents són lineals.

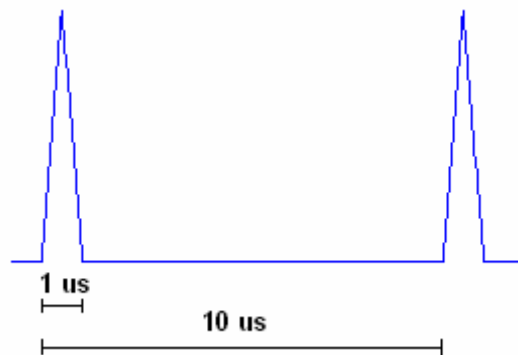


Fig. 5.13 Polsos triangulars que imiten a un senyal UWB

Els nivells reals de la senyal no es poden preveure, perquè depenen de moltes coses, com la distància entre l'emissor, les interferències, o la potència de l'emissor. Per tant el receptor haurà de funcionar d'igual forma si rep una senyal gran o una senyal petita. Per tal de poder fer la simulació cal establir uns nivells. Aquests es podrien triar aleatòriament, però per motius pràctics, s'han triat els nivells de 0V a 1V.

5.4.2. Generació de la senyal amb el *Matlab*

Per simular el comportament del receptor des del *Simulink* cal generar la senyal descrita a l'apartat anterior i guardar-la en una variable de l'entorn de treball del *Matlab*, de manera que s'hi pugui accedir des del *Simulink*. El tros de codi que genera aquesta senyal i la guarda a la variable 'simusignal', és el de l'*annex A*. La variable 'simusignal'

és un array de 2 vectors, un amb valors de temps i un altre amb valors d'amplitud, que després es farà servir en el model del receptor en el simulink.

Aquesta és la senyal generada i que es farà servir per a la simulació del sistema de sincronisme.

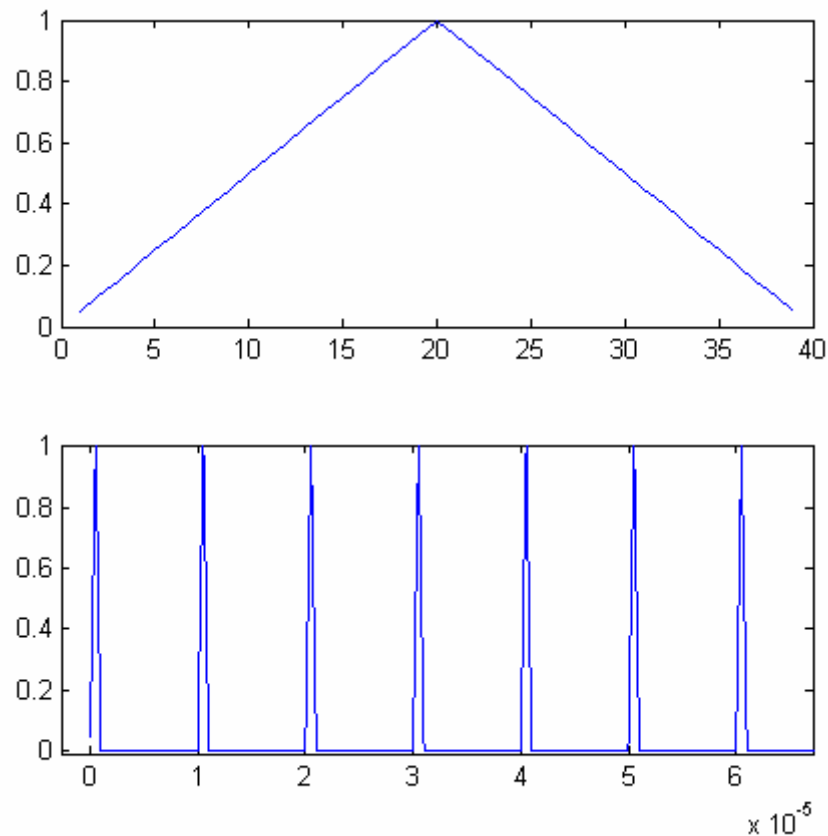


Fig 5.13. Vista d'un pols i tren de polsos triangulars generat amb el Matlab

5.4.3. La tècnica de sincronisme 'Single- Flank'

Aquest és un dels models proposats per al sistema de sincronisme del receptor. El seu funcionament es basa en el detector de fase 'track & hold'. Aquest és l'esquema implementat amb el *Simulink*.

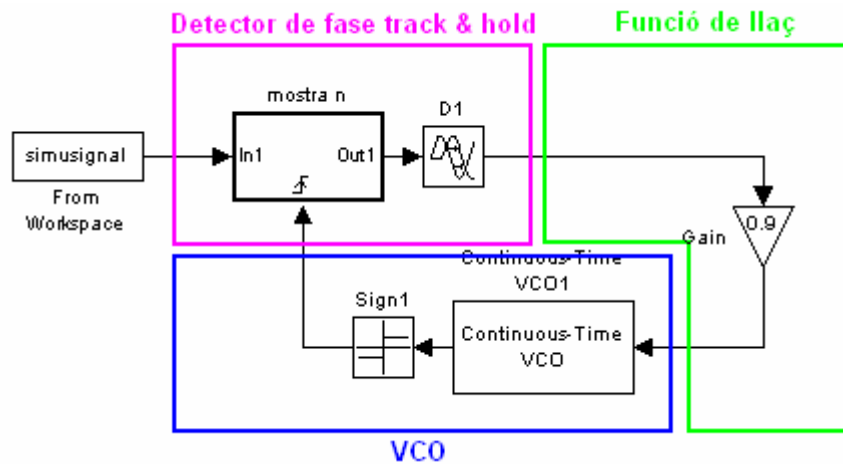


Fig. 5.14. Model implementat amb el Simulink que fa servir la tècnica 'Single Flank'

Com es pot veure, l'esquema és molt senzill. El detector de fase track & hold, que es un simple mostrejador o flip-flop, està connectat amb el VCO a través d'un guany per donar lloc a un PLL.

El funcionament és el d'un PLL corrent, un exemple per entendre'l pot ser el següent: suposem que inicialment la senyal rebuda i la senyal de rellotge generada pel VCO estan en un error de fase determinat molt pròxim a zero, de tal manera que el PLL està enganxat. En aquest estat el mostrejador està prenent mostres en algun punt del flanc de pujada dels polsos rebuts. Llavors, per qualsevol causa, una de les fases, ja sigui la del senyal de rellotge o la del senyal rebut, canvia. En aquest moment les mostres tindran un altre valor. És aquest mateix valor que, després de ser multiplicat per una constant (guany de la funció de llaç), farà que el VCO generi una freqüència que torni a fer que l'error de fase sigui 0.

A continuació es mostren els resultats d'algunes de les simulacions que s'han fet per a diferents guanys de la funció de llaç. En totes, la sensibilitat del VCO és de 2KHz/V i la freqüència central de 99.8 KHz

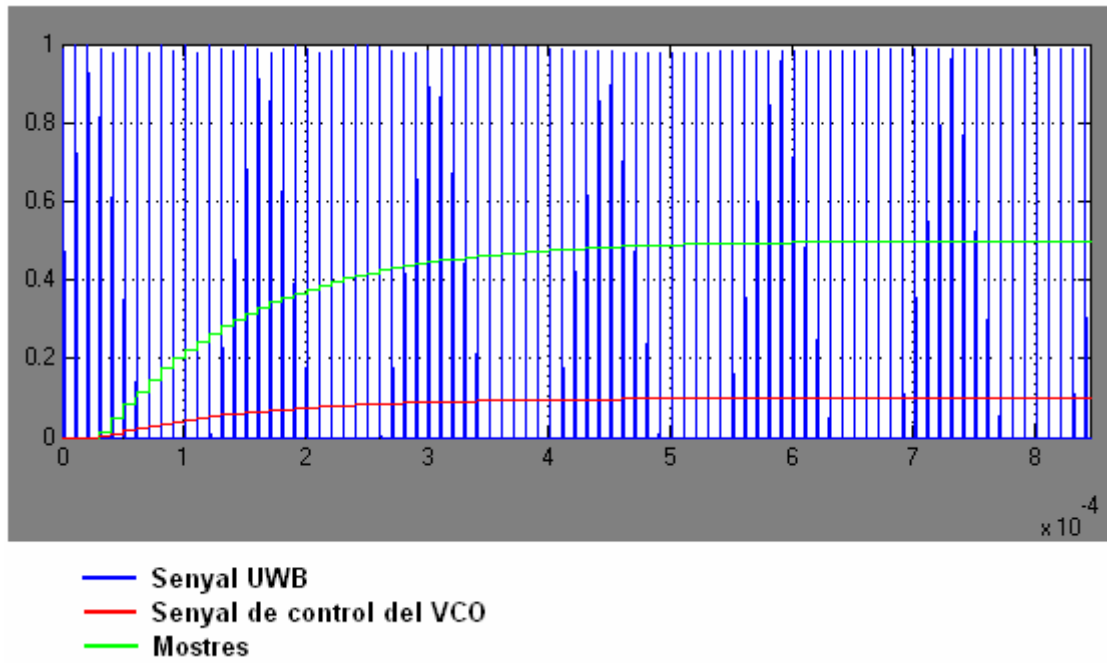


Fig. 5.15. Resultat de la simulació amb guany =0.2

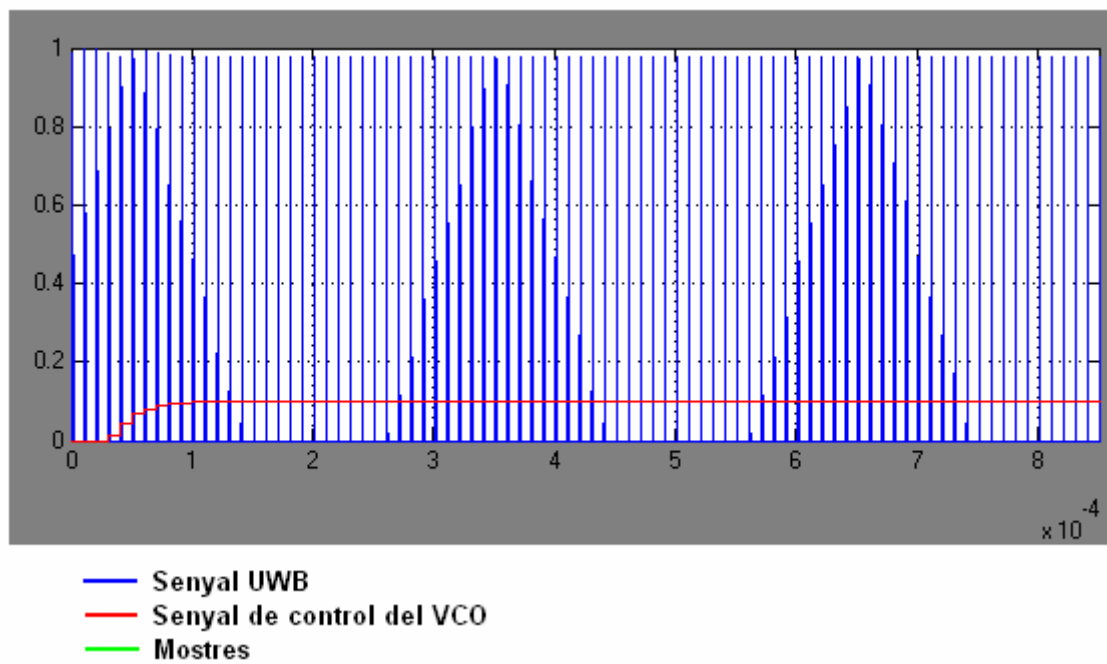


Fig. 5.16. Resultat de la simulació per un guany =1

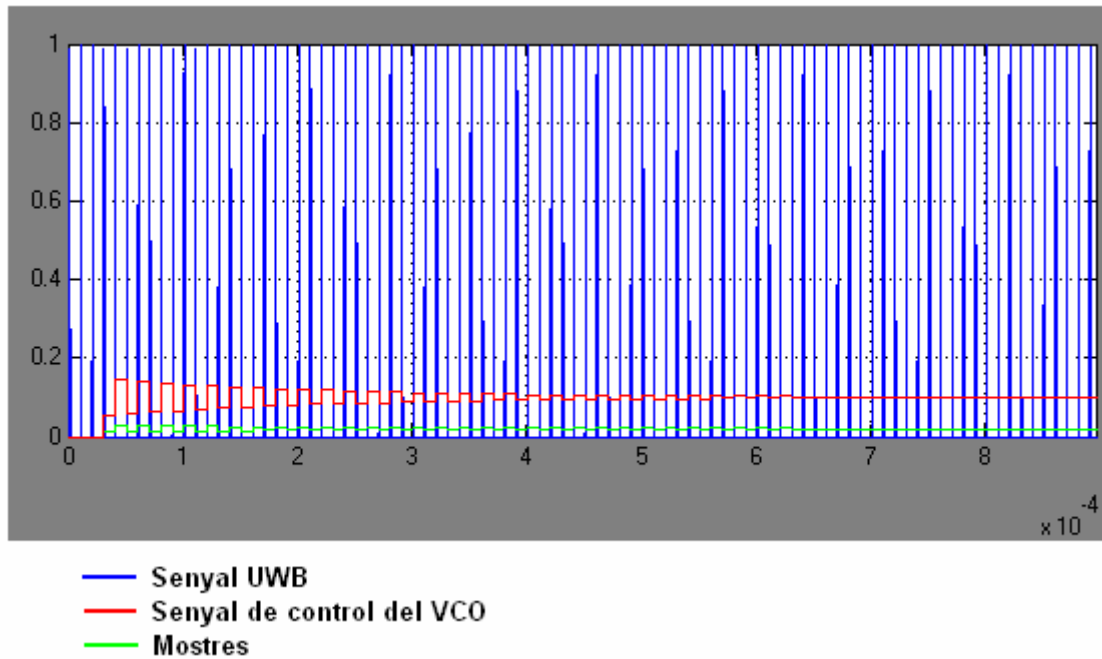


Fig. 5.17. Resultat de la simulació amb guany = 5

Com es pot veure en les simulacions, com més guany té la funció de llaç, més ràpida és la resposta, fins i tot, en l'últim cas té sobreimpulsos.

El problema que presenta aquesta tècnica és que no permet enganxar amb error de fase 0, sinó que, en funció del guany, el PLL enganxa amb un error de fase o un altre. Un altre problema que presenta i que no es reflexa en les simulacions és que aquesta tècnica depèn dels nivells de senyal per a funcionar correctament. Això vol dir que si els nivells canvien, la tècnica deixa de ser vàlida. Per altra banda presenta l'avantatge que és molt simple d'implementar. A continuació es presenten dos tècniques de sincronisme que intenten resoldre aquests problemes.

5.4.5. Tècnica de sincronisme amb el detector de fase implícit

Aquesta tècnica no té antecedents i és completament experimental. El seu funcionament és basa en el detector de fase implícit vist en apartats anteriors.

L'esquema d'aquest model en el simulink és el que segueix a continuació:

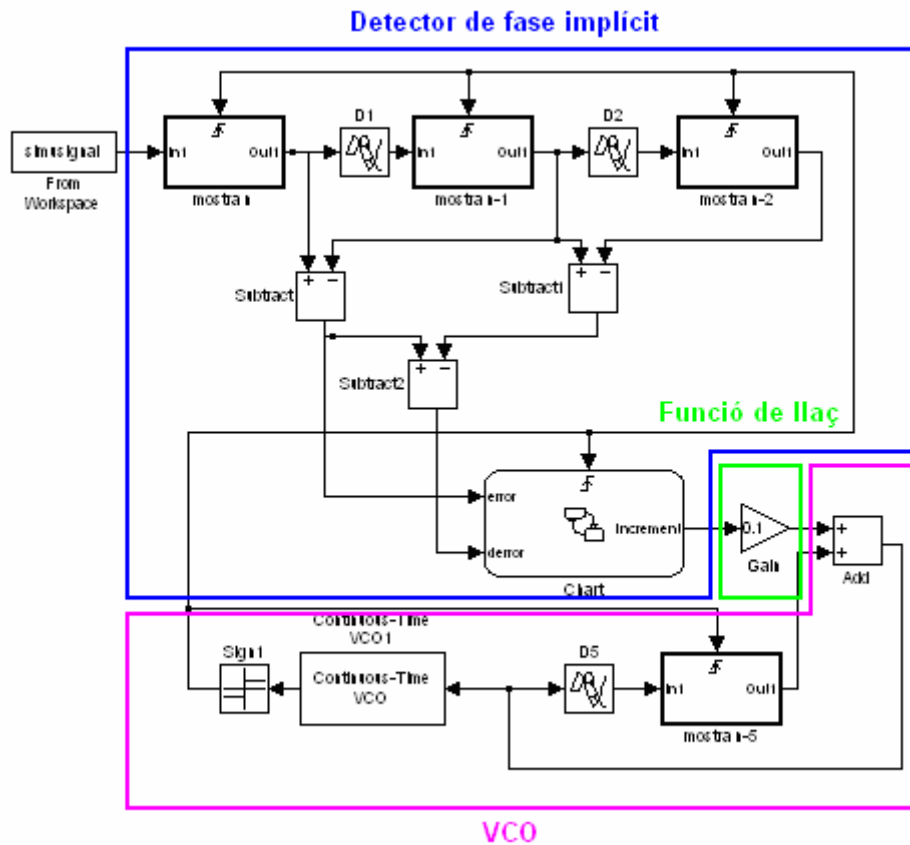


Fig. 5.18. Model implementat amb el Simulink amb el detector de fase implícit

El detector de fase implícit i un VCO estan connectats formant un PLL; en aquest cas, un PLL discret. Un sistema discret és aquell en que el valor de les variables només s'actualitza un cop cada cicle de rellotge, com en aquest cas.

Cada cop que es produeix un flanc de pujada del senyal de sincronisme, la màquina produeix canvis a la sortida de manera incremental. Si l'increment és 0, el valor de la tensió de control del VCO serà el mateix que l'anterior. Depenent de la sensibilitat del VCO i del guany de la funció de llaç, els increments diferents de 0 tindran un efecte o un altre sobre la freqüència generada pel VCO.

El que es vol aconseguir es que, quant la màquina fagi un increment en la seva sortida, la freqüència generada pel VCO canviï de tal manera que passi a mostrejar el següent pols una mica abans o una mica després en relació amb l'últim mostrejat. Però quant abans o quant després? Això és molt difícil imposar-ho teòricament, per això és millor fer simulacions i anar provant diferents valors. Inicialment s'han deixat els valors per defecte de cada component i s'apreciava com els canvis en la senyal de control eren

massa grans i no s'aconseguia el sincronisme. Més tard i a ull s'han fixat la sensibilitat del VCO a 2000 Hz/V i la seva freqüència central a 99.9KHz i el guany de la funció de llaç a 0.1 Així un increment en una unitat fet per la màquina d'estats afecta a la freqüència de la següent manera:

$$1 \cdot \text{guany} \cdot \text{sensibilitatVCO} = 1V \cdot 0.1 \cdot 2000 \frac{\text{Hz}}{\text{V}} = 200\text{Hz}$$

Llavors la pròxima mostra es prendrà al cap de:

$$\frac{1}{100\text{KHz} + 200\text{Hz}} = 9.98\mu\text{s}$$

Que amb relació a l'últim pols mostrejat, s'ha mostrejat abans:

$$10\mu\text{s} - 9.98\mu\text{s} = 20\text{ns}$$

Que sobre un pols de 1us, representa un

$$\frac{20\text{ns}}{1\mu\text{s}} \cdot 100 = 2\% \text{ abans}$$

Llavors ja hem aconseguit el que volíem, que era mostrejar el pols abans o després depenent del que fes falta en cada moment. Efectivament això també es compleix en cas contrari; si la màquina d'estats decremetés la seva freqüència en 1 unitat, les mostres es prendrien 20ns més tard.

A continuació segueixen les simulacions per aquest model amb diferents valors del guany de la funció de llaç.

