

Exploitation des Ressources du CNFM pour l'Apprentissage de l'Électronique Numérique en DUT GEII

V. Frick, *Maître de conférences, Université de Strasbourg*
CNFM – Pôle MIGREST
vincent.frick@iness.c-strasbourg.fr

Résumé— Cet article présente un ensemble d'activités s'inscrivant dans le cadre des modules d'apprentissage de l'électronique numérique et de l'informatique industrielle proposés dans la formation au DUT en Génie Électrique et Informatique Industrielle à l'IUT de Haguenau (Université de Strasbourg). Ces activités reposent sur l'utilisation de ressources FPGA, qui offrent de nombreuses possibilités en terme de fonctionnalités matérielles et logicielles pouvant être associées aux concepts de bases abordés dans les niveaux L1 et L2. Elles permettent de ce fait d'appréhender les problématiques de conception de systèmes numériques de manière plus complète.

Mots clés— CAO, langage VHDL, programmation C, synthèse de processeur, prototypage, FPGA.

I. INTRODUCTION

L'ÉVOLUTION des systèmes électroniques est telle que les développeurs doivent être capable d'avoir une vision plus complète d'un système complexe, hiérarchisé et dont l'association matériel-logiciel est de plus en plus forte. Dans le cadre de leur formation au DUT en Génie Électrique et Informatique Industrielle (GEII), les étudiants de l'IUT de Haguenau, composante de l'Université de Strasbourg, ont l'occasion d'exploiter des outils de conception avancés dont les licences sont fournies par le pôle de Micro et nanoélectronique du Grand-EST (MIGREST) du CNFM. L'utilisation de ces outils constitue une excellente opportunité d'effectuer leurs premiers pas vers l'ingénierie des systèmes complexes, sachant qu'à l'heure actuelle 80% des étudiants titulaires d'un DUT poursuivent leurs études, dont 50% en école d'ingénieur.

Construit autour de la **progression** des étudiants dans leur cursus en GEII, cet article s'articule sur 3 parties. La première expose les activités liées à l'apprentissage des fondamentaux en électronique numérique, exploitant un environnement logiciel de conception de systèmes numériques associé à des ressources matérielles de type FPGA. La seconde partie présente les activités abordées dans le module d'enseignement traitant de la mise en œuvre de circuits FPGA par le biais de la conception en

le langage VHDL. La dernière partie est consacrée à Processeurs.

II. TP ÉLECTRONIQUE NUMÉRIQUE EN 1^{ÈRE} ANNÉE

A. Contexte

Le programme pédagogique national de la formation GEII (PPN GEII) prévoit un module d'apprentissage des bases de l'électronique numérique destiné aux étudiants de 1^{ère} année GEII. Étant donnée la diversité de leurs baccalauréats d'origine, ce module, survenant en tout début d'année, a pour but d'harmoniser leur compétences et aborde donc les fondamentaux des systèmes numériques tels que la logique booléenne, les portes logiques de base, les notions de logique combinatoire et séquentielle ainsi que celles de systèmes asynchrones et synchrones.

Ce module comprend un ensemble de 6 séances de travaux pratiques d'une durée de 4 h chacune. La plupart de ces séances consistent à concevoir et simuler des systèmes à l'aide des outils Altera®.

Il est important de remarquer que le degré de virtualisation et d'abstraction des outils de CAO actuels est tel que les aspects électriques peuvent aisément échapper aux néophytes. Aussi, afin qu'ils prennent conscience des paramètres physiques et électriques des circuits numériques qu'ils seront amenés à manipuler, les étudiants sont-ils préalablement invités à faire quelques montages à partir de composants discrets de technologie TTL lors de la première séance pratique. Ils appréhendent par ce biais les notions de tension d'alimentation, d'impédance d'entrée et de sortie, de fan out, etc., qui sont autant de caractéristiques qui n'apparaissent a priori pas explicitement dans les outils de développement de systèmes numériques tels que Quartus II®. Ils prennent également conscience de la problématique, voire la pénibilité, du routage manuel en réalisant des câblages à base de bascules JK et D pour réaliser des registres et des compteurs synchrones ou asynchrones.

À partir de la seconde séance, la pratique se déroule exclusivement sur station de travail. Pour les étudiants, il s'agit dans en premier temps de prendre en main les outils de CAO Altera®. Cette seconde séance est donc consacrée à l'initiation à Quartus II® par l'élaboration de projets de conception simples consistant à saisir des schémas en

□Manuscrit proposé aux Journées Pédagogiques de la Coordination Nationale pour la Formation en Micro et nanoélectronique 2010.

mode graphique, à simuler et à implanter sur cible FPGA quelques fonctions logiques combinatoires et séquentielles de bases. Un document didacticiel de référence a été élaboré. Il est à tout moment à leur disposition sur l'intranet de l'IUT pour les guider en cas de problème lorsqu'ils travaillent en autonomie.

