الجـمـهـوريـة الجـزائـريـة الديـمـقـراطـيـة
الشـعـبـيـة
République Algérienne Démocratique
et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement
Supérieur et de la Recherche
Scientifique

C.D.T.A

جـامعة سعد دحلب البليدة
Université SAAD DAHLAB BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoirede fin d'étude en vue de l'obtention du diplôme de

Master

Filière

Microélectronique

Thème-

Intégration d'un convertisseur Analogique numérique dans l'architecture d'un nœud de capteur.

Réalisé par :
Benlalam Asma
&
Halfaoui Maroua

Encadré par :
Mme. IZABOUDJEN Nouma
Ms. Bounemri Ammar

Année universitaire 2019 – 2020

Résumé

Un convertisseur analogique-numérique est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique, proportionnelle au rapport entre la grandeur analogique d'entrée et la valeur maximale du signal. Le signal converti est plus souvent une tension électrique .

Dans le cadre de ce projet de fin d'études Master, il s'agit d'implémenter un CAN dans l'architecture d'un nœud de capteur implémenté sur un circuit FPGA. la partie pratique correspond à l'intégration d'un CAN tdc son code appartiennent au domaine public (opensource/opencores) . L'ensemble constitue un système dans un seul circuit ;L'outil de conception Vivado est utilisé pour la synthèse et l'implémentation du CAN sur le circuit FPGA Arti 7 deXilinx hébergé sur la carte de développement Nexys 4 DDR de Diligent disponible au niveau du CDTA.

La fonctionnalité du système implémenté est validée pour l'acquisition et l'affichage des informations provenant d'un capteur de température hébergé aussi sur la carte de prototypage FPGA Nexys 4 DDR.

Abstract

An analog-to-digital converteris an electronicassemblywhosefunctionis to translate an analogquantityinto a digital value, proportional to the ratio between the input analog quantity and the maximum value of the signal. The converted signal ismostoften an electrical voltage.

As part of thismaster's graduation project, the aimis to implement an ADC in the architecture of a sensornodeimplemented on an FPGA circuit. thepractical part corresponds to the integration of a CAN tdcits code belongs to the public domain (opensource / opencores) (opensource / opencores). The wholeis a system in one circuit. The Vivado design toolisused for the synthesis and implementation of ADC on the Xilinx Artix 7 FPGA circuitryhosted on Diligent'sNexys 4 DDR developmentboardavailableat the CDTA.

The functionality of the implemented system isvalidated by usingit for the acquisition and display of information from a temperaturesensoralsohosted on the Nexys 4 DDR FPGA prototypingboard.

ملخص

، لمحولالتناظريإلىالرقميعبارةعنتجميعإلكترونيتتمثلوظيفتهفيترجمةكميةتناظريةإلىقيمةترقمية
غالبًاماتكونالإشارةالمحولةعبارةعنجهدكهربائي .متناسبةمعالنسبةبينكميةالإدخالالتناظريةوالقيمةالقصوىللإشارة.

فيبنيةعقدةاستشعارمطبقةعلىدائرة tdc ADC فإنالهدفهوتنفيذ ، كجزءمنمشروعالتخرجلهذاالماجستير FPGA.
التييتمتصميكودبرنامجهإلىالمجالالعام CAN tdc الجزءالعمليتوافقمعتكامل (مفتوح / مفتوحالمصدر).
علىدائرة ADC لتركيبوتنفيذVivado تُستخدمأداةتصميم ؛ كلشيءيشكلنظامًافيدائرةواحدة Arti 7 FPGA
المستضافةعلىلوحةتطويرXilinxمن Diligent Nexys 4 DDR المتاحةفيمن CDTA.

يتمالتحققمنصحةوظيفةالنظامالمنفذللحصولعلىالمعلوماتوعرضهامنمستشعردرجةالحرارةالموجودأيضًاعلىلوحةالنماذجالأولية
Nexys 4 DDR FPGA.

Remerciements

Le travail présenté dans ce manuscrit est réalisés au sein de la division microélectronique et nanotechnologies (DMN) du Centre de Développement des Technologies Avancées (CDTA) à Alger.

Nous tenons à exprimer nos sincères remerciements et reconnaissances, tout d'abord, à notre encadreur Mme. IZEBOUDJEN Nouma d'avoir proposé ce sujet qui nous a mené vers la découverte du monde de l'expérimental et qui nous a bien accueilli pendant toute la période de notre stage, elle nous a bien dirigé afin de pouvoir réaliser ce travail de mémoire, nous n'oublions pas de la remercier pour sa disponibilité et ses conseils précieux.

De même, nous adressons notre profonde gratitude à notre promoteur Ms.BOUNEMERI Ammar et notre chef de spécialité Ms.NACER Said pour sa confiance, son soutien et tous ses conseils précieux.

Sur un plan plus personnel, nous remercions vivement nos parents pour leurs encouragements et leur soutien pendant notre cursus, ainsi que tous les membres de nos chères familles et tous nos chère amis.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin au bon déroulement de ce travail.

Table des matières

Table des figures

LISTE des tableaux

Liste des abréviations

CAN : Convertisseur Analogique Numérique

BRAM : Bloc RAM

CMOS : ComplementaryMetalOxideSemiconductor

CPLD : Complex Programmable LogicDevice

CPU : Central Processing Unit

DARPA : Defence Advanced ResearchProjects Agency

DDR : Double Data Rate

DSN : DistriutedSensor Network

DSP : Digital Signal Processor

EEPROM : ElectricallyErasable Programmable Read Only Memory

FPGA : Field Programmable Read Only Memory

GPS : Global Positioning System

IP : IintellectualProperty

JTAG : Joint Test Action Group

LSB : LeastSignificant Bit

LUT : Look Up Table

MAC : Multiply and Accumulate : Medium Access Control

MEMS : Micro Electro-MecanicalSensor

MSB : Most Significant Bit

PAL : Programmable LogicArray

PLD :programmableLogicDevice

RCSF : Réseaux de Capteur Sans Fils

ROM : Read Only Memory

RTL :Register Transfer Level

SCL : Serial Clock

SDA : Serial Data

Asic : Application SpecificIntegreted Circuit

I2C : Internet Integrated Circuit

VHDL :Very high speep Hardware Description Language

XDC :Xilinx Design Constraint

MHz :Mega Hertz

LCD :Liquid Crystal Display

CPU : CentralProcessing Unit

CTL :CenturyLink

CLB : ConfigurableLogic Bloc

IOB : Input Output Bloc

DMA : Direct Memory Access

Chapitre 1

Les nœuds de capteur basés sur FPGA

1.1 Généralité

A travers ce chapitre nous allons présenter l'architecture générale des nœuds de capteursen citant ces différents blocs et en se basant sur le CAN(convertisseur analogiquenumérique) ; nous intéressions par la suite au domaines d'application , enfin nous présentons quelques contraintes qui peut attendentles nœuds de capteurs.

I.2 l'architecture d'un nœud de capteur

De nos jours, grâce à la technologie avancée et plus particulièrement à l'évolution de la - microélectronique et de la communication sans fil, les réseaux de capteurs ont connu un- essor considérable. Généralement, ces réseaux se composent d'un grand nombre denœuds de capteur qui sont des entités autonomes miniaturisées dotés d'un moyen qui leurs permet d'effectuer des calculs et de communiquer les uns aux autres à travers une transmission multi- sauts sans fil.

Selon le type de l'application, il existe une multitude de nœuds de capteurs sur le marché :des nœuds de capteurs de température, d'humidité, de pression, etc. Malgré cette diversité considérable , ils partagent une architecture matérielle similaire. Un nœud de capteur est composé principalement de cinq unités : captage, traitement, stockage, communication, et énergie. Des composants additionnels peuvent être ajoutés selon le domaine d'application, [comme par exemple un système de localisation tels qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur qui lui permettant de se déplacer. La figure 1.1 suivante, extraite de la source [1]**,** est une illustration deCes éléments principaux et optionnels des composants d'un noeud de capteur.

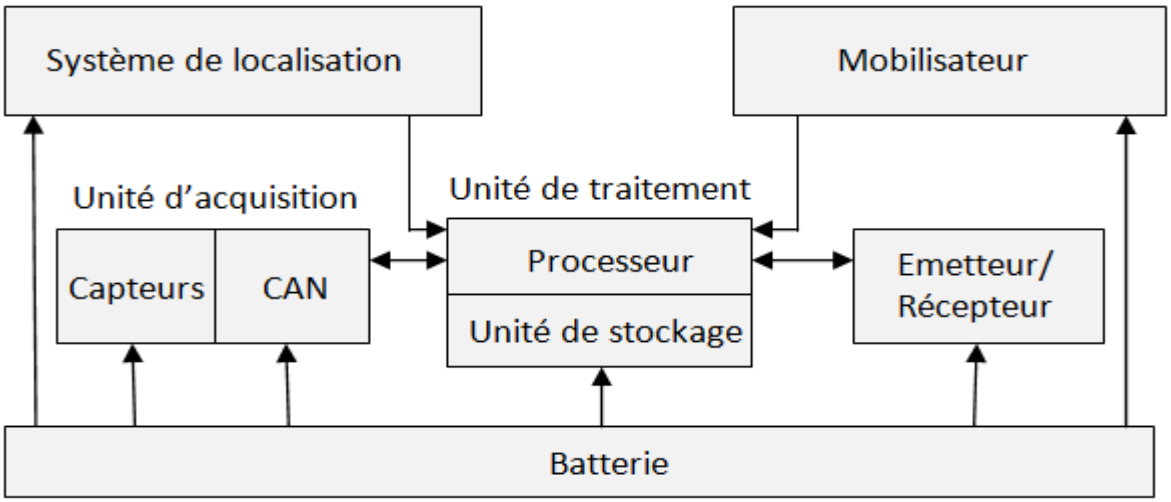


Figure 1.1 : Architecture matérielle d'un nœud de capteur

Les nœuds capteurs déployés en grand nombre, sont capables de récolter et de transmettre d'une manière autonome des données de leur environnement immédiat. La localisation géographique de ces nœuds dans le réseau n'est pas obligatoirement prédéterminée . Ils sont déployés dans une zone géographique appelée champ de capture, définissant le terrain d'intérêt pour le phénomène capturé. La figure 1.2 montre un exemple de nœud capteur MicaZ[2].

figure 1.2 :noeud capteur MicaZ

I.3les différentes blocs matérield'un nœud de capteur

I.3.1 L'alimentation

Un nœud de capteur possède une source d'énergie, généralement une batterie de type rechargeables , pour alimenter tous ses composants. Souvent, dans les environnements

sensibles, il est impossible de recharger ou de changer la batterie suivant la durée de vie des nœuds de capteurs.

Par ailleurs, un noeud capteur peut être doté d'autres unités. Citons, entre autres, la possibilité d'ajouter une unité de localisation, tel qu'un GPS (Global Positionning Système), une unité de mobilité pour assurer la mobilité du noeud capteur, ou une unité spécifique de capture comme une caméra pour l'acquisition vidé[3].

I.3.2 Unité de traitements (processeur)

cette unité constitue l'élément central d'un noeud capteur UCT (Unité central de traitement), elle est composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécialement conçus pour les capteurs. Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communication qui permettent de faire collaborer le noeud capteur avec d'autres noeuds, comme elle peut aussi analyser les données captées[3]**.**

I.3.3Unité de stockage (Mémoire)

L'unité de stockage inclut la mémoire de programme (dont les instructions sont exécutées par le processeur) et la mémoire de données (pour conserver des données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est limitée essentiellement par les considérations économiques et s'améliorera aussi probablement au fil des années[3].

I.3.4 Unité de communication (émetteur / récepteur)

Cette unité est responsable de toute émission et réception des données via un support de communication sans fil. Les différents choix de média de transmission incluent la radiofréquence (RF), le laser et l'infrarouge[4].

I.3.5 unité d'acquisition(capteur/CAN)

C'est l'unité qui nous intéresse le plus dans notre travail ; elle est composée du capteur proprement dit et du convertisseur Analogique/Numérique (ADC : Analog/Digital Converter) qui transforme les signaux analogiques en signaux numériques. En e□et, le capteur observe un phénomène et fournit les signaux analogiques correspondant au phénomène au convertisseur analogique /numérique, le ADC, qui les transforme en signaux numériques compréhensible par l'unité de traitement.

La fonction principale de l'unité de captage est de capturer ou mesurer les données physiques à partir de l'objet cible. Il est composé de deux sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur lephénomène observé, au convertisseur Analogique/Numérique(CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un nœud de capteur peut avoir un ou plusieurs unités de captage[3].

I.4 Les Applications typiques des nœuds de capteurs

La plus grande excitation à propos des nœuds de capteurs provient de l'utilisation d'un grand nombre de nœuds qui communiquent entre eux et forment des réseaux sans fil (RCSF).La miniaturisation, l'adaptabilité, le faible cout et la communication sans fil permettent à ces réseaux de capteur d'envahir plusieurs domaines d'application. Voici quelque exemple d'application potentielle dans différent domaines[5] :

I.4.1 Applications liés à la sécurité :

Les structures d'avions, navires, automobiles, métros, etc. pourraient être suivies en temps réel par des réseaux de capteurs, de même que les réseaux de circulation ou de distribution de l'énergie. Les altérations de structure d'un bâtiment, d'une route, d'un quai, d'une voie ferrée, d'un pont ou d'un barrage hydroélectrique pourraient être détectées par des capteurs préalablement intégrés dans les murs ou dans le béton, sans alimentation électrique ni connexions filaires[5].

Certains capteurs ne s'activant que périodiquement peuvent fonctionner durant des années, voire des décennies. Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisqu'il n'existe pas de point critique. La surveillance de routes ou voies ferrées pour prévenir des accidents avec des animaux ou des êtres humains ou entre plusieurs véhicules est une des applications envisagées des réseaux de capteurs [5].

I.4.2Applications militaires :

Comme dans le cas de la majorité des technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie, le projet DSN (DistributedSensor Network) au DARPA (Defense Advanced ResearchProjects Agency) était l'un des premiers projets dans les années 80 ayant utilisés les réseaux de capteurs pour rassembler des données distribuées. Un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection d'intrusions, détection des armes chimiques, biologiques ou radiations nucléaires) [3].

I.4.3Applications commerciales :

Il est possible d'intégrer des nœuds de capteur au processus de stockage et de livraison. Le réseau pourra être utilisé pour connaitre la position, état de la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières, via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les micro-capteurs en garantissent le bon état [3].

I.4.4 Applications domestiques :

Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière qui s'éteint et la musique qui se met en état d'arrêt quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, le déclenchement d'une alarme par le capteur antiintrusion quand un intrus veut accéder à la maison [5].

I.4.5Applications environnementales :

les réseau de capteur peuvent être utilisés pour surveiller les changement environnementaux ,ils servent à déterminer les valeurs de certains paramètres a un endroit donné ,comme par

exemple :la température ,la Pression atmosphériques a un endroit donné, comme par exemple : la température, la pression atmosphérique, etc.

En dispersent des nœuds de capteur dans la nature, on peut détecter des événements tels que des feux de forêts, des tempêtes au des inondations, ceci permet une intervention beaucoup plus rapide et efficace des secours. Dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques, par exemple en déclenchant le processus d'arrosage lors de la détection de zones sèches dans un champ agriculture [4]

I.4.6Applications médicales :

Dans l'application de la médecine, les réseaux de capteur peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humaine grâce à des micro-capteurs qui pourront être avalés au implantés sous la peau. Ils peuvent aussi faciliter le diagnostic de quelque maladies en effectuant des mesures physiologiques telles que : la tentions artérielle, battements du cœur, l'aide des capteurs ayant chacun une tache bien particulière.

Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient pour une ultérieure décision médicale [5].

I.4.7 Applications sportifs :

l'évolution des réseaux de capteurs est utilisée de plus en plus dans le domaine sportif, à savoir les systèmes de surveillance, les systèmes de calcule de trajectoires (comme dans le tennis), systèmes de détection d'erreur d'arbitrage (comme dans le football indiquent si le balle a franchi la ligne de but) [3]

Figure 1.3 : Exemples d'applications des nœuds de capteurs

I.5Caractéristiques et contraintes d'un nœud de capteur

La conception et la mise en place des nœud de capteur sont influencées par plusieurs contraintes qui peuvent être des contraintes conceptuelles ou matérielles. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs ; ils sont considérés également comme métriques de comparaison deperformances entre les différents travaux dans le domaine [4].

I.5.1 le problème de la localisation des noeuds de capteurs dans un milieu exempt d'erreurs :

les nœuds de capteur ce dépose d'une façon autonome sans utiliser une infrastructure fixe. Trois principales raisons justifient le besoin d'une telle localisation:

La connaissance des coordonnées d'un capteur est nécessaire pour les application qui se basent principalement sur la détection et le report des événements. Ces applications ont besoin de savoir exactement la position où l'événement s'est produit.

L'exemple de la localisation des mines et de la détection d'intrus dans un champ de bataille pour le cas d'une application militaire ainsi que l'exemple de détection d'incendie mentionné plus haut, en est une bonne illustration la localisation est aussi nécessaire pour les applications

de proximité. Celles-ci permettent à différents usagers physiquement proches les uns des autres de partager certaines de leurs informations et de localiser les données disponibles , la troisième raison réside dans le fait de permettre la gestion de certaines fonctionnalités spécifiques au réseau de capteurs. À titre d'exemple, certaines applications basées sur le routage géographique nécessitent de connaître les nouvelles coordonnées d'un noeud susceptible d'être en mouvement. Ces coordonnées vont pouvoir aider les applications à retracer la nouvelle route depuis une station source vers un noeud de capteur destinataire[2].

I.5.2Le niveau de gestion d'énergie :

Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs, dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un noeud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un noeud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de captage [2].

I.5.3 La tolérance aux pannes :

La défaillance ou le blocage de certains nœuds dans un réseau de capteurs peut être engendrés par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique ou les interférences liées à l'environnement. Ces problèmes ne devraient pas affecter le reste du réseau. C'est le principe de la tolérance aux pannes.

I.5.4 L'environnement :

Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie, humidité…

Les senseurs sont souvent déployés en masse dans des endroits tels que des champs de bataille au delà des lignes ennemies, au fond d'un océan, dans un volcan...

Par conséquent , ils doivent pouvoir fonctionner, sans surveillance dans des régions géographiques éloignées

I.5.5 Les médias de transmission :

Dans un réseaux sans-fil , les nœuds de capteur sont reliés par une architecture sans-fil tell que Le media de transmission doit être nommé ou standardisé ;On utilise le plus souvent la réalisation peut être par communication radio ,signal infrarouge ou un média optique.

