

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M.Eng.

PAR
LACASSE, David

ÉMULATION TEMPS RÉEL DE SIGNAUX DE NAVIGATION SATELLITE

MONTRÉAL, LE 9 JANVIER 2009

© David Lacasse, 2009

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Jean Belzile, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Ammar B. Kouki, codirecteur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. René Jr. Landry, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Jocelyn Dore, membre du jury
Agence Spatiale Canadienne

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 11 DÉCEMBRE 2008

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ÉMULATION TEMPS RÉEL DE SIGNAUX DE NAVIGATION SATELLITE

LACASSE, David

RÉSUMÉ

Depuis quelque temps, il apparaît fréquemment de nouveaux produits sur le marché contenant un récepteur GPS. Les cellulaires en sont un exemple. Les récepteurs GPS se généralisent depuis plusieurs années dans les voitures avec leur coût abordable. Cette disponibilité à faible prix en fait un atout majeur pour les flottes de véhicules tel que la poste, la livraison, le transport de marchandises, etc. Un nouveau marché s'ouvre alors pour la gestion des grandes flottes de véhicules, en particulier les véhicules de polices et les taxis.

Le marché de l'aviation n'est pas en reste non plus. Ce marché des récepteurs GPS de précision ne fait que s'accroître avec l'augmentation du trafic aérien.

Pour les concepteurs de récepteur GPS et du futur Galileo, il s'avère important d'augmenter l'efficacité de conception et de vérification des divers produits. La méthode la plus utilisée à cause de son faible coût est de disposer une antenne sur le toit d'un édifice et de faire passer 50 mètres de câbles et plus pour se rendre au laboratoire de test. C'est quand même efficace malgré la perte de qualité de signal dans le câble. Par contre, cela permet seulement de vérifier une position fixe. Pour tester les récepteurs en mouvements, il faut sortir à l'extérieur dans un véhicule avec un équipement volumineux souvent peu abordable puisqu'il faut un équipement portable. La portabilité coûte habituellement beaucoup plus cher.

La solution à ces problèmes est de posséder un simulateur de signaux GPS et Galileo. Sur le marché actuel, il y a déjà des simulateurs de signaux GPS pouvant s'élever à plus de 500 000\$. Le LACIME est en train de développer un simulateur de signaux GPS et Galileo permettant de diviser par 5 et même par 10 le coût d'achat d'un simulateur. Le présent mémoire expliquera brièvement les différentes sections de ce projet d'envergure de même que la conception en détail d'un circuit imprimé comprenant des convertisseurs numériques à analogiques servant à sortir les signaux GPS et Galileo en bande de base.

Ce projet est basé sur un FPGA (*Field-Programmable Gate Array*) qui permet d'avoir une architecture flexible pour suivre les développements futurs de Galileo. Cette approche permet aussi de mettre à jour le matériel tout en effectuant seulement des modifications mineures au logiciel.

Des résultats préliminaires du simulateur seront présentés de même que les résultats de la conception du circuit imprimé. Une nette amélioration de la plage dynamique de plus de 20 dB a été observée avec la nouvelle carte des convertisseurs numériques à analogiques comparée à la carte existante. Du côté de la réception GPS on remarque un léger lissage des résultats dû à une meilleure précision. Mais tant que le simulateur n'est pas complété du côté logiciel, on ne peut vérifier l'ampleur et la qualité des résultats qu'il sera possible d'obtenir.

REAL TIME EMULATION OF NAVIGATION SATELLITE SIGNAL

LACASSE, David

ABSTRACT

Since the beginning of the 21st century, there has been a marked increase in the number of mass market products that integrate GPS receivers. While cellular phones are only the latest example, GPS receivers have been present in cars for several years. They have proven to be a major asset for fleets of vehicles such as mail delivery, freight transport, etc. As costs decrease even further, new market open for the management of large fleets of vehicles, especially police vehicles and taxis.

Simultaneously with this cost decrease we also see an increase in precision. These factors are contributing to the appearance of GPS in the aviation market. The market for precision GPS receivers is increasing as fast as the air traffic.

To take advantage of these opportunities, it is important for GPS and Galileo manufacturers to reduce time-to-market. For the designer, a major time consuming task is the test and validation of a receiver. The most common method because of its low cost is to have an antenna on the roof of a building and to bring the signal to the laboratory via 50 meters or more of cable. This technique has two major drawbacks. First there is a significant signal loss in the cable but more importantly it can only test for a static position. Doing vehicular tests is much more expensive and still has geographical limitations.

In this master's thesis we present the development of a real-time hybrid GNSS (Global Navigation Satellite System) simulator that addresses these concerns. GPS signals simulators are available on the market at a cost of \$500 000 or above. This project attempts to develop a hybrid (GPS and Galileo) GNSS simulator that reduces the cost of a simulator by a factor of 5 to 10 while maintaining the signal quality. The thesis briefly explains the different sections of this project but more specifically the detailed design of a printed circuit including digital to analog converters used to output the GPS and Galileo signals at an intermediate IF of 70 MHz.

This system is based on a Field-Programmable Gate Array (FPGA) that has the flexibility to follow the evolution of the Galileo system. This approach also allows hardware updates with minimum software changes.

Preliminary results of the new digital processing board will be presented. An improvement in dynamic range of more than 20 dB was observed with the new system compared to a commercially available signal processing board.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 DESCRIPTION DU PROJET GNSS	4
1.1 Problématique des simulateurs actuels	4
1.2 Architecture du projet GNSS	5
1.2.1 Partie logicielle	6
1.2.2 Plateforme de traitement numérique en temps réel.....	8
1.2.3 Partie Analogique.....	9
1.3 Interface IF-RF.....	11
1.4 Conclusion	12
CHAPITRE 2 DESCRIPTION DE LA PLATEFORME DE TRAITEMENT NUMÉRIQUE EN TEMPS RÉEL	13
2.1 Problématique de la partie numérique du système	13
2.2 Architecture haut niveau du FPGA	14
2.3 Description des différents blocs du canal GNSS	18
2.4 Conclusion	22
CHAPITRE 3 CONCEPTION D'UN CONVERTISSEUR NUMÉRIQUE-ANALOGIQUE HAUTE VITESSE	23
3.1 Problématique et définition des besoins	23
3.2 Recherche de pièces correspondant aux besoins	24
3.3 Analyse des besoins pour l'alimentation (consommation de courant)	25
3.3.1 Distributeur d'horloge AD9512.....	26
3.3.2 DAC AD9777	27
3.3.3 Consommation totale	29
3.4 Plages de fréquences à filtrer pour l'alimentation	30
3.5 Distribution des condensateurs de découplage	31
3.6 Conception des filtres d'alimentation	32
3.7 Filtre en Pi (π)	35
3.8 Adaptation d'impédance	39
3.9 Création des « footprints » et analyse du placement des pièces	43
3.10 Longueur des traces	44
3.11 Séparation de la masse (« ground »).....	45
3.12 Conseils à prendre en note pour un bon résultat	45
3.13 Conclusion	46
CHAPITRE 4 RÉSULTATS	47
4.1 Comparaison entre les cartes DAC (Lyrtech — AD9777).....	47
4.1.1 Oscilloscope (Tektronix TDS694C)	47
4.1.2 Analyseur de spectre (Rohde & Schwarz FSQ40).....	50

4.2	Résultats avec carte RF	56
4.3	Résultats avec GPS et Galileo.....	57
4.4	Photos des différentes cartes.....	59
CONCLUSION.....		64
RECOMMANDATIONS		65
ANNEXE I	REGISTRES DU FPGA	66
ANNEXE II	EXTRAITS DES REQUIS DU PROJET SIMULATEUR GPS/GALILEO (TRADUCTION LIBRE)	77
ANNEXE III	NOTES SUR LES REQUIS.....	79
ANNEXE IV	SCHEMA GLOBAL DE LA CARTE DAC.....	116
ANNEXE V	SCHEMA DU AD9777	117
ANNEXE VI	SCHEMA DU AD9512.....	118
ANNEXE VII	SCHEMA DU CONNECTEUR DE MEZZANINE.....	119
BIBLIOGRAPHIE.....		120

LISTE DES FIGURES

	Page
Figure 1.1 Architecture du Simulateur GNSS.	5
Figure 1.2 Photo du simulateur.	6
Figure 1.3 Interface globale du simulateur.	7
Figure 1.4 Simulateur de trajectoire.	8
Figure 1.5 Plateforme RF initiale.	10
Figure 1.6 Plateforme RF finale.	11
Figure 2.1 Architecture haut niveau du FPGA.	15
Figure 2.2 Arbre d'horloges.	17
Figure 2.3 Arbre d'horloge et NCO.	19
Figure 2.4 Diagramme représentant un délai.	19
Figure 2.5 Registres, Bus FSB et XMEM.	20
Figure 2.6 Canal hybride pour GPS et Galileo.	21
Figure 3.1 Représentation des relations entre les composants du projet.	25
Figure 3.2 Tableau des puissances consommées de l'AD9512.	26
Figure 3.3 AD9777 Consommation I_{DVDD}	28
Figure 3.4 AD9777 Consommation I_{AVDD}	28
Figure 3.5 AD9777 Consommation I_{CLKVDD}	29
Figure 3.6 Impédance et ESR des condensateurs affichés avec KEMET Spice.	32
Figure 3.7 Power-Supply Ripple Rejection.	33
Figure 3.8 Représentation globale du filtrage d'alimentation.	34
Figure 3.9 Filtre en Pi avant le régulateur.	35
Figure 3.10 Transformation du filtre en Pi en réseau d'admittance pour le calcul.	36

Figure 3.11	Comparaison entre ferrites de 11 ohms et de 600 ohms à 100 MHz.....	38
Figure 3.12	Graphique d'impédance des ferrites 11 et 600 ohms.	39
Figure 3.13	LVPECL dans une ligne de transmission parallèle.	40
Figure 3.14	Ligne microstrip couplée 50 ohms de LineCalc.....	41
Figure 3.15	Ligne de transmission CMOS.	42
Figure 3.16	Ligne microstrip de 50 ohms de LineCalc.	42
Figure 4.1	Sinus de 30 MHz de Lyrtech.....	48
Figure 4.2	Sinus de 30 MHz du AD9777.	49
Figure 4.3	Sinus de 20 MHz avec modulation à 70 MHz du AD9777.....	49
Figure 4.4	Sinus de 30 MHz de Lyrtech sur 100 MHz.....	50
Figure 4.5	Sinus de 30 MHz du AD9777 sur 100 MHz.	51
Figure 4.6	Sinus de 30 MHz de Lyrtech sur 400 MHz.....	52
Figure 4.7	Sinus de 30 MHz du AD9777 sur 400 MHz.	52
Figure 4.8	Sinus de 20 MHz modulé à 70 MHz du AD9777 sur 400 MHz.	53
Figure 4.9	Harmonique à 70 MHz du sinus de 30 MHz de Lyrtech.....	55
Figure 4.10	Harmonique à 70 MHz du sinus de 30 MHz du AD9777.....	55
Figure 4.11	Sinus de 20 MHz modulé à 70 MHz du AD9777.	56
Figure 4.12	Sortie de la carte RF modulation GPS.....	57
Figure 4.13	Résultat du récepteur GPS sur 20 minutes (Lyrtech).	58
Figure 4.14	Résultat du récepteur GPS sur 20 minutes (AD9777).....	58
Figure 4.15	Résultats du récepteur GPS en temps normal.	59
Figure 4.16	Carte de Lyrtech sans le boîtier.....	60
Figure 4.17	Carte de Lyrtech avec boîtier.	61
Figure 4.18	Carte AD9777 initiale.	61

Figure 4.19	Dessus de la Carte AD9777 finale.	62
Figure 4.20	Dessous de la Carte AD9777 finale.	62
Figure 4.21	Photo du simulateur avec les nouveaux convertisseurs.....	63

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ADS	Advanced Design System
ASC	Agence Spatiale Canadienne
BPF	Band-pass filter (Filtre passe-bande)
CAN	Convertisseur analogique à numérique
CMOS	Complementary Metal Oxide Semiconductor
CNA	Convertisseur numérique à analogique
C/N ₀	Carrier to Noise Power Density Ratio (Rapport signal à bruit)
cPCI	CompactPCI ou Compact Peripheral Component Interconnect
CPLD	Complex Programmable Logic Device (Circuit logique complexe programable)
DAC	Digital to Analog Converter (Convertisseur numérique à analogique)
DCM	Digital Clock Manager (Contrôleur d'horloge digitale)
DR	Dynamic Range (Plage dynamique)
dBc/Hz	Decibel relative to the carrier per Hz (Décibel relatif à la porteuse par Hertz)
E5	Fréquence Galileo (1191,795 MHz)
E5a	Fréquence Galileo (1176,45 MHz)
E5b	Fréquence Galileo (1207,14 MHz)
E6	Fréquence Galileo (1278,75 MHz)
ESR	Electrical series resistance (Résistance série)
ÉTS	École de technologie supérieure
FPGA	Field Programmable Gate Array (Réseau de portes programmable)
FSB	Fast Synchronous Bus (Bus synchrone rapide)

GLONASS	GLObal'naya NAvigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System (Système de navigation satellite global)
GPS	Global Positioning System (Système de positionnement global)
Hz	Hertz
IF	Intermediate Frequency (Fréquence intermédiaire)
MCLK	Message clock (Horloge du message)
L1	Fréquence GPS (1575,42 MHz)
L2	Fréquence GPS (1227,60 MHz)
L5	Fréquence GPS (1176,45 MHz)
LACIME	Laboratoire de communication et d'intégration de la microélectronique
LO	Local oscillator (Oscillateur local)
LUT	Look-Up Table
NCO	Numerically Controlled Oscillator (Oscillateur contrôlé numériquement)
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
PCLK	Primary Clock (Horloge primaire)
PI	Lettre grec (π)
PLL	Phase Locked Loop (Boucle à verrouillage de phase)
PRN	Pseudo Random Noise (Bruit pseudo-aléatoire)
ps	Pico seconde (10^{-12} seconde)
PSRR	Power supply rejection ratio (Ratio de réjection de l'alimentation)
RF	Radiofréquence

SCLK	Secondary Clock (Horloge secondaire)
SDRAM	Synchronous Dynamic Random Access Memory (Mémoire dynamique synchrone)
SNR	Signal to Noise Ratio (Rapport signal à bruit)
Soft	Software (Logiciel)
VHDL	VHSIC Hardare Description Language (Langage de description matériel VHSIC)
VHSIC	Very High Speed Integrated Circuit (Circuit intégré très haute vitesse)
Xmem	eXternal MEMory (Mémoire externe)

INTRODUCTION

Depuis le début des années 2000, l'utilisation des récepteurs GPS (Global Positioning System) est rendue omniprésente. Que ce soit pour les voyages, la randonnée, le repérage de voiture, le suivi d'une flotte de livraison ou la guerre. Jusqu'à présent, les États-Unis étaient les seuls à fournir un système fiable comparé au système Russe GLONASS (GLObal'naya NAVigatsionnaya Sputnikovaya Sistema). Ce monopole géré par les militaires sera brisé prochainement par des nouveaux venus sur le marché de la navigation par satellite, la Chine et l'Europe. La Chine avec son système « Beidou » (désigne l'astérisme du « chariot » de la constellation de la Grande Ourse) est déjà opérationnelle avec cinq satellites géostationnaires couvrant la Chine et certains pays connexes. La couverture du reste du monde suivra avec 35 satellites orbitaux d'ici 2012). L'Union européenne de son côté arrive avec Galileo sous contrôle civil au lieu de militaire comme le GPS. En date de mai 2008, seulement deux satellites de test sont en orbite autour de la planète. Le système devrait être utilisable en 2010 et pleinement opérationnel en 2013.

Le marché de la navigation par satellite étant en expansion, un programme de recherche et développement fut mis en route par le LACIME (Laboratoire de communication et d'intégration de la microélectronique) avec la collaboration de l'ASC (Agence Spatiale Canadienne) et de CMC électronique pour développer un simulateur de signaux hybride GPS et Galileo. Ce simulateur a pour but de simuler les constellations de satellites, ce qui permettrait de tester des récepteurs GNSS (Global Navigation Satellite System).

Pour tester un récepteur GPS, les méthodes utilisées sont très contraignantes. La première nécessite l'installation d'une antenne sur le toit de l'édifice. Avec plusieurs dizaines et même de centaines de mètres de câble pour relier l'antenne au laboratoire de test, on obtient une perte non négligeable de puissance et de qualité du signal. Les tests sont donc limités et par le fait même contraints à tester seulement une position fixe. La deuxième méthode consiste à monter l'équipement de test dans un camion et de partir sur la route pour tester le

récepteur sur une trajectoire donnée. Cette méthode est très coûteuse en temps et argent et se limite à la région où se situe le laboratoire.

Les simulateurs existants pour GPS fournissent une grande flexibilité permettant de simuler des trajets en avion ou en voiture. Ils permettent de simuler la constellation de satellites de n'importe quel endroit dans le monde. Que ce soit une position vue de l'équateur ou du pôle Nord. Ces possibilités se retrouvent toutes dans un même boîtier permettant de tester toutes les fonctionnalités du récepteur tout en restant dans le laboratoire de test.

Le simulateur développé par le LACIME aura des fonctionnalités similaires aux simulateurs GPS existants, mais fournira aussi la simulation Galileo. L'utilité d'un simulateur hybride GPS et Galileo est accentuée par l'absence de suffisamment de satellites Galileo pour obtenir une position. Il est donc très difficile de tester de façon efficace d'éventuels récepteurs de signaux Galileo sans un simulateur.

Pour concevoir ce simulateur, il est nécessaire d'avoir trois parties distinctes : une partie logicielle, une partie numérique et une partie analogique.

La partie logicielle permet d'entrer les paramètres à simuler tels que la date et l'heure de simulation, les coordonnées de départ et la trajectoire. Elle permet aussi de contrôler la partie matérielle numérique du projet à travers d'un port compact PCI (cPCI). La partie numérique se compose d'une carte de développement FPGA pour faire le traitement des signaux. Ce traitement permet de créer les signaux de sorties GPS et Galileo à partir des informations reçues de la partie logicielle. Ces signaux GPS et Galileo doivent traverser une carte avec des convertisseurs numériques-analogiques pour ensuite rejoindre la partie analogique. La partie analogique est composée d'une carte électronique permettant de prendre les signaux GPS et Galileo générés par les convertisseurs numériques-analogiques et de les sortir à la fréquence utilisée par les satellites. De cette façon, n'importe quel récepteur GPS ou Galileo sera en mesure de capter ces signaux et de révéler la position programmée par la partie logicielle.

Le présent mémoire expliquera le cheminement de conception du simulateur et présentera en profondeur l'implémentation de la partie numérique du projet. Le Chapitre 1 fournira une description globale du projet GNSS avec ses trois parties majeures. Le Chapitre 2 décrira la partie matérielle numérique plus en détail avec ses différents composants. Le Chapitre 3 détaillera la conception de la carte avec les convertisseurs numériques-analogiques qui sert d'interface entre le numérique et l'analogique. Le Chapitre 4 terminera ce mémoire en démontrant les résultats obtenus par les différentes parties du simulateur. Ces parties seront décrites dans les prochaines pages.

Il est à noter que ce projet multidisciplinaire a débuté environ un an avant mon arrivée dans l'équipe de projet en janvier 2007. Le projet a reçu la contribution de plusieurs personnes, ce qui se reflètera par la présence de plusieurs informations dans ce document provenant des différents collaborateurs au projet. Ces informations serviront à faciliter la compréhension de l'ensemble du projet. Pour ma part j'ai travaillé principalement sur la partie numérique du projet qui consiste en un programme VHDL et la conception d'une carte électronique. Par contre j'ai aussi beaucoup travaillé sur les autres parties du projet pour aider à l'intégration et au déverminage de l'ensemble du projet. Pour aider à se situer dans la partie numérique du projet, une brève description de l'ensemble des parties sera effectuée au Chapitre 1.

CHAPITRE 1

DESCRIPTION DU PROJET GNSS

Comme il fut mentionné précédemment, il faut trois parties principales pour concevoir un simulateur GNSS. Ce chapitre décrira l'architecture globale des parties logicielles, numériques et analogiques de même que la problématique reliée aux simulateurs actuels.

1.1 Problématique des simulateurs actuels

Au départ de ce projet, il n'existait pas de simulateur de signaux Galileo sur le marché. Il y avait seulement des simulateurs de signaux GPS, mais ils étaient extrêmement chers et peu flexibles. C'était donc une opportunité pour développer un simulateur hybride GPS et Galileo. Par la suite, des simulateurs de signaux Galileo sont sortis sur le marché, mais ils étaient encore une fois extrêmement chers (autour de 800 000\$ pour celui de SPIRENT sorti en avril 2008).

L'objectif était donc d'offrir un simulateur hybride GPS et Galileo offrant plusieurs canaux de chaque simultanément à un coût nettement inférieur à ceux existants.

Pour atteindre ce but, nous avons choisi d'utiliser une plateforme de développement FPGA (Field Programmable Gate Array). L'architecture de ce programme a été conçue pour s'assurer une grande flexibilité puisque les spécifications des satellites Galileo n'étaient pas finales à ce moment. Cette flexibilité se traduit par l'utilisation de registres dans le FPGA permettant de modifier le schéma de modulation des signaux de sortie du FPGA. La flexibilité passe aussi par une interface logicielle polyvalente permettant les changements d'une multitude de paramètres de la génération des signaux.

Un bref aperçu des différentes parties du simulateur sera donné aux sous-sections suivantes. Il est à noter que ce projet fut mené par une équipe comprenant des compétences variées où chacun avait sa propre section à développer. Comme mentionné précédemment, ma tâche

était surtout axée sur le développement de la partie numérique du projet. Mais puisque la partie numérique se retrouve entre le logiciel et la partie analogique, j'ai eu à travailler sur toutes les sections du projet. Les parties majeures des sections logicielle et analogique ont donc été conçues par une tierce personne, mais seront quand même détaillées dans les sections suivantes pour faciliter la compréhension globale du projet.

1.2 Architecture du projet GNSS

Le simulateur GNSS comporte trois sections majeures : les parties logicielle, numérique et analogique. On peut voir sur la Figure 1.1 ces différentes parties reliées ensemble pour aboutir au final à un récepteur GPS pour valider les résultats.

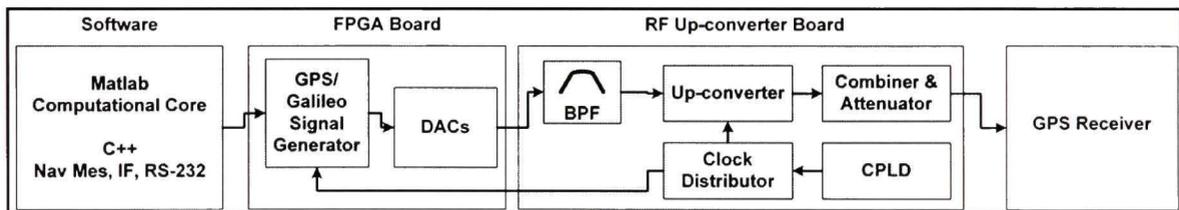


Figure 1.1 Architecture du Simulateur GNSS.

La Figure 1.2 nous montre le simulateur dans sa forme matérielle. Il est à noter que cette version a des limitations matérielles du côté de l'interface entre le numérique et l'analogique. Ces limitations proviennent de la carte avec les convertisseurs numériques-analogiques (CNA ou « DAC » en anglais) et seront expliquées au CHAPITRE 3. La section suivante présentera globalement la partie logicielle.

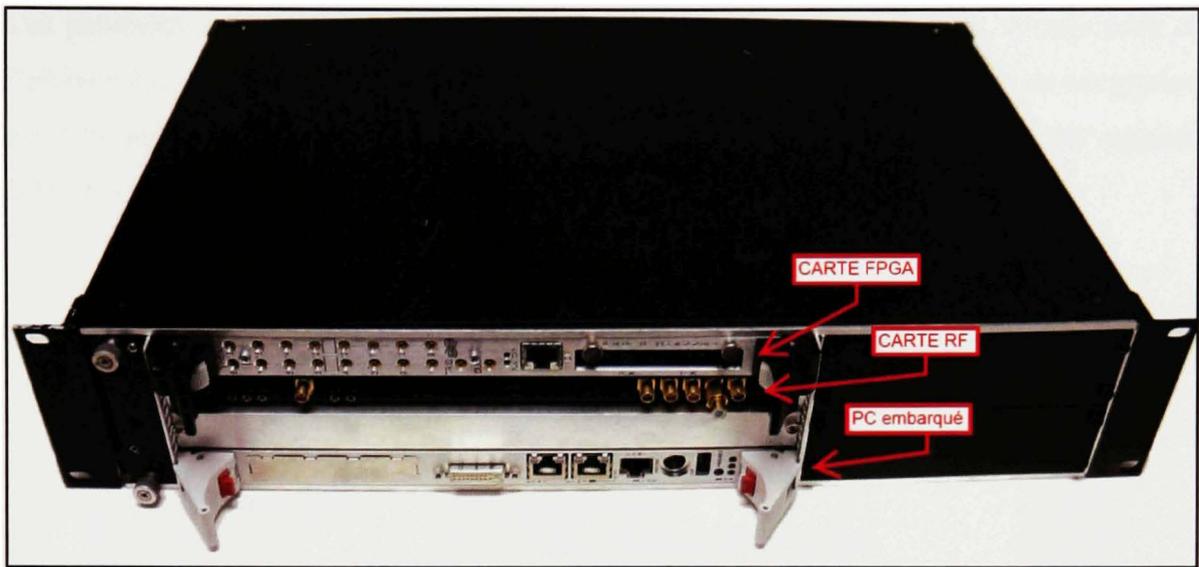


Figure 1.2 Photo du simulateur.

1.2.1 Partie logicielle

La partie logicielle, plus communément appelée « soft », est constituée d'un programme de base en MATLAB (MATLAB core) de même que d'une interface en C++ pour communiquer avec la partie numérique du projet. La partie MATLAB sert à configurer le scénario voulu pour la trajectoire et calculer les différents paramètres servant à la construction des messages provenant des différents satellites. Ces paramètres sont extrêmement nombreux et requièrent une puissance de calcul non négligeable. Ces différents paramètres sont ensuite mis en forme par la partie C++ pour être transférés à la partie numérique via le port compact PCI (cPCI) de la plateforme de développement.

Nous pouvons voir à la Figure 1.3 et Figure 1.4 différentes fenêtres de la partie logicielle conçue par des collaborateurs. Ces interfaces sont importantes puisqu'elles permettent de tester le mode fonctionnel du simulateur. Sans ces interfaces il est seulement possible de tester chaque composant du simulateur individuellement. Les interfaces permettent donc de tester la fonction globale du simulateur et de faire des modifications en temps réel sur les différents paramètres de contrôle.

Ces paramètres tels que la date et l'heure, les fichiers Almanachs ou bien la trajectoire et l'altitude font partie des nombreux calculs nécessaires pour obtenir un message de navigation qui sera transmis par le port cPCI à la partie matérielle numérique pour générer les signaux GPS et Galileo.

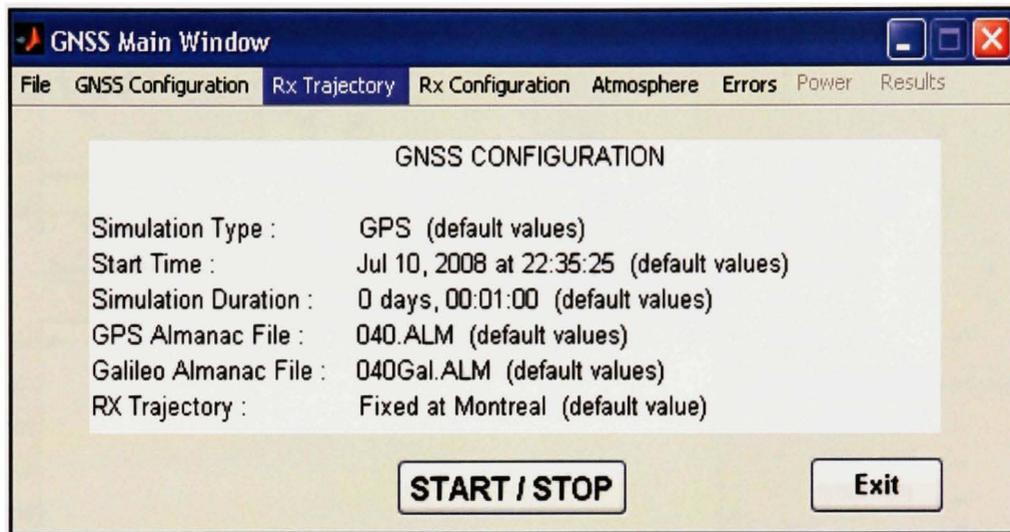


Figure 1.3 Interface globale du simulateur.

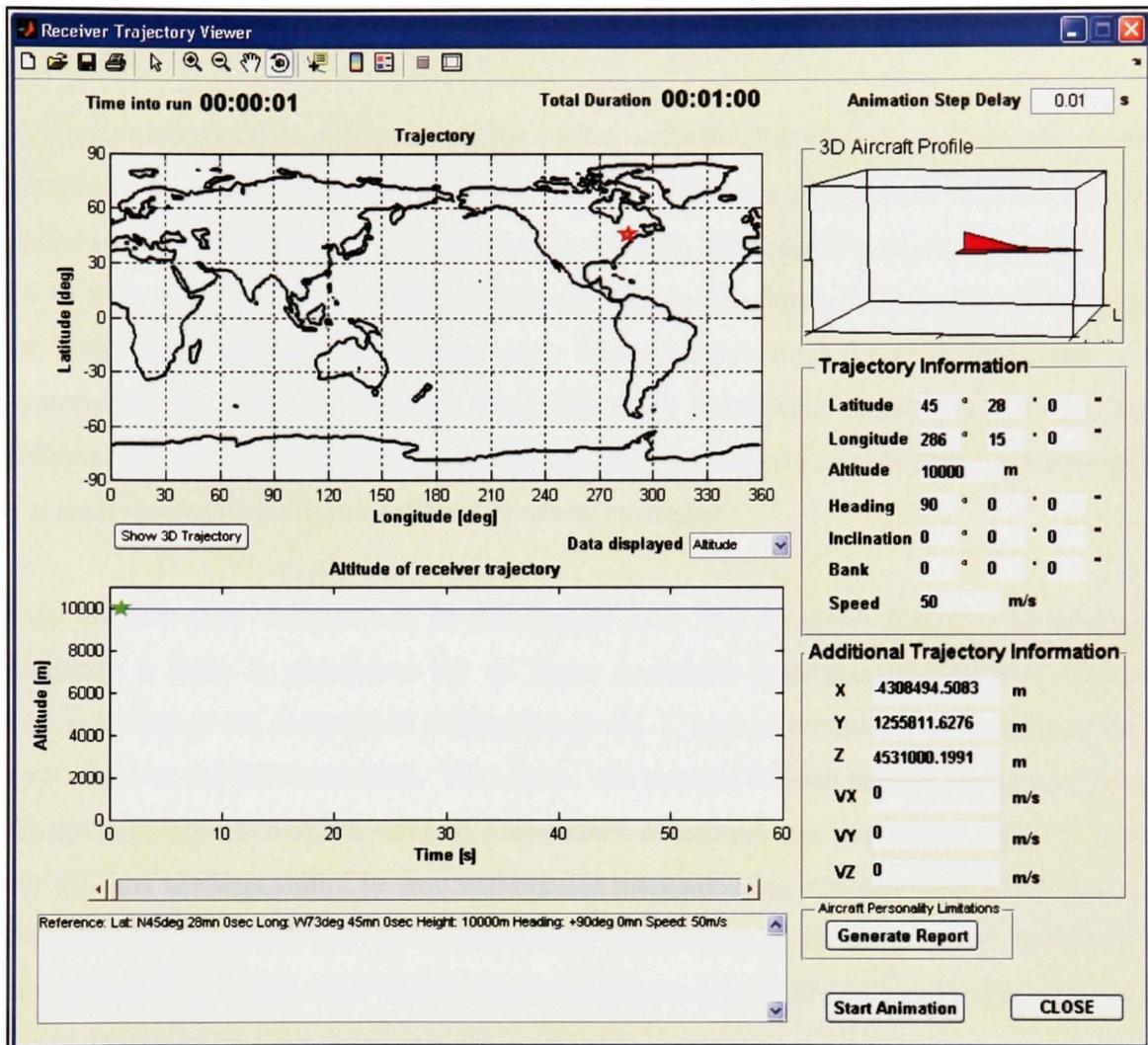


Figure 1.4 Simulateur de trajectoire.

1.2.2 Plateforme de traitement numérique en temps réel

La partie numérique, appelée « partie IF » (« Intermediate Frequency » ou Fréquence intermédiaire), sert à générer le signal GPS ou Galileo à partir des informations provenant de la partie logicielle. Les principaux composants de cette section seront détaillés au CHAPITRE 2. Les signaux GPS et Galileo sont générés au travers d'un CNA (Convertisseur Numérique-Analogique), ou plus communément appelé « DAC » (Digital to Analog Converter). Les signaux GPS et Galileo sortent des DACs à 70 MHz pour ensuite être envoyés à la partie analogique.

1.2.3 Partie Analogique

La partie analogique du système, appelée « RF » (« Radio Frequency » ou Fréquence radio), effectue la translation du signal IF vers les hautes fréquences utilisées par les satellites GPS et Galileo. Ces fréquences se situent au-delà du GHz. Par exemple, un signal de type GPS L1 se situe à 1575.42 MHz. L'architecture de la carte RF utilisée dans le système comporte un distributeur d'horloge fournissant trois PLL (« Phase Lock Loop») qui à leur tour syntonisent les trois LO (« Local Oscillator » ou Oscillateur local) donnant les trois fréquences utilisées par le simulateur. Un contrôle de gain est aussi utilisable en sortie pour s'assurer que les trois canaux sortent à la même puissance.