Dans un second temps, il leur est progressivement demandé d'exploiter les potentialités des outils pour analyser les caractéristiques et performances des systèmes étudiés (temps de propagation, stabilité, etc.). Outre l'étude de systèmes à complexité croissante, le but de ces séances est d'inciter les étudiants à développer leur esprit critique.

Dans le cadre de ce module, l'implantation sur cible FPGA de type Cyclone est essentiellement destinée à effectuer les vérifications fonctionnelles des circuits étudiés. Les étudiants n'utilisent que les ressources disponibles sur la carte de test (horloge, afficheurs, interrupteurs et boutons poussoirs). À ce stade de leur formation, il ne leur est pas encore demandé d'associer cette carte avec d'autres dispositifs (transmission de données sur PC, interfaçage avec carte de commande, etc.)

B.Exemple de sujet de TP ENSLI 1^{ère} année

Le système décrit ici est un exemple de sujet proposé aux étudiants en fin de module. Il permet de d'aborder les notions telles que la conception d'un design hiérarchique associant des blocs combinatoires et séquentiels, l'étude des temps de propagation, le synchronisme et le temps réel. Le temps imparti à sa réalisation est de 4 h. Chaque étudiant travaille individuellement.

Il s'agit de réaliser un compteur de vitesse numérique pour scooter. L'affichage de la vitesse doit être compris entre 15 et 97 km/h par palier sur les unités : 1 – 3 – 5 – 7. Par exemple : 21 km/h, puis 23 km/h, puis 25 km/h, puis 27 km/h et 31 km/h, etc.

L'affichage est effectué au moyen des afficheurs 7 segments de la carte FPGA et est réactualisé toutes les 500ms. En outre, si la vitesse est inférieure à 15 km/h ou excède 97 km/h, les symboles « -- » sont affichés.

Le schéma bloc du système est présenté dans la figure 1. Il y apparaît 1 bloc permettant d'émuler un capteur qui envoie des impulsions à une fréquence proportionnelle à la vitesse :

- Pour 15 km/h, le capteur envoie 150 impulsions en 500 ms
- Pour 21 km/h, le capteur envoie 210 impulsions en 500 ms
- etc.

Afin de contenir la séance de TP dans son temps imparti, ce bloc ainsi que le diviseur de fréquence paramétrable sont fournis aux étudiants depuis une bibliothèque accessible en ligne.

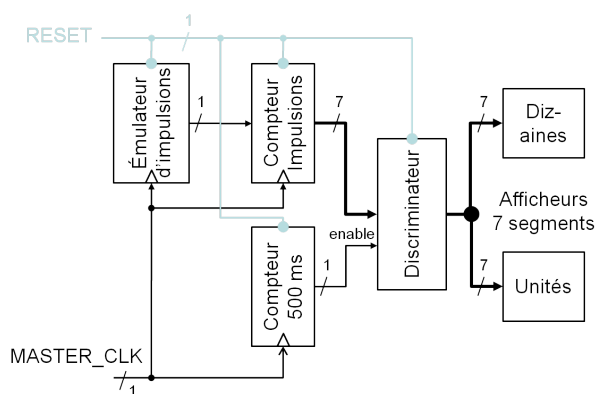


Fig. 1. Schéma bloc du système compteur de vitesse.

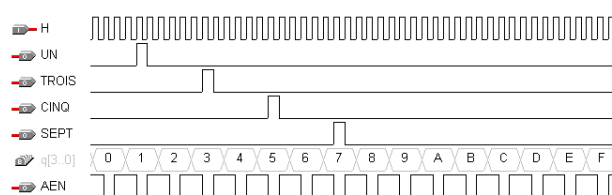


Fig. 2 : Exemple de simulation fonctionnelle d'un sous-ensemble du compteur de vitesse numérique.

Après compilation, simulation (cf. figure 2), analyse fonctionnelle et temporelle du système, celui-ci est implanté sur la carte de développement FPGA Cyclone Board NIOS II[®]. L'horloge principale MAIN_CLK est fournie par le quartz à 10 MHz de la carte et la vitesse est simulée au moyen de deux boutons poussoirs qui permettent d'accélérer ou de décélérer par pas de 1 km/h à chaque impulsion.

