RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ IBN-KHALDOUN DE TIARET FACULTÉ DES SCIENCES APPLIQUEES DÉPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Électronique

Spécialité : Électronique des Systèmes Embarqués

THEME :

Filtrage Optimal de l'impulsion Nucléaire issue d'un détecteur gamma de type HPGe

Préparé par : Leith ABADI et ALLADJO Parfait Mustapha

Devant le Jury :

Noms et Prénoms	Grade	Qualité
B. SAHLI	MCA	Président
M. BELARBI	MCA	Encadreur
D. NASRI	MCA	Examinateur
M. SEBAA	Pr	Examinateur

Promotion : 2021/2022

Dédicace

Nous dédions ce travail à nos parents

Remerciements

À l'issue de cette fin d'étude, nous adressons nos sincères remerciements premièrement à « Allah » tout puissant qui nous a donné la santé, la patience.
Et je tien a remercié également mon prophète « MOHAMED » pour nous guider à la bonne voie.

Nous remercions sincèrement Monsieur NASRI Djillali, pour le grand honneur d'avoir accepté de présider le jury de soutenance.
Ensuite, on tient à adresser nos plus vifs remerciements à nos promoteurs Mr : Belarbi Moustapha et Mr : BELMASAOUD Nahir pour nous avoir encadré, suivi et encouragé.
Nos vifs remerciements vont également à Messieurs M. SEBAA et B. SAHLI d'avoir accepté d'examiner notre travail et d'être membre du jury de soutenance.
On remercie également, tous les membres du jury de bien vouloir juger ce travail.
Ainsi que tous nos enseignants du département de GE.
On n'exclue pas de ces remerciements toutes les personnes qui ont aidé de près ou de loin dans la réalisation de cette PFE.

Enfin, on remercie tout particulièrement nos parents, pour leur soutien inconditionnel tout au long de nos longues années d'études.
On veut également remercier nos familles et nos amis pour leur soutien moral.
Les discussions, les remarques et les commentaires de nos collègues ont été sources d'idées et ont contribué au développement et à l'amélioration de cette étude.

Table des matières

Liste des Figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Liste des Abréviations

HPGe	High Purity Germanium
ADC	Analog to Digital Converter
AXI	Advanced eXtensible Interface
BLR	Base Line restorer
CCS	Code Composer Studio
DAC	Digital to Analog Converter
D.S.P	Density Spectral de Puissance
DSP	Digital Signal Processor
DC	Direct Current
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FET	Field Effect Transistor
IIR	Infinite Impulse Response
IDE	Integrated Development Environment
MCA	Multi Channel Analyzer
ZED-Board	Zynq Evaluation Development Board
TRP	Preamplifications Reinitialization Transistor
RFP	Preamplifications Resistive
PL	Programmable Logic
PS	Processing System
DC-S	Direct Current Offset
PS	Pulse Shaper
FF	Fast Filter
PUR	Pile-Up Rejector
PHA	Pulse Height Analyzer

Introduction Générale

Introduction Générale

Toutes les installations et institutions nucléaires utilisent des détecteurs de rayonnement dans une certaine mesure, que ce soit à des fins de recherche, de surveillance ou d'analyse. Chaque détecteur de rayonnement utilisé doit disposer d'une électronique nucléaire compatible, adaptée de manière optimale aux caractéristiques du détecteur, afin d'obtenir les meilleures performances. Les détecteurs au germanium de haute pureté (HPGe) offrent une excellente résolution énergétique.

Ce mémoire s'inscrit dans le cadre de l'implémentation d'un filtre trapézoïdal optimal pour la mesure d'une impulsion nucléaire gamma. Deux approches ont été abordées dans ce document. L'implémentation de ces fonctions dans un FPGA est une tâche exigeante et peut être compliquée par le codage VHDL. Par conséquent, les outils IDE Xilinx System Generator et Matlab/Simulink sont utilisés pour son développement. Ces outils permettent d'utiliser des blocs prédéfinis et testés, un langage de script pour l'implémentation de la logique de contrôle et enfin de générer du code VHDL. Tout ceci est réalisé dans un environnement graphique CONVIVIAL. Toutefois, compte tenu de la consommation des ressources (des blocs logiques configurables), nous allons implémenter ce type de filtre en langage VHDL et comparer les deux méthodes utilisées afin d'en ressortir une solution optimale. Il existe d'autres méthodes, comme une implémentation en Verilog, ou en Java qui supportent l'orientée objet.

La méthode de mise en forme trapézoïdale est largement appliquée à l'extraction de l'amplitude des impulsions dans un système de spectrométrie nucléaire numérique, afin de mesurer avec une haute précision, la distribution d'énergie des rayons gamma. Dans le domaine analogique, une impulsion de courant produite par le détecteur est intégrée à l'aide d'un préamplificateur sensible à la charge qui produit une impulsion en forme de pas ou une impulsion exponentielle à décroissance lente. Dans un système de spectroscopie nucléaire numérique typique, le signal du détecteur est numérisé à l'aide d'un convertisseur analogique-numérique (CAN) rapide, directement après l'étage du préamplificateur, et les opérations de traitement des impulsions, telles que la correction de la ligne de base, la mise en forme des impulsions et la correction de l'empilement, sont effectuées sur un matériel numérique (par exemple, un FPGA). Le filtre trapézoïdal transforme la caractéristique générée par le préamplificateur sensible aux charges en une impulsion trapézoïdale, avec des temps de montée et de descente identiques. Le sommet plat, dont la durée est supérieure au temps de collecte, est insensible aux variations du temps de montée de l'impulsion d'entrée.

Par conséquent, le sommet plat nous offre la flexibilité pour la suppression du bruit afin d'obtenir le meilleur rapport signal/bruit. Le signal transformé alimente le sous-système BLR (Base Line Restorer) qui élimine tout décalage DC offset du signal à son entrée. La sortie du sous-système BLR alimente le sous-système PHP qui calcule l'amplitude de chaque impulsion et la transmet vers un sous-système externe pour l'analyse et le stockage [1].

Puisque le traitement est effectué à l'aide d'un ordinateur, le throughput peut être affecté si le nombre de canaux analogique (ADC) est important. Une approche pour réaliser le traitement de l'impulsion en temps-réel est d'implémenter un filtre numérique trapézoïdal pour l'analyse du spectre de radiation gamma.

Ce document est consacré à l'étude et à la mise au point de ce filtre, tout en minimisant l'effet de l'empilement et du déficit balistique, susceptibles de fausser les mesures. Les propriétés de ce type de filtrage créent une excellente correspondance dans le choix du matériel Zynq-7000, qui posséde des caractéristiques intéressantes pour l'implémentation du filtre numérique. La compilation du code VHDL dans l'IDE Vivado permet de générer le fichier de configuration appelé bitstream servant à programmer le composant FPGA.

Ce mémoire est organisé en quatre chapitres. Dans un premier temps, un aperçu général sur la spectroscopie Gamma et les chaines de mesure associées sera présenté au chapitre un. Le chapitre deux sera consacré à la prise en main de l'IDE Vivado et de la plateforme matériel Zynq SoC. Il sera suivi par le chapitre trois dédié à l'estimation de la densité spectrale de puissance du bruit à l'entrée du spectromètre. Le chapitre quatre sera réservé à l'implémentation du filtre trapézoïdal optimal pour la mesure d'énergie interprétée par l'impulsion captée aux bornes du détecteur. C'est ce filtre qui s'occupera de la tache de l'estimation de la ligne de base (BLR : Base Line Restorer) liée à la référence de mesure. Enfin, nous en terminerons par une conclusion, où nous dresserons un bilan englobant les résultats et les éventuelles débouchés et extensions du travail décrit tout au long de ce document.

Chapitre I :

Généralités sur les Chaines de Mesures Spectrométriques

I.1 Introduction

Pendant longtemps, le filtrage optimal était possible seulement avec les composants analogiques. Ces dernières décennies, avec l'apparition des FPGAs, il est possible de numériser les impulsions, même après pré-amplification ou photo-tube, et de les traiter en temps-réel. Lorsque le nombre de canaux ADC est élevé, la vitesse d'acquisition est affectée, d'où l'intérêt d'implémenter un filtre numérique. L'implémentation de ce système peut se faire sur un Zynq-7000 de la société Xilinx, grâce à la disponibilité des FPGAs à haute résolution, et à haute densité des blocs logiques. Le parallélisme dans l'architecture d'un FPGA se soldera par les résultats ayant de bonne performance. Notre étude se portera sur le filtrage optimal de l'impulsion nucléaire issue d'un détecteur gamma de type HPGe. La carte ZedBoard (Zynq Evaluation Development Board) qui est une carte de développement issue de Xilinx Inc, possède des caractéristiques qui intéressent l'implémentation du filtre numérique à base du composant FPGA.

I.1.1 Propriétés générales des détecteurs de rayons γ

Considérons un détecteur soumis à l'irradiation de rayons gamma mon-énergétiques provenant d'une source. Pour que le détecteur produise une réponse, un rayon gamma doit subir une interaction. Le rayon gamma cède tout ou partie de son énergie à un électron et le libère. L'électron libéré entre en collision avec d'autres atomes du matériau et les ionise, libérant ainsi d'autres électrons.

Les électrons libérés peuvent être collectés pour générer une impulsion électrique, soit directement (comme avec les détecteurs à semi-conducteurs à l'état solide), soit indirectement (comme avec les détecteurs à scintillation). La charge collectée enregistre la présence d'un rayon gamma et est directement proportionnelle à l'énergie que le rayon gamma a déposée dans le milieu de détection.

Une première considération pour la propriété d'un détecteur est d'évaluer la probabilité qu'un rayon gamma interagisse avec le détecteur. La probabilité qu'un rayon gamma interagisse avec un électron dans un matériau est directement proportionnelle à la densité électronique du matériau. Les gaz ayant une densité très faible, les détecteurs remplis de gaz sont très inefficaces pour détecter les rayons gamma. Cela exclut les détecteurs remplis de gaz pour la spectroscopie des rayons gamma.

Les détecteurs à l'état solide et liquide sont les plus couramment utilisés pour la spectroscopie des rayons gamma. Les deux types de détecteurs à semi-conducteurs

couramment utilisés pour la spectroscopie des rayons gamma sont les détecteurs à scintillation et les détecteurs à semi-conducteurs. Une grande variété de matériaux est utilisée pour la détection par scintillation.

Le terme de spectrométrie en lui-même est vague, car il ne fait référence qu'à l'analyse de données en composants élémentaires susceptibles d'être disposés le long d'un ou plusieurs axes (numériques). C'est la représentation sur un graphique de cette répartition que l'on nomme « spectre ». Il faut donc à chaque fois préciser la nature des données et la dimension des axes (fréquence en Hertz, énergie en électron-volt ou encore masse en kilogramme). En spectrométrie nucléaire, ou plus précisément en spectroscopie gamma, il s'agit de mesurer comment se répartissent, ou se distribuent, les photons gamma selon leur énergie individuelle en électron-volts, ou un multiple de cette unité. Les radionucléides sont ainsi caractérisés par l'énergie des photons qu'ils émettent. Ces énergies sont discrètes, connues et archivées dans des bases de données. Idéalement, il est donc possible à partir du spectre d'identifier les radionucléides.

Il est par conséquent possible de caractériser la composition, en termes de radionucléides présents, et l'activité, c'est-à-dire le nombre de désintégrations par seconde ou Becquerels, d'une source émettrice de photons gamma. Le résultat final d'une mesure de spectrométrie gamma est, par exemple, un énoncé de la forme : « la source radioactive contient du Césium 137 d'activité 4500 Becquerels et de l'Europium 152 d'activité 3700 Becquerels ».

En spectrométrie, on mesure la distribution en énergie (ou spectre en énergie) des particules émises par une source radioactive ou produites par une réaction nucléaire. Le terme de spectre peut sembler mal choisi, car il fait référence à des méthodes (analyse spectrale, transformations de Fourier) qui n'ont pas lieu d'être appliquées dans le cadre de la spectrométrie gamma. Toutefois, l'énergie des photons étant proportionnelle à leur fréquence d'après la relation de Planck, cet abus de langage est conventionnellement utilisé. Il existe deux sortes de spectre en énergie :

- Le spectre en énergie différentiel.
- Le spectre en énergie intégral.

$$N(E) = \int_{\infty}^{\infty} n(E)dE \tag{I.1}$$

Dans toute la suite nous nous intéresserons uniquement à des spectres en énergie différentiels. La figure I.1 représente un spectre en énergie de l'élément Césium 137 simulé, alors que la figure I.2 représente un spectre en énergie réel de l'élément Cs^{137}.

Nous insistons encore sur le fait que ces spectres en énergie ne sont pas obtenus par transformée de Fourier, mais qu'il s'agit d'un histogramme des énergies enregistrées sur une certaine période de temps.

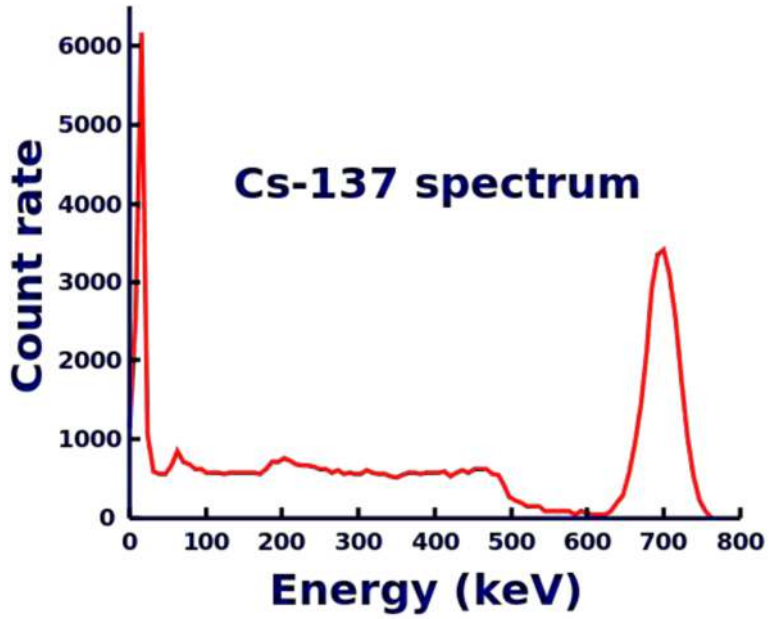


Fig.I.1 : Spectre du Cs_{137} [30].

Nous remarquons d'emblée que les spectres présentés semblent contredire ce qui a été dit précédemment, puisque qu'ils ne sont pas uniquement constitués de raies mono-énergétiques, mais également on distingue l'apparition d'autres raies parasites ainsi qu'un élargissement tangible de la raie principale au sein du spectre réel.

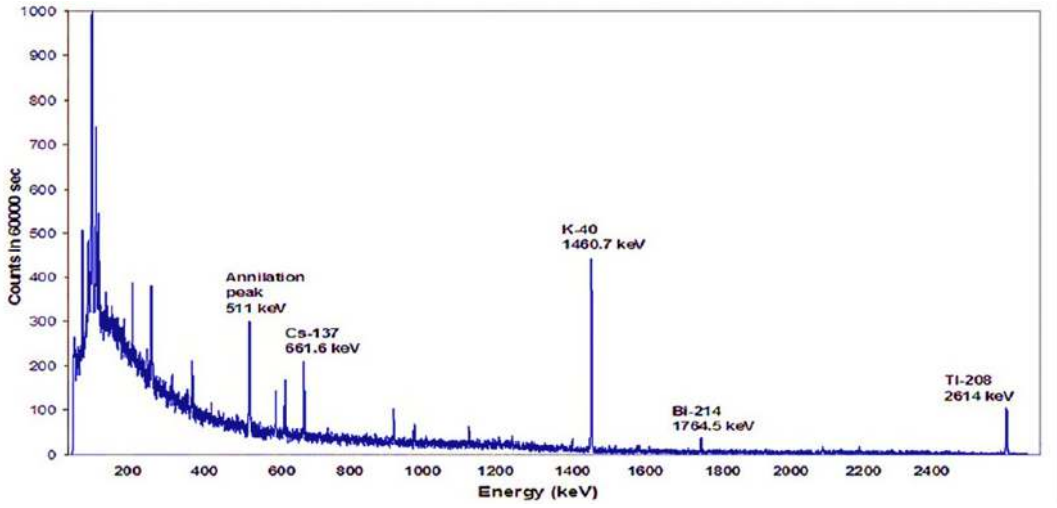


Fig.I.2 : Spectre réel [31].

Ceci est dû à la présence de plusieurs types de perturbations qui dégradent le spectre de raies idéal ; celles-ci sont classées dans trois groupes :

1. Les perturbations provenant de l'interaction des photons incidents avec la matière du détecteur,
2. Les perturbations provenant de la chaîne d'instrumentation et de l'acquisition du signal,
3. Les perturbations provenant du caractère aléatoire du signal incident.

I.1.2 L'instrumentation de mesure spectroscopique

Le principe d'une chaîne de détection nucléaire est de convertir chaque photon, parcourant la zone sensible du détecteur, en un signal mesurable, proportionnel à l'énergie du photon incident. En général, un spectromètre nucléaire, indépendamment des modes et des technologies utilisées, effectue les opérations suivantes [2] :

- Un filtrage linéaire qui minimise le bruit de nature électrique,
- Une classification des valeurs de ces maximas dans un histogramme.
- Une détection des maximas du signal temporel mis en forme (si la forme est constante, le maximum d'une impulsion est proportionnel à son intégrale.),

I.2 Les chaînes de mesure analogiques

Un spectromètre gamma classique (analogique) comprend un détecteur en germanium (Ge) avec son cryostat refroidi de manière mécanique ou par azote liquide, un préamplificateur, une source de polarisation du détecteur, un amplificateur, un convertisseur analogique/ numérique (CAN) associé à un dispositif de stockage multicanal du spectre et des afficheurs de données.

I.2.1 Le détecteur à semi-conducteurs

C'est l'élément clef du système de mesure de radiation ionisante. Il s'agit d'un transducteur qui génère un signal électrique constituant le message transportant l'information sur le phénomène physique (Interaction avec la matière) qui a lieu dans le détecteur lui-même. L'information extraite, mesure un paramètre du signal, un tel paramètre peut être une tension Vi, un courant Ii, une charge qi ou enfin le temps d'apparition du signal tout en respectant une certaine origine. Le signal en question est généralement très faible, d'où l'impossibilité d'une mesure directe du paramètre à la sortie du détecteur et la nécessité de l'emplacement d'un préamplificateur en aval.

En fait, un détecteur semi-conducteur n'est rien d'autre qu'une diode polarisée en inverse, avec une zone de déplétion épaisse (voir la figure I.3). Et lors de son interaction

avec le détecteur, la particule incidente, cède toute ou une partie seulement de son énergie [3]. La charge créée par ionisation dans le détecteur est directement proportionnelle à la perte d'énergie de la particule. Les détecteurs habituellement employés en spectroscopie Gamma sont des matériaux semi-conducteurs (germanium hyper pur HPGe, ou dopé au lithium GeLi).

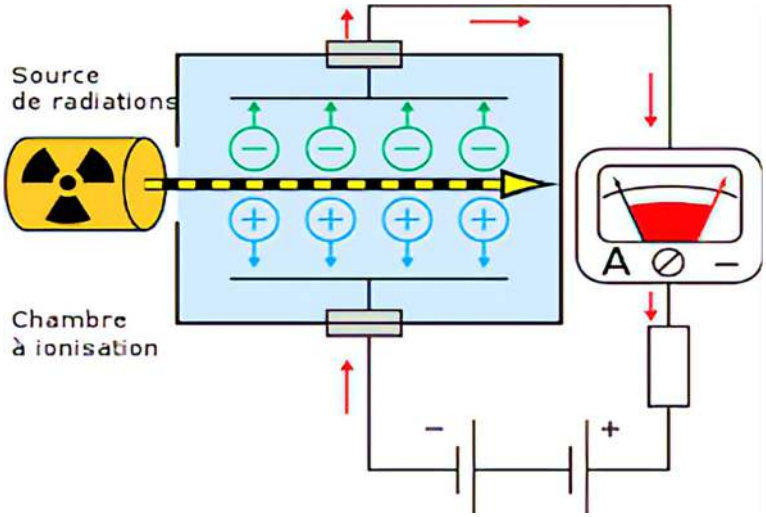


Fig.I.3 : Détecteur de radiation à base de semi-conducteur (HPGe) [32].

Les paramètres importants pour ces mesures sont l'efficacité de détection (rapport de rayonnements détectés sur les rayonnements émis par la source radioactive), la résolution énergétique (largeur de la raie à mi-hauteur) et le rapport pic/Compton. Ces critères conditionnent le type de détecteur choisi pour l'application visée.

I.2.2 Le détecteur au Germanium

Un détecteur au germanium de haute pureté est un grand monocristal de germanium avec des concentrations d'impuretés aussi faibles que 1010 atomes/cm^3. La plupart des détecteurs modernes sont de type coaxial. Si le détecteur est de type p (les impuretés de l'accepteur dominent), un contact au lithium est formé sur la surface extérieure. Cela crée une jonction np à la surface externe du cristal.

Un contact en bore est implanté sur la surface intérieure pour se connecter au préamplificateur. Si le détecteur est de type n (les impuretés du donneur dominent), les deux contacts sont inversés, créant une jonction pn. Puisque l'implant en bore peut être rendu très fin, le détecteur de type n peut détecter des rayons gamma de plus faible énergie, puisque les rayons gamma peuvent pénétrer dans le contact plus faible.

Une polarisation positive est sur le contact n+ du détecteur de type p pour épuiser complètement le cristal, et une polarisation négative est placée sur le contact p+ pour

appauvrir le détecteur de type n. La polarisation est généralement supérieure à la valeur requise pour épuiser le détecteur afin d'augmenter le champ électrique. Le germanium ayant une bande interdite de seulement 0,7 eV, un grand nombre de porteurs induits thermiquement sont présents dans le détecteur à température ambiante. Ces porteurs créent un courant de fuite important qui ajoute un bruit excessif. Pour cette raison, Les détecteurs HPGe sont refroidis à 77 K avec de l'azote liquide. [4]

I.2.3 Le Préamplificateur de Charge

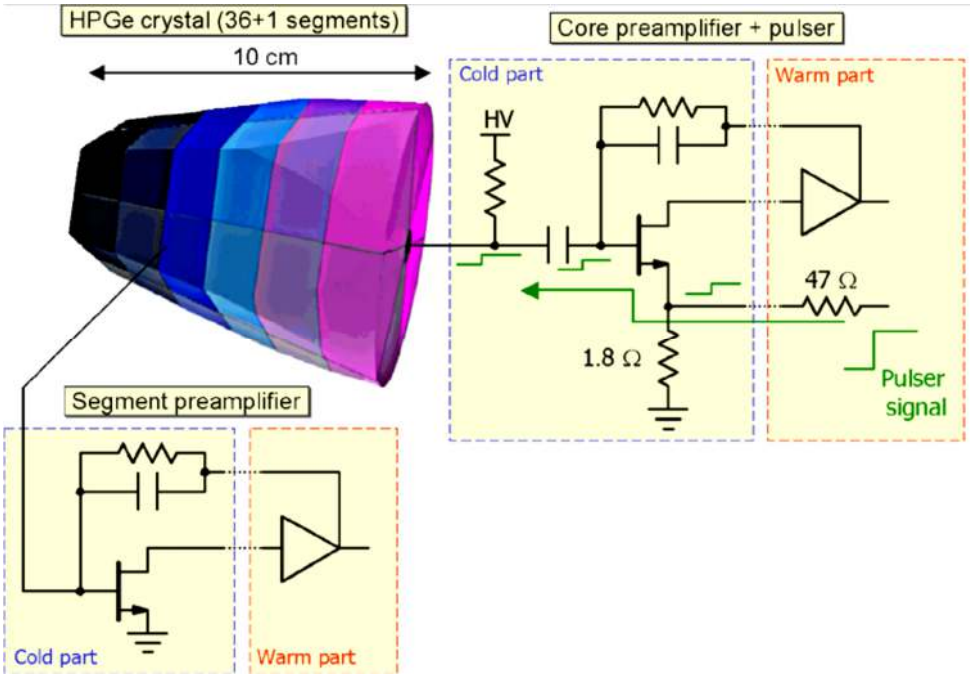
Un préamplificateur de charge est en fait un intégrateur électronique à faible bruit. Sa réalisation la plus simplifiée peut consister en un amplificateur inverseur à très grand gain et à retour purement capacitif, tel illustré dans la figure I.4. Pour diminuer le bruit électronique, le préamplificateur doit être placé le plus près possible du détecteur et il est solidaire au cryostat.

Le préamplificateur est isolé de la haute tension par une capacité. Le signal à l'entrée du préamplificateur n'est pas forcément proportionnel à l'énergie déposée dans le cristal. Par contre l'intégrale de ce signal est égale à la charge collectée qui dépend de l'énergie déposée. C'est pourquoi, dans la plupart des applications spectroscopiques sont utilisés des préamplificateurs « sensibles à la charge ». Il s'agit d'un montage intégrateur.

Le temps de montée du signal de sortie est relié au temps de collection de charge alors que le temps de descente ne dépend que de la constante de temps RC du montage intégrateur. Le temps de montée peut varier de quelques ns à quelques µs et le temps de descente est en général fixé à 50 µs. Ceci signifie que l'on a un temps de montée rapide et un temps de descente beaucoup plus lent.

Deux types de préamplificateurs sont généralement utilisés avec les détecteurs HPGe. Les préamplificateurs résistifs (RFP) et les préamplificateurs à réinitialisation par transistor (TRP). Le RFP est composé de deux étages. Le premier étage est connecté

au détecteur avec un transistor à effet de champ (FET) à faible bruit.



(b) (a)

Fig.I.4 : Préamplificateur de charge. a) : schéma de principe. b) : Schéma simplifié [33].

Un condensateur de rétroaction C_f intègre la charge collectée par le détecteur, générant un échelon de tension à la sortie du premier étage. La résistance R_f décharge le condensateur pour ramener la sortie du premier étage à son niveau initial de courant continu d'origine. Afin d'obtenir une performance de bruit optimale du détecteur, le condensateur de rétroaction doit être assez petit (< 1pf) et la résistance assez grande (- 2 000 M ohms). Par conséquent, la constante de temps de l'exponentielle de sortie est assez longue (1 à 2 ms). Le deuxième étage se compose d'un réseau pôle-zéro et d'un étage de gain. Le réseau pôle-zéro réduit la constante de temps de l'exponentielle à environ 50 µs, en appliquant un filtre avec la fonction de transfert suivante :

$$H(s) = \frac{s - \dfrac{1}{R_f C_f}}{s + \dfrac{1}{\tau_{new}}} \qquad \text{(I.2)}$$

En prenant les transformées de Laplace inverses, on obtient une impulsion exponentielle avec une constante de temps égale τ_{new}. Cette constante de temps plus courte permet un gain plus important sans exiger une grande plage dynamique de la sortie du préamplificateur.

I.2.4 L'Amplificateur de Forme

L'amplificateur (Shaper-Amplifier) réalise deux tâches essentielles : la mise en forme du signal et son amplification.

Il s'agit de filtrer le signal de manière à avoir le meilleur rapport signal sur bruit. Le signal de sortie du préamplificateur étant assez lent, avant qu'il atteigne zéro volt, il est possible qu'un autre signal arrive [3].

Pour éviter ceci, le signal de sortie du préamplificateur est différencié pour éliminer la composante lente. Cela a pour effet de préserver uniquement l'information relative aux caractéristiques du détecteur contenues dans la constante de temps du signal montant. Ensuite, le signal est intégré pour réduire le bruit et on obtient un signal quasiment gaussien. La largeur à mi-hauteur de la gaussienne obtenue s'exprime en termes de constante de temps que l'on peut régler. Si elle est choisie trop courte, le bruit est amplifié, si elle est trop longue, on peut assister à l'empilement de deux signaux électriques. En général, la constante de temps est comprise entre 3 et 30 µs suivant le type du détecteur.