S'han fet varies simulacions amb diferents guanys de la funció de llaç però a continuació només es presenten les més significatives. Totes s'han fet amb la freqüència central del VCO de 99.9 KHz i amb una sensibilitat de 2KHz/V.

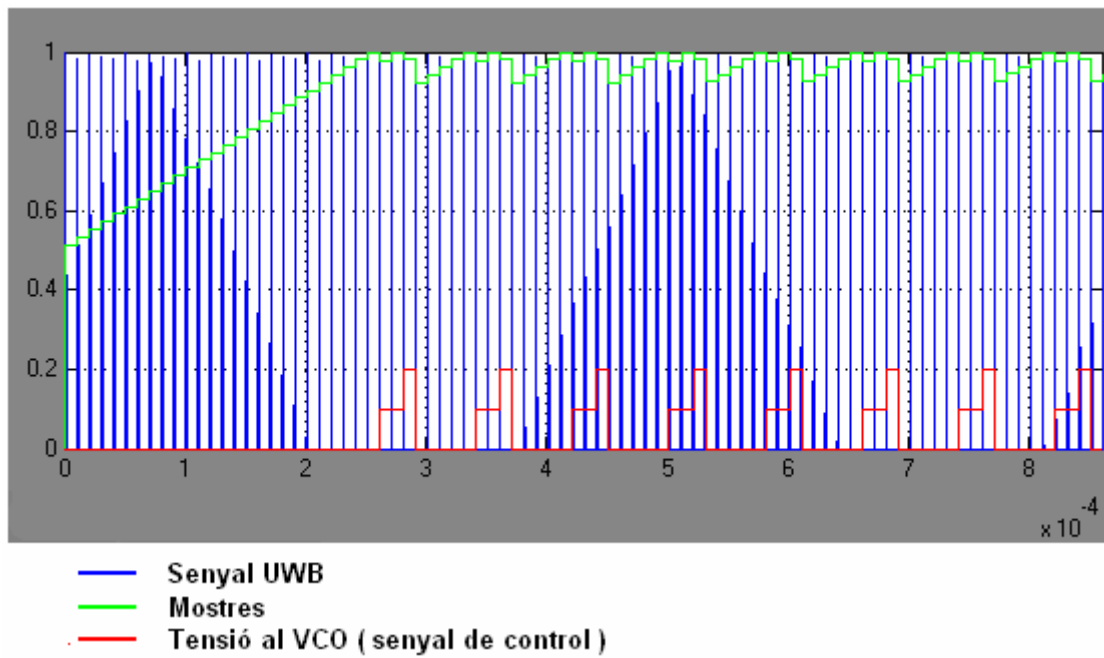


Fig. 5.19. Resultat de la simulació amb un guany = 0.1

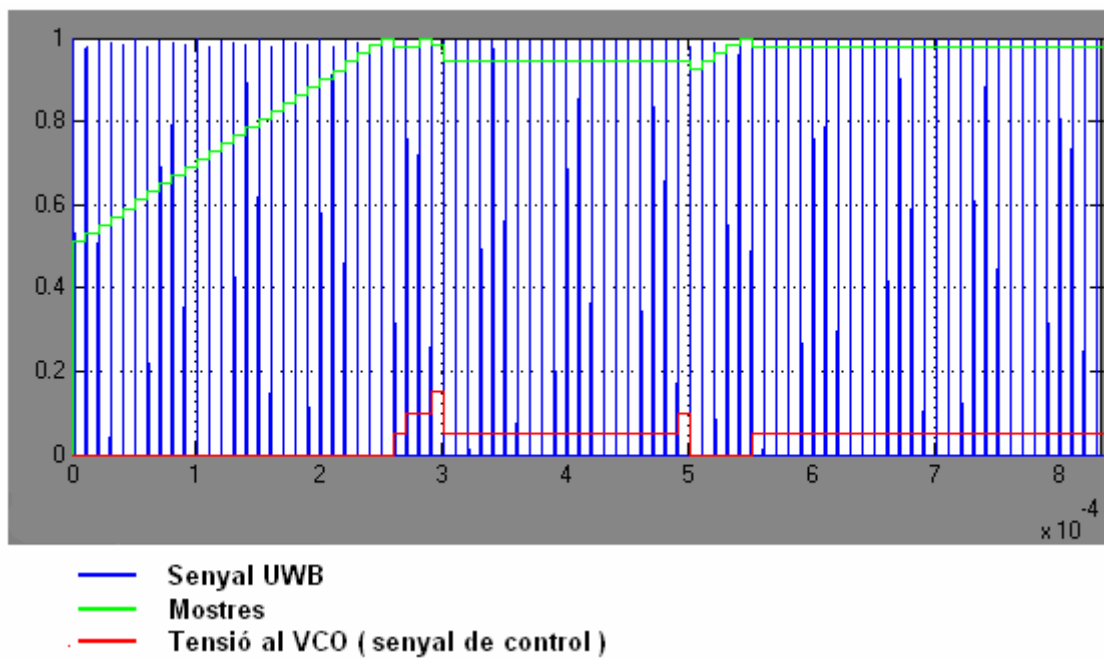


Fig. 5.20. Resultat de simulació amb un guany = 0.05

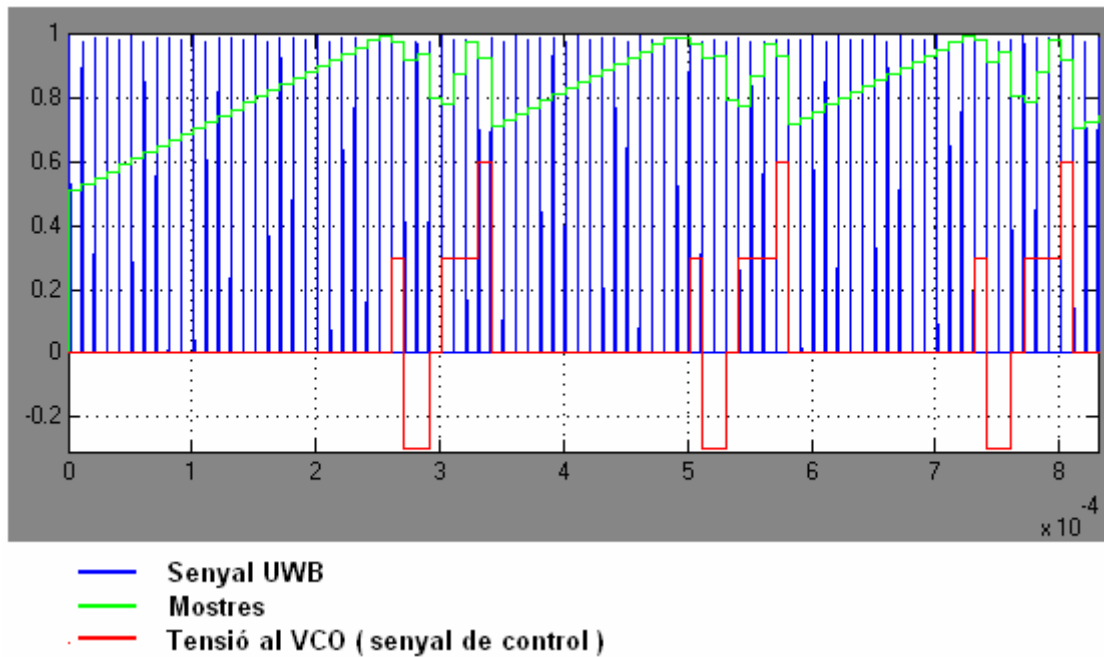


Fig.5.21. Resultat de la simulació amb un guany = 0.3

En totes les simulacions, s'aprecia com les mostres es prenen a una freqüència diferent a 100KHz, en aquest cas a 99.9KHz fins a $2.5 \cdot 10^{-4}$ s. En aquest moment la màquina d'estats, que fins ara estava a l'estat anembe, passa a altres estats per donar lloc al control del sincronisme.

S'observa que com més gran és el guany de la funció de llaç, pitjor són els resultats, però això no vol dir que el millor guany en sigui un de molt petit. Si el guany fos molt petit, el canvi fet a la tensió de control seria tant petit que no es notaria. Una manera de solucionar aquest problema podria ser implementant un control de guany, de tal manera que, quant la freqüència del senyal de rellotge i la freqüència del senyal rebut siguin molt pròximes, el guany sigui petit i que quant estiguin molt separades, el guany sigui més gran.

Per saber si les freqüències són molt diferents o no s'ha fet amb el valor de la variable error, que s'obté fent la diferència entre la mostra actual i la mostra anterior. Si el valor absolut d'aquesta variable és gran, la diferència entre freqüències també ho és, i si és petit, la diferència també és petita. Per tant, per valors de la variable error petits, el guany haurà de ser petit i per valors grans, el guany haurà de ser gran.

Per implementar aquest control s'ha fet així:

$$\text{increment} = \text{abs}(\text{error}) * (\pm 1 \text{ ó } \pm 2) \pm (0.01 \text{ ó } 0.02)$$

Els valors i els signes de la fórmula depenen de l'estat en que es troba la màquina d'estats veure *fig. 5.18*. El control del guany en aquest cas no es fa des de la funció de llaç sinó des de la variable increment. L'efecte acaba sent el mateix perquè les dues variables es troben multiplicant.

Aquest és el resultat de la simulació del model amb el guany variable.

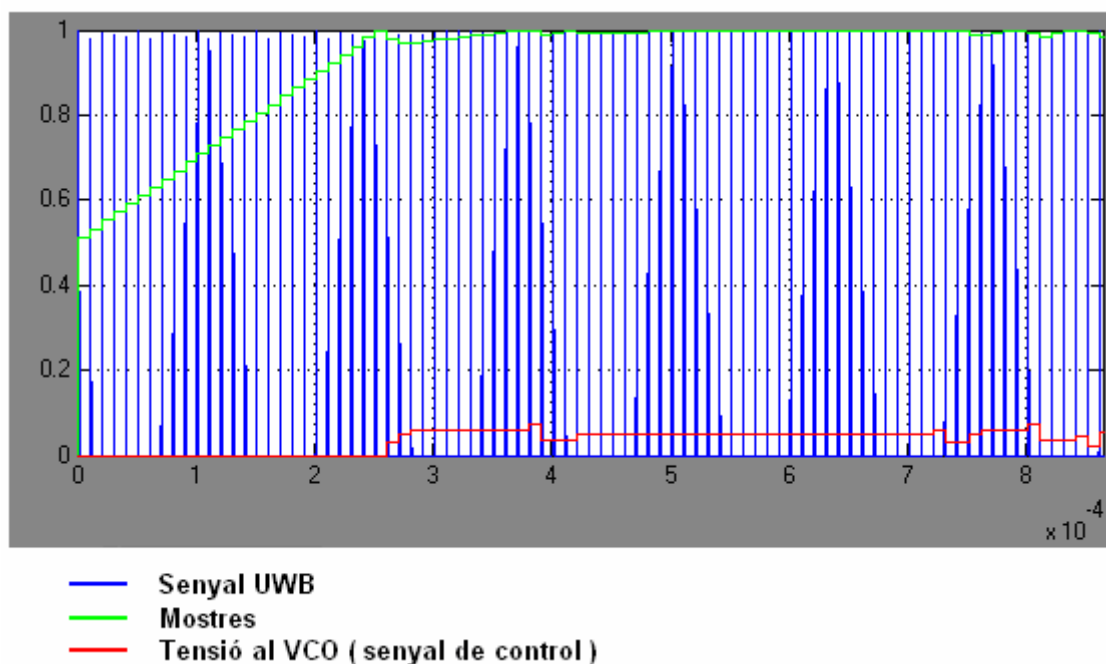


Fig. 5.22. Resultats de la simulació amb guany variable

En aquesta simulació s'aprecia com les mostres es prenen a una freqüència diferent a 100KHz, en aquest cas a 99.9KHz fins a 2.5×10^{-4} s. En aquest punt la màquina d'estats comença a fer canvis a la tensió de control, que a diferència de les simulacions anteriors son proporcionals a l'error entre una mostra i la anterior. Això fa que el control del sincronisme sigui molt millor i que, com es pot veure, els polsos es mostregin en el seu màxim.

5.4.6. La tècnica de sincronisme ‘Early- Late’

Aquesta tècnica, així com ho és la de single-flank, també és coneguda. A diferència de la tècnica anterior, el funcionament d’aquest model es basa en un altre dels detectors de fase vistos a l’apartat del PLL: el ‘early-late’.

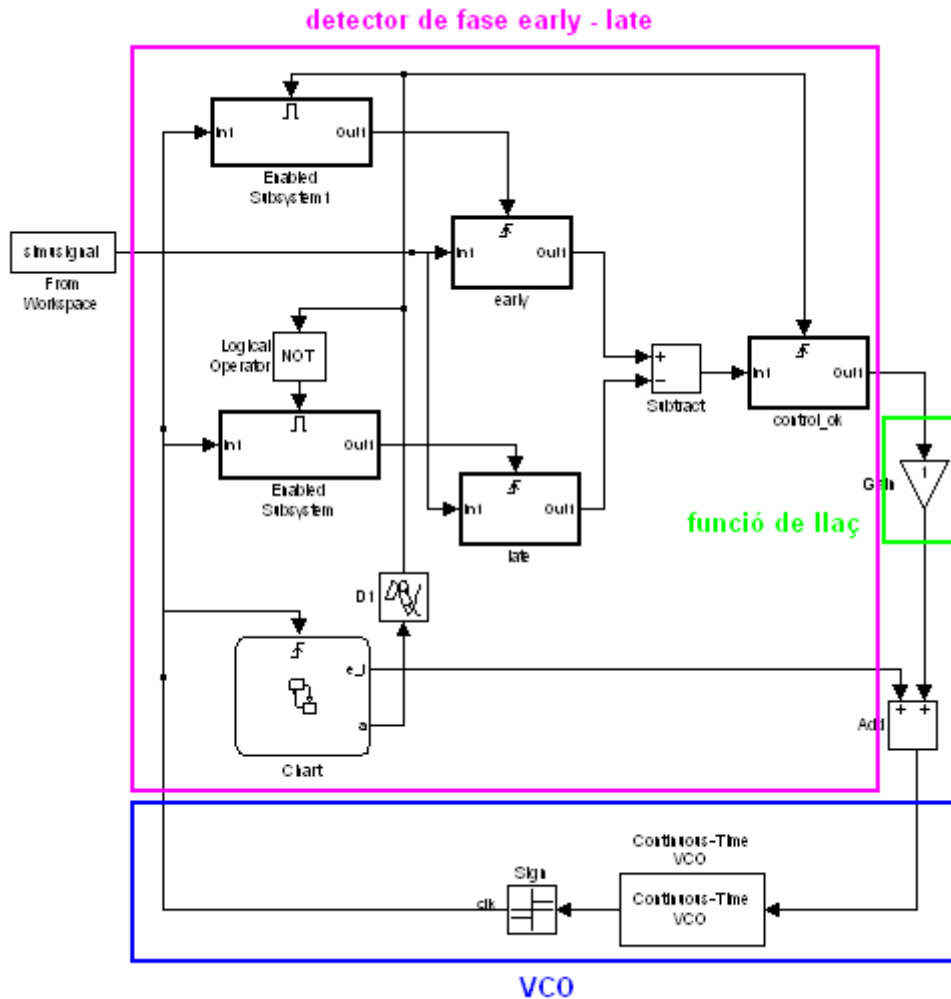


Fig.5.23. Model implementat amb el Simulink que fa servir la tècnica Early-Late

El muntatge vol venir a ser un PLL i com a tal té un detector de fase, un VCO i una funció de llaç. El detector de fase que s’ha fet servir en aquest model és l’early-late, que com s’ha vist en apartats anteriors informa directament de l’error de fase. Això és un gran avantatge ja que per controlar la tensió del VCO només cal aplicar-li un cert guany. En el detector implícit això no passa ja que per poder determinar la tensió de control s’havien de mirar estats anteriors i altres variables.

Així doncs, tot i ser un model de sistema discret perquè les variables només s'actualitzen cada cicle de rellotge, el control del sincronisme es farà 'sol', sense la necessitat de aplicar cap tipus d'algorisme.

El funcionament és el següent: suposem que inicialment la senyal de rellotge (generada pel VCO) i la senyal rebuda estan en fase. D'aquesta manera les mostres early i les mostres late es prenen a la mateixa alçada del pols, amb la diferència que les early es prenen abans del màxim i les late, després. En aquest moment la diferència entre les mostres early i les late val 0. Si per a qualsevol motiu una de les fases canvia, les mostres early i les late ja no es prendran a la mateixa alçada del pols sinó que ho faran en diferents punts. Llavors la diferència entre les mostres early i les mostres late prendrà un valor diferent de 0 que, multiplicat pel guany de la funció de llaç, donarà lloc a la tensió de control del VCO. Aquesta tensió farà que el VCO generi una senyal amb freqüència tal que l'error de fase entre aquesta senyal i la rebuda torni a ser 0.

La màquina d'estats que es pot veure a la figura és la que controla els cicles de early i late.

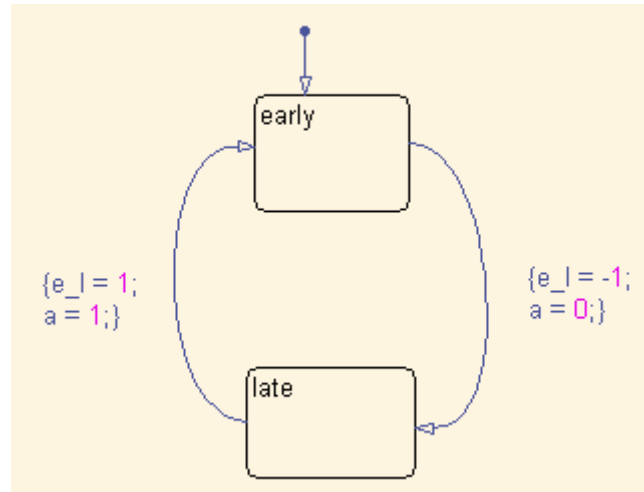


Fig. 5.24. Màquina d'estats que commuta entre els estats early i late

És una màquina molt simple que té 2 estats, un estat early i un estat late.

Té dues sortides cap al simulink, la e_1 i la a. La e_1 és la que es fa servir per poder prendre les mostres early i late, això s'aconsegueix sumant el seu valor a la senyal de control del VCO. Així, quant aquesta variable val 1, la senyal generada pel VCO és de freqüència superior i per tant la pròxima mostra serà early. Efectivament durant l'estat

early aquesta variable val 1 i durant l'estat late val 0. D'aquesta manera s'aconsegueix alternar entre una mostra early i una mostra late però el valor mig de la senyal no es veu afectat. Per altra banda, la variable a diu quina mostra s'ha de prendre, si early ($a = 1$) o late ($a = 0$). A més en el flanc de pujada d'aquesta variable, s'actualitza la senyal de control del VCO. Això és lògic que es fagi cada 2 cicles perquè per poder tenir la informació d'error de fase, s'ha de disposar de les dos mostres.

Simulació del model

A continuació es presenten els resultats d'algunes de les simulacions que s'han fet per a diferents guanys de la funció de llaç, totes elles s'han fet amb una sensibilitat del VCO de 2KHz/V i una freqüència central de 99.9KHz.