I.5.6Le consommateur d'énergie :

La caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie, de calcul et de stockage. Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque nœud.

Un senseurs est limité en énergie et dans le plupart des cas le remplacement de la batterie est impossible.

La durée de vie d'un senseur dépend grandement de la durée de vie de la batterie.

Le dysfonctionnements de quelques noeuds nécessite un changement de la topologie du réseau, du routage des paquets et donc changement de la consommation d'énergie**.**

I.5.7 La topologie de réseau et L'extensibilité (passage à l'échelle) :

L'une des caractéristiques des RCSF est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie du réseauDéploiement initial ,post-déploiement et le Redéploiement de nœuds additionnels.

I.5.8 Agrégation de données :

Dans les nœuds de capteur , les données produites par les nœuds capteurs voisins sont corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'information redondante. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données, appelée aussi fusion des données.

On peut aussi définir des contraintes matérielles liées aux nœuds de capteur , citant :

***Dimension :**

La taille réduite des capteurs peut présenter de nombreux avantage, elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée par la petite taille de ces derniers.

***Puissance de calcul :**

Les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique carils utilisent souvent des microcontrôleurs de faibles fréquences .

La consommation d'énergie doit être minimale pour le réseau ;le senseur doit être autonomes et très résistants [6]

I.6CONCLUSION

Nous avons présenté dans ce chapitre une généralité sur les nœuds de capteur, ainsiQue leurs architecture matériel et leurs domaines d'applications. Les réseaux de capteurs sans fil ne cessent de prendre uneplace très appréciée au sein de la communauté de recherche vu leurs déploiement assez simpleet leurs applications qui se développent chaque jour pour élargir leurs horizons. Initialement,réservés pour les applications militaires, c dernier ont réussi à conquérir d'autres domaines civils plus larges et plus pratiques changeant lequotidien des êtres humains. Cependant, lescontraintes relatives au hardware du capteur, l'élément clé du réseau rendent la conception duréseau une tâche difficile, et nécessite beaucoup de travail sur le volet des softwares .Dans le chapitre suivant, nous allons aborder l'État de l'art des nœuds de capteur et plus précisément l'implémentation de CAN sur FPGA.

Chapitre 2

L'Etat de l'art sur l'implémentationde CAN sur FPGA

II.1 Introduction

Les FPGA, sigle anglais qui signifie « Field Programmable Gates Arrays » traduit en français par réseau de portes programmables, sont des circuits intégrés reprogrammables.

Ils offrent la possibilité de réaliser des fonctions numériques plus ou moins complexes, tout comme leurs homologues figés : les ASIC.

Structurés sous forme de matrices, les FPGA sont composés d'éléments logiques de baseconstitués de portes logiques, présentes physiquement sur le circuit. Ces portes sont reliées par un ensemble d'interconnexions modifiables : Dou l'aspect programmable du circuit.

Pour pouvoir être programmé, le FPGA a besoin d'une carte de développement. C'est unenvironnement de test et de conception.

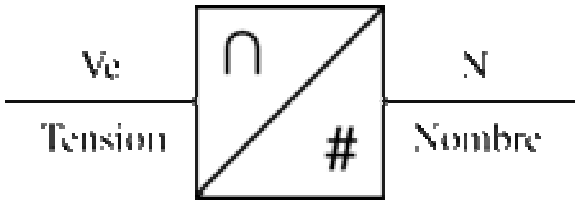
Pour communiquer avec le monde réel qui est analogique, nous avons besoin d'un convertisseuranalogique numérique (CAN, ou ADC pour Analog to Digital Converter en anglais)

Dans ce chapitre, nous allons présenter un état de l'art sur l'implémentation et / ou lacommunication des CAN avec les circuits FPGA. Nous commencerons, d'abord, par donner un aperçue générale sur les CAN ainsi que sur les circuits FPGA; ensuite nous présenterons quelques travaux de recherche sur l'implémentation des CAN avec les FPGA.

II.2Généralités sur les CAN

Un convertisseur analogique-numérique (CAN, parfois convertisseur A/N, ou en anglais ADC pour Analog to Digital Converter ou plus simplement A/D) est un montage électronique dont la fonction est de traduire une grandeur analogique en une valeur numérique (codée sur plusieurs bits), proportionnelle au rapport entre la grandeur analogique d'entrée et la valeur maximale du signal.

Le signal converti est le plus souvent une tension électrique.Le rôle d'un CAN est de convertir un signal analogique en un signal numérique pouvant être traité par une logique numérique.



FigureII.1:Symbole normé du convertisseur analogique numérique

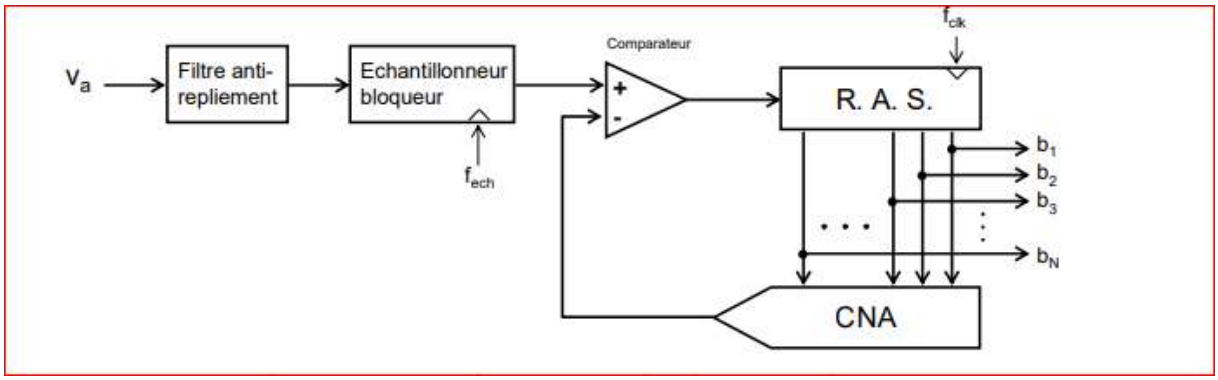
Il existe plusieurs techniques pour convertir un signal analogique en signal numérique. Elles sont classées ici dans l'ordre de la moins rapide à la plus rapide.

On distingue deux grandes familles de CAN basées sur deux approches différentes de l'échantillonnage : les CAN classiques dont la fréquence d'échantillonnage est telle que le spectre du signal converti occupe quasiment toute la bande de Nyquist (Nyquist Rate ADC) et les CAN à sur échantillonnage (Oversampling ADC) dont seule une partie réduite du bruit dequantification affecte le signal converti[6].

Citant quelques modèles différents des convertisseurs :

***Convertisseur à approximations successives :**

La figue II.2 donne la topologie d'un convertisseur à approximations successives (SAR converter, pour Successive Approximation Register).



FigureII.2:Convertisseur à approximations successives

Ce CAN possède une boucle de rétroaction, constituée d'un CNA de même résolution et d'un comparateur qui commande un Registre à Approximation Successive (RAS, qui donne sonnom à cette architecture).

Le principe de conversion est basé sur une recherche du code de sortie par dichotomie (figure II.2), à chaque coup d'horloge l'intervalle de recherche est divisé par 2. En début de

conversion tous les bits de sortie (RAS et CAN) sont positionnés à zéro à l'exception du MSB, b1, qui est fixé à un. Le mot binaire correspondant (100…0) est présenté au CNA qui délivre en sortie une tension Vref/2. Cette dernière est comparée à va. Si va est inférieur à Vref/2 alors b1 passe à zéro, dans le cas contraire il reste à un ; dans les deux cas il s'agit de la valeur finale de conversion du bit considéré.

C'est une architecture de conception ancienne, mais encore très répandue. On trouve des CAN SAR jusqu'à 18 bits et quelques MHz[7].

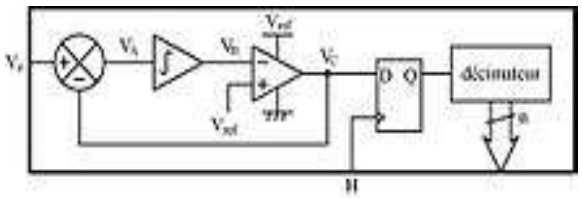
II.2.1Convertisseur Sigma Delta :

Ce type de convertisseur est basé sur le principe du suréchantillonnage d'un signal d'entrée.Un comparateur est en général utilisé pour convertir sur un bit (c'est-à-dire 0 ou 1) la différence (delta) entre le signal d'entrée et le résultat de la conversion (0=plus petit, 1=plus grand).Le résultat de la comparaison est alors entré dans un filtre appelé le décimateur, qui somme (sigma) les échantillons du signal d'entrée. Cela revient à calculer l'intégrale de la différence entre l'entrée et la sortie.

Cela crée un système asservi (la sortie est rebouclée sur l'entrée) qui fait osciller la valeur de l'intégrale du signal à convertir autour d'une valeur de référence (le résultat de la conversion).

La sortie numérique du comparateur est sur 1 bit à haute fréquence (la fréquence d'échantillonnage), qui est filtrée par le décimateur qui augmente le nombre de bits en réduisant la pseudo fréquence d'échantillonnage .L'intérêt de ce genre de convertisseur réside dans sa grande résolution de sortie possible (16, 24, 32 bits voire plus) pour des signaux d'entrée avec une bande passante modérée.

Ces convertisseurs sont adaptés à la conversion de signaux analogiques issus de capteurs dont la bande passante est souvent faible (par exemple les signaux audio). Les convertisseurs Sigma/Delta sont, par exemple, utilisés dans les lecteurs de CD dans le cas d'une conversion numérique-analogique.



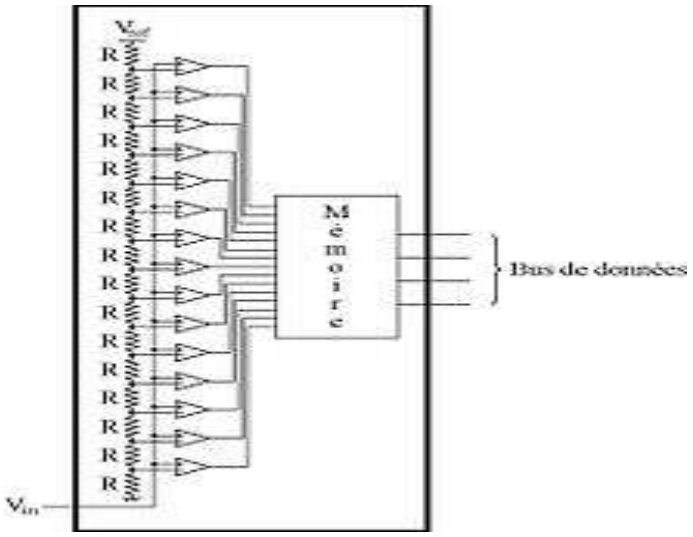
FigureII.2 :Schéma de principe d'un convertisseur Delta-Sigma

Les technologies Sigma-Delta ont quasi totalement remplacé les technologies à simple ou double rampe[8]..

II.2.2 Convertisseur flash :

Cette technique de conversion est très rapide, mais coûteuse en composants et donc utilisée pour les applications critiques.

La figure II. 3donne la topologie d'un convertisseur cations critiques.



La figure II. 3 : Structure d'un convertisseur flash

Il est difficile d'assurer une bonne linéarité de la conversion, car cela nécessite que toutes les résistances soient égales avec une précision d'autant plus grande qu'il y a un grand nombre de résistances.

Les convertisseurs Flash ont des temps de conversion inférieurs à la microseconde mais une résolution assez faible (de l'ordre de 8 bits) et sont souvent très chers[8].

II.3 Généralités sur les circuits FPGAs

Un FPGA est un réseau de portes qui peut être programmé pendant la production ou plus tard lors du déploiement. Ils sont produits en grand volume ,ce qui réduit le coût de chaque appareil.

Les fournisseurs de FPGA annoncent leurs plus grands appareils de «portes système» de plusieurs millions. Bien que peu les applications en ont réellement besoin, elles permettent d'implémenter d'autres produits. Gros appareils ont besoin des technologies de traitement au silicium les plus avancées. Étant donné que chaque famille FPGA comprend plusieurs membres allant du grand au petit nombre de portes, de petits appareils sont disponibles sur le même technologie. Avec la haute densité de grille disponible dans ces processus avancés

technologies, même les petits appareils fournissent des ressources suffisantes pour les intégrations de systèmes.

Du coup, les petits FPGA proposent des solutions qui n'étaient pas disponibles auparavant et ne coûtent que quelques-unes [9].

II.3.1Structure de base

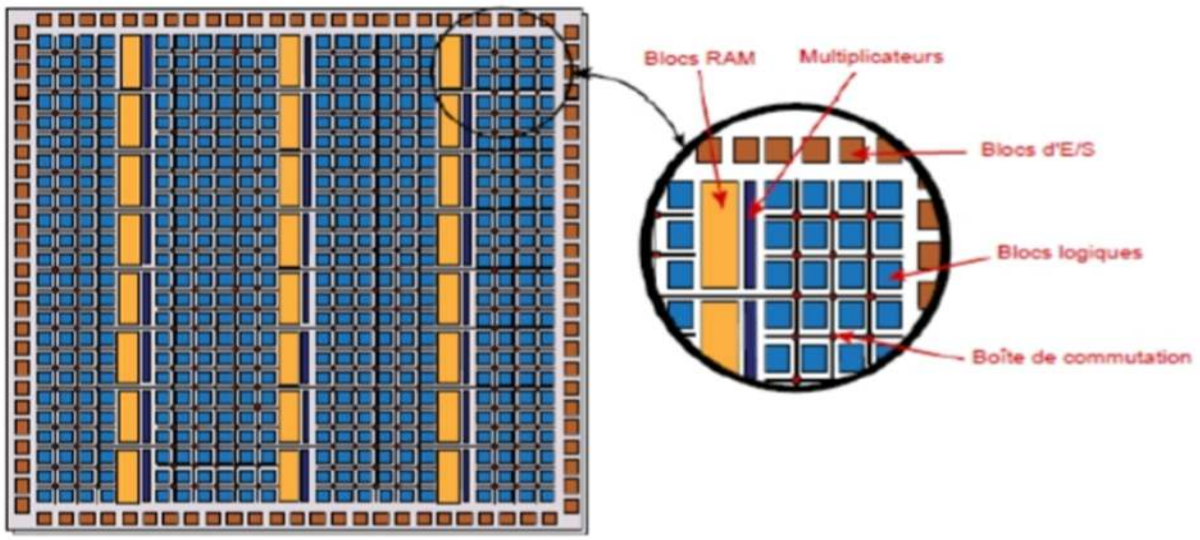
Un circuit intégré classique contient :

- des portes logiques,
- des connexions entre les portes logiques,
- des éléments de mémorisation (registres et/ou mémoires),
- des entrées/sorties,
- une (ou des) horloge(s)

Un circuit reconfigurable (circuit qui peut être reconfiguré après sa fabrication) a les mêmes éléments de base qu'un circuit statique avec la notion de reconfigurabilité. Partant du principe qu'une fonction logique peut s'exprimer sous forme canonique (somme de produits), cette fonction logique peut être représentée par un réseau logique programmable constitué d'une matrice ET et d'une matrice OU. De cette façon, en agissant sur les liaisons de la matrice ET ou la matrice OU, on peut changer la fonction réalisée. C'est ce qu'on appelle programmabilité/reconfigurabilité

II.3.2 les modèles les Plus commun des FPGA (Les circuits FPGAs de Xilinx)

Les circuits FPGA du fabricant XILINX, structurés sous forme des matrices, utilisent deux types de cellules de base, les cellules logiques appelées CLB (Configurable logic Bloc), et les cellules d'entrées/sorties appelées IOB, Ces différentes cellules sont reliées entre elles par un réseau d'interconnexions configurable. La figureII. 3.2 montre es différents blocs d'un circuit FPGA.



FigureII. 3.2.1 : Les différents blocs d'un FPGA Xilinx

* Les circuits FPGAs de la série-7 :

La série 7 des circuits FPGAs comprend 3 familles de circuits: ARTIX7, KINTEX7 et VIRTEX7 comme le montre la figure. Chacune de ces familles du circuit FPGA offre des performances en termes de nombre de cellules logiques, circuits DSP, transcrives (émetteurs-récepteurs : Ethernet), mémoire, nombre d'entrée sortie et tension d'utilisation. Comparée aux circuits KINTEX et VIRTEX-7, la famille ARTIX-7 offre les meilleurs performances en terme de consommation de puissance et de coût. Les circuits FPGAs de la famille KINTEX-7 et VIRTEX-7 sont conçues pour être utilisées dans les applicationsindustrielles. La famille VIRTEX-7 est utilisée dans les systèmes à très haute performances alors que la famille Kintex-7 offre un compromis entre le prix et les performances. Dans notre travail nous utilisons le circuit FPGA ARTIX-7 [5].

	ARTIX-7	KINTEX-7	VIRTEX-7
	Lowest Power & Cost	Industry's Best Price/Performance	Industry's Highest System Performance
Logic Cells	20K – 355K	30K – 410K	285K – 2,000K
DSP Slices	40 – 700	120 – 1540	700 – 3,960
Max. Transceivers	4	16	80
Transceiver Performance	3.75Gbps	6.6Gbps 10.3Gbps	10.3Gbps 13.1Gbps 28Gbps
Memory Performance	800Mbps	2133Mbps	2133Mbps
Max. SelectIO™	450	500	1200
SelectIO™ Voltages	3.3V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below

Table II. 3.2.2 : Les différentes familles des circuits FPGAs série-7

II.4 Etat de l'art sur l'implémentation des CAN sur FPGA

les petits FPGA proposent des solutions qui n'étaient pas disponibles auparavant, a fin de voir comment réaliser l'interface entre un nœud de capteur implémenté sur FPGA et le CAN nous allons inspirer de quelques travaux et simulation(les différentes implémentations FPGA) déjà réalisée .