I.2.5 L'ADC et l'analyseur multicanaux

À la sortie de l'amplificateur, on a un signal continu analogique. Pour l'acquisition, il est préférable de travailler avec des grandeurs discrètes, c'est-à-dire qui ne peuvent prendre qu'un ensemble fini de valeurs. C'est pourquoi on utilise une ADC, qui transforme l'amplitude du signal électrique fourni par l'amplificateur en un nombre qui est alors proportionnel à l'énergie déposée dans le cristal.

La sortie de l'ADC est enregistrée dans une mémoire qui possède autant d'adresses que le maximum de numéro de canaux servant à découper le spectre final [3]. Il y a au total 4096 canaux disponibles pour l'acquisition. L'analyseur multicanaux sert à collecter et à enregistrer les évènements issus de l'ADC en les classant. Dès qu'un signal a été analysé par l'ADC, l'adresse mémoire dans laquelle il a été enregistré est bloquée et son contenu est incrémenté d'une unité [3].

Cela permet d'obtenir un histogramme en temps réel dans lequel l'énergie déposée dans le cristal est reliée au numéro de canal (plus l'énergie est élevée, plus le numéro de canal est élevé) et le nombre de coups dans un canal donné est proportionnel au nombre de photons ayant déposé la même énergie dans le cristal. Cette opération nécessite entre 1,5 et 3 µs. L'entrée convertisseur analogique-numérique (CAN) par un circuit de détection et d'étirement des pics. Ce circuit capture l'amplitude maximale d'une

impulsion et la conserve pour que le convertisseur analogique-numérique la traduise en une valeur numérique.

Comme les détecteurs HPGe ont un bruit très faible, l'ADC doit avoir une très haute résolution (14 bits). L'ADC doit également être relativement rapide (< 10ns) et avoir une excellente linéarité différentielle (< 1%). Un curseur Gatti est implémenté sur la plupart des ADC pour atteindre cette spécification de linéarité différentielle. La valeur numérique de l'ADC est disponible pour le stockage ou l'histogramme [5].

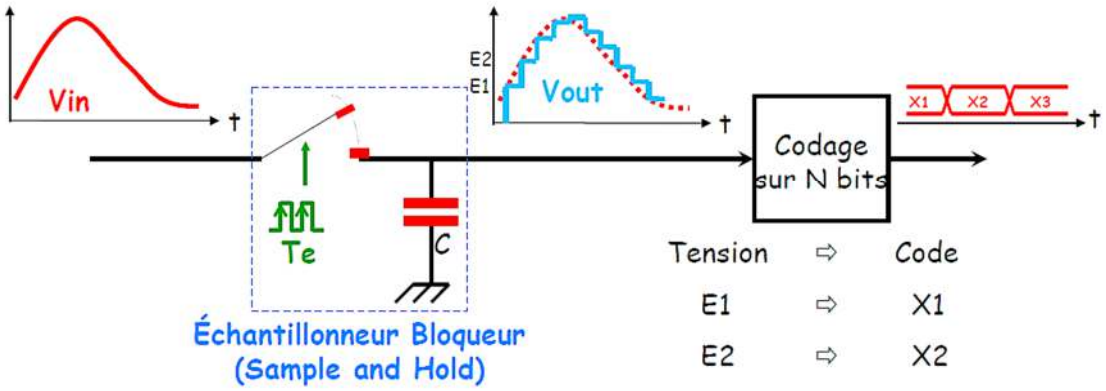


Fig.I.5 : Schéma fonctionnel ADC pour la spectroscopie [28].

I.3 Mode de fonctionnement typique

L'électronique numérique, connectée au préamplificateur, joue le rôle d'amplificateur, de convertisseur analogique numérique et d'analyseur multicanaux et va par conséquent accepter en entrée des impulsions en tension. Ces signaux sont amplifiés à l'aide d'un gain comme pour une électronique analogique. Une fois le signal amplifié, l'électronique numérique réalise trois étapes : l'identification du signal (filtre rapide), le codage numérique (filtre lent) et le remplissage du spectre. La figure I.6 représente les deux filtres appliqués au signal numérique.

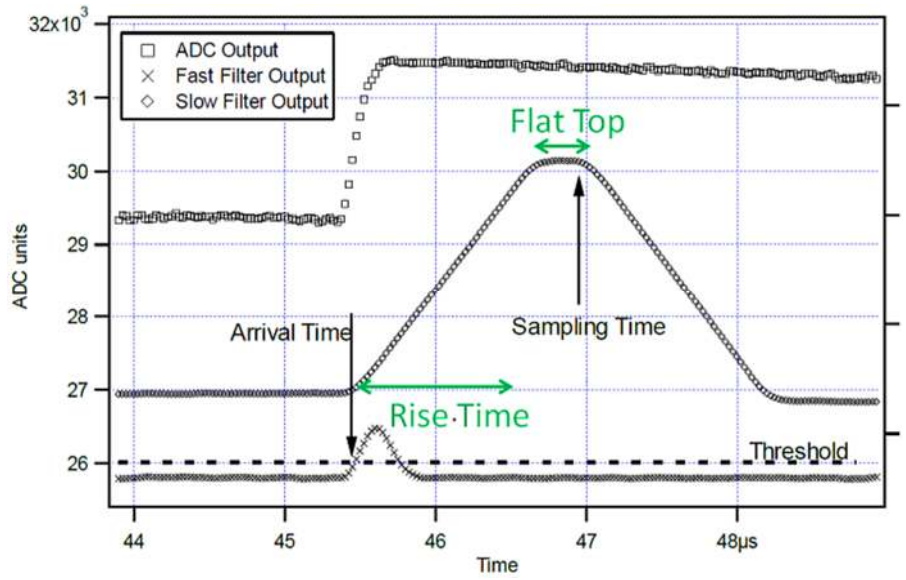


Fig.I.6 : Signaux de l'ADC (carré), filtre rapide (croix) et du filtre lent (losange) [29].

Un filtre comporte deux constantes du temps : le temps de montée (Rise Time) et le temps du plateau (Flat Top).

Le filtre rapide possède deux temps relativement courts car il doit juste détecter une variation de la ligne de base qui correspond à un signal : il identifie un événement. Le filtre lent a des constantes de temps plus longues afin de quantifier précisément la hauteur de l'impulsion. Si ces constantes sont augmentées, alors la hauteur de l'impulsion sera déterminée plus précisément, ce qui permet d'avoir une meilleure résolution en énergie. Une fois l'événement détecté et son amplitude déterminée, l'analyseur va incrémenter le canal correspondant dans le spectre.

I.4 Les chaînes de mesures numériques

D'un point de vue conceptuel, les traitements numériques et analogiques partagent plusieurs buts communs, qui incluent :

1) La suppression de l'information non essentielle (c.-à-d. « bruit ») ;
2) La réduction du flux de données entrant à un niveau maniable ;
3) Le tri et la présentation des données extraites de manière intelligibles.
4) L'extraction de l'information d'intérêt (hauteur de l'impulsion, forme d'impulsion, temps d'arrivée, etc.) à partir d'un flux de données entrant ;

Mais, on affirme bien que les systèmes numériques du traitement de l'impulsion montrent un excédent de bord sur ceux dits analogiques. Et même si les systèmes de spectroscopie analogiques sont encore dominants sur le marché et sont toujours le seul choix possible pour des applications qui requièrent des temps de traitement de l'ordre

de 10ns, ils ne peuvent toujours pas atteindre l'exactitude et la compatibilité électromagnétique exigées en spectroscopie (X et gamma) haute résolution.

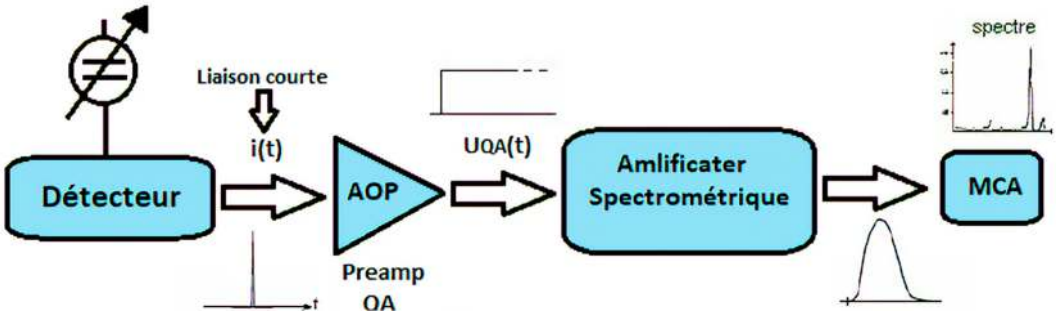


Fig.I.7 : Chaîne spectroscopique classique.

Ceci revient essentiellement aux difficultés rencontrées lors de la synthèse des fonctions de pondération proches de celles dites optimales sous une forme analogique et de leur sensibilité structurelle envers les contaminations par des perturbations dans les environnements bruyants. À cet égard, une approche numérique a le meilleur potentiel, à condition qu'une conversion analogique/numérique appropriée et une vitesse de calcul soient réalisées à un coût raisonnable.

D'une part le calibrage d'un système numérique implique une procédure d'installation (de démarrage) raffinée, qui est pour la plupart du temps évitée dans les contreparties analogiques, et d'autre part ceci permet de synthétiser pratiquement des réponses impulsionnelles de n'importe quelles forme de durée et ce en changeant juste les coefficients des filtres numériques implémentés.

Dans des implémentations typiques d'une chaîne mixte, le signal est échantillonné et numérisé à la sortie du préamplificateur. Les échantillons sont alors traités numériquement. Ils subissent une action de pesage dans des filtres numériques, par exemple des filtres Trapézoïdal [6].

Le traitement numérique ainsi réalisé, offre beaucoup d'avantages. Par exemple, dans des expériences modernes de la physique nucléaire le traitement numérique est plus une nécessité qu'un choix ; dans des situations pareilles, et avec des milliers de canaux parallèles, cette nouvelle tendance assure :

1. Une flexibilité pour des mesures d'amplitude ou avec la résolution maximale,
2. Une manipulation entièrement commandée par ordinateur,
3. Une transmission du signal affranchie de toute sorte de couplage, même lorsque les données doivent être transférées par des longs câbles.

Pour ce type de chaînes (généralement dites mixtes), deux philosophies différentes sont généralement utilisées dans l'architecture de ces systèmes : La première met le

convertisseur analogique numérique (ADC) et les modules de traitement numérique immédiatement après le préamplificateur de charge. Le signal obtenu à la sortie de ce module est échantillonné à une fréquence assez élevée à cause du front montant caractérisant l'impulsion à cette étape. Et concernant le traitement en aval, i.e. les mises en forme demandées, elles sont soit réalisées par le biais de la technique, devenue très populaire dans ce domaine, appelée technique de la déconvolution par fenêtre mobile (Moving Window Deconvolution Technique) [7]. Ou bien par l'implémentation d'un filtre numériques trapézoïdal directement inspirés de ceux déjà existants dans le domaine analogique [8] obtenus par les transformations appropriées.

Par contre, lors de la seconde configuration le convertisseur ADC est placé à la sortie d'un préfiltre analogique de conditionnement. Les traitements en amont sont réalisés en utilisant les techniques du filtrage optimal et/ou adapté et ceci pour filtrer efficacement le bruit de nature électronique capable de fausser les mesures.

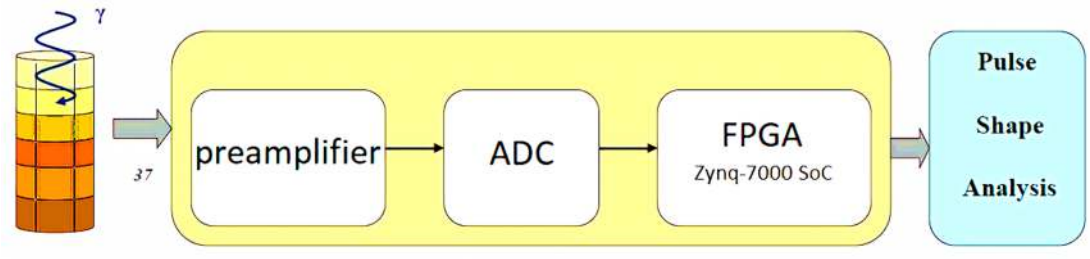


Fig.I.8 : Chaîne spectroscopique numérique (mixte).

I.5 Les sources d'erreurs dans une chaîne de mesure spectroscopique

Les sources d'erreurs dans une chaîne spectroscopique donnée sont très variées et difficiles à recenser. Et c'est la raison qui nous a poussé à ne discuter que les sources d'erreurs prépondérantes et qu'on entend corriger à travers les traitements numériques présentés lors des prochains chapitres [6].

I.5.1 Le bruit électrique

Au sens large, par bruit on entend dire une perturbation qui s'interfère avec le signal désiré. Le bruit d'origine électrique est un signal parasite pouvant affecter le signal issu d'un détecteur à radiations nucléaires, peut avoir diverses origines : bruit des sources excitatrices, bruit photonique, bruit électronique des appareils, ... etc. Suivant ces origines, deux grandes catégories émergent [6] :

- Bruits extérieurs
- Bruit thermique

I.5.2 Les bruits extérieurs

Il s'agit des contaminations pouvant altérer le fonctionnement du spectromètre, des fréquences parasites et ondulations des alimentations et champs électromagnétiques et du bruit blanc généré par des résistances connectées autour du préamplificateur.

Nous supposerons que des mesures adéquates sont prises dans la conception pour réduire ou éliminer ces sources de bruit dans les chaînes considérées tout au long de ce rapport. Ils sont aussi de différentes natures et dépendent en grande partie des composants utilisés par le procédé de fabrication du préamplificateur.

Le bruit d'un transistor MOSFET dont le principe a été suggéré par Lilienfeld dès 1930, a trois principales origines (en négligeant le courant de fuite de la grille d'un transistor MOSFET) : le bruit thermique ou bruit Johnson, le bruit de grenaille ou bruit Schottky, et le bruit en l/f ou 'Flicker noise'.

Lors de ce présent chapitre, on essayera de mettre l'accent sur la deuxième catégorie (bruit électronique des parties : détecteur + préamplificateur), et ce à cause de son impact négatif en détection des faibles signaux.

On entamera cette partie par l'introduction des principales sources de bruits mis en jeu dans l'électronique frontale d'une chaîne spectroscopique, et on fera un tour d'horizon sur les origines physiques et les modélisations mathématiques des densités spectrales de puissance de chaque variante à part [6].

I.6 Les sources du bruit électronique

Elles sont très nombreuses, tant externes qu'internes à la chaîne de mesure, dans une bande de fréquences pouvant aller de 1 MHZ à 100 MHz. Pour notre étude, on s'intéresse plus particulièrement au niveau de bruit intrinsèque des modules électroniques de la chaîne. Pour l'implantation du filtre optimal, on est contraint de rentrer un peu plus dans les détails, à cause des exigences de la haute résolution. Il existe une importante littérature concernant les bruits cités ci-dessous [9]. Aussi, nous ne ferons ici donc que de simples résumés.

I.6.1 Le bruit thermique

Le bruit thermique provient des fluctuations affectant les trajectoires des porteurs (électrons et trous dans les semi-conducteurs) dues aux interactions, et aux chocs avec le réseau. Ce mouvement aléatoire des porteurs est analogue au mouvement brownien des particules (possédant une énergie cinétique de (1/2) kT par degré de liberté). Ce

bruit a été observé par J. Johnson en 1927, et analysé théoriquement par Nyquist en 1928. Les autres dénominations sont mouvement brownien, bruit de Nyquist ou bruit de Johnson. La densité spectrale du bruit thermique d'un conducteur de résistance R est proportionnelle à la température absolue T de celui-ci, à la bande passante Δf et à la valeur de la résistance. Deux représentations équivalentes du bruit thermique généré par une résistance réelle R dans un circuit sont possibles.

La première est de type Norton, comportant une résistance idéale sans bruit, en parallèle avec une source de courant, représentant le bruit. La seconde représentation totalement équivalente, est de type Thevenin. Elle comporte une résistance idéale sans bruit en série avec une source de tension représentant le bruit, de densité spectrale indépendante de la fréquence, ce bruit est dit blanc.

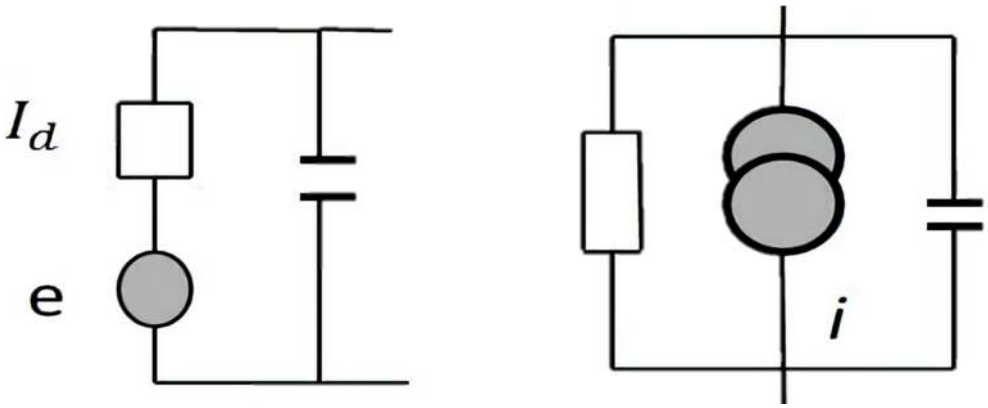


Fig.I.9 : Modèle du bruit thermique d'une résistance [34].

I.6.2 Le bruit de grenaille

Il se produit dans tous les dispositifs collectant un flux de particules électriques qui est dû à la nature granulaire de l'électricité [10] : c'est à dire qu'un courant I ne doit pas être considéré comme un flux uniforme, mais comme la composition d'un grand nombre d'impulsions élémentaires. Soit [t_j] la suite aléatoire que forme les instants où les électrons quittent l'électrode d'injection. En régime permanent, cette suite est supposée poissonnière. En négligeant l'influence du temps de transit entre les électrodes d'injection et de collection, le courant instantané peut s'écrire sous la forme d'une somme d'impulsions de Dirac :

$$I(q) = q\sum_{t_j}^{\delta}(t - tj) \tag{I.3}$$

Nous en déduisons le spectre de I(t) :

$$S_1(f) = I_0^2\delta(f) + 2qI0 \tag{I.4}$$

Dans les composants semi-conducteurs, ce bruit est associé aux courants qui franchissent les barrières de potentiel.

$$I = I_s(\exp(\frac{qV}{KT}) - 1) = -I_s + I_s \exp(\frac{qV}{KT}) = I_1 I_2 \quad \text{(I.5)}$$

Il est à noter que, tout comme le bruit thermique, le bruit de grenaille est un bruit blanc.

I.6.3 Le bruit Lorentzien

L'état d'un électron ou d'un trou dans un semi-conducteur peut être de deux types, soit délocalisé (dans la bande de conduction pour un électron ; dans la bande de valence pour un trou), soit localisé. Lorsque le porteur de charge, l'électron ou le trou, est dans un état délocalisé, il participe à la conduction. Par contre, lorsqu'il se trouve dans un état localisé, comme une impureté ou un défaut, il ne participe pas à la conduction.

De plus, la transition d'un électron ou d'un trou d'un état localisé vers un état délocalisé ou la création d'une paire électron-trou est appelée génération. Le processus inverse est appelé recombinaison. Le terme « piégeage » a également employé lorsqu'un électron ou un trou est capturé par une impureté, et le terme « dépiégeage » qualifie l'émission d'un porteur de charge. Comme les processus de génération-recombinaison sont aléatoires, le nombre de porteurs (électrons ou trous) dans les états délocalisés fluctuent autour d'une valeur moyenne qui définit la conductance moyenne.

Ces fluctuations du nombre de porteurs produisent une fluctuation de la résistance et par conséquent une fluctuation du courant et/ou de tension. Ces variations sont appelées bruit de génération-recombinaison, noté bruit de g − r, et sont dues à la fluctuation du nombre de porteurs. Le bruit de g − r dépend fortement des qualités du semi-conducteur : la conduction unipolaire ou bipolaire, le type de pièges, les centres de recombinaison. Dans le cas simple de transitions entre une bande et des pièges de même niveau d'énergie, le spectre Lorentzien est donné par la relation suivante :

$$\frac{S_{\Delta N}(f)}{N^2} = \frac{S_R(f)}{R^2} = \frac{\overline{(\Delta N)^2}}{N^2} \frac{1}{1+\omega^2\tau^2} \quad \text{(I.6)}$$

I.6.4 Le bruit de scintillation

Les autres dénominations du bruit de scintillation sont le bruit de papillotement, ou le « Flicker noise » ou encore bruit fondamental en 1/f [11]. Le bruit de scintillation a été observé expérimentalement sur une grande variété de composants et de dispositifs électroniques, linéaires et non linéaires, incluant des résistances carbones, des diodes à

jonctions pn, des transistors bipolaires et des transistors à effet de champ. Ce type de bruit est associé à un courant continu passant dans le composant. Une expérimentation simple pour observer ce bruit est la suivante : lorsqu'une tension continue est appliquée à une résistance carbone, une composante du bruit, dépendant de la fréquence, dite en excès, est observée dans le courant, superposée au bruit thermique, qui lui est toujours présent. De même, quand un courant continu traverse une résistance carbone, il apparaît une composante en excès dans la tension, superposée au bruit thermique. Ces composantes du bruit en excès, qui sont observables dans la plupart des résistances en présence d'un courant continu ou d'une tension continue (analogique), présentent une densité spectrale de puissance qui varie en $|f|^{-\alpha}$, où α est quasi constant et est habituellement compris entre 0,8 et 1,4. Sa densité spectrale est de la forme :

$$S_i(f) = 4q\,|\,I_S\,| \qquad \text{(I.7)}$$

Néanmoins, les mécanismes précis impliqués dans ce type de bruit sont complexes, varient énormément d'un dispositif à l'autre, et font l'objet de spéculations et de controverses. Toutefois, ce type de bruit se manifeste comme une fluctuation de la conductivité, ce point faisant l'unanimité. Dans un échantillon ohmique de résistance R, cela se traduit soit par une fluctuation de la tension lorsqu'un courant constant la traverse, soit par une fluctuation de courant lorsqu'une tension continue est appliquée.

Le bruit en 1/f est une fluctuation de la conductivité (notée σ). Mais la conductivité est définie par le produit du nombre de porteurs de charge n, et de la mobilité μ, à la charge élémentaire près :

$$n.q.\mu = \sigma \qquad \text{(I.8)}$$

La question qui se pose est donc : « Qu'est ce qui fluctue avec un spectre en 1/f, le nombre de porteurs ou la mobilité ? ». A l'heure actuelle, aucune réponse n'a été donnée, ou plus exactement aucun fait n'a permis d'écarter une des deux possibilités. Comme le fait remarquer Hooge dans son article « 1/f Noise Sources » de 1994, les deux types de bruit, bruit de fluctuation de mobilité (noté $\Delta\mu$) et bruit de fluctuation du nombre de porteurs (noté Δn), ont été observés sur des transistors de type MOS : bruit $\Delta\mu$ dans les transistors à canal P et bruit Δn dans les transistors à canal N.

Il semblerait donc qu'il existe différents types de bruit en 1/f, requérant différentes théories afin d'expliquer tous les faits expérimentaux [12]. Il existe également d'autres types de bruit électronique ; tels le bruit de quantification, le bruit RTS (Random

Telegraph Signal) [13], bruit d'avalanche (Zener), ..., etc. Mais du fait de leur non apparence dans le cas considéré, nous avons préféré de les omettre

I.7 Le phénomène du défaut balistique

On sait déjà que pour aboutir au spectre lors d'une mesure donnée, les impulsions électriques induites dans le détecteur doivent être amplifiées, mises en forme, et analysées. Le temps de montée des impulsions en sortie d'un préamplificateur de charges correspond au temps de transit des porteurs de charge dans le détecteur.

La mesure de l'amplitude est proportionnelle, à la perte de charges près, à l'énergie déposée dans le détecteur. Pour que cette amplitude soit totalement mesurée, le temps de mise en forme doit au moins être égal au temps de montée le plus long. Le temps de mise en forme, choisi suivant le bruit en sortie, est directement relié à la résolution spectrale.

Une constante de temps courte favorise le bruit série généralement causé par le bruit thermique du transistor à effet de champ du préamplificateur alors qu'une constante de temps longue favorise le bruit parallèle comme celui lié au courant de fuite par exemple. Alors un compromis doit être fait entre les deux valeurs pour faire un choix optimal et ceci dans le but de pallier au problème du défaut balistique dont le phénomène est explicité juste en bas. Pour arriver à expliquer ce phénomène nuisible aux mesures spectroscopiques, nous devons nous référer au schéma électronique simplifié du préamplificateur de charge, tel qu'il est habituellement utilisé pour les détecteurs à germanium.

La collection des charges créées par le rayonnement dans le cristal du détecteur induit un courant dans le circuit « détecteur préamplificateur ». Ce courant charge la capacité C_f du préamplificateur. Le condensateur se décharge à travers la résistance Rf. Dans cette description simple, le signal à la sortie du détecteur peut être exprimé par l'équation différentielle suivante [4]. Le courant de décharge est proportionnel à la tension, i.e. au niveau de charge dans le condensateur, à tout moment, y compris pendant la collection de charge. Une partie de la charge produite sera, donc, dissipée pendant la collection et le niveau de charge atteint par le condensateur restera légèrement inférieur à celui qui serait créé en absence de la résistance Rf. L'expression couramment utilisée pour cette différence est le « défaut balistique ». [14]

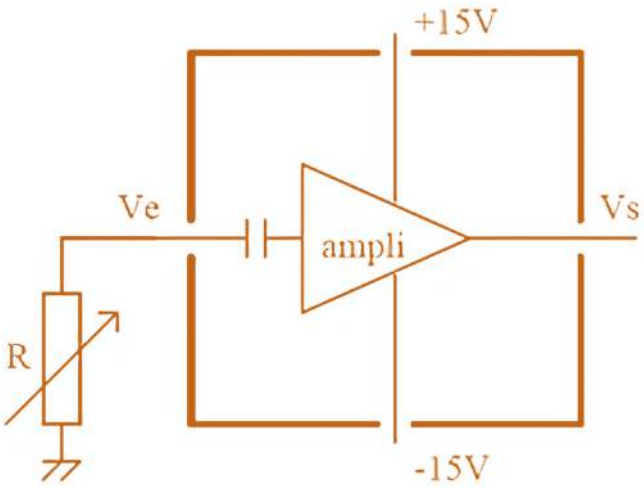


Fig.I.10 : Schéma électronique simplifié d'un préamplificateur de charge [35] .

I.8 Le phénomène d'empilement

Du fait du temps non nul de la collection de charges, le signal temporel dû à un seul photon, i.e. l'impulsion électrique qui lui est associée, est d'une durée courte mais non nulle, tout juste inférieure à la microseconde pour les détecteurs Germanium.

Cela signifie que deux photons peuvent être émis à des instants proches pour que leurs signaux respectifs se superposent partiellement ou en totalité [15]. En réalité, comme les instants d'arrivée des photons dans le détecteur forment un processus de Poisson, la probabilité qu'un empilement se produise n'est presque jamais nulle. Le phénomène provoque l'apparition des pics multiples dans l'histogramme des énergies. Il a été pris en considération lors de la mise au point des filtres de traitement numérique de l'impulsion pour amenuiser voire éliminer ses effets néfastes sur les spectres.