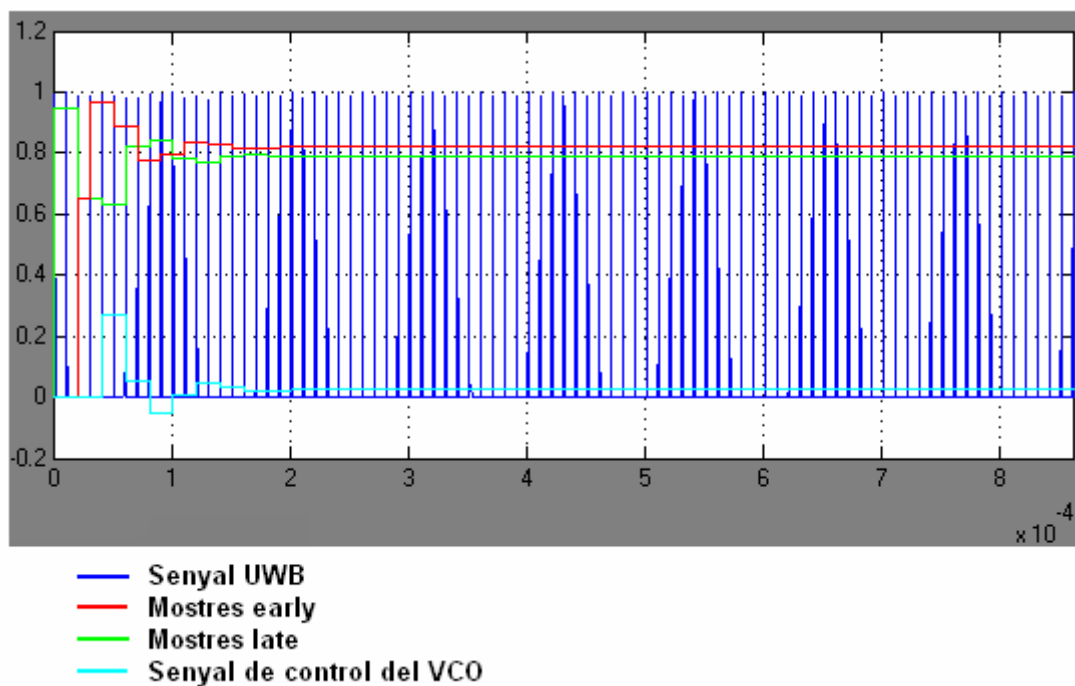


Fig. 5.25. Resultat de la simulació amb guany = 0.8

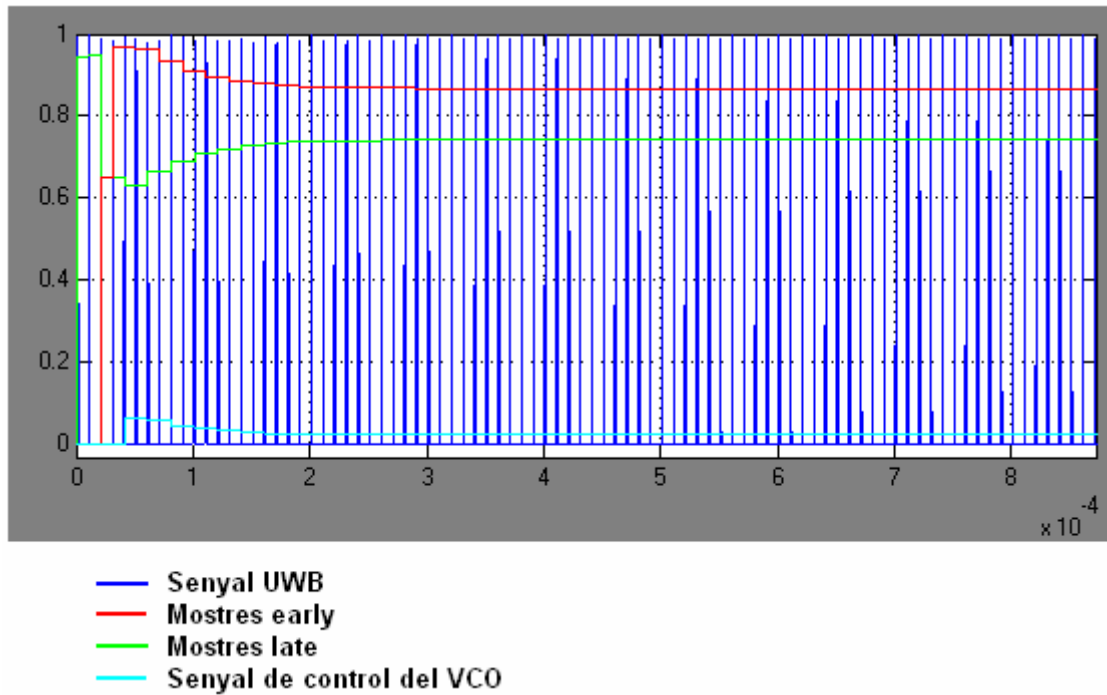


Fig. 5.26. Resultat de la simulació amb guany = 0.2

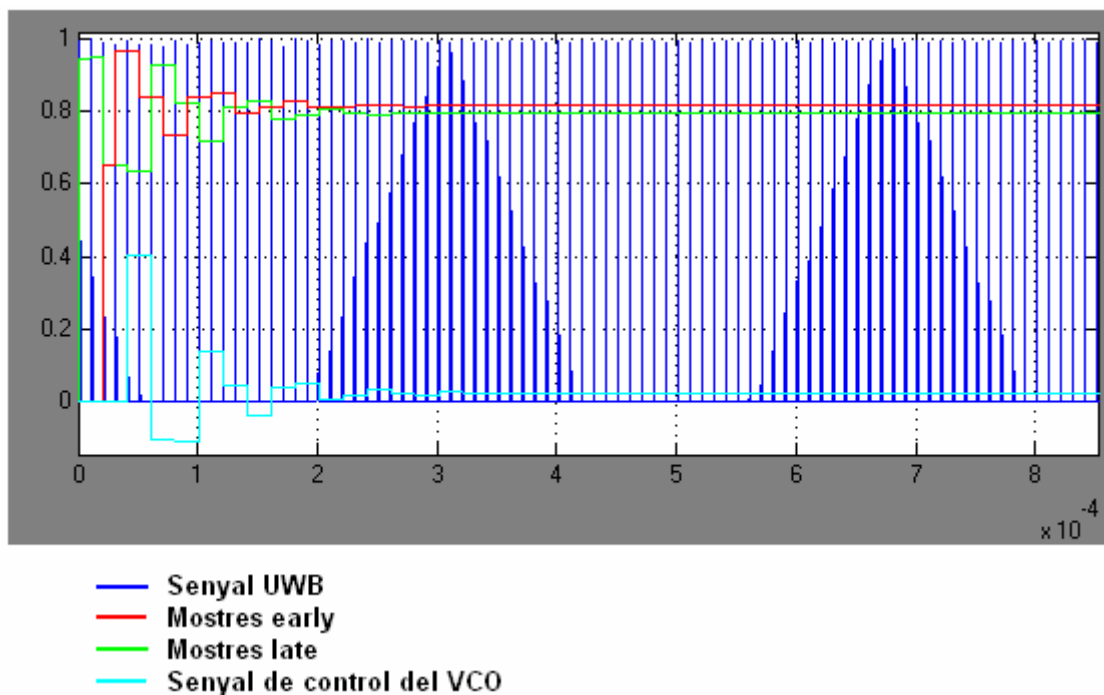


Fig. 5.27. Resultat de la simulació amb guany = 1.2

Com s'aprecia en aquestes simulacions, per a diferents valors de guans tenim diferents respostes. Un guany petit fa que el sistema no tingui sobreimpulsos en la seva resposta i que aquesta sigui una mica més lenta que amb un guany més gran, en aquest cas però la resposta sí que té sobreimpulsos. En tots els casos però, com que la funció de llaç

només implementa un guany proporcional, hi ha un error d'establiment. Aquest problema s'hauria de resoldre amb un guany integral.

Aquesta simulació s'ha fet amb una funció de llaç que implementa un guany proporcional de 1 i un guany integral de 5000.

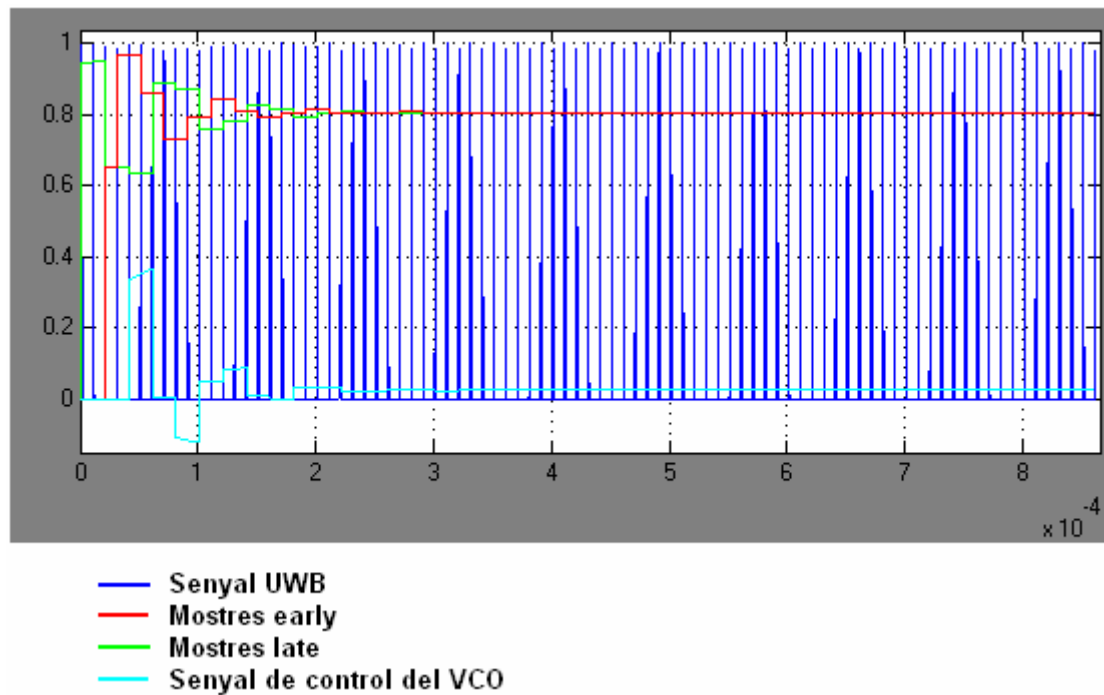


Fig. 5.28. Resultat de la simulació amb acció proporcional i integral

Amb l'acció integral s'aconsegueix corregir l'error d'establiment.

Com s'aprecia en aquestes simulacions, inicialment l'error de fase és 0 o molt pròxim a 0. Això s'ha fet expressament perquè si no fos així, aquest model no seria capaç d'enganxar. Això passaria perquè la característica del detector de fase early-late només té un rang en que el pendent és positiu, en els altres llocs és negatiu o és nul. Això és un gran inconvenient perquè es necessitarà un altre detector de fase auxiliar que engegui el principal quan l'error de fase es trobi en aquest rang de pendent positiu.

6. Disseny pràctic del sistema de sincronisme

Un cop s'ha comprovat que les diferents tècniques de sincronisme funcionen amb el Simulink, s'ha procedit a implementar els models físicament. En aquest apartat es pretenen aconseguir els mateixos models que els simulats en Simulink, però amb components reals. A continuació s'explica el procés de disseny pràctic i implementació física del sistema.

Per a poder portar a la pràctica els models dissenyats i simulats, cal trobar l'equivalent real de cada bloc del *Simulink*. En general l'esquema que segueixen tots els models proposats anteriorment és aquest:

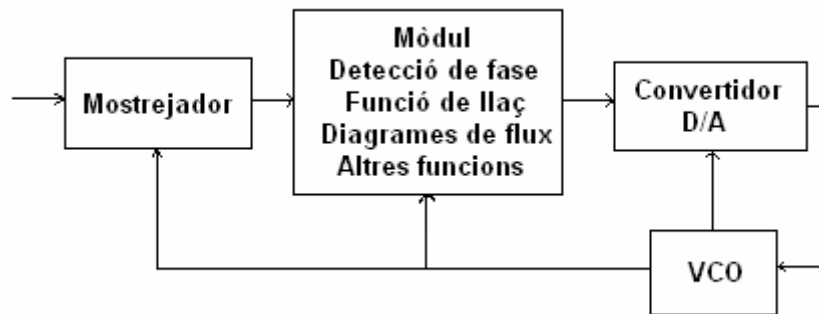


Fig.6.1. Diagrama de blocs del sistema real

L'objectiu d'aquest apartat és aconseguir un sistema real que compleixi aquest esquema general.

6.1. Etapa de mostreig, el convertidor A/D

Aquesta etapa ha de llegir el senyal d'entrada en el moment que li diu el senyal de rellotge i passar el valor al següent mòdul. Com es veurà en el següent apartat, el següent mòdul és una FPGA i necessita les dades en digital. Per tant d'això se'n encarrega un convertidor AD. Els criteris per a triar el convertidor seran: el convertidor ha de tenir com a mínim una entrada analògica i 8 bits com a sortida digital. Per anar bé hauria de consumir poc i hauria de poder treballar a la freqüència que nosaltres volem.

Un convertidor que compleix aquests requisits i estava disponible al laboratori és el convertidor ADS831E de la casa Burr-Brown, que té les següents característiques:

- Entrada simple o diferencial
- 8 bits de sortida
- Consumeix molt poc
- Bona relació senyal soroll (SNR = 49 dB)
- Pot treballar fins a 80MHz
- El seu funcionament és pipelined i dona la conversió en digital 4 clk's més tard.



Fig. 6.2. Detall de l'encapsulat del convertidor

El seu timing de funcionament és aquest:

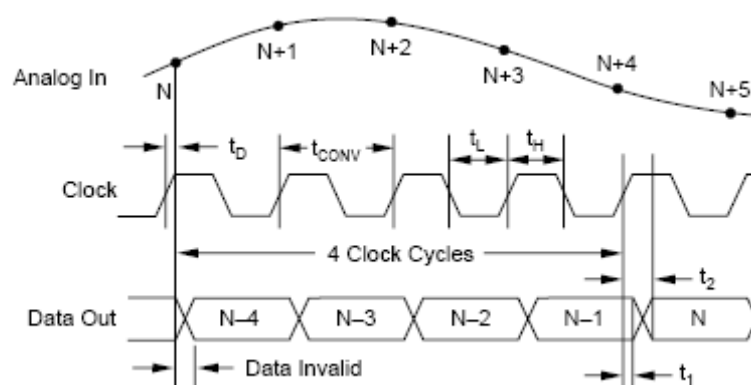


Fig. 6.3. Timing de funcionament del convertidor

Com es pot veure presenta un petit inconvenient i es que tarda 4 cicles de clock en donar la conversió a la seva sortida. Per evitar aquest problema, es fa treballar el convertidor 4 vegades més ràpid que la resta del sistema. Així, des de la FPGA, es rep la conversió en un sol cicle de retard. Això es veu més endavant.

El convertidor està configurat amb entrada simple referenciada a massa i amb referència de conversió interna.

Amb el convertidor alimentat a 3.3V, els valors digitals corresponents a diferents tensions d'entrada són els que es mostren a la següent taula:

Tensió entrada (V)	Valor digital (8 bits)
1.2	00000000
1.95	10000000
2.7	11111111

El convertidor ja estava muntat en una placa al laboratori de teoria del senyal i comunicacions i s'ha aprofitat aquesta per a fer el muntatge. L'aspecte d'aquesta placa és aquest:

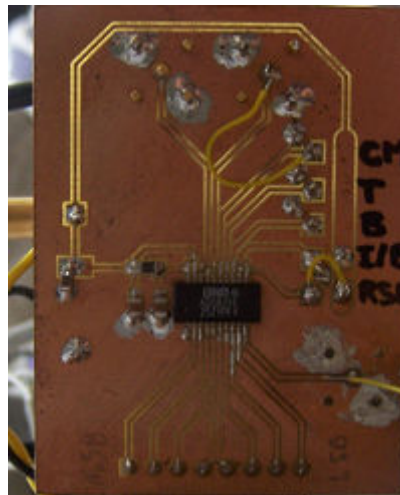


Fig. 6.4. Aspecte de la placa. En el centre es pot apreciar el convertidor.

Els pins de la placa principal (FPGA) en els que va connectada aquesta placa són els següents:

nº de bit de dades	Pin FPGA
7 (MSB)	68
6	62
5	60
4	58
3	56
2	54
1	52
0 (LSB)	50

Pin Placa	Pin FPGA
Clk	75

6.2. Etapa d'implementació de funcions digitals, la FPGA

Aquesta etapa és el nucli del sistema. Se'n ha d'encarregar de donar la senyal de control adequada al VCO perquè la senyal de rellotge estigui en fase amb la senyal d'entrada. En aquesta etapa s'han d'implementar funcions digitals com la detecció de fase o algorismes de flux. Això s'ha de fer en un sistema programable per tal de poder-lo implementar mida. En el nostre cas hem triat una FPGA perquè a diferència d'un PIC, aquesta fa totes les operacions demanades en un sol cicle de rellotge. És necessari que les operacions es facin en un sol cicle de rellotge, perquè en tractar-se d'un sistema de control, interessa eliminar tots els retards possibles.

La FPGA que es fa servir és la EP1C3T144C6 de la família Cyclone II de la casa Altera. Una FPGA (Field Programmable Gate Array), que vol dir 'matriu de portes lògiques programable' és un dispositiu en el que es pot definir el hardware lògic que l'usuari vulgui. El dispositiu es programa amb llenguatge VHDL (Verilog Hardware Definition Language) o altres llenguatges de més alt nivell. Per les FPGA de Altera, el mateix fabricant subministra un software per a programar-les, aquest és el Quartus II. En una FPGA s'hi pot programar des de una simple porta inversora fins a un microcontrolador sencer. En el nostre cas hi programarem la màquina d'estats i les operacions necessàries per fer el control de la tensió del VCO.

Aquesta FPGA de 144 pins té 2910 elements lògics i 59904 bits de memòria. A més permet treballar per sobre dels 200 MHz, així que està molt per sobre del que nosaltres necessitem.



Fig. 6.5. Detall d'una FPGA

La FPGA també estava muntada sobre una placa de caire general en el laboratori i s'ha aprofitat. L'aspecte de la placa és aquest:

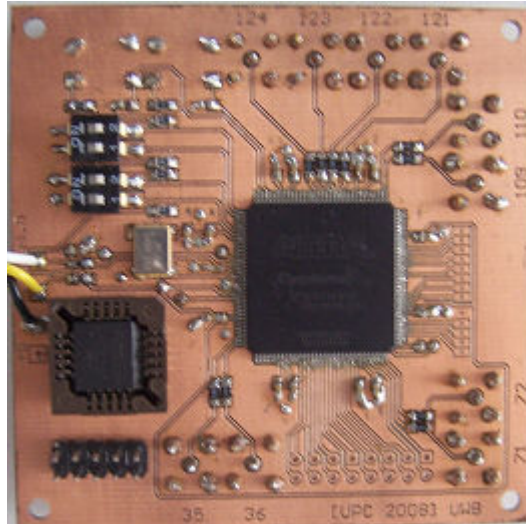


Fig.6.6. Aspecte de la placa de la FPGA. En el centre es pot apreciar la FPGA i a la part inferior esquerra el mòdul EPC2

L' alimentació d'aquesta placa és de 3.3V i de 1.5V, que són les tensions que necessita la FPGA per funcionar correctament. Els components perifèrics són condensadors per filtrar l'alimentació, pulsadors, interruptors i sortides coaxials. Aquests són components que fan la placa útil per a proves.

Aquesta placa també disposa d'un dispositiu EPC2, que guarda el programa i el grava a la FPGA cada cop que s'alimenta. La programació del dispositiu es fa amb el programador Ethernet Blaster, que també és de la casa Altera, directament sobre aquesta placa amb els 10 pins que es veuen a la part inferior esquerra de la placa. Aquesta placa és la que farà de base per a totes les altres plaques.

6.3. Adaptador entre la FPGA i el VCO: el convertidor D/A

Com que el VCO necessita una entrada de tensió analògica i la sortida de la FPGA és digital, es necessària una etapa per convertir el valor digital en valor analògic.

El component que s'ha triat per a fer això és un convertidor D/A. Aquest ha de ser capaç de passar les dades digitals a una tensió analògica. El criteri per elegir-lo es que ha de tenir com a mínim 8 entrades digitals, una sortida analògica en tensió i ha de poder treballar a la velocitat que li demanem.

El convertidor que s'ha triat és el DAHI3338 de la casa Intersil. Les característiques d'aquest component són les següents:

- Baix consum
- Sortida de tensió
- 8 bits d'entrada
- Entrades compatibles amb els nivells CMOS i TTL.
- Temps d'establiment de la sortida molt baix
- Pot treballar fins a 50 MHz.