C.Exploitation du testeur CNFM de Montpellier

Les outils de CAO tels que Quartus II[®] sont essentiellement orientés vers la conception de systèmes numériques. Comme cela a été évoqué plus haut, certaines notions telles que les niveaux électriques, les effets de collage, les courants de sortie, la robustesse, etc. sont néanmoins autant de critères, souvent déterminant dans le choix d'une technologie, auxquels les étudiants risquent de ne plus être suffisamment sensibilisés. Aussi, depuis la rentrée 2010, l'IUT de Haguenau propose une séance de TP en ligne exploitant le testeur du CNFM de Montpellier.

Ce TP d'une durée de 4 heures a été écrit par Mme Pradarelli [1], ingénieure de recherche chargée du test et de la formation au test pour le pôle CNFM de Montpellier. Il consiste, pour les étudiants en IUT travaillant en binômes, à mettre en œuvre un banc de caractérisation d'un registre universel 8 bits de type 74ACT299 sur une plateforme HP93000 à partir du programme Verigy SmartTest.

Outre la nouveauté liée au TP sur testeur, il s'agit également d'une opération expérimentale de TP en ligne de supervision en temps réel au moyen d'une connexion de type Virtual Network Computing (VNC).

III. MODULE COMPLÉMENTAIRE D'ÉLECTRONIQUE NUMÉRIQUE EN 2^{ÈRE} ANNÉE

A. Contexte

Un module complémentaire intitulé « Mise en œuvre de FPGA » est proposé en deuxième année de DUT GEII. Déjà familiarisés avec les cartes de développement FPGA lors de leur 1^{ère} année, les étudiants ont ici l'occasion de découvrir un langage de description matériel, en l'occurrence le VHDL. Il permet ainsi d'exploiter plus pleinement les cartes FPGA et donc d'aborder des sujets plus complexes et plus complets en terme de transversalité, ce qui n'est pas aisé lorsqu'un projet est développé en mode graphique en raison de la lourdeur de manipulation des designs.

Les séances de cours sont réduites au nombre de 3 (soit 6 h) pour laisser une plus grande part à la pratique du langage VHDL. Elles sont essentiellement consacrées à la présentation de la technologie des circuits FPGA et aux généralités sur les concepts et la structuration du VHDL. Lors des 6 séances pratiques d'une durée de 4 h chacune les étudiants travaillent individuellement sur PC. La complexité des sujets est croissante au fil des séances. La première séance est dédiée à la prise en main du langage à travers la description des fonctions logiques de base (ET, OU, XOR, additionneur binaire, D latch, D flip-flop, etc.). L'objectif en fin de module est la maîtrise des instructions permettant de décrire des modèles de systèmes simples mais paramétrables et hiérarchisés mettant en œuvre des notions transversales tel que le traitement du signal (filtrage numérique).

B. Exemple de sujet de TP MCENSL1 2^{ème} année

Un des sujets proposés porte sur la réalisation d'un filtre de décimation pour modulateur Sigma-Delta. Notons qu'en raison de la transversalité du sujet le module de 2^{ème} année intitulé « filtrage numérique » est pré-requis.

La topologie de ce filtre à Réponse Impulsionnelle Finie (FIR) est présentée dans la figure 3. Les échantillons provenant du modulateur sont émis sur 1 bit à une cadence de 1 MHz et la décimation doit permettre de réduire à environ 3,9 kHz la cadence des échantillons de sortie codés sur 10 bits. Le facteur de suréchantillonnage du modulateur est donc de 256.

Le principe de filtrage consiste à remplir une RAM FIFO avec n échantillons issus du modulateur. Lorsque les n échantillons sont stockés, le calcul du FIR est déclenché. Une ROM contient les n (par exemple 128) coefficients caractérisant le filtre. Chaque coefficient est additionnés ou soustrait dans un accumulateur en fonction de l'état du bit correspondant dans la FIFO.

L'étude de la réponse spectrale du filtre et le calcul des coefficients est réalisée préalablement à l'aide de l'outil FDATool de Matlab[®]. Ils sont exportés dans un fichier texte sous forme d'entiers codés sur 16 bits puis importés dans le code VHDL.

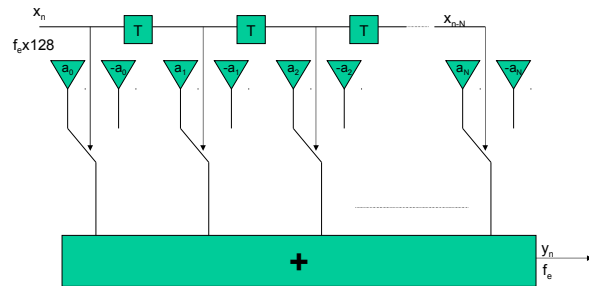


Fig. 3. Topologie du filtre FIR.