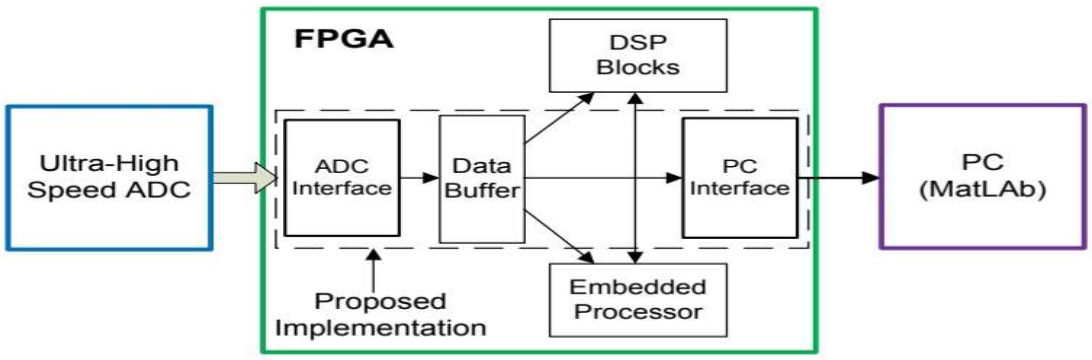
II.4.1 implémentationd'une interface ultra rapide ADC sur FPGA :

D'après l'article [10] le convertisseur analogique-numérique **(ADC**) est uncomposant dans la plupart des applications les plus exigeantes dans le monde de l'électronique aujourd'hui. . Ces dernières années,l'amélioration des architectures et de la technologie ADC etle développement rapide de FPGA hautes performances a faciliter l'interface entre eux

Les auteurs proposent une méthode de mise en œuvre pour interconnecter avec succès un ADC ultra-rapide et un FPGA. Pour pouvoir tester l'interface ADC-FPGA.

L'utilisation du circuit FPGA offre plusieurs avantages ;principalement le nombre de données et la capacité de traitement des données en temps réel.Le traitement des données peut être effectué à l'aide des Blocs de traitement des données numérique (DSP) disponibles dans le circuit FPGA, qui peuvent travailler en parallèle pour traiter une énorme quantité de données. Les données peuvent également être traité en utilisant un puissant noyau souple intégré de Processeurs, dont certains sont disponibles gratuitement sur le marché (Exo-noyau pour routeur actif ;Exo-noyau pour le cloudcomputing). De plus, certains FPGA ont des processeurs codés en dur, comme leprocesseur PowerPC dans la famille Virtex de Xilinx.

Par ailleurs, les données peuvent également être traitées dans un PC via uneinterface PCI implémentée dans le FPGA.

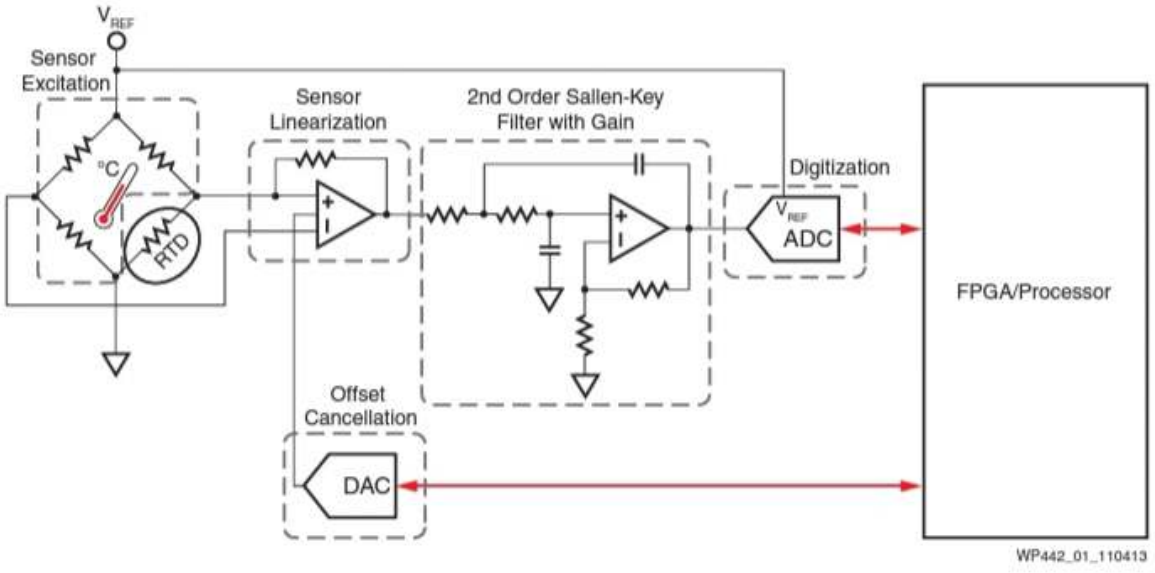


figureII.4.1:Différentes options de traitement des données avec l'implémentation proposée

Cette approche a jeté les bases pour la mise en œuvre des algorithmes DSP complexes dans FPGA ou sont intégrés des processeurs capables de traiter le très haut débit de données sortant de l'ADC, pour des applications telles que télécommunications, systèmes radar / sonar, données à haut débit acquisition, etc. Les techniques de conception FPGA à grande vitesse présentés dans ce travail ont déjà été testés avec succès dans le matériel [9].

II.4.2 Implémentation d'une Fonction analogue de traitement du signal dans Xilinx (Exemple d'un capteur de température) :

Dans cette article[11] les auteurs ont réalisé une interface entre un nœud de capteur capteur de température et un CAN Pour illustrer davantage à quoi ressemble généralement le conditionnement et le traitement du signal analogique, une conception pour mesurer la température à l'aide d'un détecteur de résistance à la température (RTD), illustré dans La **figure II.4.2** a été examinée. Un RTD est un capteur qui change sa résistance en fonction de la températures. Les détecteurs de température à résistance platine (PRTD) sont largement utilisés en raison de leur précision et leur capacité à fonctionner sur une très large plage de températures. Le circuit illustré est typique des conceptions de mesure de température qui utilisent un PRTD. Ce circuit a cinq sous-fonctions: excitation du capteur, linéarisation, annulation d'offset, filtrage et numérisation.



figureII.4.2.1: Circuit de mesure de température PT100.

d'après cette article l'excitation du capteur Un capteur RTD doit être excité par un courant ou une tension pour qu'il produise . L'excitation de tension est généralement la plus simple à mettre en œuvre car les références de tension sont relativement peu coûteux. Les références actuelles sont généralement construites sur le **PCB** en utilisant plusieurscomposants .

Aussi,Le circuit présenté montreun filtre analogique passe-basSallen-Key du second ordre. Le filtre est mis en œuvre à l'aide d'un amplificateur opérationnel et de plusieurs résistances et condensateurs. Ce filtre est utilisé pour minimiser les bruits indésirables et pour éviter les effets secondairelorsque le signal est numérisé. Si les ressources de traitement du signal FPGA sont utilisées avec un **ADC**rapide,précis, plus alorsune solution plus efficace peut être réalisée [11].

Le FPGA capte et traite la température en communiquant avec le capteur de température implémenté dans XADC comme ce suit :

La figure montre le Circuit de pont de Wheatstone pour mesurer le changement de résistance du capteur

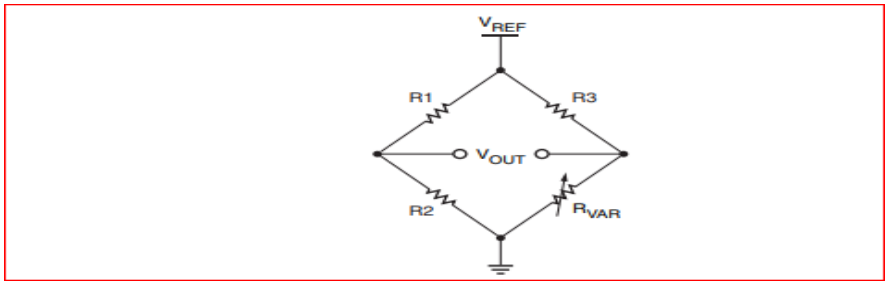


figure II.4.2.2 : pont de Wheatstone

cetterésistance détecteur (RTD),incluent une jauge de contrainte ou un capteur de température. Pour une jauge de contrainte et un RTD, la résistance du capteur change avec le paramètre mesuré. Une méthode typique pour mesurer ce changement de résistance consiste à construire un Circuit en pont de Wheatstone, comme le montre la figure II.4.2.2, où trois des résistances du bridge ont des valeurs fixes et le capteur à une résistance variable forme la quatrième résistance. la tension entre les deux points médians du pont varie proportionnellement au changement de résistance du capteur, devenant ainsi une mesure du paramètre .

Le pont de Wheatstone alimente un ADC pour numériser la tension VOUT. Le pont peut s'interfacer directement à un amplificateur d'instrumentation.. Avec cette configuration, le circuit et l'analyse sont les mêmes que dans le cas d'utilisation. La mesure par capteur nécessite un haut degré de précision. Ainsi, le XADC devrait fonctionner aussi vite que possible pour sur échantillonner ou décimer sa sortie pour maximiser la plage dynamique. Si le ADC utilisé n'a pas assez de bande passante pour piloter le XADC à ces vitesses, un amplificateur tampon est obligatoire .

Une conception du signal de la température captée est effectuer par la suite dans lesles FPGA Xilinx.

II.4. 3 interfaçage des FPGA avec un ADC a la Sortie des données numériques d'unconvertisseur :

Dans cette article [12] Le terme d'interfaçage de réseaux de portes programmables (FPGA) pour la sortie du convertisseur analogique-numérique (ADC) est un défi d'ingénierie. ce titre comprend un aperçu de divers protocoles et normes d'interface .

La tâche est compliquée par le fait que Les ADC utilisent une variété de styles et de normes de données numériques. Le CMOS à débit de données (SDR) est très courant pour les données à faible vitesse interfaces, généralement sous 200 MHz. Dans ce cas, les données sont transmet sur un bord de l'horloge par l'émetteur et reçu par le récepteur sur l'autre front d'horloge. Cela garantit que les données ont suffisamment de temps pour se stabiliser avant d'être échantillonnées par le destinataire.

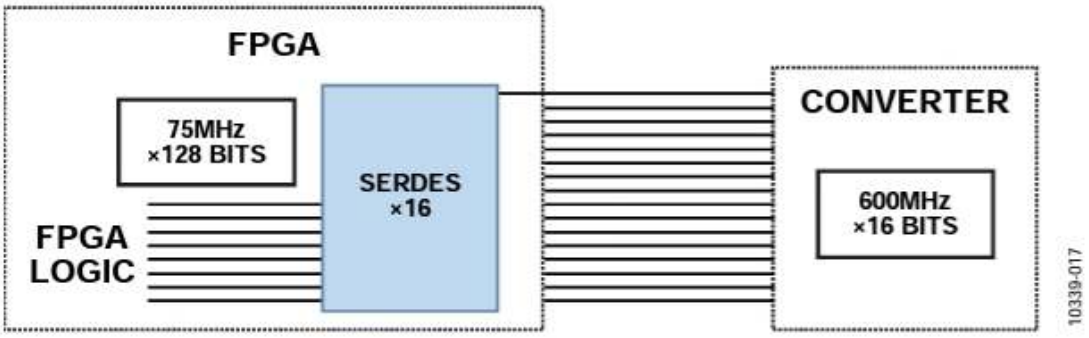


Figure II.4.3.1: Un blocs ES dans l'interface FPGA avec interfaces série haute vitesse sur un convertisseur

*Quelque recommandations :

Certaines recommandations générales sont utiles pour l'interfaçage entre ADC et FPGA.

• ils ont utilisé des terminaisons de résistance externe au niveau du récepteur, du FPGA ou

ASIC, plutôt que les terminaisons FPGA internes, pour éviter les réflexions dues à une discordance qui peuvent briser le timing budget.

• ils exige de ne pas utiliser un DCO d'un ADC si on utilisez plusieurs ADC dans le système.

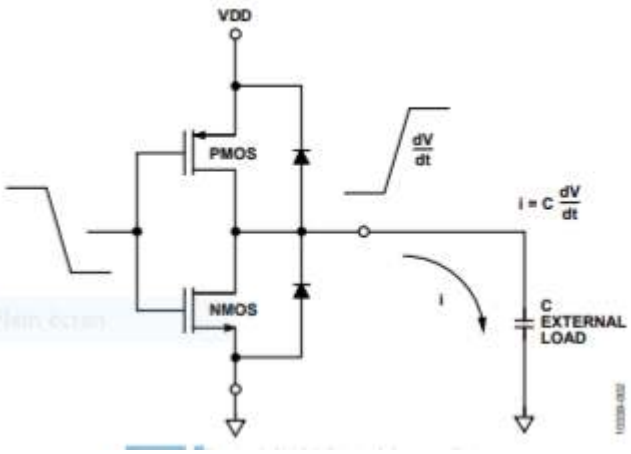


figure II.4.3.1 :LA sortie numérique CMOS typiques

Les sorties numériques ADC doivent être traitées avec précaution car les courants transitoires peuvent augmenter le bruit et la distorsion l'ADC en le couplant à l'entrée analogique. Les pilotes CMOS typiques illustrés à la figure II.3.3.b sont capables de générant d'importants courants transitoires, en particulier lors de la conduite charges capacitives. Un soin particulier doit être apporté avec CMOS ADC de sortie de données afin que ces courants soient minimisés et ne génèrent pas de bruit et de distorsion supplémentaires dans le CAN [11].

II.4.4Conversion analogique-numérique basée sur FPGA via Modulation optimale du cycle :

Dans cet article[13], une architecture de conversion analogique-numérique (ADC) basée sur FPGA via ODCM (modulation de rapport cyclique optimal), est conçue et mise en œuvre à l'aide de plates-formes de co-simulation logicielles et matérielles.(Simulink / Xilinx dans lesquels l'ODCM-ADC est implémenté, et de l'outil de programmation Xilinx ISE pour les puces FPGA).

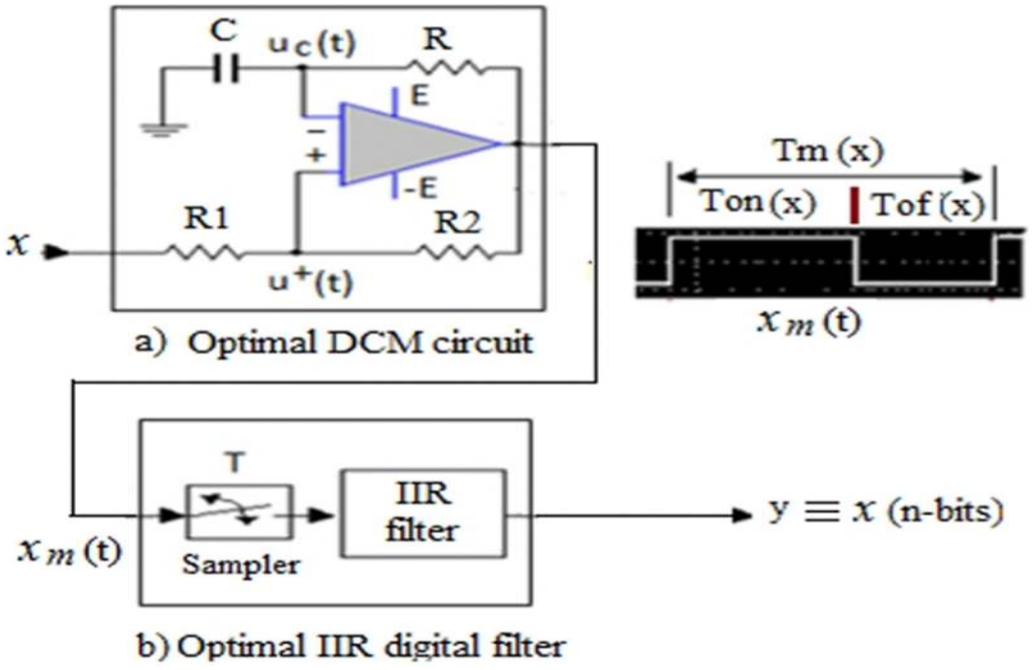


Figure II.4.4. 1 : l'architecture ODCM_ADC.

La conception ODCM-ADC basée sur FPGA est réalisée sous l'espace de travail Simulink, selon le diagramme schématique présenté sur la figureII.4.4. 1. Le sous-système en amont

illustré sur la figure II.4.4. 1 (a) est un modèle basé sur Simulink du circuit DCM donné par (1), avec une entrée de modulation analogique Xs ≡ x comme sur la figure 1 (a), et avec une sortie de modulation échantillonnée xm. Ensuite, le sous-système en aval présenté sur la figure 2II.4.4. 1 (b) est modélisé comme un filtre IIR numérique donné par (2), qui est mis en œuvre en utilisant des ressources de construction visuelles et des panneaux de configuration disponibles dans le cadre du générateur Xilinx System[12].

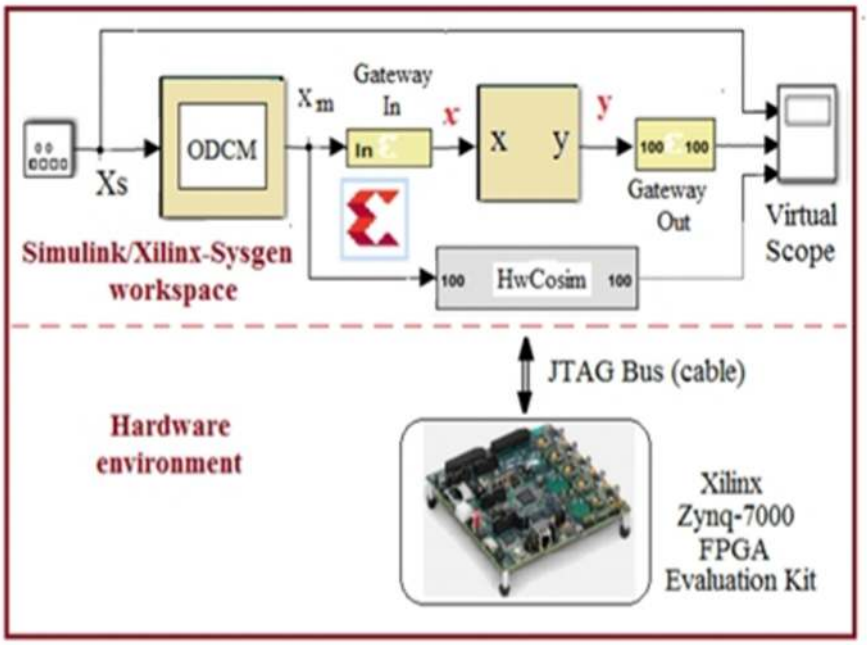


Figure II.4.4. 2: Environnement de co-simulation matérielle

La co-simulation est une technique moderne puissante, largement utilisée de nos jours pour tester et évaluerles performances prévues et réalistes d'un système basé sur DSP rapidement au moment de la conception . En effet, les processus de co-simulation nécessitent une application de simulation virtuelle sur PC pour DSP (traitement du signal numérique). Le PC et la carte DSP sont connectés via un câble / des connecteurs de communication appropriés (USB, Ethernet ou JTAG). Lors d'une session de co-simulation, le simulateur virtuel et le DSP matériel sont simultanément lancés et pilotés dans les mêmes conditions de fonctionnement (entrée et paramètres). L'environnement de co-simulation créé dans ce travail de recherche pour le calcul rapide et l'évaluation des caractéristiques à la fois prédites et expérimentales du prototypage ODCM-ADC basé sur FPGA, est présenté à la figure II.4.4.2

II.5 Conclusion

Dans ce chapitre, nous avons présenté L'état de l'art sur l'implémentation de CAN sur FPGA en s'inspirent des travaux des articles ; d'abord nous avons commencer par citer un aperçois

sur le CAN et particulièrement sur la partie XADC qui nous intéresse le plus dans notre travail.