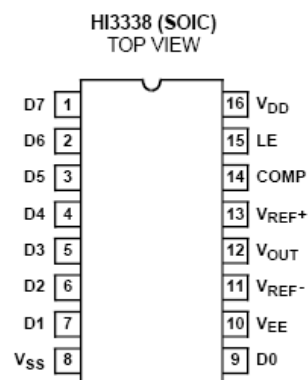


Fig. 6.7. Encapsulat del convertidor

L'esquema del mòdul convertidor D/A és el que ve a continuació. El circuit està format pel component principal, que és el convertidor D/A i uns quants perifèrics, com condensadors d'alimentació i una resistència que actua d'impedància de sortida.

Les línies de dades són les que entren per l'esquerra de l'esquema (JP1), la sortida analògica de tensió és V_{OUT} i el senyal de rellotge LE.

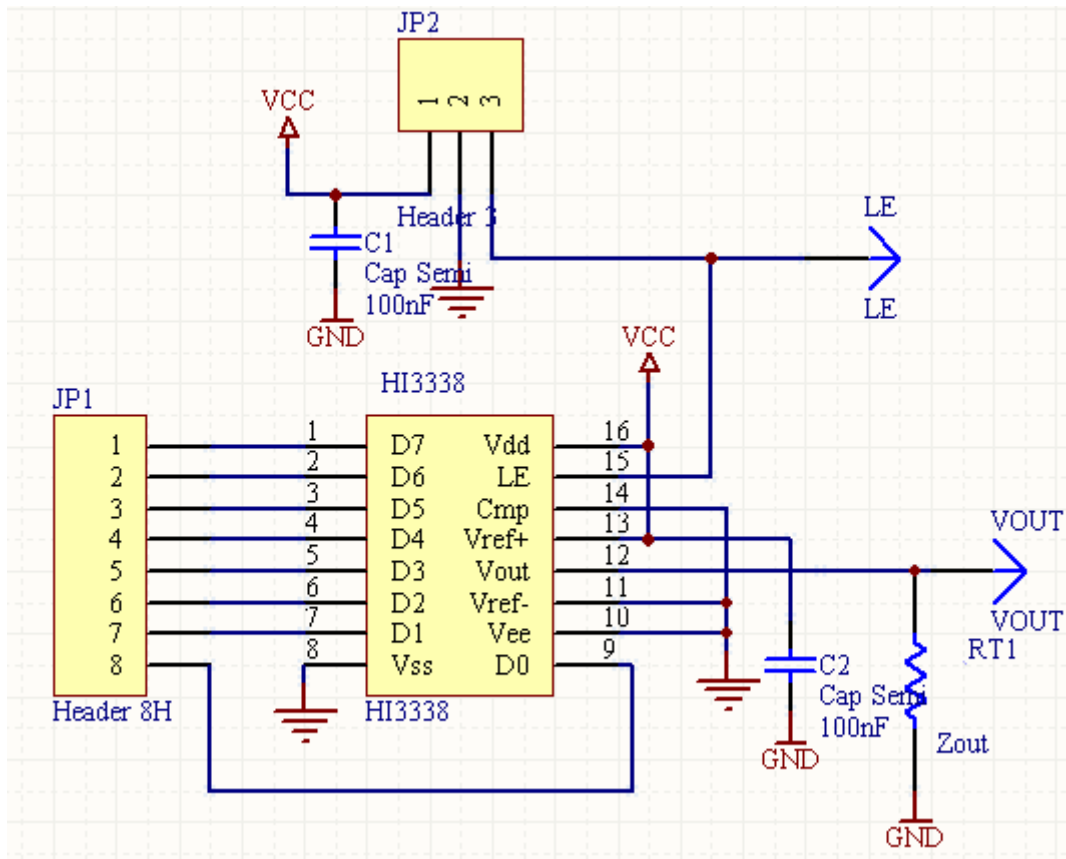


Fig. 6.8. Disseny de l'esquema de la placa en Protel DXP

El timing de funcionament d'aquest circuit és aquest:

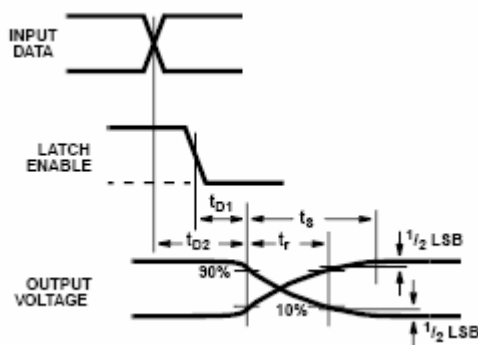


Fig. 6.9. Timing de funcionament del mòdul convertidor

Amb el convertidor alimentat a 3.3V, les tensions generades per cada entrada digital són les que es mostren a la següent taula:

Valor digital (8 bits)	Tensió generada (V)
00000000	0
10000000	0.55
11111111	1.1

La tensió de sortida està mesurada sobre la resistència Z_{out} , que és de 75ohms, tal i com el fabricant aconsella.

Aquest circuit s'ha hagut de muntar sobre una placa de circuit imprès perquè no se'n ha pogut aprofitar cap de feta. El disseny de la placa s'ha fet amb el protel DXP i s'ha fresat al laboratori. L'aspecte final de la placa amb els components soldats és aquest:

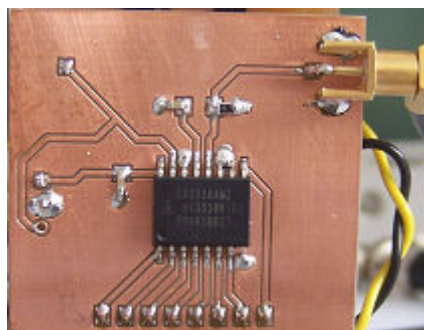


Fig.6.10. Aspecte de la placa del convertidor

Els pins de la placa de la FPGA en els que va connectada aquesta placa són els següents:

nº de bit de dades	Pin FPGA
7 (MSB)	49
6	51
5	53
4	55
3	57
2	59
1	61
0 (LSB)	67

Pin Placa	Pin FPGA
Clk	77

6.4. El mòdul VCO

La etapa del VCO és la encarregada de subministrar el senyal de rellotge a tots els mòduls anteriors. La freqüència d'aquest senyal és proporcional a la tensió d'entrada del VCO. Aquest component ha de ser capaç d'oscil·lar al voltant dels 400 KHz, això és una freqüència 4 vegades més gran que la del senyal de sincronisme necessari. S'ha fet així per poder obtenir les mostres del A/D amb un sol cicle de retard.

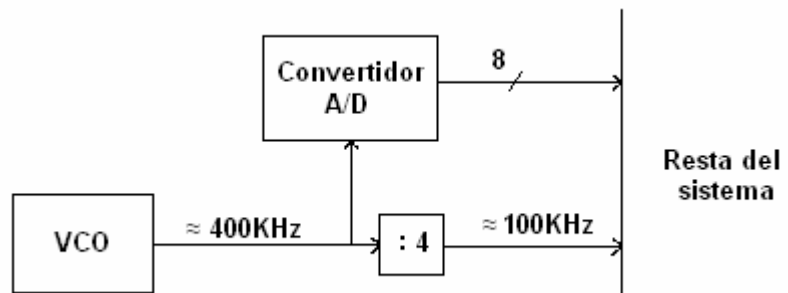


Fig.6.11. Freqüència del senyal de sincronisme en diferents punts del sistema

Hi ha varis models de VCO's disponibles al mercat. En el nostre cas n'hem de trobar un que oscil·li a prop dels 400 KHz . Una opció fàcil seria fer servir una font de tensió del laboratori com a VCO ja que aquestes permeten una modulació de la sortida en FM, que això acaba tenint el mateix comportament que un VCO. Es va fer la prova i s'observava com la senyal de sortida tenia una freqüència proporcional a la tensió aplicada a la entrada de modulació de la font. Així el comportament era perfectament el d'un VCO, però quant es va mesurar el temps de reacció es va veure que era massa gran i es va descartar. Més endavant s'explica com es va mesurar aquest temps.

Seguint amb la recerca d'un VCO real que s'assemblés al del *Simulink*, vaig mirar les característiques de tots els VCO's disponibles al laboratori i n'hi havia 2 que s'adapten a les nostres condicions. El primer model amb el que es va provar era un POS-150.



Fig.6.12. Aspecte del VCO POS-150

Aquest VCO està pensat per aplicacions RF i oscil·la al voltant dels 150 MHz, tot i que amb una tensió de control adequada es pot fer oscil·lar a 100 MHz. Per obtenir els 100 KHz només faria falta dividir la freqüència de rellotge entre 1000. L'inconvenient d'aquest VCO és que s'ha d'alimentar a 12V i la tensió que genera és alterna.

L'altre tipus de VCO que hi havia al laboratori és el 74HCT4046. Aquest integrat està pensat per actuar com a PLL, però com a tal té un VCO que es pot aprofitar, i així s'ha fet.



Fig. 6.13. Encapsulat del VCO 74HCT4046

La senyal de sortida d'aquest VCO és quadrada i va de 0 a 3.3V si s'alimenta a aquesta tensió. Això fa que sigui un senyal de rellotge perfecte per al nostre cas. Per això s'ha triat aquest VCO davant el POS-150. Aquest VCO té una sensibilitat aproximada de 31 KHz/V, això vol dir que per a cada volt que varia a la tensió d'entrada, la freqüència de sortida varia en 31 KHz.

La freqüència central del dispositiu es pot sintonitzar a través d'un condensador i un parell de resistències. El fabricant no dona cap fórmula per calcular-la a partir d'aquests valors però dona una gran quantitat de gràfics des dels que es pot deduir el valor d'aquests components per aconseguir una freqüència determinada. Com que no hi havia cap gràfic amb la freqüència de 400KHz, que és la freqüència que necessitem, s'ha anat

probant fins a aconseguir-la. Amb una entrada de 2V, els valors que fan oscil·lar el VCO a 400 KHz són: 5Kohms per la resistència i 100pF per al condensador. Aquest é l'esquema elèctric del VCO.

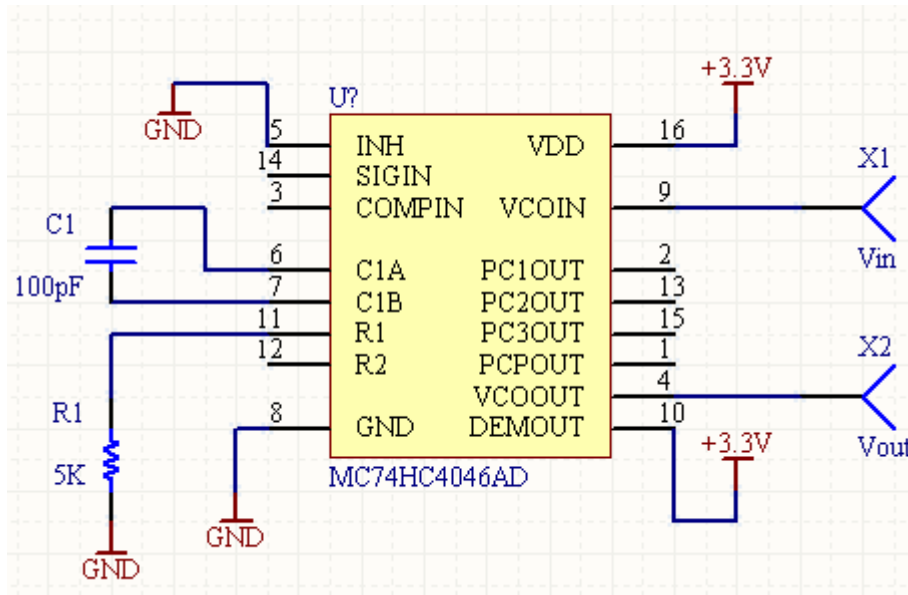


Fig.6.14. Esquema de connexions del VCO amb el protel DXP.

Un cop sintonitzat, aquesta és la característica del VCO

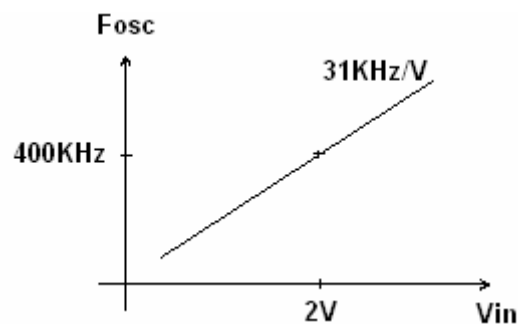


Fig. 6.15. Característica del VCO

Si connectéssim aquest VCO directament a la sortida del DA, la sensibilitat sobre un LSB seria:

$$sens = \frac{31KHz}{1V} \cdot \frac{1.1V}{256LSB} = 133 Hz/LSB$$

Això tenint en compte que la sortida del DA varia 1.1V en 256 LSB's. Com que nosaltres necessitem més sensibilitat, farà falta un adaptador d'escala entre el mòdul convertidor i el VCO.

6.5. Adaptador d'escala

L'adaptador d'escala sorgeix del problema de que la tensió generada pel D/A no és la correcta perquè l'VCO tingui la sensibilitat adequada i la freqüència central desitjada.

Com s'ha vist a l'estudi teòric, nosaltres necessitem més sensibilitat que 31KHz/V per poder fer canvis en la freqüència prou petits. Per aconseguir això, s'ha posat un adaptador d'escala a l'entrada del VCO.

La sensibilitat màxima que ens interessa aconseguir és aproximadament de 10 Hz/LSB.

Això vol dir que per 256 LSB's, el VCO s'ha de moure:

$$\frac{10\text{Hz}}{\text{LSB}} \cdot 256\text{LSB's} = 2560\text{Hz}$$

Però com que ha de generar freqüències 4 vegades més grans, s'haurà de moure:

$$\frac{4 \cdot 10\text{Hz}}{\text{LSB}} \cdot 256\text{LSB's} = 10240\text{Hz}$$

Llavors el guany de l'adaptador d'escala ha de ser de:

$$\frac{10240\text{KHz}}{1.1\text{V}} \bigg/ \frac{31\text{KHz}}{1\text{V}} = 0.3$$

Per poder fer el control de sincronisme, aquesta s'ha de poder augmentar i disminuir per sobre i per sota de 400 KHz. Per això s'ha triat que el VCO oscil·li a aquesta freqüència quant la entrada del D/A val 128LSB's, que és el centre de l'escala. D'aquesta manera el valor pot augmentar i disminuir sense arribar al fons d'escala.

La tensió que genera el D/A en rebre 128 LSB's és:

$$\frac{1.1V}{256LSB} \cdot 128LSB = 0.55V$$

Per tant, l'adaptador d'escala, en rebre 0.55V, haurà de donar-ne 2 per tal de que el VCO oscil·li a 400KHz. A més seria interessant poder ajustar manualment el valor d'aquesta tensió quan es volgués. És a dir que si en lloc de 0.55V es vol que sigui de 0.57 es pugui fer.

Aquest adaptador s'ha fet amb un amplificador operacional LM741 configurat com a amplificador inversor. El problema que presenta aquest muntatge és que és inversor. Això vol dir que si la tensió d'entrada augmenta, la de sortida disminueix. Per solucionar aquest problema s'ha de tenir en compte des de la FPGA, que quan es vulgui augmentar la freqüència s'haurà de disminuir el valor enviat al D/A i que quan es vulgui augmentar, aquest s'haurà de disminuir. Aquesta és la configuració com a amplificador inversor.

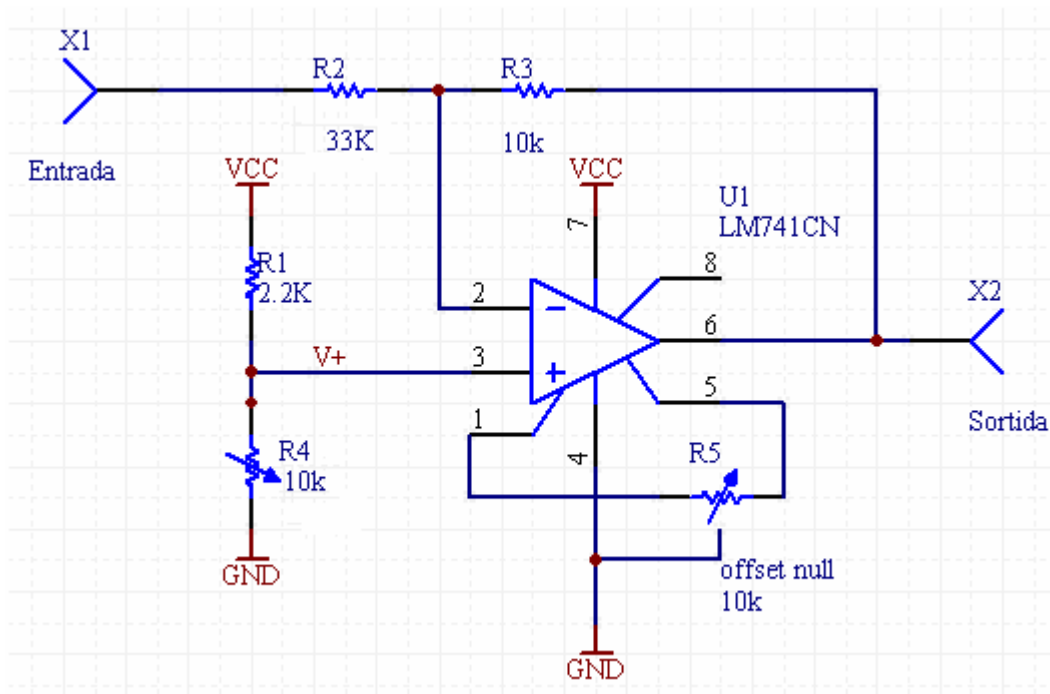


Fig. 6.15. Esquema del adaptador d'escala amb el Protel DXP

Si el circuit s'alimenta a 3.3V, la tensió de sortida en funció de la tensió d'entrada és:

$$V_{out} = -\frac{R3}{R2} \cdot V_{in} + \frac{R3}{R2} \cdot V_+ + V_+$$

On:

$$V_+ = 3.3 \cdot \frac{R4}{R1 + R4}$$

Com es veu en la fórmula de la tensió V_{out} , el guany de l'amplificador, deixant de banda el seu signe, és $R3/R2$. En apartats anteriors s'ha calculat que aquest ha de ser de 0.3. Per tant deduïm que:

$$R3 = 0.3 \cdot R2$$

Si la tensió de sortida ha de ser 2V quant a l'entrada n'hi hagi 0.55, podem trobar el valor de V_+ .

$$V_{out} = -\frac{R3}{R2} \cdot V_{in} + \frac{R3}{R2} \cdot V_+ + V_+$$

$$2 = -0.3 \cdot 0.55 + 0.3 \cdot V_+ + V_+$$

$$V_+ = 1.66$$

D'aquí podem determinar els valors de $R1$ i $R4$

$$V_+ = 3.3 \cdot \frac{R4}{R1 + R4}$$

$$1.66 = 3.3 \cdot \frac{R4}{R1 + R4}$$

$$R1 \cong R4$$

Els valors absoluts que s'han adoptat per a cada resistència són:

$R1 = 2.2K$
 $R2 = 33K$
 $R3 = 10K$
 $R4 = 0 - 10K$

Amb aquests components i el potenciòmetre degudament ajustat, la característica del adaptador d'escala és aquesta:

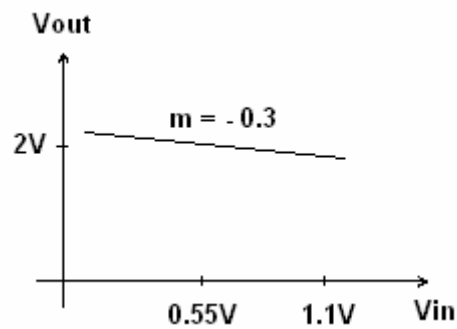


Fig. 6.16. Característica de l'adaptador d'escala

6.6. Generació de la senyal de rellotge, el nou VCO

Ara si ens mirem el conjunt adaptador-VCO, com un sol bloc, tenim el component que necessitàvem, un oscil·lador controlat per tensió amb la característica que nosaltres volíem:

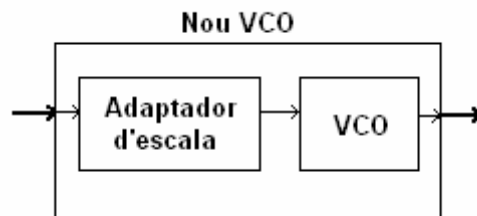


Fig. 6.17. Conjunt adaptador d'escala i el VCO es poden veure com un sol VCO

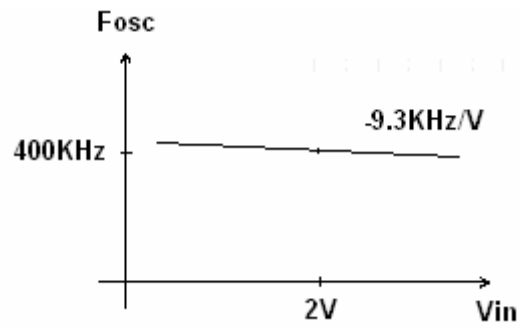


Fig.6.18. Característica de la parella adaptador-VCO

Tant l'adaptador d'escala com el VCO s'han muntat sobre la protoboard per tal de poder canviar algun component fàcilment en cas de que s'hagués de fer. Mes tard s'ha muntat sobre una placa de prototipatge. L'aspecte del circuit muntat sobre la protoboard i sobre la placa de prototipatge és aquest:

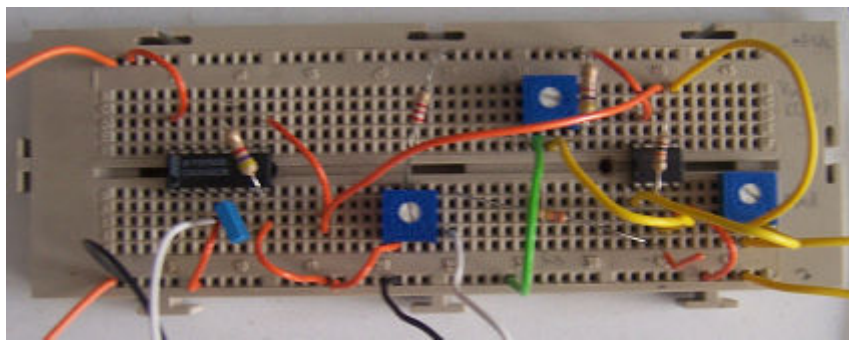


Fig. 6.19. Aspecte de la parella adaptador - VCO sobre la protoboard

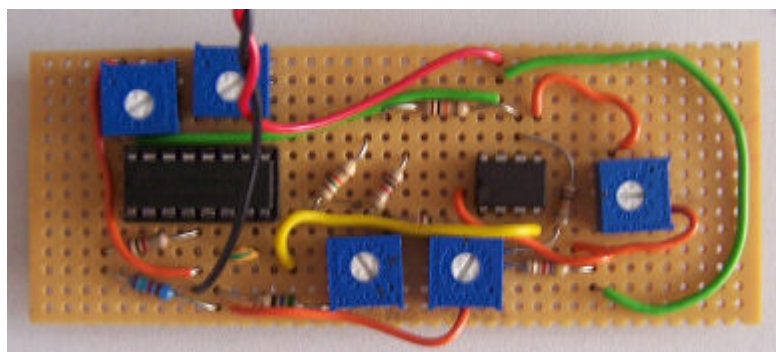


Fig. 6.20. Aspecte de la parella adaptador - VCO sobre la placa de prototipatge.

6.7. Connexió entre tots els mòduls

Per donar lloc a tot el sistema de sincronització, cal interconnectar tots els elements anteriors. Les connexions entre els mòduls anteriors s'han fet d'aquesta manera:

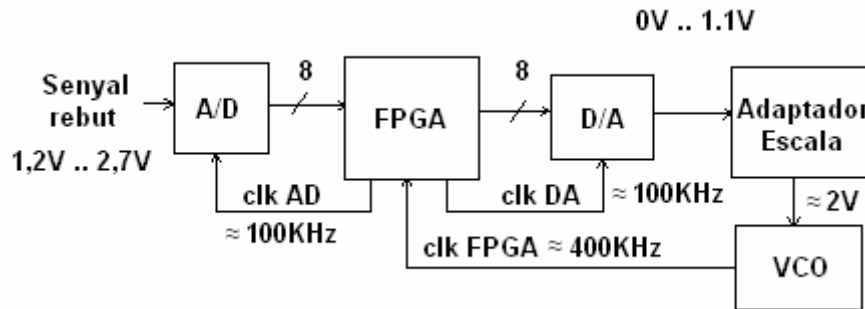


Fig. 6.21. Diagrama de blocs de tots els mòduls interconnectats

Per poder connectar els elements entre ells, cal que s'entenguin. És a dir s'han de respectar els temps de propagació, s'han de llegir les dades quant toca i altres coses que s'han de tenir en compte. En les següents figures es mostren els timings de funcionament.

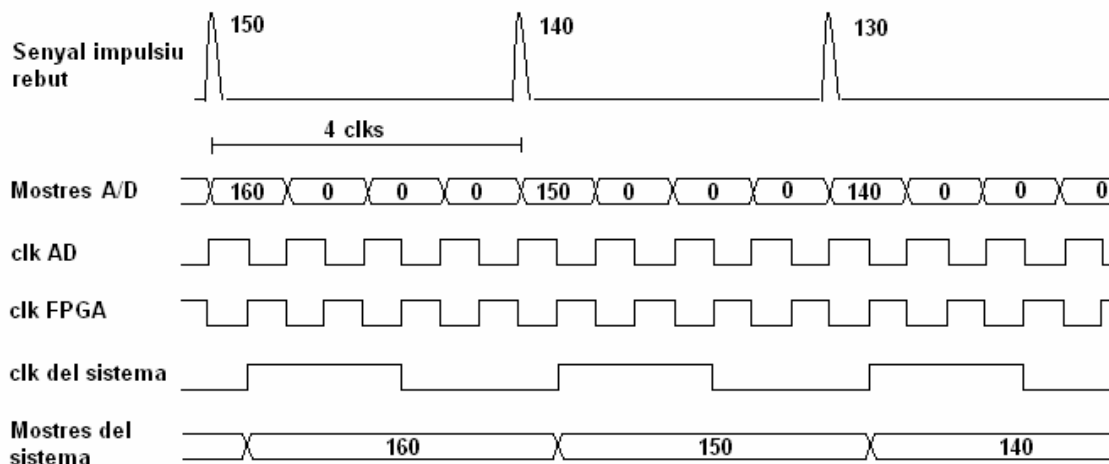


Fig. 6.22. Timing de la connexió entre el convertidor A/D i la FPGA

Per solucionar el problema de que el convertidor A/D tarda 4 cicles de clk en donar una conversió, es fa treballar aquest mòdul 4 vegades més ràpid que el clk amb el que funciona la resta del sistema. Així, les mostres només tarden un clk de sistema en aparèixer. La divisió del clk FPGA entre 4 per obtenir el clk del sistema es fa dins la mateixa FPGA.

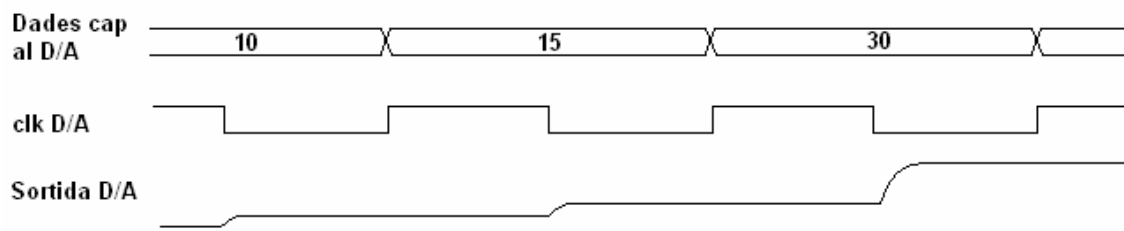


Fig. 6.23. Timing de la connexió entre la FPGA i el convertidor D/A

Les dades cap al D/A s'actualitzen en els flancs de baixada de la senyal del sistema. El convertidor D/A també actualitza la seva sortida en el flanc de baixada del rellotge. Per evitar problemes de glitch, el clk D/A és el clk del sistema invertit.

Per poder connectar els mòduls AD i DA amb la FPGA, s'han soldat dos connectors de 8 pins en aquesta última. És aquesta placa, la que fa de base de tot el muntatge. L'aspecte de totes les plaques juntes és aquest:

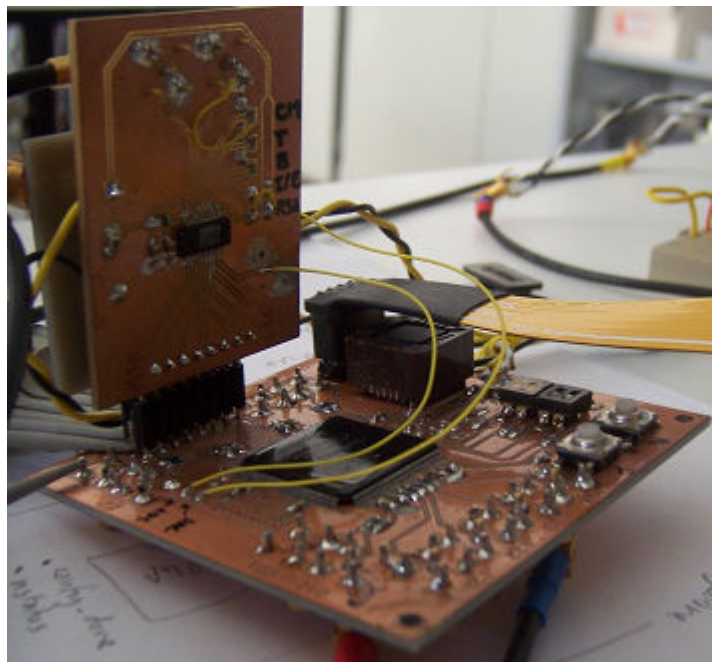


Fig. 6.24. Aspecte de tots els mòduls junts; el sistema a punt per implementar-hi els models de sincronisme

6.8. Test del hardware i proves

Un cop tenim el sistema real muntat, només cal programar la FPGA per tal de que implementi la funció que es vulgui depenent de les necessitats. La programació de la FPGA s'ha fet amb l'entorn de programació Quartus II que subministra la mateixa casa que fabrica els components: Altera. L'entorn de programació Quartus II permet programar a nivell de esquemes lògic o bé amb llenguatge VHDL.

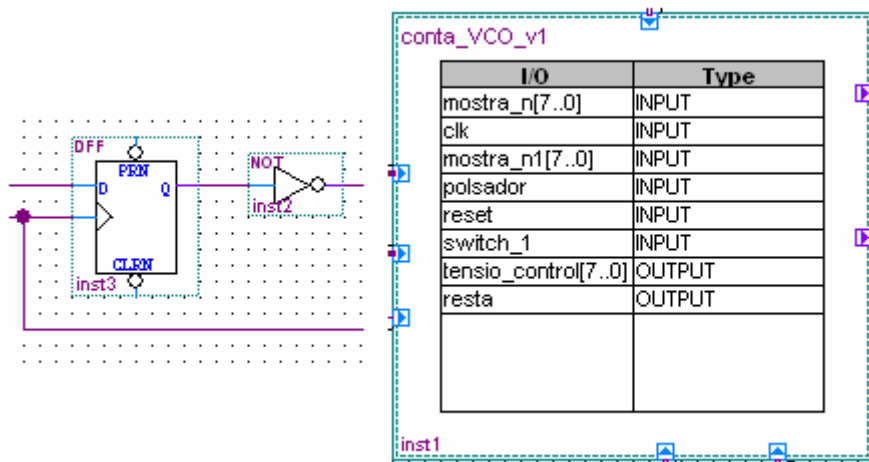


Fig. 6.25. Diferents formes de programació en el Quartus II, amb esquemes (a l'esquerra) o amb VHDL (a la dreta)

Al programa només cal indicar-li quina FPGA i quin memòria EPC2 es farà servir perquè pugui compilar el programa. Per programar el dispositiu s'ha fet a través del Programador Ethernet Blaster de la casa Altera, que com el seu nom diu, es comunica amb el PC a través d'Ethernet.

Inicialment per a provar que tots els components funcionen, s'ha programat la FPGA amb un programa molt simple. Aquest programa, que s'acabarà traduït en hardware, implementa el divisor de la senyal de rellotge entre 4 i inverteix la senyal del D/A. És a dir implementa la lògica necessària perquè el convertidor A/D, el D/A i la FPGA s'entenguin. A més aquest programa agafa les dades del convertidor A/D i les porta al convertidor D/A. D'aquesta manera, és fàcil veure si tot el hardware funciona. Si s'aplica una senyal de baixa freqüència a l'entrada del convertidor D/A, s'hauria de veure la mateixa a la sortida.

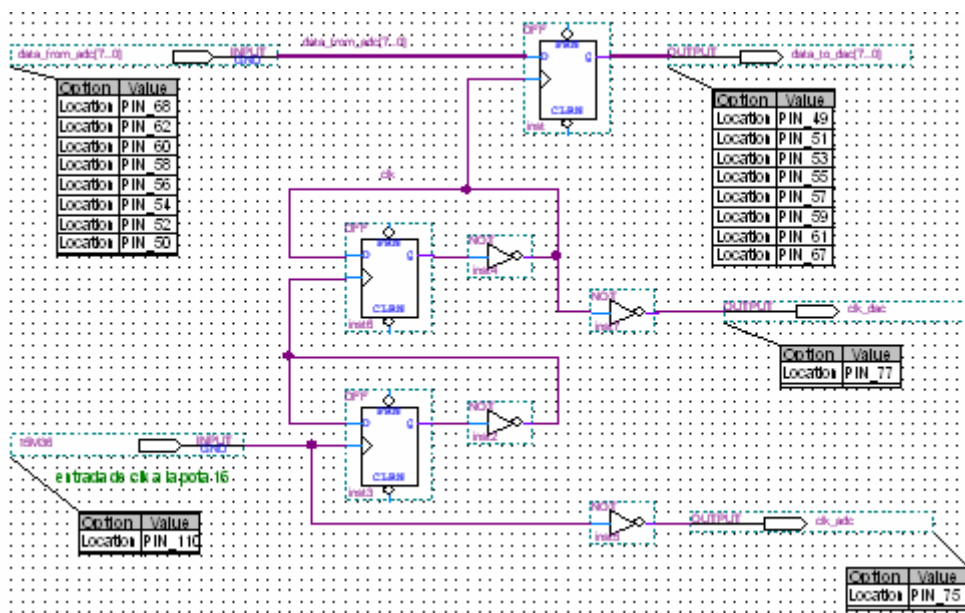


Fig.6.26. Detall del programa per a provar el hardware

El hardware resultant de programar la FPGA es pot simular en el mateix Quartus. La simulació per aquest programa de prova és aquesta.

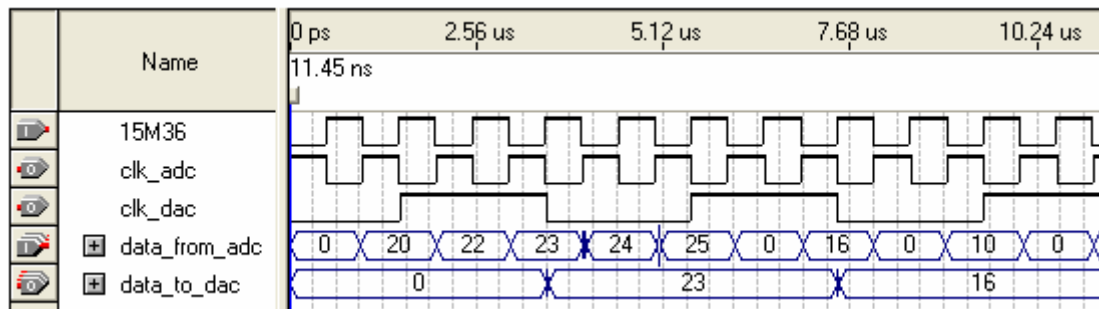


Fig.6.27. Simulació en entorn Quartus II del programa anterior

La prova d'aquest hardware el model real, ha funcionat correctament i s'ha obtingut a la sortida el mateix que a l'entrada, tal i com s'esperava.

6.9. Mesura del temps de reacció del VCO

Un paràmetre que cal conèixer és el temps de resposta del VCO. En les simulacions en el Simulink, els components eren ideals i no tenien retards en els seus temps d'establiment però ara en tractar-se d'un sistema real, és possible que existeixin retards. A més aquests temps poden afectar en la metodologia de sincronització. A continuació es realitza un programa per a tal de mesurar aquest temps.

La metodologia que s'ha fet servir per a mesurar-lo és la següent: Suposem que a l'entrada del convertidor hi apliquem una senyal UWB de 100KHz. Inicialment fixem la tensió de control del VCO perquè oscil·li a una freqüència més baixa. Llavors, les mostres es prendran de tant en tant sobre els polsos. Si en un d'aquests punts es fa un canvi en la freqüència, s'observarà com la tendència de les mostres és una altra.

Per entendre-ho millor vegem-ho amb els casos pràctics que s'han dut a terme per a mesurar els temps de reacció dels diferents VCO's.

6.9.1. Mesura del temps de reacció de la font de tensió configurada com a VCO

Inicialment es va proposar fer servir una font de laboratori com a VCO però s'ha descartat perquè el seu temps de reacció és massa gran. Aquesta és la mesura que s'ha fet:

Inicialment la freqüència generada pel VCO és més petita que la del senyal UWB, d'aquesta manera les mostres es prenen a vegades sobre els polsos i a vegades no.

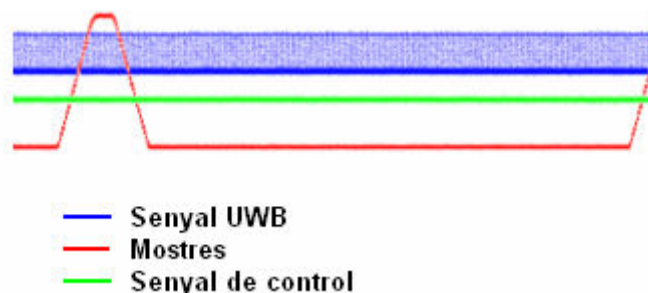


Fig.6.28. Captura de l'oscil·loscopi amb freqüències de senyal UWB i senyal de rellotge diferents

Si esperem que les mostres comencin a baixar i llavors fem un canvi en la tensió de control, podrem veure el temps de reacció del VCO.

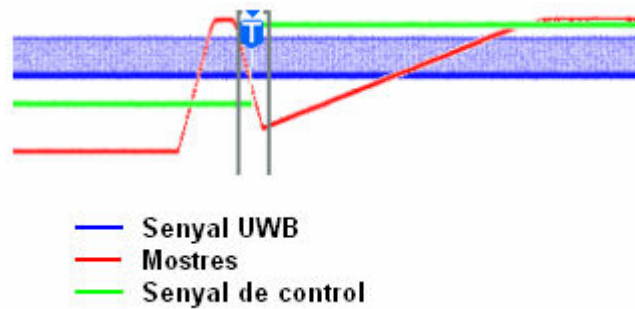


Fig.6.29. Captura de l'oscil·loscopi on s'aprecia el canvi de tendència de les mostres.

En la següent figura s'aprecia com el temps de reacció del VCO és de 7 cicles de rellotge.