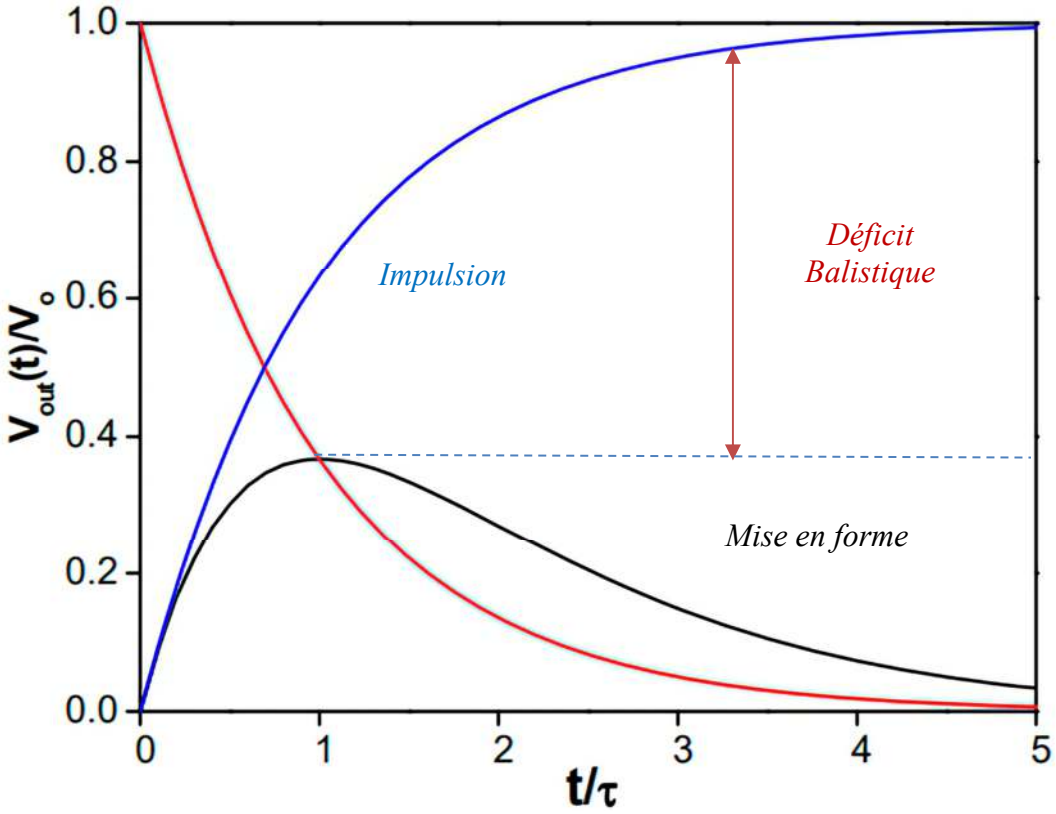


Fig.I.11 : Illustration de la perte balistique [35].

I.9 Conclusion

Au cours de ce chapitre, nous avons investi les phénomènes physiques qui se cachent derrière l'opération de mesure spectroscopique nucléaire et plus particulièrement la spectroscopie du rayonnement « Gamma ». On s'est attardé un peu plus sur les effets et les sources d'erreurs susceptibles de fausser l'allure du spectre fourni à la sortie d'une chaîne de mesure générique. L'accent a été précisément mis sur le bruit électronique qui règne dans une expérience typique, un fait dicté par la réalité stipulant que c'est le facteur déterminant et dont la connaissance et la bonne caractérisation jouent un grand rôle pour une conception soignée des filtres de traitement numériques qu'on a l'intention de développer dans la suite de ce manuscrit.

Chapitre II :

Généralités sur ZYNQ-7000 SoC et
Exploitation de l'IDE VIVADO Design Suite

II.1 Introduction

Jusqu'à une date assez récente, les FPGA n'avaient pas la capacité de portée nécessaire pour mettre en œuvre des algorithmes DSP exigeants et ne disposaient pas d'un bon support d'outils pour la mise en œuvre de tâches DSP. Ils étaient également perçus comme étant chers et gourmands en énergie. Tout cela pourrait changer, cependant, avec l'introduction de nouveaux produits orientés DSP d'Altera [16] et de Xilinx [17]. Le débit élevé et la souplesse de conception ont positionné les FPGA comme une solution silicium solide par rapport aux dispositifs DSP traditionnels dans les applications de traitement du signal à haute performance.

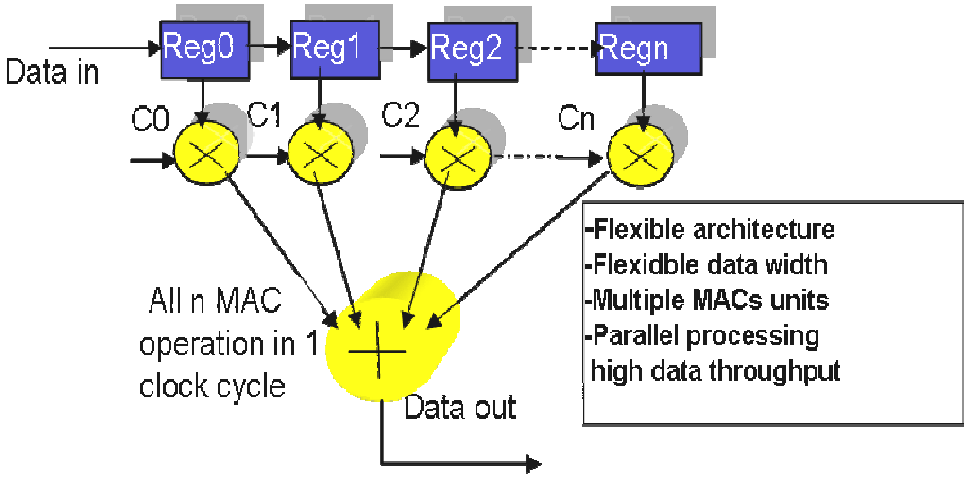


Fig.II.1 : Traitement du signal numérique dans le FPGA. [18]

II.2 Aperçu de la famille Zynq

La première chose à souligner est que tous les appareils Zynq 7000 partagent le même système de traitement, le même contrôleur de mémoire, les mêmes périphériques et les mêmes interfaces de la logique programmable. L'intention ici est de couvrir une plateforme de base commune qui permettra aux concepteurs de faire migrer leurs conceptions d'un dispositif à un autre. La différence réside dans la logique de programmation, qui effectue plus ou moins de personnalisation en fonction de l'application ciblée, comme cela est utilisé dans les dispositifs Zynq pour obtenir plus de mémoires intégrées et plus de blocs DSP, et également plus d'E/S pour connecter des dispositifs externes [19].

		Cost-Optimized Devices						Mid-Range Devices			
	Device Name	Z-7007S	Z-7012S	Z-7014S	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	XC7Z007S	XC7Z012S	XC7Z014S	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Processing System (PS)	Processor Core	Single-Core ARM® Cortex™-A9 MPCore™ Up to 766MHz			Dual-Core ARM Cortex-A9 MPCore Up to 866MHz			Dual-Core ARM Cortex-A9 MPCore Up to 1GHz[1]			
	Processor Extensions	NEON™ SIMD Engine and Single/Double Precision Floating Point Unit per processor									
	L1 Cache	32KB Instruction, 32KB Data per processor									
	L2 Cache	512KB									
	On-Chip Memory	256KB									
	External Memory Support[2]	DDR3, DDR3L, DDR2, LPDDR2									
	External Static Memory Support[2]	2x Quad-SPI, NAND, NOR									
	DMA Channels	8 (4 dedicated to PL)									
	Peripherals	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO									
	Peripherals w/ built-in DMA[2]	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO									
	Security[3]	RSA Authentication of First Stage Boot Loader, AES and SHA 256b Decryption and Authentication for Secure Boot									
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)	2x AXI 32b Master, 2x AXI 32b Slave 4x AXI 64b/32b Memory AXI 64b ACP 16 Interrupts									
Programmable Logic (PL)	7 Series PL Equivalent	Artix®-7	Artix-7	Artix-7	Artix-7	Artix-7	Artix-7	Kintex®-7	Kintex-7	Kintex-7	Kintex-7
	Logic Cells	23K	55K	65K	28K	74K	85K	125K	275K	350K	444K
	Look-Up Tables (LUTs)	14,400	34,400	40,600	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	28,800	68,800	81,200	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Total Block RAM (# 36Kb Blocks)	1.8Mb (50)	2.5Mb (72)	3.8Mb (107)	2.1Mb (60)	3.3Mb (95)	4.9Mb (140)	9.3Mb (265)	17.6Mb (500)	19.1Mb (545)	26.5Mb (755)
	DSP Slices	66	120	170	80	160	220	400	900	900	2,020
	PCI Express®	—	Gen2 x4	—	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC[2]	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs									
	Security[3]	AES & SHA 256b Decryption & Authentication for Secure Programmable Logic Config									
	Speed Grades Commercial	-1			-1			-1			-1
	Speed Grades Extended	-2			-2,-3			-2,-3			-2
	Speed Grades Industrial	-1, -2			-1, -2, -1L			-1, -2, -2L			-1, -2, -2L

Fig.II.2 : Aperçu de la famille Zynq [37].

II.2.1 Zynq-7000 System On-Chip

Le Zynq combine un processeur ARM à double cœur avec le traditionnel réseau de portes programmables par l'utilisateur. Le concept de combinaison d'ARM et de FPGA existe depuis un certain temps, et l'objectif est qu'une seule puce de silicium puisse être utilisée pour mettre en œuvre la fonctionnalité d'un système entier plutôt que de nécessiter plusieurs puces physiques différentes. C'est le concept de système sur puce présenté dans la figure II.4. Au lieu d'avoir toutes les fonctions requises dans des systèmes physiquement séparés, le SoC les combine dans un seul dispositif. Cette combinaison s'est avérée flexible et constitue la plate-forme de compilation pour une grande variété d'applications dont nous allons parler. En fait, le nombre de raisons de préférer les SoCs aux systèmes équivalents à composants discrets.

- La solution SoC permet un transfert de données plus rapide et plus sûr entre les mêmes éléments du système a une vitesse globale du système plus élevée.
- Consommation d'énergie plus faible, taille physique réduite, meilleure fiabilité et coût plus bas. Le cortex ARM doit fournir un processeur capable d'exécuter un système d'exploitation complet.

Le FPGA est basé sur l'architecture de la série 7 de Xilinx et fournit un dispositif flexible qui peut être configuré pour implémenter n'importe quel système arbitraire qui peut inclure des systèmes embarqués si nécessaire.

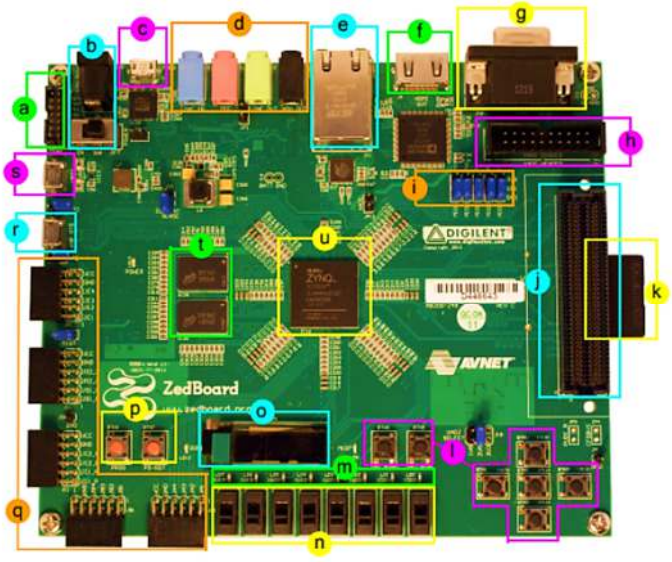
- Les FPGA ont trois rôles essentiels dans l'architecture Zynq.

- Rapidité de mise sur le marché
- Flexibilité et la possibilité de mettre à niveau.

Maximum Capability	ARTIX-7 Lowest Power and Cost	KINTEX-7 Industry's Best Price/Performance	VIRTEX-7 Industry's Highest System Performance	ZYNQ-7000 Extensible Processing Platform
Logic Cells	20K – 355K	70K – 480K	285K – 2,000K	30K – 350K
Block RAM	12 Mb	34 Mb	65 Mb	240KB – 2180KB
DSP Slices	40 – 700	240 – 1,920	700 – 3,960	80 – 900
Peak DSP Perf.	504 GMACS	2,450 GMACs	5,053 GMACS	1080 GMACS
Transceivers	4	32	88	16
Transceiver Performance	3.75Gbps	6.6Gbps and 12.5Gbps	12.5Gbps, 13.1Gbps and 28Gbps	6.6Gbps and 12.5Gbps
Memory Performance	1066Mbps	1866Mbps	1866Mbps	1333Mbps
I/O Pins	450	500	1,200	372
I/O Voltages	3.3V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below

Fig.II.3 : La famille de l'FPGA [38].

L'architecture est complétée par des interfaces AXI de l'industrie qui fournissent une connexion à faible latence entre deux parties du dispositif. La figure II.4 est la vue d'ensemble du kit Zedboard, le dispositif Zynq étant mis en encadré en jaune.



a Xilinx JTAG connector
b Power input and switch
c USB-JTAG (programming)
d Audio ports
e Ethernet port
f HDMI port (output)
g VGA port
h XADC header port
i Configuration jumpers
j FMC connector
k SD card (underside)
l User push buttons
m LEDs
n Switches
o OLED display
p Prog & reset push buttons
q 5 x Pmod connector ports
r USB-OTG peripheral port
s USB-UART port
t DDR3 memory
u Zynq device (+ heatsink)

Fig.II.4 : Vue d'ensemble du kit ZedBoard [18].

II.2.2 Relation du Zynq 7000 avec Zedboard

La ZedBoard est un kit de développement complet destiné aux concepteurs désireux d'explorer les conceptions utilisant le SoC programmable Xilinx Zynq 7000. La carte contient toutes les interfaces et fonctions de support nécessaires pour permettre une large gamme d'applications. Il existe plusieurs autres cartes compatibles avec la série Zynq-7000 SoC, ce qui la rend encore plus attrayante pour un large éventail d'applications. Pour cette présentation, nous allons nous concentrer sur l'architecture Zynq. La famille Zynq-7000 est basée sur l'architecture programmable de Xilinx.

Ce produit intègre le système de traitement de base ARM cortex A9 à cœur unique situé au niveau du système de traitement qui sera désigné par PS, et la logique programmable de 28 nm de Xilinx qui sera désignée par PL dans un seul dispositif. Les caractéristiques du CPU ARM cortex A9 comprennent une mémoire sur puce, des interfaces externes et un riche ensemble d'interfaces de connectivité périphérique.

II.2.3 Domaines d'application du Zynq 7000

Maintenant que nous avons réalisé l'architecture du Zynq et mis en évidence les composants clés et les caractéristiques du processeur, disons la dernière clé générale d'un système applicable pour lequel le Zynq peut être utilisé. Le processeur Zynq peut être utilisé dans l'automobile, l'armée, l'aérospatiale, le traitement de l'image, les communications avec et sans fil, la médecine, le contrôle industriel, le traitement du signal et bien d'autres exemples. La principale raison pour laquelle le Zynq peut être utilisé dans une large gamme de produits est que l'architecture Zynq peut être utilisée pour répondre à un grand nombre d'applications ayant des exigences importantes en termes de calcul haute performance et de traitement séquentiel en termes de fonctionnalité [19].

II.2.4 Raisons techniques de l'utilisation du Processeur Zynq

Les FPGA constituent un bon choix pour la mise en œuvre de systèmes numériques car ils peuvent : Inclure des cœurs de processeurs embarqués. Il peut s'agir de microprocesseurs durs implémentés en tant que blocs prédéfinis dédiés, tels que PowerPC dans les FPGA de Xilinx FPGA de Xilinx, ou des blocs IP de microprocesseurs logiciels tels que Microblaze.

- Supporter des schémas d'horloge complexes en utilisant des boucles à verrouillage de délai et des boucles à verrouillage de phase (PLL) intégrées.

- Inclure des multiplicateurs, des additionneurs et des blocs de multiplication et d'accumulation (MAC) intégrés, qui sont utiles pour construire des processeurs à grande vitesse massivement parallèles et/ou pipelines (réseaux systoliques).
- Offrir une grande capacité de stockage dans les blocs de RAM intégrés, en plus des mémoires de table de consultation distribuées.
- Offrir une grande capacité logique, dépassant 5 millions de portes système.
- Offrir un grand nombre de broches d'entrée/sortie à usage général (jusqu'à 1000 ou plus) et prendre en charge des protocoles série à haut débit.
- Supporter un large éventail de normes d'interconnexion, telles que la mémoire à double débit (DDR SRAM) et le PCI.
- Inclure des blocs émetteurs-récepteurs câblés spéciaux tels que RocketIO dans les FPGA de Xilinx, qui permettent une connectivité série gigabit entre les bus, les fonds de panier et les sous-systèmes.

En plus des caractéristiques ci-dessus [19], les FPGA offrent un avantage significatif en tant que puces qui sont programmées par l'utilisateur et peuvent être reprogrammées autant de fois que nécessaire pour apporter des modifications ou corriger des erreurs.

II.2.5 Comparaison Zynq par rapport à une combinaison FPGA/process discret

La dernière architecture à comparer avec le Zynq est celle d'une combinaison processeur- FPGA, c'est-à-dire où ces deux éléments existent en tant que composants physiquement séparés. La motivation pour ce type de système est généralement que le système doit supporter à la fois un traitement de type flux de données à forte intensité de calcul (idéalement adapté au FPGA), et des algorithmes ou applications logicielles sophistiqués (idéalement adaptés à un processeur dédié). Il se peut que l'élément logiciel du système dépasse la portée de l'implémentation MicroBlaze, ce qui signifie qu'un FPGA et un processeur sont tous deux nécessaires. Le Zynq offre la possibilité de remplacer cette configuration par une seule puce [19]. Du point de vue de la conception, Zynq offre également la possibilité de réaliser des gains de productivité, ce qui permet d'accélérer les temps de développement. Cela peut être attribué au flux de conception intégré et à la suite d'outils de développement de logiciels pour Zynq, qui sont basés sur une philosophie de conception au niveau du système, en tirant parti de la réutilisation de la conception et de la synthèse rapide et de haut niveau des algorithmes basés sur le langage C. D'autre part, l'alternative pour un système à composants discrets pourrait être de deux flux de conception, ensembles d'outils et processus très distincts.

Il est également significatif que le flux de conception Zynq prenne en charge un ensemble d'interfaces AXI entre le PL et le PS, ce qui supprime une partie de conception et de mise en œuvre des interfaces appropriées.

La grande disponibilité d'IP tiers compatibles avec AXI est un autre avantage. Enfin, les frais généraux de communication entre le processeur et le FPGA sur des connexions externes sont évités, ce qui peut être la cause des contraintes de bande passante et de la latence accrue dans le modèle à deux puces. Les connexions internes du dispositif Zynq sont intrinsèquement plus sûres que les liaisons externes et, en fait, des fonctions de sécurité supplémentaires sont intégrées pour faciliter une séquence de démarrage sécurisée et pour empêcher toute falsification [19].

II.3 Modèle simplifié de l'architecture Zynq

Le Zynq est composé de plusieurs éléments mentionnés précédemment, de l'ARM A9 et du FPGA. Le système de traitement PS est formé d'un processeur ARM cortex A9 à double cœur, et la logique programmable PL est équivalent à celle d'un FPGA. Le système PL est idéal pour mettre en œuvre des sous-systèmes d'arithmétique logique et de flux de données à grande vitesse. Alors que le PS supporte des routines logicielles et des systèmes d'exploitation, ce qui signifie que les fonctionnalités globales de tout système de conception peuvent être créées de manière appropriée entre le matériel et le logiciel. Les liens entre le PS et le PL sont réalisés à l'aide de connexions AXI (Advanced extensible Interface) standard de l'industrie. Les communications entre les éléments du système se font par des interconnexions. Elles sont réalisées sous la forme de liaisons directes point à point ou de bus desservant plusieurs composants. Les périphériques sont des composants fonctionnels résidant à l'écart du processeur.

En général, ils remplissent une à trois fonctions. Les coprocesseurs sont des éléments qui implémentent le processeur primaire, généralement utilisés de manière optimisée pour des tâches réduites et contraints d'interagir avec des interfaces externes, par exemple en se connectant à des LEDs et des interrupteurs, ou peuvent être des éléments de mémoire supplémentaires.

Le PS a une architecture fixe et héberge le processeur et les mémoires du système tandis que le PL est complètement flexible, donnant au concepteur une Caneva vierge pour créer des périphériques personnalisés ou utiliser des périphériques standard. Il est important de savoir que le zynq est un SoC, ce qui signifie que la grande variété de propriétés intellectuelles standard est disponible.

Cela signifie qu'il n'est pas nécessaire de redessiner ce composant. Le PS et le PL peuvent être utilisés indépendamment l'un de l'autre. En fait, les circuits d'alimentation sont configurés dans des domaines séparés, ce qui permet au PS ou au PL de se mettre hors tension s'il n'est pas utilisé. Sur le côté inférieur gauche de l'architecture Zynq, nous pouvons remarquer la présence d'un processeur microblaze optionnel.

Le rôle du microblaze est de coordonner des fonctions spécifiques de bas niveau dans les tâches les moins exigeantes du système qui pourraient être déléguées au processeur principal ARM cortex A9 afin d'améliorer les performances globales. Nous allons maintenant présenter le PS et le PL individuellement et comprendre chacune de leurs tâches, caractéristiques et propriétés. Commençons par le PS.

II.3.1 Entrée/Sortie à usage général

Les entrées-sorties à usage général sont collectivement appelées ressources d'E/S sélectionnées et sont organisées en banques de 50 IOB. Chaque IOB contient un pad qui fournit la connexion physique au monde extérieur pour une sortie ou une entrée unique. Les banques d'E/S sont classées en deux catégories : haute performance (HP) et haut débit (HR). Elles supportent une variété de normes d'E/S et de tensions, comme indiqué en détail. Les interfaces HP sont limitées à des tensions de 1,8 V et sont typiquement utilisées pour des interfaces à haute vitesse vers la mémoire et d'autres puces, tandis que les HR sont des interfaces qui permettent des tensions allant jusqu'à 3,3 V et couvrent une grande variété de normes d'E/S. Ces deux interfaces sont prises en charge et nécessitent respectivement une IOB et deux IOB pour la connexion [19].

II.3.2 Système de traitement PS

Tous les dispositifs Zynq ont la même architecture de base, et tous contiennent, comme base du système de traitement (PS), un processeur ARM Cortex-A9 à double cœur. Il s'agit d'un processeur "hard", il existe en tant qu'élément de silicium dédié et optimisé sur le dispositif. À des fins de comparaison, l'alternative à un processeur hard est un processeur "soft" comme le MicroBlaze de Xilinx, qui est formé en combinant des éléments de la matrice logique programmable [20].

La mise en œuvre d'un processeur logiciel est donc l'équivalent de tout autre bloc IP déployé dans la matrice logique d'un FPGA. En général, l'avantage des processeurs logiciels est que le nombre et l'implémentation précise des instances du processeur sont flexibles. D'autre part, les processeurs hard peuvent atteindre des performances considérablement plus élevées, comme c'est le cas du processeur ARM du Zynq [19].

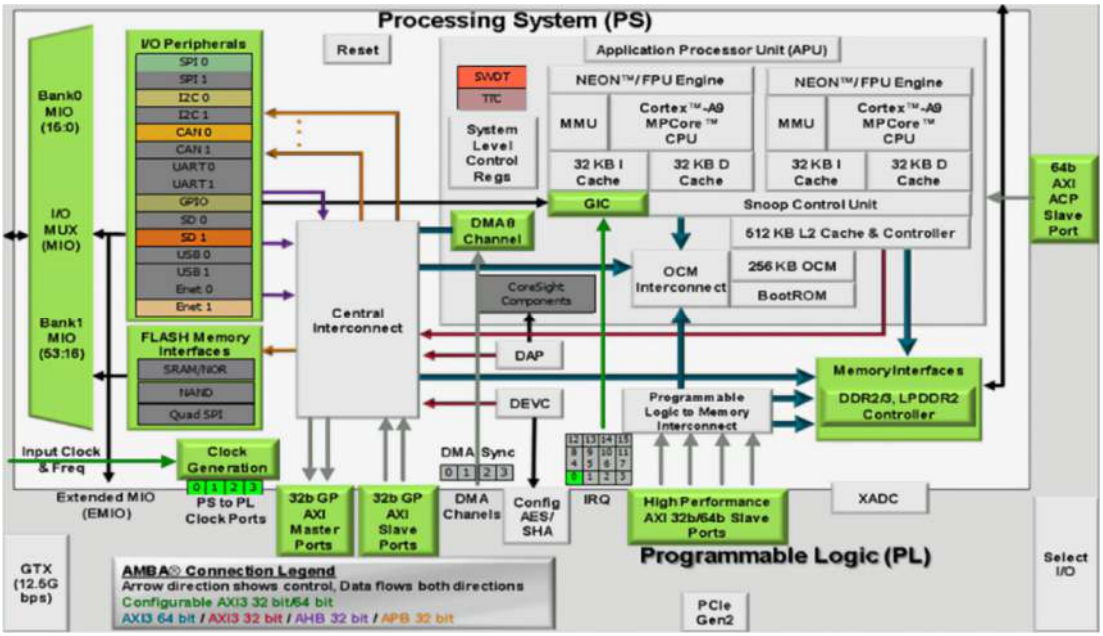


Fig.II.5 : Système de traitement du Zynq [39].

II.3.2.1 Unité de traitement des applications

L'APU se compose principalement de deux cœurs de traitement ARM, chacun avec des unités de calcul associées : un moteur de traitement multimédia (MPE) et une unité à virgule flottante (FPU) ; une unité de gestion de la mémoire (MMU) ; et une mémoire cache de niveau 1 (en deux sections pour les instructions et les données). L'APU contient également une mémoire cache de niveau 2 et une autre mémoire sur puce (OCM). La figure II.6 présente un schéma fonctionnel simplifié de l'APU.

Enfin, une unité de contrôle Snoop (SCU) forme un pont entre les cœurs ARM et les mémoires cache et OCM de niveau 2 ; cette unité a également une certaine responsabilité dans l'interface avec le PL.

En tant que fonctionnalité supplémentaire au processeur ARM principal, le NEON engine fournit des facilités SIMD (Single Instruction Multiple Data) pour permettre une accélération stratégique des algorithmes de type média et DSP [21]. Les instructions NEON engine sont une extension du jeu d'instructions ARM standard et peuvent être utilisées soit explicitement, soit en s'assurant que le code C suit une forme attendue et permet ainsi aux opérations NEON d'être déduites par le compilateur [22].