Dans le prochain chapitre, nous allons présenter l'implémentation matériel de CAN sur FPGA ; en précisons la méthode de garantir la communication entre le monde digitale FPGA et le monde analogique .

c'est-à-dire choisir une approche qui permet l'intégration du CAN dans l'architecture du nœud de capteur implémenté su FPGA .

Chapitre 3

Intégration d'un CAN dans l'architecture d'un nœud de capteur

III.1 Introduction

A travers ce chapitre, nous allons présenter l'intégration d'un CAN dans l'architecture d'un nœud de capteur implémenté sur FPGA. Il est important de signaler que l'architecture de base liée à l'unité de traitement est un système sur puce (SOC) construit autour du processeur Neo430 et de certains composants IPs (intellectuelproperties). Ces derniers sont du domaine publique (opensource/opencores) et leur code VHDL est téléchargeable gratuitement.

Afin de valider la fonctionnalité de notre (SOC), nous avons ciblé deux modes d'utilisation correspondant à deux applications différentes :

- Le premier mode concerne l'utilisation de notre SOC avec un capteur de température permettant de mesurer et d'afficher la température externe de l'environnement

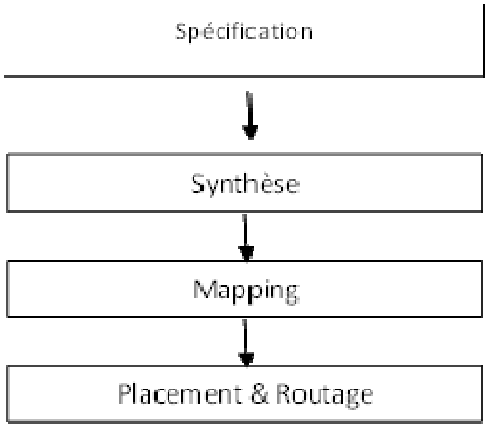
- Le deuxième mode effectué dans notre(SOC) ; concerne l'association d'un ADC avec notre capteur de température pour garantir la conversion analogique numérique et permettre la communication avec n'importe quelle type capteur située a l'intérieur ou l'extérieur de notre carte FPGA.

Nous commencerons d'abord par présenter le synoptique général et l'architecture du système liée à l'implémentation du nœud de capteur sur FPGA. Ensuite, nous présenterons la méthodologie de conception sur FPGA, l'architecture interne détaillée du SOC en question, ainsi que les résultats de synthèse et d'implémentation sur FPGA.

III.2 Méthodologie de conception sur FPGA

La figure III.2 montre le flot de conception adopté l'implémentation hardware du SOC. D'abord, on commence par spécifier la fonction générale du système comme décrit dans la section précédente, la prochaine étape serait de concevoir notre système à travers la

description VHDL et de le relier avec les autres composants (laptop, capteurs et afficheur 7 segments). Cette étape est suivie par l'implémentation sur FPGA selon les étapes suivantes : la synthèse, le mapping, le placement/routage et enfin la configuration et programmation du circuit FPGA .



La figure III.2: Flot de conception

Pour réaliser ce travail ; on a choisie un **CAN** sigma delta de type **TDC** son code est téléchargeable gratuitement(un IP open source).

III. 3 Présentation de l'architecturedécrite du CAN implémenté sur FPGA en utilisant un TDCs

Actuellement, la technique principale utilisée pour implémenter un **CAN** dans un **FPGA** est la modulation « delta sigma ». Bien qu'une résolution élevée puisse être atteinte, de l'ordre de 10 à 15 bits, la fréquence d'échantillonnage est de l'ordre de quelques dizaines de kHz. Comme la modulation delta sigma dépend d'une boucle de rétroaction, elle est intrinsèquement plus lente qu'une méthode sans rétroaction.

Depuis 2009, le taux d'échantillonnage des FPGA TDC était déjà de l'ordre de centaines de MHz. Ainsi, si le signal analogique peut être converti en une variable du domaine temporel, il peut être mesuré avec ces TDC haute vitesse.

Nous montrons ici une de ces techniques pour créer un convertisseur analogique-numérique haute vitesse à partir d'un TDC, que nous appelons un convertisseur analogique-numérique.

III.3.1 De l'idée a la pratique

'ADC 200 MS / s (conversion à la vitesse d'horloge) est fait en convertissant le signal analogique en une variable de domaine temporel par un schéma de comparaison de pente. L'horloge de 200 MHz du FPGA génère une pente à laquelle l'entrée analogique est comparée. Ceci est mis en œuvre à l'aide d'un tampon d'entrée LVDS faisant office de comparateur. Le temps pendant lequel le LVDS génère un 1 en sortie est en corrélation directe avec la tension analogique. Par conséquent, la simple mesure de ce temps dans un TDC donne une mesure de la tension analogique. Le chronogramme est illustré dans la figure suivante.

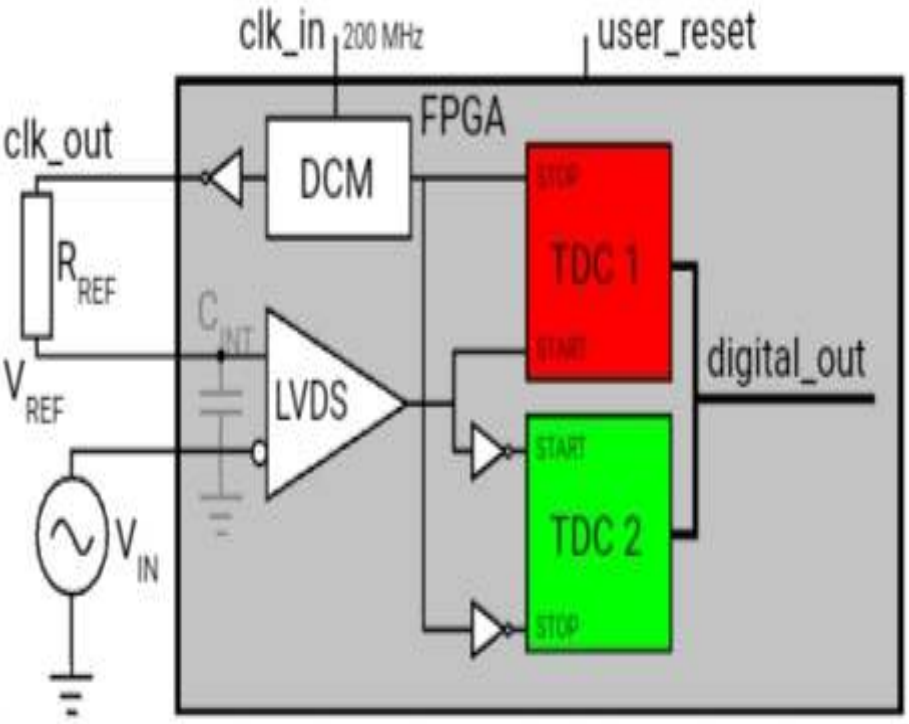


Figure III.3.1 :schéma ADC de niveau supérieur avec les entrées et sorties FPGA

La figureIII.3.1représente le système implémenté avec deux TDC de base comme décrit précédemment dans les TDC FPGA de base .

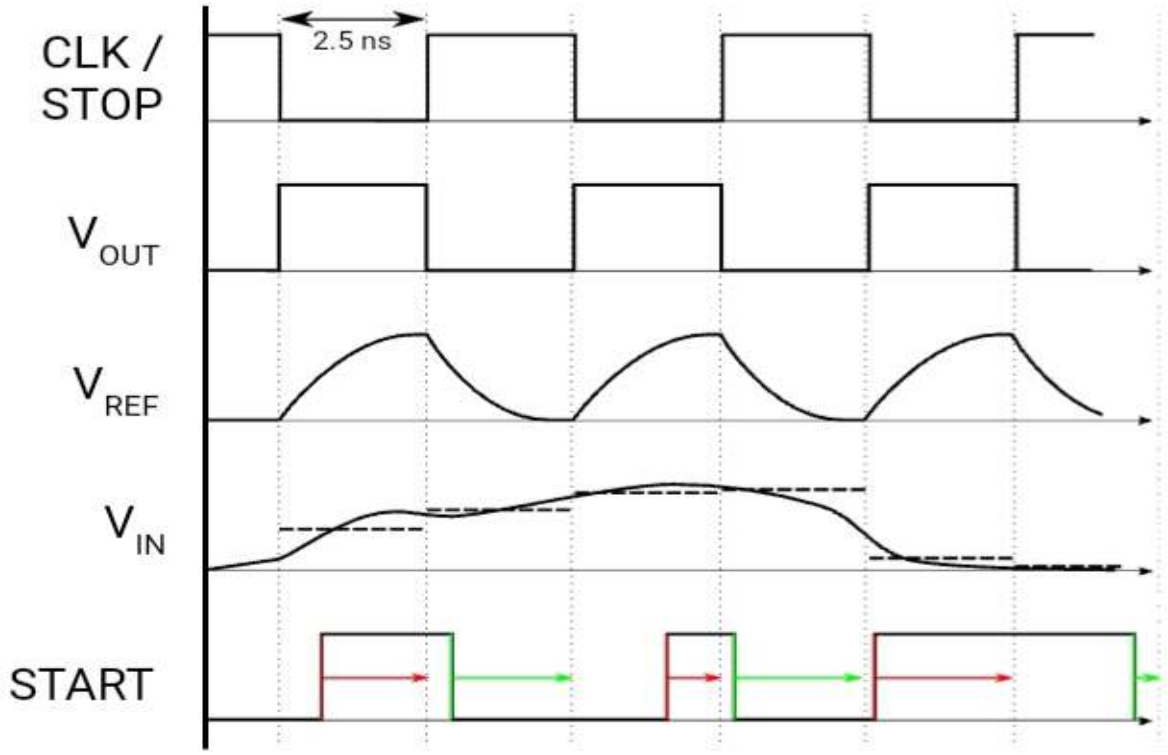


Figure III.3.2 : Chronogramme de comparaison de pente du CAN

De plus, une résistance externe **RREF** est nécessaire pour générer régler la tension de référence. Le système a trois entrées (le signal analogique d'entrée **VIN**, la tension de référence générée par le circuit **RC VREF**, et un signal d'horloge **CLK** fonctionnant à 200 MHz) et deux sorties (la conversion numérique de l'entrée mettre le signal analogique (DOUT) et la tension de sortie (VOUT) utilisé pour générer la tension de référence pour le COM LV DS). Les formes d'onde représentant le chronogramme de l'ensemble des systèmes sont représentés sur la figure III.3.2 ,les commandes de signal d'horloge et le fonctionnement global du circuit. Il est utilisé comme signal d'arrêt pour TDC 1 etinversé comme signal d'arrêt pour le TDC inversé, il pilote le signal de sortie VOUT connecté au réseau RC, générant un semi-exponentiel rampe[14].

la rampe de référence VREF devient supérieure à la tension d'entrée VIN. L'intervalle de temps entre une transition «0» → «1» causé par le LV DS est le prochain front montant de l'horloge.

III.3.2 phase implémentation :

la conception est dépend de la structure de tranche FPGA. Bien qu'il soit possible de mettre en œuvre des versions reconfigurables d'oscillateurs TDC et de compteurs de temps, leur

utilisation est limitée en raison de la faible résolution ou des exigences d'étalonnage exigeantes.

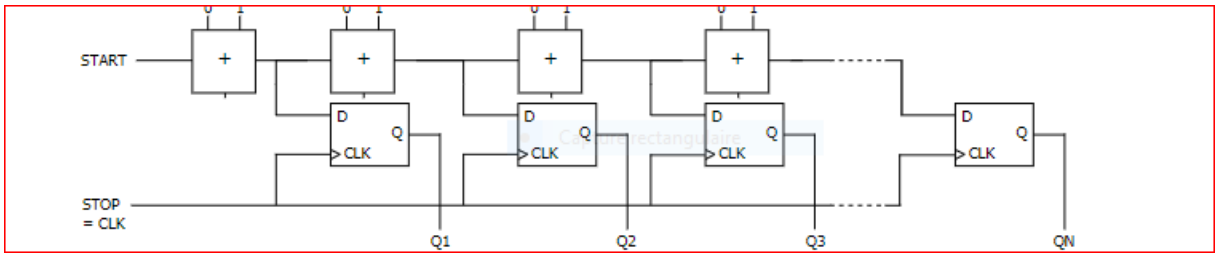


Figure III.3.2.a : exemple d'un TDC basé sur une ligne à retardimplémenté dans un FPGA

Par la suite ;Une représentation de bloc plus détaillée de notre ADC est représentée sur la figure **III.3.2.b**. Sur la figure, nous pouvons voir les nacelles, les bascules, le décodeur de thermomètre et la lecture. Dans à chaque période d'horloge, le LV DS génère un «1». Ce signal est retardé à travers la carry Chain, implémenté à l'aide de Carry4 blocs. La valeur comptable de chaque bloc Carry4 est verrouillée le prochain front d'horloge. L'horodatage mesuré avec ce le thermomètre est converti en un horodatage binaire. le décodeur pour le thermomètre est un décodeur implémenté en utilisant multiplexeurs qui choisissent de transférer le haut ou le bas partie du thermomètre en fonction de la valeur stockée dans la partie centrale du thermomètre, comme cela sera détaillé dans le reste de notre section.

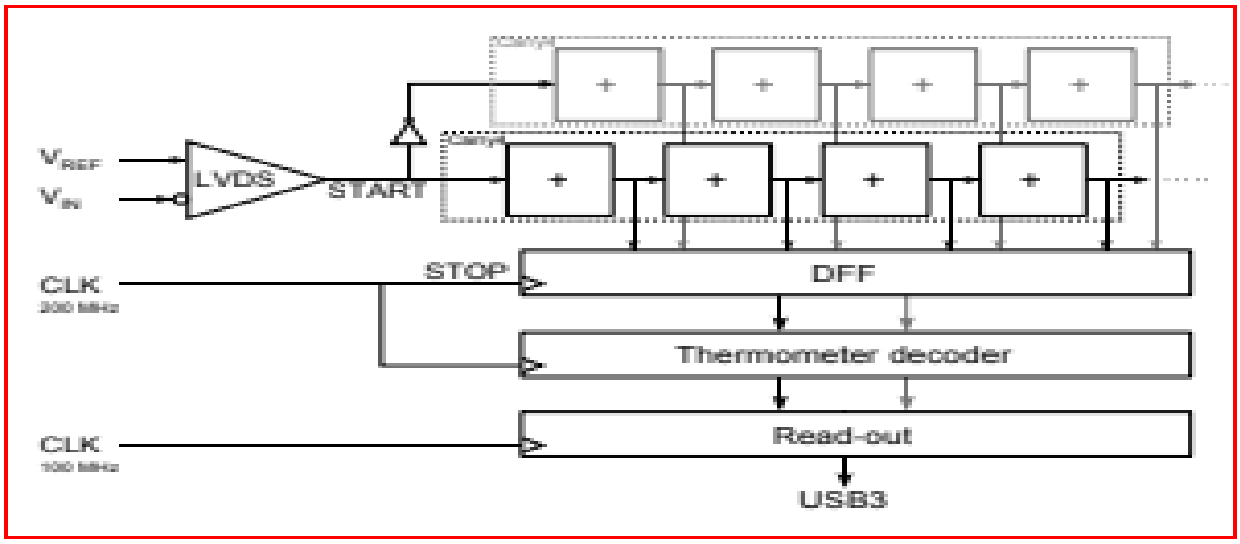


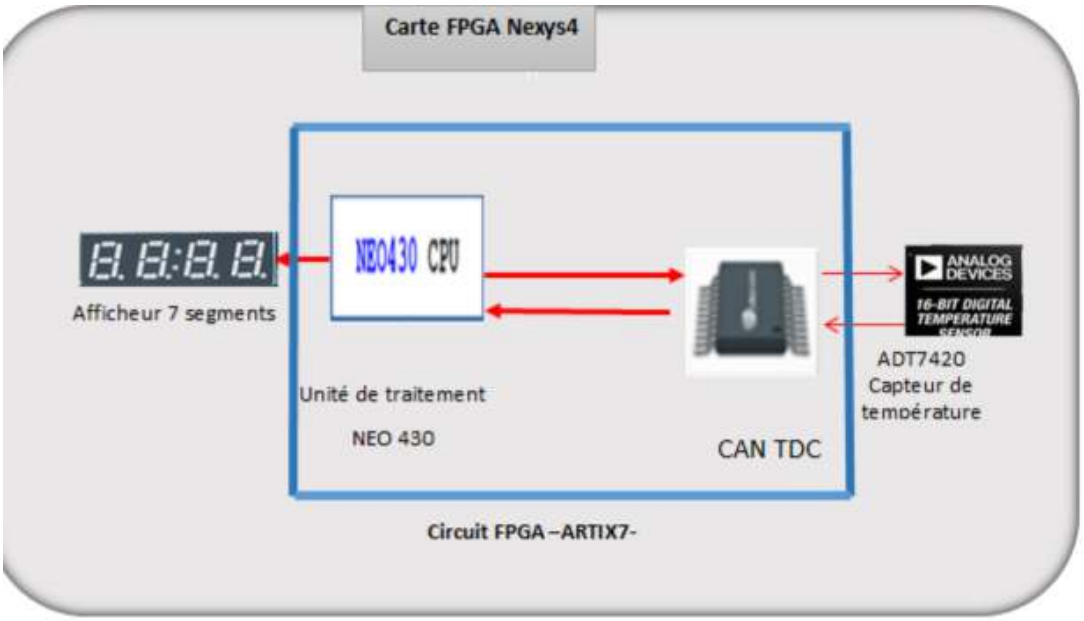
Figure III.3.2.b :Schéma fonctionnel plus détaillé de notre TDC

C'est possible de voir le thermomètre implémenté avec Carry4 éléments, le LVDS, les bascules, le décodeur de thermomètre, et l'interface pour la lecture

III.4Synoptique général du nœud de capteur sur FPGA

La figure III.4 représente le synoptique général de l'implémentation du nœud de capteur sur FPGALa partie digitale correspondant à l'unité de traitement de notre nœud de capteur, est un SOC implémentée au sein du circuit FPGA Artix7 de Xilinx. Le circuit FPGA communique avec deux composant : un capteur de température ADT7420 pour acquérir la température de l'environnement extérieur et un convertisseur analogique numérique pour convertir le signal capte en un langage digitale que notre FPGA pourra comprendre .