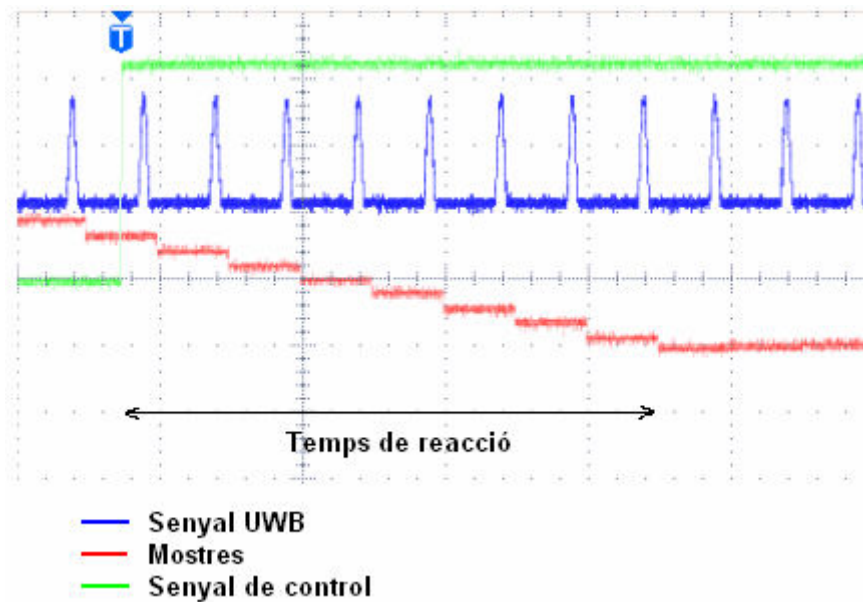


Fig.6.29. Captura de l'oscil·loscopi on s'aprecia el temps de reacció del VCO.

Aquest temps és massa gran per a un sistema de control que ha de ser molt més ràpid. Per això es va descartar l'ús de la font d'alimentació com a VCO.

El programa en VHDL per a mesurar aquest temps està en l'annex D.

6.9.2. Mesura del temps de reacció del VCO 74HCT4046

Per a mesurar el temps de reacció d'aquest VCO no ha estat tant fàcil com en el cas anterior, allà s'ha pogut aplicar una tensió de control de 0 a 3.3V i per a fer això s'ha fet servir un pin de la FPGA. En el cas actual, això no ho podem fer perquè el VCO només accepta tensions entre 0 i 1.1V i necessitem que la tensió de sortida del D/A sigui la tensió de control del VCO. D'aquesta manera no serà possible visualitzar el valor de les mostres i per tant no es podran veure amb l'oscil·loscopi els canvis en aquesta.

Per a superar aquest problema s'ha fet un programa amb VHDL que el que fa és contar els cicles de rellotge que passen des de que es fa un canvi fins que les mostres tenen una tendència positiva. Després, la FPGA dona a la sortida cap al D/A, el numero de cicles que han passat. Aquest numero es podrà llegir amb l'ajuda d'un analitzador lògic. A més aquest programa també controla el punt en el que es fa el canvi: es fa en el flanc de baixada de la tendència de les mostres com es veu en la figura. L'algorisme s'ha implementat amb una màquina d'estats, que es la que segueix a continuació.

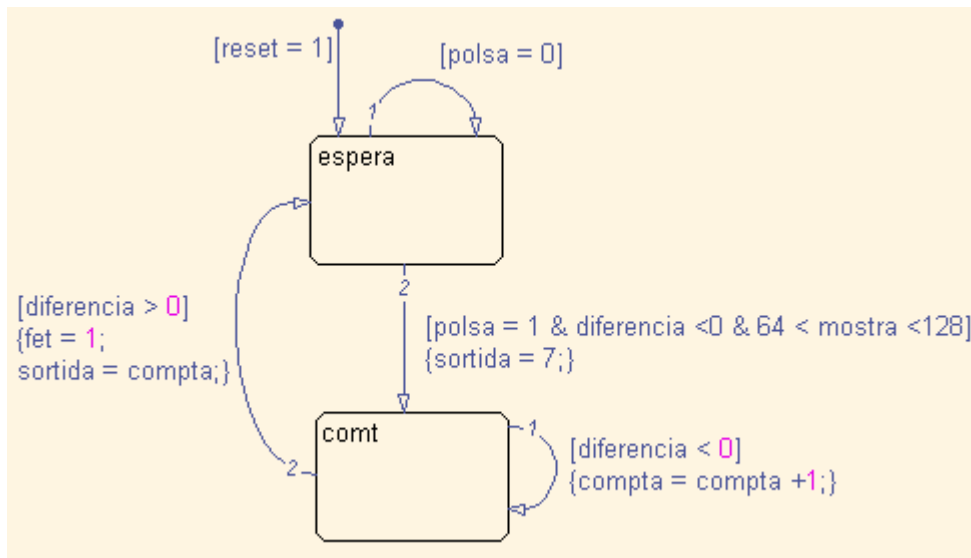


Fig.6.30. Màquina d'estats per a poder mesurar el temps del VCO.

El codi en VHDL per implementar la màquina en la FPGA es troba a l'annex D

La mesura que s'ha obtingut en aquest cas és la següent:

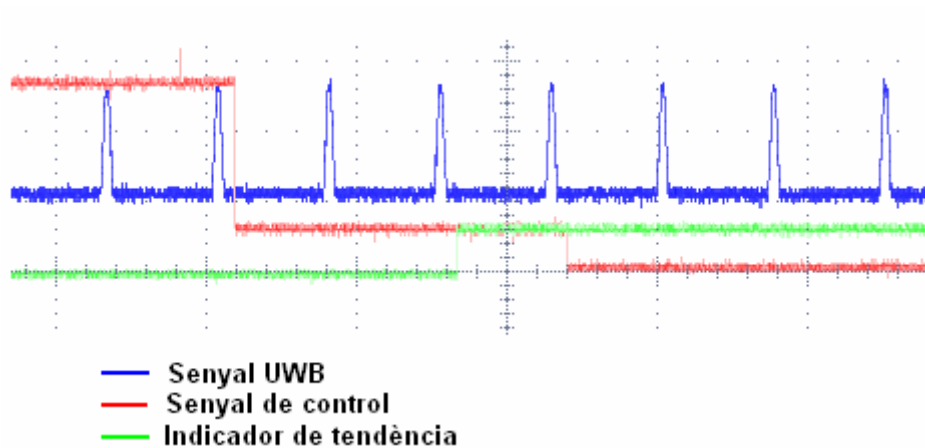


Fig.6.31. Captura de l'oscil·loscopi on s'aprecien els diferents valors de la senyal de control i del la qual es pot deduir el temps de reacció del VCO.

La senyal indicador de tendència està connectada en un pin de la FPGA i com a tal només pot prendre valors de entre 0 i 3.3V. Aquesta variable indica que la diferència entre una mostra i la anterior és negativa quant val 0 i positiva quant val 1.

Inicialment, la senyal de control val 0.13V i fa oscil·lar al VCO a una freqüència determinada, en el moment en que es compleixen les condicions de la transició de l'estat espera al comt, es fa un canvi en la senyal de control i la màquina d'estats compta els cicles de rellotge que tarda la variable diferència (representada per la senyal verda) a ser positiva. En aquest moment la senyal de control passa a indicar en binari el nombre de cicles que ha comptat mentre estava a l'estat comt, que en aquest cas són 2. Aquest és el temps de reacció del nostre conjunt adaptador-VCO.

Aquest temps (2 cicles de rellotge) és molt millor que 7 que tardava el primer. Tot i així aquests cicles de rellotge no provenen tots de el temps de reacció del VCO. Si s'analitza amb detall la *fig. 6.22* s'aprecia que un pols tarda 2 cicles de clock entre que apareix a l'entrada del convertidor A/D fins que és llegit pels flip-flops de la FPGA. Aquest temps que s'ha observat doncs és el corresponent a aquest retard i no al retard del VCO. Per tant s'ha arribat a la conclusió de que la resposta del VCO és gairebé immediata.

Tot i així s'haurà de tenir en compte que el temps entre que es fa un canvi i s'en aprecien els resultats és de 2 cicles de rellotge.

6.10. Implementació pràctica del model de sincronisme 'Single-Flank'

La implementació d'aquest model es va dur a terme sense voler, és a dir, un dia fent proves al laboratori me'n vaig adonar que un programa semblant al que vaig fer per a provar el hardware havia fet que el sistema enganxés i que la fase del senyal de rellotge seguís a la fase de la senyal d'entrada. El programa amb el que estava provant era el de l'annex D que està pensat per determinar el temps de resposta del VCO, però si el switch està a 0 l'únic que fa es posar al D/A el que hi ha al A/D. Llavors el funcionament del model coincideix amb del model de sincronisme Single-Flank. El programa adient per a implementar aquest model és el de l'annex C

Aquests són els resultats obtinguts.

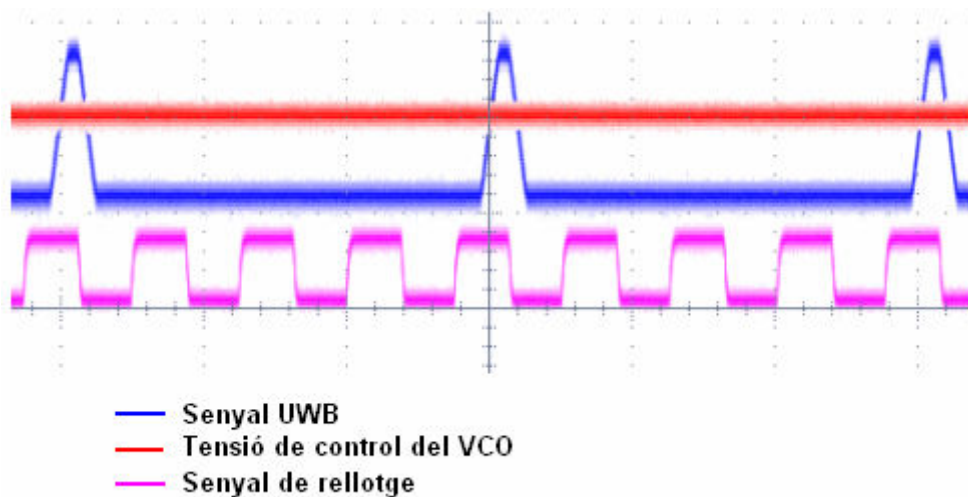


Fig.6.32. Captura de l'oscil·loscopi on s'aprecia que la senyal de rellotge i la senyal UWB estan sincronitzades.

En aquesta captura s'observa com la senyal de rellotge i la senyal UWB estan enganxades. Com que aquest model no és capaç d'enganxar amb error de fase 0 és lògic que la senyal de rellotge i la senyal UWB no estiguin exactament en la mateixa fase. L'error de fase, però, no és molt diferent de 0 ja que es pot veure com el flanc de

baixada de la senyal de sincronisme coincideix amb el flanc de baixada de la senyal UWB.

En el model teòric es donava que els flancs de sincronisme de la senyal de rellotge, coincidien amb el flanc de pujada de els polsos UWB. I ara perquè no es així? Això es justifica perquè el guany real de la funció de transferència, a diferència del teòric, és negatiu. La funció de guany en aquest cas és l'adaptador d'escala del VCO i aquest té guany negatiu.

S'han provat de fer canvis en la freqüència d'entrada i el model aconsegueix que la senyal de rellotge els segueixi. La resposta de la senyal de control és aquesta.

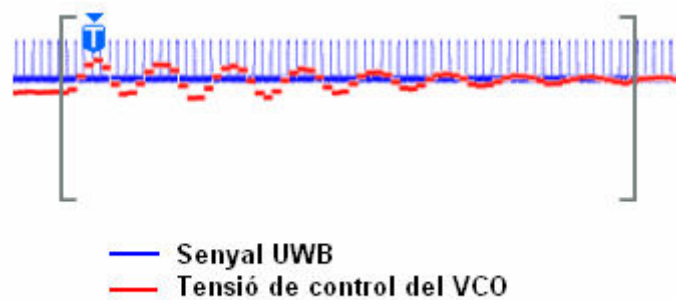


Fig.6.33. Captura de l'oscil·loscopi on s'aprecien les variacions de la senyal de control en produir-se un canvi en la freqüència del senyal UWB.

En la figura es pot veure com la tensió de control té una resposta amb sobreimpulsos. Això a la simulació teòrica no passava i és degut al retard que provoca el convertidor A/D.

Aquest model presenta l'avantatge que és molt simple, i que tarda relativament poc a enganxar, el problema és que el seu funcionament depèn del nivell de la senyal rebuda i aquesta, en un receptor inalàmbric, es pot veure afectada per a molts factors. Un altre inconvenient que presenta és que no pot enganxar amb error d fase 0. Tot i així no deixa de ser una possibilitat per a receptors amb els nivells de senyal constants i coneguts.

6.11. Implementació pràctica del model de sincronisme amb el detector de fase implícit

Per portar a la pràctica aquest model, cal un bon programa que permeti implementar la màquina d'estats corresponent. El programa que s'ha creat és el de l'*annex E*. Va ser el primer que es va crear i no es van tenir en compte moltes coses, per això quant s'ha portat a la pràctica no ha funcionat.

Per tal de depurar el programa s'han fet simulacions més reals, en les que s'ha inclòs soroll i el retard del convertidor A/D.

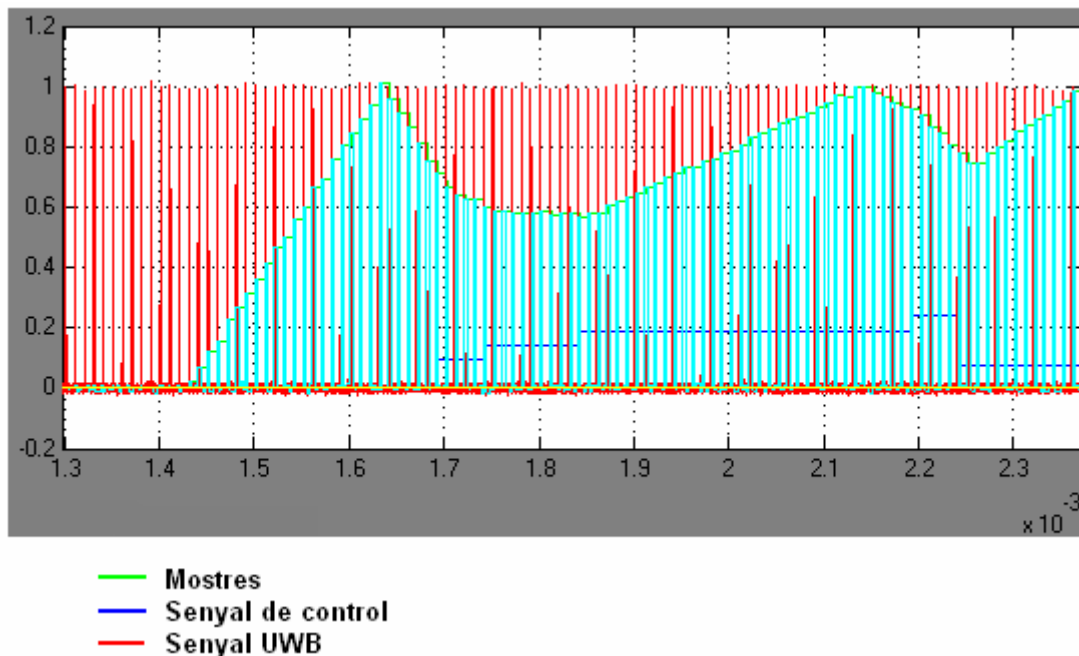


Fig.6.34. Resultats de la simulació amb soroll i retards reals

Com es pot veure a la figura, el model continua donant bons resultats en les simulacions tenint en compte els retards i el soroll.

Per tal de que el model funcioni a la realitat s'ha fet el programa de la FPGA el màxim ajustat al model del Simulink. Tot i així no s'ha aconseguit fer funcionar aquest model a la realitat. Això pot ser degut a que, en el model real, hi ha alguna cosa que no es té en compte en les simulacions i que fa que les coses no vagin com haurien d'anar. O també podria ser perquè el model real s'ajusta exactament al model del Simulink.

Per aconseguir que el model funcionés, aquest s'hauria de depurar amb detall, mirar en cada moment quin es el valor de les mostres, mirar si els canvis en la senyal de control són els que toquen en funció d'aquestes mostres, si la senyal de control actua correctament sobre l'VCO i altres aspectes que són molt difícils de mesurar en temps real. Sobretot pel fet de no poder fer pauses en el sistema i de no poder mesurar el valor d'algunes variables.

No sé si algú ha aconseguit que un model amb aquesta metodologia de sincronisme funcioni, però és un model interessant que sembla que pugui funcionar, ja que presenta avantatges com una gran rapidesa d'adquisició i la capacitat d'enganxar amb error de fase molt pròxim a 0.

6.12. Implementació pràctica del model 'Early-Late'

Aquest model no s'ha arribat a implementar físicament degut a que s'ha donat preferència a obtenir unes simulacions molt aproximades a la realitat per no trobar-nos com en el cas anterior. Tant és així que el temps disponible per a portar a terme el projecte s'ha invertit en aquestes simulacions.

Per a la simulació d'aquest model s'han fet moltes modificacions respecte el primer model teòric, només cal mirar l'esquema del nou model per donar-se'n compte.

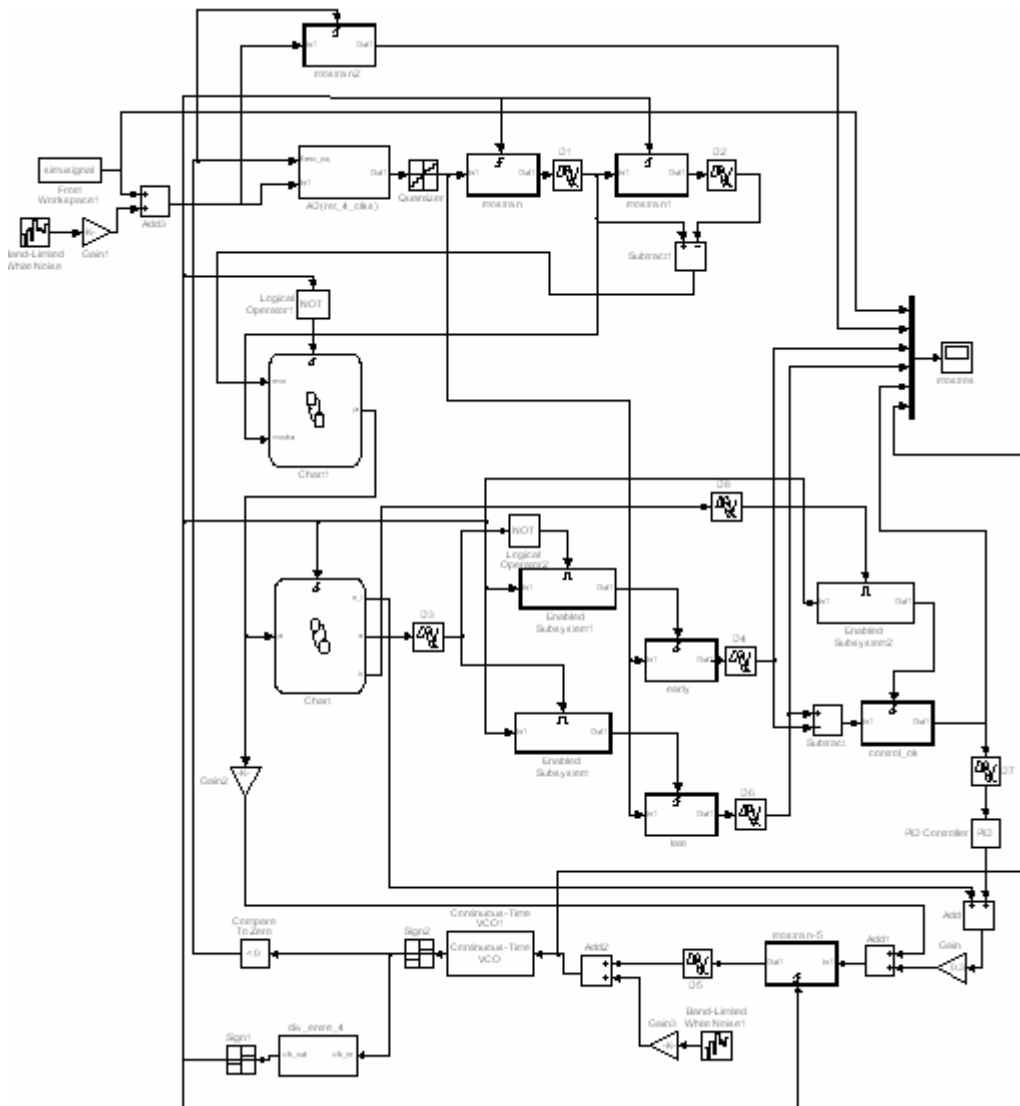


Fig.6.35. Model implementat amb Simulink que té en compte els sorolls i retards reals

A diferència del primer model teòric aquí s'ha implementat de més el divisor de freqüència per al convertidor A/D, el convertidor A/D, retards als flip-flops, el convertidor D/A i soroll a la senyal de control i a la senyal d'entrada.

