

Conception de système embarqué sur cible FPGA : une approche par compétences

V. Frick^a, B. Boyer^b

^a IUT de Haguenau, Université de Strasbourg, Haguenau, France

^b INSA, Strasbourg, France

Contact email : vincent.frick@unistra.fr

Cet article témoigne d'une expérience d'approche par compétence appliquée dans le cadre de modules d'enseignement aux systèmes électroniques embarqués. Les projets proposés aux étudiants visent à développer les compétences qui leur permettront de répondre efficacement à un cahier des charges dans un domaine où matériels et logiciels sont en constante évolution. En l'occurrence, il s'agit d'encourager l'initiative dans les choix techniques de co-conception de circuits numériques impliquant le langage de description matériel VHDL, la synthèse de processeur embarqué, la programmation en langage C. Le retour d'expérience montre qu'au prix d'un suivi régulier de la progression des étudiants, les résultats et le taux de satisfaction des étudiants sont très élevés et peuvent même dépasser les objectifs initiaux.

I. Introduction

Les systèmes électroniques embarqués sont aujourd'hui omniprésents dans tous les domaines et à toutes les échelles. L'offre de solutions techniques de développement est conséquente et en perpétuel renouvellement, tant sur le plan matériel (microcontrôleurs, FPGA, interfaces, périphériques, etc.) que logiciel (langages, outils CAO, etc.). En l'occurrence, cette diversité démultiplie le nombre de possibilités de co-conception permettant de répondre efficacement à un cahier des charges donné. Les futures générations de concepteurs doivent acquérir la capacité de s'adapter régulièrement à ces évolutions. Or, « *la simple transmission des savoirs est devenue impossible avec la multiplication des connaissances* » (1) relatives aux systèmes embarqués. En outre, les profils d'étudiants tendent à couvrir des spectres plus larges, aux parcours et affinités différentes. Ceci se traduit donc par la nécessité d'un changement de paradigme dans l'enseignement de cette discipline.

Cet article présente une expérience d'approche par compétence mise en œuvre dans le cadre d'un module en BUT à l'IUT de Haguenau et d'un module en formation d'ingénieurs à l'INSA de Strasbourg. Le choix du type de production (solution technique, outil, langage, etc.) est l'essence même de l'apprentissage par projet, qui permet à l'étudiant de passer le seuil au-delà duquel la connaissance est pérennisée en compétence. C'est donc cette forme d'enseignement qui a été adoptée. Les projets proposés aux étudiants reposent sur une base technologique commune, évolutive et ludique, adaptable aux niveaux licence (L2-L3) ou master/ingénieur (M2). Les cahiers des charges de ces projets sont détaillés dans la section II, la section III est dédiée aux aspects pédagogiques et la section IV conclut cet article.

II. Systèmes « niveaux à bulle »

Les projets proposés reposent sur l'utilisation de cartes de développement Terasic® DE10-Lite (2) ou DE1-SoC (3) adossées à la suite d'outils CAO : Quartus, ModelSim et Eclipse (4). Ces cartes comportent des circuits FPGA dont le nombre d'éléments logiques est suffisant pour héberger au moins un processeur couplé à des fonctions numériques conçues sur mesure. Elles servent donc de cibles matérielles pour l'implantation des systèmes développés par les étudiants. En outre, elles disposent toutes les deux d'un accéléromètre de type ADXL345 (5), appelé G-sensor. C'est autour de ce composant que sont construits les deux systèmes embarqués de base, l'un pour la carte DE10-Lite, l'autre pour la carte DE1-SoC, constituant les points de départ des projets. En fonction du cahier des charges et de la carte de développement, qui lui seront soumis, l'étudiant fera évoluer le système avec un fort degré de flexibilité selon les solutions techniques qu'il aura choisies pour accomplir ses missions.

Deux exemples de cahier de charges sont présentés ci-dessous. Le premier s'adresse aux étudiants de niveau L2-L3 à l'IUT de Haguenau, le second est proposé aux élèves ingénieurs de niveau M2 à l'INSA de Strasbourg.

Projet « niveau à bulle » électronique en GEII

Ce projet s'adresse à tous les étudiants en GEII¹, qu'ils soient en formation initiale ou en alternance.