Pour réaliser cette architecture, la conception a eu lieu en deux phases. La première consistait à créer la plateforme RF de façon modulaire pour pouvoir tester et modifier aisément chacun des composants indépendamment. L'approche modulaire fut utile lors de la conception et des différents tests. Par contre, elle pouvait devenir pénible lorsque le design fut arrêté puisque le nombre élevé de connecteurs constituait une importante source d'erreur par ses faux contacts suite à de trop nombreuses manipulations. On remarque à la Figure 1.5 l'ampleur de la plateforme RF initiale. Ma contribution à ce boîtier s'est limitée à la conception d'une petite carte électronique comprenant un CPLD (« Complex Programmable Logic Device » ou Circuit logique programmable complexe). Ce circuit se programmant avec le langage de programmation VHDL sert à contrôler et programmer le distributeur d'horloge et les PLLs de la chaîne RF pour être aux bonnes fréquences.

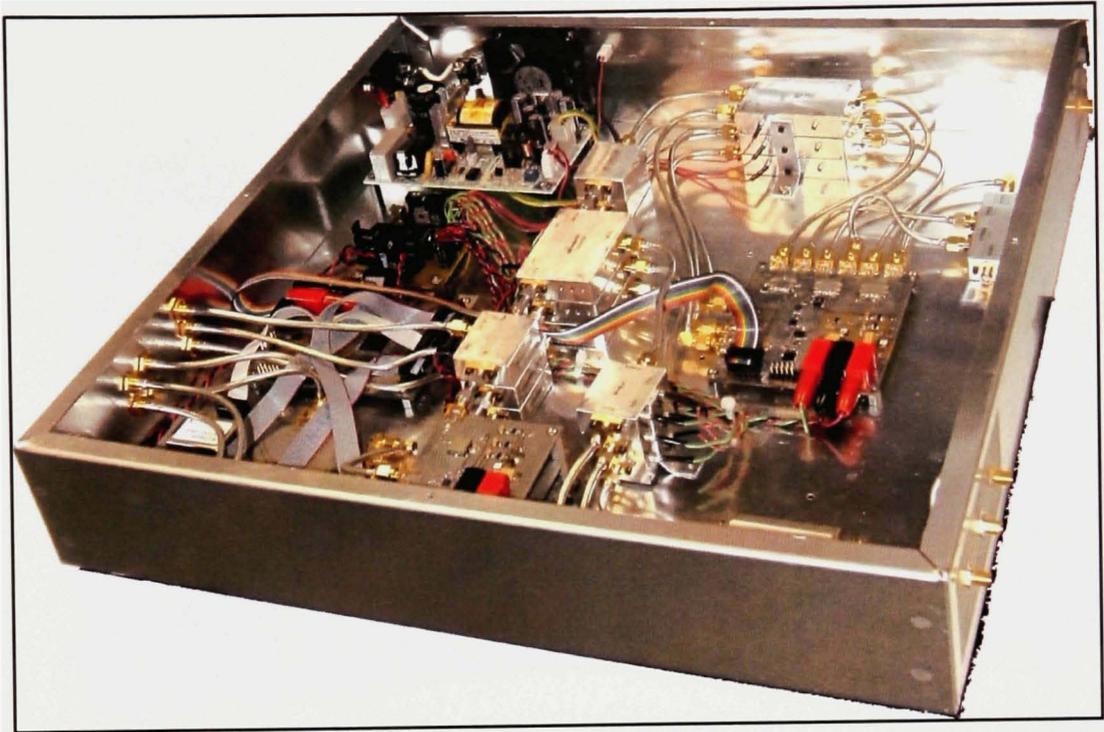


Figure 1.5 Plateforme RF initiale.

Lorsque les tests préliminaires furent concluants, la seconde phase consistait à créer une carte électronique regroupant tous les composants au même endroit de façon concise. La Figure 1.6 représente la carte finale créé par un collaborateur. Cette carte a été conçue pour s'insérer dans le boîtier de la plateforme globale tel qu'aperçu à la Figure 1.2.

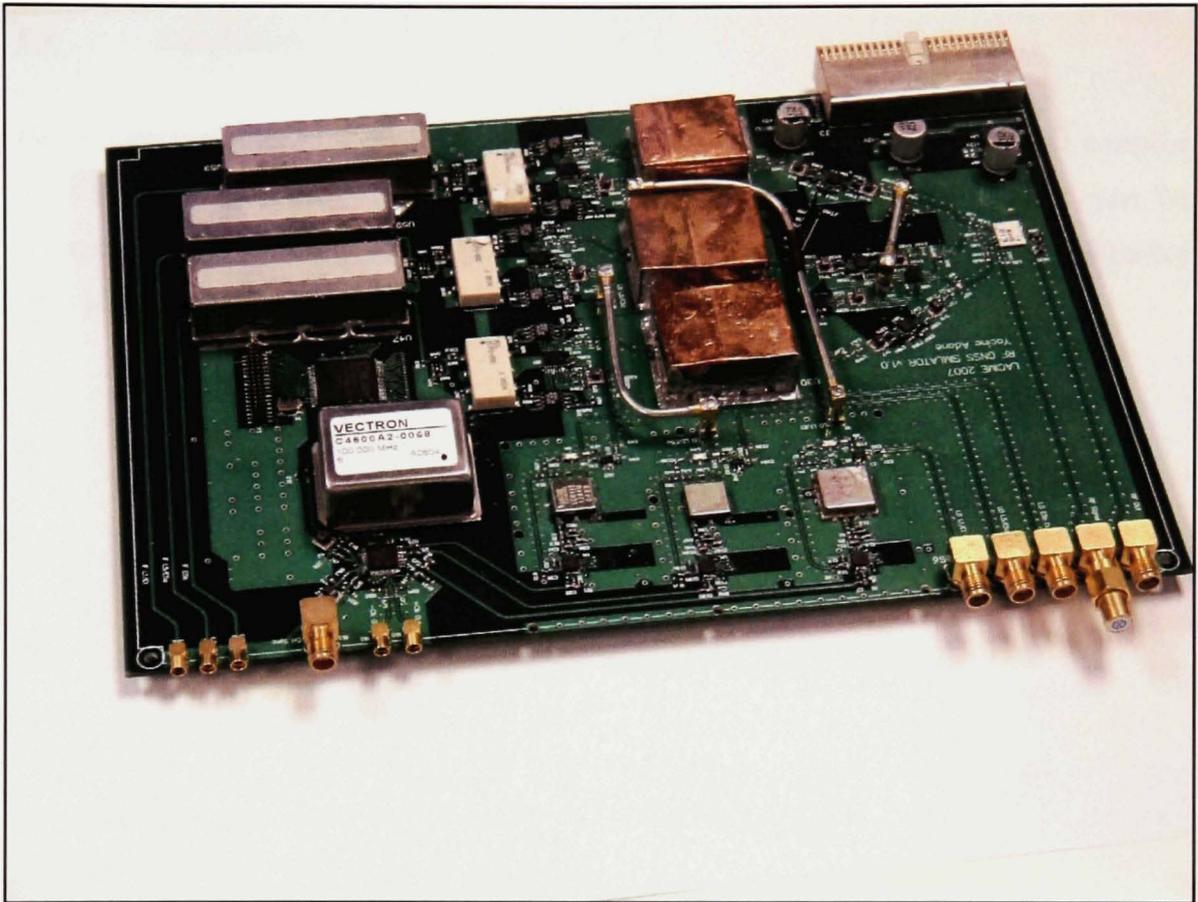


Figure 1.6 Plateforme RF finale.

Pour s'assurer un signal de qualité à la sortie de cette carte, il est important d'avoir une interface de qualité entre la partie numérique et la partie analogique.

1.3 Interface IF-RF

L'interface entre la partie IF et la partie RF est constituée principalement de la carte fille comprenant les convertisseurs numériques à analogiques de même qu'un distributeur d'horloge. Cette carte prend comme entrées les données numériques provenant du FPGA et les distribues aux trois DACs qui sortiront le signal analogique requis à l'entrée de la partie RF. Cette carte possède six sorties analogiques dont trois sont connectées à la carte RF. La carte fille sera détaillée dans les prochains chapitres.

1.4 Conclusion

Ce chapitre a donné un aperçu des trois sections essentielles au projet. On peut conclure avec ce survol qu'il peut s'avérer difficile de bien coordonner la communication entre les sections pour mener à bien le projet. Il est maintenant temps d'avoir une vue plus détaillée de la partie centrale du projet que constitue la partie numérique.

CHAPITRE 2

DESCRIPTION DE LA PLATEFORME DE TRAITEMENT NUMÉRIQUE EN TEMPS RÉEL

Pour mener à bien cette partie du projet, nous avons décidé d'acheter une carte de développement FPGA existante sur le marché pour accélérer le développement du projet (Lyrtech VHS-ADC). Ce chapitre décrira en détail les différents composants de la partie numérique du projet. Il est à noter que l'architecture originale a été créée par un collaborateur et corrigée par mes soins. Toutes les figures de ce chapitre ont donc été créées par ce collaborateur mais adaptées et corrigées par moi.

2.1 Problématique de la partie numérique du système

La problématique majeure de la partie numérique vient des spécifications du produit final. Ces spécifications peuvent être retrouvées en ANNEXE II, mais voici un bref aperçu d'une des spécifications reliées à la partie numérique.

- Le simulateur doit être en mesure de simuler 36 signaux comme suit :
 - 12 signaux GPS L1
 - 12 signaux GPS L5
 - 6 signaux Galileo E1
 - 6 signaux Galileo E5a
 - 6 signaux Galileo E5b

Cette spécification est directement reliée aux capacités de mémoire interne du FPGA. C'est cette mémoire interne du FPGA qui permet le traitement en temps réel du simulateur. Bref, plus il y a de mémoire disponible plus il est possible d'entrer de canaux.

Dans le cas de ce projet, les spécifications font état de 36 canaux. Pour l'instant on limite le FPGA à huit canaux pour augmenter la rapidité des tests, mais il serait possible seulement

d'en avoir 16 au total. De futures optimisations du code sont nécessaires pour atteindre les 36 canaux demandés dans les spécifications pour éviter de devoir acheter une carte avec un plus gros FPGA.

Le fonctionnement en temps réel du système avait une autre problématique : la synchronisation. En effet il est crucial vu que l'on utilise plusieurs composants internes au FPGA et que l'on utilise des blocs mémoires pour synchroniser le tout. Cette synchronisation passe par un calcul minutieux de tous les cycles d'horloges nécessaires pour passer au travers chaque composant. On peut donc par la suite ajouter les délais nécessaires sur certains chemins pour que tous les signaux arrivent en même temps au même endroit pour générer le signal de sortie. Il sera plus aisé de visualiser ces problématiques avec de plus amples détails sur l'architecture interne du FPGA.

2.2 Architecture haut niveau du FPGA

L'architecture haut niveau du FPGA est composée de cinq principales sections. Ces sections servent essentiellement à structurer les transferts de données vers le FPGA et à l'intérieur de celui-ci. La Figure 2.1 nous montre le haut niveau (« top level ») du programme FPGA.

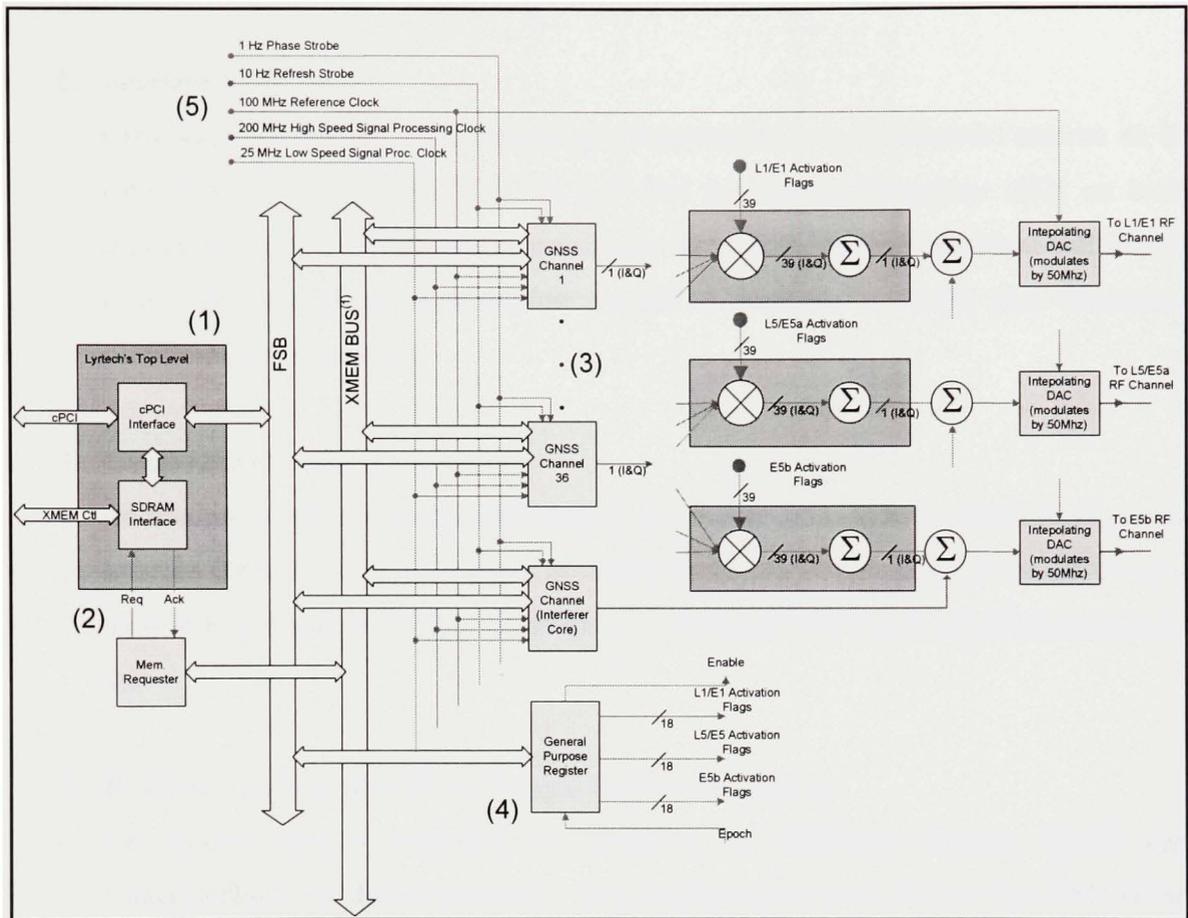


Figure 2.1 Architecture haut niveau du FPGA.

Les différents composants de ce « top level » se décrivent comme suit :

1. Interface cPCI

Cette interface sert exclusivement à la communication avec la carte mère de la plateforme. L'interface cPCI a été fournie par la compagnie Lyrtech avec leur carte de développement FPGA.

2. Interface SDRAM

Cette interface permet de transférer de l'information vers la mémoire externe de la carte FPGA. Le module SDRAM peut être accédé par l'interface cPCI ou bien directement par le FPGA par l'intermédiaire du « XMEM bus » (eXternal MEMory). L'utilisation de la mémoire externe permet de soulager le FPGA pour permettre d'utiliser plus de canaux.

3. Canal GNSS (« GNSS channel »)

Ce module contient tous les composants nécessaires pour faire fonctionner un canal hybride GPS et Galiléo. L'idéal du projet serait d'être capable d'avoir 36 canaux dans le FPGA plus un canal d'interférence. Ce canal GNSS sera détaillé à la section 2.3.

4. Registres d'utilité générale (« General Purpose Register »)

Ce module sert à contrôler et surveiller les différents canaux GNSS par l'intermédiaire de divers registres. Ces registres sont tous liés au bus FSB (Fast Synchronous Bus) pour la facilité d'implémentation et d'accès.

5. Arbre d'horloge

Ce composant sert à créer les différentes horloges nécessaires au bon fonctionnement du système. Sur la Figure 2.1 on remarque seulement les signaux entrants provenant de l'arbre d'horloge. La Figure 2.2 détaille l'architecture de cet arbre d'horloge.

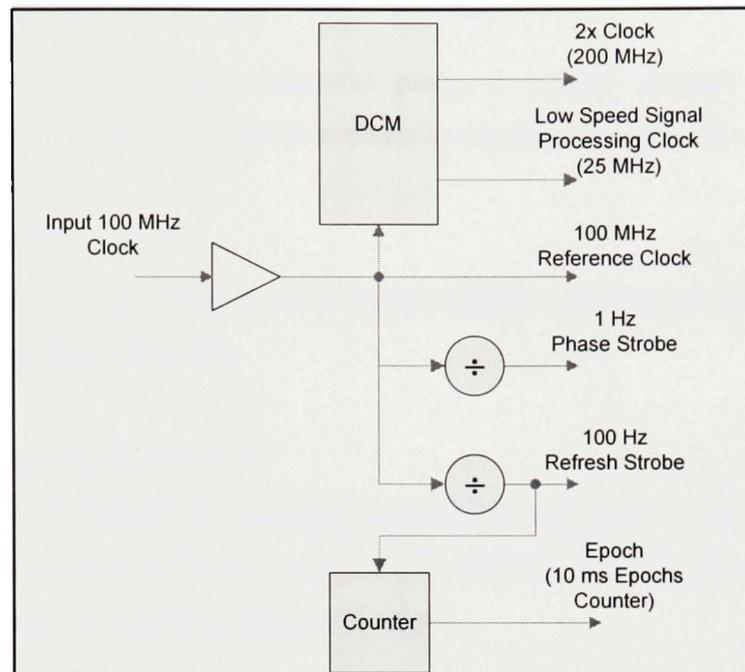


Figure 2.2 Arbre d'horloges.

Pour être en mesure de bien se synchroniser à l'interne avec les bonnes vitesses, il faut être en mesure de bien définir l'utilité de chacun des signaux suivants :

- 200MHz
Cette horloge sert à partager une table de sinus/cosinus entre 2 canaux. Cela permet de sauver de l'espace pour entrer plus de canaux.
- 25MHz
Cette horloge sert à la communication avec le bus cPCI
- 100MHz
Cette horloge sert au fonctionnement du FPGA en général.

- 1Hz
Ce pulse sert à corriger la phase à chaque seconde pour éviter une accumulation d'erreur sur la phase de la porteuse et du code.

- 100Hz
Ce pulse sert à synchroniser les sorties des tampons pour les fréquences de la porteuse et du code.

- 10ms Epoch Counter
Le résultat de ce compteur peut être lu par le PC. Ce compteur sert au programme PC pour savoir où le FPGA est situé dans les tampons des fréquences de la porteuse et du code.

Le canal GNSS discuté précédemment est la pièce maîtresse du programme, c'est ce canal qui met en forme chacun des « satellites » simulés.

2.3 Description des différents blocs du canal GNSS

Le canal GNSS est constitué de plusieurs modules interconnectés. L'arbre d'horloge et les NCO (« Numerically Controlled Oscillator » ou oscillateur contrôlé numériquement) sont représentés à la Figure 2.3. L'arbre d'horloge du canal GNSS fournit les différents pulses nécessaires à la synchronisation des différentes composantes du schéma de modulation tels que le message, la porteuse et le code. Les NCO de leur côté gèrent la table de sinus et cosinus pour la porteuse et le code.

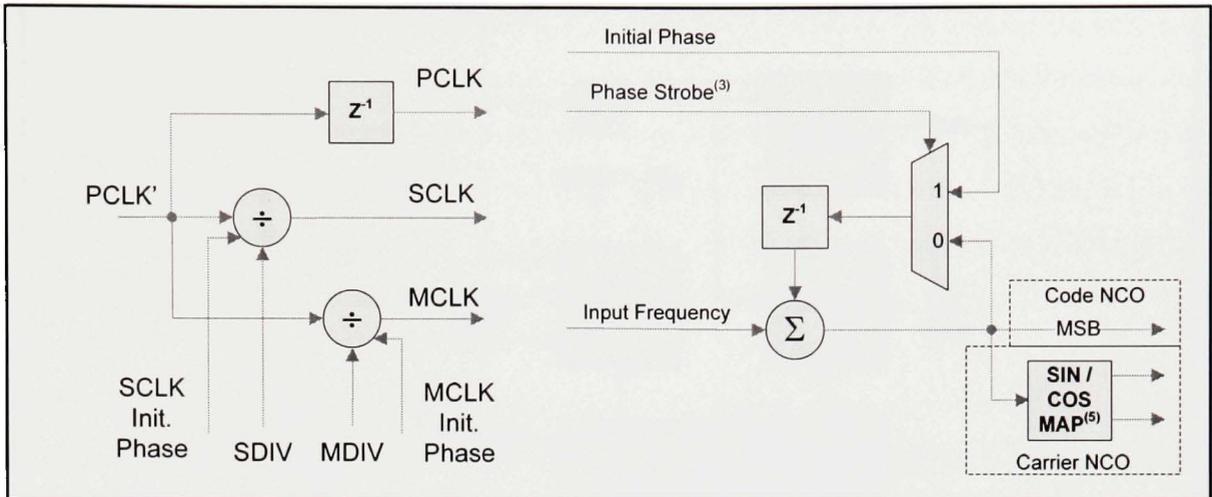


Figure 2.3 Arbre d'horloge et NCO.

La Figure 2.5 nous explique les différentes interactions entre les composants du canal GNSS. Il est possible de voir les nombreux délais ajoutés pour synchroniser les différents composants comme il en avait été question en début de chapitre. Ces délais sont représentés par le diagramme à la Figure 2.4. Le « X » représentant le nombre de cycles d'horloge de délai. Ces délais sont très importants et permettent d'assurer une bonne synchronisation des signaux nécessaires à la modulation GPS.

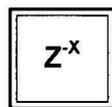


Figure 2.4 Diagramme représentant un délai.

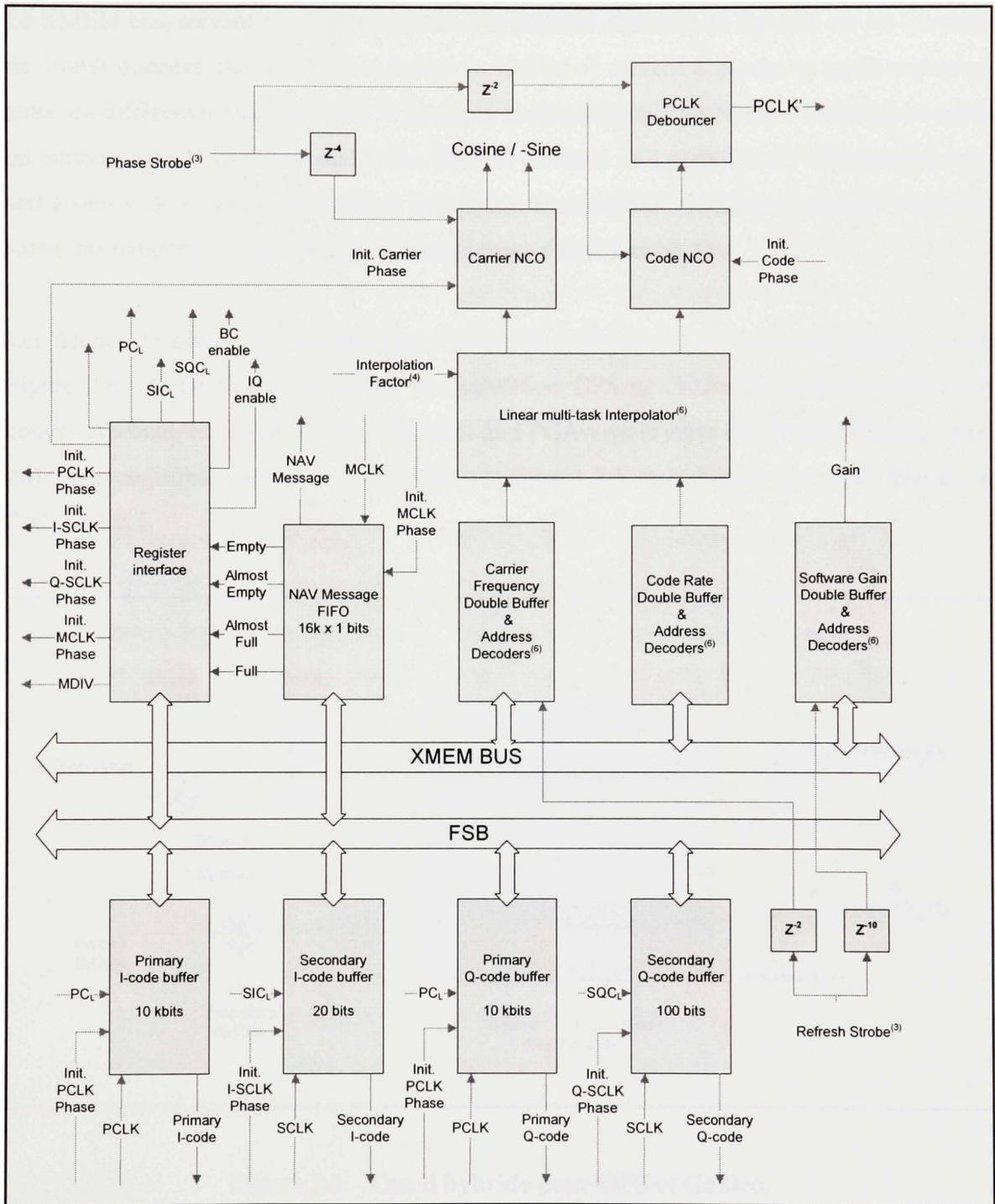


Figure 2.5 Registres, Bus FSB et XMEM.

Le XMEM bus, servant à communiquer avec la mémoire externe, est branché sur les tampons de 10000 données chacun. Ces tampons (« buffers ») servent à garder la synchronisation entre les différentes fréquences de la porteuse et du code pour les NCOs. On retrouve aussi un tampon pour le gain du canal. Plus haut on retrouve un interpolateur. Cet interpolateur sert à sauver de l'espace mémoire en interpolant d'un facteur 10000 entre chaque fréquence sortie des tampons du code et de la porteuse avant d'aller aux NCOs.

Les flèches dirigées vers l'extérieur dans le bas de la Figure 2.5 se trouvent à entrer dans la Figure 2.6. Cette figure représente la modulation GPS et Galileo à partir des différents codes. Au final, les signaux I et Q sortiront du FPGA vers la carte fille avec les DACs. Les différents registres que l'on retrouve dans la Figure 2.5 et la Figure 2.6 sont détaillés à l'ANNEXE I.

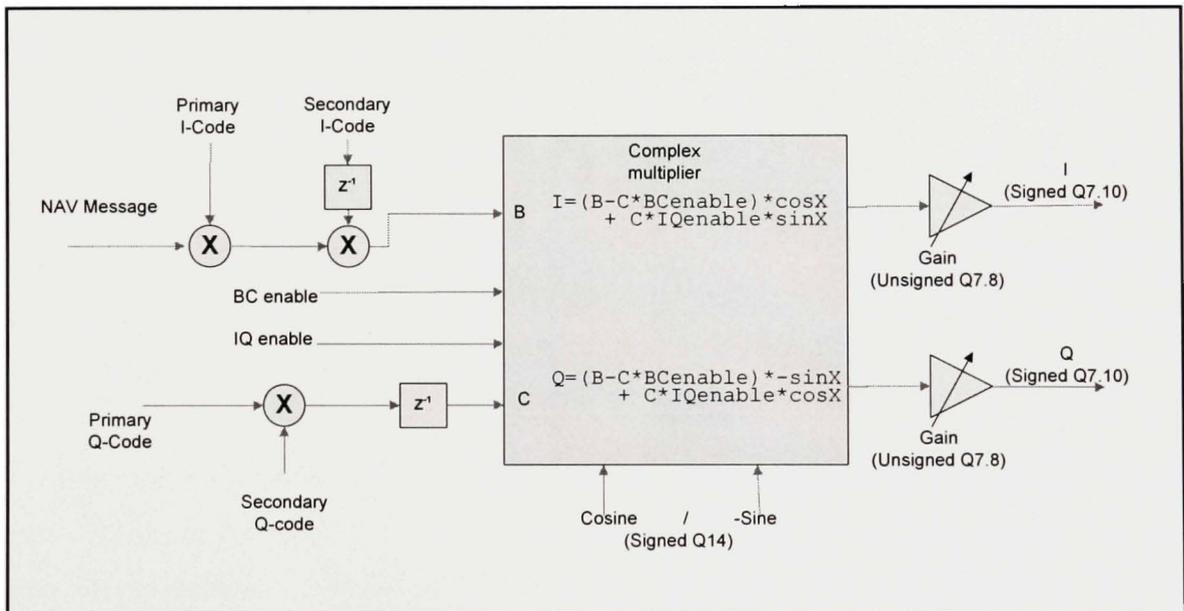


Figure 2.6 Canal hybride pour GPS et Galileo.

2.4 Conclusion

Ce chapitre a servi à donner une vue détaillée de l'architecture VHDL de la partie numérique. Nous avons parlé précédemment de l'interface entre la partie numérique et la partie analogique du projet. La carte fille avec ses convertisseurs numériques-analogiques provenant de la carte de développement FPGA ne convenait pas aux requis énoncés à l'ANNEXE III. Nous avons donc choisi de concevoir une nouvelle carte mezzanine pour respecter les requis. Il est important que ces DACs soient bien conçus pour s'assurer une précision maximale des signaux de sorties vers la plateforme RF. Le chapitre suivant sera entièrement consacré aux différentes étapes de conception de cette carte mezzanine.

CHAPITRE 3

CONCEPTION D'UN CONVERTISSEUR NUMÉRIQUE-ANALOGIQUE HAUTE VITESSE

La conception d'une carte électronique peut s'avérer très complexe lorsque les contraintes de conceptions s'accumulent. Ces contraintes reliées aux spécifications du projet amènent différentes problématiques. Ce chapitre se déroule comme une suite d'étapes à respecter pour bien concevoir une carte électronique mixte (analogique-numérique).

3.1 Problématique et définition des besoins

Le lien entre la partie numérique et analogique est critique dans ce projet. Le calcul de la position à partir des satellites GPS se fait avec l'aide de l'effet Doppler. En gros, cet effet Doppler décale la fréquence du signal original tout au long de son trajet jusqu'au récepteur. En sachant le temps que le signal a mis pour atteindre le récepteur il est possible de connaître la distance du satellite. Dans notre cas, la problématique est de sortir un signal le plus précis possible pour éviter de biaiser les données calculées par le simulateur. La conception du convertisseur numérique à analogique est donc cruciale puisqu'il faut tenir compte de tous les paramètres affectant le signal de sortie tel que le bruit de phase, bruit thermique, interférence électromagnétique, biais d'horloge, etc.

Comme mentionné précédemment, nous avons décidé de concevoir une nouvelle carte fille pour remplacer l'originale. La carte originale provenant de la compagnie Lyrtech a une mauvaise distribution de l'horloge. Cette mauvaise distribution de l'horloge ajoute du bruit au signal de sortie des convertisseurs. Les performances et la qualité du signal de sortie sont donc atténuées.

Nous avons aussi un autre problème avec la carte de Lyrtech. Pour obtenir un signal à 70 MHz avec une horloge à 100 MHz pour entrer dans la carte RF, nous devons sortir notre signal GPS à 30 MHz pour utiliser la deuxième harmonique comme entrée de la RF. Cette

façon de faire nous faisait perdre de la puissance, ce qui affectait la plage dynamique à l'entrée des « mixers » de la carte RF. Les autres harmoniques générées pouvaient aussi générer du bruit supplémentaire dans le circuit.

Avec les nouveaux convertisseurs, ces problèmes disparaissent puisque nous avons la possibilité de faire de l'interpolation et de la modulation interne dans le DAC. Avec une horloge à 100 MHz, nous pouvons donc avoir un signal de sortie à 70 MHz sans la moindre harmonique grâce à la réjection d'image interne du convertisseur.

Les 16 bits du nouveau DAC procurent aussi plus de précision que les 14 bits originaux.

Il est possible de voir au chapitre 4 différents résultats comparant les deux cartes DACs. Mais avant de passer aux résultats comparatifs, procédons plutôt aux étapes de conception de la nouvelle carte DAC.

3.2 Recherche de pièces correspondant aux besoins

Bien choisir ses pièces est crucial dans une conception. Puisqu'il est nécessaire d'avoir trois sorties analogiques, il nous faut donc trois convertisseurs numériques à analogiques (« DACs »). Pour assurer une bonne synchronisation des sorties des DACs, l'utilisation d'un distributeur d'horloge s'avère préférable. Ce distributeur d'horloge permettra aux horloges d'arriver toutes en même temps aux convertisseurs, ce qui assurera la synchronisation entre les sorties des DACs. La Figure 3.1 montre un schéma des différents composants du projet. Il est à noter que le « Board RF » représente la carte analogique vue précédemment et que le « Board DAC » représente la nouvelle carte mezzanine discutée dans ce chapitre.

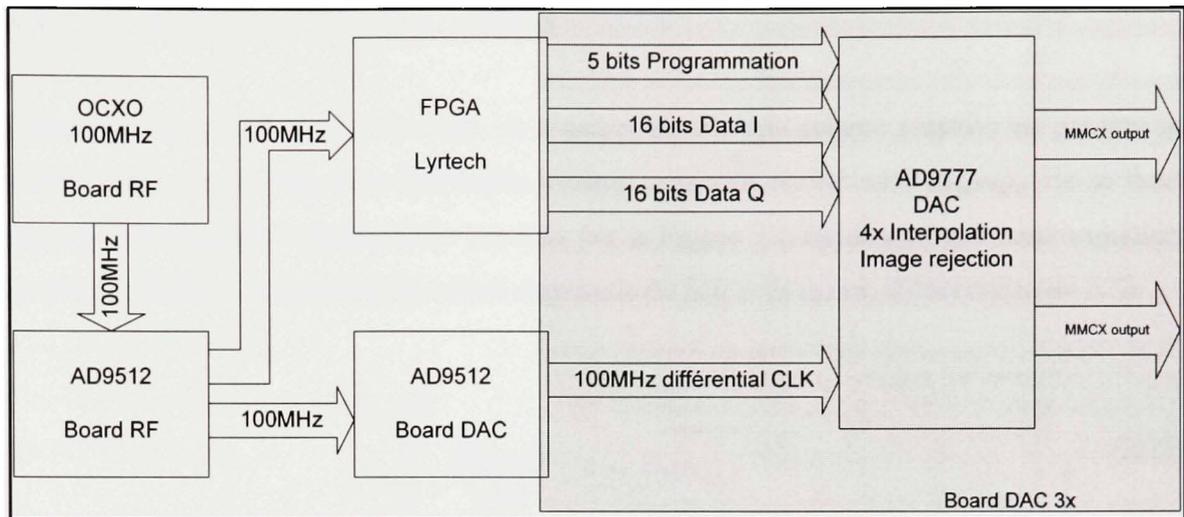


Figure 3.1 Représentation des relations entre les composants du projet.