Aquest model té dues màquines d'estats, a diferència del primer que només n'hi havia una. La màquina principal és la que regula els cicles early i late i la altra màquina és auxiliar. Aquesta serveix per fer entrar el detector early-late en el punt on la seva característica té pendent positiu.

A continuació es presenten algunes de les simulacions que s'han fet:

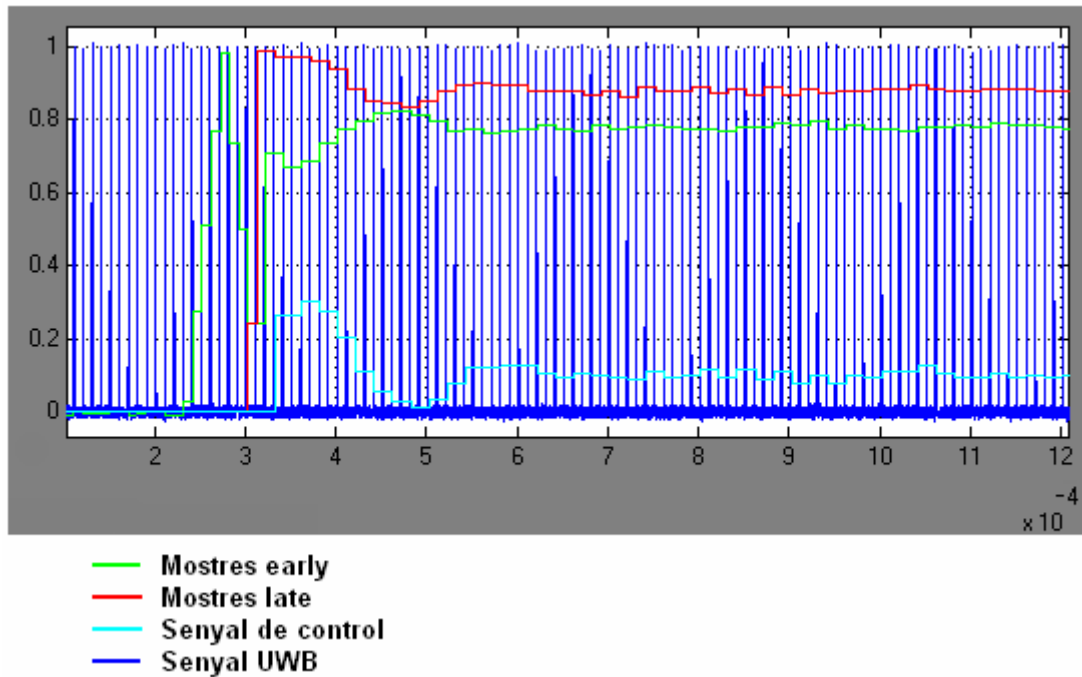


Fig.6.36. Resultat de la simulació amb el model 'realista amb guany = 0.5

En aquesta simulació s'aprecia com inicialment la freqüència de la senyal de rellotge es molt diferent de la senyal rebuda. En el moment en que el valor d'una mostra actual és inferior a una mostra anterior entra en joc la màquina d'estats principal i comença la metodologia early-late.

La dificultat que sembla presentar aquest model a l'hora d'implementar-lo en el sistema real és que la màquina auxiliar aconsegueixi posar el detector de fase early late en el punt de pendent positiu. Si això s'aconsegueix, sembla que el model pot funcionar correctament, sempre i quant els guanys de cada element siguin els adients.

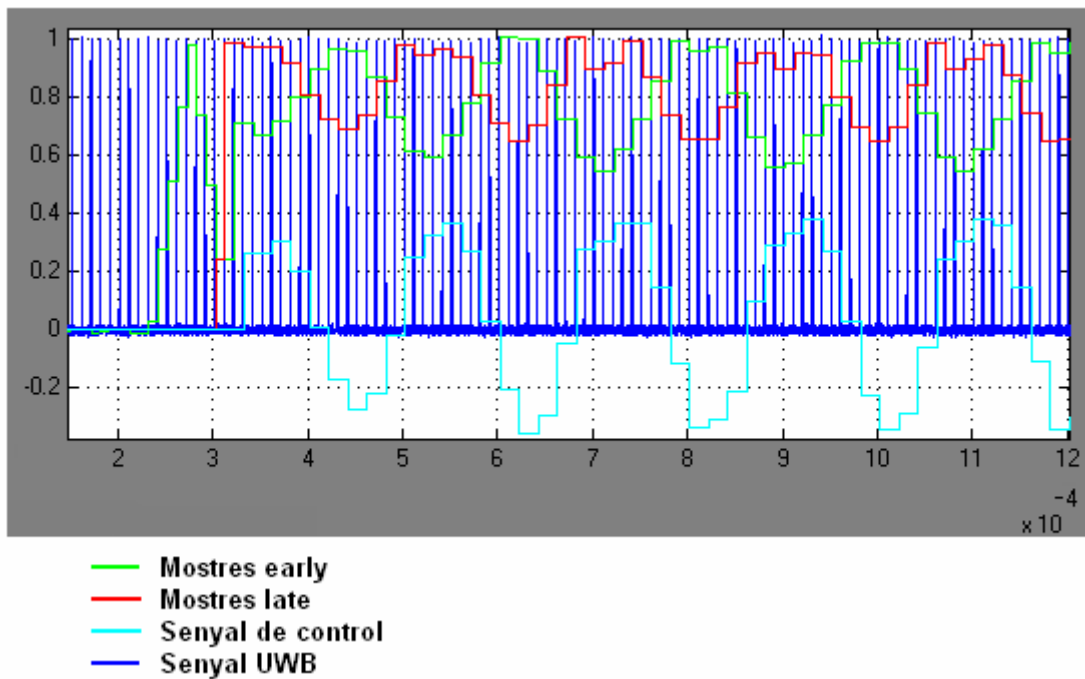


Fig.6.37. Resultat de la simulació amb el model 'realista' amb guany = 1

En aquesta simulació s'aprecia com la resposta de la senyal de control és inestable. Això vol dir que s'ha de triar un guany tal que no passi això. En aquest cas per a guanys més petits de 0.6, la resposta és estable.

Aquest és un bon model de sincronisme ja que és molt poc sensible al soroll i un cop està enganxat és molt difícil de que es desenganxi. A més no depèn del nivell de senyal, així que si hi ha pèrdues en els nivells o altres deficiències, el model seguirà funcionant igualment. Per tant és un bon model per a ambients amb soroll. Per altra banda presenta l'inconvenient que és complex d'implementar.

6.13. Comparació de les tècniques de sincronisme anteriors

Les tres tècniques de sincronisme anteriors són més o menys efectives, per aconseguir una sincronització UWB. Però n'hi ha que presenten més avantatges que d'altres. Per exemple el model amb el detector de fase 'Track & Hold' presenta l'avantatge que és molt fàcil d'implementar, tant que es podria fer amb un simple condensador i un interruptor. Aquesta tècnica presenta l'inconvenient de que és absolutament dependent del nivell de la senyal rebuda i de la forma dels polsos transmesos. A més no és capaç

d'enganxar amb error de fase 0. Això fa que aquesta tècnica només pogués funcionar en condicions ideals i com que en la majoria dels casos aquestes no es compleixen, l'ús d'aquesta tècnica es veu molt limitat. Per fer front a aquest problema hi ha la tècnica de sincronisme early-late que és més difícil d'implementar però no és sensible als nivells de la senyal. Això tampoc és del tot cert perquè al principi si que s'han de mirar els nivells per saber si estem sobre un pols o no i per tant també depèn en part dels nivells de senyals rebuts. Un receptor amb la tècnica de sincronisme early-late podria treballar amb llocs amb més deficiències en el canal de transmissió que no pas el primer model vist. A més aquest és un receptor molt robust al soroll, cosa que el fa útil per la majoria dels casos reals. També cal destacar que aquest model és capaç d'enganxar amb error de fase 0, cosa necessària si la intensitat del senyal és baixa. Un gran inconvenient que presenta és que un receptor amb només aquesta tècnica no és capaç d'adquirir mai. El temps d'adquisició és el temps que tarda el sistema de sincronisme des de que està completament desenganxat en enganxar. Per tal de solucionar el problema a la dependència del nivell de senyal i a l'adquisició, s'ha dissenyat la tècnica amb el detector de fase implícit, que ha estat una llàstima que no acabés funcionant. Però això no vol dir que no pugui funcionar. Aquesta tècnica no es sensible al nivell de les senyals rebudes, és a dir que un receptor amb nivells baixos de senyal, que fes servir aquesta tècnica, seguiria funcionant. A més presenta l'avantatge de que adquireix relativament ràpid i autònomament. A més permet enganxar amb error de fase gairebé nul.

Per tant no hi ha una millor i indiscutible tècnica per aplicar en un receptor de senyals UWB de tipus impulsiu, sinó que dependrà de cada situació. Tot i que les tècniques early – late o la del detector de fase implícit, si funcionés, serien bones candidates per fer servir en la majoria dels receptors d'aquest tipus.

6.14. Aplicació de les tècniques a freqüències altes

Fins ara s'han fet les proves i els estudis amb senyals de baixa freqüència on es podien fer servir tot tipus de components i les senyals tenien angles i pendents perfectes. En les senyals UWB reals això no passa. Com que els polsos són tant petits (de l'ordre dels ps), aquests prenen formes rares, com aquestes:

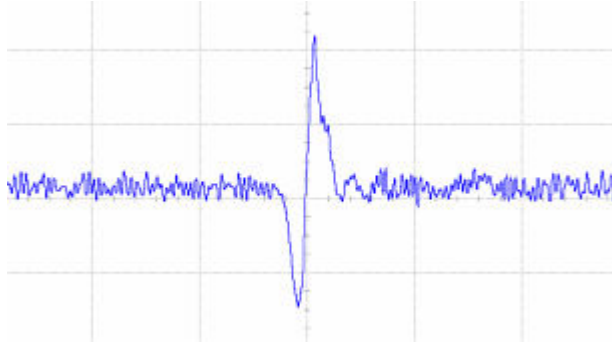


Fig.6.38. Pols UWB real

En aquestes temps tant petits no es poden fer servir convertidors A/D i la feina que fins ara feia el convertidor, ara l'hauria de fer un altre component.

Aquest component, hauria de ser un *receptor superregeneratiu*. Això és el nom que se li dona a un circuit que té el següent funcionament: En cada cicle d'un senyal periòdic, el component es mira l'entrada i en funció d'aquesta dona una sortida o una altra. En general el valor de la sortida és proporcional al de l'entrada. A més a part d'això s'haurien de tenir en compte més coses com les formes dels polsos generats o la seva freqüència de repetició.

Per comprovar que les tècniques de sincronització vistes en apartats anteriors també funcionarien en altes freqüències s'han fet simulacions amb formes de pols no tant perfectes, concretament amb polsos gaussians.

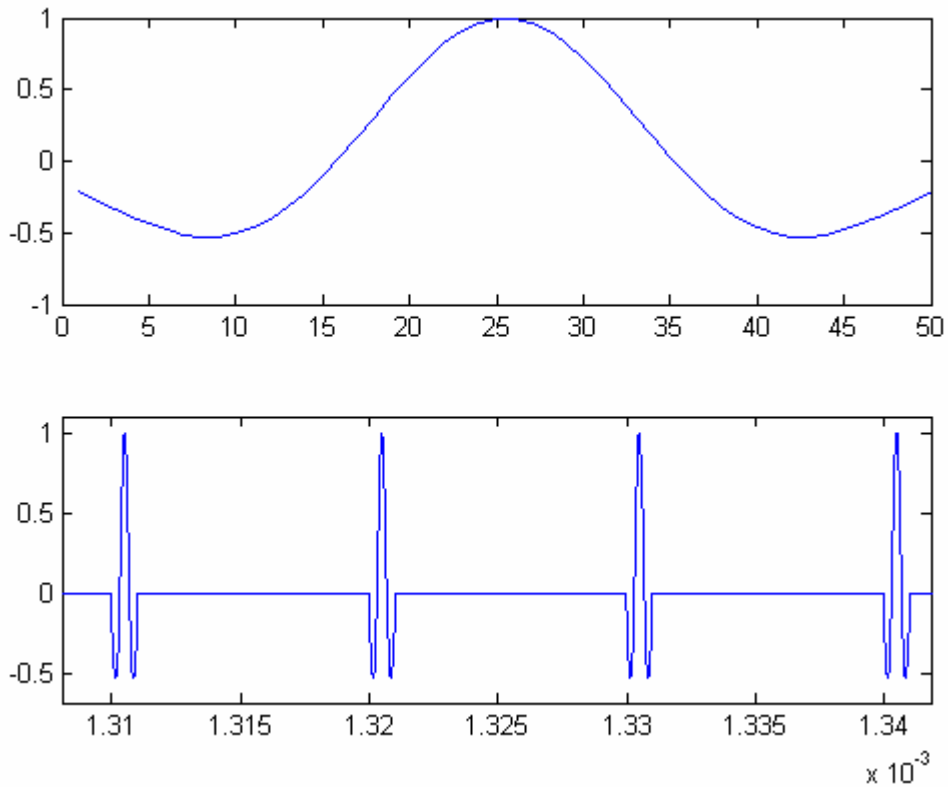


Fig.6.38. Pols i tren de polsos gaussians generats amb el Matlab

Aquests són els polsos amb els que s'ha simulat el funcionament a alta freqüència. El programa en *Matlab* per a generar aquests polsos es troba en l'*annex B*.

L'única tècnica de sincronització que s'ha simulat amb polsos gaussians en lloc de triangulars ha estat la 'early - late'. Això ha estat així perquè, deixant de banda el model amb el detector de fase implícit, s'ha estimat que aquesta tècnica és la més adient per a funcionar en altes freqüències.

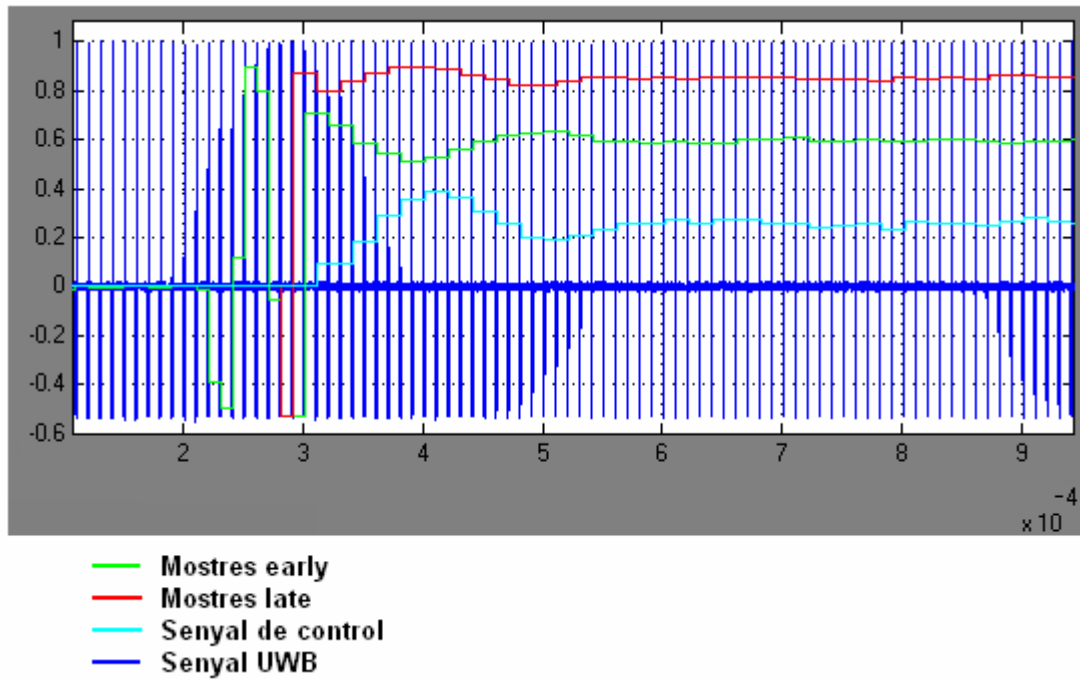


Fig.6.39. Resultat de la simulació amb guany de la funció de llaç = 0.2

Com es pot veure a la figura, la tècnica early -late també funcionaria per a altes freqüències tot i que amb un altre guany de la funció de llaç, en aquest cas 0.2.

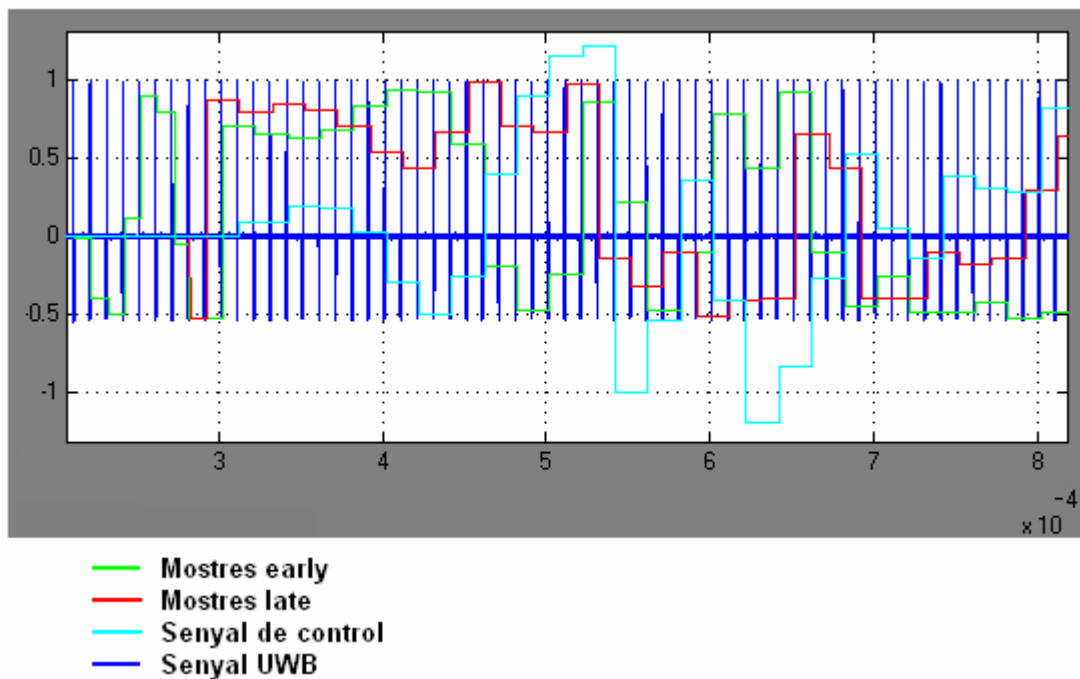


Fig.6.39. Resultat de la simulació amb guany de la funció de llaç = 1

Com es pot veure a la figura, amb un guany massa elevat, el sistema es torna inestable.
Per tant caldrà anar molt en compte en el disseny de no cometre errors com podrien ser aquest.