Le sujet porte sur le développement d'un système de type « niveau à bulle » électronique « intelligent » qui doit satisfaire aux contraintes suivantes :

- il doit être implanté sur la carte DE10-Lite,
- il doit comporter un processeur embarqué de type « softcore » NIOS II® chargé d'exécuter un programme développé en langage C,
- il doit comporter une machine d'état décrite soit en langage de description matériel VHDL, soit en langage informatique C.

Indépendamment de ces trois contraintes, les étudiants ont une grande latitude quant au choix des solutions techniques (architectures, méthode de design, outils, etc.) apportées pour répondre au cahier des charges des fonctionnalités du système. Ils sont néanmoins guidés et conseillés dans le choix de la méthodologie de conception et des outils de développement (cf. section III). Par ailleurs, afin d'éviter les écueils et permettre la réalisation du projet dans le temps imparti au module MCFPGA (50 h), une base de système est fournie aux étudiants. Cette base comporte tous les fichiers (HDL, symbole, etc.) du bloc de contrôle (driver) du G-sensor.

Le cahier de charges des fonctionnalités est évolutif. Les nouvelles fonctionnalités sont proposées au fur et à mesure de la progression des séances et doivent s'ajouter aux versions précédentes. En **première version**, le système doit donner une information lumineuse de type « bargraphe » sur l'inclinaison de la carte de développement au moyen des leds qui s'y trouvent. Simultanément, le degré d'inclinaison doit être affiché de manière numérique sur la console du PC relié à la carte de développement. La figure Fig.1 montre le schéma synoptique et le fonctionnement de base du système.

¹ Jusqu'ici placé au 4^{ème} semestre (2^{ème} année) dans le cadre du module complémentaire « MCFPGA » (50 h) du DUT GEII à l'IUT de Haguenau, ce projet sera dorénavant proposé au parcours « Électronique et Systèmes Embarqués » au 5^{ème} semestre (3^{ème} année) de la nouvelle formation au BUT GEII.

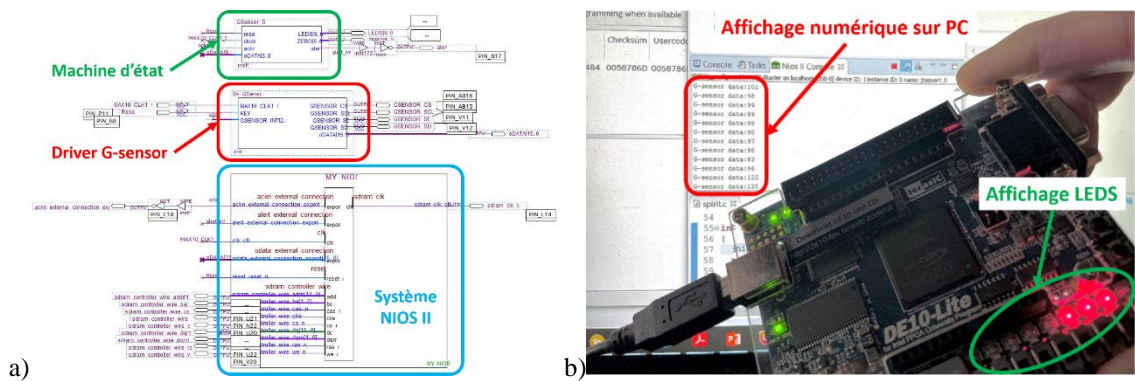


Fig.1. a) Exemple de schéma Quartus du système « niveau à bulle » électronique, b) Photo du système en fonctionnement : les données issues du G-sensor sont simultanément affichées sous forme de bargraphe de leds et en valeurs numériques dans la console PC (Eclipse) liée au processeur embarqué NIOS II.

Le système comporte un processeur NIOS II permettant d'exécuter un programme, développé en langage C à l'aide du logiciel Eclipse. Le rôle du programme est d'une part de lire et convertir les données envoyées par le G-sensor pour afficher l'angle d'inclinaison de la carte de développement dans la console du PC. D'autre part, il doit gérer une interruption déclenchée par la machine d'état en cas de butée du G-sensor aux valeurs extrêmes (trop incliné d'un côté ou de l'autre). Le programme d'interruption doit déclencher l'affichage d'un message, par exemple « Alerte butée droite » ou « Alerte butée gauche », selon le cas correspondant et doit mettre le système en attente d'un acquittement. Cet acquittement est effectué en appuyant sur une touche du clavier. Au niveau interne, il se traduit par l'envoi, depuis le processeur NIOS II, d'un signal de déblocage vers la machine d'état.