Il existe deux registres d'entrée, A et B, qui contiennent chacun un ensemble de N vecteurs d'entrée individuels. Une opération définie unique est effectuée entre les N ensembles de vecteurs d'entrée pour produire un ensemble correspondant de vecteurs de sortie qui sont écrits dans le registre de sortie. La taille des vecteurs peut varier, tout comme le nombre de vecteurs composant chaque registre ;

La caractéristique importante est que chaque "voie" produit des résultats découlant de la même opération, qui est effectuée sur plusieurs ensembles différents d'entrées en même temps (instruction unique, données multiples). [19]

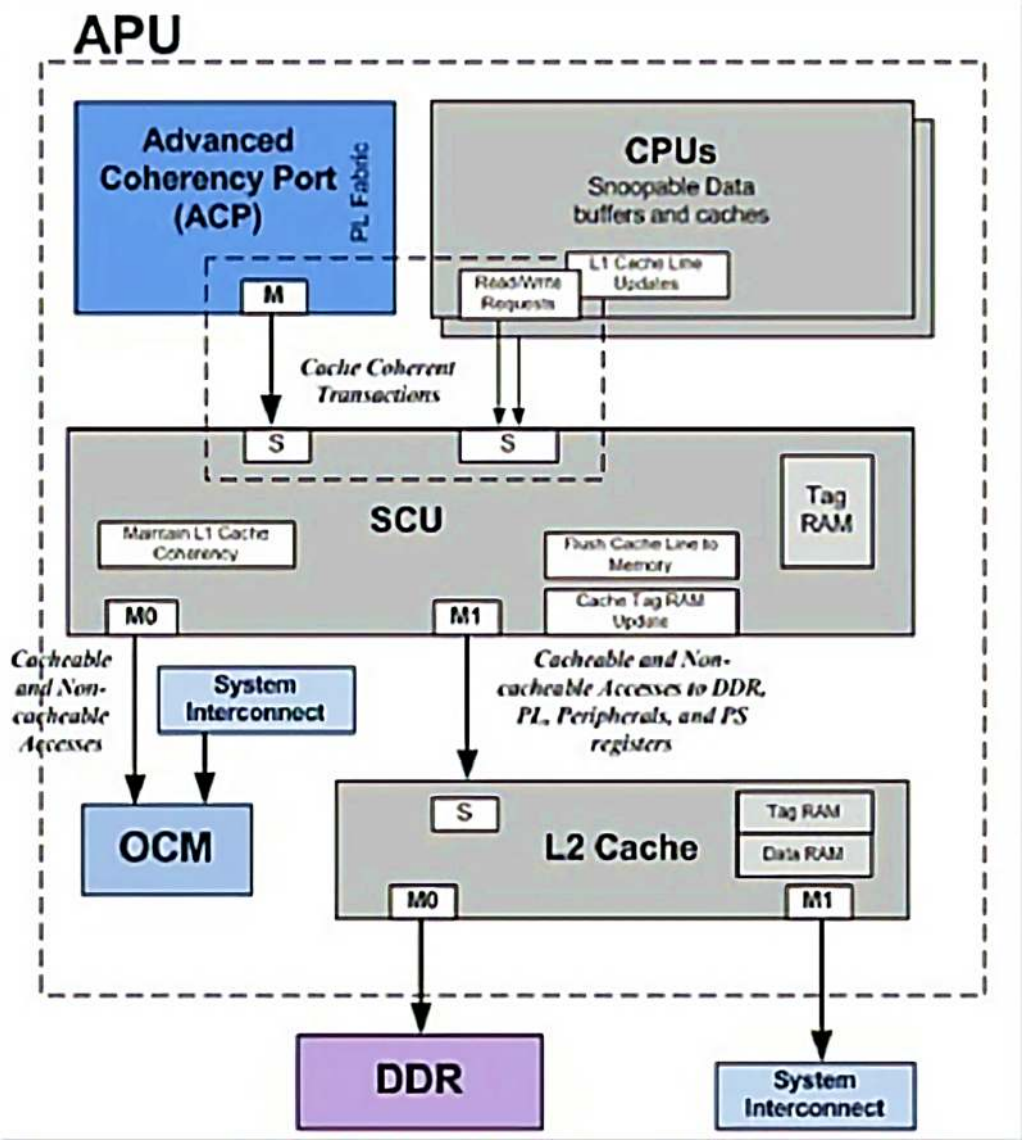


Fig.II.6 : Diagramme de l'unité de traitement des applications (APU) [38].

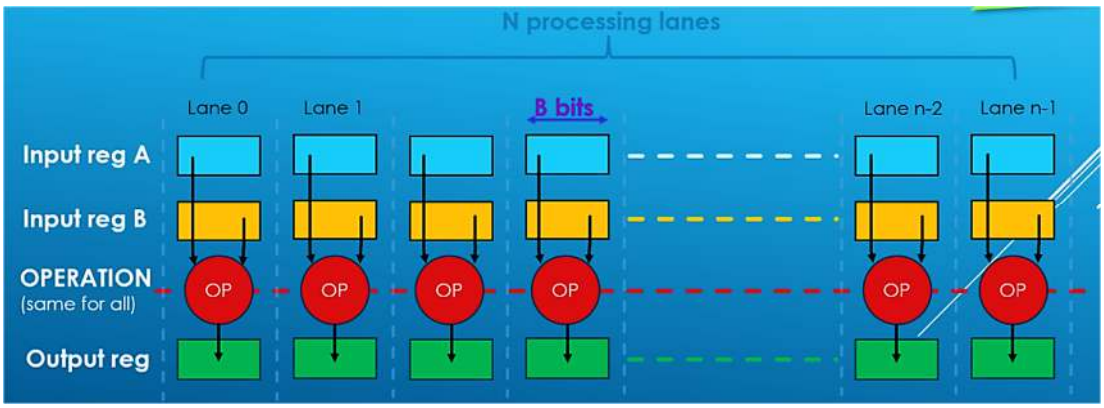


Fig.II.7 : Traitement SIMD dans le MPE de NEON [40].

II.3.2.2 Performances de traitement

La fréquence maximale atteinte par MicroBlaze dépend de sa configuration et de certains autres facteurs tels que le placement et le routage sur le PL. Une configuration MicroBlaze typique peut atteindre environ 70 % de la fréquence maximale du PL, ce qui équivaut à deux ou trois cents MHz à titre de comparaison, la fréquence maximale de fonctionnement des processeurs ARM est de 800 MHz à 1 GHz.

Les performances du processeur sont évaluées à l'aide d'un benchmark, afin de comparer l'ARM Cortex A9 et le MicroBlaze : 1-DMIPS (Dhrystone Millions of Instructions Per Second). La quantité DMIPS exprime le nombre d'opérations réalisées par seconde par le processeur lors de l'exécution de l'application de test Dhrystone standard. 2-Score CoreMark - Le score CoreMark établit un score numérique simple pour les performances des processeurs, qui peuvent être directement comparé aux scores d'autres processeurs. DMIPS et CoreMark sont des métriques utilisées pour quantifier la capacité de traitement. Selon la documentation de Xilinx, les trois configurations MicroBlaze énumérées ci-dessous ne peuvent atteindre plus de 260 DMIPs sur Zynq en grade de vitesse 3, alors que le processeur ARM à double cœur devrait atteindre 5000 DMIPs (2500 DMIPs par cœur), en supposant une fréquence d'horloge PS de 1 GHz. Cela signifie que le processeur ARM offre environ 20 fois plus de performances de traitement par rapport à un seul cœur MicroBlaze [19].

II.3.3 Logique de programmable PL

La logique programmable (PL) est basée sur la matrice FPGA. Maintenant que nous avons totalement couvert le système de traitement de la PS qui était le processeur ARM A9 cortex, parlons de la PL. Le PL est principalement composé du tissu logique général FPGA qui est tout d'abord composé de tranches et de blocs logiques configurables CLBs. Le PL a deux ressources spéciales connues sous le nom de DSP48A1s et de blocs de RAM. Enfin, le PL est composé d'entrées-sorties à usage général pour communiquer avec le monde extérieur. Discutons maintenant de chaque partie pour comprendre le tissu logique. Les blocs logiques configurables (CLB) sont de petits groupements réguliers d'éléments logiques disposés en réseau bidimensionnel sur le PL et connectés à d'autres ressources similaires par des interconnexions programmables. Chaque CLB est positionné à côté d'une matrice de commutateurs et contient 2 tranches logiques. Mais qu'est-ce qu'une tranche ?

Une tranche est la sous-unité au sein du CLB qui contient des ressources pour la mise en œuvre de circuits logiques combinatoires et séquentiels. Les tranches Zynq sont composées de 4 tables de consultation, de 8 flip-flops et d'autres logiques. La table de consultation LUT est une ressource flexible capable d'implémenter une fonction logique jusqu'à 6 entrées, une petite ROM, une petite RAM ou même un registre à décalage.

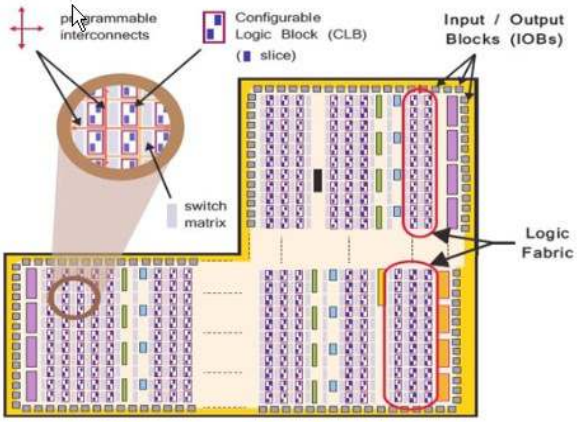


Fig.II.8 : Structure interne du PL [40].

Les LUT peuvent être combinées entre elles pour former une fonction logique plus importante, des mémoires ou des registres à décalage, selon les besoins. La bascule est un circuit à éléments séquentiels mettant en œuvre un registre de 1 bit avec des fonctionnalités de réinitialisation. L'une des fonctionnalités FF sera utilisée pour implémenter un latch. Une matrice de commutation située à côté de chaque CLB fournit des facilités de routage flexibles pour établir des connexions d'abord entre les éléments au sein des CLB ou d'une CLB à une autre ressource sur le PL.

La logique de report est un circuit arithmétique dont les signaux doivent être propagés entre des tranches adjacentes. Et ceci est réalisé par une logique de report. La logique de report comprend une chaîne de routes et de multiplexeurs le long des tranches en colonnes verticales. Les blocs d'entrée-sortie sont des ressources qui assurent l'interface entre les ressources logiques de l'API et le bloc de dispositifs physiques utilisé pour se connecter aux circuits externes. Chaque IOB peut gérer un signal d'entrée ou de sortie de 1 bit. Les IOB sont généralement situés sur le périmètre du dispositif.

Il y a 2 composants à usage spécial :

- ❖ Blocs de RAM pour répondre aux besoins en mémoire.
- ❖ DSP48E1s pour l'arithmétique haute vitesse.

Les blocs de RAM peuvent mettre en œuvre une mémoire vive, une mémoire morte et des tampons FIFO tout en prenant en charge le codage correcteur d'erreurs.

L'utilisation d'une mémoire vive en bloc signifie qu'une plus grande quantité de données peut être stockée dans une petite taille physique sur un dispositif dans un élément de mémoire dédié et optimisé. Le DSP48E1 utilise des circuits de multiplexage pour promouvoir une utilisation flexible des registres et pour supporter une modification dynamique du calcul. Différents calculs sont possibles, impliquant plusieurs opérateurs arithmétiques.

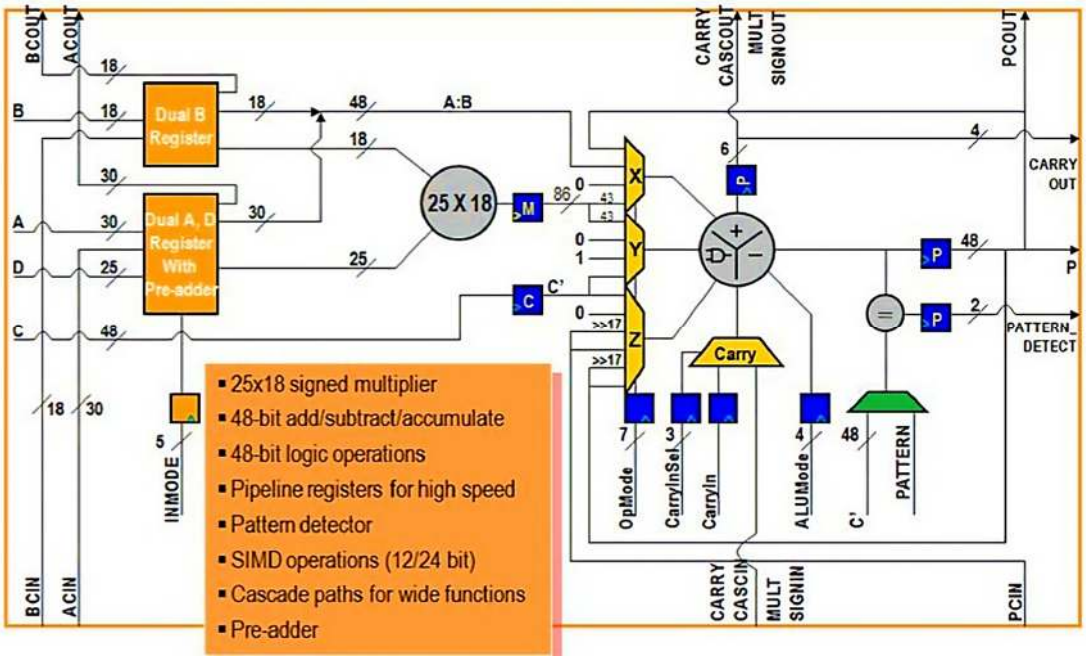


Fig.II.9 : Schéma bloc DSP48E1 [38].

Il est également capable d'un traitement SIMD mettant en œuvre l'addition, la soustraction et l'accumulation de 2 ou 4 bits plus courts, respectivement de 24 et 12 bits. Lorsqu'il est utilisé en mode logique, il peut exécuter des fonctions logiques au lieu d'arithmétiques et supporte toutes les opérations booléennes fondamentales : NOT, AND, OR, NAND, NOR, XOR et XNOR bit à bit.

La fonction principale du Zynq-7000 dans le prototype est d'assurer l'interface avec toute l'électronique frontale, à savoir le système de commande de Zynq-7000, les comparateurs et le ADC. En outre, PL doit être en mesure de communiquer avec le processeur, de recevoir et d'interpréter les commandes, ainsi que d'envoyer les données reçues de Zynq-7000. Le PL gère également l'ADC de la partie d'électronique frontale par le biais d'un contrôleur I2C. La carte Zynq-7000 possède trois interfaces distinctes, chacune ayant un protocole personnalisé : une interface de registre série, une interface d'échantillonnage et d'écriture, et une interface de numérisation et de lecture. De plus, le PL doit fournir trois horloges pour PS.

Un bloc firmware de contrôleur MicroBlaze a été développé, gère toutes les fonctionnalités des signaux d'entrée et de sortie du Zynq-7000. L'ADC échantillonne en permanence, les impulsions fournissant une horloge et une adresse pour stocker la tension analogique dans une mémoire interne. La mémoire interne du Zynq-7000 est à accès aléatoire et le FPGA a le contrôle total sur la façon de la gérer. La gestion du stockage de la mémoire est une tâche cruciale à des taux de déclenchement élevés pour éviter les temps morts ou l'écrasement de données utiles. Le FPGA est également responsable de la réception des données brutes de forme d'onde provenant du ADC. Le FPGA doit être prêt à recevoir et à désérialiser les données, à les emballer et à les envoyer à la RAM externe via une interface AXI Stream.

II.3.3.1 Interface de communication

L'interface de communication est un dispositif zynq hautement spécifié et comprend généralement l'émetteur-récepteur GTX, des blocs d'interface de communication à haut débit qui sont intégrés dans le tissu logique. Il s'agit de blocs de silicium dédiés, capables de prendre en charge un certain nombre d'interfaces standard, notamment PCI Express, RapidIO série, SCSI et SATA. Les interfaces peuvent être utilisées pour créer des connexions vers des dispositifs externes tels que des équipements de réseau et d'autres dispositifs FPGA.

II.3.3.2 Interface d'interconnexion AXI

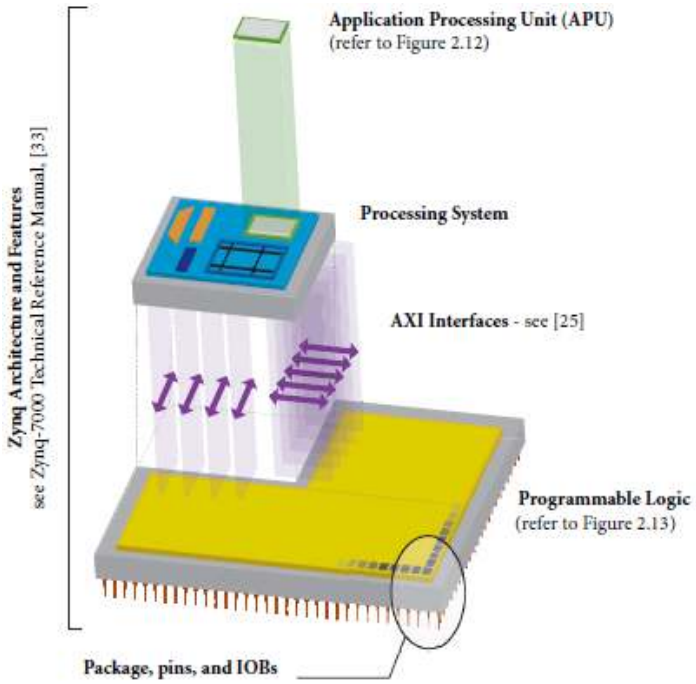


Fig.II.10 : Schéma de l'interface AXI.

Nous commençons par présenter la norme AXI sur laquelle la plupart de ces connexions sont basées. AXI signifie Advanced extensible Interface et la version actuelle est AXI 4, qui fait partie de la norme ARM AMBA. La norme AMBA n'est pas décrite par ARM comme "la norme de facto pour la communication sur puce". Les bus AXI peuvent être utilisés de manière flexible, et d'une manière générale sont utilisés pour contrôler le processeur et les systèmes embarqués. En fait, il existe trois types d'AXI4, chacune d'entre elles représentant un élément du bus différent et sont résumées ci-dessous. Le choix du protocole de bus AXI pour une connexion particulière dépend des propriétés de conception de cette connexion.

- **L'AXI4** est utilisé pour les flags de carte mémoire et offre une performance maximale et une alimentation ajustée, suivie d'un verset de données d'environ 256 mots de données.
- **AXI4-Lite** est une liaison simplifiée ne supportant qu'un seul transfert de données par connexion. AXI4-Lite est également mappé en mémoire, dans ce cas, une adresse de mots de données uniques est transférée.
- **AXI4-Stream** est utilisé pour la transmission de données en continu à grande vitesse et prend en charge les transferts de tranches de circuits.

Ce troisième type est le mieux adapté au flux de données direct d'adresse entre la source et la destination des liens non mappés de mémoire.

L'interface primaire entre le PS et le PL est réalisée par un ensemble de 9 interfaces AXI, chacune d'entre elles étant composée de plusieurs canaux. Celles-ci réalisent la connexion décadence au sein du PL et les interconnexions au sein du PS. Il est utile de définir brièvement ces deux termes importants : Interconnexion : une interconnexion est effectivement un commutateur qui gère et dirige le trafic entre les interfaces AXI. Il existe des interconnexions similaires au sein du PS, certaines étant directement interfacées avec le PL et d'autres étant réservées à un usage externe. Les connexions entre ces interconnexions se forment également à l'aide d'interfaces AXI. Une interface est une connexion point à point permettant de transmettre des données, des adresses et donc le signal entre le client maître et le client esclave dans le système.

Le bus AXI à usage général est un bus de données de 32 bits qui convient aux communications à faible et moyen débit entre le PL et le PS. L'interface est directe et ne comprend pas de mise en mémoire tampon. Il y a 4 interfaces d'usage général au total. Le PS est le maître de deux d'entre elles, et le PL est le maître de deux autres.

Le port de cohérence de l'accélérateur (ACP) est une connexion unique entre le PL et le PS dans l'APU. Avec une largeur de bus de 64 bits, ce port est utilisé pour assurer la cohérence entre les caches de l'APU et les éléments du PL, le PL étant le maître [19]. Il y a quatre interfaces AXI haute performance comprenant des tampons FIFO pour accumuler le meilleur comportement de raison et supporter la communication à haut débit entre les éléments PL et mémoires et le PS. La largeur des données est de 32 ou 64 bits et le PL est le maître des quatre interfaces.

Notez sur le schéma que toutes les interfaces sont spécifiquement connectées aux interconnexions AXI résidant dans le PS, à l'exception de l'interface ACP, qui est connectée directement à l'unité de contrôle Snoop dans l'APU. À l'intérieur du système de traitement, les interfaces AXI sont utilisées à la fois dans l'APU ARM (pour établir des connexions entre les cœurs de traitement et le SCU, la mémoire cache et l'OCM), et plus généralement pour connecter les diverses interconnexions dans le PS. Ces connexions s'ajoutent à celles de la frontière PS-PL. En particulier, les trois interconnexions illustrées à la figure II.10 (les interconnexions de mémoire, de maître et d'esclave) sont connectées en interne à l'APU [19].

II.3.3.3 Interface EMIO

Comme nous l'avons mentionné dans la section précédente, plusieurs connexions du PS peuvent être acheminées vers les interfaces externes du PL, et cela peut être fait par le MIO étendu ou EMIO.

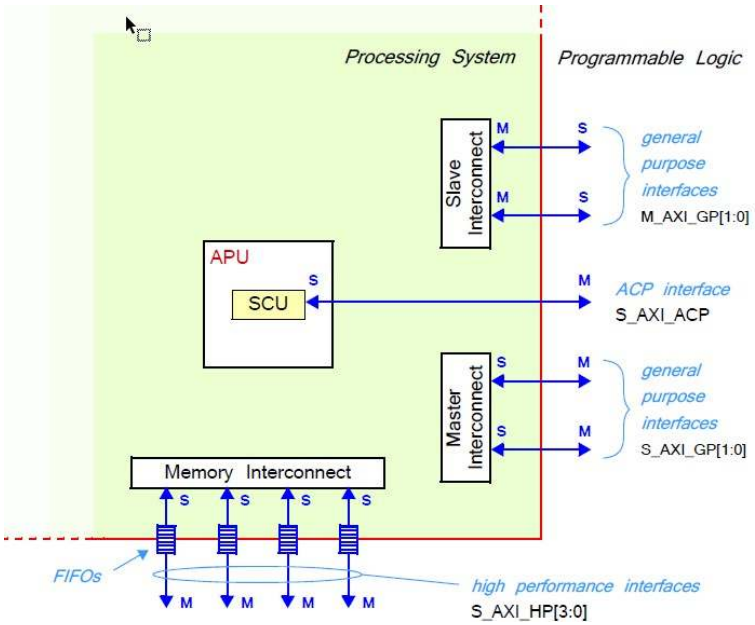


Fig.II.11 : Structure des interconnexions AXI et des interfaces reliant le PS et le PL [40].

EMIO implique le transfert de signaux entre les deux domaines, et ceci est réalisé par un simple ensemble de connexions câblées. Par conséquent, toutes les interfaces MIO ne sont pas prises en charge par EMIO. Et certaines de celles qui sont supportées ont des capacités réduites. Les connexions sont organisées en deux banques de 32 bits. Les interfaces acheminées par l'EMIO sont dans de nombreux cas connectées directement aux broches externes souhaitées de l'API, comme spécifié par l'inclusion d'entrées appropriées dans un fichier de contraintes. Dans ce mode, l'EMIO peut être divisé en 64 entrées et 64 sorties supplémentaires avec les activations de sortie correspondantes [19]. Une autre option consiste à utiliser EMIO :

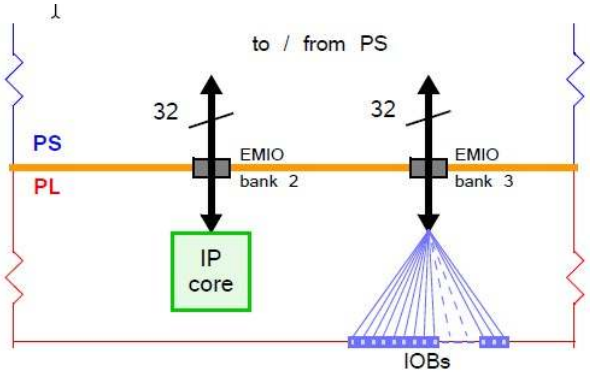


Fig.II.12 : L'interface EMIO [40].

II.3.3.4 Autres interfaces PL

D'autres interfaces logiques programmables sont la conversion analogique-numérique, le PL comprend un autre composant IP élevé, le bloc XADC. Il s'agit d'un ensemble de convertisseurs analogiques-numériques, de matériel à signaux mixtes, qui comporte deux outils distincts, avec un CAN capable d'échantillonner des signaux d'entrée analogiques externes à 1 MSPS. Le contrôleur du XADCA est une puce utilisant le bloc d'interface XADC situé dans le PS et le PS XADC qui est le bloc de contrôle du convertisseur analogique-numérique de traitement peut également être programmé à partir du logiciel exécutant l'APU. Le PL reçoit quatre entrées d'horloge distinctes du PS et a en outre la possibilité de générer et de distribuer ses propres signaux d'horloge indépendamment du PS. Les ressources indépendantes du PL sont équivalentes à celles des FPGA de la série 7. Un ensemble de JTAG fourni dans la section PL facilite la configuration et le débogage du PL.

Des méthodes plus sûres sont normalement préférées lors du déploiement. La configuration JTAG est souvent utilisée dans la phase de développement. Cela facilite le débogage du support JTAG overdrive à la fois sur les outils Xilinx [19].

II.4 L'environnement de développement intégré VIVADO

L'environnement de développement intégré (IDE) Vivado offre une interface utilisateur graphique (GUI) intuitive et de puissantes fonctionnalités. Tous les outils et options d'outils sont écrits dans le format natif du langage de commande d'outils (Tcl), ce qui permet de les utiliser à la fois dans l'IDE Vivado ou dans Vivado Design Suite Tcl-shell. L'analyse et l'affectation des contraintes sont activées tout au long du processus de conception. Par exemple, on peut effectuer des estimations de timing ou de puissance après la synthèse, comme la base de données est accessible par Tcl, les modifications des contraintes, de la configuration de la conception ou des paramètres des outils se font en temps réel, souvent sans nécessiter de ré-implémentation.

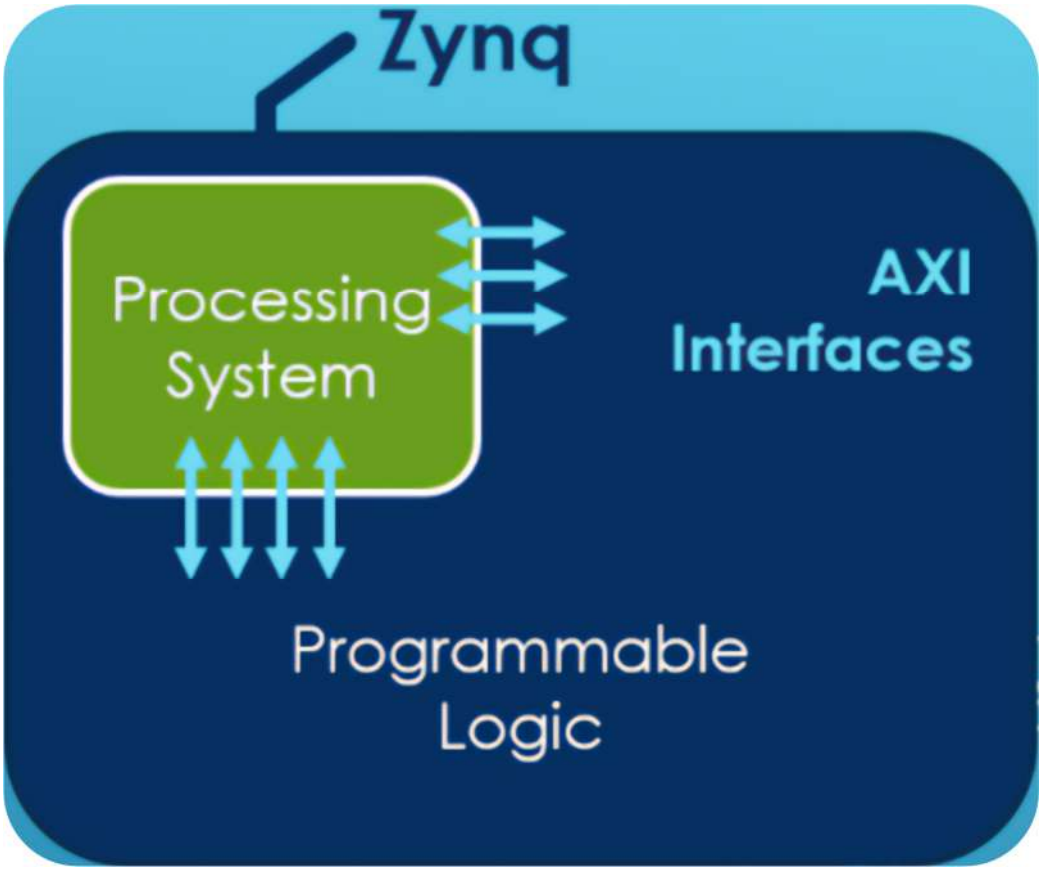


Fig.II.13 : Comparaison entre les combinaisons de systèmes de traitement et de PL .

L'IDE VIVADO utilise un concept d'ouverture des conceptions en mémoire. L'ouverture charge la liste de conception à cette étape particulière du flux, affecte les contraintes à la conception, puis applique la conception au dispositif cible. Cela permet de visualiser et d'interagir avec la conception à chaque étape de la conception.