7. Conclusions i línies futures

Tenint en compte que els objectius inicials del projecte eren estudiar, dissenyar i implementar físicament un sistema de sincronisme, puc dir que s'han assolit gairebé tots. El que sí que vull dir és que personalment esperava que en finalitzar el projecte s'aconseguissin més objectius, sobretot en la part pràctica. Si he de ser sincer m'ha sabut greu que el model de sincronisme que tant temps vaig estar dissenyant no hagi funcionat en el sistema físic real. Per altra banda estic orgullós d'haver aconseguit que en les simulacions, tots els models funcionessin i d'haver aconseguit fer funcionar el model més simple a la realitat.

Tot i que m'he passat moltes hores pensant en coses que finalment no han acabat funcionant, no considero que hagin estat hores llençades perquè que no hagin funcionat no vol dir que en algun moment no ho puguin fer si s'acaben de perfeccionar algunes coses. Per això de cara a la presentació del projecte, hi dedicaré més hores perquè m'agradaria que al menys un dels models més complexos acabés funcionant sobre el sistema real.

Per a poder dur a terme aquest projecte he fet ús de coneixements adquirits durant la carrera, sobretot de les matèries més relacionades amb circuits de comunicacions. Per altra banda, considero que totes les hores que he passat fent aquest projecte que són moltes m'han servit per adquirir nous coneixements que no he adquirit durant la carrera i que crec que són molt interessants i que estan íntimament relacionats amb l'àmbit de les telecomunicacions.

Seria interessant continuar aquest projecte i donar lloc a un prototipus de receptor UWB. Precisament un dels objectius d'aquest projecte és obrir camí cap a un receptor d'aquest tipus, i de que serveix obrir un camí cap a un lloc si després ningú no hi passa? M'agradaria animar a aquells que estiguin interessats en desenvolupar un receptor de última tecnologia, que abans de posar-s'hi, perdin un temps fullejant aquest treball, si mes no perquè no cometin els mateixos errors que he comés jo i perquè puguin agafar exemple d'aquelles coses que han funcionat correctament.

Concretament seria interessant desenvolupar un receptor de dades pensat per anar molt pròxim a un emissor i consumir molt poc. Es podria aplicar en aplicacions mèdiques: implantar un receptor de baix consum en alguna part del cos per fer una determinada funció. També es podria desenvolupar un receptor per a dades en àmbit de les PAN, com per exemple per fer funcionar altaveus, monitors i altres perifèrics d'un ordinador personal sense fils. Aquest és el futur d'aquestes tecnologies.

8. Bibliografia

Format paper

-Ian Oppermann, Matti Hämäläinen, Jari Iinatti (Editors). UWB: Theory and Applications. Ed. Wiley

-Ricard Sanahuja. Disseny de sistemes digitals (I)
Transparències i apunts.

-Pere Palà. Electrònica de comunicacions
Transparències i apunts

-Rosa Giralat. Projectes
Transparències i apunts

Format digital

Central de Matlab

<http://www.mathworks.es/company/events/webinars/>

<http://www.mathworks.com/products/stateflow/>

Informació sobre la tecnologia UWB i mètodes de sincronisme

<http://www.informit.com/articles/article.aspx?p=433381&seqNum=5>

http://www.gmmaggio.com/publications/conferences/2004/euroem_2004.pdf

<http://www.idg.es/comunicaciones/articulo.asp?id=167630>

Informació de la FPGA i Quartus II

<http://www.altera.com/products/software/flows/fpga/flo-fpga.html>

<http://www.fpgacentral.com/docs/fpga-tutorial/how-start>

Annex A: Programa per a Matlab per a generar un tren de polsos triangulars

```
durpols=1e-6; %durada del pols en segons
freq_fon=100e3; %frequencia fonamental de repeticio de polsos en Hz
tempsfinal=0.003; %temps de simulacio en segons

%construccio d'un pols triangular
inicial=[5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100];

%aquesta secuencia reflexa el vector inicial sobre la ultima mostra
pulse=0;
pulse=inicial./100;
j=length(pulse)-1;
i=length(pulse)+1;
while (j>0)
pulse(i)=pulse(j);
i=i+1;
j=j-1;
end
subplot(2,1,1), plot(pulse), figure(gcf);

%definicio de t, de 0 a tempsfinal amb separacions de mostra de
%durpols/llargada del pols
t=[0:(durpols/length(pulse)):tempsfinal];

%creacio del tren de polsos per convolucio d'un tren de deltes amb el
%pols que he creat abans-- deltes separades-> (1/freq)/(separacio de
%mostra).
trendelta=zeros(1,length(t));
trendelta(1:length(pulse)/(durpols*freq_fon):length(t))=1;
trenpolsos=conv(trendelta,pulse);
subplot(2,1,2), plot(t,trenpolsos(1:length(t)))

%creacio de la senyal de simulacio simusignal
data=([zeros(1,35) trenpolsos(1:length(t))]);
simusignal=[t' (data(1:length(t)))'];
```

Annex B: Programa per a Matlab per a generar un tren de polsos gaussians

```
durpols=1e-6;    %durada del pols en segons
freq_fon=100e3;    %frequencia fonamental de repeticio de polsos en
Hz
tempsfinal=0.003;    %temps de simulacio en segons

%construccio d'un pols gaussià
tc = gauspuls('cutoff',100e3,0.9,[],-10);
t1 = linspace(-tc,tc,50)
pulse = gauspuls(t1,100e3,0.9);

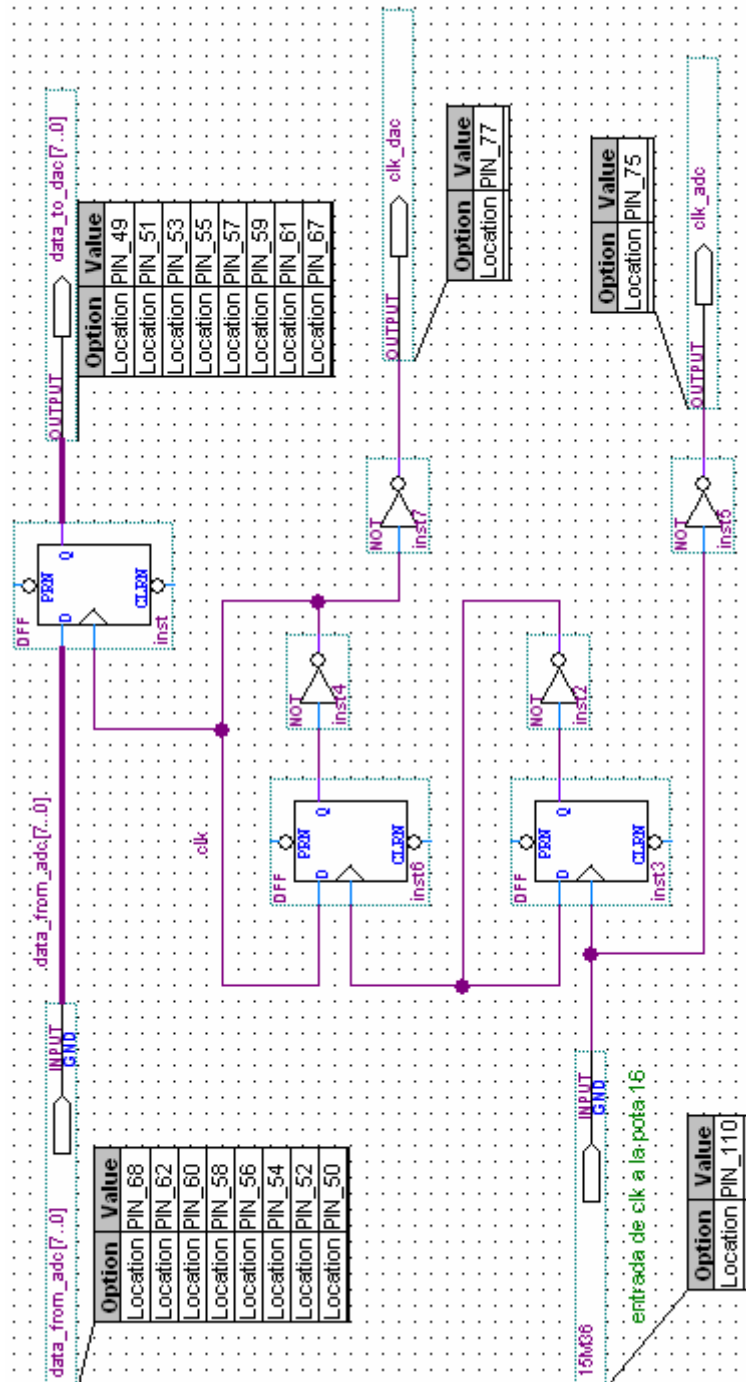
subplot(2,1,1), plot(pulse), figure(gcf);

%definicio de t, de 0 a tempsfinal amb separacions de mostra de
%durpols/llargada del pols
t=[0:(durpols/length(pulse)):tempsfinal];

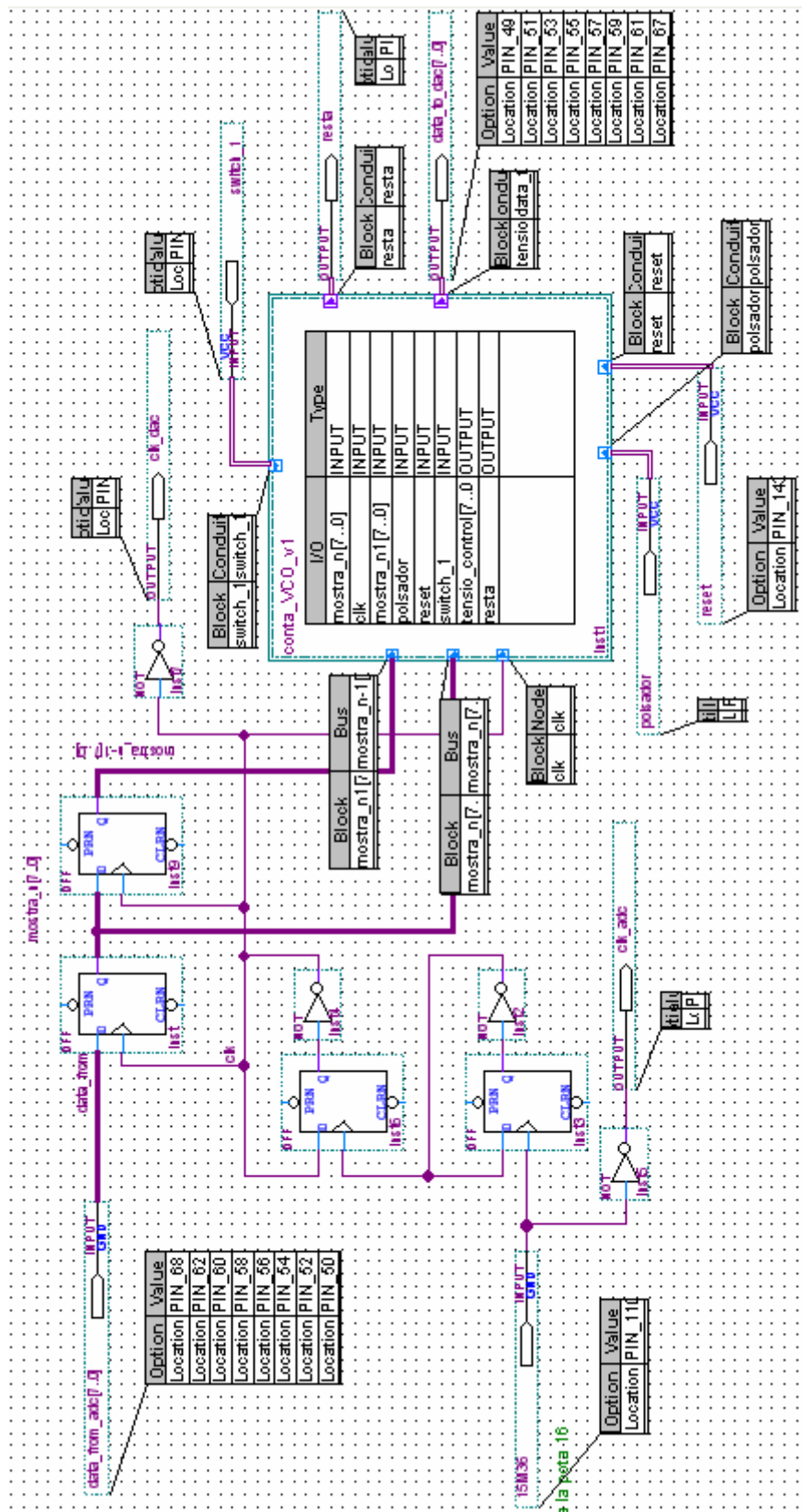
%creacio del tren de polsos per convolucio d'un tren de deltes amb el
pols
%que he creat abans-- deltes separades-> (1/freq)/(separacio de
mostra).
trendelta=zeros(1,length(t));
trendelta(1:length(pulse)/(durpols*freq_fon):length(t))=1;
trenpolsos=conv(trendelta,pulse);
subplot(2,1,2), plot(t,trenpolsos(1:length(t)))

%creacio de la senyal de simulacio simusignal
data=( [zeros(1,35) trenpolsos(1:length(t))] );
simusignal=[t' (data(1:length(t)))'];
```

Annex C: Programa per la FPGA per a provar el funcionament del hardware i implementar el detector de fase 'Track & Hold'



Annex D: Programa per la FPGA per a mesurar el temps de reacció del VCO



Programa en VHDL

```
-- Created on Tue Apr 07 18:26:58 2009

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_signed.ALL;

-- Entity Declaration

ENTITY conta_VCO_v1 IS
  -- {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
  PORT
  (
    mostra_n : IN STD_LOGIC_VECTOR(7 downto 0);
    clk : IN STD_LOGIC;
    mostra_n1 : IN STD_LOGIC_VECTOR(7 downto 0);
    polsador : IN STD_LOGIC;
    reset : IN STD_LOGIC;
    switch_1 : IN STD_LOGIC;
    resta : OUT STD_LOGIC;
    tensio_control : OUT STD_LOGIC_VECTOR(7 downto 0)
  );
  -- {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!
END conta_VCO_v1;

-- Architecture Body

ARCHITECTURE comportament OF conta_VCO_v1 IS
  signal int_n, int_n1, diferencia : integer range 0 to 255;
  signal controla : integer range 0 to 255 := 0;
  signal comptador : integer range 0 to 255 := 0;
  type estat is (espera, comt);
  signal estat_actual : estat := espera;
  signal polsa, fet : std_logic;

  -- ull!! no es poden fer servir comparacions (a < 0) pero si(a > 0)
  -- nombres positius --> < 127
  -- nombres negatius --> >= 128

BEGIN
  ----- no cal process perque no hi ha paraula clau -----

  int_n <= conv_integer(mostra_n);
  int_n1 <= conv_integer(mostra_n1);
  diferencia <= int_n - int_n1;

  ----- proces 1 -----

  process(reset, polsador, fet)
  begin
    if ((reset = '0') or (fet = '1')) then
      polsa <= '0';
    elsif (polsador = '0') then
      polsa <= '1';
    else null;
    end if;
  end process;

  ----- maquina d'estats -----

  process(reset, clk, polsa, diferencia, int_n, estat_actual, controla, comptador)
  begin
    if (reset = '0') then
      estat_actual <= espera;
      fet <= '0';
      controla <= 30;
    end if;
  end process;
end comportament;
```

```

comptador <= 0;

elsif (clk'event and clk = '0') then
--wait until (clk'event and clk = '0');

case estat_actual is
  when espera =>
    if ((polsa = '1') and (diferencia >= 128) and (int_n < 128) and (int_n >
//64)) then
      estat_actual <= comt;
      controla <= 5;
    else
      estat_actual <= espera;
    end if;

  when comt =>
    if (diferencia >= 128) then
      estat_actual <= comt;
      comptador <= comptador + 1 mod 256;
    else
      estat_actual <= espera;
      fet <= '1';
    end if;

  end case;

end if;

end process;

----- proces 3 -----

process(fet,switch_1)
begin

if (switch_1 = '1')then          --amb el switch es pot triar entre veure a la
//sortida les mostres o que fagi el control.
  if (fet = '0') then
    tensio_control <= conv_std_logic_vector(controla,8);
  else
    tensio_control <= conv_std_logic_vector(comptador,8);
  end if;

elsif (switch_1 = '0') then
  tensio_control <= mostra_n;
  --tensio_control <= conv_std_logic_vector(134,8);

else null;
end if;

end process;

----- proces 4 -----

process(diferencia)
begin

if (diferencia < 127) then
  resta <= '1';
else
  resta <= '0';
end if;

end process;

END comportament;

```

Annex E: Programa en VHDL per al sistema de sincronisme amb el detector de fase implícit

Veure l'esquema implementat amb el Quartus II. El fitxer de projecte està disponible en el CD dins el directori /CONTROLADOR.

Programa en VHDL

```
-- Created on Tue Apr 07 18:26:58 2009

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_signed.ALL;

-- Entity Declaration

ENTITY controlador IS
  -- {{ALTERA_IO_BEGIN}} DO NOT REMOVE THIS LINE!
  PORT
  (
    mostra_n : IN STD_LOGIC_VECTOR(7 downto 0);
    clk : IN STD_LOGIC;
    mostra_n1 : IN STD_LOGIC_VECTOR(7 downto 0);
    mostra_n2 : IN STD_LOGIC_VECTOR(7 downto 0);
    resta : OUT STD_LOGIC_VECTOR(7 downto 0);
    deriva : OUT STD_LOGIC_VECTOR(7 downto 0);
    tensio_control : OUT STD_LOGIC_VECTOR(7 downto 0)
  );
  -- {{ALTERA_IO_END}} DO NOT REMOVE THIS LINE!
END controlador;

-- Architecture Body

ARCHITECTURE comportament OF controlador IS
  type estat is (anembe, hemapujat, hemabaixat);
  signal int_n, int_n1, int_n2, diferencia, derivada : integer range 0 to 255;
  signal controla : integer range 0 to 255 := 128;
  signal estat_actual : estat := anembe;

BEGIN

  int_n <= conv_integer(mostra_n);
  int_n1 <= conv_integer(mostra_n1);
  int_n2 <= conv_integer(mostra_n2);
  diferencia <= int_n - int_n1;
  derivada <= diferencia - int_n1 + int_n2;
  resta <= conv_std_logic_vector(diferencia,8);
  deriva <= conv_std_logic_vector(derivada,8);

  process
  begin

  wait until clk'event and clk = '0';

  case estat_actual is
    when anembe =>
      if (diferencia >= 128) then
        estat_actual <= hemapujat;
        controla <= controla + 1 mod 256;
      else estat_actual <= anembe;
```

```

        end if;

    when hemapujat =>
        if (diferencia < 127) then
            estat_actual <= anembe;
        elsif ((diferencia >= 128) and (derivada >= 128)) then
            estat_actual <= hemabaixat;
            controla <= controla - 2 mod 256;
        elsif ((diferencia >= 128) and (derivada < 127)) then
            controla <= controla + 1 mod 256;
            estat_actual <= hemapujat;
        end if;

    when hemabaixat =>
        if (diferencia < 127) then
            estat_actual <= anembe;
        elsif ((diferencia >= 128) and (derivada >= 128)) then
            estat_actual <= hemapujat;
            controla <= controla + 2 mod 256;
        elsif ((diferencia >=128) and (derivada < 127)) then
            controla <= controla - 1 mod 256;
            estat_actual <= hemabaixat;
        end if;
    end case;
end process;

-- faltara implementar un control de la variable controla, es a dir que no su
-- peri els 255 ni baixi de 0.

tensio_control <= conv_std_logic_vector(controla,8);

END comportament;

```


Annex F: CD amb el tots els fitxers digitals generats en la realització d'aquest projecte

El CD que s'adjunta amb aquesta memòria conté tots els fitxers que s'han generat en aquest projecte. Aquest inclou aquesta mateixa memòria, tots els models de sincronisme implementats sobre *Simulink*, programes de *Matlab*, programes de *Quartus II*, circuits impresos per *Protel DXP*, documentació consultada i altres fitxers.