Le processeur NIOS II est synthétisé grâce à l'outil Platform Designer de Quartus (Fig.2). Cet outil permet de construire des systèmes sur puce à partir de bibliothèques de blocs IP tels que le cœur de processeur NIOS II lui-même, des ports microcontrôleurs, nécessaires par exemple pour la transmission de signaux entre le processeur et la machine d'état ou les leds, ou encore des protocoles de communication (USB ou RS232).

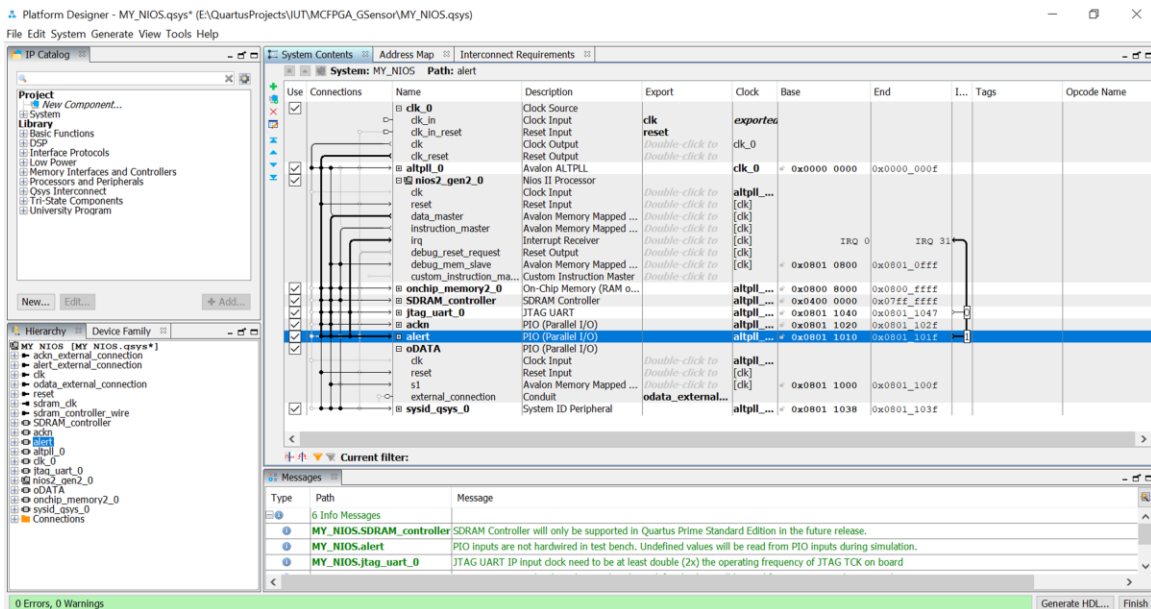


Fig.2. Exemple de système NIOS II réalisé avec l'outil Platform Designer.

Le rôle de la machine d'état est d'une part de gérer l'affichage des leds qui doivent s'allumer de manière « bargraphe » en fonction des données du G-sensor, avec 5 leds pour l'inclinaison à droite et 5 leds pour l'inclinaison à gauche. Quand le G-sensor est à l'équilibre (carte de développement à plat), toutes les leds doivent être éteintes et un afficheur 7 segments affiche « 0 ». En revanche, quand le G-sensor arrive en butée, toutes les leds du côté concerné doivent rester allumées, même si on remet la carte de développement à plat. La machine d'état ne se débloque que par le signal d'acquittement du processeur ou par un reset complet du système.

La méthode de réalisation de cette machine d'état diffère en fonction du parcours des étudiants (formation initiale ou alternance). En effet, le programme du module MCFPGA destiné aux étudiants en formation initiale comporte une initiation au langage de description matériel VHDL. Aussi, il leur est demandé de concevoir la machine d'état ainsi que son banc de test dans ce langage. Le banc de test est alors simulé à l'aide de ModelSim, comme illustré dans Fig.3. Concernant les étudiants en formation par alternance, ils programment la machine d'état en langage C pour le processeur NIOS II.