La page de démarrage de l'IDE Vivado nous aide à créer et à ouvrir des projets, à exécuter les commandes de l'IDE Vivado et à consulter la documentation comme suit :

- **Create Project :** l'ouverture de l'assistant Nouveau projet va nous guider pour la création de divers types de projets pris en charge. On peut également utiliser l'assistant pour importer des projets précédemment créés depuis l'outil PlanAhead (extension .ppr) ou depuis ISE Design Suite (extension .xise).
- **Open Project :** Ouvre un navigateur qui nous permet d'ouvrir n'importe quel fichier de projet Vivado IDE (extension .xpre).
- **Open Example Project :** l'ouverture de l'assistant Ouvrir projet va nous guider pour la création de l'un des projets d'exemple suivants, qui comprend la spécification d'un nom et d'un emplacement de projet et le choix parmi une liste de parties valides :
- **BFT :** Petit projet RTL.

II.4.1 Principaux composants de l'environnement de visualisation

On peut interagir avec l'IDE Vivado par le biais de la souris, du clavier ou de l'interface Tcl. Les principaux composants de l'environnement de visualisation sont :

1. Barre de menu
2. Main Toolbar
3. Flow Navigator
4. Data Windows Area
5. Commande de menu Accès rapide Champ de recherche
6. Espace de travail
7. Barre d'état du projet
8. Layout Selector
9. Barre d'état
10. Zone de la fenêtre des résultats

Lorsque on ouvre l'IDE Vivado, la page de mise en route s'affiche comme indiquer sur la figure II-14. La disposition par défaut de l'environnement Vivado IDE est présentée à la figure II-14 (d'autres dispositions peuvent être choisies en sélectionnant différentes perspectives). Cette disposition est spécifiquement destinée à la Zedboard.

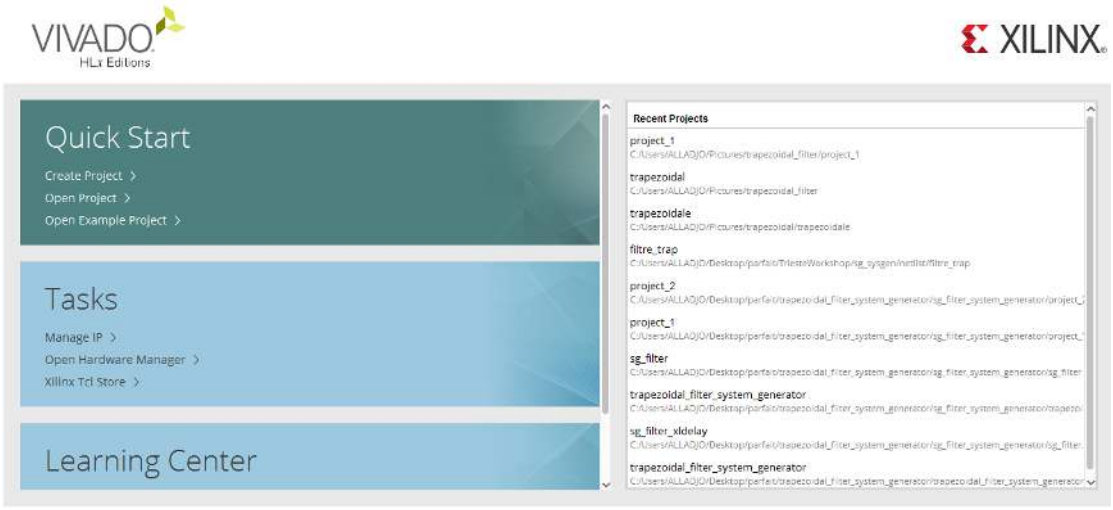


Fig.II.14 : Environnement de visualisation de l'IDE Vivado.

En référence aux étiquettes numérotées de la figure II.14, les principaux composants de l'environnement Vivado IDE sont les suivants :

1. **Barre de menu** : La barre d'accès principale permet d'accéder aux commandes de Vivado IDE.

2**. Barre d'outils principale** : La barre d'outils principale permet d'accéder facilement aux commandes les plus couramment utilisées de Vivado IDE. Des infobulles fournissent des informations sur chaque commande de la barre d'outils et celles-ci peuvent être visualisées en passant le pointeur de la souris sur les boutons.

3. **Flow Navigator** : Le Flow Navigator permet d'accéder facilement aux outils et aux commandes nécessaires pour guider votre conception du début à la fin, en commençant par la section Project Manager avec la saisie de la conception et en terminant par la génération du flux binaire dans la section Program and Debug. Des commandes d'exécution sont disponibles dans les sections Simulation, Synthèse et Implémentation pour simuler, synthétiser et implémenter la conception active.

4. **Volet Fenêtres de données** : Par défaut, le volet Fenêtres de données affiche les informations relatives aux données et sources de conception, notamment les données et les sources de conception, notamment :

- **Fenêtre Propriétés :** Affiche des informations sur les objets logiques ou les ressources de périphérique sélectionnés.
- **Fenêtre Netlist :** Fournit une vue hiérarchique de la conception logique synthétisée ou élaborée.
- **Fenêtre Sources :** Affiche les vues Sources IP, Hiérarchie, Bibliothèques et Ordre de compilation.

5. **Barre d'état :** La barre d'état affiche une variété d'informations, notamment :

- Des informations détaillées sur les commandes de la barre de menu et de la barre d'outils s'affichent dans le Les informations détaillées concernant les commandes de la barre de menu et de la barre d'outils s'affichent dans la partie inférieure gauche de la barre d'état lorsque la commande est accessible.
- Lorsque on survole un objet dans la fenêtre Schématique avec le pointeur de la souris, les détails de l'objet apparaissent dans la barre d'état.
- Pendant la création de contraintes et de placements dans les fenêtres Device et Package, la validité et le type de contrainte sont affichés dans la barre d'état, le type de contrainte s'affichent sur le côté gauche de la barre d'état, les coordonnées et le type de site s'affichent dans la partie droite.
- La progression d'une tâche en cours d'exécution est déplacée vers le côté droit de la barre d'état lorsque le bouton Arrière-plan est sélectionné.

6. **Layout Selector :** L'IDE Vivado fournit des dispositions de fenêtres prédéfinies pour faciliter diverses tâches du processus de conception. Le sélecteur de disposition permet de modifier facilement les dispositions des fenêtres. On peut également changer de disposition à l'aide du menu Layout de la barre de menus.

7. **Barre d'état** : La barre d'état du projet affiche les éléments suivants :

- L'état actuel du modèle actif
- Les descriptions des commandes de menu et des boutons de la barre d'outils.
- Les informations sur le fichier texte sélectionné que' on a modifié, y compris les numéros de ligne et les modes
- La position des éléments sur lesquels passer dans les fenêtres Device et Package.

8. **Zone de la fenêtre des résultats :** Zone de la fenêtre des résultats - La fenêtre des résultats affiche l'état et les résultats des commandes dans un ensemble de fenêtres regroupées dans la partie inférieure de l'environnement Vivado IDE. Au fur et à mesure que les commandes progressent, des messages sont générés et des fichiers journaux et des rapports sont créés. Les informations correspondantes sont affichées ici. Les fenêtres par défaut sont les suivantes :

- **Design Runs :** Gère les exécutions pour le projet en cours.
- **Messages :** Affiche tous les messages pour la conception active.
- **Tcl Console :** Les commandes Tcl peuvent être saisies ici et un historique des commandes précédentes et des sorties est également disponible.

- **Rapports :** Un accès rapide est fourni aux rapports générés tout au long du flux de conception.
- **Log :** Affiche les fichiers journaux générés par les processus de simulation, de synthèse et d'implémentation.

Les fenêtres supplémentaires qui peuvent apparaître dans cette zone selon les besoins sont les suivantes : La fenêtre Find Results, la fenêtre Timing Results et la fenêtre Package Pins. Après avoir présenté la structure de l'environnement Vivado IDE, nous pouvons maintenant passer à la création du système Zynq.Dans la fenêtre Flow Navigator, sélectionnez Create Block Design dans la section IP Integrator. Cliquez sur OK. Le canevas de diagramme de Vivado IP Integrator s'ouvre dans l'espace de travail. Le premier bloc que nous allons ajouter à notre conception sera un système de traitement Zynq. Dans le canevas Vivado IP Integrator Diagram, cliquez avec le bouton droit de la souris n'importe où et sélectionnez Add IP (Ajouter une IP).

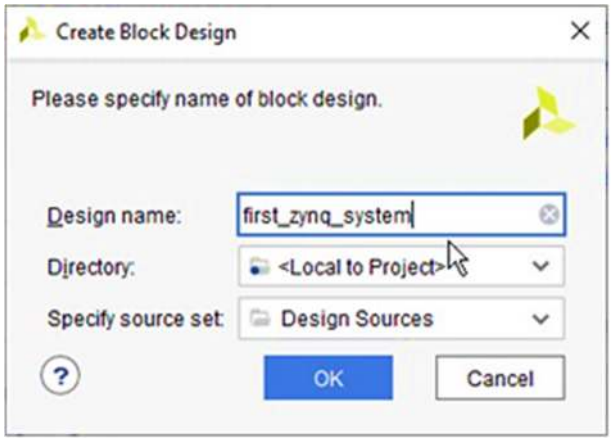


Fig.II.15 : Créer un dialogue de conception de blocs.

On peut également sélectionner le bouton Add IP dans la barre d'outils située à gauche du canevas. La fenêtre pop-up du catalogue IP s'ouvre. Entrez Zynq dans le champ de recherche et sélectionnez le système de Zynq Processing, puis appuyez sur la touche Entrée de clavier. Sélectionnez l'option Run Block Automation (Exécuter l'automatisation du bloc) dans le message d'assistance du concepteur en haut de la fenêtre Diagramme.

Sélectionnez OK, en assurant que l'option Appliquer le préréglage de la carte est sélectionnée, afin de générer les connexions externes pour les interfaces DDR et FIXED_IO, et d'appliquer les préréglages de la carte correspondants. Comme la plate-forme Zedboard est la carte de développement cible, et que cela a été spécifié lors de la création du projet, Vivado configurera le bloc processeur Zynq en conséquence.

Dans la boîte de dialogue Run Block Automation, on assure que l'option Apply Board Preset est sélectionnée et cliquez sur OK. Les connexions externes pour les interfaces DDR et FIXED_IO sont maintenant générées. Maintenant que le PS Zynq principal a été ajouté à notre conception et configuré, nous pouvons ajouter d'autres blocs qui seront placés dans le PL pour ajouter des fonctionnalités au système. Dans ce cas, nous avons besoin d'un bloc GPIO AXI pour les LEDs et d'un autre pour les boutons poussoirs.

II.4.1.1 Éditeur de texte

L'éditeur de texte de l'IDE de Vivado est un éditeur de texte intégré et configurable qui prend en charge la mise en évidence de la syntaxe, la vérification de la syntaxe à la volée, l'assistance en cas d'erreurs et d'avertissements, le pliage de code, la complétion de code et la comparaison de fichiers. L'éditeur de texte prend en charge les types de fichiers suivants :

- Fichiers Verilog et d'en-tête Verilog
- Fichiers SystemVerilog
- Fichiers VHDL
- Fichiers de contraintes
- Scripts Tcl
- Journal et fichiers journaux de l'IDE Vivado
- Fichiers texte simples

L'éditeur de texte prend en charge les fonctionnalités suivantes :

- Contrôle syntaxique à la volée
- Assistance en cas d'erreurs et d'avertissements
- Compléter le code

On peut générer une fenêtre Schematic pour tout niveau de la hiérarchie logique ou physique. On peut sélectionner un élément logique dans une fenêtre ouverte, comme une primitive ou un réseau dans la fenêtre Netlist, et utiliser la commande Schematic dans le menu contextuel pour créer une fenêtre Schematic pour l'objet sélectionné. Une conception élaborée s'ouvre toujours avec une fenêtre schématique du niveau supérieur de la conception.

Dans la fenêtre Schematic, on peut visualiser l'interconnexion de la conception, la structure hiérarchique ou tracer les chemins de signaux pour la conception élaborée, la conception synthétisée ou la conception implémentée.

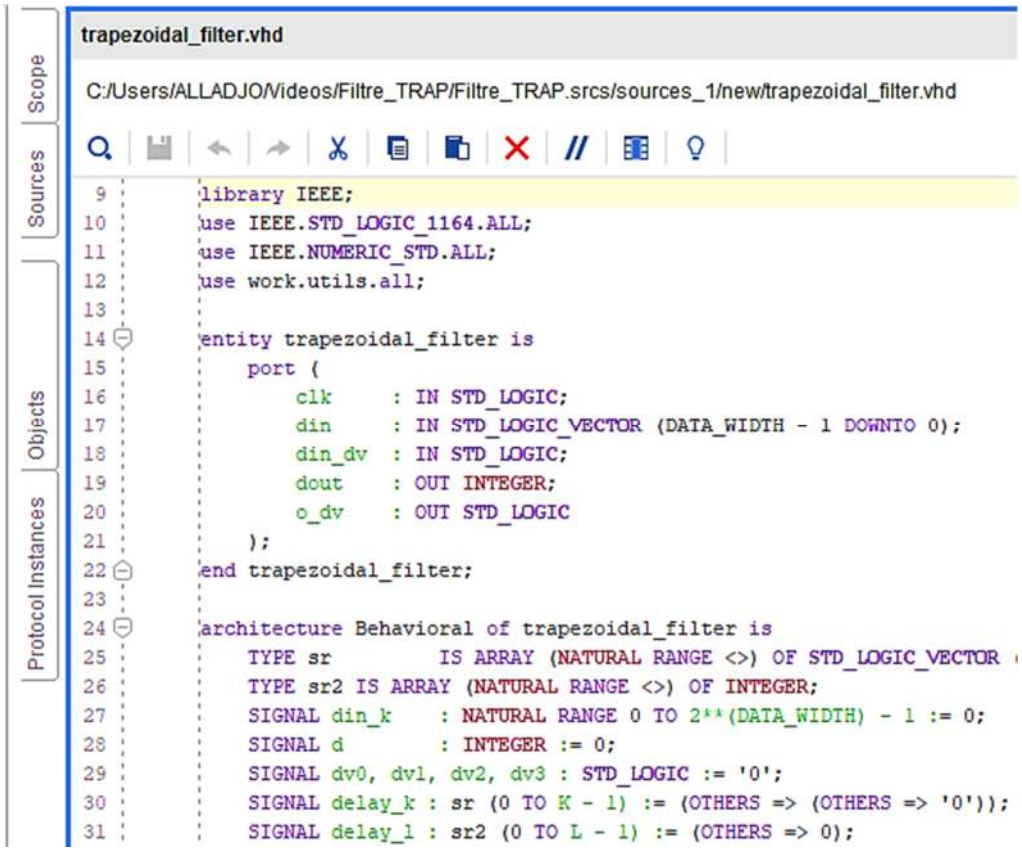


Fig.II.16 : Caractéristiques de l'éditeur de texte.

Dans la conception élaborée, la logique de bas niveau connectée aux bus est représentée en logique groupée pour faciliter la visualisation du schéma RTL.

II.4.1.2 Console Tcl

La console Tcl (illustrée dans la figure suivante) affiche :

- ❖ Les messages des commandes Tcl précédemment exécutées.
- ❖ Les erreurs de commande, les avertissements et les réussites.
- ❖ L'état des charges de conception et des contraintes de lecture.

Pour ouvrir la console Tcl, sélectionnez Window ↔ Tcl Console.

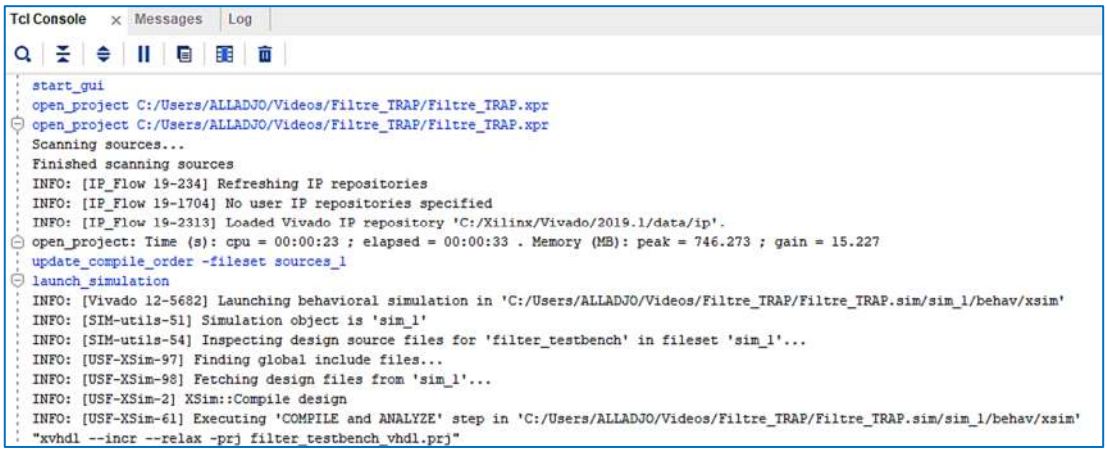


Fig.II.17 : Console Tcl.

II.4.1.3 Contraintes IP

La plupart des IP de Xilinx sont livrées avec des fichiers de contraintes (.xdc). Ils peuvent contenir des contraintes physiques, telles que la définition de normes ou d'emplacements d'E/S, et des contraintes de timing, telles que des faux chemins. Ces deux types peuvent être mélangés dans le même fichier.

Les contraintes sont écrites comme si l'IP était le sommet de la conception. Les contraintes sont automatiquement scopées à l'instance ou aux instances d'IP. Il est fortement recommandé de ne pas modifier les contraintes fournies par une IP. Il existe deux sources de contraintes utilisées par IP. Cependant, cette distinction serait importante lors la création d'un IP :

1. Les fichiers XDC créés pendant la génération de la PI et contenus dans le répertoire de la PI ou dans le conteneur du noyau.
2. Contraintes créées automatiquement par Vivado pendant le traitement de l'IP.

II.5 Conclusion

Ce chapitre a passé en revue l'architecture générale du dispositif Zynq, et a détaillé ses deux composants, PS et PL, ainsi que les ressources permettant de les interfacer. Le système de traitement comprend un double processeur ARM cortex A9 avec des extensions associées pour les SIMD et le processeur à virgule flottante, regroupés dans une unité APU avec des ressources de mémoire ; la capacité d'une unité APU et son interfaçage avec le reste des systèmes de traitement ont été soulignés.

La structure des ressources de la section de logique programmable a également été examinée, y compris la logique du tissu, la RAM de bloc et les ressources du DSP48E1, ainsi que les ressources d'interfaçage, et l'interfaçage important entre le PL et le PS en utilisant AXI a également été examiné. Enfin, la comparaison entre les dispositifs de la famille Zynq 7000 a également été examinée. Pour créer un système embarqué, le processus de conception consiste à créer le matériel du système et à développer le logiciel qui fonctionnera sur les processeurs du système. Les fabricants de FPGA fournissent une suite d'outils automatisés pour faciliter les deux parties de ce flux de conception [23].

Pour créer les circuits matériels, les outils permettent à l'utilisateur de personnaliser la logique matérielle du système en utilisant des blocs de construction préconçus pour les processeurs, les contrôleurs de mémoire, les circuits de traitement des signaux numériques et divers modules de communication.

Les blocs IP peuvent être développés par l'utilisateur ou obtenus auprès du fournisseur de l'outil ou d'un tiers. La suite d'outils de conception Vivado contient des services qui prennent en charge toutes les phases des conceptions FPGA - depuis la saisie de la conception, la simulation, la synthèse, le placement et le routage, la génération de flux binaires, le débogage et la vérification, ainsi que le développement de logiciels ciblés pour ces FPGA.

Chapitre III :

Estimation de la Densité Spectrale
de la Puissance

III.1 Introduction

Dans cette partie, nous donnons des résultats généraux pour la densité spectrale de puissance qui facilitent l'évaluation de la densité spectrale de puissance de processus aléatoires spécifiques. Le calcul de la fonction de pondération du filtre optimal utilisé en mesure d'énergie ainsi que celui prévu pour l'évaluation de la ligne de base, nécessite la caractérisation de l'électronique frontale de la chaîne de mesure par rapport aux composantes du bruit qui y règnent. Une telle caractérisation passe inévitablement par le calcul et/ou la mesure de la densité spectrale de puissance totale N(ω) du bruit électronique à l'entrée du spectromètre. Cette densité spectrale peut se référer soit à une tension, à un courant ou à une puissance.

Au cours de ce chapitre, nous allons essayer d'estimer cette entité étant donnée une chaîne spectroscopique dont les modules de l'électronique frontale sont bien définis. Nous ne prétendons nullement épuiser le sujet de l'estimation spectrale et nous oublions certainement plusieurs aspects fondamentaux. Ce domaine est en fait couvert par une très vaste littérature et nous renvoyons le lecteur à la référence [24] qui en donne une bonne introduction.

III.2 Estimation de la densité spectrale

Deux approches sont utilisées pour l'évaluation de la Densité Spectrale de Puissance. La première est celle dite l'approche directe de Fourier ; la seconde porte sur le calcul de la transformée de Fourier de la fonction d'autocorrélation du signal à étudier.

$$G(T,f)=\frac{|X(T,f)|^2}{T} \qquad \text{(III.1)}$$

L'approche alternative est de déterminer l'autocorrélation du signal, définie par :

$$R(T,t-\tau)=\begin{cases} x(t)*x(t-\tau), T\varepsilon[0,T], T-\tau\varepsilon[0,T] \\ 0 \qquad ailleurs \end{cases} \qquad \text{(III.2)}$$

Tout en sachant que x(t) est un signal définit sur [0, T]. Cette fonction d'autocorrélation est non nulle dans la région du plan (τ, t) illustrée par la figure III.1). Et prendre alors une moyenne temporelle pour former une fonction de corrélation moyennée. Ces deux approches mènent à des résultats identiques.

Les deux définitions peuvent être facilement généralisées pour les processus aléatoires et à des intervalles de temps infinis. Mais, la nature directe de l'approche de Fourier facilite la dérivation de la DSP des signaux des processus aléatoires.

Une projection des deux approches qui viennent d'être citées, dans le domaine temporel discret a donné naissance aux deux méthodes généralement utilisées pour estimer la DSP dans le domaine discret à partir d'une réalisation du processus limitée dans le temps.

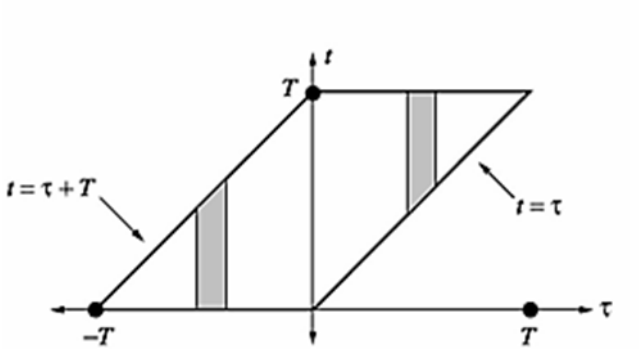


Fig.III.1 : Région où la fonction d'autocorrélation est non-nulle.

III.2.1 Méthode directe

C'est une méthode utilisant directement le signal temporel tronqué. Le spectre est estimé par :

$$\hat{S}_{PER}(f) = \left| \frac{1}{N} \sum_{n=0}^{N-1} X(n) \exp(-2\pi jfn) \right|^2 \qquad \text{(III.3)}$$

Du fait de la troncature du signal, le périodogramme est en fait la convolution du spectre par une fenêtre en sinus cardinal. Le fait de tronquer le signal induit deux phénomènes :

- ❖ Élargissement du lobe principal ⇒ perte de résolution ;
- ❖ Apparition de lobes secondaires.

On montre que la résolution du périodogramme est de l'ordre de 1/N, c'est-à-dire que l'on peut discriminer 2 fréquences distantes de 1/N. Pour remédier aux problèmes de truncate, on utilise en général des fenêtres permettant soit de réduire le lobe principal, soit d'atténuer les lobes secondaires. On montre [25] que le periodogramme est un estimateur biaise :

$$E\left\{\hat{S}_{PER}(f)\right\} = \int_{-1/2}^{1/2} W_b(f-u) S_x(u) du \qquad \text{(III.4)}$$

Le périodogramme est donc en moyenne la convolution du véritable spectre avec la transformée de Fourier de la fenêtre triangulaire. Néanmoins, lorsque N → ∞, le biais devient nul. On peut montrer également que la variance est pratiquement indépendante de N et proportionnelle au spectre :

$$Var\left\{\hat{S}_{PER}\right\}(f) \cong S_x(f)^2 \qquad \text{(III.5)}$$

Le périodogramme n'est donc pas un estimateur consistant de la d.s.p. Afin de diminuer la variance de cet estimateur, on peut utiliser un périodogramme moyenne. Ceci consiste à séparer le signal en K tranches (de longueur N/K), à calculer le périodogramme sur chaque tranche et à faire la moyenne. Du fait des K moyennages, la variance est presque divisée par K : néanmoins, les tranches étant plus courtes, la résolution diminue.

III.2.2 Méthode indirecte

Une autre approche consiste à utiliser la définition du spectre à partir de la fonction de corrélation. On estime alors le spectre comme :

$$\hat{S}_{BT}(f) = \sum_{n=-M}^{M} \hat{r}_{xx}(m)\exp(-2\pi jmf) \qquad \text{(III.6)}$$

Blackman-Tuckey ont suggéré de prendre M de l'ordre de 10% N. L'application d'une fenêtre est possible pour diminuer la variance sur l'estimation de la fonction de corrélation. En effet, on a :

$$\hat{r}_{PER}(f) = \sum_{m=-(N-1)}^{N-1} \hat{r}_{xx}(m)\exp(-2\pi jmf) \qquad \text{(III.7)}$$

Pour ce qui est des travaux d'estimation de la d.s.p., adressés tout au long de ce document, nous avons utilisé le périodogramme moyenné de la méthode directe. Tout en supposant que les processus à traiter (bruit électronique de la chaîne de mesure) sont stationnaires et ergodiques.

III.3 Décomposition spectrale

Lors du chapitre deux, nous avons essayé d'énumérer, d'une manière théorique, les différents types de bruit susceptibles d'exister dans un système d'électronique frontale utilisé en mesure spectroscopique. Mais il faut noter qu'en pratique, on est loin d'arriver à déceler et/ou à séparer chaque type de bruit à part.

Et quel que soit la représentation adoptée, un spectre de bruit est, dans la plupart des cas, la superposition de plusieurs composantes de bruit dont la prépondérance des unes par rapport aux autres diffèrent d'une situation à une autre.

III.3.1 Expression de la densité spectrale du bruit

Puisque le bruit considéré est supposé être additif, stationnaire et non corrélé au signal utile [12], la représentation fréquentielle (Densité spectrale de puissance du bruit) obtenue par la méthode décrite précédemment et adaptée au présent contexte, peut être exprimé mathématiquement, dans la gamme de fréquence considérée, par : le plus proche possible suivant « Série de Laurent » [16] donnée par la formule :

$$N(\omega) = \sum \alpha_k \, |\omega|^k \quad K = -2,-1,0,1,2 \ldots \qquad \text{(III.8)}$$

Avec : αk : constantes ne dépendant pas de la fréquence. ω = 2π f : fréquence angulaire.