On remarque sur la figure précédente que le distributeur d'horloge (AD9512) de la carte RF fournit l'horloge principale au FPGA de la carte de développement de même qu'à un autre distributeur d'horloge situé sur la nouvelle carte DAC avec trois convertisseurs numériques-analogiques (AD9777). Ces deux pièces (AD9512-AD9777) proviennent de la compagnie Analog Devices.

3.3 Analyse des besoins pour l'alimentation (consommation de courant)

Il est primordial d'étudier correctement les besoins en alimentation d'un futur circuit pour éviter des bris ou une surchauffe. Une consommation de courant aux limites d'un régulateur risquerait de causer un réchauffement inadéquat du système de même que des risques de bris. On doit donc avoir une idée claire des consommations de courant prévues pour mieux prévoir les pièces requises pour réguler la tension. Il est aussi important de faire cette étude puisque la nouvelle carte DAC se retrouve à remplacer l'ancienne carte de Lyrtech sur le même connecteur. Si la consommation de notre carte dépasse largement celle de Lyrtech, il est possible que la source d'alimentation soit insuffisante pour notre nouvelle carte. Nous devons donc calculer la consommation du distributeur d'horloge et des trois convertisseurs.

3.3.1 Distributeur d'horloge AD9512

Concernant le distributeur d'horloge, on s'assure de n'avoir aucune surprise en prenant la valeur maximale de puissance demandée lorsque toutes les sorties sont actives. En se fiant aux spécifications du fabricant montrés par la Figure 3.2, on obtient une consommation de près de 260 mA en utilisant la valeur maximale de 850 mW et une alimentation de 3.3V.

$$P = VI$$

$$I = \frac{P}{V} = \frac{850mW}{3.3V} = 257.5757mA \quad (3.1)$$

AD9512					
POWER					
Table 9.					
Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
POWER-UP DEFAULT MODE POWER DISSIPATION		550	600	mW	Power-up default state, does not include power dissipated in output load resistors. No clock.
POWER DISSIPATION			800	mW	All outputs on. Three LVPECL outputs @ 800 MHz, two CMOS out @ 62 MHz (5 pF load). Does not include power dissipated in external resistors.
Full Sleep Power-Down			850	mW	All outputs on. Three LVPECL outputs @ 800 MHz, two CMOS out @ 125 MHz (5 pF load). Does not include power dissipated in external resistors.
Power-Down (PDB)		35	60	mW	Maximum sleep is entered by setting 0Ah<10> = 01b and 58h<4> = 1b. This powers off all band gap references. Does not include power dissipated in terminations.
Power-Down (PDB)		60	80	mW	Set FUNCTION pin for PDB operation by setting 58h<6:5> = 11b. Pull PDB low. Does not include power dissipated in terminations.
POWER DELTA					
CLK1, CLK2 Power-Down	10	15	25	mW	
Divider, DIV 2 – 32 to Bypass	23	27	33	mW	For each divider.
LVPECL Output Power-Down (PD2, PD3)	50	65	75	mW	For each output. Does not include dissipation in termination (PD2 only).
LVDS Output Power-Down	80	92	110	mW	For each output.
CMOS Output Power-Down (Static)	56	70	85	mW	For each output. Static (no clock).
CMOS Output Power-Down (Dynamic)	115	150	190	mW	For each CMOS output, single-ended. Clocking at 62 MHz with 5 pF load.
CMOS Output Power-Down (Dynamic)	125	165	210	mW	For each CMOS output, single-ended. Clocking at 125 MHz with 5 pF load.
Delay Block Bypass	20	24	60	mW	Vs. delay block operation at 1 ns/fs with maximum delay; output clocking at 25 MHz.

Figure 3.2 Tableau des puissances consommées de l'AD9512.

Tiré du site web d'Analog Devices [2]

3.3.2 DAC AD9777

Pour le convertisseur numérique-analogique, la composante utilisée par Lyrtech sur l'ancienne carte était un DAC de Analog Devices (AD9767). Notre choix à nous pour le remplacer s'est fait en considérant l'utilité de l'interpolation et de la réjection d'image pour notre projet. Notre choix s'est donc tourné vers le AD9777 de la même compagnie qui offre une interpolation interne de 4x de même qu'une modulation interne avec réjection d'image. Ces spécifications vont nous permettre d'obtenir un signal de sortie vers la carte RF de 70 MHz directement.

Concernant la consommation, le manufacturier (Analog Devices) fournit des courbes définies selon les différentes configurations. Dans notre cas, on utilise la configuration suivante :

- $f_{DATA} = 100 \text{ MHz}$
- Interpolation à 4x
- Modulation ON
- PLL Enabled

En se fiant aux spécifications du manufacturier représentées par la Figure 3.3, la Figure 3.4 et la Figure 3.5, on obtient les consommations suivantes :

- $I_{DVDD} = 380 \text{ mA}$
- $I_{AVDD} = 75.75 \text{ mA}$
- $I_{CLKVDD} = 30 \text{ mA}$

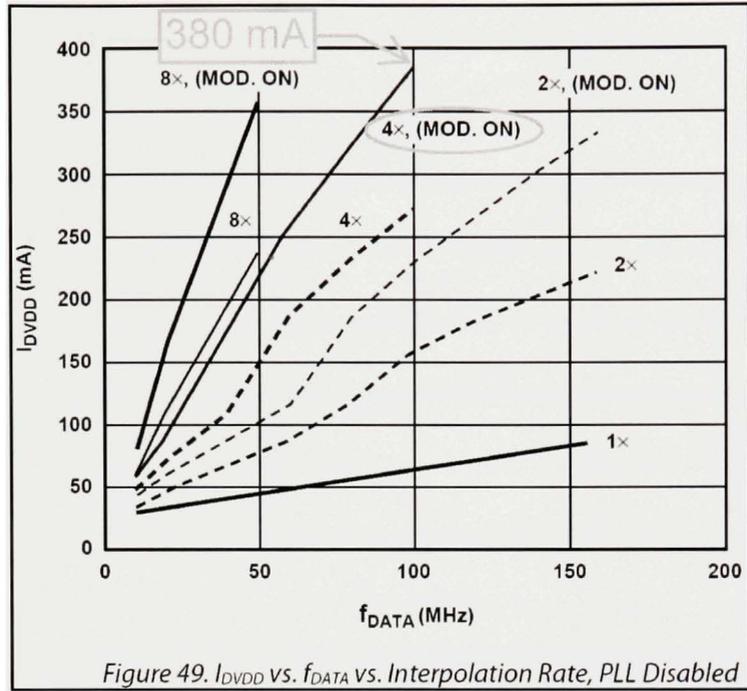


Figure 3.3 AD9777 Consommation I_{DVDD} .

Tiré du site web d'Analog [1]

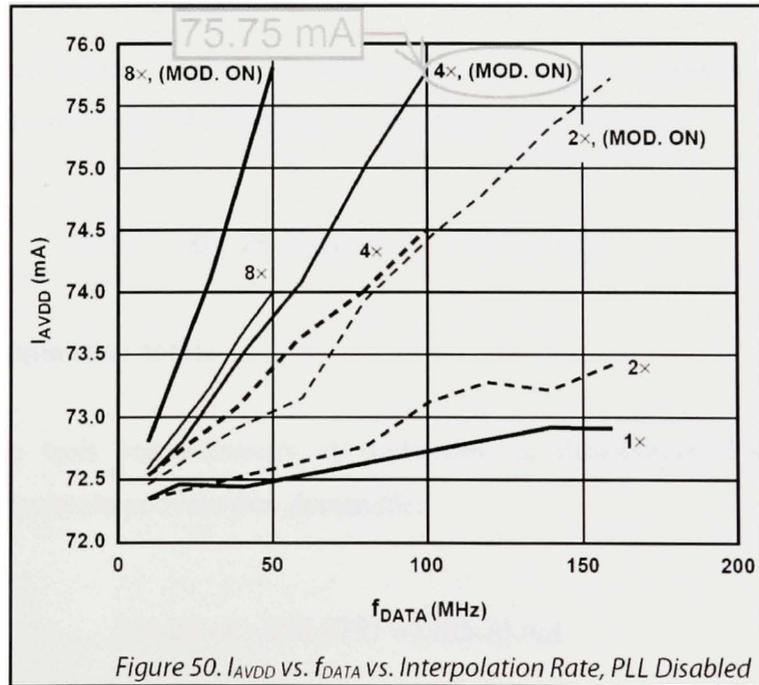


Figure 3.4 AD9777 Consommation I_{AVDD} .

Tiré du site web d'Analog [1]

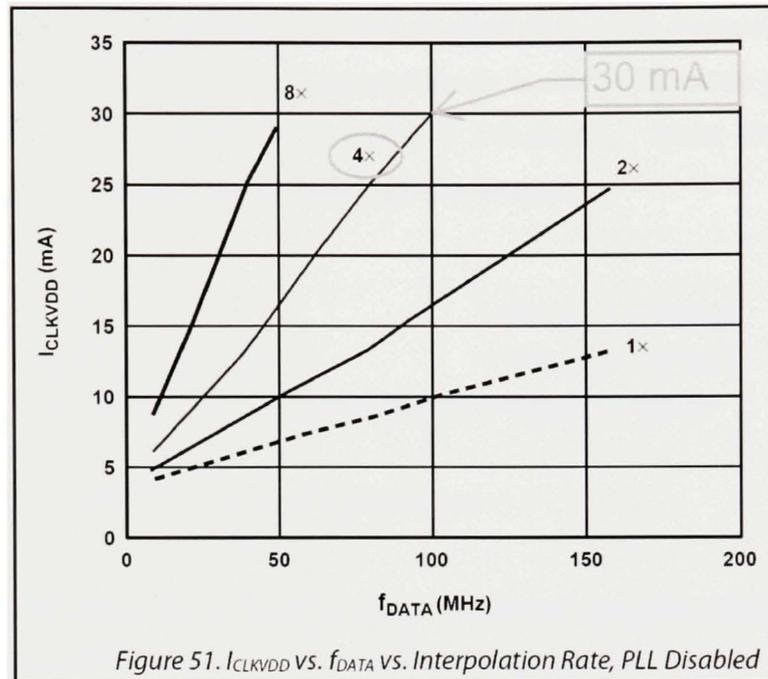


Figure 3.5 AD9777 Consommation I_{CLKVDD} .

Tiré du site web d'Analog [1]

Ces graphiques représentent les consommations sans PLL, selon le tableau des spécifications du AD9777 tiré du site web d'Analog [1], on rajoute 23.5mA à I_{CLKVDD} .

$$\text{Total} = 380 + 75.75 + 30 + 23.5 = 509.25 \text{ mA} \quad (3.2)$$

3.3.3 Consommation totale

Puisqu'on utilise trois convertisseurs et seulement un distributeur d'horloge, voici la consommation maximale pouvant être demandée.

$$509.25 \times 3 + 257.5757 = 1785.33 \text{ mA} \quad (3.3)$$

Il faut donc s'assurer que le connecteur d'alimentation et l'alimentation elle-même peuvent supporter et fournir 2 ampères.

Puisque la nouvelle carte DAC sera installée à la place de la carte mezzanine de Lyrtech, il est important d'avoir une idée de la consommation de la carte de Lyrtech pour évaluer si la nouvelle ne consommera pas trop comparée à l'ancienne. Voici l'évaluation de la consommation de cette ancienne carte appelée à être remplacée.

$$\begin{aligned}
 \text{DACs AD9767} &= 90mA \times 4 = 360mA \\
 \text{5V -5V Ampli Op AD9632} &= 17mA \times 8 = 136mA \\
 \text{5V Potentiomètre numérique DS1267A1} &= 4mA \times 4 = 16mA
 \end{aligned} \tag{3.4}$$

$$\text{Total} = 360mA + 136mA + 16mA = 512mA$$

On remarque clairement que la nouvelle carte consomme beaucoup plus que l'ancienne. Mais puisque le bloc d'alimentation de la plateforme peut fournir 5 ampères sur le 5V, aucun problème en vue de ce côté. Même si aucun problème ne peut venir de la consommation de courant, nous avons décidé de mettre un connecteur pour utiliser une alimentation externe au cas où il y aurait effectivement des problèmes.

Puisque nous n'avons aucune certitude concernant la source d'alimentation, nous devons filtrer correctement l'alimentation pour ne pas que le bruit provenant de cet alimentation affecte le reste du circuit.

3.4 Plages de fréquences à filtrer pour l'alimentation

Il est important de bien filtrer à la fréquence de l'horloge principale puisque c'est cette fréquence qui possède l'intensité la plus élevée dans le circuit. Les fréquences GPS et Galileo sont seulement générés à l'intérieur du FPGA et risquent donc beaucoup moins d'affecter la carte avec les convertisseurs. Un mauvais filtrage peut affecter le niveau de bruit et par le fait même la plage dynamique des convertisseurs.

3.5 Distribution des condensateurs de découplage

La distribution des condensateurs de découplage peut se faire avec une simple règle du pouce. Il s'agit tout simplement de couper les hautes et les basses fréquences pour chaque pin d'alimentation. Pour ce faire, il faut utiliser un condensateur créant presque un court-circuit aux fréquences désirées puisque sa résistance série (« ESR » Electrical Series Resistance) sera près de zéro. Pour s'assurer de bien couper les basses fréquences, il est impératif d'utiliser un gros condensateur au tantale comme un 10 μF , qui conserve une résistance série minimale même aux basses fréquences. Pour les hautes fréquences, en général on cible l'horloge principale, c'est-à-dire 100 MHz pour la carte DAC. Les condensateurs en céramique offrent habituellement un gouffre à certaines fréquences précises. Dans notre cas si on se fie au logiciel KEMET Spice de la compagnie KEMET, un condensateur de 3.9 nF a une « ESR » basse pour 100 MHz. L'idéal serait un condensateur de type C0G puisque la pointe peut descendre sous la barre des 20 mOhms. Mais par contre ce type coûte en général huit à dix fois plus cher que le type X7R, qui lui se tient sous les 230 mOhms. Puisque la différence n'est pas énorme, l'utilisation du X7R sera retenue dans le cas présent pour diminuer les coûts.

La Figure 3.6 nous montre l'impédance et l'ESR des condensateurs de 10 μF et de 3.9 nF. On remarque la pointe du 3.9 nF vers les 100 MHz. Ce qui va permettre une bonne filtration à cette fréquence tandis que la courbe évasée du condensateur de 10 μF filtrera les fréquences plus basses.

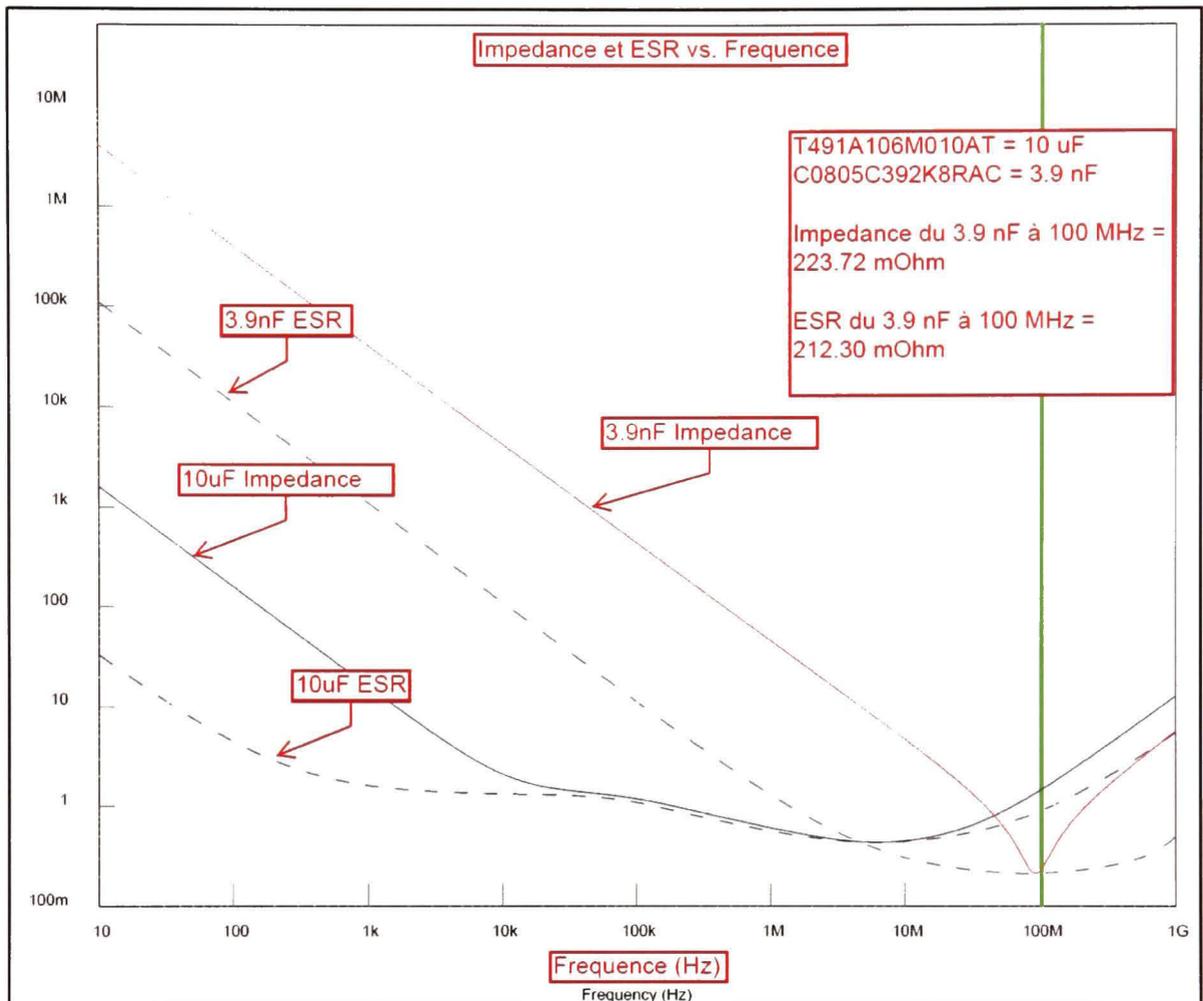


Figure 3.6 Impédance et ESR des condensateurs affichés avec KEMET Spice.

3.6 Conception des filtres d'alimentation

Les filtres d'alimentation sont importants dans un circuit, tout particulièrement dans les circuits analogiques comme celui conçu ici.

Habituellement, les circuits intégrés possèdent du filtrage interne sur l'alimentation appelé plus communément PSRR (« power supply rejection ratio » ou « power supply ripple rejection »). Voici les PSRRs du AD9777 et de régulateur de tension utilisé TPS76833 de Texas Instruments.

- AD9777

Pour le AD9777, son PSRR sur le AVDD (alimentation de la partie analogique du circuit intégré) est indiqué comme étant 0,4% du Full-Scale Range/V. Par contre, aucune donnée ne permet de savoir jusqu'à quelle fréquence le PSRR est valide.

$$\begin{aligned} \text{Full-Scale Range} = \text{FSR} &= [-1\text{V } 1\text{V}] = 2\text{V} \\ 20 \log\left(\frac{0.4\%}{2}\right) &= -54\text{dB} \end{aligned} \quad (3.5)$$

- TPS76833

Concernant le régulateur, on peut voir à la Figure 3.7 (tirée des spécifications du fabricant) l'étendue de sa couverture en fréquence.

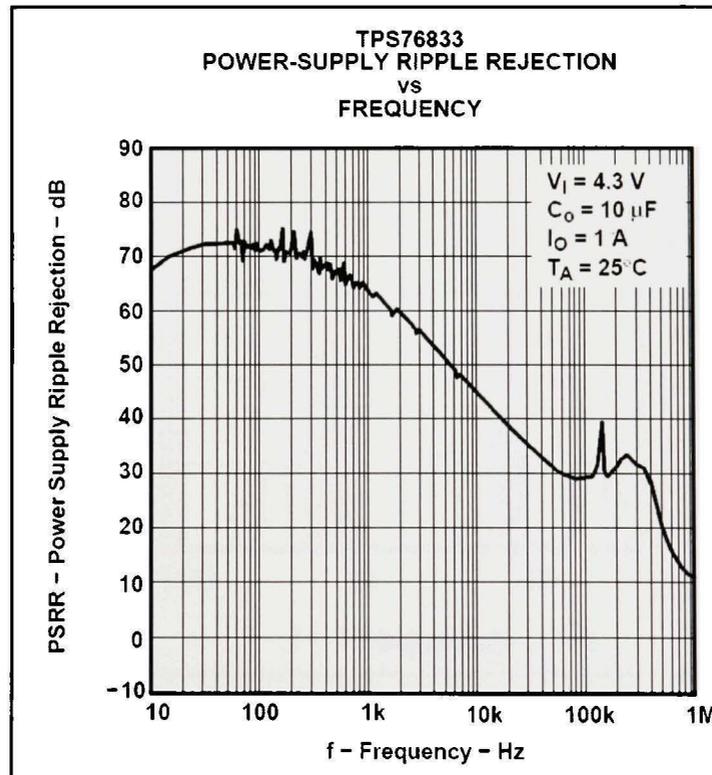


Figure 3.7 Power-Supply Ripple Rejection.

Tiré du site web de Texas Instruments [10]

Il est impératif de filtrer à la source pour supporter ces filtrages internes. C'est-à-dire directement à l'entrée du régulateur de tension. Un filtre de la forme de la lettre grec Pi (plus communément appelé « filtre en Pi » ou « Pi filter ») est un circuit polyvalent et donne d'excellents résultats comme filtre passe-bas. Il sera discuté plus en détail à la sous-section suivante. Le filtrage résultant des filtres intégrés et du filtre en PI est dessiné sommairement sur une réutilisation de la Figure 3.7 à la Figure 3.8.

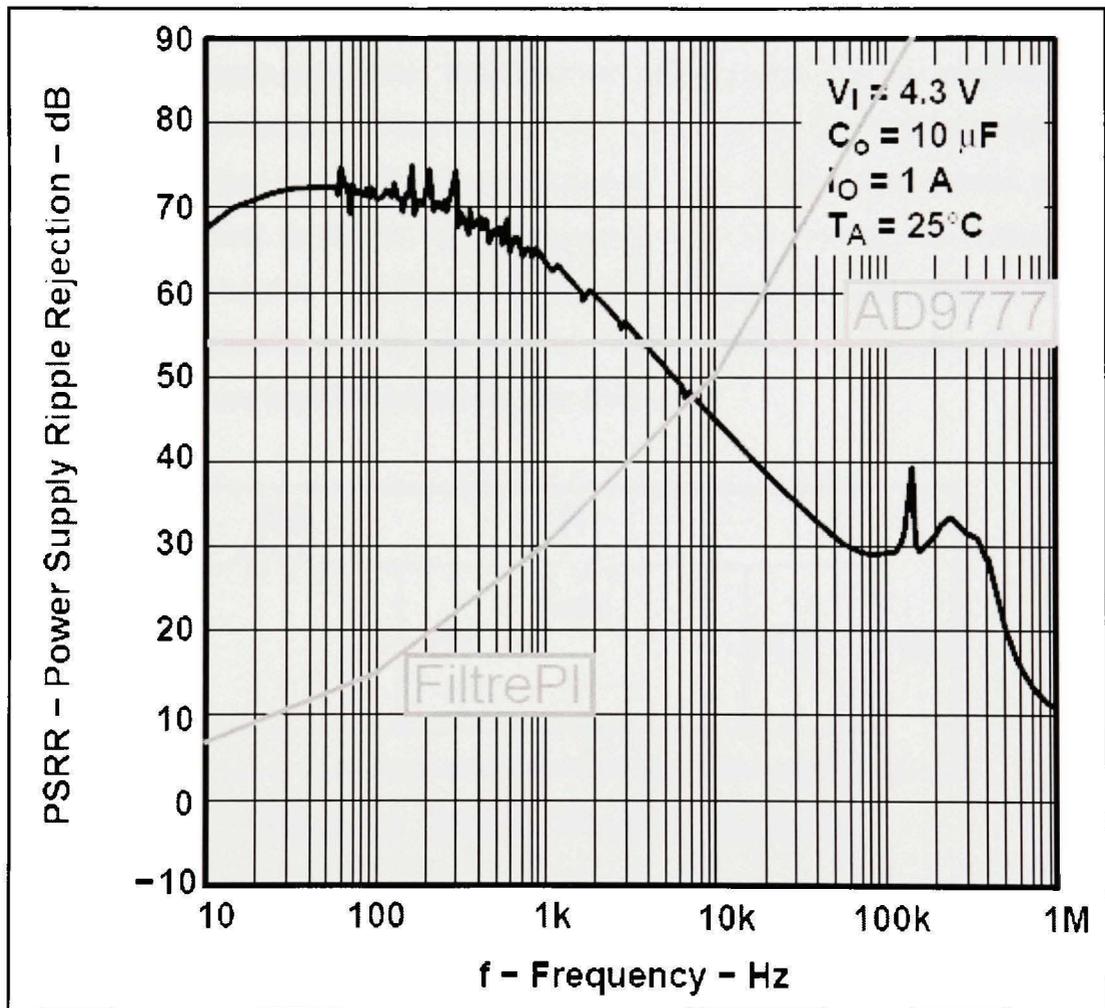


Figure 3.8 Représentation globale du filtrage d'alimentation.

Image originale tirée du site web de Texas Instruments [10]

On remarque que tous ces filtres sont complémentaires. Il est à noter que pour le AD9777, la plage de fréquence fut supposée constante jusqu'à 1 MHz. Il est fort possible que cette hypothèse diffère un peu de la réalité, mais cela permet d'avoir une idée globale.

3.7 Filtre en Pi (π)

Le filtre en Pi permet d'ajouter une filtration complémentaire dans les plus hautes fréquences que le filtre intégré du régulateur de tension et des autres circuits intégrés. Les filtres en PI possèdent généralement une bobine. Dans notre cas précis, au lieu d'utiliser une bobine pour la filtration il est préférable d'utiliser une « ferrite ». Les ferrites ont la particularité d'avoir une impédance variant en fonction de la fréquence. Elles offrent donc comme avantage d'être comme un bout de fil aux basses fréquences et d'avoir une grande résistance aux hautes fréquences choisies. À la Figure 3.9, la ferrite est représentée par l'élément R. Dans notre cas, il est préférable de s'assurer de filtrer le plus possible la fréquence d'horloge de 100 MHz en ayant une impédance élevée à cette fréquence.

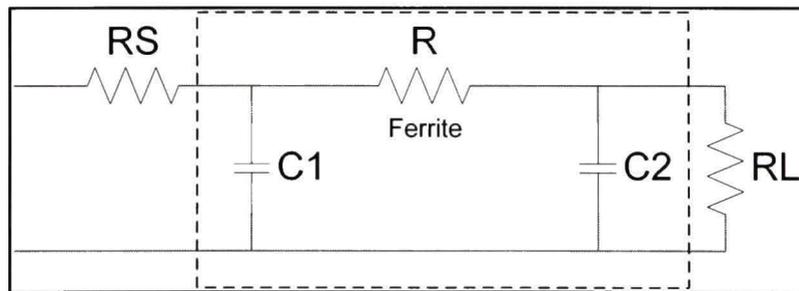


Figure 3.9 Filtre en Pi avant le régulateur.

Pour être en mesure d'évaluer théoriquement la réponse en fréquence de ce filtre, il est utile de transformer le circuit en quadripôle en admittance tel que vu à la Figure 3.10.

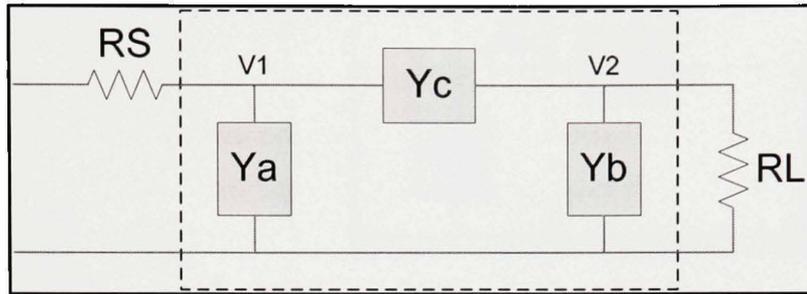


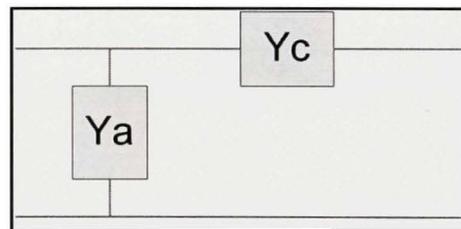
Figure 3.10 Transformation du filtre en Pi en réseau d'admittance pour le calcul.

Nous sommes donc en mesure d'identifier les équations utiles au calcul de la réponse en fréquence.

- Équation des différents paramètres d'admittances :

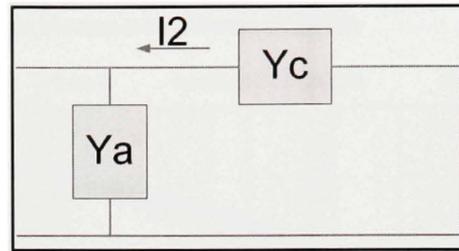
$$\begin{aligned}
 I_1 &= y_{11}V_1 + y_{12}V_2 \\
 I_2 &= y_{21}V_1 + y_{22}V_2 \\
 y_A &= C_1s \\
 y_B &= C_2s \\
 y_C &= \frac{1}{R}
 \end{aligned}
 \tag{3.6}$$

$$y_{11} = \left. \frac{I_1}{V_1} \right|_{V_2=0} = y_A + y_C$$



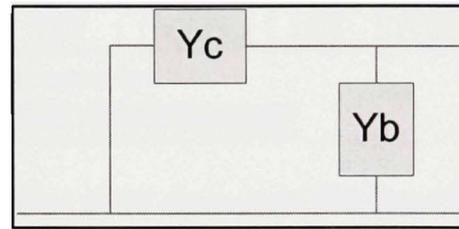
(3.7)

$$y_{21} = y_{12} = \frac{I_2}{V_1} \Big|_{V_2=0} = -y_c$$



(3.8)

$$y_{22} = \frac{I_2}{V_2} \Big|_{V_1=0} = y_b + y_c$$



(3.9)

- Équation représentant la réponse en fréquence du filtre en Pi :

$$\frac{V_L(s)}{V_i(s)} = \frac{-y_{21}}{y_{22} + \frac{1}{R_L} + R_s \left(-(y_{12}y_{21} - y_{11}y_{22}) + y_{11} \frac{1}{R_L} \right)} = \frac{\frac{20000000}{R}}{s^2 + s \left(2200 + \frac{110000}{R} \right) + 400000 + \frac{40000000}{R}} \quad (3.10)$$

La Figure 3.11 donne le résultat de l'équation précédente avec des ferrites de 11 ohms et de 600 ohms.

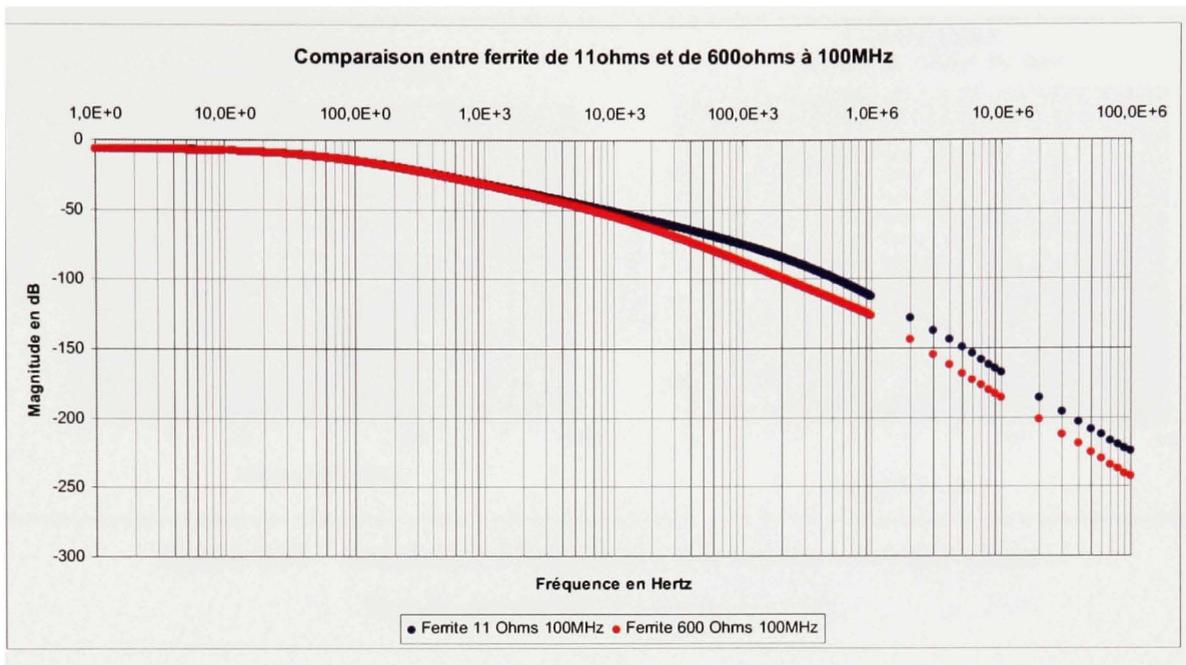


Figure 3.11 Comparaison entre ferrites de 11 ohms et de 600 ohms à 100 MHz.