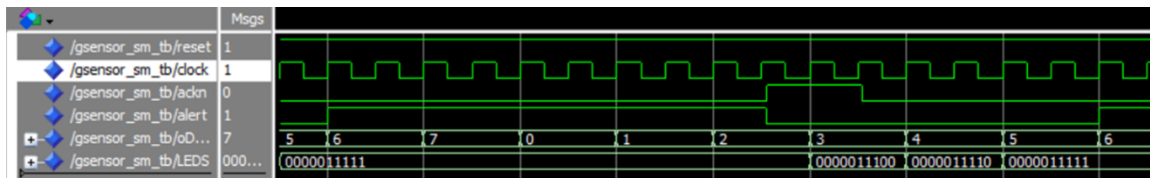


Fig.3. Exemple de simulation de la machine d'état, réalisée avec ModelSim. Un signal d'alerte (« alert ») est levé lorsque la sortie du G-sensor « oD » atteint une limite (ici fixée arbitrairement à 6). L'affichage des leds en mode bargraphe reste figé jusqu'à acquittement de l'alerte par le signal « ackn ».

Lorsque la **première version** du système est achevée et validée (cf. section III, « Déroulement des séances ») le système peut évoluer vers la **deuxième version**. Le cahier des charges porte sur l'ajout d'une fonction qui, lorsque le système arrive en butée, génère un nombre aléatoire « vrai » compris entre 0 et 15. Pour pouvoir débloquer le système, l'utilisateur doit alors trouver la combinaison binaire correspondant à ce nombre à l'aide des interrupteurs de la carte. Une fois une combinaison sélectionnée, elle est validée par un appui sur un bouton poussoir. Le système affiche un message d'erreur sur la console du PC tant que la combinaison sélectionnée est différente du nombre aléatoire généré par le programme. Lorsque la bonne combinaison est trouvée, le système se débloque et le programme reprend son cours normal.

La **troisième version** reprend les éléments des versions précédentes, mais lorsque le système arrive en butée et génère un nombre aléatoire, une temporisation est également déclenchée. L'utilisateur dispose alors un temps limité (par exemple 1 minute) pour trouver la bonne combinaison. Au-delà, une nouvelle combinaison est générée et la temporisation redémarre. Ainsi, tant que le système n'est pas débloqué, un nouveau nombre aléatoire est généré chaque fois que le temps limite est atteint. Une solution possible, mais non exclusive, consiste par exemple à modifier le système NIOS II pour y rajouter un module « interval timer » et développer le programme permettant de gérer la temporisation.

La **quatrième version** est proposée en option aux étudiants souhaitant approfondir leurs compétences à la fin du module MCFPGA. Elle consiste à rendre la durée de temporisation reconfigurable. Ainsi l'utilisateur peut choisir le temps de renouvellement du nombre aléatoire au lancement de l'application. En outre, pour les étudiants en formation initiales, le « timer » programmable doit être indépendant du NIOS II® et développé en langage VHDL.

Projet « niveau à bulle » acoustique

Ce projet est proposé dans le cadre du module « CAO microélectronique » aux élèves ingénieurs de 5^{ème} année suivant l'option Systèmes embarqués & IoT à l'INSA de Strasbourg.

Outre le G-sensor, la carte DE1-SoC, utilisée comme cible matérielle pour implanter le système développé, comporte un CODEC audio haute-fidélité de type WM8731 (6). Le but du projet est de développer un « niveau à bulle » acoustique dont le principe repose sur la variation de la hauteur (fréquence) d'un signal acoustique en fonction de l'inclinaison de la carte.

Le CODEC WM8731 est un composant programmable dont la configuration (mode acquisition/enregistrement ou émission, réglage des volumes, fréquence d'échantillonnage, gestion de l'alimentation, etc.) est gérée par le par le protocole I²C (cf. Fig.4).

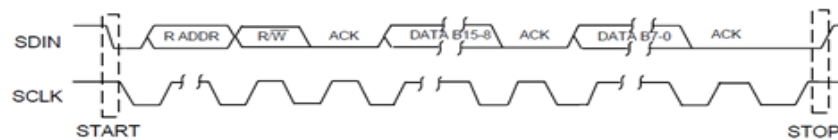


Fig.4. Protocole standard I²C.