Les éléments de complexité liés à la conception et à la synthèse de ce système sont minimales. Il s'agit de :

- générer une ROM de 2 kbits (128 coefficients de 16 bits)
- créer une machine d'état capable de gérer le calcul du FIR par accumulation des coefficients sans perte d'échantillon du modulateur.

L'architecture générale du système est présentée dans la figure 4. Elle illustre la simplicité de concept de filtrage. La difficulté ne résidant ici que dans le séquençage et la gestion des pointeurs de la RAM et de la ROM.

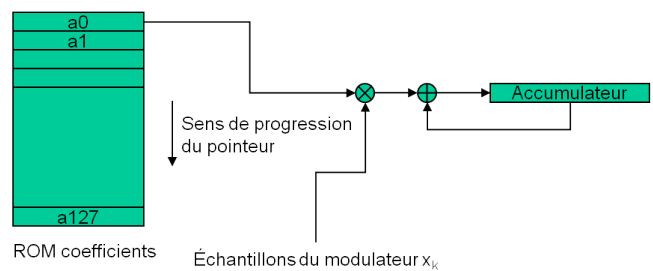


Fig. 4. Architecture générale du FIR. Lorsque le pointeur de coefficients atteint la fin du filtre, la somme contenue dans l'accumulateur contient le prochain échantillon de sortie du filtre. L'échantillon est sorti et l'accumulateur est remis à 0. Note: x_k vaut 1 ou -1. Le multiplieur se limite donc à un choix entre le coefficient ou son opposé.

Pour les simulations, les étudiants utilisent le logiciel ModelSim[®] dont les fonctionnalités sont plus avancées que celles du simulateur natif de Quartus II[®]. Le vecteur de test contient un enregistrement de 100 kpoints d'un train de bits issus d'un modulateur Sigma-Delta réel. Ce train de bits est fourni aux étudiants sous forme d'un fichier texte disponible dans une bibliothèque en ligne. Il est également enregistré dans une ROM codée en VHDL, aussi fournie aux étudiants et permettant l'implantation sur la carte de développement FPGA.

Pour le test du filtre, cette ROM est lue de manière cyclique par le système afin d'émuler un modulateur en régime permanent. La lecture des échantillons de sortie du filtre est réalisée au moyen d'un convertisseur numérique/analogique de type DAC0808 associé à un convertisseur courant/tension selon le schéma de la figure 5. Cette tension est visualisée sur oscilloscope.

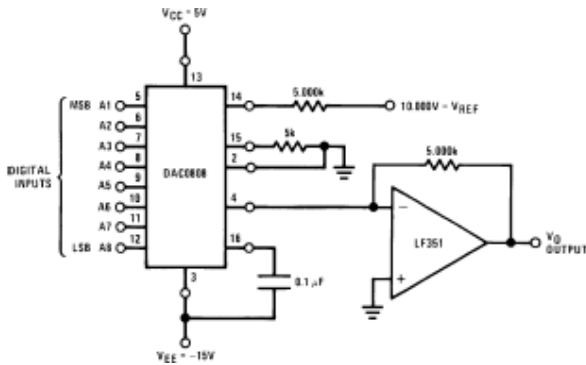


Fig. 5 : Convertisseur N/A en sortie du FIR.

IV.PROJETS TRANSVERSAUX

A. Contexte

Outre les modules d'enseignement comportant des séances pratiques encadrée, les étudiants de DUT sont également amenés à réaliser des projets en autonomie partielle. Les sujets traités à cette occasion peuvent leur permettre d'adopter une approche « système », dans laquelle ils abordent les notions de co-design matériel-logiciel.

Pour mener à bien leur projet ils disposent d'une dizaine de créneaux « libres » de 4 h dans leur emploi du temps. Ils travaillent en binômes et disposent de l'ensemble des ressources nécessaires à la réalisation du projet (PC portables équipés des suites logicielles Altera® et Matlab® cartes de développement FPGA, composants discrets, etc.)

B.Exemple de sujet de projet : système d'acquisition de données d'une caméra numérique

Dans ce sujet, librement inspiré des travaux exposés dans [2], il est demandé aux étudiants de réaliser une interface entre une caméra numérique et un processeur embraqué de type « softcore » NIOS II® développé à partir d'un outil de synthèse spécifique, SOPC Builder®, d'Altera®. Ce processeur est chargé d'assurer le transfert des données acquises par la caméra vers un PC au moyen d'un liaison RS 232.