On remarque que l'utilisation de la ferrite de 600 ohms offre une meilleure atténuation aux hautes fréquences, mais qu'aux basses fréquences l'atténuation est similaire entre les deux ferrites. Ces valeurs d'impédances correspondent aux valeurs idéales des ferrites avec une consommation de 0 ampère. Selon nos spécifications, avec une consommation de plus de 500 mA (prenons 750 mA par sécurité), les impédances seraient de 8,5 ohms au lieu de 11 et de 70 ohms au lieu de 600. Puisque la marge est restreinte dans la différence d'atténuation entre les deux, il faut donc trouver une autre caractéristique qui nous indiquerait laquelle serait mieux dans notre cas. Il faut savoir que les courbes des impédances des ferrites suivent la forme d'une cloche.

On peut voir à la Figure 3.12 tirée des spécifications du manufacturier (Steward) que la courbe de la ferrite de 11 ohms est plus évasée que celle de la ferrite de 600 ohms. Puisque la ferrite de 11 ohms atteint son sommet vers les 20 ohms à 1 GHz et que la ferrite de 600 ohms atteint 600 ohms vers 450 MHz, il est préférable d'utiliser la ferrite de 600 ohms qui filtrera de manière plus efficace dans les fréquences autour du 100 MHz.

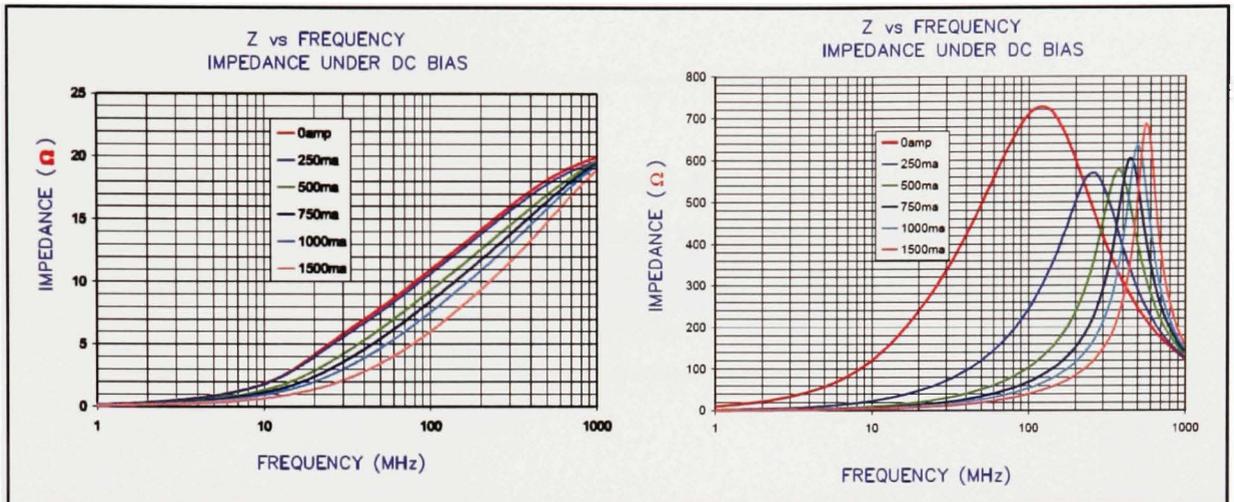


Figure 3.12 Graphique d'impédance des ferrites 11 et 600 ohms.

Tiré du site web de Steward [8] et [9]

Pour le filtre de l'alimentation, il faut aussi s'assurer de rajouter quelques composantes pour couper les hautes fréquences de la même façon vue pour le découplage des alimentations des circuits intégrés. Pour la carte DAC, un condensateur de 150 nF fut rajouté au filtre en Pi en parallèle avec C2 pour couper aux environs de 10MHz.

3.8 Adaptation d'impédance

L'adaptation d'impédance pour la distribution des horloges est importante. Cela permet d'obtenir les bons niveaux de tensions à destination avec le moins de bruit possible. Évidemment, moins de bruit signifie de meilleurs résultats.

Il est utile de se fier aux spécifications du distributeur d'horloge lui-même pour adapter correctement les sorties. Pour le distributeur d'horloge AD9512, on retrouve la Figure 3.13 dans les spécifications de Analog Devices. Cette figure montre le circuit recommandé pour les trois lignes LVPECL desservant les trois convertisseurs AD9777.

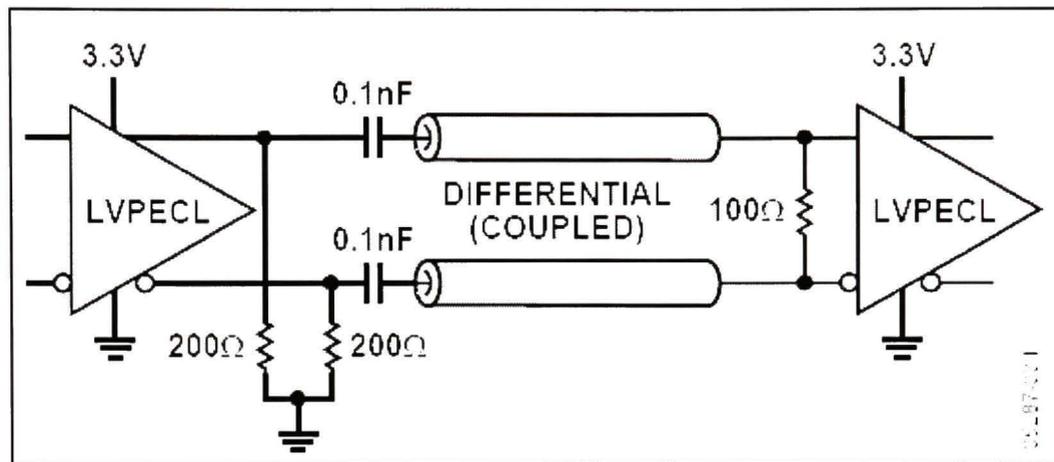


Figure 3.13 LVPECL dans une ligne de transmission parallèle.

Tiré du site web d'Analog [2]

Pour la conception de la carte électronique, des calculs sont nécessaires pour avoir une ligne de transmission (cylindre sur la figure) de la bonne impédance. Comme les calculs peuvent s'avérer complexes, un logiciel comme « LineCalc » de ADS (« Advanced Design System ») peut être utile. Des logiciels sont aussi disponibles gratuitement sur internet, mais on ne sait jamais la précision et la justesse des calculs effectués. La Figure 3.14 nous montre l'interface du logiciel LineCalc. Plusieurs paramètres doivent être entrés pour obtenir ce que l'on désire. Il est important d'indiquer les paramètres suivants correctement.

Paramètres de substrat :

Er Permittivité relative de la carte (FR4 donne habituellement 4.3)

H Épaisseur de la couche de substrat séparant le dessus du dessous

T Épaisseur de cuivre

Paramètres physiques

W Largeur de la trace de cuivre

S Espacement entre les deux traces de cuivres couplées

En appuyant sur le bouton « Analyze » après avoir bien configuré le modèle, nous obtenons l'impédance caractéristique des traces (Z_0). Habituellement il n'est pas possible de jouer sur les paramètres de substrat, il suffit donc de modifier la largeur et/ou l'espacement (W et S) pour obtenir l'impédance voulue.

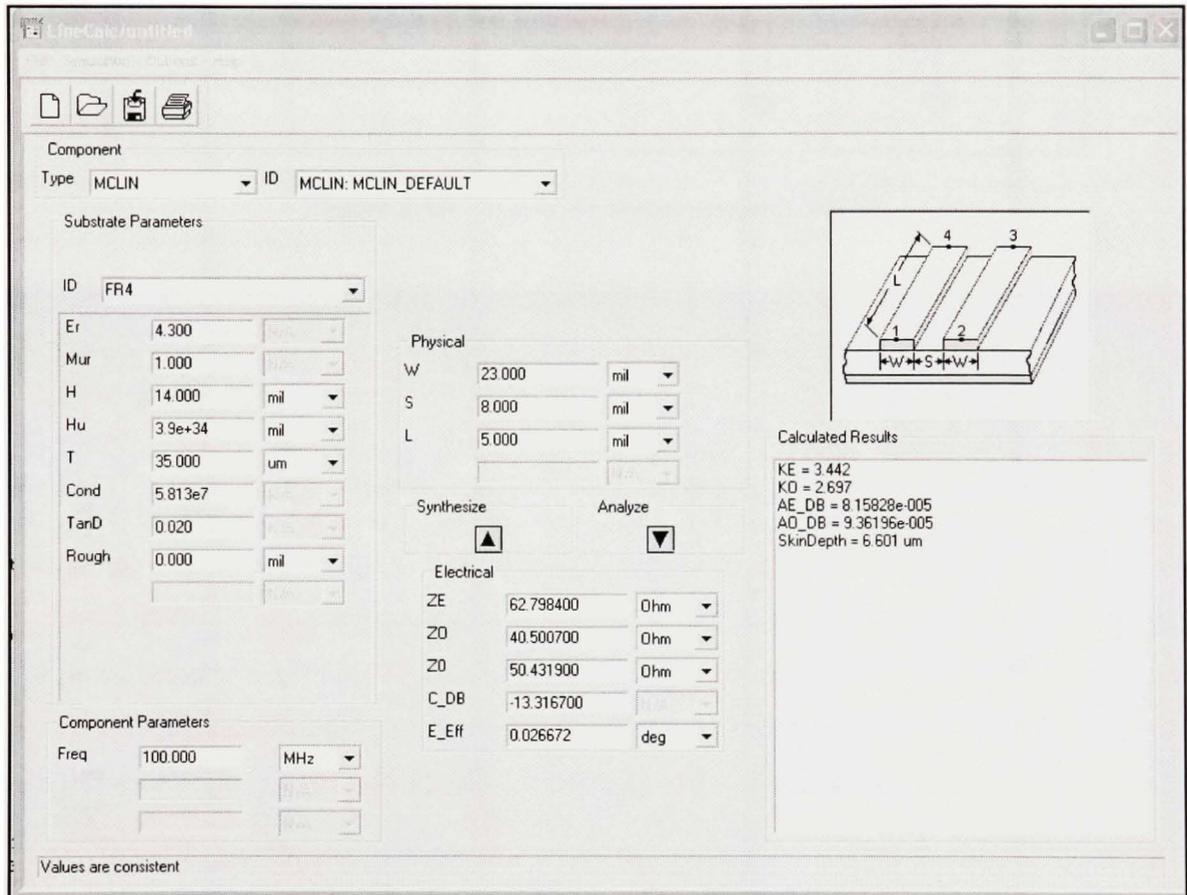


Figure 3.14 Ligne microstrip couplée 50 ohms de LineCalc.

Pour la sortie des signaux CMOS pour le FPGA et la sortie de test, le circuit recommandé est celui de la Figure 3.15 et l'analyse de l'impédance caractéristique se trouve à la Figure 3.16.

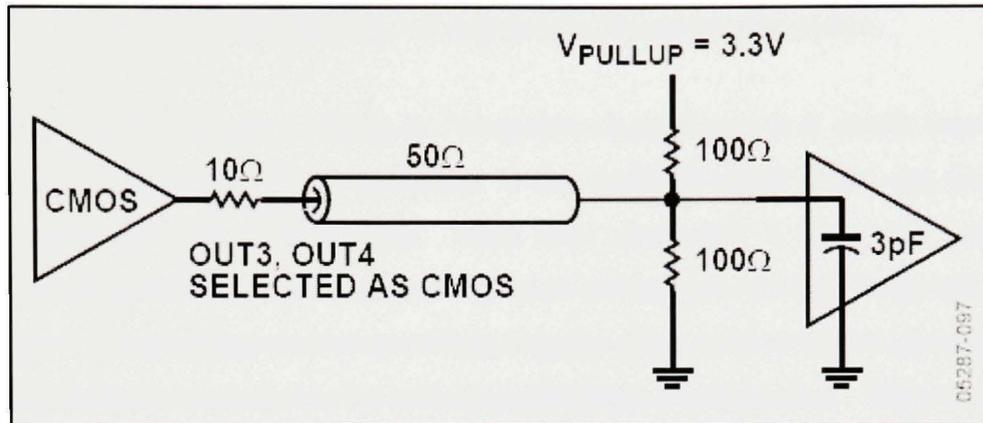


Figure 3.15 Ligne de transmission CMOS.

Figure 3.16 Ligne microstrip de 50 ohms de LineCalc.

3.9 Création des « footprints » et analyse du placement des pièces

La création des « footprints » (dessin de l'empreinte de la pièce sur le circuit imprimé) est une autre tâche importante de la conception. Il faut impérativement se fier aux dimensions recommandées dans les spécifications. Mais d'un autre côté, il faut aussi vérifier les contraintes de fabrication du fournisseur qui sera choisi pour la fabrication de la carte électronique. Ce qu'il faut surtout surveiller, c'est les dimensions des trous de perçage, car les fabricants n'ont pas tous les mêmes équipements de fabrication. Alors si on prend comme exemple un circuit où la spécification d'un composant indique qu'il faut des trous de 20 millièmes de pouces (« mil »), dépendamment du fabricant on aura au final des trous allant de 18 millièmes à aussi bas que 14 millièmes. Cette différence s'explique par le procédé de placage des trous qui diffère d'un fabricant à un autre. Il faut donc créer le « footprint » en indiquant un trou de quelques millièmes de plus si on veut avoir des trous avec un diamètre suffisant après le placage.

En général un trou après placage est d'environ 3 millièmes plus petit. Pour notre exemple on devrait donc demander des trous de 23 millièmes pour s'assurer d'avoir un trou au final de 20 millièmes. Mais comme mentionné précédemment, certains fabricants parlent d'une différence au final d'au maximum de 6 millièmes. Par sécurité, il faudrait donc demander un trou de 26 millièmes. Lorsque les dimensions sont de l'ordre de 100 mil, ce n'est pas un problème de rajouter 6 mil mais malheureusement, lorsqu'on a un connecteur haute densité et que l'on a besoin un trou de 16 mil, il peut être difficile de demander un trou de 22 mil lorsque les pins elles-mêmes ne sont espacées que de seulement 50 mil.

Concernant le placement des pièces, il est préférable de placer les pièces connectées ensemble le plus près possible l'une de l'autre pour éviter de longues traces de cuivres allant d'un bout à l'autre de la carte électronique. Pour éviter que la partie numérique pollue la partie analogique, il est aussi important de bien séparer les deux types de composantes pour que les retours de courant de l'un n'aient pas lieu dans la partie de l'autre.

3.10 Longueur des traces

La longueur des traces peut s'avérer problématique dans un système. Règle générale, les traces de cuivres sur un circuit imprimé ont un temps de propagation de 180 ps/pouce ou environ 70 ps/cm (Belzile, 2007). Il faut donc s'assurer que la longueur des traces de nos signaux répond aux spécifications des différentes pièces utilisées. Plus l'horloge synchronisant les données a une fréquence élevée, plus il est critique de s'assurer que les données demeurent synchronisées. Par exemple, lorsque les contraintes du circuit imprimé nécessitent d'utiliser une trace excessivement longue sur un bus de données, il peut s'avérer nécessaire de s'assurer que chaque trace a la même longueur en insérant des « ziggles » (zigzag).

Dans notre cas, les données allant aux convertisseurs sont à 100 MHz. Selon la spécification de l'AD9777 à la page 30, le « setup time » doit être de 0ns et le « hold time » de 2.5 ns. Le « setup time » est la durée durant laquelle la donnée doit être stable avant que le front de l'horloge arrive. Pour le « hold time », c'est la durée durant laquelle la donnée doit être stable après le front de l'horloge.

$$\begin{aligned} \frac{1}{100MHz} &= \text{Période} = 10ns \\ \frac{2.5ns}{70 \frac{ps}{cm}} &= 35.71cm \end{aligned} \quad (3.11)$$

La longueur maximale d'une trace doit être de 35 cm pour éviter un problème de synchronisation. Dans notre cas, il n'est pas nécessaire d'avoir les traces numériques du FPGA de la même longueur puisque les traces les plus longues ne dépassent même pas 15 cm.

3.11 Séparation de la masse (« ground »)

Il y a deux écoles de pensées qui s'affrontent à ce sujet. La première soutient qu'il faut avoir deux masses séparées pour l'analogique et le numérique et les relier à un même point près de l'alimentation. Mais cela implique la difficulté à concevoir une carte électronique de cette manière lorsqu'on est limité dans l'espace. Et dans le cas où un fort courant est demandé, passer tout le courant dans un même point peut finir par être fatal pour la carte électronique.

La seconde manière consiste à avoir une seule masse, mais de bien séparer physiquement les pièces analogiques du numériques sur le circuit imprimé. De cette façon beaucoup plus simple, les retours de courant de l'analogique ne pourront pas aller polluer les signaux de la partie numérique. Il faut bien entendu s'assurer de ne pas couper la masse sous les signaux analogiques pour éviter un grand détour pour le retour de ces mêmes signaux. Cette seconde approche sera utilisée pour la conception de notre carte puisqu'elle est plus simple d'implémentation et qu'à ce jour aucune étude n'a démontré la supériorité de l'une ou l'autre de ces méthodes.

3.12 Conseils à prendre en note pour un bon résultat

Lors d'une première conception, il y a souvent plusieurs détails qui nous sont inconnus. Lors des conceptions suivantes, c'est souvent notre mémoire qui nous fait défaut sur certains détails pouvant être important au final. Cette section est plus à caractère informatif que scientifique, mais permet un meilleur résultat esthétique du circuit imprimé. Pour votre information, voici donc quelques points à ne pas oublier ou à prendre en considération lors de la conception.

- Grosseur du texte (Top Overlay ou Silkscreen)

Désignateurs des pièces : 40 mils (hauteur) et 10 mils (épaisseur)

Texte d'indication (ex : 5V – GND – Input) : 60 mils (hauteur) et 10 mils (épaisseur)

Date de fabrication et nom du concepteur : 75 mils (hauteur) et 10 mils (épaisseur)

- S'assurer de prévoir des trous de fixation
- Prévoir des trous de 3 à 5 mil plus grands que ceux désirés pour prendre en compte le placage du procédé de fabrication.

Ce sont des choses simples, mais qui permettent d'avoir un meilleur rendu esthétique et pratique du circuit imprimé.

3.13 Conclusion

Ce chapitre se concentrait sur les explications des différentes étapes de conception d'une carte électronique mixte. Il ne faut pas oublier que le choix des composantes est très important puisque tout le reste de la conception dérive de ce choix. Il faut aussi tenter d'évaluer les utilisations futures de cette éventuelle carte pour bien choisir les pièces. Bien entendu, la conception d'un circuit imprimé ne se limite pas à ces informations. Des livres entiers se consacrent à la conception de circuits imprimés. Ce chapitre comprenait les étapes principales de conception permettant de mieux comprendre la carte conçue de même que leur effet sur les résultats finaux. Les différents schémas des connexions de cette carte sont présentées aux ANNEXES IV,V,VI et VII.

Le chapitre suivant montrera les résultats comparatifs entre cette nouvelle carte DAC et l'originale de Lyrtech. D'autres résultats seront aussi montrés concernant le simulateur GNSS.

CHAPITRE 4

RÉSULTATS

Ce dernier chapitre comparera les résultats obtenus avec la carte de Lyrtech et la nouvelle carte DAC (AD9777). Des résultats des solutions de navigations obtenues avec le simulateur GNSS complet seront aussi expliqués dans ce chapitre.

4.1 Comparaison entre les cartes DAC (Lyrtech - AD9777)

Les résultats présentés ici seront donc axés sur la comparaison entre la carte DAC de Lyrtech et la carte DAC basée sur le AD9777. Pour éviter de se mélanger, elles seront respectivement nommées « Carte de Lyrtech » et « Carte AD9777 ». Les résultats seront montrés premièrement à l'oscilloscope puis ensuite avec un analyseur de spectre. La méthodologie utilisée est tout simplement de faire rouler le même programme pour les deux cartes et d'ensuite de prendre des captures d'écrans des résultats obtenus sur un oscilloscope et un analyseur de spectre

4.1.1 Oscilloscope (Tektronix TDS694C)

Les figures suivantes montrent la sortie à l'oscilloscope d'un sinus sur les deux différentes cartes. Concernant la carte AD9777, deux configurations sont montrées. Il y a la configuration équivalente de Lyrtech avec une sortie sans modulation ni interpolation à 30 MHz et la sortie interpolée à 400 MHz et modulée à 70 MHz. Il faut noter que la modulation à l'interne se fait avec une fraction de la fréquence interpolée. Dans notre cas, l'interpolation se fait à 400 MHz avec une modulation à 50 MHz. Il faut donc fournir un sinus à 20 MHz plutôt que 30 MHz pour obtenir une sortie à 70 MHz ($20+50=70$ MHz)

L'information la plus utile à retenir des figures suivantes est seulement la forme de l'onde. On remarque sur la Figure 4.1 que le sinus à 30 MHz de Lyrtech a des sommets très arrondis comparé à son équivalent avec le AD9777 qui contient des plateaux (Figure 4.2). Ces

plateaux sont normaux puisque l'échantillonnage se fait à 100 MHz sur un sinus à 30 MHz. Nous avons donc seulement environ trois niveaux de tension par période du sinus pour le représenter. L'arrondissement des sommets de Lyrtech est probablement dû à l'utilisation d'un amplificateur opérationnel à la sortie du DAC AD9767 utilisé. Cet amplificateur apporte une limitation de la bande passante et provoque l'arrondissement du signal en coupant les hautes fréquences.

La Figure 4.3 nous montre la sortie du AD9777 avec interpolation et modulation. On remarque une sortie qui ressemble beaucoup plus à un sinus. Cette ressemblance est normale puisque avec l'interpolation à 400 MHz d'un sinus de 20 MHz, on dispose de 20 niveaux de tensions par période du sinus. Ce qui donne un résultat plus représentatif d'un sinus.

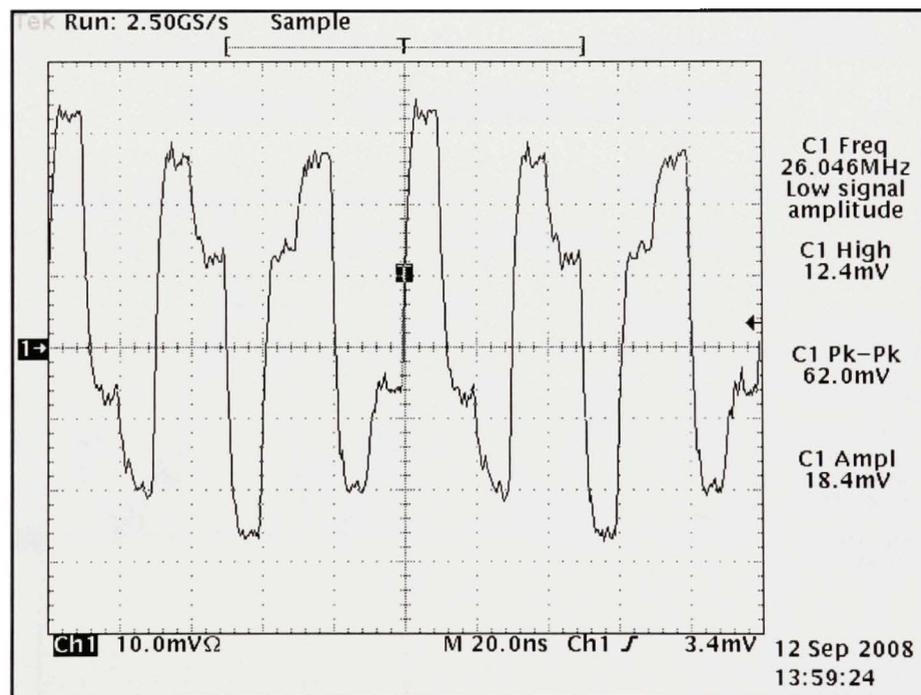


Figure 4.1 Sinus de 30 MHz de Lyrtech.

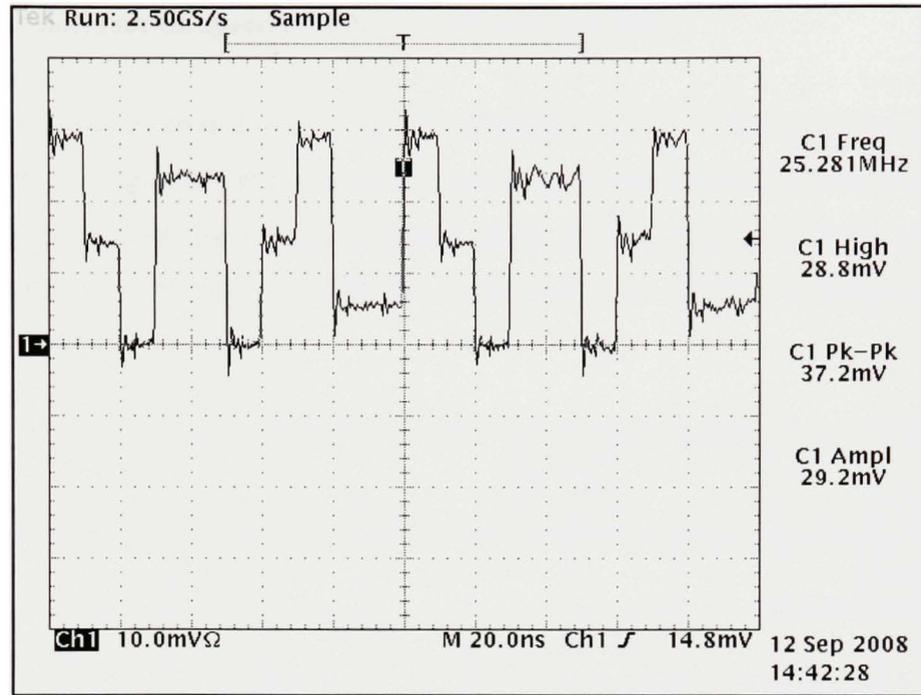


Figure 4.2 Sinus de 30 MHz du AD9777.

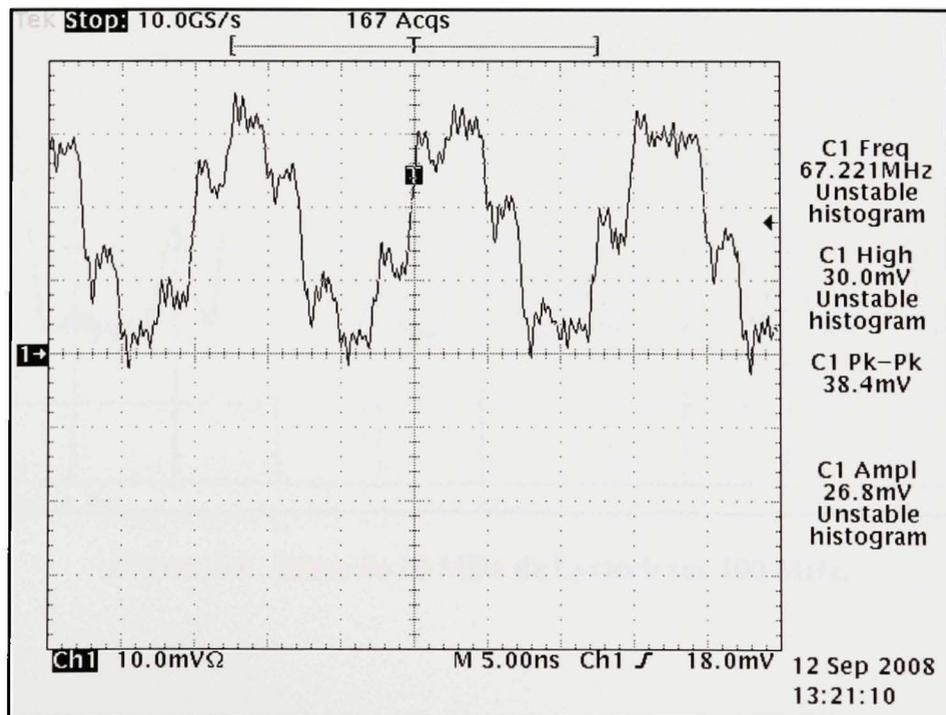


Figure 4.3 Sinus de 20 MHz avec modulation à 70 MHz du AD9777.

4.1.2 Analyseur de spectre (Rohde & Schwarz FSQ40)

Les figures suivantes nous dévoilent dans le domaine spectral la même chose que les figures précédentes. La Figure 4.4 et la Figure 4.5 représente le sinus de 30 MHz sur une plage de 0 à 100 MHz. On remarque que la sortie de Lyrtech est beaucoup plus bruitée que celle du AD9777 sans modulation ni interpolation, c'est-à-dire dans la même configuration que la carte de Lyrtech.

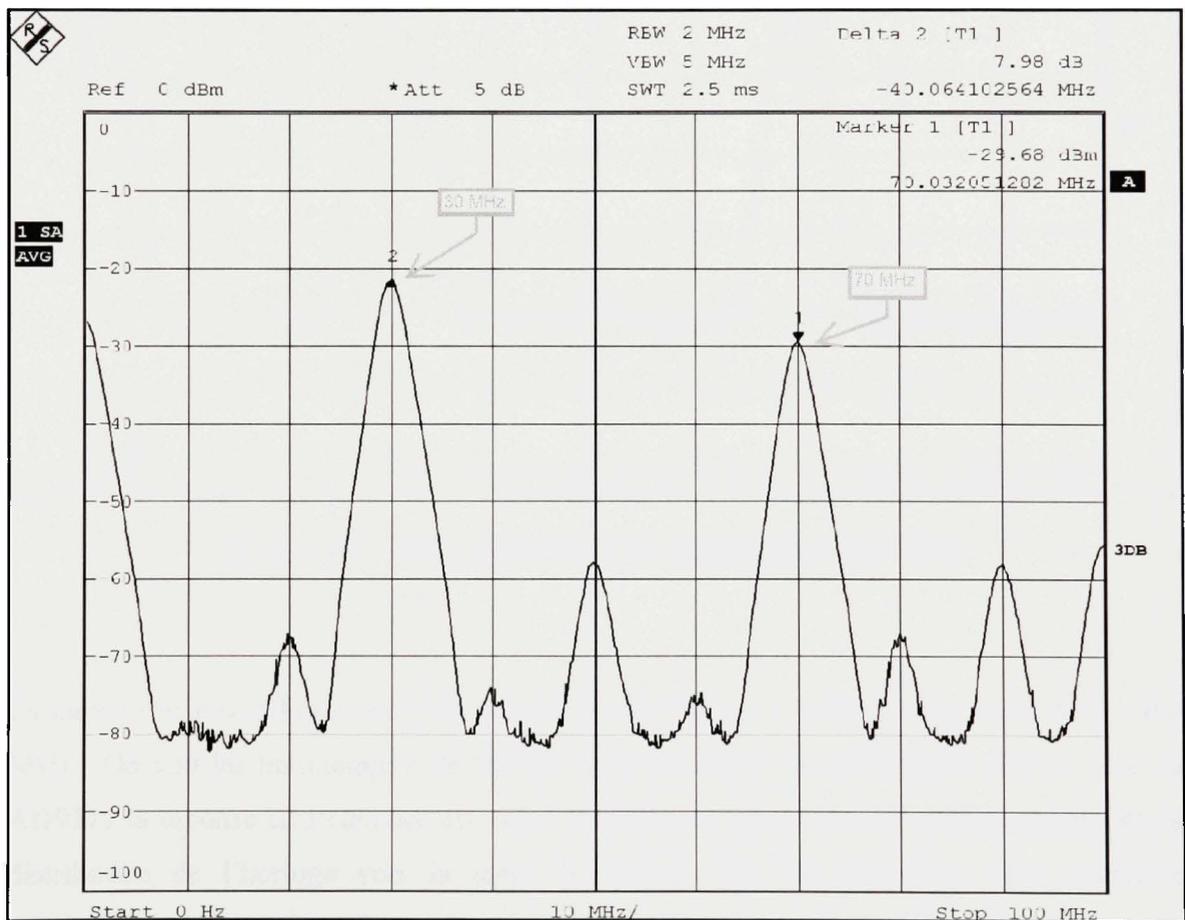


Figure 4.4 Sinus de 30 MHz de Lyrtech sur 100 MHz.