Un afficheur 7 segments est utilisé pour afficher la valeur de la température captée .Le circuit FPGA Artix7, le capteur de température ADT 7420, l'ADC et l'afficheur 7 segments sont tous des composants implantés dans la carte de développement Nexys4 DDR de Digilent.



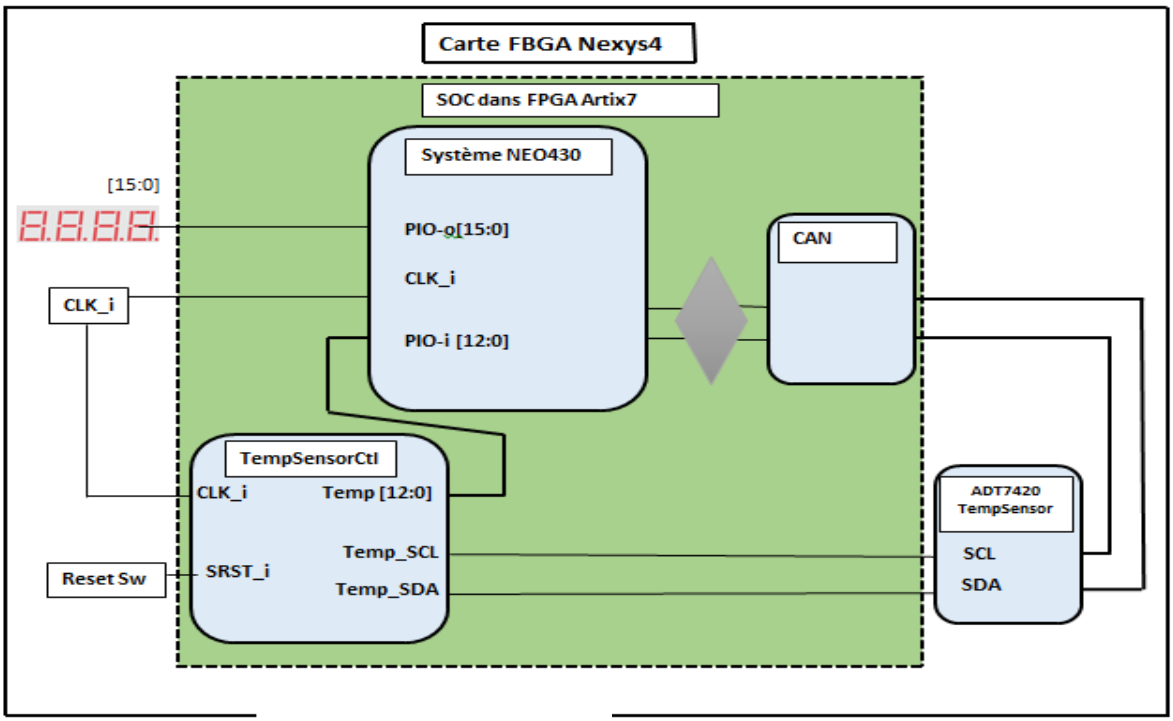
La figure III.4 : Synoptique général de l'implémentation du CAN sur FPGA

.La communication entre le lap top (ou bien le PC) et le circuit FPGA permet d'une part la configuration et la programmation de la carte FPGA et d'autre part, l'affichage, la visualisation graphique et l'exploitation des mesures en temps réel de la température .

III.5 l'architecture d'interne détaillée du système

La figure III.4 représente l'architecture interne détaillée du système, constitué principalement du système NEO430 et de Convertisseur analogique numérique (ADC) .Cette dernière assure la communication entre le capteur de température ADT7420 et le système NEO430. La communication avec l'afficheur 7 segments est assurée moyennant la sortie PIO. Et la communication avec le PC est assurée moyennant l'UART (UniversalassynchronousReceivertransmetter)

Le système à base du processeur NEO430 et convertisseur analogique numérique (ADC) forment un système sur puce SOC. L'outil Vivado de Xilinx est utilisé pour la synthèse et l'implémentation de ce système sur le circuit FPGA Artix-7 de Xilinx hébergé sur la carte Nexys 4 DDR de Diligent.



FigureIII.5: Présentation de l'architecture hardware du système implémenté sur FPGA

*Dans la suite de cette section, nous présenterons les différents éléments du système :

III.6 Le Système NEO430

Le système à processeur NEO430 est constitué de plusieurs différents modules comme le montre la figure III.6.1

Les modules présentés avec des lignes pointillées sont optionnelles et peuvent être désactivées individuellement à travers des constants dans un fichier VHDL du type package inclus dans le design, par exemple si on veut exclure l'unité MAC du design, on met 'false' à la place de 'true' (voir capture ci-dessous) [15].

Par la suite ;Nous parlerons des éléments qui nous avons besoin dans notre travail :

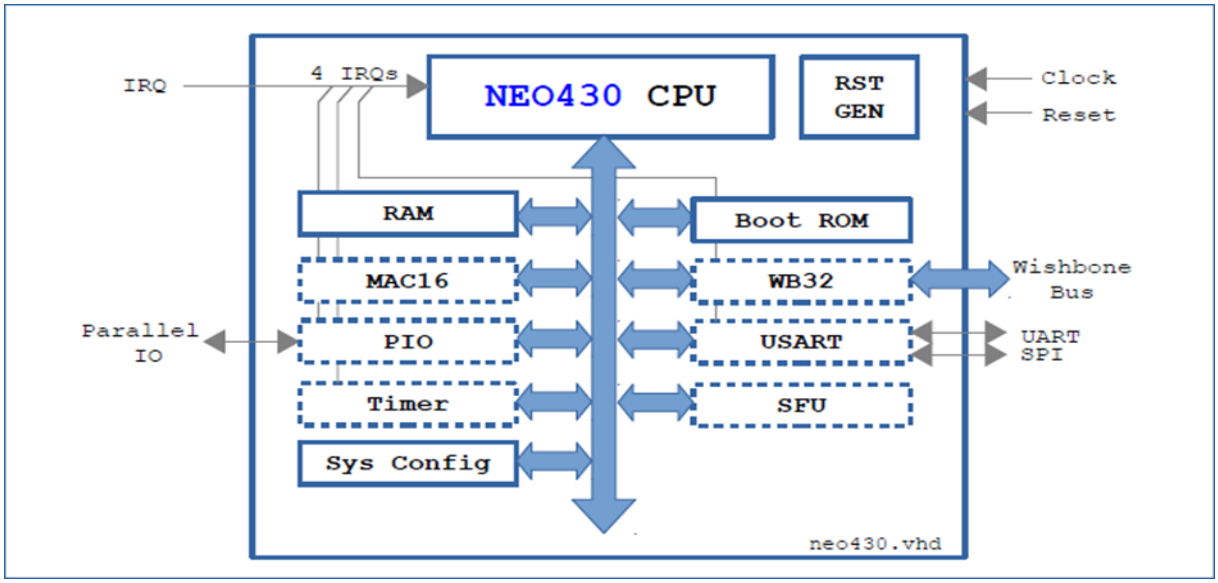


Figure III.6.1 : Le système NEO430

- Neo430 CPU : Le NEO430 est un processeur soft open source 16-bit RISC décrit en VHDL qui utilise une seule mémoire pour les données et les instructions (Architecture Va Neumann) et qui est aussi 100% compatible avec le jeu d'instruction du fameux MSP430 de Texas Instrument. C'est le cœur du système sur puce, il supporte tous le jeu d'instruction et les modes d'adressages du MSP430. Il inclut une unité de contrôle, une unité arithmétique et logique (ALU), un générateur d'adresse et un groupe de registres .

- Mémoire interne RAM (Component VHDL memory.vhd) : Elle sert comme une mémoire commune pour les données et les instructions du programme exécuté. La taille de cette mémoire peut être configurée en utilisant le fichier package comme le montre la capture ci-dessous. La taille maximale autorisée est de 60KB.

Etant donné que cette mémoire est volatile, elle ne peut pas être utilisée pour garder le programme d'application de façon permanente, par conséquent on utilise le bootloader pour copier une image du programme à partir d'une ROM ajoutée au processeur comme on va voir par la suite.

- Boot ROM (Component VHDL boot_rom.vhd): Comme son nom déjà indique, le boot Rom contient l'image du boot loader en lecture seul qui est exécuté juste après leredémarrage du système. La taille de cette mémoire est aussi configurable à travers le fichier package (Capture ci-dessous)

- Unité multiplicateur-accumulateur (MAC 16) (Component VHDL mac16.vhd)) : Unité optionnelle qui performe des opérations de multiplications suivies par additions sur des donnés de 16 bits.

- Ports d'entrés sorties parallèle (PIO) (Component VHDL parallel_io.vhd):

Unité qui offre un port d'entré de 16 bits et un port de sortie de 16 bits pour communiquer avec le monde extérieur. Dans notre système les lignes d'entrés de ce module sont utilisées pour acquérir l'information de la température et lignes de sorties pour animer les afficheurs 7 segments. Cette unité peut être exclus durant la synthèse du système en utilisant le fichier package

III.7 Capteur de température ADT7420 :

L'ADT7420 hébergé sur la carte Nexys4 DDR est un capteur de température numérique de haute précision offrant des performances de pointe sur une large gamme industrielle, logés dans un boîtier de 4mm×4mm. Il contient un capteur de température.

Le contrôle de l'ADT7420 s'effectue via ADC (Convertisseur Analogique Numérique) dans la figure III.4 CAN de 16 bits pour surveiller et numériser la température à une résolution de 0,0078 ° C. La résolution du CAN, par défaut, est réglée sur 13 bits (0.0625 °C) et c'est la résolution utilisée pour notre implémentation. L'ADT7420 est connecté à ce bus en esclave et est sous le contrôle d'un appareil maître.

Le schéma de la figure III.7représente les différentes connections d'interface conseillé par le constructeur et implémenté sur la carte Nexys4[16].

Comme toutes les stations compatibles I2C, l'ADT7420 utilise 7 bits d'adresse, les 5 MSBs de cette adresse sont liés physiquement à 10010 à l'intérieur du capteur. Les pins A0 et A1 représentent les 2 LSBs de cette adresse ce qui nous donne le choix entre 4 adresses possibles. Sur la carte Nexys4, ces deux pins sont liés au Vcc ce qui veut dire que l'adresse du capteur est fixée à 1001011.

Les pins SCL et SDA représentent le serial clock et serial data respectivement de la communication I2C. Le pin CT (Critical Température) s'active quand une température critique configurée par l'utilisateur est atteinte et le pin d'interruption INT s'active quand la température dépasse un seuil maximum prédéfini ou chute en dessous un seuil minimum prédéfini.

Les pins A0, A1, SCL, SDA, CT et INT sont du type open-drain, donc des résistances de pull-up sont nécessaires pour leurs fonctionnements comme le montre la figure III.7

L'ADT7420 contient plusieurs registres qui permettent la configuration du capteur, la lecture de température, la spécification des seuils de température pour générer des interruptions et des alarmes. Chacun de ces registres est accessible avec une adresse unique[17].

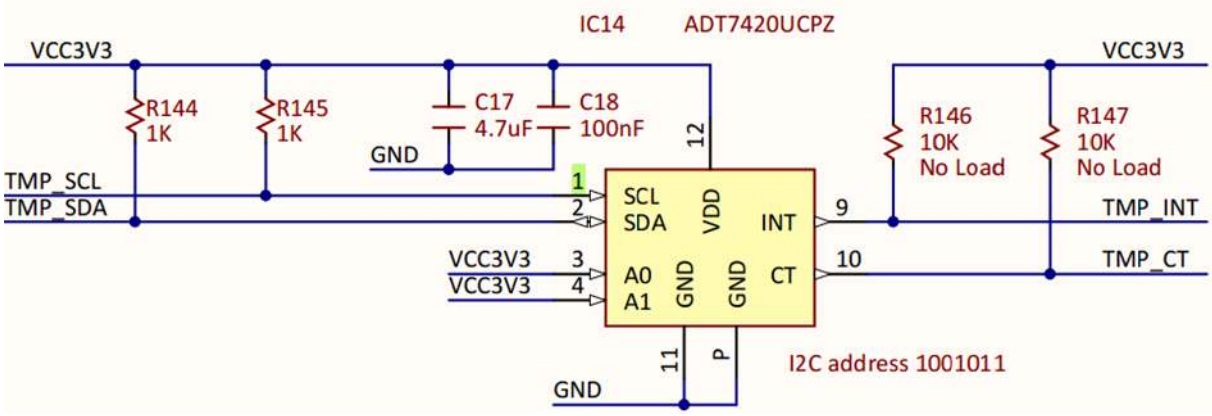


Figure III.7 : Connexion d'interface typique de l'ADT7420 implémentée sur Nexys 4

III.8 Conclusion

Dans ce chapitre,nous avons présenté l'architecture du CAN ainsi notre choix de convertisseur ; on a opté pour une approche de travailler avec un CAN-TDC son code VHDL (IP source) téléchargeable .

L'architecture du système présenté est composée de deux sous- systèmes :un système NEO430 implémenté avec un ADC-TDC sur une carte FPGA Artix7 de XiLinX et un autre système située dans la carte FPFA ,il s'agit d'un capteur de température ADT7420 .

Dans le prochain chapitre ; nous allons voir l'implémentation ainsi le système dans le circuit FPGA et les résultats de synthèse et d'implémentation .

Chapitre 4

Résultats et synthèse de l'implémentation du CAN sur FPGA

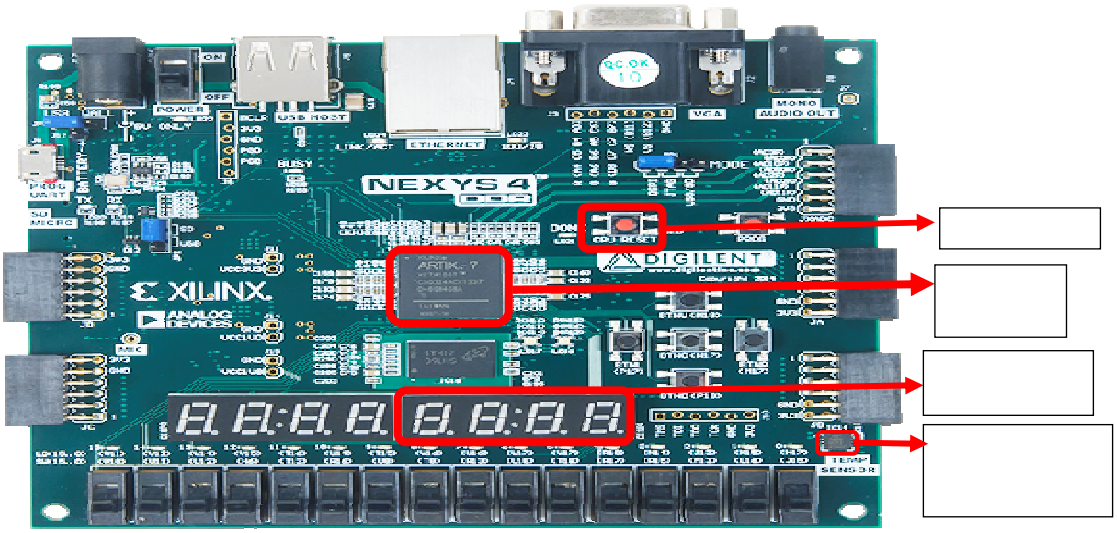
IV.1 Introduction

Dans ce chapitre, nous allons présenter l'architecture de notre système implémenté sur FPGA En revanche il faut communique le CAN et FPGA ;pour cela on a proposé d'intégré un ADC sigma delta de type TDC et l'interfacer avec cette dernière. Nous allons aussi présenter l'organigramme de notre programme avec la synthèse et les résultats d'implémentation .une étape avance par la suite et de faire la simulation et voir l'échantillonnage du signal de sortie si il saura valider .

IV.2 La Plateforme Nexys™4

La carte Nexys™4 de Digilent est utilisée pour valider l'implémentation de notre implémentation. Le Nexys 4, figureIV.2 est une plateforme de développement de circuit numérique prêt-à-utiliser, complète basée sur le tout dernier réseau de portes programmables (FPGA) Artix-7T de Xilinx. Avec son FPGA large et de grande capacité, des mémoires externes généreuses, et un grand nombre de ports USB, Ethernet, et autres, la Nexys4 peut accueillir des modèles allant des circuits combinatoires préliminaires à de puissants processeurs embarqués. Plusieurs périphériques intégrés dont un accéléromètre, un capteur de température, un microphone numérique MEMs, un amplificateur de haut-parleur et de nombreux périphériques d'E/S permettent au Nexys4 d'être utilisé pour une large gamme de modèles, sans avoir besoin d'autres composants [18].

La carte Nexys4 est compatible avec Xilinx Vivado ® Design Suite, ainsi que la boîte à outils ISE.



FigureIV.2 : Plateforme de développement Nexys 4

FPGA Artix-7 :

Le FPGA Artix-7 est utilisé pour l'implantation de notre système sur puce. Ce circuit est optimisé pour la logique de haute performance, et offre plus de capacité, plus de performances et plus de ressources que les modèles précédents. Les caractéristiques du Artix-7 100T incluent [19] :

- 15 850 slices (tranches) comprenant chacune quatre LUT (table de correspondance) à 6 entrées et 8 bascules
- Bloc RAM rapide de 4 860 kbits
- Six dalles de gestion d'horloge, chacune avec une boucle à verrouillage de phase (PLL)
- 240 tranches DSP
- Des vitesses d'horloge interne supérieures à 450 MHz
- Convertisseur analogique/numérique sur puce (XADC)

IV.3Implémentations du système sur puce :

Dans le chapitre, nous avons présenté les étapes de la méthodologie de conception sur fpga en utilisant l'outil vivado .

Le logiciel Xilinx Vivado 2015.1 est un outil de conception de circuit pour la famille 7 et la famille Ultrascale des FPGAs de Xilinx. Ce logiciel permet essentiellement d'effectuer les différentes étapes propres à la synthèse de circuits numériques sur FPGA.