Les données audio-numériques sont quant à elles véhiculées selon le protocole I²S, protocole standard des équipements multimédias et hi-fi (cf. Fig.5). Elles sont codées sur 16 ou 32 bits selon la fréquence d'échantillonnage choisie (respectivement 96 kHz ou 48 kHz). Le signal issu du convertisseur numérique/analogique du CODEC est diffusé sur la sortie « casque audio » composée d'un connecteur de type « mini-jack ».

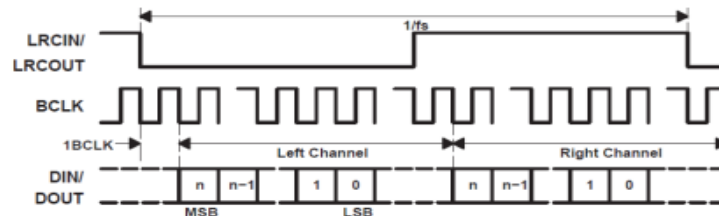


Fig.5. Protocole audio standard I²S (Philips).

Le CODEC est connecté à la partie « FPGA fabric » du circuit Cyclone V équipant la carte DE1-SoC. En revanche, l'accéléromètre (G-sensor, ADXL345) est interfacé à la partie HPS² de ce circuit. Le HPS exécute un système d'exploitation de type « Linux embarqué » (distribution Ubuntu). Les données issues du G-sensor sont envoyées en continu sur le bus « axi master » faisant l'interface entre la partie HPS et la partie FPGA du Cyclone V (3). Le cahier des charges du projet implique la réalisation d'un bloc de contrôle du CODEC implanté dans la partie FPGA. Le rôle de ce contrôleur est de lire les données du G-sensor (« Octaves »), de gérer un affichage sur leds de type bargraphe émulant le comportement d'un niveau à bulle, et de contrôler intégralement (configuration et données audio) le CODEC. Le schéma synoptique du système est présenté ci-dessous (cf. Fig.6).

² HPS : Hard Processor System, processeur ARM Cortex embarqué dans le Cyclone V.

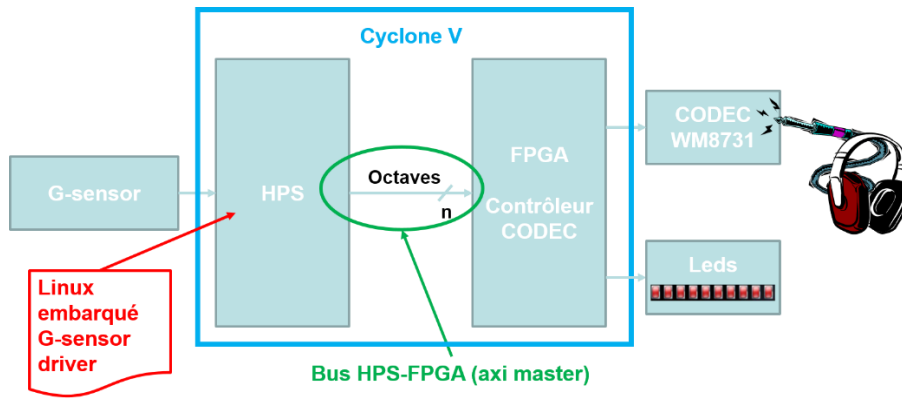


Fig.6. Schéma synoptique du système « niveau à bulle » audio.

Le programme du module de CAO microélectronique (30 h) en 5^{ème} année à l'INSA de Strasbourg porte essentiellement sur la conception en langage VHDL. Il comporte aussi une initiation à Platform Designer, l'outil de synthèse de processeur embarqué NIOS II. Ainsi les étudiants peuvent choisir de développer le contrôleur du CODEC exclusivement en VHDL ou de combiner VHDL et NIOS II programmé en langage C. Il ne leur est néanmoins pas possible de développer l'intégralité du système sur NIOS II car la réalisation de deux blocs VHDL leur est imposée. En l'occurrence, il s'agit de deux ROM (« config » et « audio »), l'une contenant les données de configuration I²C du CODEC (mode écriture, réglage des volumes, acquisition de sortie, gestion de l'alimentation du CODEC, etc.), et l'autre contenant les échantillons d'une période d'un signal sinusoïdal. Ces échantillons, stockés sous forme d'entiers (signés) codés sur 32 bits servent à générer les données audio du CODEC. La figure 7 montre un exemple de structure du contrôleur de CODEC contenant les deux ROM ainsi qu'un bloc « Séquenceur » unique chargé de gérer à la fois la configuration du CODEC et les données audio. La gestion des protocoles I²C, I²S et le traitement des données « Octaves » du G-sensor peut être réalisée soit par des blocs développés en VHDL soit au moyen d'un processeur NIOS II comportant des ports microcontrôleur dédiés.