Pratiquement, cette approximation peut conduire à des coefficients αk ayant des signes négatifs, ce qui ne reflète aucune réalité physique. Mais l'expression reste, pour le moment, la meilleure approche, pour approximer la densité spectrale du bruit électronique à l'entrée de la chaîne et dans la gamme de fréquence considérée. L'entité ainsi obtenue, exprimée souvent en [A2/Hz], est directement introduite dans les calculs de la fonction de pondération du filtre optimal requis en mesures spectroscopiques [17]. Et pour une prise en charge d'autres composantes de bruits, (comme par exemple : bruit de recombinaison/génération dont le spectre associé est une Lorentzienne), on avait ajoute le terme de contribution du bruit Lorentzien (composantes série et parallèle) :

$$N(\Phi) = \sum_k H_k \frac{1}{1+G_k^2\Phi^2} + \frac{H_N}{|\Phi|^N} + \ldots + \frac{h_1}{|\Phi|} + 1 + F(|\Phi|) + \Phi^2 + K_1 |\Phi|^3 + \ldots + K_M |\Phi|^{M+2} + \ldots \sum_k K_k \frac{L_k^2\Phi^2}{1+L_k^2\Phi^2} \qquad \text{(III.9)}$$

III.3.2 Cas particuliers

Dans des cas pratiques bien définis, des densités spectrales de puissance typiques sont données par :

1. Spectroscopie X avec détecteur (Si) à la température ambiante.

$$N(\omega) = C^2 a\omega^2 + b + C^2 af\, 2\pi\, |\omega| \qquad \text{(III.10)}$$

2. Spectroscopie X avec détecteur (GaAs) à la température ambiante.
3. Spectroscopie Gamma avec détecteur (HPGe) refroidi à l'azote liquide.

$$N(\omega) = C^2 \alpha'' \omega^2 + b'' \qquad \text{(III.11)}$$

III.4 Source de bruit équivalente ramenée au spectromètre

Le bruit total observé à la sortie du système de mesure est en fait la somme des contributions de plusieurs sources N1(ω), N2(ω), ... Néanmoins, et conformément à ce que nous venons d'énoncer en haut, l'utilisateur qui observe le signal en sortie ne peut

pas discerner les contributions de S1(ω), S2(ω), ... Pour lui, tout se passe comme s'il n'y avait qu'une seule source de bruit.

Si l'on suppose que cette unique source équivalente est placée à l'entrée du système, alors on peut la comparer directement au signal que l'on souhaite mesurer. En particulier, la variance de cette source de bruit donne directement la résolution du système complet. De ce fait, le système de mesure « détecteur + étages de traitement en amont » peut être modélisé par un circuit équivalent (voir figure III.2) composé de :

1. Un signal d'entrée : un générateur d'impulsion de courant de charge Q (modèle du détecteur).
2. Une capacité C représentant la somme des capacités du détecteur et des composants d'entrée (Ordre de grandeur de 10 à 20 pF).

III. Toutes les sources de bruits relatives à (l'étage d'entrée + détecteur) sont représentées par une seule source de bruit montée soit en en générateur de courant ou de tension. Pour une meilleure clarté on réfère le signal d'entrée et la source de bruit à un même type de source.

4. L'étage en amont est supposé être sans bruit et est modélisé par sa fonction de transfert. C'est le modèle électrique réduit de l'entrée d'un spectromètre générique, et c'est sur ce modèle que vont s'appuyer tous les calculs du filtre optimal utilisé pour la mesure de l'énergie déposée dans le détecteur par une particule ionisante.

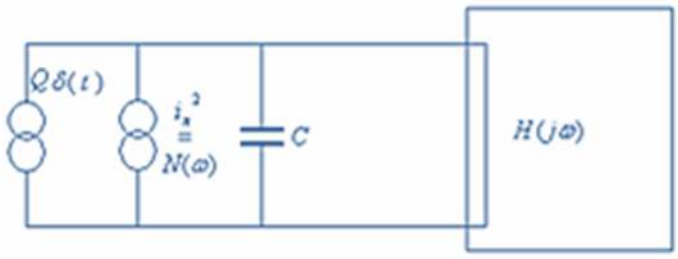


Fig.III.2 : Réduction des sources de bruits à une seule source équivalente.

III.5 Détermination de la DSP du bruit de la chaîne

Dans le but de caractériser la chaîne spectroscopique « spectroscopie Gamma avec détecteur (HPGe) refroidi à l'azote liquide », vis-à-vis du bruit électronique qui y règne, et de déterminer la densité spectrale de puissance correspondante. Pour déterminer la D.S.P on doit passer par les tâches suivantes :

1. Acquisition des enregistrements temporels du signal à la sortie du préamplificateur de charge en l'absence de tout événement radiatif.

Les acquisitions sont faites par oscilloscope numérique « Tektronix TDS220 », relié au PC via une liaison RS-232D. Plusieurs fréquences d'échantillonnage ont été essayées avant le choix final. Le nombre des échantillons de chaque acquisition est fixé par le matériel utilisé à 2500 points, au temps où la fréquence d'échantillonnage est réglable par l'utilisateur, nous avons travaillé dans la gamme (5MHz ~ 50MHz) Caractéristiques du préamplificateur de charge utilisé :

- Nomenclature : Model Canberra de type 2002CSL (refroidi par l'azote liquide)
- Constante du temps : 2 GΩ * 1 pF
- Gain : 2V/pC (pC : Pico-Coulomb).

2. En utilisant ces échantillons, nous avons calculé une estimation de la densité spectrale de puissance à l'aide de la méthode du périodogramme moyenné (méthode directe) [25]. Cela a été fait via le logiciel Matlab.

3. L'estimation de la densité spectrale de puissance ainsi obtenue, est ramenée à l'entrée de la chaîne en divisant les valeurs en question, point par point, par le carrée de la fonction de transfert du bloc préamplificateur,

4. Après cette étape, nous sommes passé à l'opération de la décomposition spectrale. À cet effet, la densité spectrale du bruit à l'entrée de la chaîne a été exprimée en termes de puissance négatifs et positifs de la fréquence (série de Laurent). Pour arriver à cette fin, nous avons essayé plusieurs méthodes d'approximation (fitting), parmi lesquelles on note celle qui a été retenue : La méthode des moindres carrés pondérés. On note à ce stade que la méthode utilisée ne converge pas à tous les coups, cependant un choix judicieux des paramètres initiaux donne des résultats probants.

III.7 Conclusion

Les spectres obtenus par la méthode d'estimation suivie lors de ce chapitre sont limités en termes de fréquences. L'inverse de la durée totale des enregistrements temporels limite le spectre dans la zone des faibles fréquences, tandis que la fréquence d'échantillonnage utilisée limite la validité de ce spectre dans la gamme des hautes fréquences. Mais cela n'empêche pas de dire que les expressions finales obtenus confirment bien la concordance des résultats avec ceux énoncés théoriquement, et surtout en ce qui concerne la chaîne spectroscopique gamma. D'où l'efficacité de la démarche suivie, pour donner des approximations acceptables des densités spectrales du bruit à l'entrée de la chaîne spectroscopique. Des données qui seront utilisées pour la conception des filtres optimaux traités dans les prochains chapitres.

Chapitre IV :
Implémentation du Filtre Trapézoïdal

IV.1 Introduction

De nombreux algorithmes ont été développés pour mettre en œuvre le filtrage trapézoïdal en utilisant respectivement la méthode de convolution et la méthode de la fonction de transfert. Dans ce chapitre, nous avons adapté l'algorithme de filtrage trapézoïdal en utilisant la méthode de convolution.

IV.2 Implémentation du système de détection de rayonnement γ

Le filtrage trapézoïdal est une méthode bien connue et efficace pour obtenir un spectre avec un rapport signal/bruit (résolution de pic), qui est proche de l'optimal pour de nombreux systèmes pratiques de spectroscopie numérique. Un tel système comprend une section analogique et une section numérique. Dans la section analogique, une impulsion de courant produite par le détecteur est intégrée sur un condensateur de rétroaction d'un préamplificateur sensible à la charge qui produit une impulsion exponentielle en forme de pas ou à décroissance lente (50 µsec). Afin de réduire l'effet de pile-up (superposition de deux impulsions proches) et d'éliminer les composantes basse fréquence du bruit (blanchiment du bruit), un différentiateur est inclus dans la section. Celui-ci raccourcit le temps de décroissance à une valeur qui correspond à la constante de temps du bruit généré par le détecteur et le préamplificateur (généralement quelques microsecondes). Dans le cas d'une impulsion exponentielle, le différentiateur peut produire une forme exponentielle déformée.

Enfin, le signal est amplifié par un amplificateur linéaire et prêt pour la conversion ADC. Dans la section numérique, le signal est converti par un CAN rapide (50 MHz environ) pour un traitement en temps réel. Dans le bloc, la composante continue est éliminée. Ceci est fait par un suppresseur ou bloqueur de courant continu (DC-S). Ensuite, le signal est mis en forme trapézoïdale. La largeur du signal doit être ajustée à la valeur optimale, qui dépend de la constante de temps de coin du bruit du système détecteur-préamplificateur. Avant que l'amplitude du signal mis en forme ne soit mesurée, il peut être corrigé du courant continu. Ceci est fait par un restaurateur de ligne de base (BLR). La distribution des amplitudes est enregistrée dans une mémoire. Afin de rejeter les événements d'empilement, le signal d'entrée peut être formé en trapèze avec une largeur très courte (beaucoup moins qu'optimale). Cela permet de résoudre plus facilement deux impulsions successives et de rejeter les événements lorsqu'ils se produisent. [27]

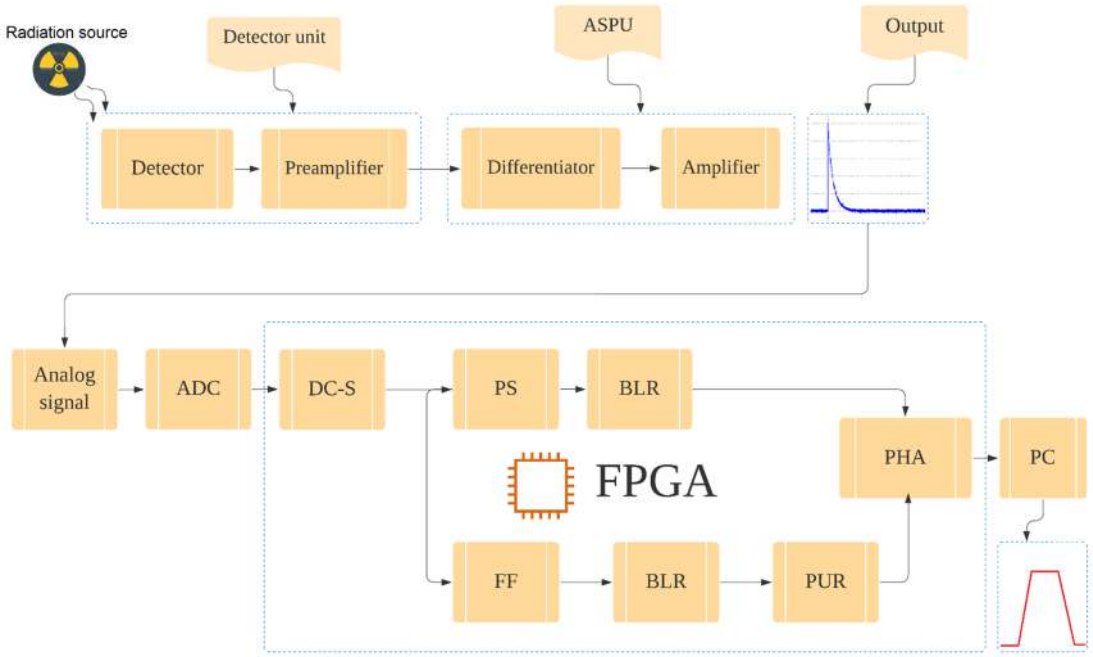


Fig. IV.1 : Chaine de spectroscopie numérique.

IV.2.1 Flux de conception

Dans cette partie, nous allons concevoir un circuit de mise en forme trapézoïdal et un DC-blocker. La section analogique sera remplacée par un générateur de formes d'onde, capable de générer une traînée de signaux exponentiels avec un temps de décroissance court de quelques microsecondes. Pour la numérisation et le traitement, l'utilisation d'une carte ZedBoard de la famille Zynq-7000 SoC est nécessaire. La configuration est montrée sur la figure IV.2.

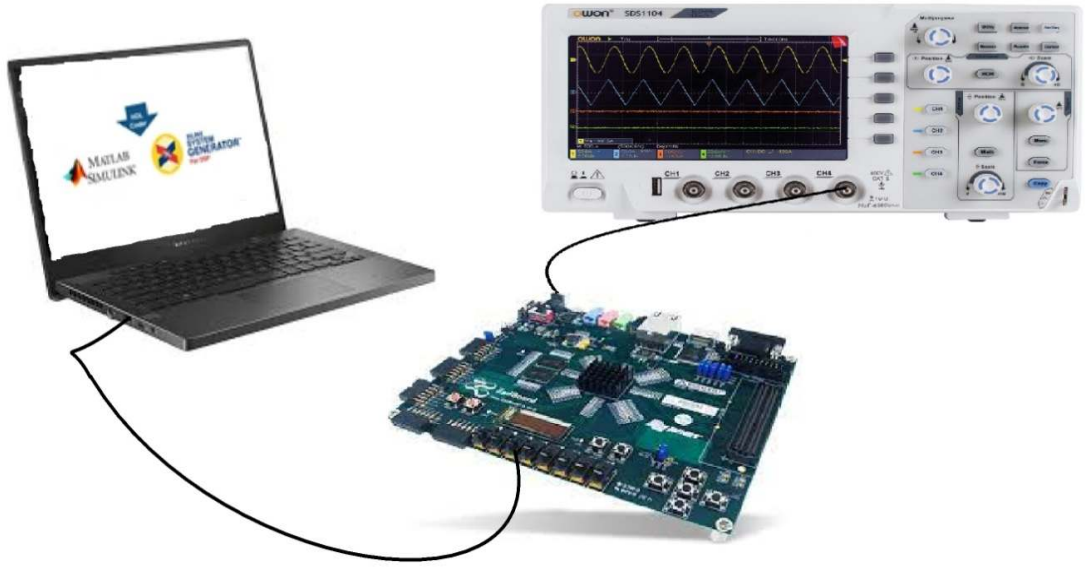


Fig. IV.2 : Acquisition par ordinateur.

La tâche principale sera de produire le fichier de configuration du FPGA qui contient le traitement des données et la logique de communication avec le PC. Le fichier de configuration aura produit en utilisant l'environnement Xilinx. La conception du FPGA sera constituée de 5 modules qui communiquent entre eux selon un protocole générique de bus en épi. Ces modules sont les suivants [27] :

1. **Syscon** - génère l'horloge système
2. **Le module maître FX-2** qui est responsable de la communication du FPGA avec le PC via USB.
3. **Module Filter Slave** qui contient l'algorithme de filtrage, l'oscilloscope (RAM + trigger) et la logique pour la communication avec le bus wishbone.
4. **Module Intercon** qui connecte tous les modules maîtres et esclaves de la conception.

Les modules 1, 2 et 4 sont standards et déjà écrits en langage VHDL en utilisant l'éditeur Xilinx ISE. Le module 5 sera conçu dans un environnement de programmation graphique (au lieu de VHDL) en utilisant un programme informatique appelé Matlab/Simulink et un ensemble de bibliothèques compatibles appelées Xilinx System generator. Ces outils utilisent des blocs graphiques au lieu du langage VHDL et ils sont capables de synthétiser du code VHDL à partir de blocs graphiques.

Le code VHDL généré par Matlab/Simulink / System Generator sera ensuite couplé avec le reste des modules (1,2,4) en utilisant Xilinx ISE. Enfin, l'ISE traduira, mappera et routera la conception dans un fichier qui pourra être chargé dans le FPGA.

La communication avec le FPGA à partir du PC sera effectuée par un programme écrit en C++. La carte ZedBoard communique avec le PC via le port USB. Comme le même port est utilisé pour alimenter les parties analogique et numérique, aucune alimentation externe n'est nécessaire, mais la carte ne peut fonctionner que si elle est connectée au PC via un câble USB.

IV.2.2 Création d'un modèle Simulink

La conception décrite dans le fichier modèle sera constituée de plusieurs parties logiques appelées sous-systèmes. Les sous-systèmes sont constitués de blocs Matlab/Simulink et de blocs Xilinx. Seuls les blocs Xilinx seront traduits en code VHDL ou en d'autres fichiers nécessaires à la production du fichier de configuration Zynq-7000 final. Les blocs Xilinx et Matlab sont séparés par des blocs appelés "gate way".

IV.2.1 Générateur d'impulsions

Le premier bloc que nous devons créer s'appelle "pulse_generator". Il est utilisé pour les besoins de la simulation et génère des signaux d'entrée pour le filtre. Les signaux d'entrée seront des signaux périodiques, à décroissance exponentielle, avec un temps de décroissance de l'ordre de quelques microsecondes (5usec dans cet exemple). Il correspond à la sortie d'un préamplificateur de type reset suivi d'un filtre passe-haut avec une constante RC de 5µsec.

La fonction du générateur d'impulsions peut être expliquée comme suit. Le bloc appelé générateur d'impulsions, qui simule le temps d'arrivée d'une particule de charge, déclenche le sous-système déclenché avec une période Tpprd microsecondes (temps normalisé par MATLAB). Le sous-système déclenché, les blocs Mémoire et Gain simulent le détecteur/préamplificateur. Chaque impulsion de déclenchement, est généré et ajouté à la valeur précédente de la sortie du bloc Mémoire. De cette façon, un signal en escalier est créé et son amplitude change toutes les Tpprd microsecondes. Le bloc Zero-pole est utilisé pour simuler un filtre passe-haut avec une constante RC égale à Taud. Par conséquent, la sortie globale du sous-système est une impulsion exponentielle périodique avec un temps de décroissance Taud (voir le code ci-dessous).

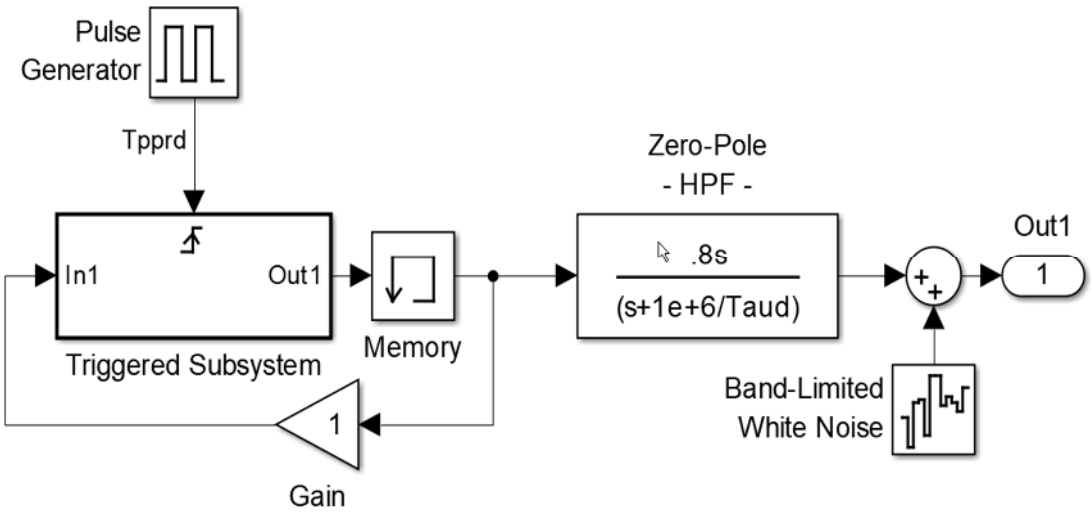


Fig. IV.3 : Générateur d'impulsions.

Script Matlab

```
clear all
clc
%% ========================================================== %%
        Parameters
%% ========================================================== %%
% Pulse period
Tpprd = 100;
% Clock period [µsec]
Tclk = 1./50 ;
Tclkn = Tclk*1e-6 ;
```

```
% High pass filter differentiation constant
Taud = 5; % RC constant -HPF-
%% ================== Filter parameters ================== %%
Taupk = 3;
Taupk_top = 2;
%% =====================================
b10 = exp(-Tclk/Taud);
%% =====================================
na = (Taupk/Tclk);
nad = na-3;
%% =====================================
nb = (Taupk_top+Taupk)/Tclk;
nbd = nb-3;
%% =====================================
na_inv=1/na;
```

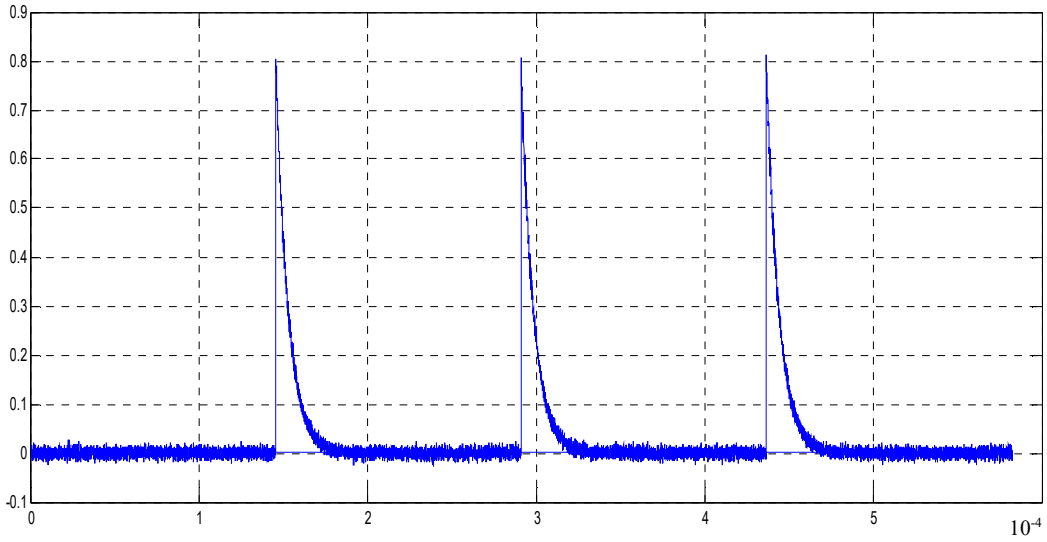


Fig. IV.4 : Simulation d'entrée de forme exponentielle

IV.2.2. Quantificateur ADC

Le module ADC convertit les signaux analogiques de l'amplificateur en données numériques qui peuvent être utilisées ultérieurement pour d'autres analyses. Les vitesses typiques de l'ADC pour le spectromètre gamma numérique se situe entre 40MHz et 100MHz. [6]

Le codeur d'amplitude capture l'amplitude max du signal analogique provenant du Shaper. Le schéma de principe présenté ici est celui du détecteur de crête. La durée de la porte est généralement comprise entre quelques 100 ns à quelques 10 µs.

Lorsque le détecteur et le préamplificateur sont utilisés, un convertisseur analogique-numérique (CAN) doit être ajouté après ces étapes pour numériser le signal. En outre, un étage de filtrage analogique optionnel avant le CAN peut également être ajouté. L'un des objectifs de cette étude était d'utiliser l'algorithme dans un environnement dans lequel les besoins en énergie sont très restrictifs.

Ainsi, l'un des principaux objectifs était de réduire la fréquence d'échantillonnage autant que possible afin de réduire la consommation d'énergie. Dans ce but, un filtre passif passe-haut a été inclus entre le préamplificateur et l'ADC. Il s'agissait d'un filtre anti-repliement, généralement utilisé avant un CAN. Le pôle génère un filtre passe-haut qui élimine les basses fréquences. Tous les traitements, y compris la mise en forme et la détection des pics, sont effectués à l'aide d'éléments numériques [3].

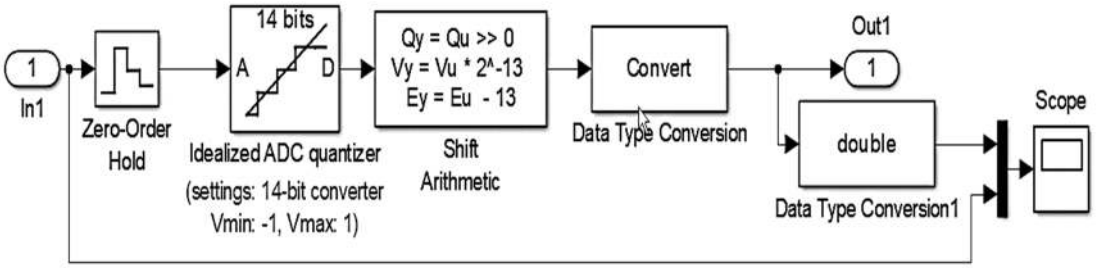


Fig. IV.5 : Schéma block ADC simule avec Matlab Simulink.

L'élément Zero-Order Hold échantillonne et maintient le signal d'entrée avec une précision totale sur le front montant de jusqu'au prochain front montant. L'ADC idéalisé convertit les valeurs de pleine précision stockées dans l'élément Zero-Order Hold en un nombre le plus proche. Ce type de quantification est appelé "Rounding".

Les valeurs numérisées sont représentées par un nombre binaire de 14 bits. En utilisant 14 bits, il est possible de définir 16384 valeurs différentes. Puisque le signal d'entrée est bipolaire et se situe dans la plage (+1,-1), la sortie du CAN sera comprise entre -8191 et +8191 [27]. A ce stade, il convient de souligner que la quantification du signal d'entrée et les opérations arithmétiques ultérieures introduisent du bruit. Tout d'abord, on peut remarquer une certaine différence entre valeur analogique exacte et sa représentation binaire. Cette erreur de quantification ε sera uniformément distribuée dans l'intervalle :

$$-\frac{2^{-B}}{2} \leq \varepsilon \leq \frac{2^{-B}}{2} \qquad \text{(IV.1)}$$

La variance de cette erreur de quantification sera de :

$$\sigma_{\varepsilon}^{2} = \frac{1}{\Delta} \int_{-\frac{2^{-B}}{2}}^{\frac{2^{-B}}{2}} (\varepsilon - m_{\varepsilon})^{2} d\varepsilon = \frac{2^{-2B}}{12} \qquad \text{(IV.2)}$$

Outre le type de quantification par arrondi ci-dessus, il existe également un autre type appelé quantification par truncate. Où la valeur est représentée par le niveau quantifié le plus proche et le plus bas.

$$m_\varepsilon = \frac{2^{-B}}{2} \qquad \text{(IV.3)}$$

$$\sigma_\varepsilon{}^2 = \frac{2^{-2B}}{12} \qquad \text{(IV.4)}$$

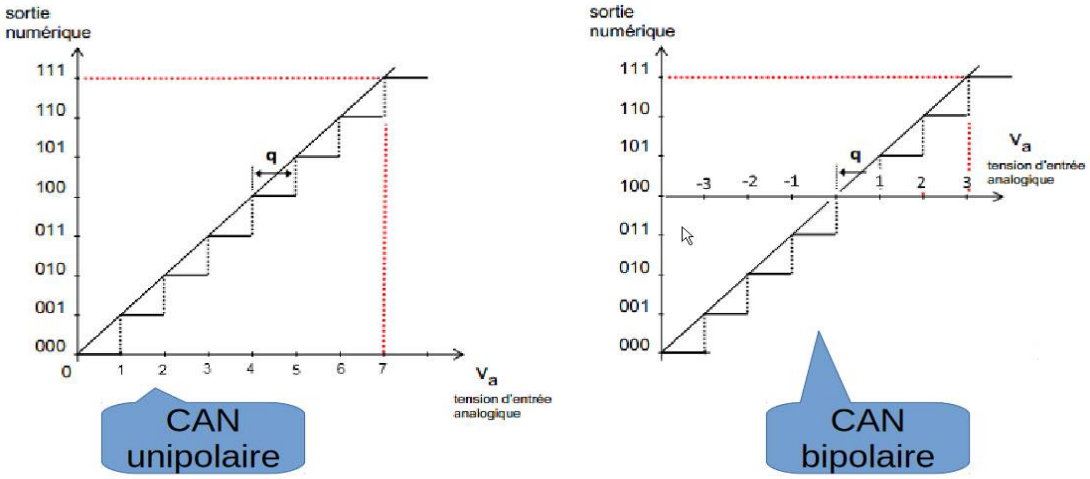


Fig. IV.6 : Erreur de quantification par truncate

IV.2.3 Interface du bloc Simulink avec Xilinx system generator

Le filtre sera implémenté en utilisant les blocs de Xilinx, au lieu de ceux de Simulink. Les blocs et la logique associée seront transférés aux portes matérielles et les données échantillonnées par un ADC réel alimenté dans le FPGA par ses broches E/S. Par conséquent, les données introduites dans le filtre doivent passer par le bloc "Gateway In" de Xilinx comme indiqué dans la figure IV.7 :

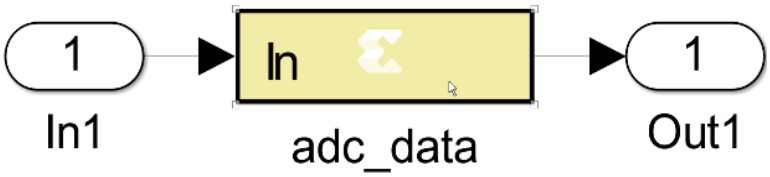


Fig. IV.7 : Interface ADC.