*Utilisant l'outil ViVADO, l'implémentation de notre système sur puce se fait en plusieurs étapes :

1. **Création de projet** : un projet permet de regrouper plusieurs fichiers sources, dans le projet crée et comme première étape on ajoute les différents fichiers source de type VHDL des différents modules (IPs) du système basé sur le processeur NEO430 décrit dans la section 3.1 en s'assurant que le fichier NEO430_top.VHD est en tête d'hiérarchie des fichiers de l'instance du système de processeur et on ajoute aussi le fichier VHDL de l'interface série TempSensorCTL comme deuxième instance. La figure montre la hiérarchie des différents fichiers source décrivant notre système sur puce. Après la création des deux instances, il est temps maintenant de les relier à travers un fichier VHDL global (NEO430TEMP) qui sera notre TOP LEVEL ENTITY utilisant la technique du PORT MAP.

2. **Affectation des Pins** : Le FPGA sur lequel nous allons implanter notre système sur puce comporte un grand nombre de pins d'entrée/sortie, cette étape consiste à crée un fichier du type XDC (Xilinx Design Constraint) qui sert à relier les différentes entrées/sorties de notre système montrées dans la figure IV.4.1 à des pins spécifiques du FPGA qui sont eux même relier à des dispositifs spécifiques de la carte de développement Nexys4.

3. **Synthèse du système** : La synthèse consiste à traduire la description de notre système décrit en VHDL en blocs et composantes disponibles sur le FPGA choisi comme les LUTs, les bascules, les blocs RAMs, etc.

4. **Implémentation du système** : cette étape est divisée en trois sous étapes :

*__Transformation (mapping)__ : consiste à regrouper les composantes obtenus lors de la synthèse dans des blocs spécifiques du FPGA.

*__Disposition (placement)__ : choisir des endroits spécifiques sur le FPGA où disposer les blocs utilisés, et choisir les pins du FPGA correspondant aux ports d'entrée et de sortie*Routage (routing) : consiste à établir des connexions électriques entre les blocs utilisés.

Dans la page suivante un Aperçue de la plateforme de Conception FPGA XILINX: VIVADO 14.2 qui est présenté :

.

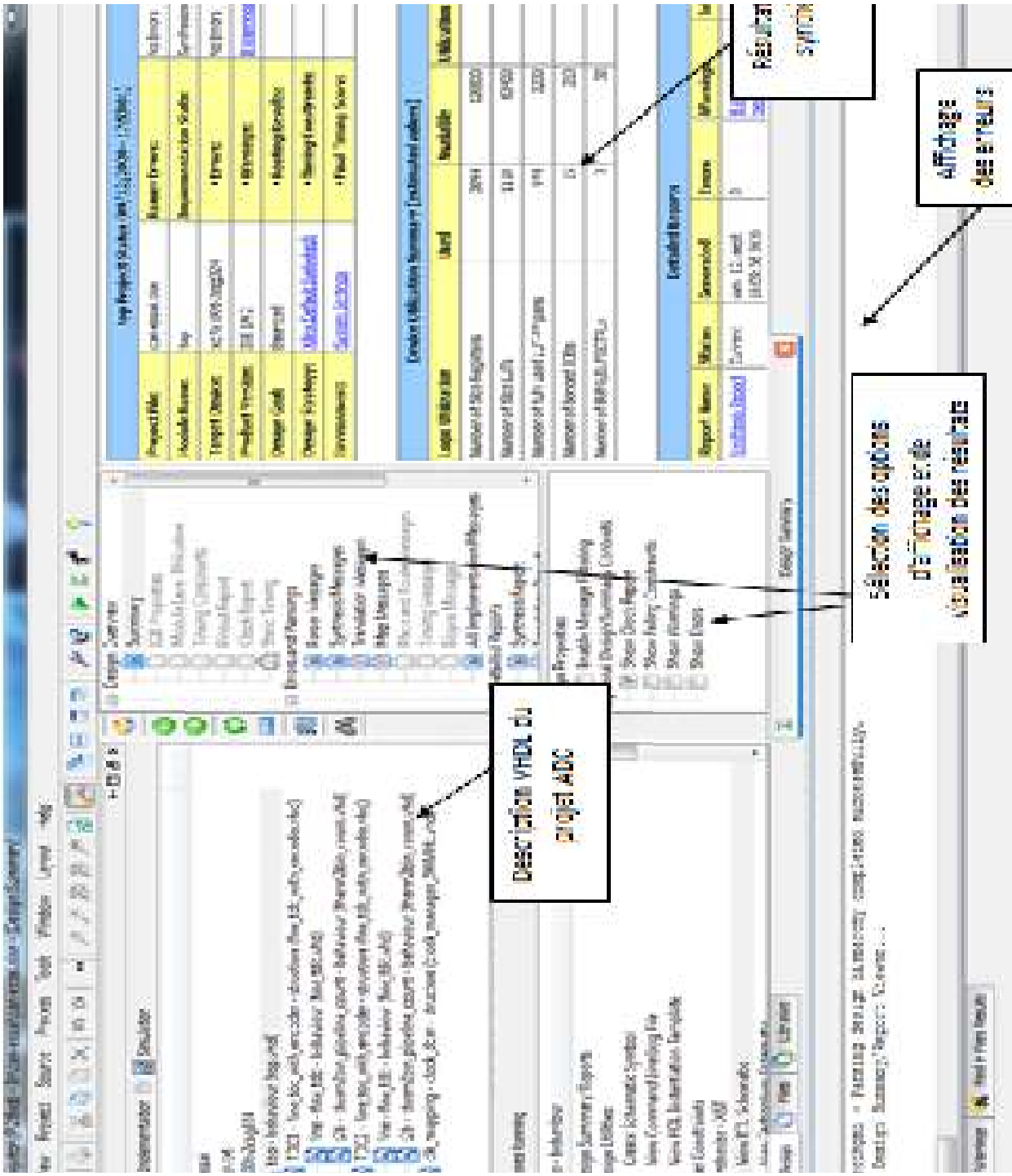


Figure IV.3.1 : Aperçue de la plateforme de Conception FPGA XILINX: VIVADO 14.2

IV.4Résultats de synthèsedu TDC:

Une fois le projet crée et compilé al'outil de conception VIVADO,on compile la description VHDL que nous avons crée et la transforme sous forme de portes et circuits logiques de base

qui sont contenues dans le circuit FPGA, que nous avons choisi pour notre travail, à savoir le circuit: Artix-7 XCA-100T-2csg 324. Le tableau ci-dessous , présente les résultats de synthèse du convertisseur analogique numérique TDC .

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	3094	126800	2%
Number of Slice LUTs	1110	63400	1%
Number of fully used LUT-FF pairs	974	3230	30%
Number of bonded IOBs	15	210	7%
Number of BUFG/BUFGCTRLs	3	32	9%

Tableau IV.4 : Utilisation des ressources du FPGA après l'implémentation TDC

*Ces résultats montrent que le CAN TDC consomme 2% de registres; 1µ de mémoire LUT, 30% de LUT a base de Filp Flop (FF), 7 % d'entrée sortie et 9% de Buffer G -BUFG) du circuit FPGA artix 7 famille XCA-100T-2 csg 324.

***Résultats RTL (Register Transfer Level)**

Par ailleur, l'outil VIVAO, nous permet de visualiser le Schéma RTL (c.a.d : description au niveau registre), du circuit ADC-TDC.

La figure **IV.4.1**montre la description du circuits ADC au niveau TOP. Ce schéma montre que les entrées /sorties sont conforment à ceux décrits en VHDL, à savoir:

```
ENTITY top IS

PORT (user_reset      : IN std_logic;
      clk_in          : IN std_logic;   -- input 200 MHz clock
      clk_out         : OUT std_logic;          -- clock to generate ra
      V_IN            : IN std_logic;   -- the analog input signal
      V_REF           : IN std_logic;   -- the reference input signal (ramp)
      digital_out     : OUT std_logic_vector(FINE_BITS DOWNTO 0));
END top;
```

La figure IV.4.1 :description du circuits ADC au niveau TOP

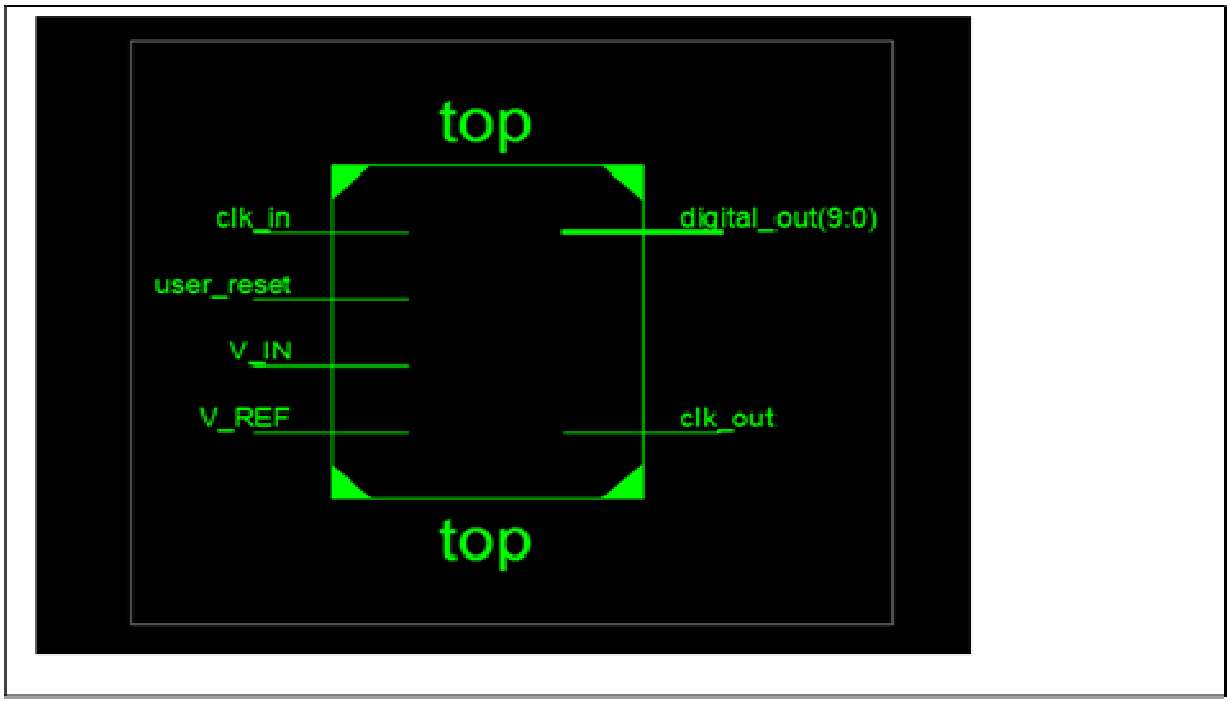
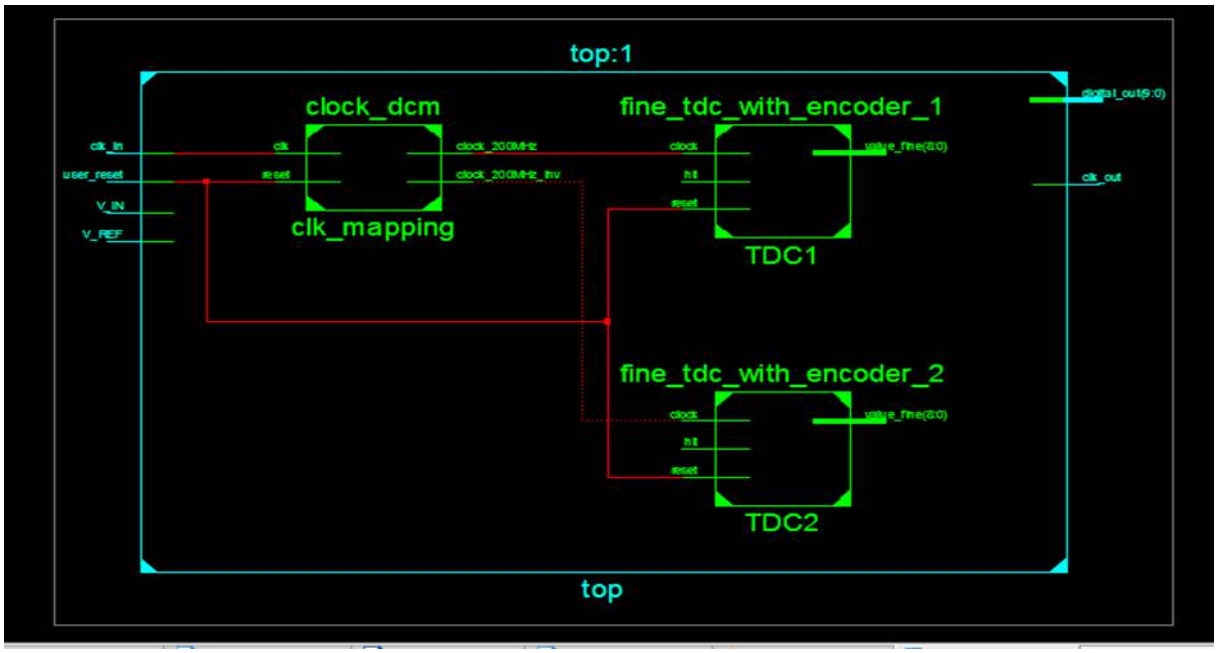


Figure IV.4.2 : Description RTL du CAN-TDC

La figure IV.4.2 montre la génération du schéma du CAN_TDC complet. ce résultats montre lagénération de l'architecture interne CAN-TDC constitué duClk manager (clk_dcm), du TDC1et TDC2.



FigureIV.4.3 :Génération de la description schématique de l'architecture interne du CAN-TDC

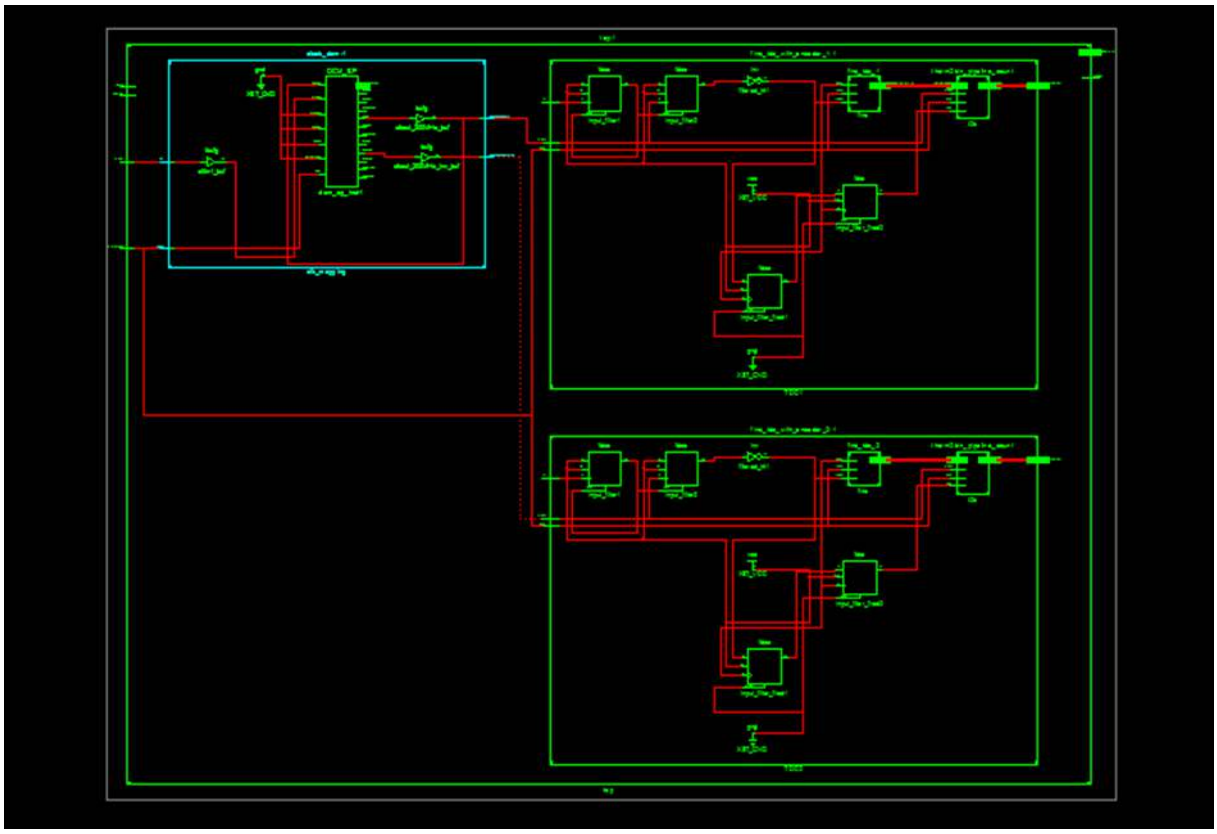


Figure IV.4.4 : Visualisation de la description interne détaillée de l'architecture du ADC-TDC

TDC1 et TDC2 doit connecté entre eux ;ils ont un comparateur qui définie leur architecture on remarque aussi des composant interne de Xlinix pour optimiser le design.