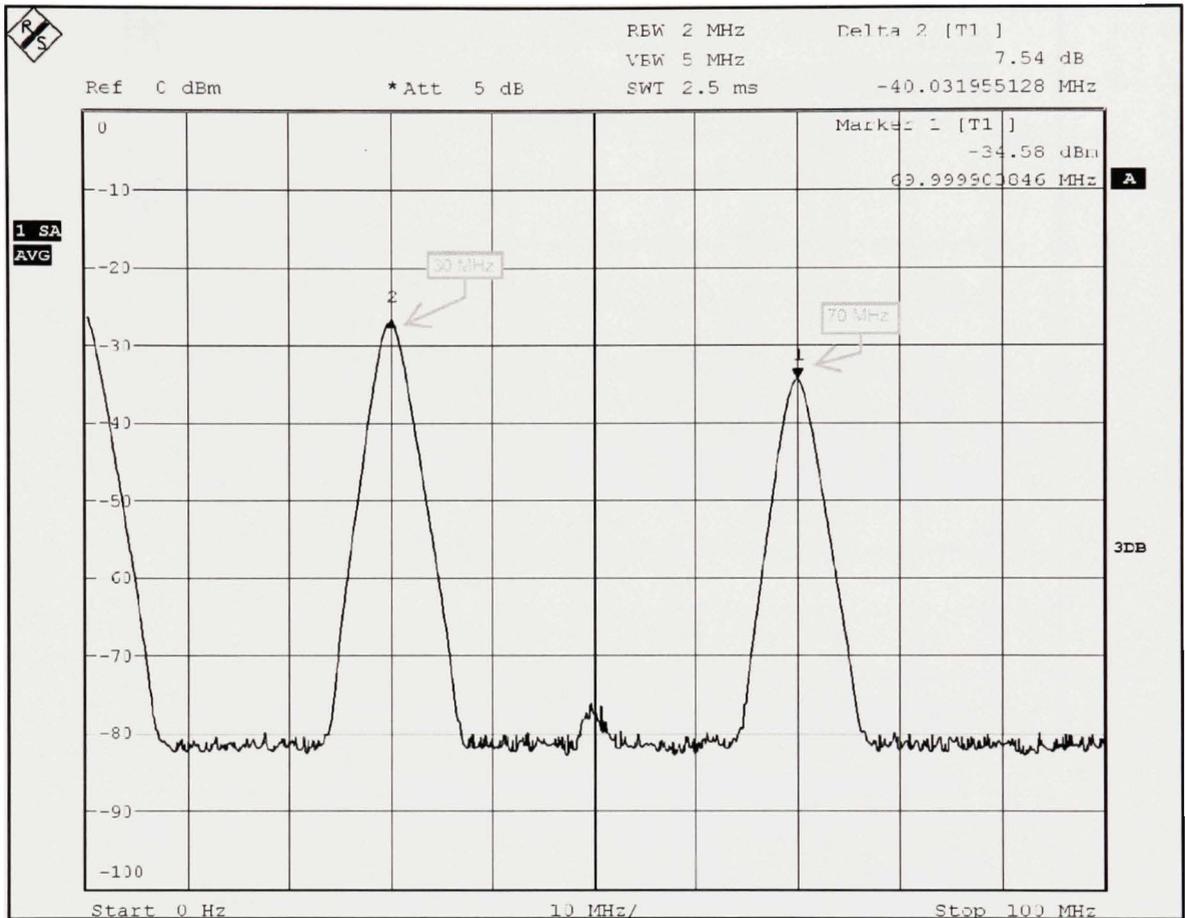


Figure 4.5 Sinus de 30 MHz du AD9777 sur 100 MHz.

La même chose se retrouve dans les Figure 4.6 et Figure 4.7 mais sur une plage de 0 à 400 MHz. On voit les harmoniques de Lyrtech avec beaucoup de bruits tandis que du côté du AD9777 la réponse en fréquence est plus « propre ». Comme mentionné précédemment, la distribution de l'horloge vers la carte de Lyrtech est très mauvaise, cette mauvaise distribution est probablement une des causes du bruit généré à la sortie des DACs de Lyrtech.

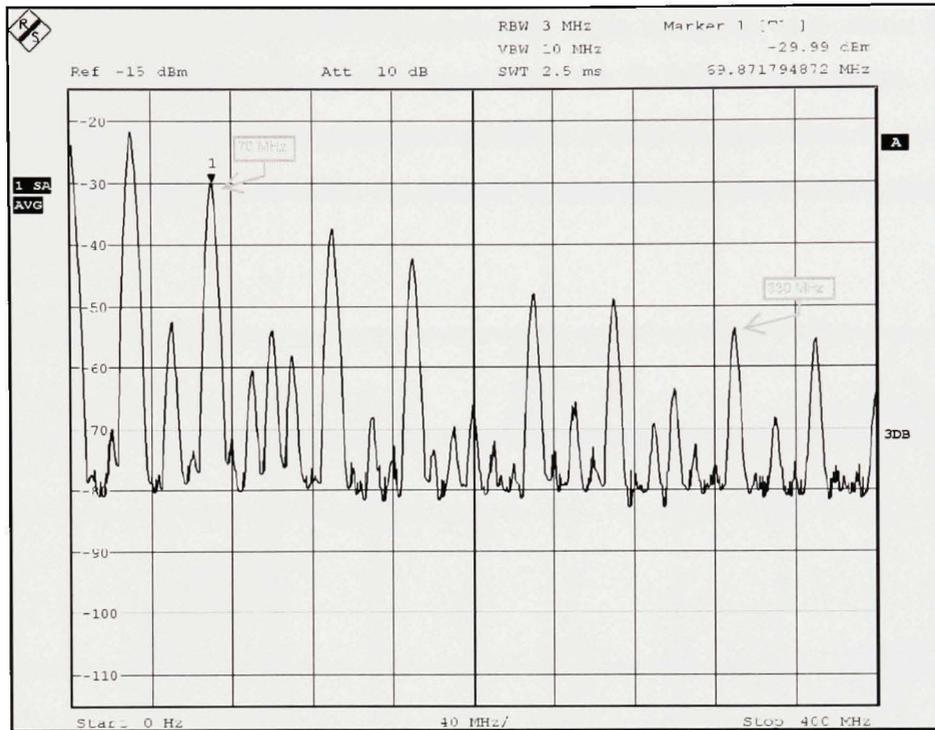


Figure 4.6 Sinus de 30 MHz de Lyrtech sur 400 MHz.

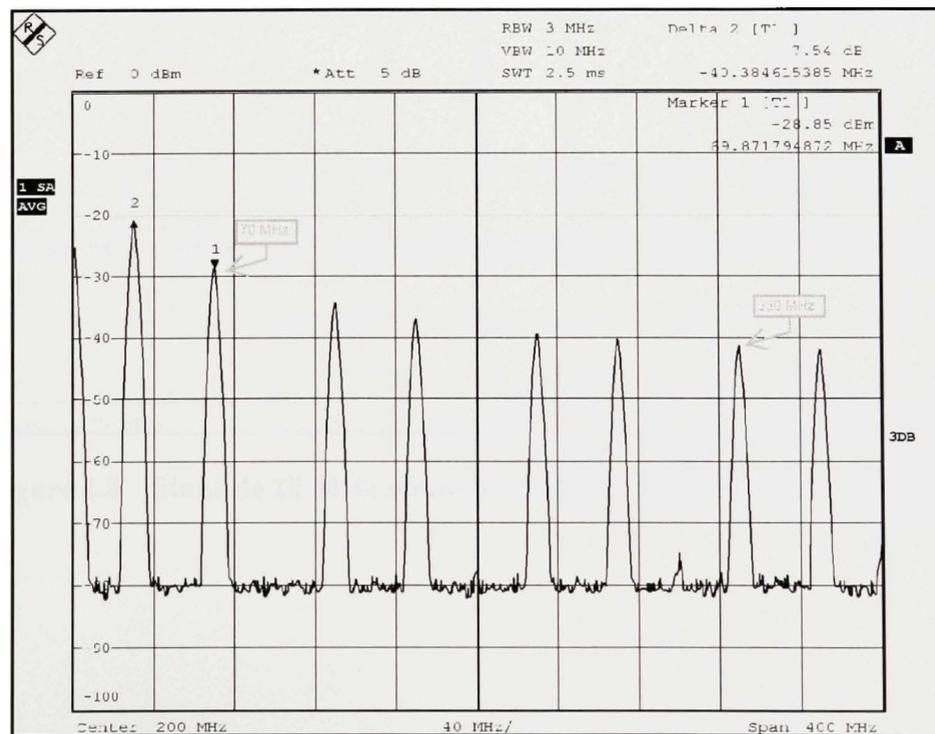


Figure 4.7 Sinus de 30 MHz du AD9777 sur 400 MHz.

À la Figure 4.8 on retrouve le AD9777 mais avec la sortie interpolée et modulée à 70 MHz. On remarque que les nombreuses harmoniques autour de 70 MHz sont absentes. Ce qui est exactement ce que l'on recherchait puisque l'entrée de la carte RF a un filtre d'une largeur de bande de 24 MHz centré sur 70 MHz. Le bruit de la carte de Lyrtech se trouve aux frontières de cette zone.

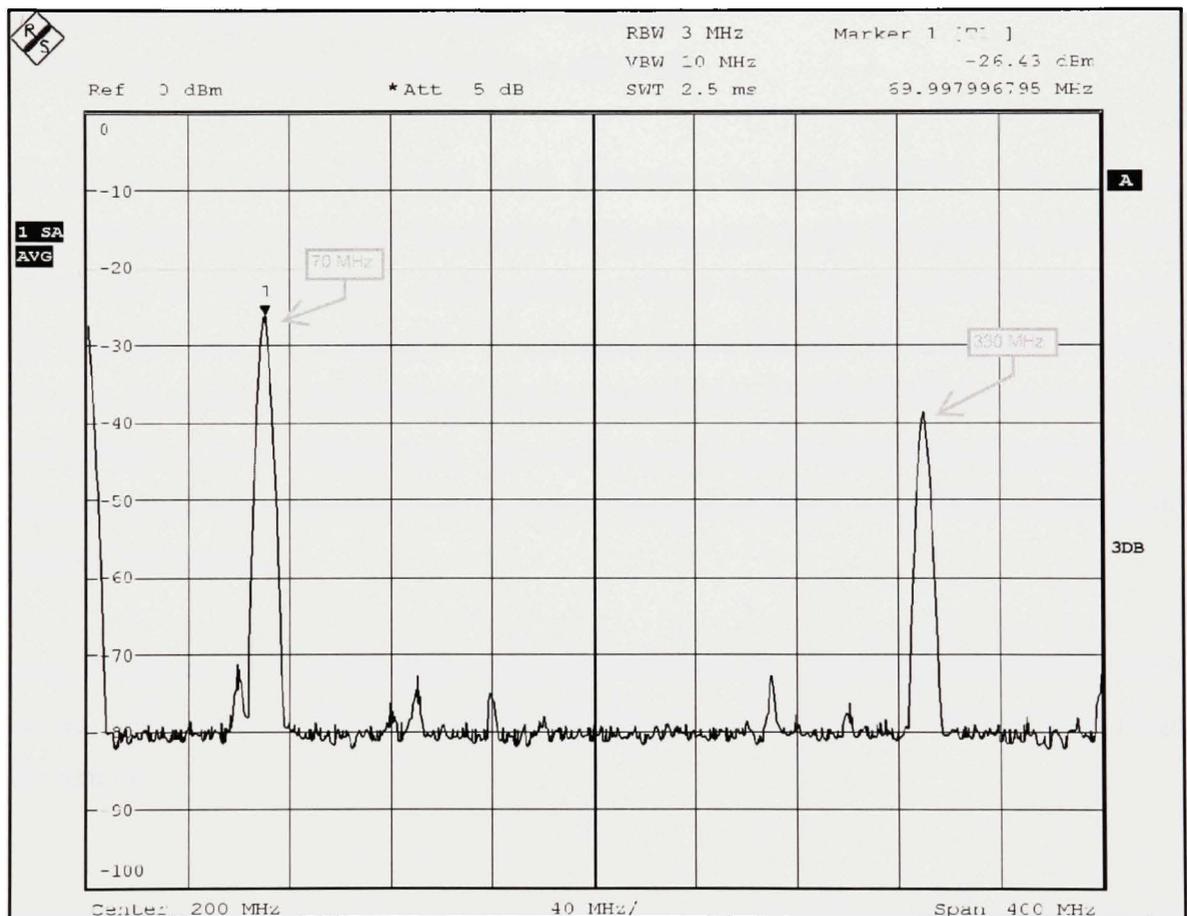


Figure 4.8 Sinus de 20 MHz modulé à 70 MHz du AD9777 sur 400 MHz.

Finalement, les trois figures suivantes nous donnent un aperçu des plages dynamiques. Le bruit aperçu sur les précédentes figures de Lyrtech se répercute sur sa plage dynamique même si la largeur de bande présente est de 10 KHz. Voici un petit résumé représentant les différentes plages dynamiques obtenues expérimentalement.

Lyrtech = 67.16 dB
AD9777 sans modulation ni interpolation = 89.43 dB
AD9777 avec modulation et interpolation = 100.79 dB

On remarque la grande différence entre Lyrtech et la carte AD9777. Cet écart est probablement dû à la mauvaise distribution de l'horloge sur la carte de Lyrtech de même que l'utilisation des amplificateurs opérationnels qui limite le domaine fréquentiel de la sortie. On voit clairement sur la Figure 4.9 le bruit situé juste à côté de l'harmonique de 70 MHz. Ce bruit est quasiment inexistant sur la nouvelle carte avec le AD9777 tel qu'on peut le voir sur la Figure 4.10 et la Figure 4.11. On remarque aussi que l'interpolation et la modulation interne ont un impact significatif sur la plage dynamique de sortie. Une différence de plus de 10 dB a été obtenue.

Ceci s'explique par la modulation qui permet d'éliminer les images de même que leur bruit en les superposant, ce qui augmente la puissance de sortie et par le fait même la plage dynamique.

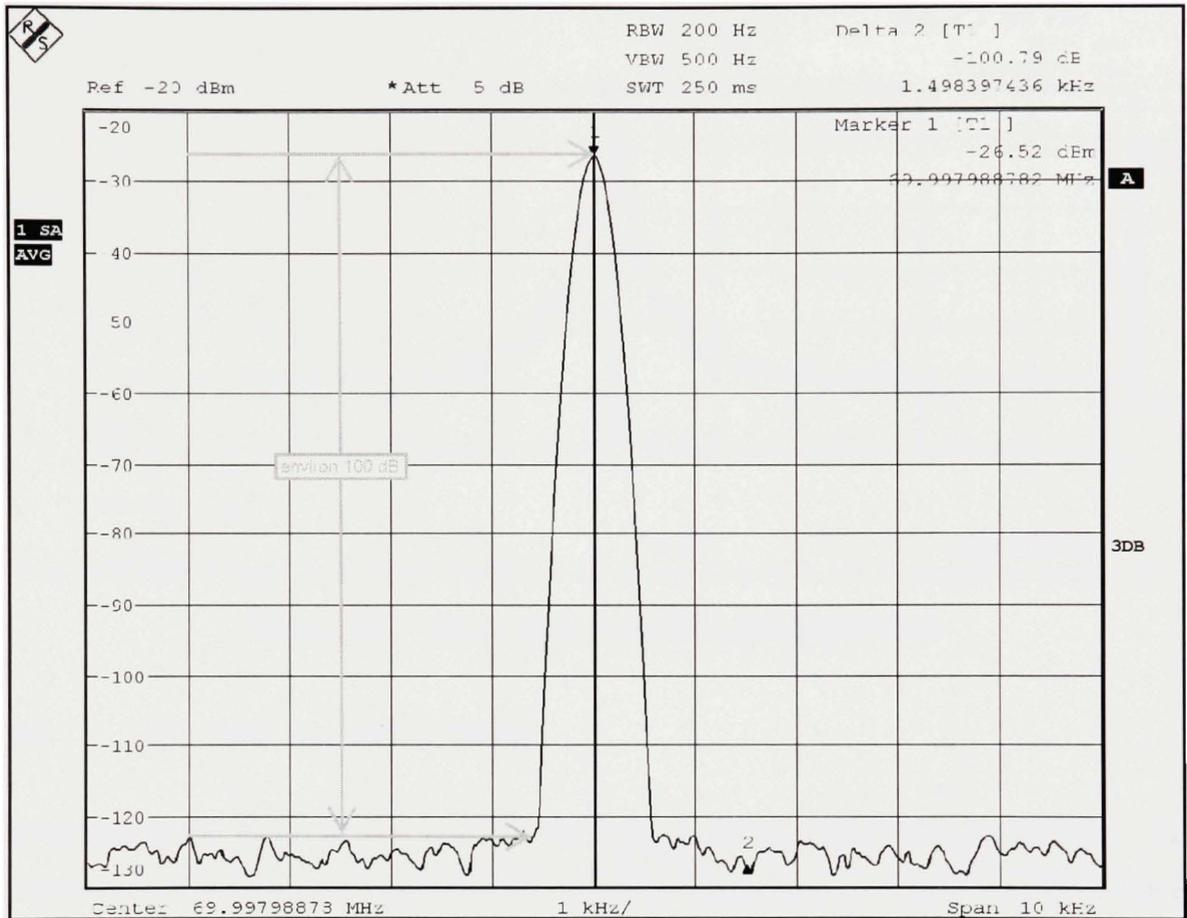


Figure 4.11 Sinus de 20 MHz modulé à 70 MHz du AD9777.

4.2 Résultats avec carte RF

La Figure 4.12 nous montre la sortie de la carte RF avec la modulation GPS. C'est un signal ressemblant à celui-ci qui sera dirigé vers un récepteur GPS. Puisque le rapport signal à bruit est de 71 dBc/Hz, il faudra atténuer le signal de sortie d'environ 30 dB pour avoir un rapport signal à bruit autour de 40 dBc/Hz à l'entrée d'un récepteur GPS.

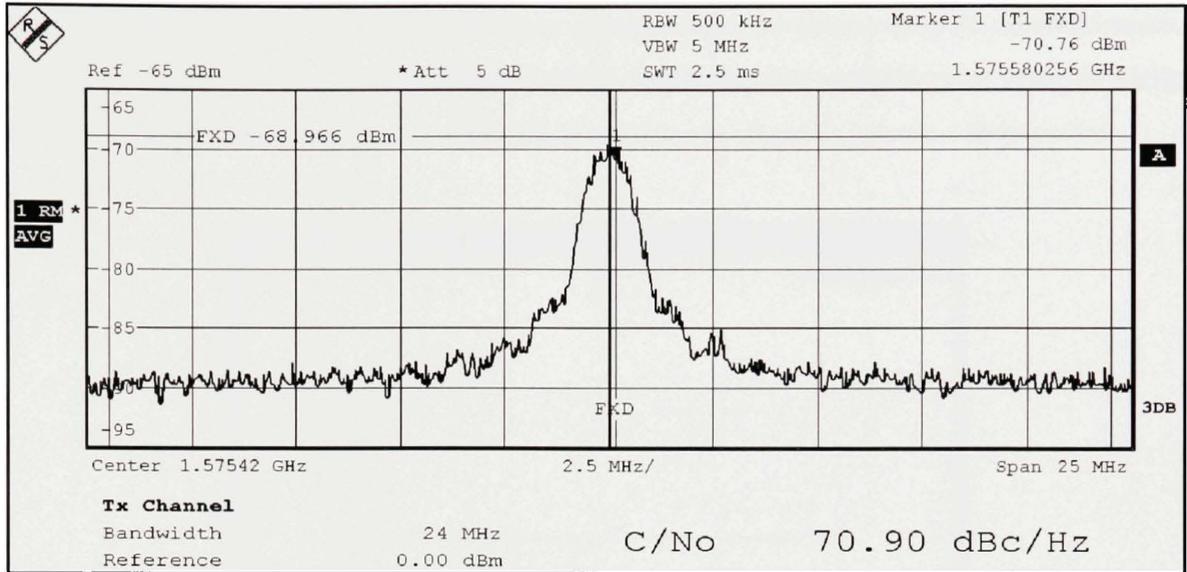


Figure 4.12 Sortie de la carte RF modulation GPS.

4.3 Résultats avec GPS et Galileo

Il est maintenant temps d'avoir une idée des résultats obtenus avec un récepteur GPS. La Figure 4.13 et la Figure 4.14 représentent les résultats provenant d'un récepteur GPS sur une durée de 20 min. L'information utile à retenir de ces captures d'écran est seulement l'allure des courbes. Les captures d'écrans ne représentent pas la terre mais plutôt une sphère de 15 mètres de diamètres permettant d'observer la précision de la solution de navigation sur une position fixe. La position étant supposée statique le décalage (« drift ») obtenu sur la solution de navigation correspond probablement à l'accumulation d'erreur sur l'horloge lors de l'enregistrement et lors de la simulation.

Il est à noter que ce résultat a été obtenu avec des données enregistrées par un autre récepteur et simplement rejoué par notre simulateur. Mais puisque le projet est toujours en développement côté logiciel pour pouvoir générer une solution de navigation complète sans partir d'enregistrement, il est difficile de savoir si ce décalage est causé par le matériel ou si c'est un artefact causé par l'enregistrement et la simulation de cet enregistrement.

On remarque qu'il n'y a aucune différence notable entre les deux cartes de convertisseurs. Encore une fois c'est seulement lorsque la partie logicielle sera complètement opérationnelle que l'on pourra distinguer si une amélioration notable est observée entre les deux cartes.

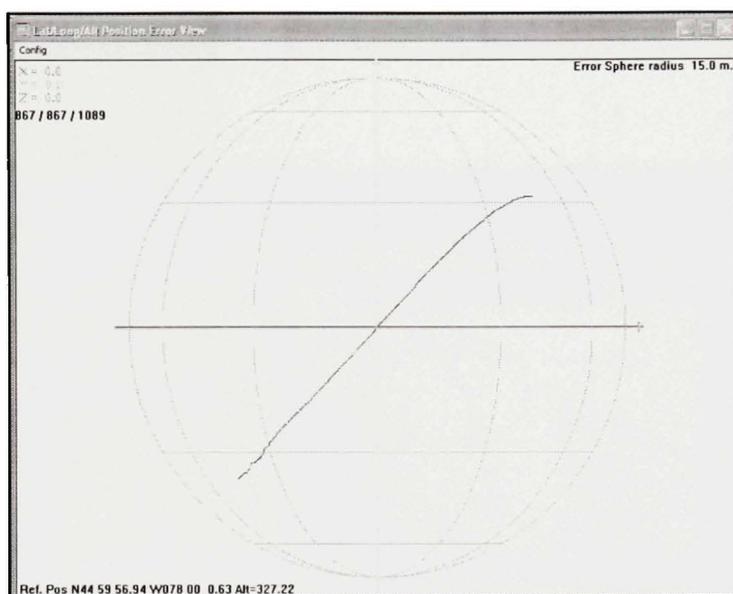


Figure 4.13 Résultat du récepteur GPS sur 20 minutes (Lyrtech).

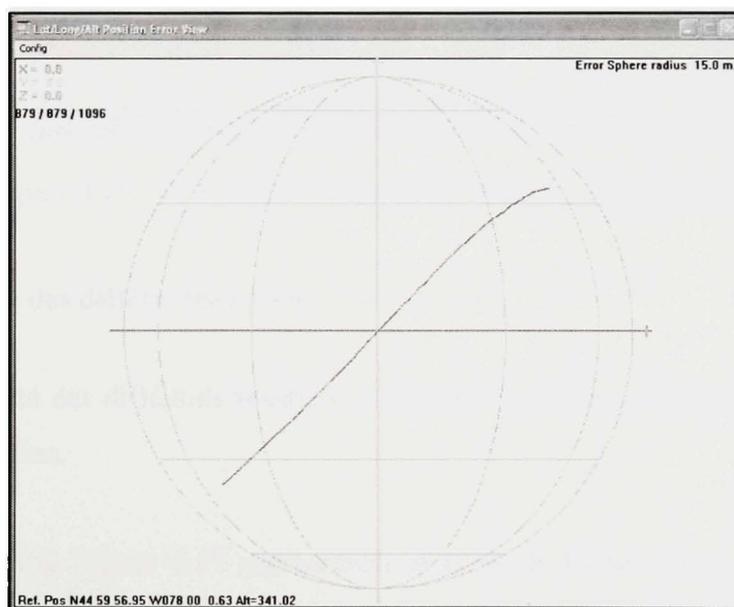


Figure 4.14 Résultat du récepteur GPS sur 20 minutes (AD9777).

Dans le cas où le décalage d'horloge serait inexistant, la solution de navigation aurait ressemblé à la Figure 4.15. Sur la Figure 4.13 et la Figure 4.14, nous avons un décalage d'environ 10 mètres par période de 20 minutes de simulation. Dans le cas de la Figure 4.15, on voit clairement que la solution de navigation se situe dans un rayon de cinq mètres. Il est clair qu'il reste encore beaucoup de développement à faire côté logiciel pour en arriver à cette précision.

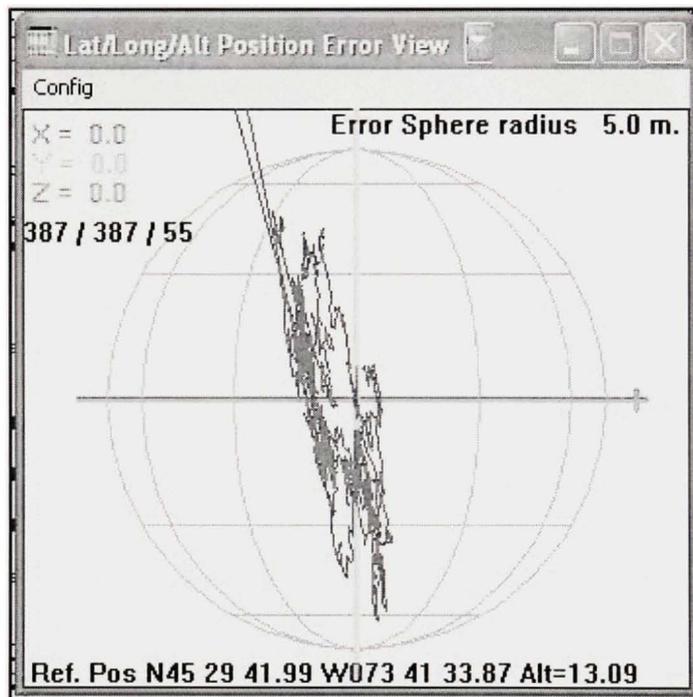


Figure 4.15 Résultats du récepteur GPS en temps normal.

4.4 Photos des différentes cartes

Après avoir discuté des différents résultats, il est intéressant de vous montrer concrètement les cartes en question.

La Figure 4.16 et la Figure 4.17 représentent la carte de Lyrtech avec et sans le boîtier protégeant du bruit électromagnétique. On remarque des lignes argentées entourant chacun des convertisseurs et des amplificateurs. Ces lignes servent de contact pour le boîtier. À la

Figure 1.2 on voit que la carte de développement FPGA est située au-dessus de la carte RF. Dans cette configuration, nous avons remarqué que le boîtier n'affecte pas la qualité des signaux de sorties puisqu'il est situé loin de toute sources de perturbation. Par contre quand la carte RF est située au-dessus de la carte FPGA, la carte mezzanine est située très proche de la carte RF et ses sorties sont très bruitées dans le boîtier de protection. Ce bruit est généré par les composants analogiques de la carte RF.

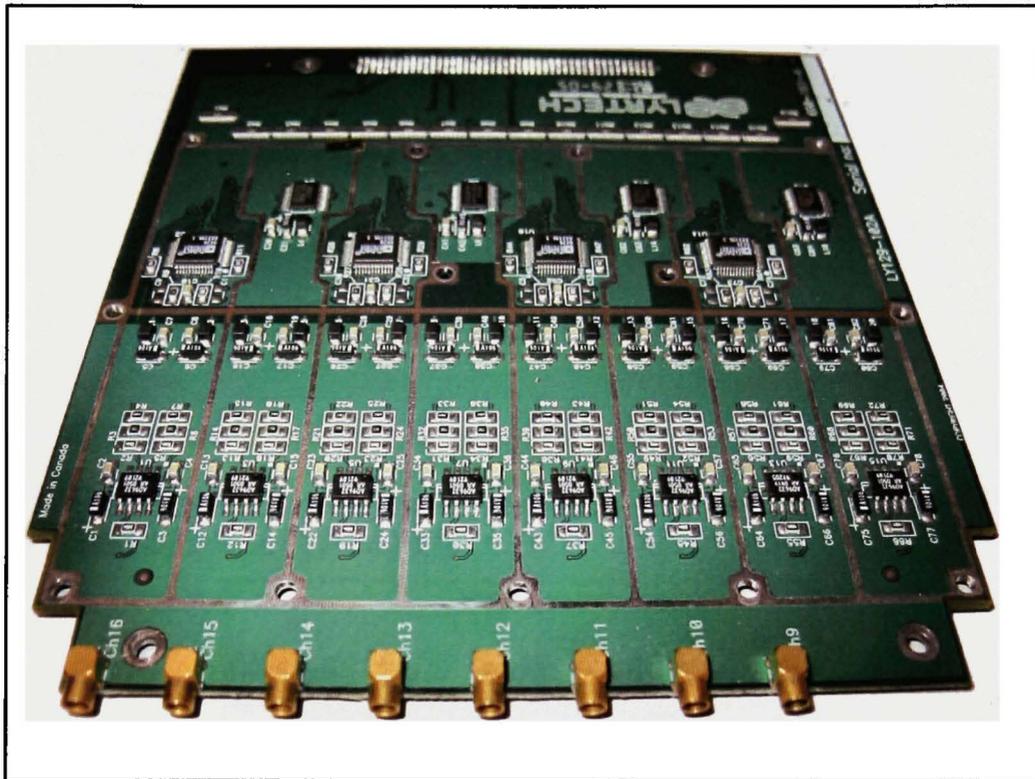


Figure 4.16 Carte de Lyrtech sans le boîtier.



Figure 4.17 Carte de Lyrtech avec boîtier.

La Figure 4.18 représente le prototype conçu pour tester le DAC AD9777 et sa capacité de modulation des signaux. Finalement la Figure 4.19 et la Figure 4.20 montrent la version finale de la carte conçue au CHAPITRE 3.

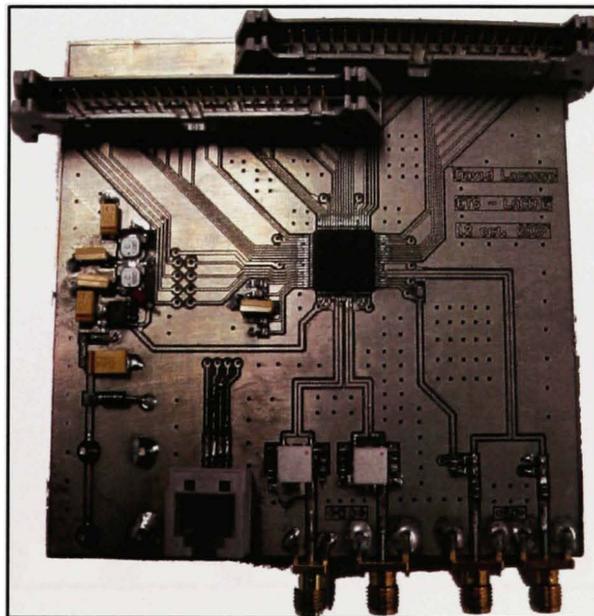


Figure 4.18 Carte AD9777 initiale.

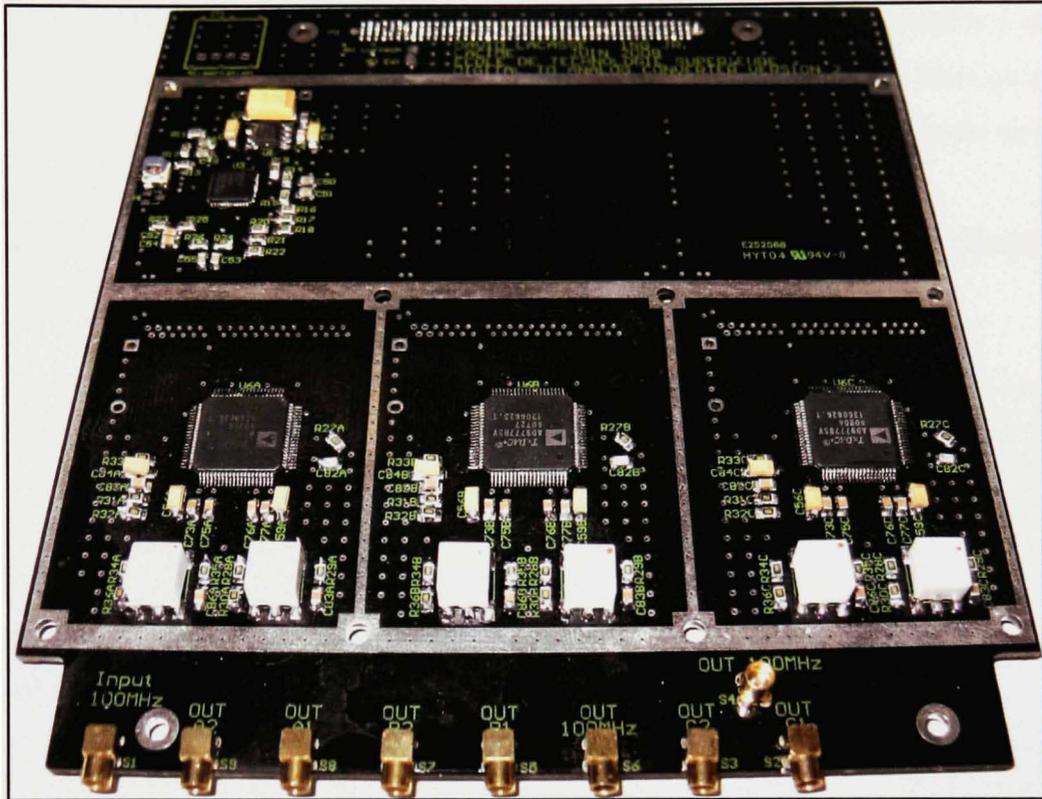


Figure 4.19 Dessus de la Carte AD9777 finale.