L'outil SOPC Builder® permet également de générer et de synthétiser les périphériques du NIOS II® tels que le bus Avalon® faisant l'interface avec la caméra et les ports de communication JTAG et UART. La figure 6 montre le schéma synoptique du système.

Après compilation, le système est simulé grâce au logiciel ModelSim®. La figure 8 présente une simulation du système en configuration d'acquisition de données.

La figure 7 montre la structure de l'interface décrite en langage VHDL et permettant de connecter la caméra sur le bus Avalon®.

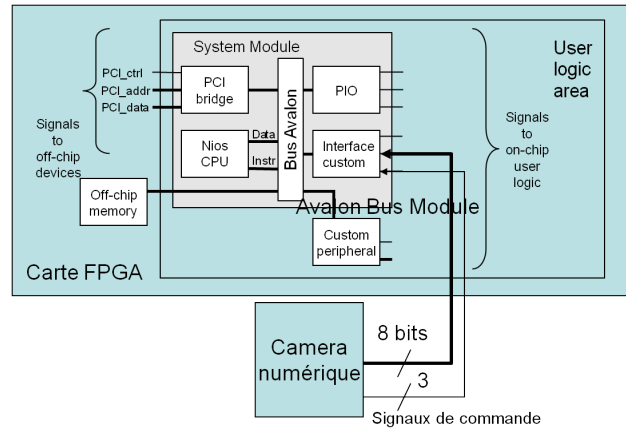


Fig. 6. Synoptique du système d'acquisition de données issues d'une caméra numérique.

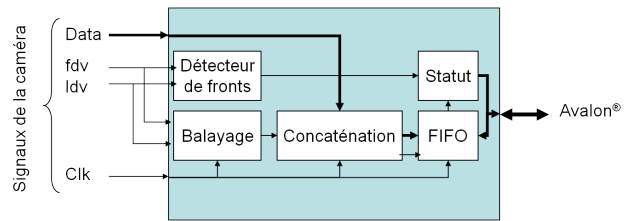


Fig. 7. Structure de l'interface « camera – bus Avalon® »

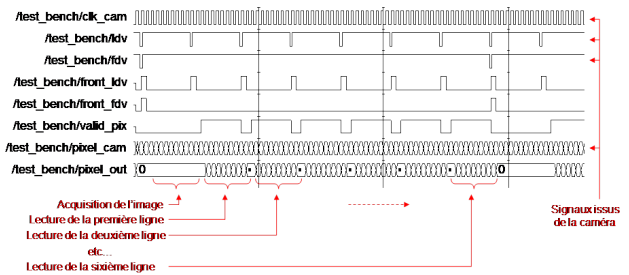


Fig. 8 : Extrait de la simulation d'une séquence d'acquisition.

Une fois le NIOS II® synthétisé et implanté sur la carte de développement FPGA, sa communication avec la caméra est gérée par le protocole propre au bus Avalon® dont la conception du driver par l'outil de développement NIOS IDE C++ est fortement assistée. Ce logiciel permet de développer en langage C et de piloter le NIOS II® à partir des fonctions disponibles dans des bibliothèques spécifiques. La caméra est connecté sur les broches disponibles de la carte de test au moyen d'un nappe ad-hoc.

Il est ainsi possible de stocker les données issues de la caméra dans la RAM du NIOS II®. Ces données sont ensuite transmises au port UART du NIOSII® et récupérées sur un PC via le port série RS232 disponible sur la carte de développement et dont la gestion est assurée par un programme d'émulation de terminal PC. Les données peuvent ensuite être traitées avec un logiciel de traitement du signal ad-hoc tel que Matlab®.

V.CONCLUSION

Le potentiel des ressources FPGA est tel qu'il facilite l'apprentissage de l'électronique numérique et permet une

forte progression des étudiants qui, débutants dans le domaine de l'électronique pour la plupart lorsqu'ils intègrent la 1^{ère} année de formation au DUT GEII, parviennent assez rapidement à appréhender des systèmes relativement complexes.

Les outils de CAO étant souvent fastidieux à mettre en œuvre, il est donc évident qu'une assistance soutenue et la mise à disposition de blocs complexes préalablement développés peut se révéler indispensable au succès de cette technique pédagogique.

L'expérience montre que les sujets transversaux associant aspects matériels et logiciels sur un même système suscitent le plus grand intérêt et ouvre la voie à des perspectives nouvelles.

RÉFÉRENCES

- [1] Béatrice Pradarelli, Production Testing Labs, CNFM Montpellier, LIRMM.
- [2] Fahmi Ghazzi, Optimisation d'une Bibliothèque de Modules Matériels de Traitement d'Images. Conception et Test VHDL, Implementation sous forme FPGA