IV.5 résultats de synthèse du NEO 430 :

De même comme le titreIV.4 ;on prend les résultats de synthèse d'un travail déjà fait du NE0 430[18] :

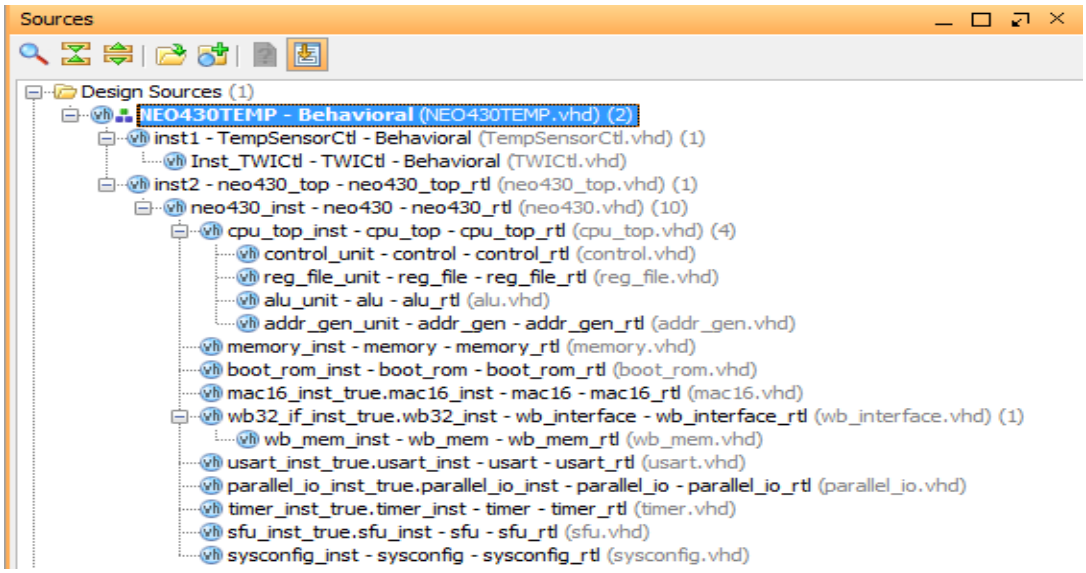


Figure IV.5.1 : Hiérarchie des différents fichiers source VHDL

NEO430TEMP.vhd

D:/Xilinks/NEO430TEMP_WbRom/NEO430TEMP_WbRom.srcs/sources_1/new/NEO430TEMP.vhd

```
 98 begin
 99
100 inst1: TempSensorCtl
101        port map(
102                 TMP_SCL => tmp_scl,
103                 TMP_SDA => tmp_sda,
104                 TEMP_O(12 downto 0) => s(12 downto 0),
105                 RDY_O => rdy_o,
106                 ERR_O => err_o,
107                 CLK_I => clk_i,
108                 SRST_I => srst_i);
109
110 inst2: neo430_top
111        port map(
112                 clk_i => clk_i,
113                 rst_i => rst_i,
114                 pio_o (15 downto 0) => pio_o (15 downto 0),
115                 pio_i (12 downto 0) => s (12 downto 0),
116                 pio_i (15 downto 13) => pio_i (15 downto 13),
117                 uart_txd_o => uart_txd_o,
118                 uart_rxd_i => uart_rxd_i,
119                 spi_sclk_o => spi_sclk_o,
120                 spi_mosi_o => spi_mosi_o,
121                 spi_miso_i => spi_miso_i,
122                 spi_cs_o (5 downto 0) => spi_cs_o(5 downto 0));
123
124 end Behavioral;
```

Figure IV.5.2 : Partie du code VHDL reliant l'interface série et le système NEO430

Dans notre système on affecte CLK_i du processeur NEO430 et de l'interface série TempSensorCtl au clock de la carte qui est de 100MHZ, le RST_i du processeur NEO430 , le SRST_i de l'interface série à un switch, les lignes de communication I2C Temp_SCL et Temp_SDA de l'interface série aux SCL et SDA du capteur de température.

Dans la suite ; des tableaux qui montrent les résultats de synthèses du NEO 430 :

Resource	Utilisation	Disponible	Pourcentage %
Bascule (Flip Flop)	759	126800	0.60
Table de vérité (LUT)	1135	63400	1.79
Mémoire LUT	33	19000	0.17
E/S (I/O)	37	210	17.62
Blocs mémoire (BRAM)	4.5	135	3.33
DSP	1	340	0.42
Global Buffer (BUFG)	1	32	3.12

Table IV.5.1 : Utilisation des ressources du FPGA après l'implémentation

Le tableauIV. 5.1 résume l'utilisation des ressources du FPFA après l'implémentation, on constate que notre système ne consomme pas beaucoup de ressource du circuit FPGA Artix 7 ce qui laisse l'espace pour élargir notre système et rajouter d'autres fonctionnalités.

Type de consommation	Resource	Consommation (mW)
Dynamique	Horologes (Clocks)	5
	Signaux	8
	Logique	6
	Bloques RAMs (BRAM)	3
	DSP	1
	E/S (I/O)	1
	TotaleDynamique	24
Statique	TotaleStatique	97
	Totale Circuit	121

Table IV.5.2 : Consommation d'énergie après implémentation

Le tableau IV.5.2 montre la consommation énergétique de notre système, on constate que sur la totalité de 121 mW seulement 20% est consommée par le circuit implémenté, le reste (97 mW) représente la consommation statique du circuit FPGA Artix7. La consommation totale reste un peu plus élevée en comparaison avec les nœuds de capteurs commerciaux ,mais le circuit FPGA peut compenser ça en offrant plus de flexibilité et une capacité de traitement supérieure en cas d'extension du présent système.

IV.6 résultats d'implémentation du TDC

par la suite une autre visualisation avec une version VIVADO 2014 qui montre un schéma détaillée du TDC sur FPGA

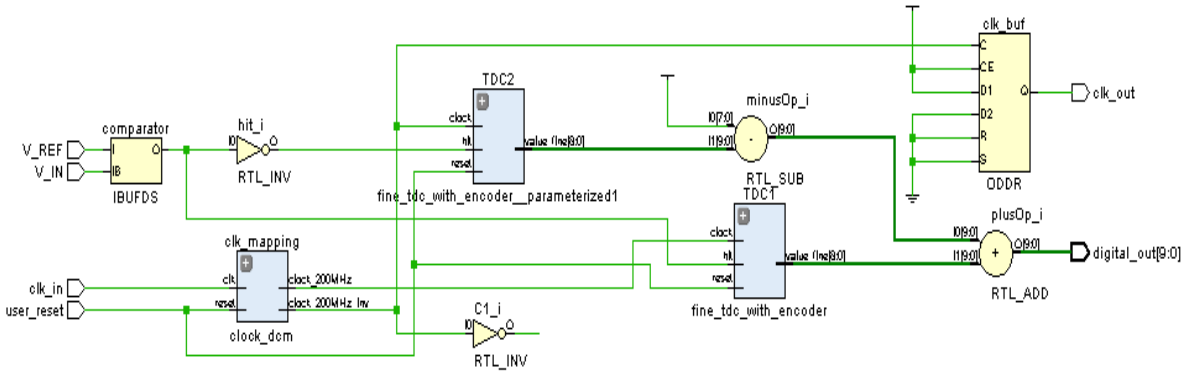


Figure IV.6.1 : visualisation schématique détaillé du TDC

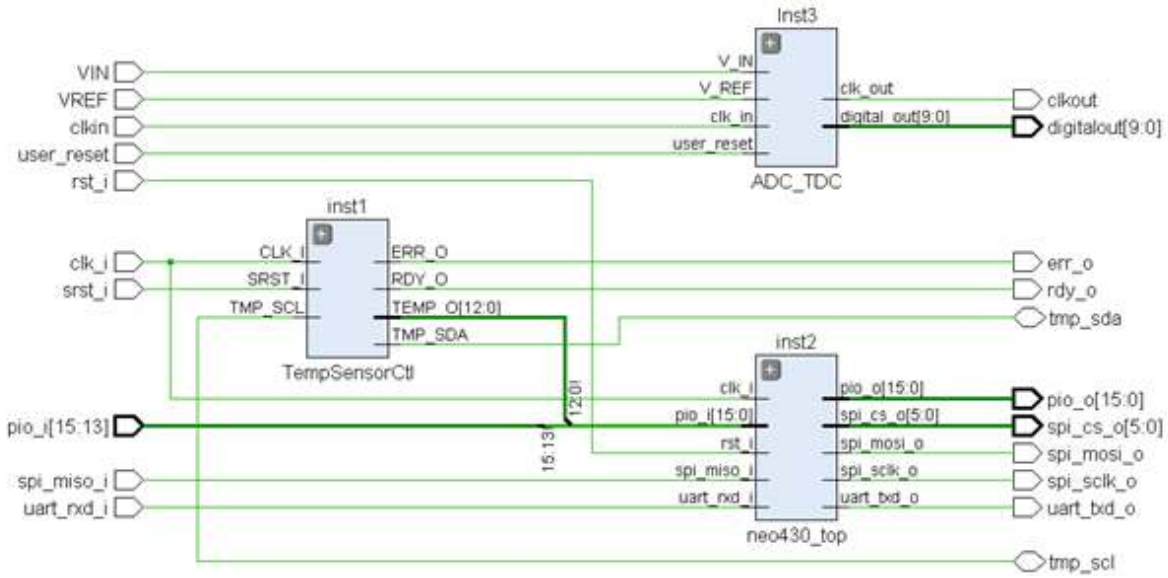
FigureIV.6.2 :implémentation du TDC sur FPGA

On remarque contrairement a la synthèse ;le TDC a occupé une large phase presque tout la surface après implémentation et le routage des signaux interne des composants existe .

IV.7 intégration du NEO 430 avec le TDC :

on propose de faire une combinaison entre le NEO 430 et le TDC ;c'est ta dire créer un circuit de connexion pour pouvoir l'intégrer dans notre FPGA .cette démarche exige la création d'un nouveaux projet vivado par la contribution d'un travail en description VHDL du NEO TOP en déclarant tout les entiers et leur architecture (NEO top ; contrôleur de température ;CAN tdc) ; on s'inspire du code réutilisée du NEO top pour régler le code du nouveaux circuit temp sensor +NEO top+CAN tdc .

on aura par la suite après l'intentassions un nouveaux circuit qui regroupe plusieurs composant l'entité devient component et on lance la synthèse .



FigureIV.7.1 : le schéma RTL après l'implémentation de tous les composants sur la carte FPGA

La Figure IV.7.1 représente implémentationgénérale des résultats de registre et le routages des circuits interne :inst 1 représente le contrôleur de température, inst 2 le NEO top et enfin inst 3 le CAN_TDC avec tout les entres et les sortie considéré

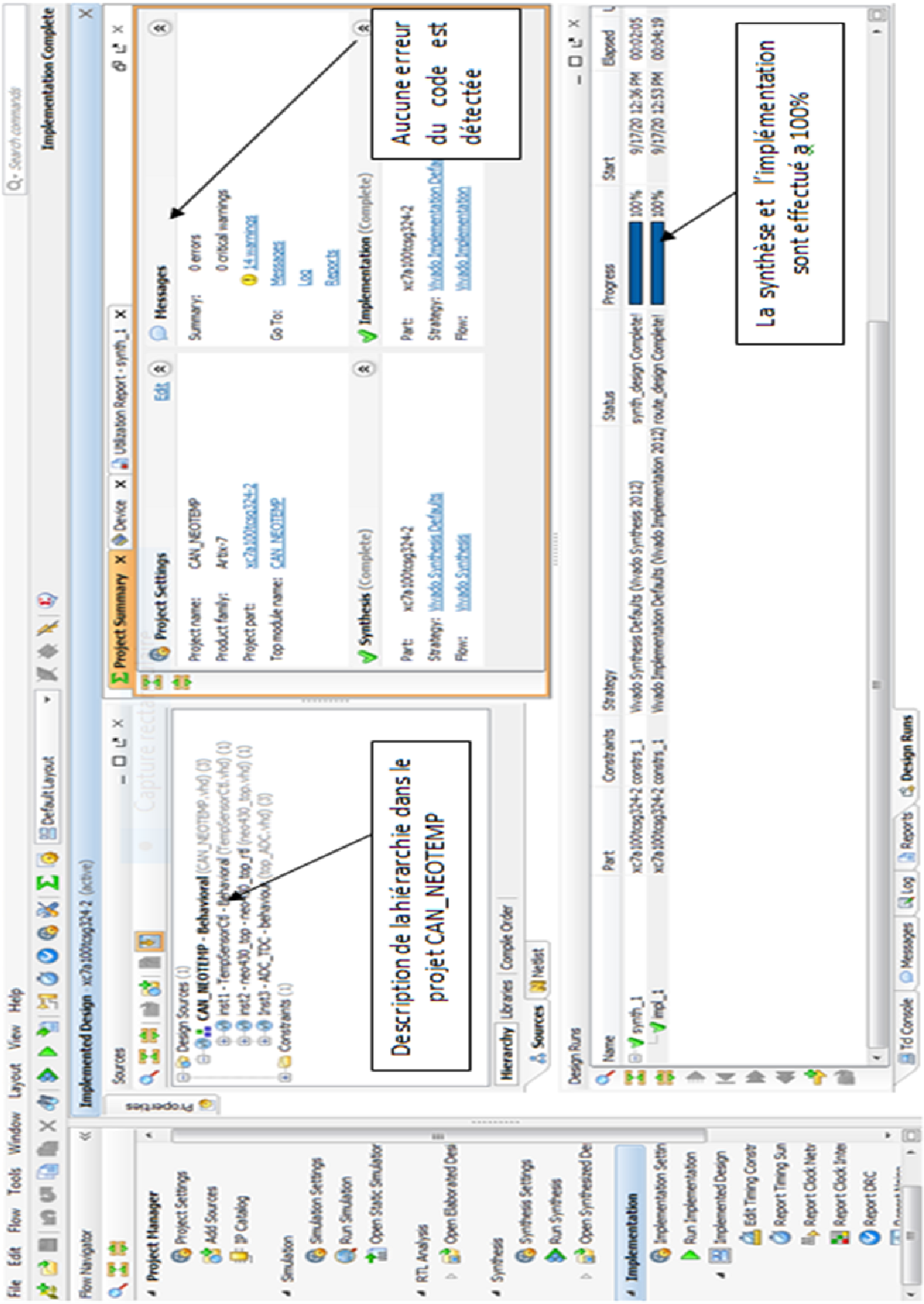


Figure IV.7.2 : la création du projet CAN_NEO_TEMP et la validation de la synthèse et l'implémentation sans aucune erreur

Le tableau IV.7représente la synthèse après l'implémentation :

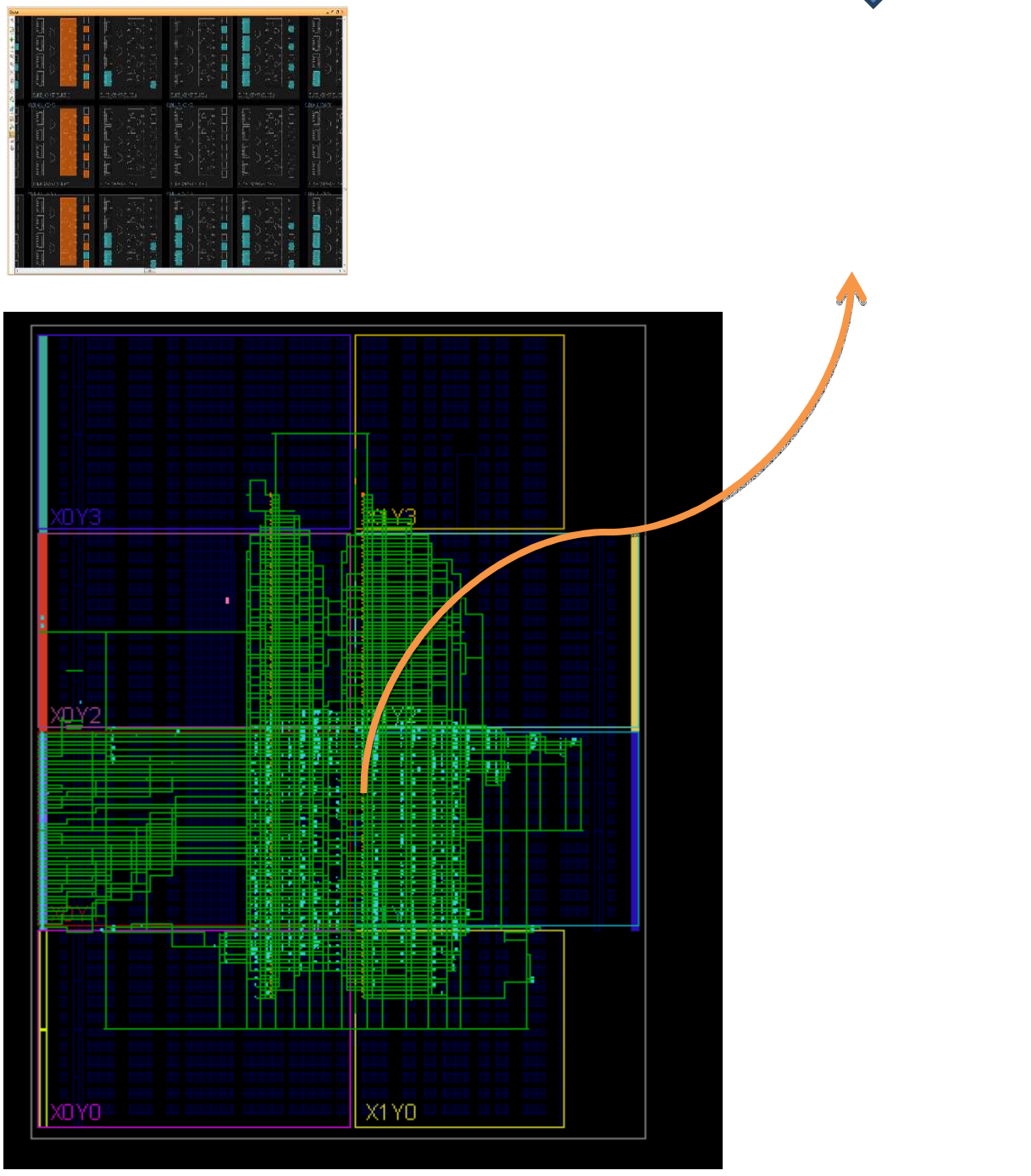
Site Type	Used	Loced	Available	Util%
Slice LUTs*	2400	0	63400	3 .78
LUT as Logic	2360	0	63400	3 .72
LUT as Memory	40	0	19000	0.21
LUT as Distributed RAM	32	0		
LUT as Shift Register	8	0		
Slice Registers	3862	1022	126800	3.04
Register as Flip Flop	3862	1022	126800	3.04
Register as Latch	0	0	126800	0.00
F7 Muxes	36	0	31700	0.11
F8 Muxes	16	0	15850	0.10

Tableau IV.7: utilisation des ressources du FPGA après l'implémentation

Le tableau IV.7Montre la consommation énergétique de notre système intégrée , on constate que seulement 7.71% est consommée par les différents LUTs des composants implémenté, le reste 6.08 % représente la consommation des registres du circuit FPGA Artix7. Une totalité de 0.21% est réservé pour les multiplexeurs.

le circuit FPGA peut compenser ça en offrant plus de flexibilité et une capacité de traitement supérieure pour le système

Exemple d'une porte logique



FigureIV.7.3 : placement finale des composants après la synthèse et l'implémentation sur FPGA

La figure IV.7. 3 représente le placement et le routage finale après l'intégration sur FPGA ,elle montre les différents blocs logiques de base : luts ;RAM ;registre et leur ordre de placements dans la carte FPGA .

Plus on approfondie dans ce circuit des milliers de porte logique s'ouvre.

On constate que notre intégration est possible après la validation du code avec le logicielle utilisée et les résultats de synthèses présentés .La vérification fonctionnelle de l'intégration est validée .

Par contre ; l'étape de simulation après l'implémentation n'est pas validée probablement a cause du signale d'entrés analogique proposée par les fournisseurs du code open source utilisée de cette conception . une possibilité de faire la simulation avec un oscilloscope et une proposition d'utilisé autre outils développé pour simulée ce signal analogique numérique en même temps .