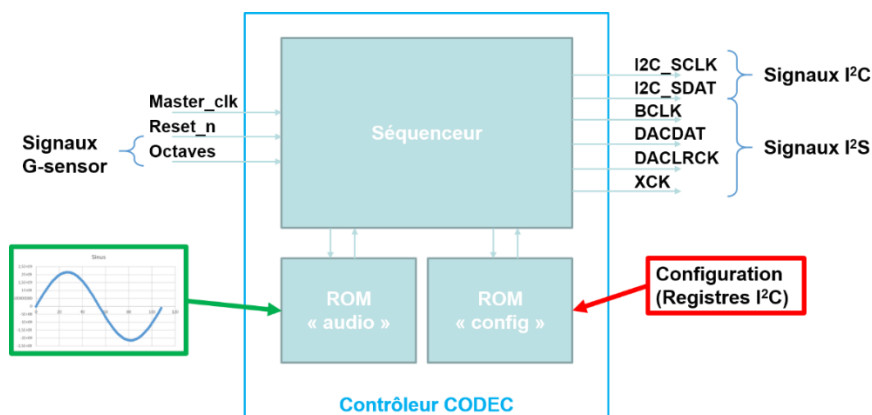


Fig.7. Exemple de structure du CODEC audio. Les ROM « audio » et « config » sont développées en VHDL tandis que le séquenceur est développé soit en VHDL, soit programmé sur NIOS II. Notons qu'un bloc « Séquenceur » unique n'est pas forcément nécessaire, les deux protocoles I²C et I²S étant indépendants.

D'un point de vue des fonctionnalités, le signal audio doit correspondre à la note « La » à 440 Hz lorsque la carte est à l'horizontale et monter ou descendre d'une octave selon

qu'elle soit respectivement en butée droite ou gauche (valeurs d'inclinaison extrêmes du G-sensor). La figure ci-dessous montre un exemple de simulation de la ROM « audio » développée en langage VHDL.

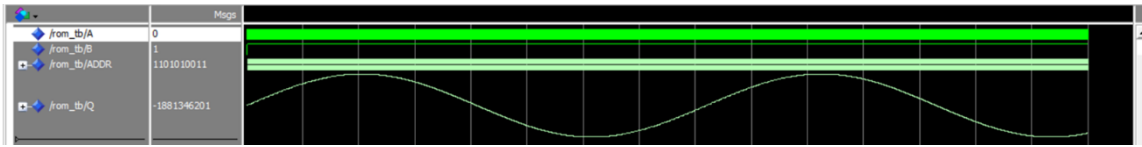


Fig.8. Simulation de lecture de la ROM « audio » développée en langage VHDL, réalisée à l'aide du logiciel ModelSim. La ROM « audio » contient 1024 échantillons d'une période d'un signal sinusoïdal codé sur 32 bits.

III. Aspects pédagogiques

Le concept d'approche par compétence implique une adaptation des méthodes d'enseignement, tant dans la transmission des ressources que dans l'évaluation. Une compétence est validée dès lors qu'un l'étudiant est capable de répondre à un cahier des charges en s'appuyant sur les apprentissages critiques de la formation. Les projets de systèmes embarqués sont particulièrement adaptés aux Situations d'Apprentissage et d'Évaluation (SAé), qui sont notamment mises en place dans le cadre du nouveau BUT. Ces SAé doivent permettre l'acquisition et l'évaluation des compétences visées dans des situations professionnelles mettant en œuvre les ressources (connaissances et matériels) dont dispose l'étudiant.