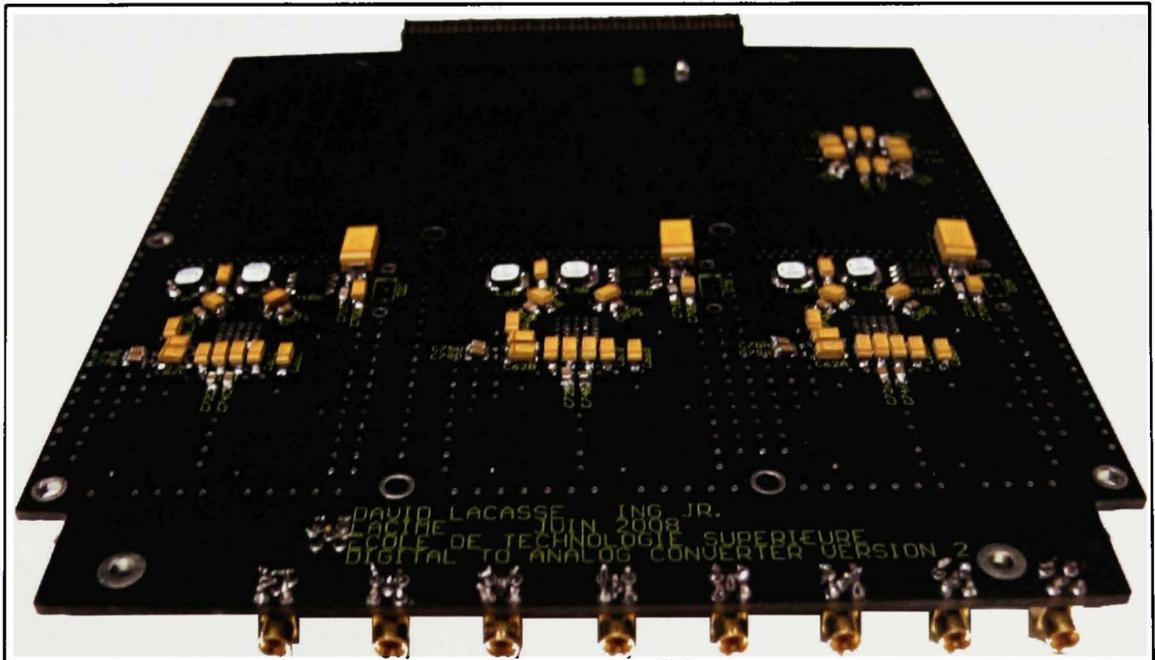


Figure 4.20 Dessous de la Carte AD9777 finale.

La carte AD9777 fut conçue de manière à pouvoir accueillir un boîtier semblable à celui de Lyrtech comme on peut remarquer les contours argent sur la Figure 4.19. Mais puisque la carte FPGA se situe au-dessus de la carte RF, il n'est pas nécessaire de concevoir un boîtier de protection.

La Figure 4.21 représente la version finale du simulateur. La plaque noire du dessus a été enlevée pour mieux montrer l'intérieur du simulateur.

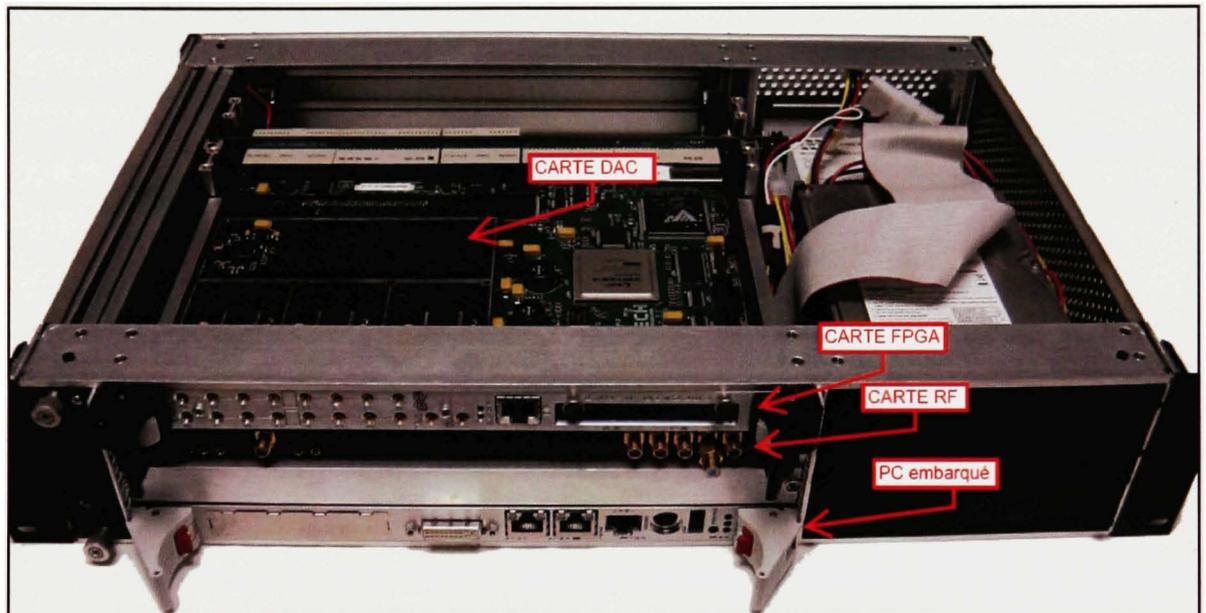


Figure 4.21 Photo du simulateur avec les nouveaux convertisseurs.

CONCLUSION

Il a été démontré lors de ce mémoire qu'un projet d'envergure comme ce simulateur de signaux GPS et Galiléo peut s'avérer complexe. Il est toujours important de bien planifier les différentes interactions possibles entre les différentes branches d'un projet (Logicielle, IF et RF). Cela ne peut qu'aider à mener le projet à terme.

Ce mémoire a présenté plusieurs facettes de la solution globale retenue, mais demeure quand même insuffisant pour expliquer l'envergure du projet dans le détail. Il serait prématuré d'énoncer que la problématique a été résolue. De futurs développements sont nécessaires pour parvenir au résultat escompté. Puisque ce projet est toujours en cours, de nombreuses informations contenues dans ce mémoire peuvent changer radicalement.

Pour conclure, ce projet complexe et complet par les nombreux domaines d'expertise auquel il touche fut très intéressant et m'a permis de développer mes connaissances dans bien des domaines auquel je ne m'attendais pas au début de ce projet.

Le projet du simulateur a de l'avenir, puisque davantage de produits utilisés tous les jours contiennent des récepteurs GPS et dans un futur proche des récepteurs Galileo aussi. Le positionnement en temps réel est loin de tomber dans l'oubli.

RECOMMANDATIONS

Comme dans tout projet, il y a toujours de la place pour l'amélioration. Évidemment, il faut savoir où s'arrêter lorsque les efforts d'amélioration ne valent pas l'amélioration elle-même. Concernant la partie IF, des efforts d'amélioration doivent être faits pour être en mesure de rentrer le plus de canaux possible dans le FPGA. La solution venant à l'esprit le plus rapidement serait de changer le FPGA pour un plus gros, mais cela impliquerait de profondes modifications du programme FPGA puisque le FPGA interagit avec de la mémoire externe et des convertisseurs numériques-analogiques. Dans cet éventualité, il faudrait aussi concevoir une carte pour adapter les connecteurs entre la carte DAC et les sorties du nouveau FPGA.

Il serait donc plus pratique d'optimiser le code de manière à diminuer les ressources que prend chaque composant du programme FPGA. Les options les plus plausibles pour optimiser les ressources sont simples, mais demandent quand même beaucoup de travail. Premièrement, il faudrait transférer tous les tampons en mémoire externe. Ce qui sauverait un espace non négligeable par canal. Par contre ces changements ne devraient pas se faire avant d'avoir analysé l'espace mémoire disponible en externe et les implications sur la partie logicielle et les registres du FPGA. Ensuite il faudrait analyser chaque composant en profondeur pour tenter d'utiliser les nombreux multiplicateurs internes (DSP48) pour remplacer les additionneurs ou multiplicateurs utilisant de la logique.

Concernant la carte AD9777, une analyse approfondie s'avérerait nécessaire pour balancer expérimentalement les sorties à 50 ohms. Puisque la carte AD9777 est difficilement accessible lorsqu'elle est connectée dans la plateforme, il est ardu de vérifier expérimentalement si les sorties sont à 50 ohms comme le prévoient les calculs théoriques. Par contre si on se fie à la plage dynamique qui est largement supérieure à celle de Lyrtech, on peut en déduire que les sorties sont convenablement balancées.

En résumé, puisque le projet est toujours en cours, ces optimisations peuvent attendre puisque le développement peut se poursuivre sans problème avec une dizaine de canaux.

ANNEXE I

REGISTRES DU FPGA

L'interface entre le Soft et la IF est essentiellement composée de registres. Ces registres sont situés dans le FPGA et dans la mémoire externe de la carte IF gérée par le FPGA. Voici un tableau spécifiant les différents registres.

Tableau I.1

Registres logiciel dans le FPGA

FSB address offset	R/W	Description
0x000	R/W	Channel Control Register
0x004	W	Codes Length Register 0
0x008	W	Codes Length Register 1
0x00C	R/W	Code Pointers Register 0
0x010	R/W	Code Pointers Register 1
0x014	W	Primary I-Code Write Register
0x018	W	Primary Q-Code Write Register
0x01C	-	Reserved
0x020	W	Secondary Codes Register 0
0x024	W	Secondary Codes Register 1
0x028	W	Secondary Codes Register 2
0x02C	W	Secondary Codes Register 3
0x030	R/W	Nav. Message FIFO pointers
0x034	W	Nav. Message Write Register
0x038-0x03F	W	Reserved
0x040-0x044	W	Carrier Initial Phase
0x048-0x04C	W	Code Initial Phase
0x050	W	Clocks Register 0
0x054	W	Clocks Register 1
0x05C	W	Clocks Register 2

Tableau I.2

Modèle de la description des registres

Bit number
Register description
Mnemonic
Initial value at reset

I.1 Contrôle du canal (Offset 0x000)

- **ON** : Active la modulation sur le canal

Quand ce bit est mis à 0, cela permet de fermer le canal, c'est-à-dire que la sortie est forcée à 0 pour ce canal. En pratique dans le FPGA c'est la valeur du gain du canal qui est forcée à 0.

- **IQ** : Active la modulation QPSK
- **BC** : Active la modulation Binary + carrier
- **Empty** : Drapeau de FIFO vide du message de Navigation
- **Almost Empty** : Drapeau indiquant que le FIFO est rempli à 5% ou moins
- **Almost Full** : indiquant que le FIFO est rempli à 95% ou plus
- **Full** : indiquant que le FIFO est rempli à 5%
(Avertissement : Si ce drapeau est levé, les données du FIFO risquent d'être désynchronisées)

Tableau I.3

Offset 0x000 - Contrôle du canal

31 to 7	6	5	4	3	2	1	0
Reserved	Message FIFO				Modulation		
R	Full	Almost Full	Almost Empty	Empty	IQ	BC	ON
0	0	0	0	1	0	0	0

I.2 Registres de longueur du Code (Offsets 0x004-8)

- **PICL** : Primary I-code (phase code) length [en chips]
- **PQCL** : Primary Q-code (quadrphase code) length [en chips]
- **SICL** : Secondary I-code (phase code) length [en chips]
- **SQCL** : Secondary Q-code (quadrphase code) length [en chips]

Tableau I.4

Offset 0x004 Longueur du code Registre 0

31 to 30	29 to 16	15 to 14	13 to 0
Reserved	Primary codes length		
R	PQCL	R	PICL
0	0x1	0	0x3FF

Tableau I.5

Offset 0x008 Longueur du code Register 1

31 to 23	22 to 16	15 to 5	4 to 0
Reserved	Secondary codes length		
R	SQCL	R	SICL
0	0x1	0	0x1

I.3 Registres des pointeurs du Code (Offsets 0x00C-10)

- PICP: Primary I-code pointer
- PQCP: Primary Q-code pointer

Ces deux pointeurs sont utilisés pour la lecture et l'écriture des « buffers » (tampon) du Code. Ces pointeurs sont incrémentés de 32 bits à chaque écriture pour faciliter l'entrée d'un nouveau code. Lors de chaque écriture, les cinq LSB de chaque pointeur ne sont pas pris en charge puisque 32 bits sont écrits à chaque écriture.

Lors de l'activation du canal, les 2 pointeurs sont incrémentés de 1 chip à chaque front montant de l'horloge primaire (Primary clock - PCLK). Ces pointeurs doivent être initialisés à la bonne phase du chip initial lorsque l'écriture du nouveau code est complétée.

- SICP: Secondary I-code pointer
- SQCP: Secondary Q-code pointer

Ces pointeurs servent essentiellement à garder une trace sur l'état actuel de la sortie et doivent être initialisés à la bonne phase du chip initial.

Tableau I.6

Offset 0x00C Pointeurs du code Register 0

31 to 30	29 to 16	15 to 14	13 to 0
Reserved	Primary code pointers		
R	PQCP	R	PICP
0	0	0	0

Tableau I.7

Offset 0x010 Pointeurs du code Register 1

31 to 23	22 to 16	15 to 5	4 to 0
Reserved	Secondary codes pointers		
R	SQCP	R	SICP
0	0	0	0

I.4 Registres d'écriture des Codes Primaires I/Q (Offsets 0x014-8)

Ces registres servent à entrer des données dans les tampons de code primaire. L'adresse d'écriture du tampon est donnée par les registres décrits précédemment. Le premier chip correspond au MSB du registre et le dernier au LSB.

Tableau I.8

Offset 0x014 Code Primaire I Registre d'écriture

31 to 0
Primary I-Code LUT Input
PICW
LUT table initialized with GPS PRN #1 Code

Tableau I.9

Offset 0x018 Code Primaire Q Registre d'écriture

31 to 0
Primary Q-Code LUT Input
PQCW
LUT table initialized with 0

I.5 Registres des Codes Secondaire (Offsets 0x020-4-8-C)

- Contrairement au tampon de code primaire, les codes secondaires sont entrés directement dans ces quatre registres. Ici aussi le premier chip correspond au MSB et le dernier au LSB.

Tableau I.10

Offset 0x020-0x028 Code Secondaire Registres 0-2

Register 0: 31 to 0	Register 1: 31 to 0	Register 2: 31 to 0
Secondary Q- Code 0 to 31	Secondary Q- Code 32 to 63	Secondary Q- Code 64 to 95
SQCW2	SQCW1	SQCW0
0	0	0

Tableau I.11

Offset 0x02C Code Secondaire Registre 3

31 to 24	23 to 4	3 to 0
Reserved	Secondary I- Code	Secondary Q- Code 96 to 99
R	SICW	SQCW3
0	0x00001	0

I.6 Pointeurs du FIFO du Message de Navigation (Offset 0x30)

- MWP: Navigation message write pointer
- MRP: Navigation message read pointer

Ces pointeurs ont la même utilité et la même fonctionnalité que les pointeurs des codes primaires. Par contre, ces deux pointeurs sont utilisés pour pouvoir écrire de nouvelles données tout en continuant de lire les données existantes à une adresse différente. Au même type que le code primaire, le pointeur d'écriture s'incrémente lui-même de 32 chips à chaque écriture dans le FIFO. D'un autre côté, le pointeur de lecture s'incrémente de 1 chip à chaque front

montant de la MCLK (horloge du message). Il est à noter que ces pointeurs doivent seulement être initialisés lors du transfert initial du message au FPGA et ne devraient jamais être altérés par la suite durant l'utilisation normale.

Tableau I.12

Offset 0x030 Message de Navigation pointeurs FIFO

31 to 30	29 to 16	15 to 14	13 to 0
Reserved	Read/write pointers		
R	MRP	R	MWP
0	0x3FFF	0	0

I.7 Registre d'écriture du Message de Navigation (Offset 0x34)

- Ces registres servent à entrer des données dans le FIFO du message de navigation. L'adresse d'écriture est donnée par le pointeur discuté précédemment. Pour éviter que le message devienne désynchronisé, il est important de vérifier continuellement le drapeau « Almost Full » et d'arrêter l'écriture lorsque ce drapeau est levé. Encore une fois le premier bit correspond au MSB du registre et le dernier au LSB.

Tableau I.13

Offset 0x034 Message de Navigation Registre d'écriture

31 to 0
Navigation Message FIFO Input
MW
FIFO initialized with 0

I.8 Registres des phases initiales de la porteuse et du Code (Offsets 0x40-4-8-C)

- CrP0/1: Carrier initial phase
- CdP0/1: Code initial phase

Dues à la précision limitée de leur fréquence, les phases des oscillateurs ont tendances à dériver de leur valeur initiale à mesure que le temps avance. Pour corriger cette erreur accumulée, les oscillateurs prennent automatiquement une nouvelle phase corrigée des registres CrP et CdP à chaque front montant du signal « phase strobe » (généralisé toutes les secondes, c'est-à-dire chaque modulo 100 « epochs »).

Tableau I.14

Offset 0x044-0x040 Porteuse (Phase initiale)

Register 1: 31 to 4	Register 1: 3 to 0	Register 0: 31 to 0
Reserved	Init. Phase 35 to 32	Init Phase 31 to 0
R	CrP1	CrP0
0	0	0

Tableau I.15

Offset 0x04C-0x048 Code (Phase initiale)

Register 1: 31 to 16	Register 1: 15 to 0	Register 0: 31 to 0
Reserved	Init. Phase 47 to 32	Init Phase 31 to 0
R	CdP1	CdP0
0	0	0

I.9 Registres des horloges (Offsets 0x50-4-C)

- SCLKR: Primary to secondary codes clock frequency ratio
- SCLKP: Initial secondary code clock frequency divider phase value
- MCLKR: Primary code to navigation message clocks frequency ratio
- MCLKP: Initial navigation message clock divider frequency phase value

Les registres SCLKP et MCLKP n'ont pas à être rafraîchis à chaque seconde comme les phases de la porteuse et du code. Ces valeurs sont seulement prises en compte quand le bit indiquant le canal « ON » est à 0.

Tableau I.16

Offset 0x050 Horloge Registre 0

31 to 30	29 to 16	15 to 14	13 to 0
Reserved	SCLK Init. Phase	Reserved	PCLK to SCLK Ratio
R	SCLKP	R	SCLKR
0	0	0	0x1

Tableau I.17

Offset 0x054 Horloge Registre 1

31 to 20	19 to 0
Reserved	PCLK to MCLK ratio
R	MCLKR
0	0x04FEC

Tableau I.18

Offset 0x058 Horloge Registre 2

31 to 20	19 to 0
Reserved	MCLK Init. Phase
R	MCLKP
0	0

ANNEXE II

EXTRAITS DES REQUIS DU PROJET SIMULATEUR GPS/GALILEO (TRADUCTION LIBRE)

Besoins concernant la partie IF du projet

- Signaux à Générer :
 - 12 signaux GPS L1
 - 12 signaux GPS L5
 - 6 signaux Galileo E1
 - 6 signaux Galileo E5a
 - 6 signaux Galileo E5b

- La partie IF devra générer les signaux précédents centrés à la fréquence intermédiaire de 35 MHz. Les signaux IF devront être basés sur la spécification de Novatel pour le convertisseur IF/RF [NOVATEL].

- Le simulateur devra fournir une sortie IF pour chaque fréquence de porteuse pour un total de cinq sorties.

- Le simulateur devra permettre de suivre un minimum de 15 satellites Galileo sur une période de simulation de 24 heures, minimum.

- La partie IF devra fournir des sondes internes (comme le signal d'horloge) pour déverminer ou tester le système.

- Quand une simulation est en route, la partie logicielle devra afficher les paramètres pour chaque canal généré. [SPIRENT section 10.1.1.1]

- Le simulateur devra être conçu pour faciliter les mises-à-jours pour des besoins futurs. En particulier pour les mises-à-jours dans la constellation Galileo.
- Le simulateur doit permettre de simuler des CNo supérieurs à +50dBHz (but : 60dBHz), mesuré à la sortie de la IF et RF.

ANNEXE III

NOTES SUR LES REQUIS

Title: Notes on IF signal processing

Prepared for: CMC Electronics

Jean Belzile
By : _____ Scientific Authority
Signature : _____

LIMITED RIGHTS NOTICE

This document is produced under contract with CMC Electronics Inc., dated November 23, 2005. It contains information proprietary to École de technologie supérieure or to a third party to which École de technologie supérieure may have legal obligation to protect such information from unauthorized disclosure, use or duplication of this document or of any of the information contained herein for other than the specific purpose for which it was disclosed is expressly prohibited except as École de technologie supérieure may otherwise agree to in writing.

III.1 Clocks

The OCXO reference clock is set up on the RF board. It has the following characteristics:

- HCMOS output at 100 MHz
- Aging by year ± 50 ppb
- Tuning range ± 2 ppm
- Rise and fall time less than 15 ns

Besides the 100 MHz reference clock, there are four other frequencies considered in this design:

- Code Clock
- Symbol Clock
- Tick clock
- High speed signal processing clock

III.1.1 Available clocks

The 100 MHz reference clock, tick clock and high speed signal processing clocks are actual clocks. The others are clock enable signals related to the reference clock. This avoids unnecessary skew, jitter and noise that may occur in the processes of frequency generation and distribution.

III.1.1.1 100 MHz reference clock

The 100 MHz reference clock is the RF time reference for the entire design. This clock can be calibrated to within 0.04PPM and has extremely low phase noise. To maximize the purity of the clock, it should enter the FPGA and go to the first clock buffer available. No DCM should be used on this clock to process it in any way.

This clock will serve as the reference clock for all of the signal processing. The other clocks will be clock enables referred to this clock.

The phase noise of this OCXO is given by:

-65dBc/Hz	1 Hz
-95dBc/Hz	10 Hz
-120dBc/Hz	100 Hz
-130dBc/Hz	1 KHz
-135 dBc/Hz	10 KHz
-140 dBc/Hz	100 KHz

This leads to an Allan variance of 1×10^{-11} at 1Hz. Integrated from 1Hz to 10KHz we get a total RMS jitter of 5.6×10^{-4} rads RMS (0.03 degrees RMS or 1×10^{-12} seconds RMS).

III.1.1.2 Code Clock

This clock is used inside the SV Core. When the SV Core is configured as a GPS signal generator the clock rate is 1.023MHz. When configured in Galileo mode, the clock has a rate of 10.230MHz.

These rates are generated via a Modulo-N integrator. The Modulo-N integrator adds a phase value to the current accumulated phase. Every time the integrator goes over the 0 value then it has performed a full cycle. To generate a clock signal then only the MSB of the Integrator is required.

If N is chosen to be the integrator operating frequency then phase value corresponds to the value in Hz of the desired clock. For example a 1 second clock would use 1 as phase value whereas a 1.023MHz rate is obtained by using 1.023M for phase value.

Using the integrator frequency, in this case the reference clock, as N implies a 27 bit integrator. There is little need for more as the precision of the 100 MHz clock is not expected to exceed 0.04 PPM or 4 Hz.

III.1.1.3 Symbol Clock

The bit clock is a divided version of the code clock. It corresponds to the symbol rate. If the modulation is BPSK, as in the GPS case, then this also corresponds to the bit rate. When a QPSK modulation is used then this corresponds to half of the bit rate.

This clock rate is used to pull complex data out of the message FIFO. When a BPSK modulation is used then the imaginary part is assumed to be 0. It is further assumed that the content of the FIFO represents properly framed, interleaved, coded, scrambled and modulated signals.

III.1.1.4 Tick clock

The tick clock is used to generate a programmable clock of 50% duty cycle. It is directly derived from the reference clock and will have the same precision as the original. Since it is derived from the 100MHz reference clock, the precision on the rising edge will be 10 ns peak-peak.

III.1.1.5 High speed signal processing clock

The high speed signal processing clock is a 200 MHz version of the reference clock. It is used to clock DSP blocks of the Virtex 4 in order to minimize area and power consumption. It is generated by a DCM.

III.1.2 Clock distribution

An ultra low jitter clock distributor provides an 8 output clock distribution and 2 inputs reference clocks. One of the inputs is connected to the OCXO while the other is connected to an external SMA connector to allow external synchronisation.

Four of the eight outputs are LVPECL (1.2 GHz), and four are selectable as either LVDS (800 MHz) or CMOS (250 MHz) levels. Each output has a programmable divider that may be bypassed or set to divide by any integer up to 32. The phase of one clock output relative to another clock output may be varied by means of a divider phase select function that serves as a coarse timing adjustment. Two of the LVDS/CMOS outputs feature programmable delay elements with full-scale ranges up to 10 ns of delay. This fine tuning delay block has 5-bit resolution, giving 32 possible delays from which to choose for each full-scale setting.

Three of the LVDS/CMOS are used to drive the reference input of the 3 frequency synthesiser of the local oscillators. One LVPECL output is used to drive the IF board at 100 MHz. Finally another LVPECL output will be use to provide the 10 MHz trigger output (TBD dBm, 50 Ohms SMA connector).

III.2 Quantization Noise

This section will address some of the quantization noise issue on the various cores of the IF section:

- SV Core
- Top level Core
- Interferer Core
- Tick Clock Core

As a general rule, all numbers are expressed in two's complement notation expressed with the notation [a,b] where a-1 represents the weight of the MSB and b represents the number of digits after the decimal point.

III.2.1 SV Core

The SV Core has two important sections. The clock generation section which has been dealt with previously and the signal processing section. This is the section we want to treat here.

III.2.1.1 Carrier Generation

The carrier has to have 0.02 rad RMS of phase jitter at 1.6 GHz this leads to a requirement of 2.5×10^{-4} rad RMS at 20 MHz which is the first IF frequency. If we consider this to be a target for the peak-to-peak error we then must be able to quantize the phase down to this level we must quantize the circle into 25132.74 steps or a requirement for 15 bits of phase resolution. A 15 bit phase resolution implies a ± 0.01 rad peak-to-peak error. This is a minimum of 10 dB lower than the requirement.

In order to keep the quantization noise of the sine and cosine values at approximately the same level they also need to be of the same values because $\sin(x) \approx x$ when x is small. We consider that the sine and cosine amplitude are to be defined between $]-1,1[$. The bits assignments is thus [1, 14].

III.2.1.2 Interpolation Filters

There are two interpolation filters in the chain. The first interpolation filter increases the sample rate from 1.023 Msps (GPS) or 10.23 Msps (Galileo) to 100 Msps. The second interpolation filter is located inside the AD9777 DAC from Analog Device. It interpolates by 4 and is composed of two half-band filters.

DIGITAL FILTER SPECIFICATIONS

Table 4: Half-Band Filter No. 1: 43 Coefficients

Tap	Coefficient
1: 43	8
2: 42	0
3: 41	-29
4: 40	0
5: 39	67
6: 38	0
7: 37	-134
8: 36	0
9: 35	244
10: 34	0
11: 33	-414
12: 32	0
13: 31	673
14: 30	0
15: 29	-1009
16: 28	0
17: 27	1772
18: 26	0
19: 25	-3290
20: 24	0
21: 23	10364
22	16384

Table 5: Half-Band Filter No. 2: 19 Coefficients

Tap	Coefficient
1: 19	19
2: 18	0
3: 17	-120
4: 16	0
5: 15	438
6: 14	0
7: 13	-1288
8: 12	0
9: 11	5047
10	8192

Table 6: Half-Band Filter No. 3: 11 Coefficients

Tap	Coefficient
1: 11	7
2: 10	0
3: 9	-53
4: 8	0
5: 7	302
6	512

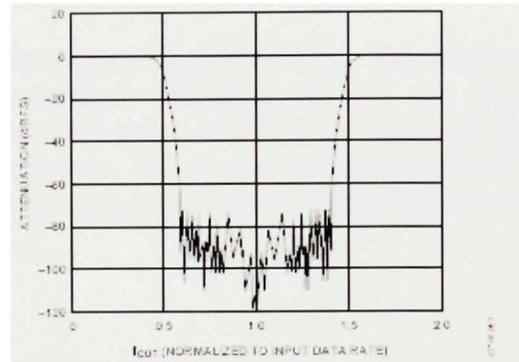


Figure 2: 2x Interpolating Filter Response

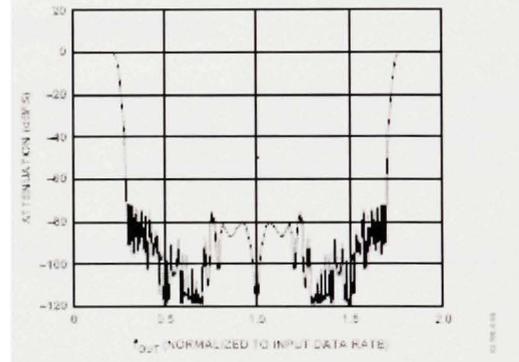


Figure 3: 4x Interpolating Filter Response

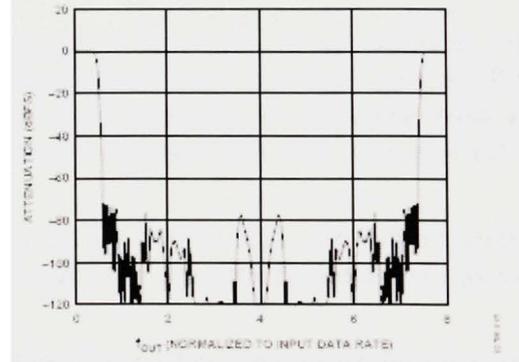


Figure 4: 8x Interpolating Filter Response

III.2.1.3 100 Msps Interpolation Filter

The 100 Msps interpolation filter is a 0th order a sample and hold filter. This implies that this filter will not introduce any phase or amplitude distortion. It will not add any quantization noise nor will it eliminate any either. This means that both the GPS and Galileo signals after interpolation will look like $\sin(x)/x$ signals. This is not a problem as there is no requirement on spectrum shape other than the minimum 3 dB bandwidth desired is to be no less than 24 MHz.

III.2.1.4 DAC interpolation filter

The half-band filters 1 and 2 are used when interpolating by 4. The coefficients and frequency response of the three half-band filters are given above.

From these figures, one can set filter 1 to a half-band filter because the spectral content of the data is centered at 20 MHz and has a span of +/-12 MHz. The upper frequency is 32 MHz. Clearly the second filter must be set to zero stuffing only because the low pass response is too narrow.

Note that since the half-band filters are symmetrical FIR filters, i.e. $h(n) = h(-n)$, then they are zero phase filters.

III.2.1.5 Gain Control

The control should add 40 dB of dynamic range to its input. This is an additional 7 bits or 42 dB. A control gain of 1 represents the minimum output level. A control gain of 10 is the nominal value and a control gain of 100 is the maximum output level. The range from 101 to 127 should never be used is left for head room for the top level core. The value 0 is valid and represents no signal.

A minimal resolution of 0.5 dB on the signal power implies that gain control shall have additional bits after the decimal point. An additional 5 bits after the decimal point provides a maximum step size 0.26 dB.

The command of the AGC thus requires 12 bits where 5 bits are located after the decimal point. This is denoted by [7, 5].

III.2.1.6 Doppler and Position Correction Update Rates

Acceleration max (see specs) : $a_{\max} = 45 \frac{m}{s^2}$

Jerk max (see specs) : $j_{\max} = 55 \frac{m}{s^3}$

Wavelength L1 : $\lambda_{L1} = 190mm$

$a_{\max} \Rightarrow$ Maximum rate of change of Doppler due to receiver dynamic :

$$d_{RXDop} = \frac{a_{\max}}{\lambda_{L1}} = \frac{45}{19e-3} = 236.8 \frac{Hz}{s}$$

$j_{\max} \Rightarrow$ Maximum accel. of change of Doppler due to receiver dynamic :

$$dd_{RXDop} = \frac{j_{\max}}{\lambda_{L1}} = \frac{55}{19e-3} = 289.5 \frac{Hz}{s^2}$$

Maximum rate of change of Doppler due to satellite motion : $d_{SATDop} \approx 1 \frac{Hz}{s}$

Maximum accel. of change of Doppler due to satellite motion : $dd_{SATDop} \approx 0.3 \frac{mHz}{s}$

$d_{RXDop} + d_{SATDop} \Rightarrow$ Maximum total rate of change of Doppler : $d_{Dop} \approx 238 \frac{Hz}{s}$

$dd_{RXDop} + dd_{SATDop} \Rightarrow$ Maximum accel. of change of Doppler : $dd_{Dop} \approx 290 \frac{Hz}{s}$

Max error = 10 mm (see specs)

Max rate error = 1 mm/s (see specs)

The maximum error on the Doppler is

$$\Delta ErrDoppler_{\max} = \frac{\text{Max rate error}}{\lambda_{L1}} = \frac{1e-3}{19e-2} = 5.26mHz . \text{ So the data update has to be done in}$$

order to respect the maximal Doppler error of 5.26 mHz.

Case 1: Influence of $d_{Dop} = 238 \frac{Hz}{s}$

$$d_{Dop} = \frac{\Delta ErrDoppler_{\max}}{\Delta t} \Rightarrow \Delta t = \frac{\Delta ErrDoppler_{\max}}{d_{Dop}} = \frac{5.26mHz}{238} = 22\mu s \text{ (or 45 kHz)}$$

So the Doppler values should be updated at least every $22\mu s$ in order to verify the acceleration specifications.

Case 2: Influence of $j_{\max} = 290 \frac{Hz}{s^2}$

$$dd_{Dop} = \frac{2 \cdot \Delta Doppler_{\max}}{\Delta t^2} \Rightarrow \Delta t = \sqrt{\frac{2 \cdot \Delta Doppler_{\max}}{dd_{Dop}}} = \sqrt{\frac{2 \cdot 5.26e-3}{290}} = 6ms$$

So the Doppler values should be updated at least every $6ms$ (133 Hz) in order to verify the maximal jerk specifications.