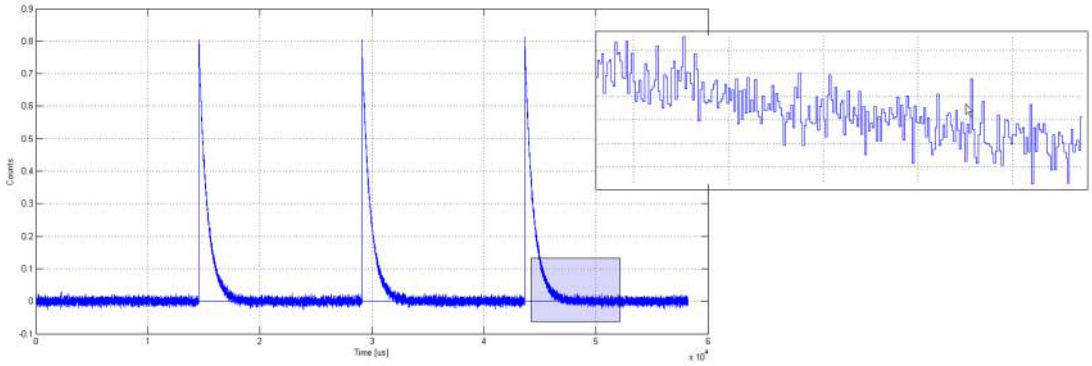


Fig. IV.8 : Simulation de la sortie ADC.

IV.3 Implémentation de filtre trapézoïdale

Pour l'implémentation du filtre trapézoïdal, deux méthodes sont développées dans ce manuscrit. Notamment l'implémentation en utilisant Xilinx System Generator, et l'implémentation en utilisant la programmation en VHDL.

IV.3.1 Implémentation avec Xilinx system generator

Nous allons maintenant procéder à l'implémentation du filtre. La fonction de transfert globale du filtre peut être exprimée sous forme de quatre fonctions de transfert correspondant aux mises en forme successives pour l'obtention du filtre trapézoïdal. La fonction de transfert du filtre peut être factorisée de la manière suivante [27] :

$$h(z) = (1 - e^{-\frac{Tclk}{\tau d}} Z^{-1})(\frac{1 - Z^{-n_a}}{1 - Z^{-1}})(\frac{1 - Z^{-n_b}}{1 - Z^{-1}})(\frac{Z^{-1}}{n_a}) \quad \text{(IV.5)}$$

En plus de la période d'horloge et de la constante du filtre passe-haut, la formule comporte deux autres paramètres qui définissent le sommet $\tau_{ft} = n_a + n_b$ plat et les temps de pic en unité de période d'horloge.

$$H_1(z) = 1 - e^{-\frac{Tclk}{\tau_d}} z^{-1} \quad \text{(IV.6)}$$

La formule ci-dessus, écrite dans l'espace de la transformée en z, correspond à la récurrence suivante dans l'espace n entre les séries d'échantillons d'entrée et de sortie x_n et y_n respectivement.

$$y_n = x_n - b_{10} x_{n-1} \quad \text{(IV.7)}$$

$$b_{10} = e^{-\frac{Tclk}{\tau d}} \quad \text{(IV.8)}$$

Cette forme peut être calculée en entrant la formule de la fonction de décroissance exponentielle d'entrée dans la relation récursive du filtre :

$$x_n = \begin{cases} e^{-\frac{Tclk}{\tau_d} n} & n \geq 0 \\ 0 & n \prec 0 \end{cases}$$

Ceci nous donne :

$$\begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

La deuxième fonction de transfert peut être écrite sous la forme suivante :

$$H_2(z) = (\frac{1-Z^{-n_a}}{1-Z^{-1}}) = \frac{1}{1-z^{-1}}(1-Z^{-n_a}) \quad \text{(IV.9)}$$

Le facteur $H_2'(z) = \frac{1}{1-z^{-1}}$ représente un intégrateur. La relation récursive correspondant à cette expression est :

$$y_n = y_{n-1} + x_n \quad \text{(IV.10)}$$

Le deuxième facteur $H_2''(z) = 1 - z^{-n_a}$ est appelé filtre en peigne. Sa relation récursive est :

$$y_n = x_n - x_{n-n_a} \quad \text{(IV.11)}$$

Où $n_a = (Tau_{pk}/T_{clk})$;

et $n_{ad} = n_a - 3$;

Comme l'entrée $\mathbf{x_n}$ dans la deuxième fonction de transfert est un signal de type delta, la sortie ne sera différente de zéro que pendant les $\mathbf{n_a}$ horloges successives. Pendant cette période, la somme totale est toujours constante. Par conséquent, la sortie a la forme d'un signal rectangulaire de largeur égale à et l'amplitude étant égale à l'amplitude de la fonction delta d'entrée.

Afin de comprendre la forme, rappelons que la fonction de transfert du troisième système est :

$$H_3(z) = (\frac{1-Z^{-n_b}}{1-Z^{-1}}) \quad \text{(IV.12)}$$

Il peut être réécrit sous la forme dite FIR (Finite Impulse Response) comme suit :

$$H_3(z) = 1 + z^{-1} + ... + z^{-(n_a-1)} \quad \text{(IV.13)}$$

La relation récursive correspondante est :

$$y_n = x_n + x_{n-1} + ... + x_{n-(n_a-1)} \quad \text{(IV.14)}$$

Le dernier système a pour fonction de transfert :

$$H_4(z) = (\frac{Z^{-1}}{n_a}) \quad \text{(IV.15)}$$

Pour mettre en œuvre la division par n_a, il est plus pratique d'utiliser la multiplication avec un nombre précalculé, qui peut être stocké dans un registre. Par conséquent, le système 4 ne nécessite qu'un seul multiplicateur, à part un élément de retard qui n'affecte pas l'amplitude de la sortie finale du filtre global. La constante doit être déclarée dans le fichier script comme indiqué ci-dessous :

$$n_{a\_inv} = 1/n_a ;$$

Il convient de souligner que la précision du signal rectangulaire à la sortie du système 2 est de 14 bits, mais le résultat après le système 4 est effectivement obtenu après avoir fait la moyenne des n_a valeurs qui ne diffèrent que par le bruit (les valeurs qui forment le rectangle). Le résultat est donc $\sqrt{n_a}$ fois meilleure en précision. Puisque n_a est habituellement plus grand que 100 (temps de pic 2microsec), l'amélioration de la précision est au moins 10 fois supérieure, c'est-à-dire 3 bits.

Cela signifie que la précision de sortie du multiplicateur du système 4 peut être augmentée de 3 bits supplémentaires. Le gain total à la fréquence DC est :

$$G = G_1 G_2 G_3 G_4 = (\frac{T_{clk}}{\tau_d})(\frac{\tau_p}{T_{clk}})(\frac{\tau_p + \tau_{ft}}{T_{clk}})(\frac{T_{clk}}{\tau_p}) = \frac{\tau_p + \tau_{ft}}{T_d} \qquad \text{(IV.16)}$$

Dans le cas d'un temps de pointe très important, le gain **G** peut atteindre 8 et il faut ajouter 3bits supplémentaires plus un bit de signe (4 au total) à gauche du point binaire afin d'obtenir le résultat final (en supposant que l'entrée soit limitée à +-1). Puisqu' une suppression du courant continu DC offset sera insérée après le système 2, la sortie finale sera dans la plage +-2. Par conséquent, la précision de la sortie peut être réduite à 16 bits, avec 14 points binaire.

Dans notre cas, $T_{clk} = 20\,n\sec$ En supposant que $\tau_{ft} = 2\,\mu\sec$ $\tau_p = 3\,\mu\sec$ $\tau_d = 5\,\mu\sec$

Il s'en suit que :

$$G = \frac{20}{5000} * \frac{3000}{20} * \frac{5000}{20} * \frac{20}{3000} = \frac{1}{250} * \frac{150}{1} * \frac{250}{1} * \frac{1}{150} = 1 \qquad \text{(IV.17)}$$

Par conséquent, tout décalage (DC offset) à l'entrée de l'étage 1 est amplifié par un facteur de 1.

IV.3.1.1 Simulation du filtre Trapézoïdal

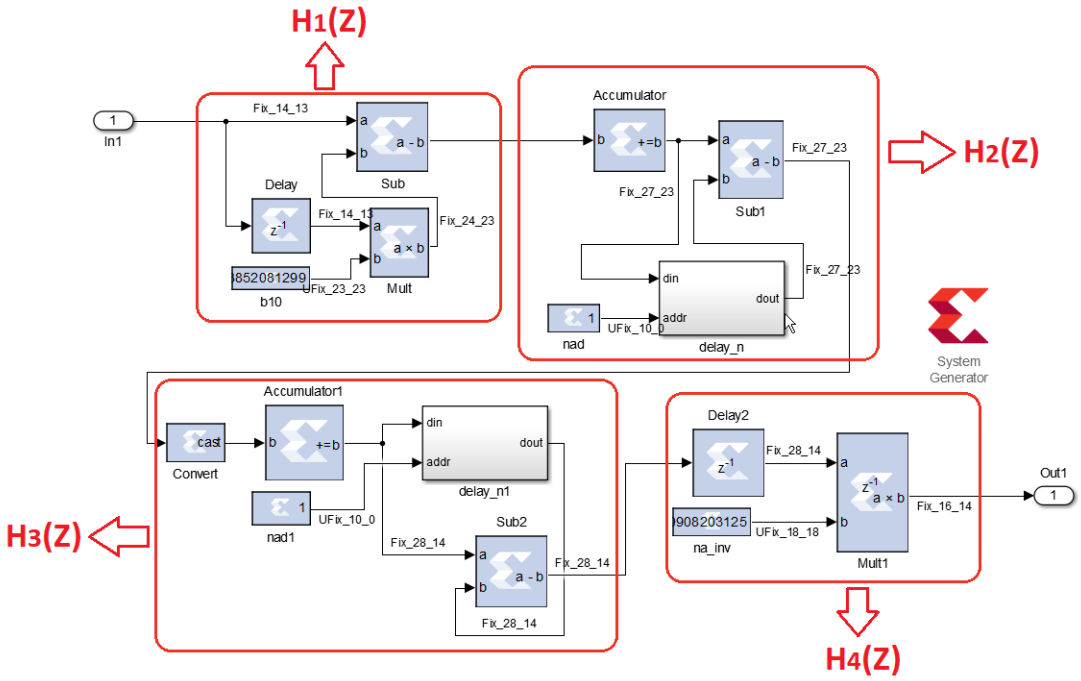


Fig. IV.9 : Schéma Bloc du filtre Trapézoïdal.

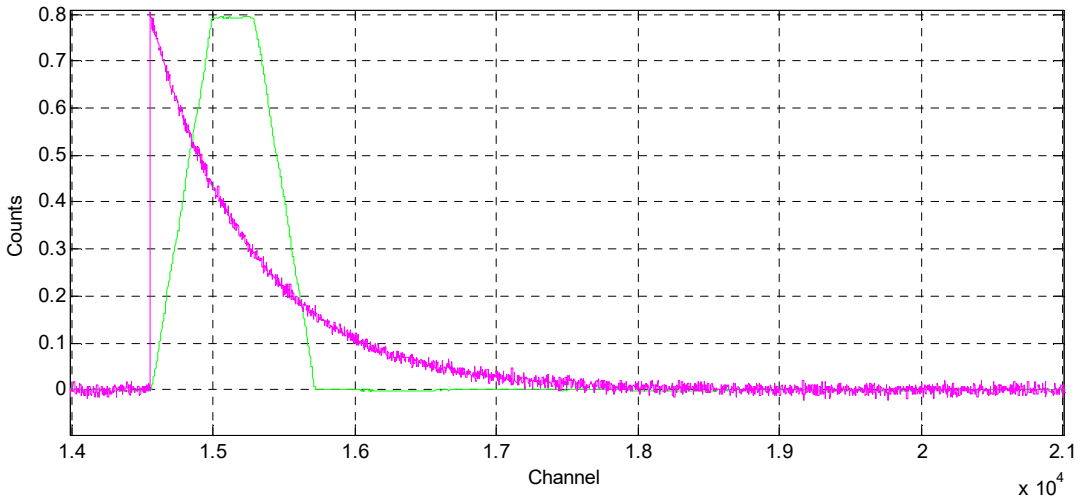


Fig. IV.10 : Traçage de l'impulsion exponentiel avec celle de filtre Trapézoïdale avec Matlab.

1) Erreurs de quantification due à l'ADC :

Afin d'estimer les erreurs dues à la quantification de l'ADC, il faut se rappeler que la quantification d'un signal d'entrée x_n qui est échantillonné avec une précision de 13 bits (le format est 14.13, signé) introduit des erreurs qui sont modélisées comme un bruit avec une variance σ_ε égale à $\sigma_\varepsilon = \frac{2^{-2B}}{12} = \frac{2^{-2*13}}{12}$.

Ce bruit va se propager dans fonctions de transferts $H_1(Z)$ et $H_2(Z)$. Afin d'estimer la variance totale du bruit à la sortie de la deuxième fonction de transfert, il est utile d'écrire la fonction de transfert totale $H_1(Z)$ et $H_2(Z)$ comme une somme de produit :

$$H_{12}(z) = (1 - b_{10}z^{-1})(1 + z^{-1} + ... + z^{-(n_a - 1)}) \quad \text{(IV.18)}$$

Après avoir effectué la multiplication, il s'en suit que :

$$H_{12}(z) = \sum_{k=0}^{n_a} h_k z^{-k} \quad \text{(IV.19)}$$

$$\text{Où} \begin{cases} 1 & k = 0 \\ 1 - b_{10} & 1 \prec k \prec n_a \\ -b_{10} & k = n_a \end{cases}$$

La forme de la fonction de transfert ci-dessus correspond à la relation somme de produits suivante :

$$y_n = \sum_{k=0}^{n_a - 1} h_k x_{n-k} = h_0 x_n + h_1 x_{n-1} + ... + h_{n_a} x_{n-(n_a - 1)} \quad \text{(IV.20)}$$

En supposant que les erreurs sont indépendantes, la variance à la sortie 2 due à la quantification du CAN sera de :

$$\sigma^2_{21} = \sigma^2_{\varepsilon} \sum_{k=0}^{n_a} h^2_k \approx \left[n_a \left(\frac{T_{clk}}{\tau_d}\right)^2 + 2 \right] \sigma^2_{\varepsilon} \approx 2\sigma^2_{\varepsilon} = 2\frac{2^{-2B}}{12} \quad \text{(IV.21)}$$

Où l'on suppose que : $n_a = \dfrac{\tau_{pk}}{\tau_{clk}} \gg 1, \quad \dfrac{\tau_p}{\tau_d} \leq 8 \quad et \quad \dfrac{\tau_{clk}}{\tau_d} \leq \dfrac{1}{50}$

Par exemple : $T_{clk} = 0.02\,\mu\text{sec}, \ \tau_d = 5\,\mu\text{sec}, \ \tau_{pk} = 3\,\mu\text{sec}$ ce qui donne : $n_a = 250 \gg 1$

$$\frac{\tau_{pk}}{\tau_d}\frac{T_{clk}}{\tau_d} = \frac{3}{5}\frac{20}{5000} = 0.0024 \ll 2 \quad \text{(IV.22)}$$

2) Erreur à la sortie du multiplicateur :

La deuxième source d'erreur est liée au calcul de la somme des produits. Si la somme des produits est calculée de telle manière que l'arrondi ou la truncate est effectué après chaque multiplication l'erreur peut être modélisée comme un bruit superposé à la valeur exacte de la somme des produits.

Dans ce cas, on peut montrer que la variance du bruitplusieurs fois plus n_a est σ^2_2 grande que la variance due à une seule multiplication. Si la multiplication est effectuée avec une précision de $\sigma^2_{22} = n_a \dfrac{2^{-2B_m}}{12}$ alors : B_m

Par conséquent, le bruit total à la sortie de la deuxième fonction de transfert à la variance suivante :

$$\sigma^2_{22} = \sigma^2_{21} + \sigma^2_{22} = \frac{2^{-2B}}{12} + n_a \frac{2^{-2B_m}}{12} = \frac{2^{-2B}}{12}(2 + \frac{n_a}{2^{2(B_m - B)}}) \qquad \text{(IV.23)}$$

Ce bruit serait dominant dans le cas où B_m qui est aussi représenté par 13, comme l'est le signal d'entrée x_n. L'influence du second terme peut être calculée en tenant compte de ce qui suit : $B_m = 23$, $\max(n_a) = 2^{10}$,Ce qui donne une valeu r de $B = 13$

Il est à noter que B_m pourrait être moins précis avec 5 bits et pourtant sa contribution au bruit dû à l'arrondi serait 2 fois plus faible que celle due à la quantification du CAN. Mais nous devons le garder beaucoup plus précis à cause de son influence sur la forme du signal de sortie.

Nous pouvons donc écrire :

$$\sigma^2_2 = 2\frac{2^{-2B}}{12} = 2\frac{2^{-2*13}}{12} = (0.498*10^{-4})^2 \qquad \text{(IV.24)}$$

Le résultat ci-dessus peut être vérifier par une simulation en ramenant le temps de pic à $\tau_p = 3\mu\text{sec}$, précision du coefficient de la première fonction de transfert à 23.23 et précision du multiplicateur.

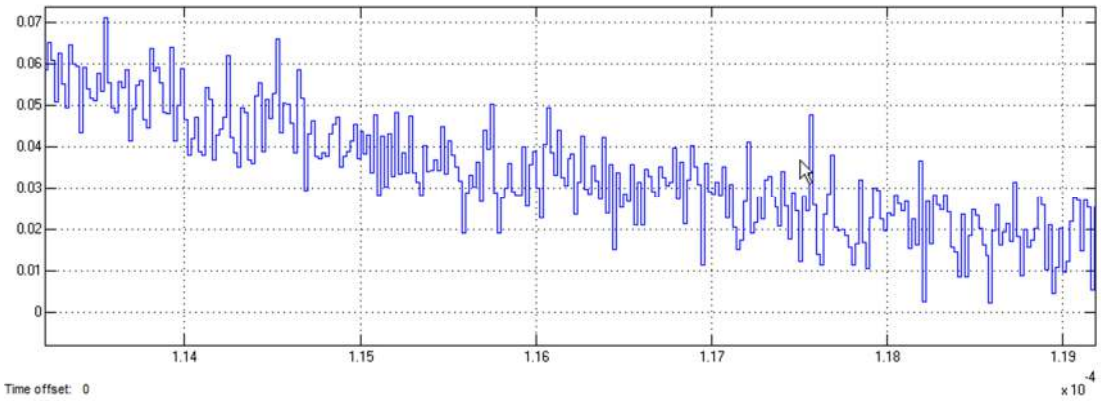


Fig. IV.11 : La sortie d'ADC quantification pour impulsion exponentielle.

En premier lieu, remarquez que la forme rectangulaire n'est pas discordante car les exigences en matière de précision de b_{10} et la multiplication dans la première fonction de transfert sont toutes deux suffisantes. Deuxièmement, remarquez que le "bruit" est plus petit que $3\sigma_2 = 1.5*10^{-4}$ ce que prédit le calcul ci-dessus. Cela signifie que seules les erreurs dues à la quantification de l'ADC contribuent aux erreurs dans la deuxième fonction de transfert, et elles apparaissent comme un "bruit".

3) Précision de la troisième fonction de transfert :

La troisième fonction de transfert ne contient que la sommation. La sommation est effectuée en pleine précision ; par conséquent il n'y aura pas de bruit dû à l'arrondi/la truncate. Seul le bruit de son entrée se propagera. La variance sur la sortie sera :

$$\sigma^2_3 = \sigma^2_2 \sum_{k=0}^{n_b-1} h^2_k = \sigma^2_2 \sum_{k=0}^{n_b-1} 1 = n_b \sigma^2_2 = 2n_b \frac{2^{-2B}}{12} \qquad \text{(IV.25)}$$

D'autre part, comme on le verra plus loin, ne contient qu'un multiplicateur par le facteur donc la variance globale du bruit due à la quantification sera de : $\sigma^2_4 = \frac{2n_b}{n_a^2} \frac{2^{-2B}}{12}$

Pour une valeur typique du détecteur de rayons gamma,

$\tau_d = 5\ \mu\text{sec},\ \tau_p = 3\ \mu\text{sec},\ \tau_{flop} = 2$.

$$\sigma^2_4 = \frac{2*200}{200^2} \frac{2^{-2*13}}{12} = \frac{2*250}{250^2} = \frac{1}{125} \frac{2^{-2*13}}{12} \cong 2^{-7} \frac{2^{-2*13}}{12} = \frac{2^{-2*16.5}}{12} \qquad \text{(IV.26)}$$

Cela signifie que la précision due à toutes les quantifications et à tous les arrondis est effectivement limitée à 16 bits, il est à noter que la précision de l'entrée est de 13 bits.

4) Précision de la quatrième fonction de transfert :

Il convient de souligner que la précision du signal rectangulaire à la deuxième sortie est de 14 bits, mais le résultat après la quatrième sortie est effectivement obtenu après avoir fait la moyenne des valeurs n_a qui ne diffèrent que par le bruit (les valeurs qui forment le rectangle). Le résultat est donc $\sqrt{n_a}$ fois meilleure précision. Puisque n_a est habituellement plus grand que 100 (temps de pic 2microsec), l'amélioration de la précision est au moins 10 fois supérieure, c'est-à-dire 3 bits. Cela signifie que la précision de sortie du multiplicateur de la quatrième sortie peut être augmentée de 3 bits supplémentaires.

IV.3.1.2 La précision du gain

Il existe deux sources principales d'erreurs qui affectent la précision de la sortie des étages, qui existent même dans le cas où le signal d'entrée dans l'ADC n'a pas de bruit. Il s'agit de :

- L'arrondi/la truncate du coefficient du filtre
- Quantification de l'ADC
- L'arrondi/la truncate effectué(e) pendant les opérations arithmétiques (multiplication).

Puisque l'entrée x_n dans le troisième système (sortie du système 2) est un signal de type delta, la sortie sera non nulle uniquement pendant n_α horloges successives.

Pendant cette période, la somme totale est toujours constante. Par conséquent, la sortie a la forme d'un signal rectangulaire de largeur égale à et l'amplitude étant égale à l'amplitude de la fonction delta d'entrée.

IV.3.1.3 Gain à la fréquence DC

En tenant compte de l'expression pour $H_3(z)$ et en supposant que l'entrée est une constante, alors la sortie n_b est la somme repliée de la constante. Cela signifie que le gain à la fréquence DC (par rapport à la sortie du système 2) est :

$$G_3 = n_b = \frac{\tau_p + \tau_{ft}}{T_{clk}} \qquad \text{(IV.27)}$$

On suppose que n_b peut être représenté par un nombre de 10 bits, $n_b \prec 2^{10}$. Par conséquent, la longueur du nombre de bit de l'accumulateur doit être 10 bits plus large que la sortie du système 2, soit 30 bits au total. La précision du soustracteur doit rester 14 bits à droite du point binaire. Il en ressort que le format du soustracteur doit être 30 bits avec 14 points binaire. Si le DC offset (courant continu) est supprimé de la sortie du système 2, la sortie aura le format 16 bit, avec 14 points binaire et par conséquent le système 3 devrait avoir : une largeur d'accumulateur de 26 bits et le format du soustracteur 26 bits avec 14 point binaire. Cette valeur fonctionnerait toujours pour tous les $\frac{\tau_p}{\tau_d} \prec 8$. [27]

Si l'on suppose que le signal d'entrée a un décalage DC offset DC= 0 et une amplitude $A_0 = 0$, alors le signal de sortie d'amplitude A sera :

$$A = A_0 G_1 G_2 G_3 = A_0 \frac{\tau_p}{\tau_d} \frac{\tau_p + \tau_{ft}}{T_{clk}} = 0.8 \frac{3}{5} \frac{3+2}{0.02} = 120 \qquad \text{(IV.28)}$$

IV.3.2 Implémentation avec code VHDL

L'algorithme récursif **(1)** qui convertit une impulsion exponentielle numérisée v(n) en une impulsion trapézoïdale symétrique s(n) est donné comme suit **:**

$$\begin{aligned} d^{k.l}(n) &= v(n) - v(n-k) - v(n-l) + v(n-k-l) && (1) \\ p(n) &= p(n-1) + d^{k.l}(n) && (2) \\ r(n) &= p(n) + Md^{k.l}(n) && (3) \\ s(n) &= s(n-1) + r(n) && (4) \end{aligned} \qquad \text{(IV.29)}$$

Où : v(n), p(n) et s(n) sont égaux à zéro pour n < 0.

Le paramètre **M** dépend uniquement de la constante de temps de décroissance τ de l'impulsion exponentielle et de la période d'échantillonnage **Tclk** du numériseur et est donné par l'équation suivante :

$$M = \frac{1}{e^{\left(\frac{T_{clk}}{\tau}\right)} - 1} \qquad \text{(IV.30)}$$

Les équations (3) et (4) définissent un algorithme récursif pour générer une forme trapézoïdale symétrique à partir d'un signal d'entrée exponentiel d'entrée exponentiel échantillonné avec l'ADC. Lorsque k =1, la réponse du système se traduit par une forme triangulaire symétrique. En utilisant l'aide des équations (3) et (4), une configuration matérielle simple peut être assemblée en utilisant seulement un petit nombre de blocs de traitement d'impulsions numériques standard. Un schéma fonctionnel simplifié de la mise en forme trapézoïdale est représenté à la Fig. IV.12 [26] :

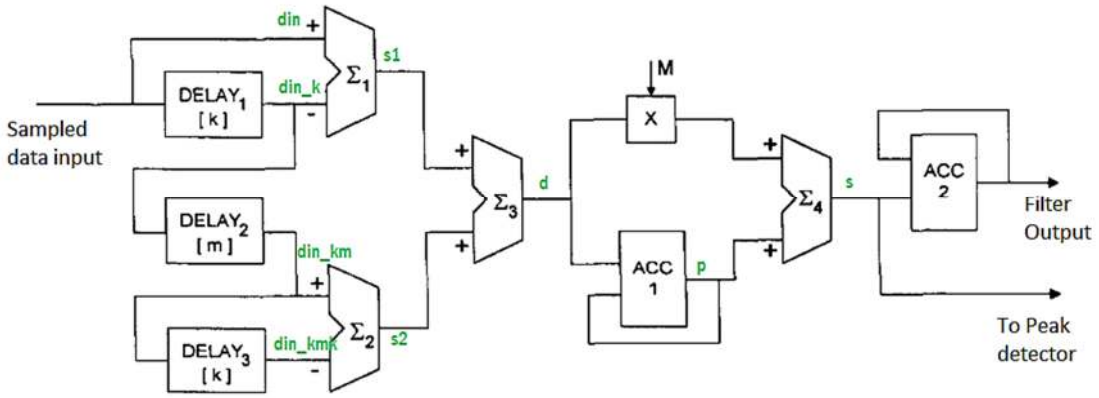


Fig. IV.12 : Schéma fonctionnel d'un système numérique trapézoïdal.

Trois pipelines à retard programmable sont utilisés pour fournir le décalage temporel nécessaire à l'échantillonnage du signal d'entrée. Dans les étapes suivantes, le signal passe à travers des unités arithmétiques selon l'algorithme décrit par les équations (3) et (4). Tous les blocs de construction sont des dispositifs arithmétiques à nombres entiers utilisant des variables signées.

Il convient de noter que le processeur numérique ne transmet qu'une seule valeur numérique par impulsion traitée correspondant à la valeur de crête de l'impulsion à l'analyseur multicanal. Les pics sont détectés en surveillant les changements de signe dans le résultat de l'addition qui précède l'accumulation finale.