Déroulement des séances

Au niveau L2-L3 : Le module MCFPGA proposé en deuxième année de BUT GEII représente un volume horaire total de 50 heures réparties en cours (2×2 h), travaux dirigés (3×2 h), travaux pratiques (7×4 h) et séances de projet non-encadrées (3×4 h). Pour les étudiants en alternance, il n'y a pas de séance de projet non-encadré³ et donc le volume horaire n'est que de 38 heures. Le découpage répond aux exigences du module, dans lequel la pratique est fondamentale et omniprésente. En effet, toutes les séances, cours y compris, ont lieu en salle de travaux pratiques (salle informatique). Les deux séances de cours (4 h), placées en début de module, sont dédiées à l'introduction aux concepts de systèmes embarqués, au langage VHDL et à la présentation des outils de développement, essentiellement ModelSim, Platform Designer et Eclipse car les étudiants ont été formés à l'utilisation de Quartus dès la 1^{ère} année. Elles se déroulent sur le principe du cours intégré. Les connaissances introduites sont simultanément illustrées par des exemples pratiques simples (exemple : modèles VHDL simples, simulations et implantation sur carte).

Les séances de TD, qui surviennent immédiatement après les cours, sont consacrées à la prise en main des nouveaux outils ainsi qu'à la préparation du projet. Les étudiants sont amenés à construire une base saine et opérationnelle du système embarqué. En l'occurrence, le bloc de contrôle du G-sensor (driver I²C) leur est fourni car son développement requerrait un volume horaire trop important et empêcherait d'atteindre les objectifs visés. Ces séances sont fortement guidées, avec des instructions claires. En particulier, les étapes de création de la couche d'abstraction matérielle du NIOS II sont décomposées pas à pas.

³ Les projets sont réalisés en entreprise.

A ce stade, l'accent est particulièrement mis sur la nécessité de rigueur dans gestion des fichiers, qui est très souvent source de difficultés chez les étudiants.

La première séance de TP est consacrée à la présentation détaillée du cahier des charges. Les autres séances de TP sont dédiées au développement des différentes versions du système. Elles sont entrecoupées de séances non-encadrées pendant lesquelles les étudiants poursuivent leur travail en autonomie. Cette autonomie est facilitée par le fait que chaque étudiant dispose de sa carte de développement, prêtée à l'année.

Au niveau M2 : Le module de CAO microélectronique proposé en 5^{ème} année à l'INSA de Strasbourg représente un volume horaire total de 30 heures réparties en cours (4×3 h) et séances de projet encadrées (6×3 h). A l'instar du BUT, toutes les séances, cours y compris, se déroulent en salle de travaux pratiques (salle informatique). Les cours sont essentiellement dédiés à la conception d'entités VHDL et à leur simulation. L'accent est particulièrement mis sur les notions d'entités paramétrables, de réutilisabilité et de hiérarchisation des architectures. Ce dernier point est particulièrement important car les étudiants sont amenés à travailler en binôme et doivent donc être capable d'organiser leur architecture système pour pouvoir en développer indépendamment chaque partie.

Par manque de temps, le volume horaire du module de CAO microélectronique étant seulement de 30 heures, il ne leur est malheureusement pas possible d'exploiter pleinement la partie HPS. En l'occurrence, afin que les étudiants puissent achever leur projet dans le temps imparti, une base de système leur est fournie. Cette base comporte les fichiers de configuration de la partie HPS ainsi que le système d'exploitation Linux contenant notamment le driver logiciel du G-sensor.

Monitoring, évaluation

En début de module, les étudiants sont informés sur les différents jalons du projet, les livrables, les modalités et critères d'évaluation.

A partir de la troisième séance de TP, des points individuels sont effectués en début de chaque séance. Ces séquences courtes, d'une durée de 2 minutes maximum par étudiant, permettent de mesurer l'état d'avancement du travail, de lever des situations bloquantes et de rectifier les éventuelles dérives. Les similitudes entre les travaux des étudiants sont inévitables dans la mesure où l'entraide est encouragée pour stimuler le travail en équipe, susciter l'émulation et ainsi créer des cercles vertueux. Mais les étudiants doivent impérativement trouver un équilibre entre le travail individuel, qui leur permet de développer leur compétence propre, et travail commun. C'est pourquoi les points individuels visent aussi à dissuader l'exploitation de blocs qui ne sont pas maîtrisés. Cela peut être le cas par exemple lorsqu'un étudiant a récupéré un code (VHD ou C) sur internet ou chez un collègue sans chercher à en comprendre le contenu.