Case 3: Influence of a_{\max} **and** j_{\max}

The fastest rate of Doppler update should be $22\mu s$ in order to verify maximal acceleration and jerk specifications. However, this rate is too high for the software. Thus, a linear interpolation algorithm shall be developed at the IF stage to reduce the software update frequency. Here, the impact of the maximal jerk on the linear interpolation is studied:

Real Doppler:

$$Doppler(t) = Doppler_0 + d_{Dop}t + \frac{dd_{Dop}}{2}t^2$$

Interpolated Doppler:

$$\begin{aligned} Doppler_{interpol}(t) &= \frac{\Delta Doppler}{\Delta t}t + Doppler_0 \\ &= \left(d_{Dop} + \frac{dd_{Dop}}{2} \Delta t \right) t + Doppler_0 \end{aligned}$$

Interpolated Doppler error due to jerk contribution:

$$\begin{aligned} Doppler_{ERR}(t) &= \left| Doppler(t) - Doppler_{interpol}(t) \right| \\ &= \left| d_{Dop}t + \frac{dd_{Dop}}{2}t^2 - \left(d_{Dop} + \frac{dd_{Dop}}{2} \Delta t \right) t \right| \\ &= \frac{dd_{Dop}}{2} \left| t^2 - t\Delta t \right| \end{aligned}$$

Or, this term is maximal where :

$$\frac{d(t^2 - t\Delta t)}{dt} = 0 \Rightarrow t = \frac{\Delta t}{2}$$

So, the maximal error is expressed as:

$$\Delta Err Doppler_{max} = \frac{dd_{Dop}}{8} \Delta t^2$$

From this last equation, it is possible to determine the largest Doppler update interval:

$$\Delta t = \sqrt{\frac{8 \cdot \Delta Err Doppler_{max}}{dd_{Dop}}} = \sqrt{\frac{8 \cdot 5.26e-3}{290}} = 12ms$$

By choosing a software update rate of 100Hz (10ms), the Doppler error becomes:

$$\Delta Err Doppler_{max} = \frac{dd_{Dop}}{8} \Delta t^2 = \frac{290}{8} (10e-3)^2 = 3.625mHz$$

Thus, the error margin left for the interpolation (in the FPGA) update rate is:

$$ErrDoppler_{margin} = 5.26 - 3.625 = 1.635mHz$$

Using Case 1 equation, the slowest interpolation rate becomes:

$$\Delta t = \frac{ErrDoppler_{margin}}{d_{Dop}} = \frac{1.635mHz}{238} = 6.87\mu s \text{ (or 146 kHz)}$$

In conclusion, with a Doppler software update rate of 100Hz (10ms) and an interpolation rate (in the FPGA) of 200 kHz (5μs) the specifications are fully verified.

At last, by verifying the 1 mm/s specification, the satellites position shall be corrected at least once in every 10 seconds to comply with the 10 mm maximal error specification.

III.2.1.7 Carrier NCO Sizing

To comply with the specifications, every SV Doppler should be updated by the software core at a rate of 100 Hz and should be interpolated by the IF core at a rate of 200 kHz. By interpolating at a rate of 1 MHz instead, the remaining Doppler error margin becomes:

$$\begin{aligned} ErrDoppler_{remmargin} &= ErrDoppler_{margin} - d_{RXDop} \times f_{interpol} \\ &= 1.635 \times 10^{-3} - 238 \times 10^{-6} = 1.397mHz \end{aligned}$$

Thus, in order to respect the 1 mm/s specification, all Carrier NCO must have a minimal frequency resolution of 1.397mHz. In that purpose, the NCO accumulator size in bits is given by:

$$N_{NCO} = \log_2 \left(\frac{f_s}{ErrDoppler_{remmargin}} \right) = \log_2 \left(\frac{100 \times 10^6}{1.397 \times 10^{-3}} \right) = 36bits$$

III.2.1.8 Code NCO Sizing

To reflect the worst case for the code NCO sizing, GPS L1 C/A code must be used for computations as it has the slowest chipping rate (1.023 MHz compared to 10.23 MHz for all

other GNSS signals generated). Thus, the code NCO accumulator size is given by the same equation has seen above, except the fact a scaling factor equal to the carrier to code frequency (chipping rate) is introduced. This scaling factor reflects the code Doppler that is smaller than the carrier Doppler. In counterpart, it implies that more precision is required for the code NCO:

$$N_{NCO} = \log_2 \left(\frac{f_{L1}}{f_{CIA}} \frac{f_s}{ErrDoppler_{remmargin}} \right) = \log_2 \left(\frac{1575.42}{1.023} \frac{100 \times 10^6}{1.397 \times 10^{-3}} \right) = 47bits$$

A precision of 48 bits will be used from here forward as it is a multiple of 8.

III.2.1.9 Phase Update Rate

Because all NCO have a limited precision, the simulated SV position will tend to drift from its exact position as time passes. As the 1 mm/s specification has been verified, all NCO phase must be also updated (forced) at least once in every 10 s to comply with the 10 mm specification. A phase update rate of 1 Hz should be rather used to guarantee specifications compliance.

III.2.1.10 Number of Channels Implemented

Number of GNSS channels required:

- 12 GPS L1 channels
- 6 GPS L5 channels
- 6 Galileo L1 channels
- 6 Galileo E5a channels
- 6 Galileo E5b channels

A total of 36 GNSS channels is best targeted to allow any combination possible.

III.2.1.11 Development Board

The development board currently available for designing is a Lyrtech VHS-ADAC-V4. It is a cPCI board capable of transfers up to 20 MB/s and it is equipped with 64 MB of external DRAM memory and a Virtex IV SX55 -10 FPGA which main resources are presented here:

- 512 Xtreme DSP slices
- 320 Block RAM (offering up to 5.8 Mbits mem. space)
- 24 576 slices

III.2.1.12 Suggested General Channel Architecture

Each GNSS signal simulation channel should be built using the following blocks
(Refer to Schematic 2/3):

- (1x) Control register interface
- (1x) Async FIFO (NAV message)
- (3x) Sync FIFO (Software gain, Code rate & carrier frequency)
- (1x) Linear multi-task interpolator
- (4x) Buffers (Primary & secondary I&Q-codes)
- (2x) NCO (Code & carrier)
- (2x) LUT (Sin & cos)
- (2x) Variable gains
- (1x) Complex multiplier
- (3x) multipliers (sign inverters)
- (1x) adder
- (2x) multiplexer
- (2x) clock dividers

III.2.1.13 FPGA Resources Attribution

III.2.1.13.1 Control Register Interface

A state machine and about 162 flip-flops will be required by this module. Therefore, about 140 slices are estimated to be used.

III.2.1.13.2 NAV Message FIFO

Because Virtex IV Async FIFO primitives does not come in 16k x 1 bit format, a single 16k x 1 bit Block RAM (BRAM) will be used for this module, plus some overhead logic implemented as a control state-machine. About 30 slices are estimated to be used.

III.2.1.13.3 Software Gain FIFO

As for the NAV message FIFO, a single BRAM will be used in a 512 x 36 bits format. Because this FIFO is synchronous, little overhead logic should be implemented, reducing down to 20 the estimated slices to be used.

III.2.1.13.4 Carrier Frequency FIFO

Same as above.

III.2.1.13.5 Code Rate FIFO

Same as above, except that 2 BRAM of 512 x 36 bits format will be used.

III.2.1.13.6 Linear Multi-Task Interpolator

To reduce the total resources consumption, a single interpolator will process both carrier frequency and code rate in multi-task mode. Because the interpolation frequency is expected to be around 1 MHz and that the system clock is of 100 MHz, a state machine coupled to a single DSP Xtreme slice should have enough time to process all data. Thus 1 DSP Xtreme slice and about 100 regular slices are expected to be used.

III.2.1.13.7 I&Q-Codes Buffers

Very little overhead logic is required to load and manage these modules compared to a FIFO as these modules are simple buffers. Primary code buffers will each use a 16k x 1 bit wide BRAM for a total of 2 BRAM. Because of their small size (20 to 100 chips), secondary code buffer will be implemented using Distributed RAM blocks of format 16 x 1 bit double ports, for a total of 9 Distributed RAM blocks. Because each Distributed RAM block uses 1 slice, a total of 60 slices is estimated to be used by all I&Q-Codes buffers, including overhead logic.

III.2.1.13.8 NCO

Code and carrier NCO generally require the use of 2 adders and of 2 buffers. Because their bus widths are of 48 and 36 bits respectively, a total of about 180 slices is expected to be used by both NCO. However, an Xtreme DSP slices might be used in each NCO to reduce the number of required slices down to 90.

III.2.1.13.9 LUT (Sin & Cos)

Each carrier NCO requires a Sin and a Cos Look-Up-Table of 14 bits peak to peak by 15 bits of precision (addressing). By using a single time-multiplexed quarter wave-length LUT, it is thus possible to generate a sine and a cosine with the use of seven (7) 8k x 2 bits wide dual port BRAM. 10 slices are also expected to be use as over-head logic.

III.2.1.13.10 Variable Gains

A single Xtreme DSP slice will be used for both variable gains.

III.2.1.13.11 Complex Multiplier

A single 4x over-clocked Xtreme DSP slice should be used for this module.

III.2.1.13.12 Sign Inverters, Adder and Multiplexers

About 5 slices will be required to implement all those modules.

III.2.1.13.13 Clock Dividers

Because of a maximal Primary Code Clock (PCLK) to Secondary Code Clock (SCLK) ratio of 10 230, a simple 14 bits counter will be sufficient to generate the SCLK from the PCLK. In the same optic, a maximal ratio of 204 600 exists for Galileo E5a, thus a 20 bits counter will be enough to generate the Navigation Message Clock (MCLK) for the PCLK in any circumstances. A total of 20 slices is therefore expected for both clock dividers.

III.2.1.13.14 Total Estimated Resources

Module	BRAM	DSP Slices	Slices
Reg. Interface	-	-	140
Msg. FIFO	1	-	30
Gain FIFO	1	-	20
Code rate FIFO	2	-	20
Car. freq. FIFO	1	-	20

Interpolator	-	1	100
I&Q codes buf.	2	-	60
NCO	-	2	90
Sin & Cos LUT	7	-	10
Variable Gains	-	1	-
Complex Mult.	-	1	-
Clock dividers	-	-	20
Others	-	-	25
Total	14	5	535
Total for 36 channels	504	180	19 260
% Used	158%	35%	78%

The table above clearly indicates that 146% of the total BRAM primitives available in the current FPGA are needed to implement 36 GNSS channels. As a solution, the FPGA could be replaced by a Virtex IV FX140 which contains up to 552 BRAMs, 192 Xtreme DSP Slices and 63 168 slices.

Another solution would be to replace all synchronous FIFO (Gain, code rate & carrier frequency FIFO) by the external RAM. Because synchronous FIFO access is relatively slow (10 ms), a memory arbiter and a series of requesters could be designed to allow every GNSS channels to retrieve data directly from external RAM, thus savings up to 4 BRAMs per channel. The Sine and Cosine LUT should also be shared among 2 GNSS channels at a time by over-clocking the BRAMs by 2x, thus saving 3.5 more BRAMs per channel. Both optimizations could allow a consumption of only 6.5 BRAM/channel for a total of 234 BRAMs (73% of total FPGA resources).

III.2.1.14 Chosen General Channel Architecture

The second solution proposed above will be chosen for the IF core design (Refer to Schematic 3/3).

III.2.1.15 Data transfer rate

To comply with the specifications, data has to be transferred from the software core to the IF core at a minimal rate of 100 Hz. As seen above, the data that has to be transferred periodically regroup the carrier frequency (36 bits), the code rate (48 bits) and the channel gain value (16 bits). Therefore, 100 bits (13 bytes) has to be transferred per channel, for a total transfer rate of:

$$T_{kB/s} = \frac{D_B \times f_t \times N_{channel}}{1024} = \frac{13 \times 100 \times 36}{1024} = 45.7 kB/s$$

Considering Lyrtech's board maximal transfer rate of 20 MB/s, $T_{kB/s}$ represents only 0.2% of this maximal transfer rate. There is thus no need to study furthermore this aspect.

III.2.2 Top level Core

At the top level core the signal is then potentially summed with up to 17 other signals. The summation process will lead to a maximum of 24.6 dB more power. This is an addition 4 bits (or 24 dB). The missing 0.6 dB should not be a problem because the dynamic full 22 bits where not fully utilized and the worst case should only happen if all 18 signals are at maximum power (unlikely situation) and all add up constructively (again unlikely situation). The total number of bits at the output of the first adder is thus 26 bits.

The next adder combines the interferer with the Lx/Ex signal. The number of bits does not need to be increased as the interferer power is 3 bits lower, or 18 dB, than the Lx/Ex signal.

After all signals are added together the output is rounded to 16 bits thus introducing quantization noise on the overall signal to the rate of $96dB + 10 \log(fs) = -176dBc/Hz$.

The typical signal is at this point $20 + 24.6 = 44.6$ dB lower than the peak power.

This truncated signal is then transmitted to the DAC where it will be interpolated to 400 MHz and modulated with a 50 MHz carrier that is generated synchronously with the sampling clock and will therefore not add any phase noise other than that of the reference clock.

This signal will then be converted by the DAC. The resulting noise should be larger than the -176 dBc/Hz due to DAC imperfections. On the other hand the noise will be spread over a 4 times larger bandwidth leading potentially to a 6 dB reduction in the noise floor.

The specifications call for a 50 dB/Hz of C/No. The C/No at the DAC output will be considerably larger than that. The attenuation in the RF chain will bring the C/No within the expected range.

The expected power output at the DAC is expected to be around 0 dBm. The DAC itself can generate between -7 dBm and 13 dBm of output power.

III.2.3 Interferer Core

III.2.3.1 Soft interferer

The soft interferer core uses the same circuitry as the SV core. It has 15 bits to generate the sin wave and 50 dB of dynamic range on the gain control requiring 8 bits (for 48 dB). This leads to a quantization requirement of 23 bits at the output of the interferer core.

III.2.3.2 Hard interferer (optional)

An optional interferer could be inserted in the RF board. It use a 2.7 GHz DDS-Based AgileRF Synthesizer which drives a large band external VCO (1100 MHz to 1600 MHz). The DDS is designed to provide fast frequency hopping and tuning resolution less than 1 Hz (48-bit frequency tuning word). Information is loaded into the DDS via a serial I/O port that has a device write-speed of 25 Mb/s. The DDS block also supports a user-defined linear sweep mode of operation.

III.2.4 Gain Control

III.2.4.1 Soft interferer

The control should add 50 dB of dynamic range to its input. This is an additional 8 bits or 48 dB. A control gain of 1 represents the minimum output level. A control gain of 256 is the maximum output level. The value 0 is valid and represents no signal.

No minimal resolution is specified for the interferer but a minimal resolution of 0.5 dB will be used. This implies that the gain control shall have additional bits after the decimal point. An additional 5 bits after the decimal point provides a maximum step size 0.26 dB.

The command of the AGC thus requires 13 bits where 5 bits are located after the decimal point. This is denoted by [8, 5].

III.2.4.2 Hard interferer (optional)

The Hard interferer amplitude gain is to be controlled by a high performance, voltage-controlled variable gain amplifier/attenuator for use in applications with frequencies up to 3 GHz. The gain control pin with a 20 mV/dB scaling factor is controlled by 16 bits DAC to ensure a high tuning precision and fast updating rate. The Dynamic range of the VGA is about 50 dB which is assumed to be sufficient for this application.

III.2.5 Tick Clock Core

The Tick clock core uses the same basic clock dividing technology as the SV core. A modulo N integrator where N represents a full cycle. The MSB serves as the derived clock. Since this signal is to be used as an external reference it will drive the MSB of the DAC with all other bits set maximize the dynamic range. The output will thus be a square wave of 50% duty cycle and 0 dBm power (subject to DAC output circuitry implementation).

The DAC will be set to interpolate to 400 MHz with no modulation.

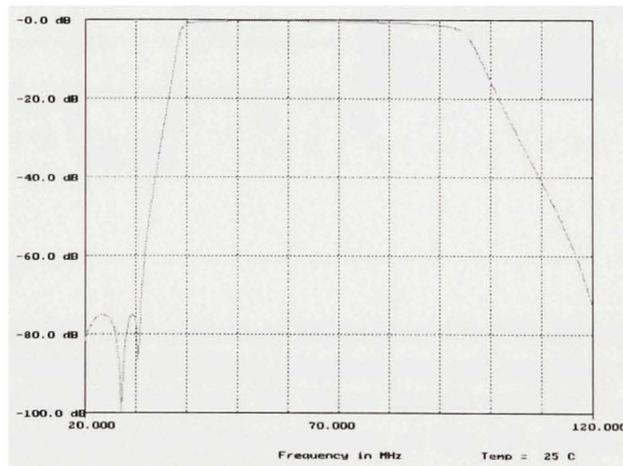
III.2.6 RF board inventory

- Inputs:
 - 3 SMA connector for the IF signal (L1, L5/E5a, E5b)
 - 1 SMA connector for external clock input
- Outputs:
 - 1 SMA connector for the RF
 - 1 SMA connector for the 100 MHz clock
 - 1 SMA connector for the 10 MHz clock
 - 3 SMA connectors for the LO signals (L1, L5/E5a, E5b)
- 1 reference clock at 100 MHz
- 1 clock distributor
- 3 IF band-pass filters
- 3 RF band-pass filters
- 3 cleaning low-pass filter for VCOs output
- 3 Image Reject Mixers
- 3 VCOs for the LO
- 3 frequency synthesizer for the LO
- 3 variable gain amplifier (VGA) for the calibration of the RF signals
- 1 combiner 4 ways
- 1 RF attenuator
- 1 DAC with 8 outputs to drive the VGAs
- 6 couplers to probe the signals
- More than 32 test points
- 1 power detector for the calibration of the RF signals
- 1 ADC for power detector signal digitalization
- 1 SP4T switch for power detector input commutation
- Optional:
 - 1 DDS for the interfere
 - 1 L band VCO
 - 1 L band filter
 - 1 VGA

III.2.7 Analog Filters

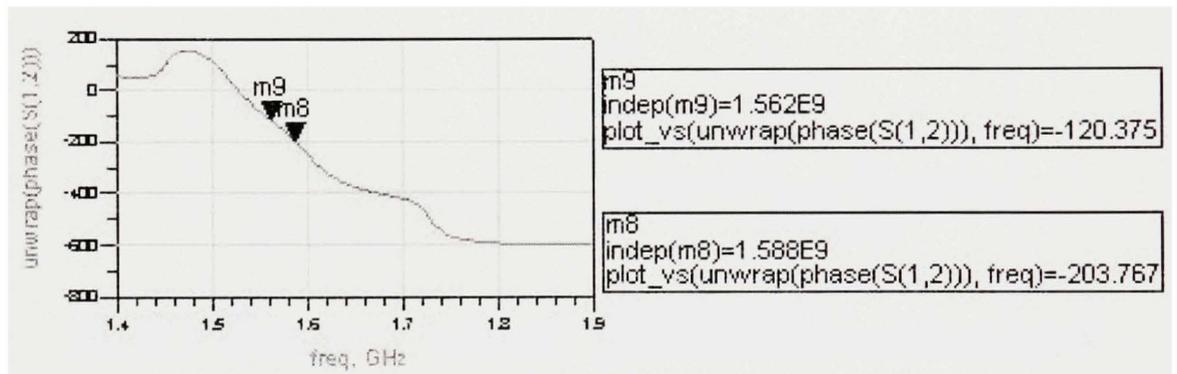
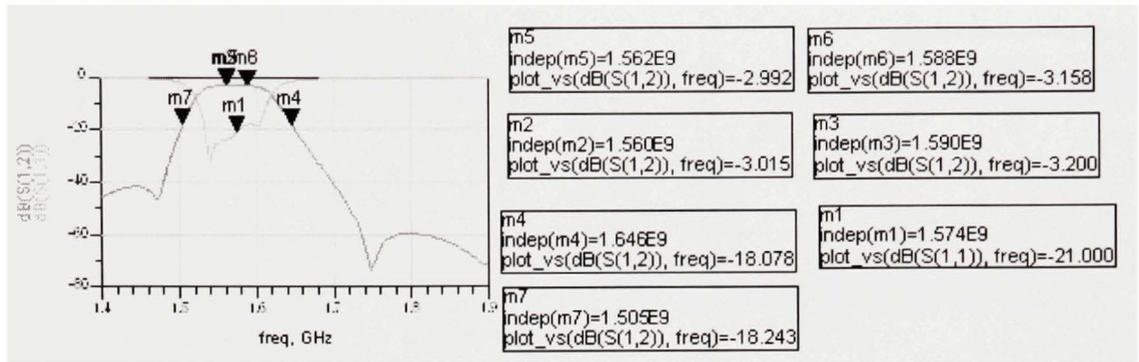
III.2.7.1 IF filters

- Type Lumped Band pass
- Center frequency 70 MHz
- Pass Band 24 MHz min.
- Pass Band loss < 1.0 dB
- Pass Band Return loss >15 dB
- Pass Band Linear Phase < 4°
- Rejection > 60 dB @ 20 & 120 MHz
- Rejection > 60 dB @ 120 & 1000 MHz
- Operating Temp-20°C to +70°C
- Package 50.80 x 16 x 12.7 millimeters S11M Leadless SMD Package



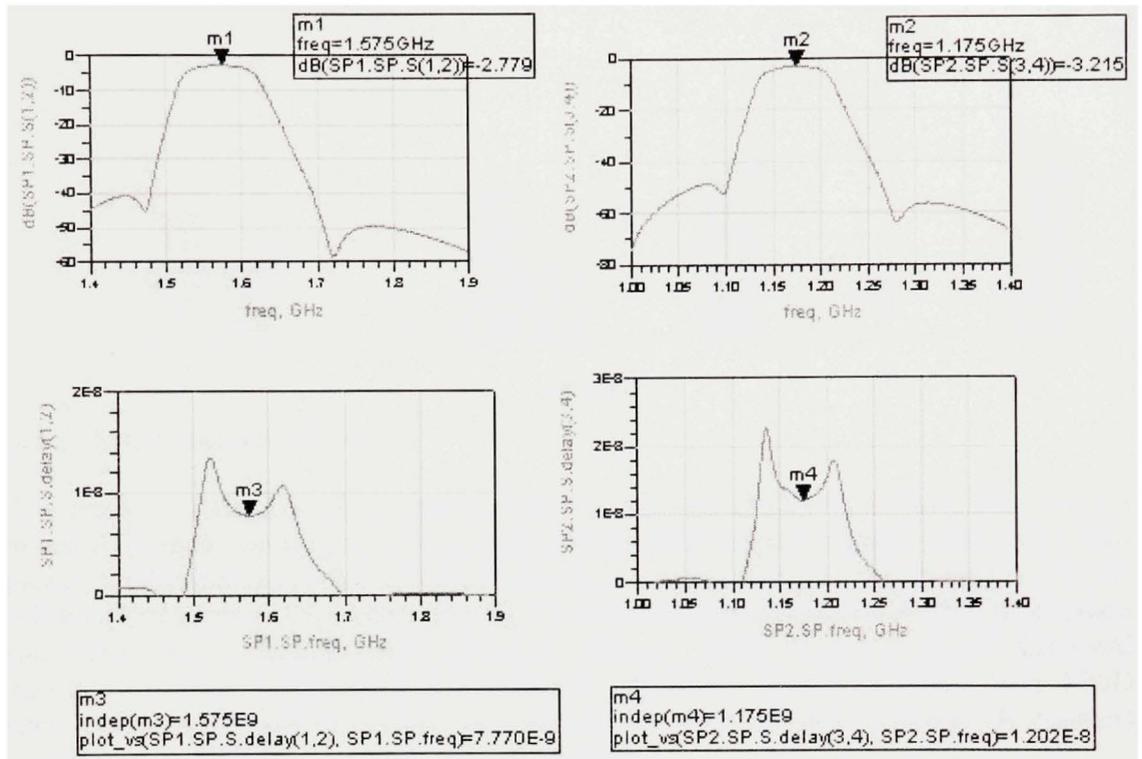
III.2.7.2 RF filters

- Type Micro-Strip Band pass
- Center frequencies 1175,45 MHz 1207,14 MHz and 1575,42 MHz
- Pass Band 30 MHz min.
- Pass Band loss < 3 dB
- Pass Band Return loss >20 dB
- Pass Band Linear Phase (see the figure below)
- Rejection > 40 dB @ 150 MHz from the center frequency
- Operating Temp-20°C to +70°C
- Package 30 x 22 x 12



III.2.7.3 Group delay

In whole RF systems the main source of delay should be the filters and the trace path of each RF channel. Assuming the IF filters identical we can consider the differential group delay between them negligible. Only the RF filters should provide a significant differential group delay. The figure below shows the response of the RF filters. L1 filter response is represented on the top at the left and L5 filter response is on the top at the right. The respective group delay of each filter is represented on below the corresponding frequency response. The simulation shows that the differential group delay between each carrier L5-L1 (worst case) is about 4 ns.

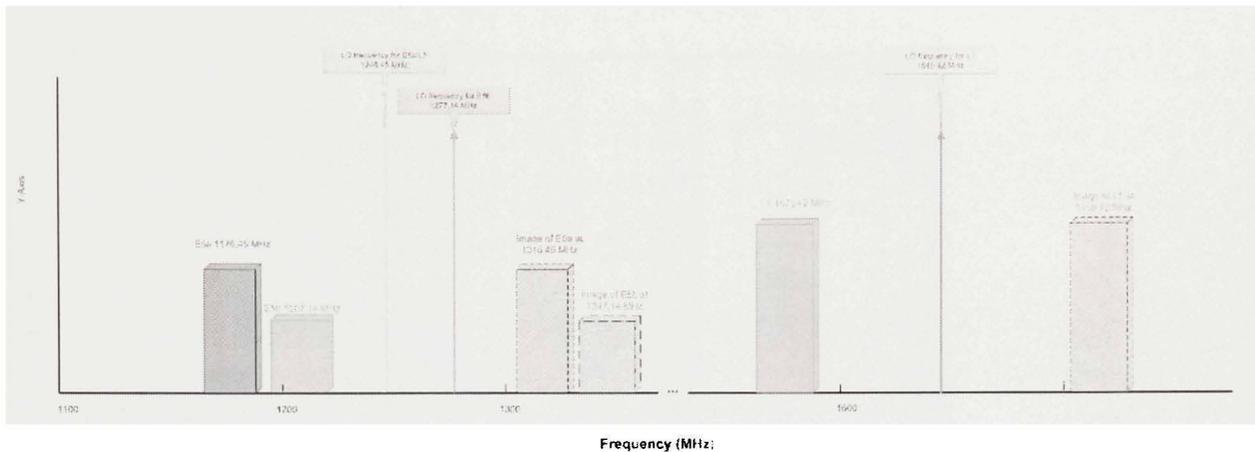


III.2.7.4 Frequency plan

The LO frequencies have been chosen to obtain the spectrum displayed on the figure below. By considering the IF at 70MHz, LO frequencies are:

- 1645,42 MHz for L1
- 1246,45 MHz for L5/E5a
- 1277.14 MHz for E5a

The figure below shows also the images frequencies outputted by the mixers. An adequate filtering and image rejection mixers will substantially reduce the images influence through the L band.



III.2.8 Local oscillators core

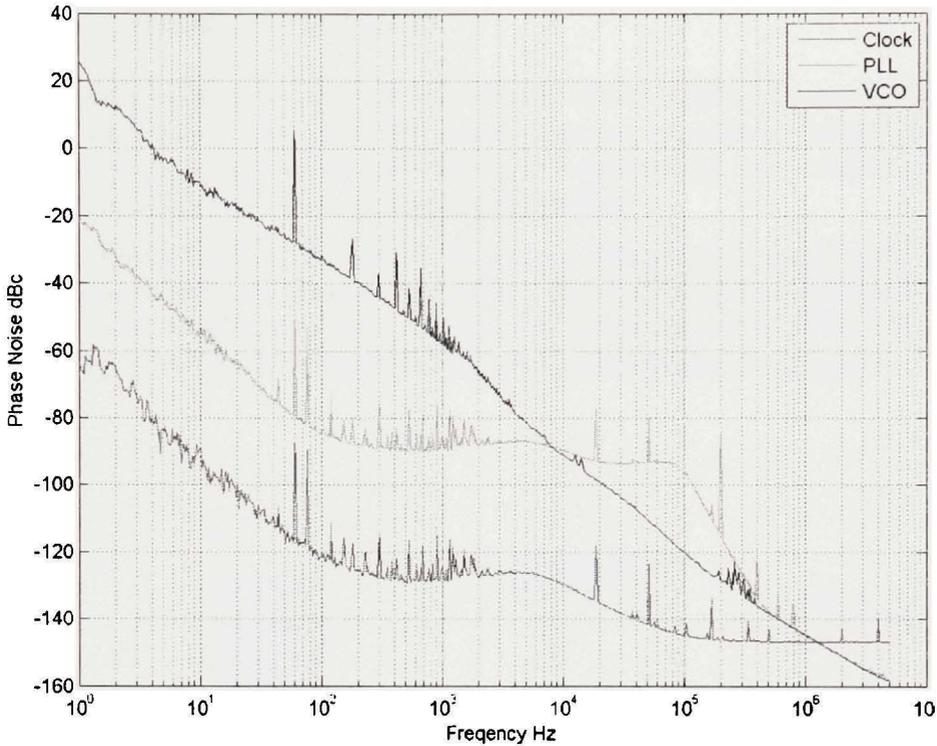
The local oscillator core is complete phase-locked loop (PLL) implemented by using a fractional frequency synthesizer with an external loop filter and a low phase noise VCO. This frequency synthesizer is suitable to implement local oscillators in the upconversion sections of the RF board. It consists of a low noise digital phase frequency detector (PFD), a precision charge pump, and a programmable reference divider. There is a Σ - Δ based fractional interpolator to allow programmable fractional-N division. The INT, FRAC, and MOD registers define an overall N divider. In addition, the 4-bit reference counter (R counter) allows selectable REF_{IN} frequencies at the PFD input.

If F_{RF} is the LO frequency, then:

$$F_{RF} = [\text{INT} + (\text{FRAC}/\text{MOD})] \times [F_{REF}/R]$$

F_{REF} is provided by the outputs of the Clock distributor driven by the OCXO at 100 MHz (see section III.1.2). The values of the registers for each LO are set as following:

- For L1: INT=164 FRAC=271 MOD=500 R=10
- For L5/E5a: INT=124 FRAC=129 MOD=200 R=10
- For E5b: INT=127 FRAC=357 MOD=500 R=10

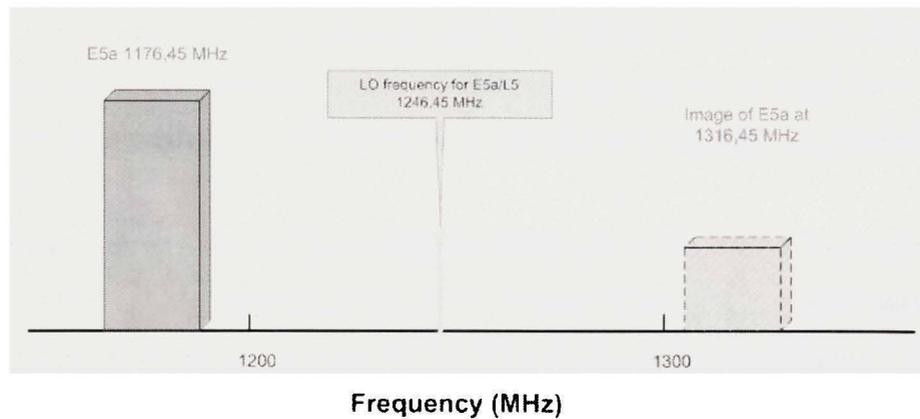


The phase noise of the LO should follow the OCXO reference. The 250 fs of jitter introduced by the clock distributor have insignificant influence on the LO phase noise. If the reference is converted by the frequency synthesizer to 1.6 GHz (worst case) then the phase noise increase by 1.6GHz/100MHz. Form the OCXO phase noise characteristics (refer to the section III.1.1.1) we deduce the phase noise of the LO is about 0.01 rad RMS. The measurements displayed on the figure above show how the phase noise of the VCO follows the reference when using the fractional frequency synthesizer.

III.2.9 The image reject mixer

In comparison to conventional mixers, IRMs achieve image-rejection through phase cancellation, not filtering, so the frequency spacing between the image and desired inputs can be negligible. Therefore the upconversion is operated by using this kind of mixer to cancel the image as we can observe it on the figure below.

To design the IRM we use RF quadrature modulator with excellent phase accuracy and amplitude balance. IF input is buffered by an ultra low phase unbalance 90° hybrid coupler which split to the modulator in phase and in quadrature IF. The image rejection is then expected to be around -40 dBc and the carrier feedthrough is -38 dBm.



III.3 Annex A

Digital Filter Terminology

From: R. G. Lyons, UNDERSTANDING DIGITAL SIGNAL PROCESSING, (Appendix F - pages 494-505). © 1997 by Addison Wesley Longman Inc., Reproduced by permission of Addison Wesley Longman. All rights reserved.

The first step in becoming familiar with digital filters is to learn to speak the language used in the filter business. Fortunately, the vocabulary of digital filters corresponds rather well with the mother tongue used for continuous (analog) filters—so we don't have to unlearn anything that we already know. This material is an introduction to the terminology of digital filters.

Allpass filter - an IIR filter whose magnitude response is unity over its entire frequency range, but whose phase response is variable. Allpass filters are typically appended in a cascade arrangement following a standard IIR filter, $H_1(z)$, as shown in Figure F-1.

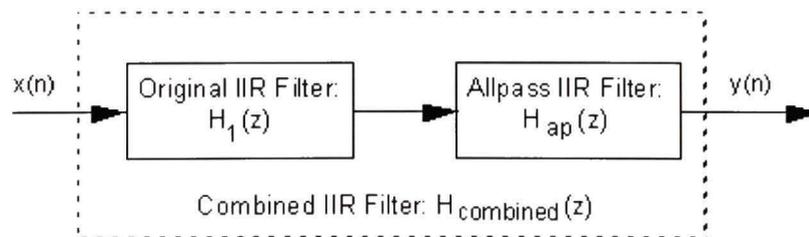


Figure F-1 Typical use of an allpass filter.

An allpass filter, $H_{ap}(z)$, can be designed such that its phase response compensates for, or *equalizes*, the nonlinear phase response of an original IIR filter. [1]-[3] Thus the phase response of the combined filter, $H_{combined}(z)$, is more linear than the original $H_1(z)$, and this is particularly desirable in communications systems. In this context, an allpass filter is sometimes called a *phase equalizer*.