Lorsque le niveau de bruit est élevé ou lorsque le CAN d'échantillonnage manque certaines séquences, des mesures doivent être prises pour éviter une détection erronée des pics, en introduisant les signaux de validations de données. [26]

Comme indiqué dans le code le schéma de filtrage comprend trois registres à décalage, deux soustracteurs, un multiplicateur, deux accumulateurs et un additionneur.

IV.3.2.1 Simulation du filtre numérique avec l'IDE Vivado

L'environnement de développement intégré Vivado, nous offre une interface graphique convivial, afin de tester le code VHDL grâce aux équations récurrentes développée précédemment.

```
10      use IEEE.STD_LOGIC_1164.ALL;
11      use IEEE.NUMERIC_STD.ALL;
12      use work.utils.all;
13
14      entity trapezoidal_filter is
15          port (
16              clk     : IN STD_LOGIC;
17              din     : IN STD_LOGIC_VECTOR (DATA_WIDTH - 1 DOWNTO 0);
18              din_dv  : IN STD_LOGIC;
19              dout    : OUT INTEGER;
20              o_dv    : OUT STD_LOGIC
21          );
22      end trapezoidal_filter;
23
24      architecture Behavioral of trapezoidal_filter is
25          TYPE sr         IS ARRAY (NATURAL RANGE <>) OF STD_LOGIC_VECTOR (DATA_WIDTH - 1 DOWNTO 0);
26          TYPE sr2 IS ARRAY (NATURAL RANGE <>) OF INTEGER;
27          SIGNAL din_k    : NATURAL RANGE 0 TO 2**(DATA_WIDTH) - 1 := 0;
28          SIGNAL d        : INTEGER := 0;
29          SIGNAL dv0, dv1, dv2, dv3 : STD_LOGIC := '0';
```

Fig. IV.13 : VHDL code du filtre trapézoïdal programmé par Python.

IV.3.2.2 Schéma RTL-viewer

Le FPGA utilisé dans notre travail Zynq dispose d'un grand nombre de slices DSP qui sont personnalisées et peuvent traiter les opérations de manière efficace. Le synthétiseur RTL reconnaît ces modèles et un alloue en conséquence les cellules logiques cellules correspondantes. En cliquant sur "Elaborated Design", sous RTL Analysis, dans le "Flow Navigator", on obtient le circuit numérique générique dérivé de la compilation (élaboration) du modèle VHDL.

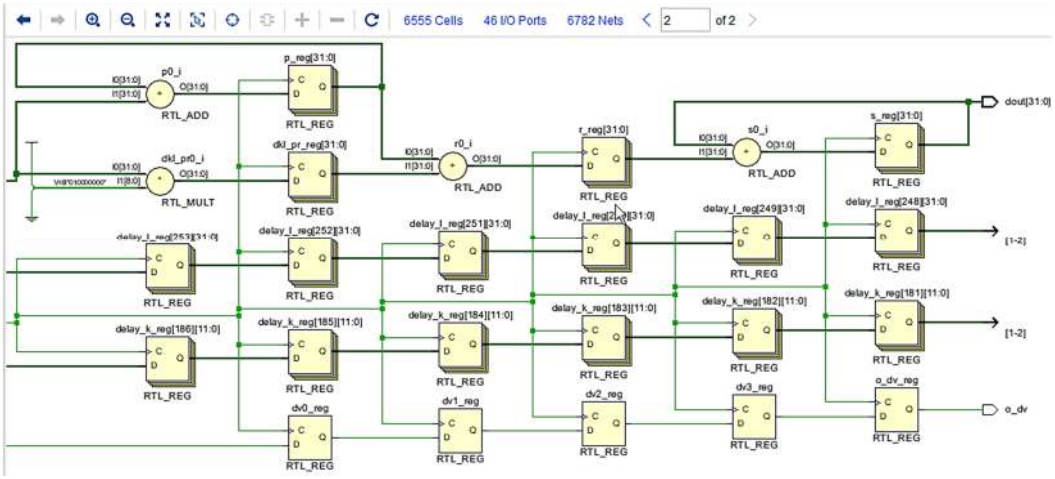


Fig. IV.14 : Schéma RTL viewer.

IV.3.2.3 Simulation de stimulus d'entrée Testbench

Afin d'effectuer une simulation VHDL, un code de test bench doit être généré comprenant le Design Under Test (DUT) et les vecteurs de test qui sont utilisés pour stimuler le DUT. Afin de simuler les sous-modules du filtre trapezoidal, des testbenchs simples ont été écrites à l'aide du logiciel XILINX Vivado. Pour l'entité trapezoidal_filter, des bancs d'essai ont été générés d'abord pour configurer l'objet sous test pour la simulation, puis appliquer les données de forme d'onde enregistrées à l'entrée de l'ADC de sorte que la sortie du filtre puisse être observée dans l'affichage de la forme d'onde du simulateur Vivado. Les formes d'onde enregistrées sont lues à partir d'un fichier et appliquées à l'entrée "din" du filtre numerique, comme illustré par le code suivant :

```
                                    -- Sample Code --
IF RISING_EDGE(CLK) THEN
        IF NOT(ENDFILE(sig_file)) THEN
                readline (sig_file, line_buffer) ;
                din<=STD_LOGIC_VECTOR (UNSIGNED (sample,DATA_WIDTH)) ;
            din_dv <= '1' ;
        ELSE
                din_dv <='0' ;
END IF ;
```

Ensuite, le code VHDL attend un cycle horloge, avant que la boucle ne recommence depuis le début. Le fichier filter_testbench.vhd contient une description matérielle permettant de réaliser cette tâche de manière autonome. L'entité filter_testbench contient une entrée, une sortie et deux signaux de validation de donnée.

L'impulsion d'entrée exponentielle a décroissance lente est converti par l'ADC et stockées dans un fichier texte, par l'intermédiaire d'un script python.

Les signaux de validation appelés handshake permettent de valider la transmission et la réception d'une donnée via l'interface AXI (Advanced eXtensible Interface). L'interaction avec le filtre numerique est modélisée comme un "process" Fdéclenché, il commence par la lecture du stimulus issu de l'ADC. Ensuite, ce "process" déclenche un autre "process" qui écrit la forme d'onde dans un fichier texte. Grâce au testbench développé, il a été possible de tester la fonctionnalité de l'implémentation du filtre trapezoidal avant tout test de l'algorithme dans un matériel.

IV.4. Interprétation des résultats

Le script (voir Annexe p.80) écrit sous forme textuel en langage python est utilisé pour calculer et enregistrer les coefficients de l'impulsion d'entrée exponentielle. Celui-ci constitue le stimulus d'entrée. Puisque l'ADC échantillonne le stimulus d'entrée en permanence, 4096 coefficients peuvent être calculés. Afin de réduire le temps de simulation, il est judicieux de limiter le nombre de canaux ADC à 12 bits.

Le choix des paramètres K, L et M sont important dans la suppression de l'effet de l'empilement et la restauration de la ligne de base. Le testbench écrit en VHDL réalise une tâche similaire, en sauvegardant l'onde de sortie du filtre numérique dans un fichier de sortie, dont le tracé peut être réalisé par l'intermédiaire d'un script python. Le résultat obtenu est une forme trapézoïdale. Ce qui justifie la validité de l'implémentation utilisée.

La première partie du code constitue la déclaration des bibliothèque python à utiliser. La seconde partie est une initialisation des paramètres clefs du filtre. La troisième partie permet de tracer l'impulsion exponentielle en rendant le signal causal. La quatrième partie contient les équations récurrentes de la mise en forme trapézoïdale. Enfin, la dernière partie du code trace la forme de l'onde trapézoïdale.

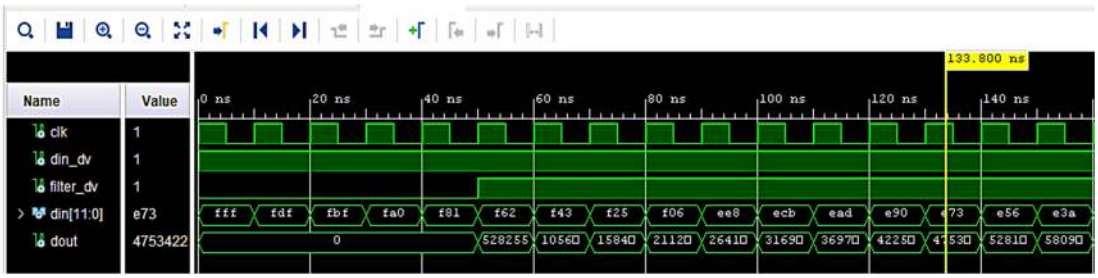


Fig. IV.15 : Simulation de l'onde de sortie

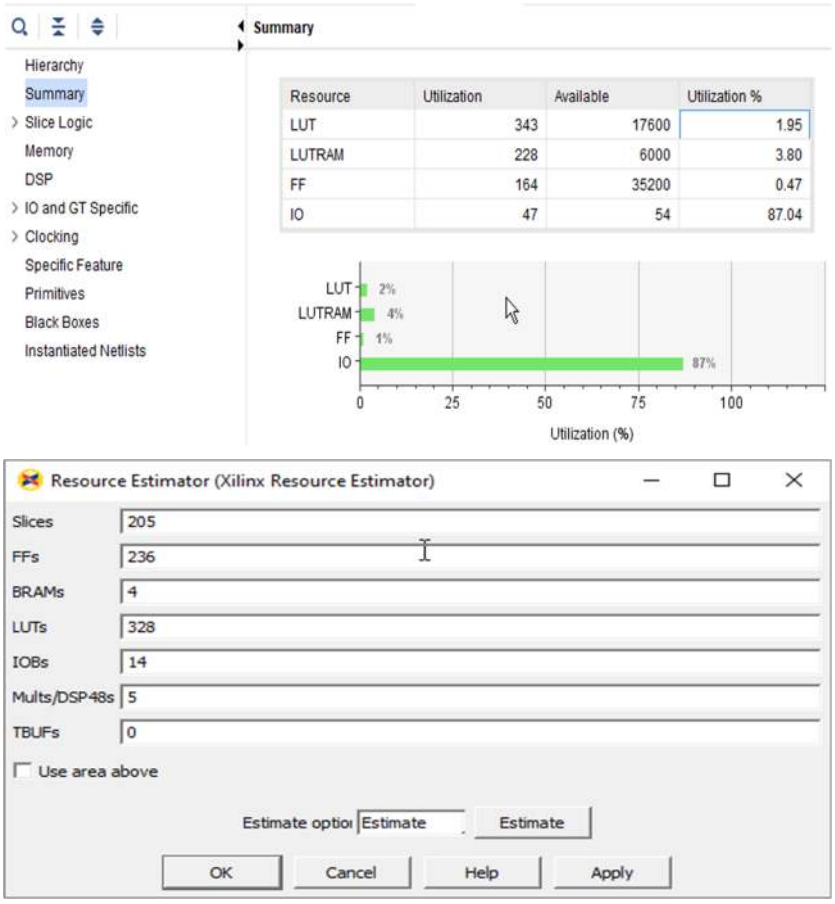


Fig. IV.16 : Utilisation des ressources VHDL et System generator xilinx.

IV.4.1 Comparaison des deux méthode (VHDL et System generator Xilinx)

Lorsqu'il est question de performances et d'une reponse temps-reels, l'implémentation en VHDL est la solution optimale, car elle consomme moins de ressources matérielles à l'intérieur du FPGA et possede une faible latence. Les tableaux precedent résument la consommation en termes de ressource, du filtre numérique implémenté. Xilinx System Generator génère des codes VHDL de manière automatisée, il en resulte que ces codes sont peu lisibles pour un humain, et par consequent, peu pratiques. De plus, l'utilisation des ressources ne peut pas etre maitrisée. L'implémentation avec Xilinx est capable d'inclure des "delay-line", les registres et des blocs DSP de manières différentes et moins professionnelle. Par contre, l'implementation en ecrivant un code VHDL fait correspondre efficacement les structures dans l'architecture matérielle, consommant ainsi très peu ressource. Ceci place le VHDL comme etant une solution avantageuse par rapport à un grand nombre de solutions.

IV.5. Conclusion

La simulation VHDL du filtre trapézoïdal a été réalisée pour des séquences exponentielles simples et bruité. La sélection des paramètres de mise en forme (c'est-à-dire K et L) joue un rôle essentiel dans la réduction de l'effet d'empilement. L'algorithme peut être mis en œuvre dans un réseau de portes programmables (FPGA) pour effectuer le filtrage trapézoïdal dans des applications en temps réel.

Conclusion Générale

Conclusion Générale

En effet, si le débit d'impulsion est trop élevé, les besoins en temps réel paralysent l'acquisition du signal durant le traitement d'une impulsion. Durant ce délai, appelé temps mort, des impulsions peuvent être perdues. Cette contrainte conduit les architectures actuelles à utiliser des solutions dédiées à base de FPGA. Les systèmes numériques basés sur le FPGA offrent une grande flexibilité pour le renouvellement de l'instrumentation nucléaire dédiée aux activités de recherche et de développement.

Dans ce travail, nous avons montré que l'utilisation d'outils logiciels tels que Vivado IDE de Xilinx Inc et Matlab system generator, peut grandement simplifier et accélérer la conception d'un processeur d'impulsions numériques pour spectrométrie à rayons γ à haute résolution. Un tel processeur d'impulsions numériques présente de meilleure performance que les systèmes analogiques avec des taux de comptage élevés. En considérant un signal d'entrée d'un préamplificateur à décroissance exponentielle, on peut démontrer qu'un filtre numérique linéaire peut être implémenté en utilisant simplement un ensemble de primitives micrologiciels comme des éléments de retard, des additionneurs, multiplicateurs et accumulateurs. L'implémentation en VHDL est la solution optimale, car elle consomme moins de ressources matérielles à l'intérieur du FPGA. Les sources d'erreurs susceptibles de paralyser la chaine de mesure spectroscopique, en y introduisant des temps-mort ont été examinées en conduisant une simulation du filtre trapézoïdale numérique avec des paramètres recommandés.

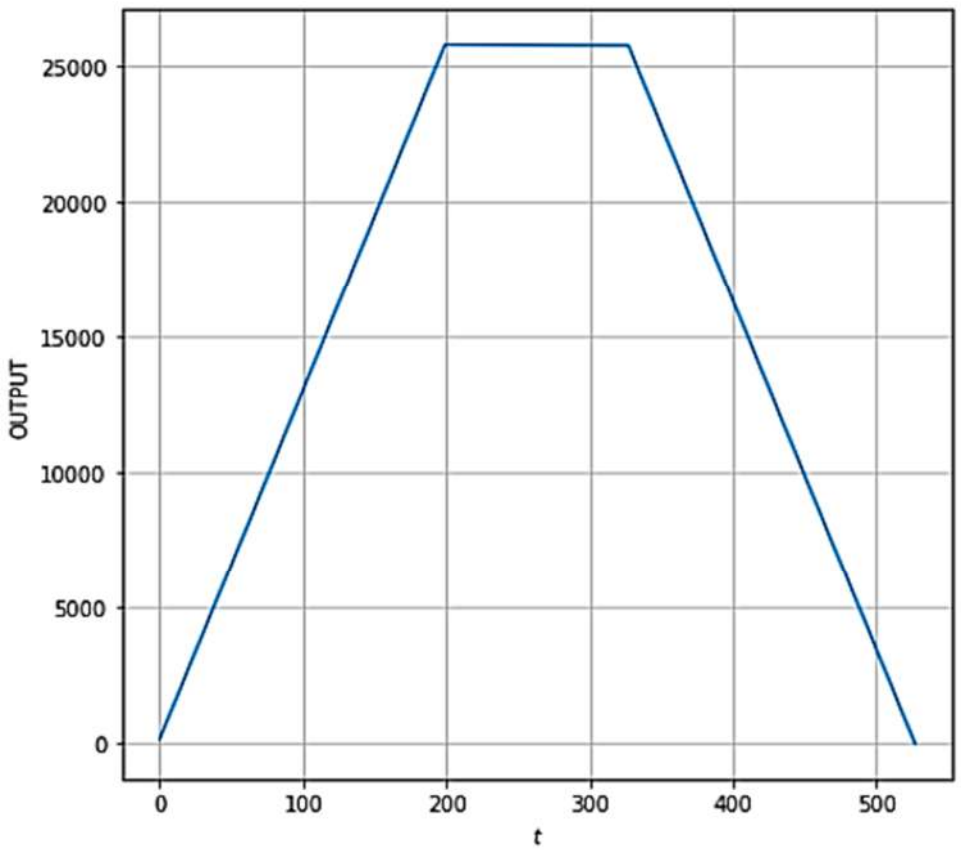
Annexe

A- Script en Python pour la génération de l'entrée exponentielle :

```python
# -*- coding: utf-8 -*-
import numpy
import numpy as np
import matplotlib.pyplot as plt
import warnings
import os
import json
import math
# Suppress warnings
warnings.filterwarnings('ignore')
# Render math equations in plot labels
# plt.rcParams['text.usetex'] = True
LENGTH = 528
TAU = 128 # LENGTH // 3
T1 = LENGTH // 3
T2 = 2 * LENGTH // 3
t = np.arange (LENGTH)
exponential = np.exp (-(t/TAU))
plt.figure (figsize=(7, 7))
plt.plot (t, exponential)
plt.xlabel ("$t$")
plt.ylabel ("$y$")
plt.grid (visible=True)
plt.show ()
# M MUST be equal to TAU
M = TAU
K = (LENGTH - TAU) // 2
L = M + K
p = 0
s = np.zeros_like (exponential)
def get_exp (idx):
    return 0 if idx < 0 else exponential[idx]
file=open(r"C:\Users\ALLADJO\Documents\EXP.TXT","w")
for a in range(LENGTH):
    data=int(exponential[a]*((2**12)-1))
    file.writelines(str(data)+'\n')
file.close()
plt.figure (figsize=(7, 7))
plt.plot (t,s)
plt.xlabel ("$t$")
plt.ylabel ("OUTPUT")
plt.grid (visible=True)
plt.show ()
```

B- Script en Python pour la génération de la sortie Trapézoïdale :

```
# -*- coding: utf-8 -*-
import numpy
import numpy as np
import matplotlib.pyplot as plt
import warnings
import os
import json
import math
# M MUST be equal to TAU
M = TAU
K = (LENGTH - TAU) // 2
L = M + K
p = 0
s = np.zeros_like (exponential)
def get_exp (idx):
    return 0 if idx < 0 else exponential[idx]
file=open(r"C:\Users\ALLADJO\Documents\FILTER_OUT.TXT","w")
for a in range(LENGTH):
    data=int(exponential[a]/((2**12)-1))
    file.writelines(str(data)+'\n')
file.close()
plt.figure (figsize=(7, 7))
plt.plot (t,s)
plt.xlabel ("$t$")
plt.ylabel ("OUTPUT")
plt.grid (visible=True)
plt.show ()
```



Références

[1] Yinyu Liu, Hao Xiong, Chunhui Dong, Chaoyang Zhao, Quanfeng Zhou, and Shun Li. Real-time signal processing in field programmable gate array based digital gamma-ray spectrometer. Review of Scientific Instruments, 91(10):104707, 2020.

[2] Strahinja Lukic. Mesure de sections efficaces de réactions (n, xn) par spectroscopie gamma prompte auprès d'un faisceau à très haut flux instantané. PhD thesis, Université Louis Pasteur-Strasbourg I, 2004.

[3] Glenn F Knoll. Radiation detection and measurement. John Wiley & Sons, 2010.

[4] Nicholas Tsoulfanidis and Sheldon Landsberger. Measurement & detection of radiation. CRC press, 2021.

[5] Russell David Bingham. A quasi-cusp digital filter for gamma-ray spectroscopy. The University of Tennessee, 1996.

[6] A Messai, A Nour, I Abdellani, et al. Digital signal processing for optimal résolution in gamma ray spectroscopy. 2013.

[7] Andrey Georgiev and Werner Gast. Digital pulse processing in high résolution, high throughput, gamma-ray spectroscopy. IEEE Transactions on nuclear science, 40(4): 770–779, 1993.

[8] Thierry Legou. Etude et réalisation d'une chaîne d'instrumentation numérique rapide pour l'identification des ions. PhD thesis, Université de Caen, 2002.

[9] V Radeka. Signal, noise and résolution in position-sensitive detectors. IEEE Transactions on Nuclear Science, 21(1):51–64, 1974.

[10] Roy M Howard. Principles of random signal analysis and low noise design: The power spectral density and its applications. John Wiley & Sons, 2004.

[11] Veljko Radeka. 1/|f| noise in physical measurements. IEEE Transactions on Nuclear Science, 16(5):17–35, 1969.

[12] Alberto Pullia and Stefano Riboldi. Time-domain simulation of electronic noises. IEEE Transactions on Nuclear Science, 51(4):1817–1823, 2004.

[13] Marshall Leach Jr. W.: Fundamentals of low-noise electronics. Proc. IEEE, 82(10): 1515–1538, 1994.

[14] Paul W Nicholson. Nuclear electronics. 1974.

[15] M. L. Simpson & al., "An ultra high-Throughput, High-Resolution, Gamma-Ray Spectroscopy System", IEEE Trans. Nucl. Sci., vol. 38, N°. 2, Apr. (1991).

[16] E. Gatti & al., “ Automatic synthesis of optimum filters with arbitrary constraints and noise : A new method ”, Nucl.Instr. and Meth., A381, (1996).

[17] A. Geraci & al., “ Optimum filter for charge measurement in the presence of 1/f current noise ”, Nucl.Instr. and Meth., A361, pp. 277-289, (1995).

[18] Abdel Kader Bousselham. FPGA based data acquisition and digital pulse processing for PET and SPECT. PhD thesis, Fysikum, 2007.

[19] Louise Helen Crockett, Ross Elliot, Martin Enderwitz, and Robert Stewart. The Zynq book: embedded processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 all programmable SoC. 2014.

[20] IP Xilinx LogiCORE. Microblaze micro controller system (v1. 1). DS865. 2012.

[21] Arria V. Device Handbook. Volume 3: Hard processor system technical reference manual. 2012.

[22] Haoliang Qin. Boost software performance on Zynq-7000 AP SoC with neon. Xilinx XAPP1206, 2014.

[23] E.Gatti & al., “Optimum filters for experimentally measured noise in high resolution nuclear spectroscopy”, Nucl.Instr. and Meth.Vol.A417,pp.131-136, (1998).

[24] P. Stoica, R. L. Moses, “Introduction to spectral analysis” Prentice Hall (1997).

[25] J. G. Proakis & D. G. MANOLAKIS, “Digital Signal Processing”, principles, algorithms, and applications”, Third Edition Prentice Hall (1996).

[26] JORDANOV, Valentin T. et KNOLL, Glenn F. Digital synthesis of pulse shapes in real time for high resolution radiation spectroscopy. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1994, vol. 345, no 2, p. 337-345.

[27] M. Bogovac, 'Digital Pulse Shaping ', Implementation of a Trapezoidal filter in a FPGA by using MATLAB & Xilinx design tools, ICTP-IAEA Workshop on Advances in Digital Spectroscopy 6 - 10 May 2013

[28] Laurent Leterrier, Acquisition Multi-détecteurs ou «L’électronique au cœur des expériences de physique», 2017.

[29] L. Netterdon, M. Elvers, J. Endres, A. Hennig, A. Sauerwein, F. Schlüter, and A. Zilges, Experimental facilities for reaction studies in Nuclear Astrophysics in Cologne, 2012.

[30] Kolbasz, Caesium-137 Gamma Ray Spectrum-de.svg, 4 mars 2015, https://fr.m.wikibooks.org/wiki/Fichier:Caesium-137_Gamma_Ray_Spectrum-en.svg.

[31] "Acquired gamma spectrum of Cs-137 preconcentrated on copper ferrocyanide impregnated filter cartridge", Radiation protection and Environment, https://www.rpe.org.in/viewimage.asp?img=RadiatProtEnviron_2015_38_3_72_169368_f3.jpg
[32] Little Shadow," Détection des rayonnements/Appareils de radioprotection", dans Radioprotection le 24 Août 2015 à 17:43, http://coursenvrac.eklablog.com/detection-des-rayonnements-appareils-de-radioprotection-a118544222
[33] F. Zocca, A. Pullia, D. Bazzacco, G. Pascovici, " Wide dynamic range front-end electronics for gamma spectroscopy with a HPGe crystal of AGATA", Published in 2007 IEEE Nuclear Science Symposium Conference Record 2007.
[34] Claude Gimenès, Les cours de Claude Gimenès, "Caractéristiques du bruit de fond", 2013 – 2022.
[35] Strahinja LUKIĆ, “Mesure de sections efficaces de réactions (n,xn) par spectroscopie γ prompte auprès d’un faisceau à très haut flux instantané ”, Thèse de l’Université Louis Pasteur de Strasbourg, (2004).
[37] Ian Cutress, "Xilinx Launches Cost-Optimized Portfolio: New Spartan, Artix and Zynq Solutions ", September 27, 2016.
[38] Cristian Sisterna, " Zynq Architecture 7-Series FPGA Architecture ", ICTP.
[39] Roman Trogan, " Parallella Platform Reference Design ", March 27, 2013, https://www.adapteva.com/white-papers/parallella-platform-reference-design/
[40] L. H. Crockett, R. A. Elliot, M. A. Enderwitz and R. W. Stewart, The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC, First Edition, Strathclyde Academic Media, 2014.

Resumé

Une simulation de l'algorithme de filtrage trapézoïdal en langage de description du matériel VHSIC (VHDL) a été réalisée. Les séquences à décroissance exponentielle ont été considérées comme l'entrée du filtre. Ces types de signaux sont généralement obtenus après traitement (c'est-à-dire dans le domaine analogique) et numérisation des signaux du détecteur nucléaire. Le filtre trapézoïdal agit comme un filtre de mise en forme des impulsions dans un système typique de spectroscopie nucléaire pour optimiser la valeur du déficit balistique, le rapport signal/bruit et les pertes par empilement d'impulsions afin de faciliter une meilleure mesure du temps d'arrivée et de l'énergie de la particule de rayonnement. La simulation qualifie l'algorithme pour sa mise en œuvre dans un réseau de portes programmables (FPGA) pour des applications en temps réel.

Mots clé : Trapézoïdal, VHDL, FPGA, déficit balistique, empilement, Vivado, Xilinx.

Abstract

A simulation of the trapezoidal filtering algorithm in hardware description language (VHDL) has been performed. The exponentially decaying sequences were considered as the input to the filter. These types of signals are generally obtained after processing (i.e. in the analog domain) and digitization of the nuclear detector signals. The trapezoidal filter acts as a pulse-shaping filter in a typical nuclear spectroscopy system to optimize the ballistic deficit value, signal-to-noise ratio, and pulse stacking losses to facilitate better time measurement. arrival and energy of the radiation particle. The simulation qualifies the algorithm for its implementation in a programmable gate array (FPGA) for real-time applications.

Keywords: Trapezoidal, VHDL, FPGA, ballistic deficit, stacking, Vivado, Xilinx.

ملخص

تم إجراء محاكاة لخوارزمية الترشيح شبه المنحرف بلغة وصف الأجهزة . تم اعتبار التسلسلات المتدهورة بشكل كبير كمدخلات إلى المرشح. يتم الحصول على هذه الأنواع من الإشارات بشكل عام بعد المعالجة (أي في المجال التناظري) ورقمنة إشارات الكاشف النووي. يعمل المرشح شبه المنحرف كمرشح لتشكيل النبض في نظام التحليل الطيفي النووي النموذجي لتحسين قيمة العجز الباليستي ونسبة إشارة الضوضاء وخسائر تكديس النبضات لتسهيل قياس الوقت بشكل أفضل. تؤهل المحاكاة الخوارزمية لتطبيقها في مصفوفة بوابة قابلة للبرمجة (FPGA) لتطبيقات الوقت الفعلي.

كلمات مفتاحية: شبه منحرف ، VHDL، FPGA ، عجز باليستي ، تكديس ، Vivado، A.Xilinx