Concernant les étudiants en BUT, plusieurs livrables sont exigés. Ils correspondent aux différentes versions exposées dans la section II. Le cahier des charges de chaque version comporte des objectifs réalisables, inscrits dans un périmètre bien défini et décomposé en une succession de tâches simples. Le fait de mettre plusieurs points de sortie dans le projet permet d'éviter les frustrations, favorise l'implication des étudiants et leur garanti un niveau de compétence minimale à chaque étape. Concernant les étudiants de l'INSA, un seul livrable est demandé en fin de module.

Les compétences sont évaluées au moment de la restitution des livrables. Chaque étudiant présente individuellement son système, expose les choix techniques effectués et fait une démonstration de son fonctionnement. Les critères d'évaluations portent essentiellement sur la maîtrise des outils et la rigueur dans la méthodologie de conception. En particulier, dans le cas de développement de blocs VHDL, des résultats de simulations

sont exigés. Pour les étudiants de l'INSA, la restitution orale est effectuée en binôme et doit être accompagnée d'un dossier de conception rédigé.

Pour les étudiants en BUT, la compétence minimale est jugée acquise lorsque l'étudiant a présenté une version aboutie du système et est capable de proposer une solution technique immédiate pour apporter une modification simple sur son système. Par exemple, lorsque la machine d'état est écrite en VHDL, l'ajout d'un bouton poussoir permet le cas échéant de prévoir un acquittement manuel du signal d'alerte de butée, et ainsi de garantir le fonctionnement du système en cas d'arrêt du NIOS II. Pour les étudiants de l'INSA, les critères comportent également des aspects plus subtils tels que l'élégance des solutions techniques, le caractère paramétrable des architectures, la réutilisabilité des modules ou encore la hiérarchisation du design. Quelle que soit la filière concernée, pour une grande majorité des étudiants, les objectifs fixés sont largement atteints, voire même dépassés.

IV. Conclusion

Les évaluations effectuées sur les modules MCFPGA et CAO microélectronique dans le cadre des démarches qualité⁴ de l'IUT de Haguenau et de l'INSA de Strasbourg ont révélé un très fort taux de satisfaction des étudiants. Les échanges, la liberté des choix techniques, l'aspect ludique du projet, le prêt permanent des ressources matérielles (cartes de développement) sont autant de facteurs propices à l'émulation et à l'investissement des étudiants. Les résultats vont souvent au-delà des attentes et font parfois émerger des idées originales et ludiques qui peuvent être réinjectées dans les projets futurs. Mais c'est sans doute dans le suivi rapproché que réside l'une des clés essentielles du succès de l'approche par compétence. Ceci requière investissement et souplesse d'esprit, suggérant un degré certain d'expérience pédagogique de la part des encadrants.

Remerciements

L'auteur salue l'initiative nationale « Une carte – Un étudiant » menée par le GIP-CNFM, dans le cadre du projet IDEFI-FINMINA (7), qui donne l'opportunité aux établissements de s'équiper d'un grand nombre de cartes de développement à des tarifs préférentiels et contribue surtout à améliorer le confort de travail des étudiants.

Références

1. G. Scallon, *Des savoirs aux compétences, Exploration en évaluation des apprentissages*, De Boeck Supérieur, ISBN 9782804187859.
2. MAX10-based DE10-Lite development board. Website: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1021>
3. Cyclone-based DE1-SoC development board. Website: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836>
4. Quartus Prime. Website: <https://www.intel.fr/content/www/fr/fr/software/programmable/quartus-prime/overview.html>
5. Analog Device ADXL345. Online datasheet : <https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>
6. Cirrus Logic WM8731. Website: <https://www.cirrus.com/products/wm8731/>
7. IDEFI-FINMINA : Initiative d'Excellence - Formation Innovante en Microélectronique et Nanotechnologies, ANR-11-IDFI-0017. Website: <http://www.cnfm.fr/VersionFrancaise/actualites/FINMINA.htm>

⁴ L'IUT de Haguenau est certifié ISO 9001 et label Marianne, l'INSA de Strasbourg est certifié ISO 9001.