Attenuation - an amplitude loss, usually measured in dB, incurred by a signal after passing through a digital filter. Filter attenuation is the ratio, at a given frequency, of the signal

amplitude at the output of the filter over the signal amplitude at the input of the filter, defined as

$$\text{attenuation} = 20 \cdot \log_{10} \left(\frac{a_{\text{out}}}{a_{\text{in}}} \right) \text{ dB} . \quad (\text{F-1})$$

As is usually the case, for a given frequency, the output amplitude of the filter is smaller than the input amplitude, making the ratio in Eq. (F-1) less than one and the attenuation is a negative number.

Band Reject Filter - a filter that rejects (attenuates) one frequency band and passes both a lower and a higher frequency band. Figure F-2(a) depicts the frequency response of an ideal band reject filter. This filter type is sometimes called a *notch filter*.

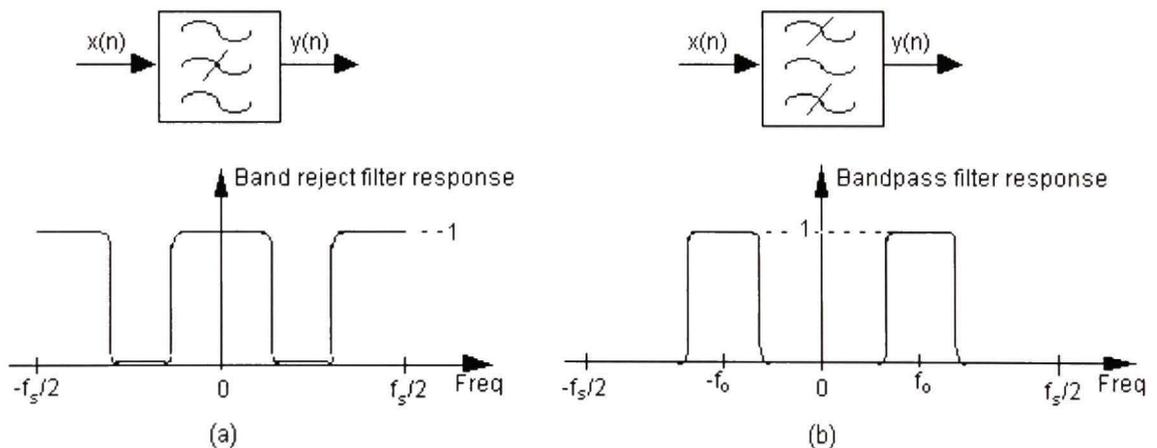


Figure F-2 Filter symbols and frequency responses: (a) Band reject filter; (b) Bandpass filter.

Bandpass Filter - A filter, as shown in Figure F-2(b), that passes one frequency band and attenuates frequencies above and below that band.

Bandwidth - few terms in signal processing have more definitions than this one. We'll define bandwidth as the frequency width of the passband of a filter. For a low-pass filter, the bandwidth is equal to the cutoff frequency. For a bandpass filter, the bandwidth is typically defined as the frequency difference between the upper and lower -3 dB points.

Bessel Function - a mathematical function used to produce the most linear phase response of all IIR filters with no consideration of the frequency magnitude response. Specifically, filter designs based on Bessel functions have maximally constant group delay. (Modeling Bessel

functions with computer software, not recommended for the faint-hearted, generally results in a significant lesson in humility.)

Butterworth Function - a mathematical function used to produce maximally flat filter magnitude responses with no consideration of phase linearity or group delay variations. Filter designs based on a Butterworth function have no amplitude ripple in either the passband or the stopband. Unfortunately, for a given filter order, Butterworth designs have the widest transition region of the most popular filter design functions.

Cascaded filters - the implementation of a filtering *system* where multiple individual filters are connected in series. That is, the output of one filter drives the input of the following filter as illustrated in Figure F-1.

Center Frequency (f_0) - the frequency lying at the midpoint of a bandpass filter. Figure F-2(b) shows the f_0 center frequency of a bandpass filter.

Chebyshev Function - a mathematical function used to produce passband, or stopband, ripples constrained within fixed bounds. There are families of Chebyshev functions based on the amount of ripple such as 1 dB, 2 dB, and 3 dB of ripple. Chebyshev filters can be designed to have a frequency response with ripples in the passband and flat passbands (Chebyshev Type I), or flat passbands and ripples in the stopband (Chebyshev Type II). Chebyshev filters cannot have ripples in both the passband and the stopband. Digital filters based upon Chebyshev functions have steeper transition region roll-off but more nonlinear phase response characteristics than, say, Butterworth filters.

Coefficients - See Filter Coefficients.

Cutoff Frequency - the upper passband frequency for low-pass filters, and the lower passband frequency for highpass filters. A cutoff frequency is determined by the -3 dB point of a filter magnitude response relative to a peak passband value. Figure F-3 illustrates the f_c cutoff frequency of a low-pass filter.

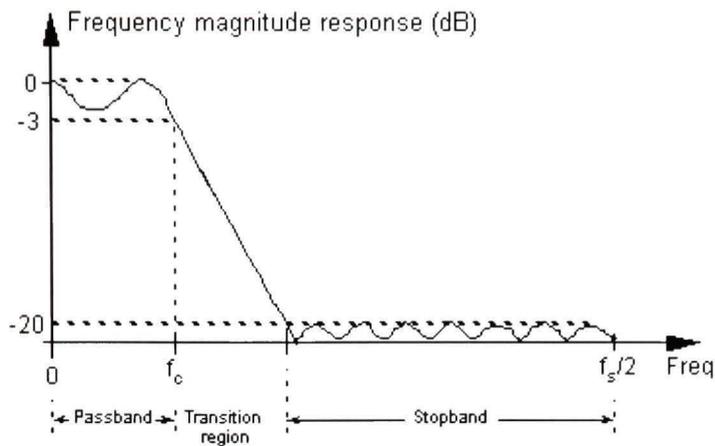


Figure F-3 A low-pass digital filter frequency response. The stopband relative amplitude is -20 dB.

Decibels (dB) - a logarithmic unit of attenuation, or gain, used to express the relative voltage or power between two signals. For filters we use decibels to indicate cutoff frequencies (-3 dB) and stopband signal levels (-20 dB) as illustrated in Figure F-3.

Decimation filter - a low-pass digital FIR filter whose output sample rate is less than the filter's input sample rate. To avoid aliasing problems the output sample rate must not violate the Nyquist criteria.

Digital filter- computational process, or algorithm, transforming a discrete sequence of numbers (the input) into another discrete sequence of numbers (the output) having a modified frequency domain spectrum. Digital filtering can be in the form of a software routine operating on data stored in computer memory or can be implemented with dedicated digital hardware.

Elliptic Function- a mathematical function used to produce the sharpest roll-off for a given number of filter taps. However, filters designed using elliptic functions, also called *Cauer filters*, have the poorest phase linearity of the most common IIR filter design functions. The ripple in the passband and stopband are equal with elliptic filters.

Envelope Delay - See Group Delay.

Filter Coefficients - the set of constants, also called *tap weights*, used to multiply against delayed signal sample values within a digital filter structure. Digital filter design is an exercise in determining the filter coefficients that will yield the desired filter frequency

response. For an FIR filter, the filter coefficients are, by definition, the impulse response of the filter.

Filter Order - a number describing the highest exponent in the numerator or denominator of the z-domain transfer function of a digital filter. For FIR filters, there is no denominator in the transfer function and the filter order is merely the number of taps used in the filter structure. For IIR filters, the filter order is equal to the number of delay elements in the filter structure. Generally, the larger the filter order, the better the frequency magnitude response performance of the filter.

Finite Impulse Response (FIR) Filter - defines a class of digital filters that has only zeros on the z-plane. The key implications of this are that FIR filters are always stable, and have linear phase responses (as long as the filter's coefficients are symmetrical). For a given filter order, FIR filters have a much more gradual transition region roll-off than digital IIR filters.

Frequency Magnitude Response - a frequency domain description of how a filter interacts with input signals. The frequency magnitude response in Figure F-3 is a curve of filter attenuation (in dB) vs frequency. Associated with a filter's magnitude response is a phase response.

Group Delay - defined as the derivative of a filter's phase with respect to frequency, $G = -\Delta \phi / \Delta f$, or the slope of a filter's $H\phi(m)$ phase response curve. The concept of group delay deserves additional explanation beyond a simple definition. For an ideal filter, the phase will be linear and the group delay would be constant. Group delay, whose unit of measure is time in seconds, can also be thought of as the propagation time delay of the envelope of an amplitude modulated signal as it passes through a digital filter. (In this context, group delay is often called *envelope delay*.) Group delay distortion occurs when signals at different frequencies take different amounts of time to pass through a filter. If the group delay is denoted G , then the relationship between group delay, a $\Delta \phi$ increment of phase, and a Δf increment of frequency is

$$G = \frac{-\Delta \phi_{\text{degrees}} / 360}{\Delta f} = \frac{-\Delta \phi_{\text{radians}} / 2\pi}{\Delta f} \text{ seconds.} \quad (\text{F-2})$$

If we know a linear phase filter's phase shift ($\Delta \phi$) in degrees/Hz, or radians/Hz, we can determine the group delay in seconds using

$$G \cdot \Delta f = G \cdot 1 = G = \frac{-\Delta \phi_{\text{degrees / Hz}}}{360} = \frac{-\Delta \phi_{\text{radians / Hz}}}{2\pi} \text{ seconds.} \quad (\text{F-3})$$

To demonstrate Eq. (F-3) and illustrate the effect of a nonlinear phase filter, let's assume that we've digitized a continuous waveform comprising four frequency components defined by

$$x(t) = \sin(2\pi \cdot 1 \cdot t) + \sin(2\pi \cdot 3 \cdot t)/3 + \sin(2\pi \cdot 5 \cdot t)/5 + \sin(2\pi \cdot 7 \cdot t)/7. \quad (\text{F-4})$$

That is, the $x(t)$ input comprises the sum of a 1 Hz, 3 Hz, 5 Hz, and 7 Hz sinewaves and its discrete representation is shown in Figure F-4(a). If we applied the discrete sequence representing $x(t)$ to the input of an ideal 4-tap linear phase low-pass digital FIR filter with a cutoff frequency of greater than 7 Hz, and whose phase shift is -0.25 radians/Hz, the filter's output sequence would be that shown in Figure F-4(b).

Because the filter's phase shift is -0.25 radians/Hz, Eq. (F-3) tells us that the filter's constant group delay in seconds is

$$G = \frac{-\Delta \phi_{\text{radians / Hz}}}{2\pi} = \frac{0.25}{2\pi} = 0.04 \text{ seconds.} \quad (\text{F-5})$$

With a constant group delay of 0.04 seconds, the 1 Hz input sinewave is delayed at the filter output by 0.25 radians, the 3 Hz sinewave is delayed by 0.75 radians, the 5 Hz sinewave by 1.25 radians, and the 7 Hz sinewave by 1.75 radians. Notice how a linear phase (relative to frequency) filter results in an output that's merely a time shifted version of the input as seen in Figure F-4(b). The amount of time shift is the group delay of 0.04 seconds. Figure F-4(c), on the other hand, shows the distorted output waveform if the filter's phase was nonlinear, for whatever reason, such that the phase shift was 3.5 radians instead of the ideal 1.75 radians at 7 Hz. Notice the distortion of the beginning of the output waveform envelope in Figure F-4(c) compared to Figure F-4(b).

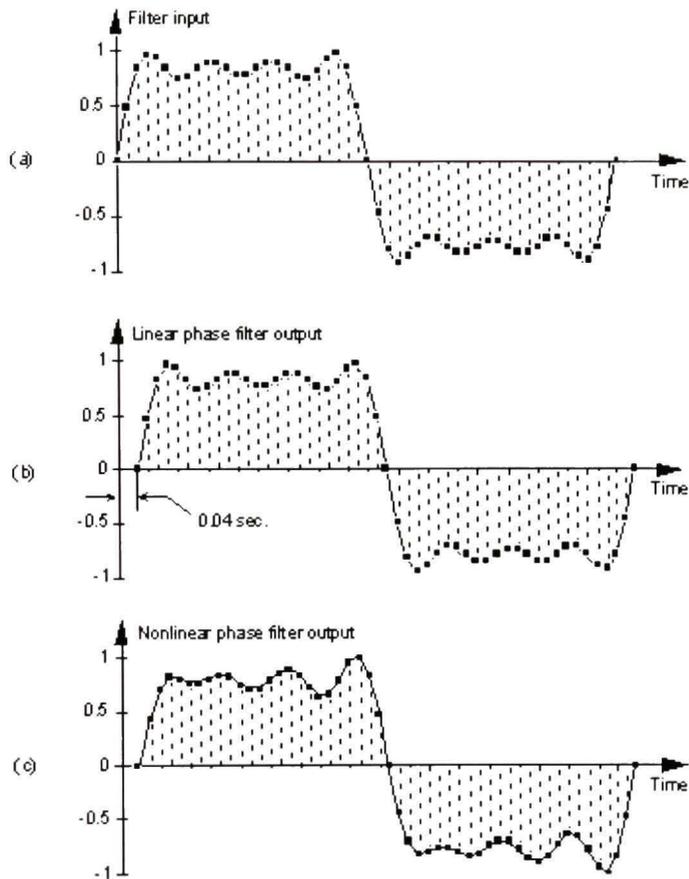


Figure F-4 Filter time domain response examples: (a) Filter input sequence; (b) Linear-phase filter output sequence that's a time shifted, by 0.04 seconds, duplicate of the input sequence; (c) Distorted output sequence due to a filter with nonlinear phase.

The point here is that if the desired information is contained in the envelope of a signal that we're passing through a filter, we'd like that filter's passband phase to be as linear as possible with respect to frequency. In other words, we'd prefer the filter's group delay to vary as little as possible in the passband.

Half-band filter - a type of FIR filter where the transition region is centered at one quarter of the sampling rate, or $f_s/4$. Specifically, the end of the passband and the beginning of the stopband are equally spaced on either side of $f_s/4$. Half-band filters are often used in decimation filtering because (almost) half their time domain coefficients are zero. This

means, for example, you can achieve the performance of an M -tap FIR filter while only paying the computational price of $(M+1)/2 + 1$ multiplications per filter output sample.

Highpass Filter - a filter that passes high frequencies and attenuates low frequencies as shown in Figure F-5(a).

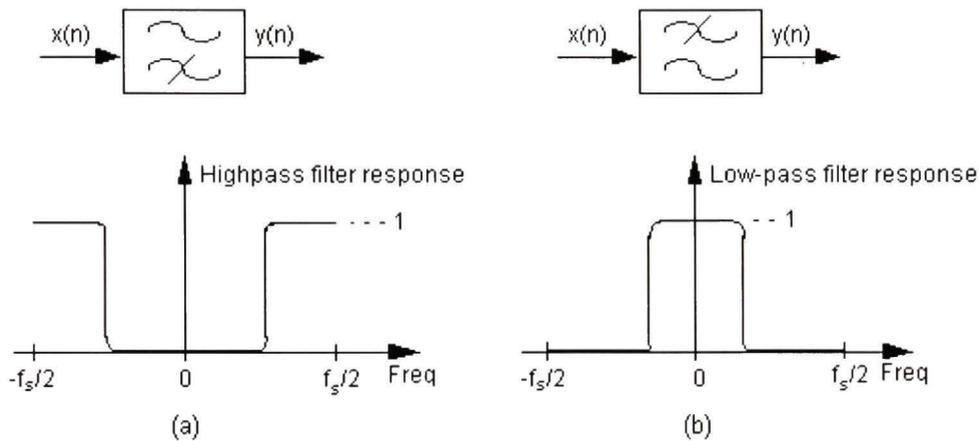


Figure F-5 Filter symbols and frequency responses:

(a) Highpass filter; (b) Low-pass filter.

We've all experienced a kind of highpass filtering in our living rooms. Notice what happens when we turn up the treble control (or turn down the bass control) on our home stereo systems. The audio amplifier's normally flat frequency response changes to a kind of analog highpass filter giving us that sharp and *tinny* sound as the high frequency components of the music are being accentuated.

Impulse Response - a digital filter's time domain output sequence when the input is a single unity-valued sample (an impulse) preceded and followed by zero-valued samples. Using perhaps the most powerful principle in signal processing, we can say that a linear digital filter's frequency domain response can be calculated by taking the discrete Fourier transform of the filter's time domain impulse response. [4]

Infinite Impulse Response (IIR) Filter - defines a class of digital filters that may have both zeros and poles on the z -plane. As such, IIR filters are not guaranteed to be stable and almost always have nonlinear phase responses. For a given filter order (number of IIR feedback taps), IIR filters have a much steeper transition region roll-off than FIR filters.

Linear Phase Filter- a filter that exhibits a constant change in output phase angle as a function of frequency. The resultant filter phase plot vs frequency is a straight line. As such, a linear phase filter's group delay is a constant. In order to preserve the integrity of their information-carrying signals, linear phase is an important criteria for filters used in communication systems.

Low-pass Filter- a filter that passes low frequencies and attenuates high frequencies as shown in Figure F-5(b). By way of example, we experience low-pass filtering when we turn up the bass control (or turn down the treble control) on our home stereo systems giving us that dull, muffled, sound as the low frequency components of the music are being intensified.

Notch Filter- See Band Reject Filter.

Passband - that frequency range over which a filter passes signal energy. Usually defined as the frequency range where the filter's frequency response is equal to or greater than -3 dB, as depicted in Figure F-3.

Passband Ripple - fluctuations, or variations, in the frequency magnitude response within the passband of a filter. Passband ripple, measured in dB, is illustrated in Figure F-6. (See Ripple.)

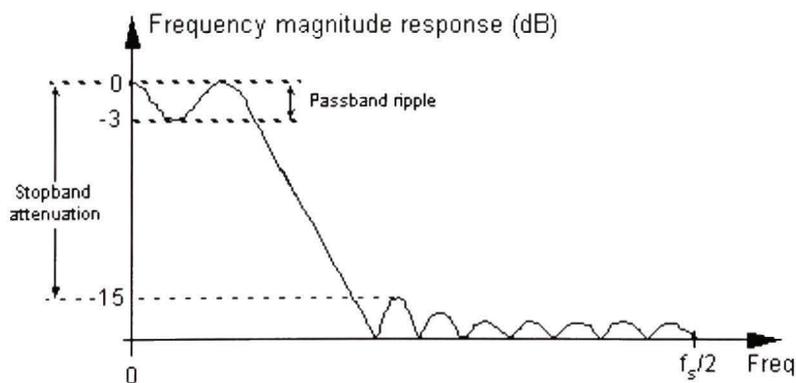


Figure F-6 Low-pass digital filter frequency response showing passband ripple and stopband attenuation.

Phase Response - the difference in phase, at a particular frequency, between an input sinewave and the filter's output sinewave at that frequency. The phase response, sometimes called *phase delay*, is usually depicted by a curve showing the filter's phase shift vs frequency.

Phase Wrapping - an artifact of arctangent software routines, used to calculate phase angles, that causes apparent phase discontinuities. When a true phase angle is in the range of -180° to -360° , some software routines automatically convert those angles to their equivalent positive angles in the range of 0° to $+180^\circ$.

Quadrature Filter- a dual-path digital filter operating on complex signal sequence, $x(n)$, as shown in Figure F-7. One filter operates on the in-phase $i(n)$ data while the other filter processes the quadrature-phase $q(n)$ signal data. Quadrature filtering is normally performed using low-pass filters on a complex sequence, $i(n) + jq(n)$, whose spectrum has been translated down so that it's centered about 0 Hz.

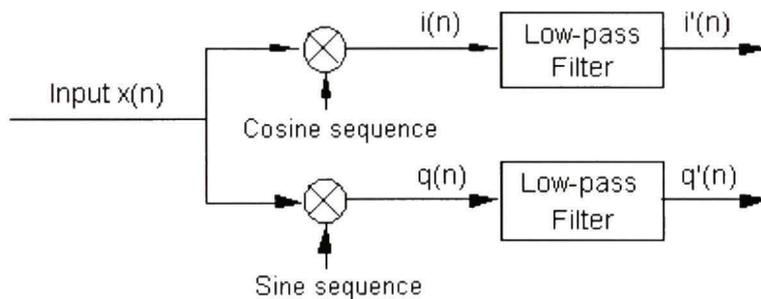


Figure F-7 Two low-pass filters used to implement quadrature filtering.

Relative Attenuation - attenuation measured relative to the largest magnitude value. The largest signal level (minimum attenuation) is typically assigned the reference level of 0 dB, as depicted in Figures F-3 and F-6, making all other magnitude points on a frequency response curve have negative dB values.

Ripple - has nothing, whatsoever, to do with an inexpensive alcoholic beverage. Ripple refers to fluctuations (measured in dB) in the passband, or stopband, of a filter's frequency magnitude response curve. Elliptic and Chebyshev-based filters have equiripple characteristics in that their ripple is constant across their passbands. Bessel and Butterworth derived filters have no ripple in their passband responses. Ripples in the stopband response are sometimes called *out-of-band ripple*.

Rolloff - a term used to describe the steepness, or slope, of the filter response in the transition region from the passband to the stopband. A particular digital filter may be said to have a rolloff of 12 dB/octave—meaning that the first octave of a frequency f_0 , or $2f_0$, would be

attenuated by 12 dB more than the filter's attenuation at f_0 . The second octave, $4f_0$, would be attenuated by 24 dB more, and so on.

Shape Factor - a term used to quantify the steepness of a filter's rolloff. Shape factor is normally defined as the ratio of a filter's the passband width plus the transition region width over the passband width. The smaller the shape factor value, the steeper the filter's roll-off. For an ideal filter with a transition region of zero width, the shape factor is unity. The term shape factor has long been used by RF folk to describe analog filters where they use the ratio of a filter's 60 dB bandwidth over its 3 dB bandwidth.

Stopband - that band of frequencies attenuated by a digital filter. Figure F-3 shows the stopband of a low-pass filter. Although the stopband attenuation in Figure F-3 is -20 dB, not all filters have stopband lobes of equal amplitude. Figure F-6 shows that stopband attenuation is measured between the peak passband amplitude and the largest stopband lobe amplitude.

Structure - a highfalutin term used by DSP "professionals" referring to the block diagram showing how a digital filter is implemented. A recursive filter structure is one in which feedback takes place and past filter output samples are used, along with past input samples, in the calculation of the present filter output. IIR filters are almost always implemented with recursive filter structures. A nonrecursive filter structure is one in which only past input samples are used in the calculation of the present filter output. FIR filters are almost always implemented with nonrecursive filter structures.

Tap Weights - See Filter Coefficients

Tchebyshev Function - See Chebyshev.

Transfer Function - a mathematical expression of the ratio of the output of a digital filter over the input of the filter. Given the transfer function we can determine the filter's frequency magnitude and phase responses.

Transition Region - the frequency range between the passband and the stopband of a digital filter. Figure F-3 illustrates the transition region of a low-pass filter. Transition region is sometimes called the *transition band*.

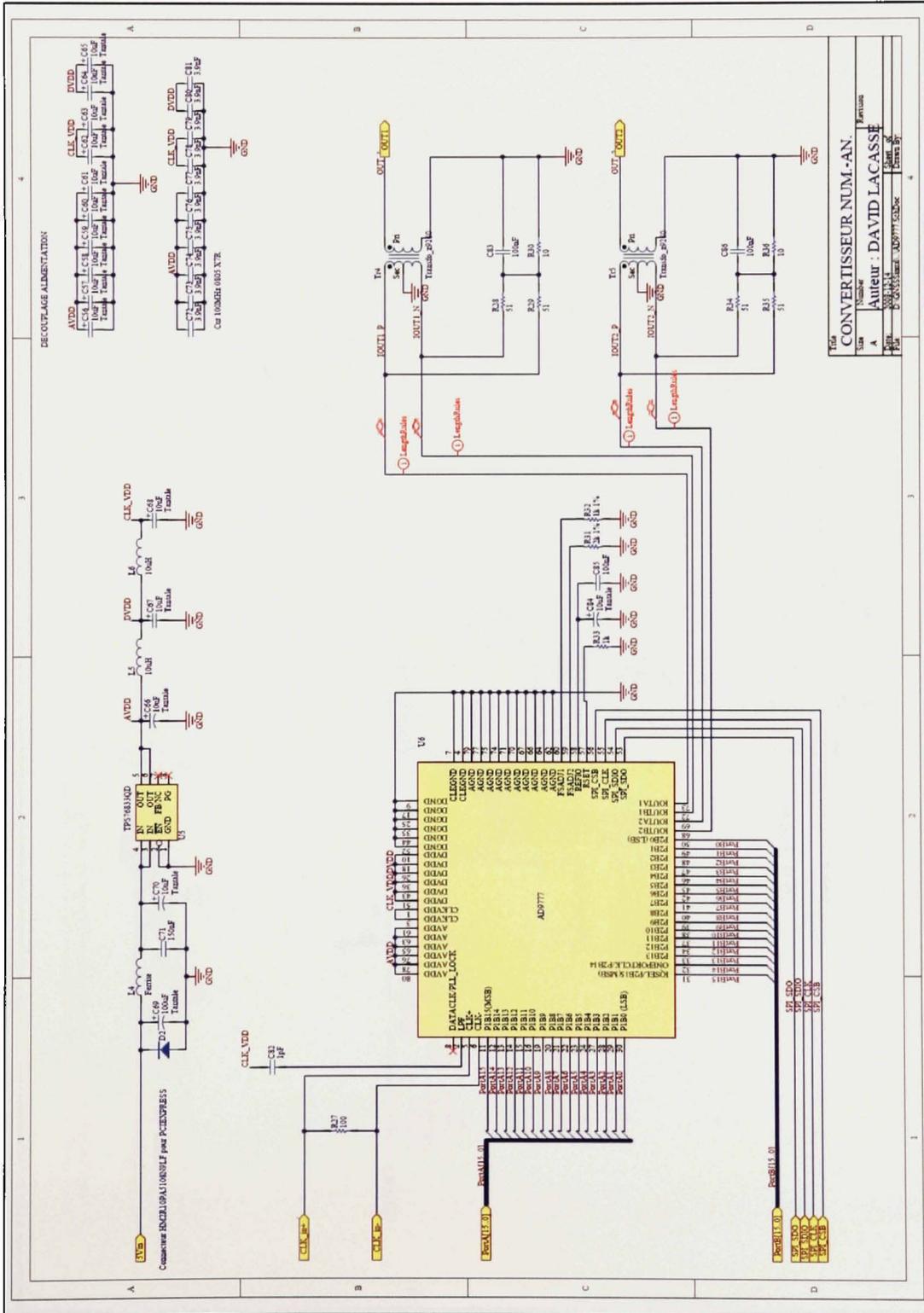
Transversal Filter - another name for standard FIR filter implementations, where the input samples traverse their way through the delay elements of a FIR filter.

III.4 References

- [1] L. R. Rabiner and B. Gold, *The Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs New Jersey, 1975, pp. 206, 273, and 288.
- [2] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989, pp. 236, and 441.
- [3] Timo I. Laakso et al., "Splitting the Unit Delay," *IEEE Signal Processing Magazine*., January 1972, pp. 46.
- [4] John Pickerd, "Impulse-Response Testing Lets a Single Test Do the Work of Thousands," *EDN*, April 27, 1995.

ANNEXE V

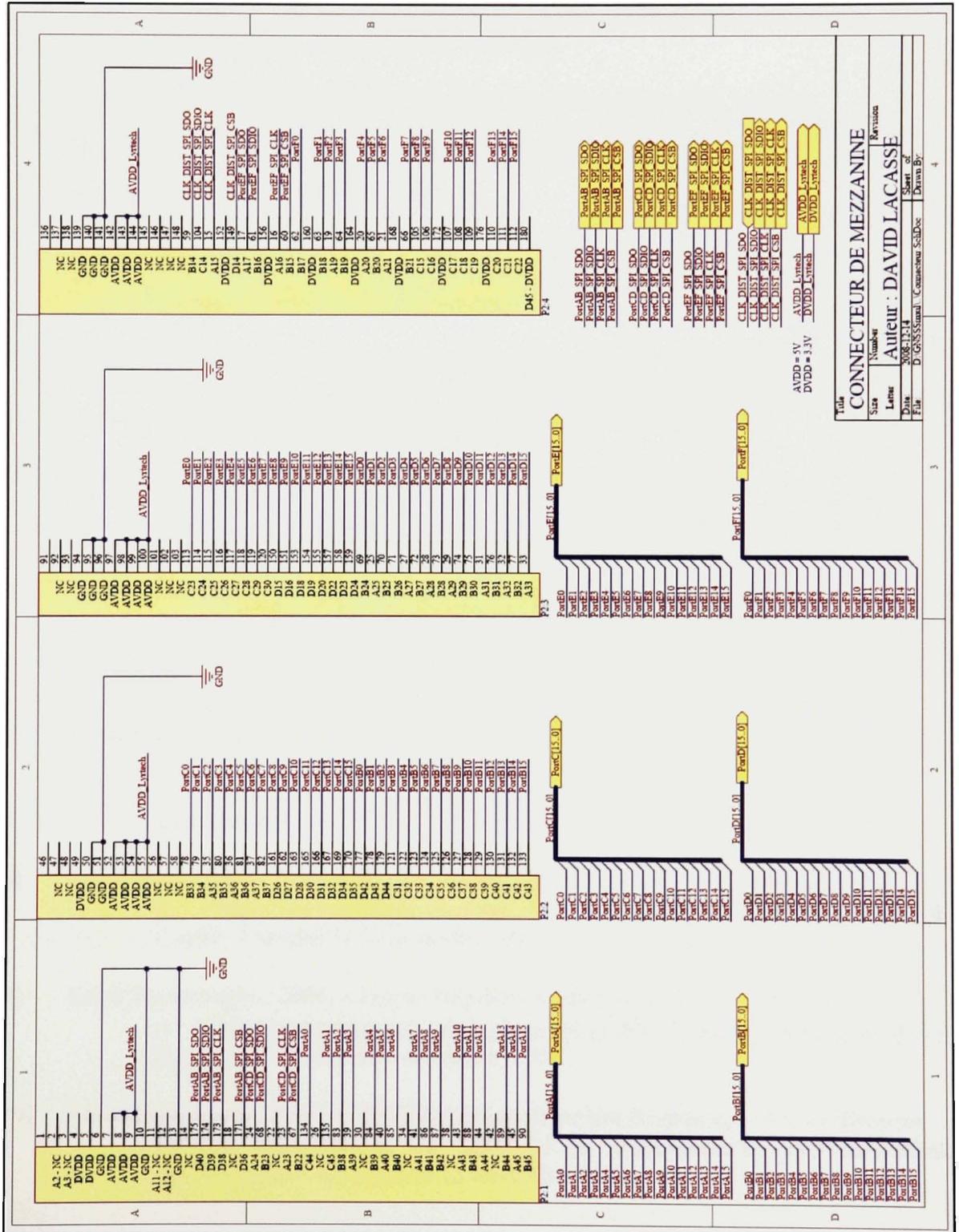
SCHEMA DU AD9777



Titre	
CONVERTISSEUR NUM-AN.	Version
Num	Autheur : DAVID LACASSE
A	00000001
Rev	00000001
Doc	00000001
Proj	00000001
Part	00000001
Sheet	00000001

ANNEXE VII

SCHEMA DU CONNECTEUR DE MEZZANINE



BIBLIOGRAPHIE

- [1] Analog Device. 2006. « AD9777 Interpolating Dual TxDAC+ D/A Converter ». En ligne. 60 p. <http://www.analog.com/static/imported-files/data_sheets/AD9777.pdf> . Consulté le 9 avril 2007.
- [2] Analog Device. 2005. « AD9512 1.2 GHz Clock Distribution IC, 1,6 GHz Inputs, Dividers, Delay Adjust, five Outputs ». En ligne. 48 p. <http://www.analog.com/static/imported-files/data_sheets/AD9512.pdf>. Consulté le 16 octobre 2007.
- [3] Bryant, James M. 2006. « Ground : Where Good Signals Go When They Die». En ligne. <http://www.analog.com/static/imported-files/rarely_asked_questions/moreInfo_raq_groundingClean.html>. Consulté le 16 juillet 2008.
- [4] Bryant, James M. 2006. « Ground Noise and Groundhogs -Folklore for EEs ». En ligne. 1 p. <http://www.analog.com/static/imported-files/rarely_asked_questions/RAQ_groundhog.pdf>. Consulté le 16 juillet 2008.
- [5] Bryant, James M. 2006. « Grounding Converters ». En ligne. 1 p. <http://www.analog.com/analog_root/static/pdf/raq/RAQ_groundingADCs.pdf>. Consulté le 16 juillet 2008.
- [6] GUAN, Hongyu. 2008. « Le chinois Beidou sort sa première puce de géolocalisation». En ligne. <http://www.chine-informations.com/actualite/le-chinois-beidou-sort-sa-premiere-puce-de-geolocalisation_8754.html>. Consulté le 15 juillet 2008.
- [7] Kester, Walt. 2006. « Evaluating High Speed DAC Performance ». En ligne. <http://www.analog.com/en/digital-to-analog-converters/da-converters/products/tutorials/CU_tutorials_MT-013/resources/fca.html>. Consulté le 4 juin 2008.
- [8] Laird Technologies. 2004. « Ferrite 11ohms MI0805K110R-10 ». En ligne. 1 p. <http://www.steward.com/web_info/CADPrints/Sales/MI0805K110R-10-C.pdf>. Consulté le 14 novembre 2007.
- [9] Laird Technologies. 2004. « Ferrite 600ohms MI0805K601R-10 ». En ligne. 1 p. <<http://www.steward.com/web_info/CADPrints/Sales/MI0805K601R-10-B.pdf>>. Consulté le 14 novembre 2007.
- [10] Texas Instruments. 2006. « TPS768xxQ Fast Transient Response, 1-A Low-Dropout Voltage Regulators ». En ligne. 23 p. <<http://focus.ti.com/lit/ds/symlink/tps76833.pdf>>. Consulté le 15 novembre 2007.