IV.8CONCLUSION

Dans ce dernier chapitre, nous avons présenté les résultats de synthèse et d'implémentation du CAN sur FPGA .le programme réutilisé pour l'intégration du Can dans l'architecture du noeud de capteur est effectué et validée avec succès. nous avant déduit le schéma de placements des différentes composants NEO top, temp sensor et notre CAN TDC sur le circuit FPGA ainsi la synthèse et l'implémentation .

Une étape avancée été de propose des solution pour pouvoir réaliser la simulation âpres l'implémentation .

Introduction Générale

De nos jours, avec l'évolution de la technologie microélectronique d'une part, et les outils de conception d'autre part; il est devenu possible d'intégrer dans une seule puce, des systèmes complexes constitués de microprocesseurs, de circuits d'acquisition et traitement du signal et des protocoles de communication. On parle alors, de systèmes intégrés sur une seule puce et aussi de systèmes embarqués. Les domaines d'applications sont nombreux : médicale, industriel, militaire, environnementale, spatiale, pour ne citer que quelques exemples. Parmi ces derniers, les nœuds de capteurs et réseaux de capteurs sans fils constituent des applications où la tendance vers l'intégration et la miniaturisation est en pleine émergence.de point de vue architectural, un nœud de capteur est composé principalement d'une unité d'acquisition (capteur), d'un Convertisseur analogique numérique (CAN), d'une unité de traitement de données (microcontrôleur ou microprocesseur), d'une unité de communication, de mémoires et d'une source d'énergie (batterie).Généralement, Les microcontrôleurs commerciaux sont les unités de traitements les plus utilisés dansles nœuds de capteurs grâce à leur faible prix et faible consommation d'énergie, cependant les défisactuels des nœuds de capteurs en termes de communication, vitesse de traitement et de reconfigurationfont appel aux caractéristiques des circuits FPGA (Field Programmable GateArray) pour intégrer lesunités de traitements.

Par ailleurs, les CANs dont le rôle est de traduire une grandeur analogique en une valeur numérique(codée sur plusieurs bits), se présentent généralement sous forme de composants discrets quicommuniquent avec les circuits FPGA grâce à une interface dédiée. Néanmoins, cette communicationentre le circuit FPGA et le CAN n'est pas toujours facile à réaliser.

Le défi actuel de la conception consiste à intégrer dans le même circuit FPGA, qui est par vocation dudomaine numérique, le CAN, qui est traditionnellement conçu en analogique. Pour cela, la technique basée sur la modulation Sigma Delta est la plus communément utilisée. Néanmoins, bien qu'une résolution élevée, de l'ordre de 10 à 15 bits, puisse être atteinte, la fréquence d'échantillonnage dans ces circuit ne dépasse pas les quelques dizaines de kHz.

Afin d'améliorer les performances d'un CAN en terme de fréquence, la technique basée surl'utilisation d'un convertisseur temps numérique, en anglais « Time delayconverter », (TDC)est proposée.

Il s'agit en premier lieu d'étudier et d'analyser les convertisseur analogiques numériques ADC (spécification, architecture); particulièrement celui basé sur la technique TDC; ensuite de le concevoir sur FPGA et enfin de réaliser l'intégration de ce dernier dans l'architecture du nœud de capteur.

La partie intégration de notre travail représente un système sur puce construit autour du processeur NEO430, et du convertisseur ADC-TDC et le processeur NEO430, sont tous du domaine public. Leur code VHDL est disponible gratuitement et peut être téléchargé moyennant une licence GPL (General Public License).

L'outil Vivado est utilisé afin de valider l'apprentissage de cette méthodologie de conception pour la synthèse et l'implémentation de ce système dans le circuit FPGA Artix 7 de Xilinx hébergé sur la carte de développement Nexys 4 DDR de Diligent.

Dans ce contexte, le projet de fin d'études master qui nous a été proposé, porte sur l'intégration sur d'un convertisseur analogique dans l'architecture d'un nœud de capteur basée sur FPGA.Dans ce contexte, l'objectif de ce projet de fin d'étude master est de réaliser l'interfaçage entre le monde réel (analogique) et le monde digital ; Il s'agit d'étudier et d'analyser le convertisseur ananlogique numérique ADC (spécification, architecture) ;ensuite réaliser l'intégration d'un ADC dans l'architecture d'un capteur de températureADT7420 implémenté sur le circuit FPGA (réutilisation d'un code IP VHDL open source téléchargeable gratuitement d'un convertisseur analogique de type TDC) .enfin faire l'implémentation et les résultats de synthèse pour chaque composant et pour tout les composants intégrés sur la carte .l'outil vivado est utilisée pour valider le fonctionnement du convertisseur et du circuit FPGA ; une étape avancée serait de faire la simulation après l'implémentation .

La partie intégration dans notre travail représente un système construit autour du processeur soft NEO430 et de certain composants IP (intellectual properties) décrits entièrement en VHDL (Very high speed Hardware Description Language). Le processeur soft NEO430 et les composants IPs appartiennent au domaine publique (opensource/opencores) et leur code source est téléchargeable gratuitement. L'outil Vivado est utilisé afin de valider l'apprentissage de cette méthodologie de conception pour la synthèse et l'implémentation de ce système dans le circuit FPGA Artix 7 de Xilinx hébergé sur la carte de développement Nexys 4 DDR de Diligent.

Ce mémoire comporte quatre chapitres , le premier chapitre décrit Les nœuds de capteur basés sur FPGA et leur domaines d'application ; leurCaractéristiques et contraintes par la suite le second chapitre présente un état de l'art sur l'implémentation de CAN sur FPGA ;a partir des différentes travaux réalisé su CAN des différentes modèles de convertisseurs et FPGAs ont été présenté aussi .

Le troisième chapitre concerne Intégration d'un CAN TDC dans l'architecture d'un nœud de capteur. l'architecture interne du processeur NEO 430 et de notre approche CAN tdc ainsi l'idée de la pratique était présenté en détail .Le quatrième et le dernier chapitre traite les Résultats et synthèse de l'implémentation du CAN sur FPGA notamment la validation de notre intégration est présenté.

Enfin, ce mémoire s'achèvera par une conclusion générale sur l'intégralité de notre travail en résumant les accomplissements de ce travail et aussi présentant les différentes perspectives qui peuvent améliorer des projets similaires dans la future.

Conclusion générale

L'objectif de ce travail est d'implémenter un convertisseur analogique numérique dans l'architecture d'un nœud de capteur implémenté sur un circuit FPGA . L'implémentation sur FPGA de cette unité permet de garantir une communication entre le monde digitale et le monde analogique dans les circuits FPGAs ; un défis qui n'est pas attribué par avant .

L'unité de traitement en question est à la base un système dont l'architecture est construite autour du processeur soft NEO430 et de certains composants CAN adc dont son IP est téléchargeable gratuitement un contrôleur de température et un capteur de température ADT 7420 .

La complexité de ces systèmes impose l'apprentissage d'une méthodologie de conception basée sur la réutilisation d'un IP open source choisie .

Pour notre part, nous avons choisi de réutiliser des IPs du domaine public, plus précisément ceux issus du site Opencores. Ce choix est justifié par le fait que le code VHDL des différents IPS constituants non seulement téléchargeable gratuitement, mais aussi les IPs peuvent être réutilisés avec un haut degré de confiance.

Nous avons ciblé le circuit FPGA Artix7 de Xilinx hébergé sur la carte de développement Nexys 4 DDR, pour pouvoir intégrer le CAN dans l'architecture du nœud de capteur implémenté sue ce dernier .

Les résultats de synthèse et d'implémentations obtenues de l'outil Vivado montre que notre système sur puce n'utilise pas beaucoup de ressources du circuit FPGA Artix 7 avec seulement 6.08 % des registres interne ,7.71% des Luts et 0.21 % des Blocs mémoires et multiplexeur offrant la possibilité d'élargir encore plus notre système. Les résultats montrent aussi que notre système fonctionne normalement dans ce processeur .

Afin de valider la fonctionnalité de notre système , nous avons proposé de l'utilisé pour récupérer et afficher des mesures à partir de d'un capteurs différents d'Analog Devices hébergé aussi sur la carte de développement Nexys 4. Le premier paramètre à récupérer concerne la température de l'environnement extérieur en communiquant avec le capteur de température numérique ADT7420 et afficher la température dans un afficheur 7 segment .

Nous avons utilisé l'outil vivado pour voir la simulation et la synthèse d'intégration au processeur NEO430. L'outil offre aussi la possibilité de règle le programme d'application automatiquement à partir des bibliothèques des composants entres tdc et NEO top avant le lancement du travail .

Le code VHDL réutilisé règle assure l'acquisition de la température sur les trois axes afin de les afficher sur les modules 7 segments de la carte de développement Nexys 4.

Les résultats obtenus montrent que l'unité de traitement CAN a été implémentée dans l'architecture du nœud de capteur et testée sur FPGA avec succès.

Notre unité de traitement possède les avantages suivants : elle est complètement configurable, elle peut être adaptée avec plusieurs types de capteurs en ajoutant ou en éliminant des modules tout dépend de l'application, et pour plus de flexibilité, le système aussi offre la possibilité de compiler et de générer l'exécutables des codes d'applications décrit en VHDL.

Comme perspective à ce travail, nous proposons d'utiliser des version d'outil plus développer pour permettre le fonctionnement de la simulation âpres l'implémentation .un simulateur qui fait la simulation d'un signal mixte analogique numériqueest exigée.

Donc pour améliorer notre système, nous proposons aussi d'utiliser un générateur de signal pour valider la simulation .

Bibliographie

[1]www.wikipedia.org/wikireseau_de_capteurs_sans_fil Applications

[2] université Montréal ,Mémoire présenté à la Faculté des études supérieures en vue de l'obtention du grade de Maîtrise sciences (M.Sc.) Informatique ; © Boushaba Mustapha, 2007

[3] Université Abou BakrBelkaid– Tlemcen, Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique ;Implémentation et test d'un protocole deprévention de l'attaque Clone dans un réseaude capteurs sans fil ;2014

[4] Yacine Younes, « Minimisation d'énergie dans un réseau de capteurs », mémoire de magister, Université Mouloud Mammeri de Tizi-Ouzou, Algérie, 2012

[5] BENZID Sofiane « Conception d'un système sur FPGA pour le traitement des données d'un nœud de capteurs », Mémoire de master, Ecole Nationale Supérieure de Technologie, 2017.

[6]Milagros roman « implémentation d'une application de surveillance de température », Mémoire de master, département des recherches opérationnelles ,université de Montréal 2008.

[7] www.emse.fr/dutertre/enseignement.html - 2009

[8] www.wikipedia.org/wiki/Convertisseur_analogique-numerique

[9] C. TRABELSI, « Contrôle matériel des systèmes partiellement recouvrables sur FPGA : de la modélisation à l'implémentation », Thèse doctorat, Université des Sciences et Technologies de Lille, 2013.

[10]CristianSisterna, Marcelo Segura, Martin Guzzo, Gustavo Ensinck, Carlos Gil Departamento de Electrónica y AutomáticaFacultad de Ingeniería, UniversidadNacional de San Juan {cristian, msegura, m.guzzo, gustavo, cgil} @unsj.edu.ar

[11] Antonio J.Ascota,Adoracion Rueda and Jose L.Huertas, "A VHDl-basedMethodology for the Design and Verification of Pipeline A/D Converters",Proceedings of Design,Automation and Test in Europe Conference and Exhibition,pp.534-538,March 2000.

[12] M. Segura, H. Hashemi, C. Sisterna, V. Mut, "ExperimentalDemonstration of Self Localized Ultra Wideband Indoor Mobile Robot Navigation System." IEEE International Conference on Indoor Positioning and Indoor Navigation

(IPIN), Zürich, Switzerland. 15-17 September 2010.

[13] WP442 (v1.0.2) March 16, 20 °°Efficient Implementation of Analog Signal ProcessingFunctions in Xilinx All Programmable Devices By: Cathal Murphy.

[14] tudelft.nl/fpga_tdc/ADC_basic.html

[15]SNolting NEO430 Processor ,juillet 2016 .

[16]Dgilent ;(Nexys4 DDRTM FPGA Board Reference Manual), Avril 2016.

[17] Nexys 4 DDR FPGA Booard Reference Manual:reference.digilentinc.com.

[18]Analogue Devices , ADT7420 Dtatasheet , 2012

[19] www .Opencores. com

Annexe

Programme d'application en VHDL

```
- Company:
-- Engineer:
--
-- Create Date: 17.09.2020 11:18:52
-- Design Name:
-- Module Name: CAN_NEOTEMP - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description: Integration du convertisseur analogique numérique TDC dans
--              l'architecture d'un noeud de capteur a base du microcontroleur NEO430,
--              TDC et NEO430 sont du domaine public
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments: Ce travail a été réalisé par Benlalam Asma et Halfaoui Maroua
-- ****Encadré par N. Izeboudjen,Projet NCIR/CDTA****
--
----------------------------------------------------------------------------------



library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.math_real.all;
use IEEE.std_logic_arith.all;
```

```
usework.TWIUtils.ALL;
library work;
use work.neo430_package.all;
USE ieee.std_logic_unsigned.ALL;
USE ieee.std_logic_misc.ALL;
LIBRARY unisim;
USE unisim.vcomponents.ALL;
USE work.tdc_library.ALL;

-- Uncomment the followinglibrarydeclaration if using
-- arithmeticfunctionswithSigned or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the followinglibrarydeclaration if instantiating
-- anyXilinx primitives in this code.
library UNISIM;
useUNISIM.VComponents.all;

entity CAN_NEOTEMP is
      GENERIC (
                  STAGES                                          : INTEGER := 512;
                  FINE_BITS                                        : INTEGER := 9;
                  Xoff_TDC1                                        : INTEGER := 34;
                  Xoff_TDC2                                        : INTEGER := 52;
                  Yoff                                            : INTEGER := 32);
   Port (clk_i : in STD_ULOGIC;
rst_i : in STD_ULOGIC;
uart_rxd_i : in STD_ULOGIC;
uart_txd_o : out STD_ULOGIC;
srst_i : in STD_ULOGIC;
rdy_o : out STD_ULOGIC;
err_o : out STD_ULOGIC;
tmp_scl : inout STD_ULOGIC;
```

```
tmp_sda : inout STD_ULOGIC;
pio_i : in STD_ULOGIC_VECTOR (15 downto 13);
pio_o : out STD_ULOGIC_VECTOR (15 downto 0);
spi_sclk_o : out STD_ULOGIC;
spi_miso_i : in STD_ULOGIC;
spi_mosi_o : out STD_ULOGIC;
spi_cs_o : out STD_ULOGIC_VECTOR (5 downto 0);
user_reset        : IN std_logic;
clkin             : IN std_logic;                                          -- input 200 MHz clock
clkout            : OUT std_logic;                                         -- clock to generateramp
        VIN               : IN std_logic;                                  -- the analog input signal
        VREF              : IN std_logic;                                  -- the reference input signal (ramp)
        digitalout        : OUT std_logic_vector(FINE_BITS DOWNTO 0)
    );
end CAN_NEOTEMP;

architectureBehavioral of CAN_NEOTEMP is
signal s: std_ulogic_vector(12 downto 0);
componentTempSensorCtl
        Generic (CLOCKFREQ : natural := 100); -- input CLK frequency in MHz
        Port (
                TMP_SCL : inout STD_ULOGIC;
                TMP_SDA : inout STD_ULOGIC;
--              TMP_INT : in STD_ULOGIC; -- Interrupt line from the ADT7420, not used in thisproject
--              TMP_CT : in STD_ULOGIC;  -- CriticalTemperatureinterrupt line from ADT7420, not used in
thisproject

                TEMP_O : out STD_ULOGIC_VECTOR(12 downto 0); --12-bit
two'scomplementtemperaturewithsign bit
                RDY_O : out STD_ULOGIC;          --'1' whenthereis a validtemperaturereading on TEMP_O
                ERR_O : out STD_ULOGIC; --'1' if communication error

                CLK_I : in STD_ULOGIC;
```

```
                SRST_I : in STD_ULOGIC
        );
end component;

component neo430_top
port (
   -- global control --
clk_i      : in  std_ulogic; -- global clock, risingedge
rst_i      : in  std_ulogic; -- global reset, async
   -- parallelio --
pio_o      : out std_ulogic_vector(15 downto 0); -- parallel output
pio_i      : in  std_ulogic_vector(15 downto 0); -- parallel input
   -- serial com --
uart_txd_o : out std_ulogic; -- UART send data
uart_rxd_i : in  std_ulogic; -- UART receive data
spi_sclk_o : out std_ulogic; -- serial clock line
spi_mosi_o : out std_ulogic; -- serial data line out
spi_miso_i : in  std_ulogic; -- serial data line in
spi_cs_o   : out std_ulogic_vector(05 downto 0) -- SPI CS 0..5
  );
end component;

component ADC_TDC
        GENERIC (
                STAGES                                          : INTEGER := 512;
                FINE_BITS                                       : INTEGER := 9;
                Xoff_TDC1                                       : INTEGER := 34;
                Xoff_TDC2                                       : INTEGER := 52;
                Yoff                                            : INTEGER := 32);
        PORT (
                user_reset: IN std_logic;
                clk_in    : IN std_logic;                       -- input 200 MHz clock
                clk_out   : OUT std_logic;                      -- clock to generateramp
```

```
                V_IN     : IN std_logic;                                    -- the analog input signal
                V_REF  : IN std_logic;                                  -- the reference input signal (ramp)
                digital_out: OUT std_logic_vector(FINE_BITS DOWNTO 0));
END component;



begin

inst1: TempSensorCtl
    port map(
        TMP_SCL =>tmp_scl,
        TMP_SDA =>tmp_sda,
        TEMP_O(12 downto 0) => s(12 downto 0),
        RDY_O =>rdy_o,
        ERR_O =>err_o,
        CLK_I =>clk_i,
        SRST_I =>srst_i);

inst2: neo430_top
    port map(
clk_i =>clk_i,
rst_i =>rst_i,
pio_o (15 downto 0) =>pio_o (15 downto 0),
pio_i (12 downto 0) => s (12 downto 0),
pio_i (15 downto 13) =>pio_i (15 downto 13),
uart_txd_o =>uart_txd_o,
uart_rxd_i =>uart_rxd_i,
spi_sclk_o =>spi_sclk_o,
spi_mosi_o =>spi_mosi_o,
spi_miso_i =>spi_miso_i,
spi_cs_o (5 downto 0) =>spi_cs_o(5 downto 0));

 Inst3: ADC_TDC
```

```
 port map(
user_reset =>user_reset,
            clk_in     =>clkin,
                        clk_out  =>clkout,
                        V_IN     =>VIN,
                        V_REF  =>VREF,
                        digital_out(9 downto 0)=>digitalout(9 downto 0));



end Behavioral;
```