

Mise en œuvre d'un testeur numérique dans le cadre de projets d'étudiants

Robin Rolland^a, Michele Portolan^b

^a CIME Nanotech CS 50-257 38016 GRENOBLE CEDEX 1, France

^b Univ. Grenoble Alpes, CNRS, TIMA, F-38000 Grenoble

Contact email robin.rolland@grenoble-inp.fr, michele.portolan@imag.fr

Cette communication décrit la mise en œuvre d'un testeur numérique de composants, dans le cadre de projets d'élèves ingénieurs des écoles Phelma et Ensimag. Cette expérience pédagogique concerne une formation de spécialistes en systèmes et logiciels embarqués sur une durée d'un semestre. Ce projet a largement contribué à la mise en œuvre de séances de travaux pratiques destinés à des élèves apprentis en conception de systèmes électroniques intégrés de Phelma.

Mots clés: Test, DUT, Vecteur de test, ATPG, Numérique, FPGA, IP, VHDL.

1. Introduction

En 2013, l'école d'ingénieurs Phelma (Physique appliquée, Physique, Electronique et Science des matériaux) (1), école du Groupe G-INP a mis en place une filière apprentissage en microélectronique intitulée "CSI: Conception des Systèmes Intégrés". Les thèmes et compétences enseignés sont centrés sur la conception de dispositifs en électronique analogique et numérique. Ce cursus est calqué sur un cursus type école d'ingénieurs en 3 années (2). La formation implique la signature d'un contrat d'apprentissage entre l'élève et une entreprise. La spécificité du cadre de l'apprentissage conduit à une formation alternant enseignement et travail en entreprise.

Le programme pédagogique a été défini pour permettre aux étudiants de mettre œuvre les savoirs acquis à la fin de chaque période scolaire dès leur retour en entreprise. Les nouvelles connaissances théoriques ainsi que les compétences acquises peuvent ainsi être mises en application en entreprise. Cela nous a conduits à mettre en place un projet de conception d'un circuit intégré, en fin de deuxième année. Les circuits sont fabriqués durant l'été, pour être testés par les étudiants en milieu de troisième année.

Nous décrivons comment des enseignements au sein de deux écoles d'ingénieurs ont conduit au développement d'un testeur numérique. Tout d'abord, nous aborderons les thèmes enseignés en conception de circuits analogique/numérique en deuxième année d'apprentissage Phelma en partie 1, puis la mise en œuvre d'un testeur numérique par des élèves de troisième année Ensimag, filière SLE: "systèmes et logiciels embarqués" en partie 2. L'aspect concret et le déroulement du projet, puis du TP de test utilisant ce testeur sur FPGA seront abordés en section 3. Le projet testeur a été réalisé en 2015, et nous aborderons les évolutions prévues concernant le TP utilisant le testeur en section 4. La section 5 fera la synthèse de cette expérience.

2. **Projet “Conception de fonctions intégrées”, apprentissage 2A Phelma:**

Ce projet est une des réalisations phare de la filière apprentissage CSI. Il s'agit d'une architecture complète d'émission-réception selon le standard Zegbee (4). Durant trois à quatre mois, les étudiants travaillent sur cette architecture mise en œuvre en technologie AMS 0.35 μm CMOS. L'étude débute par la définition d'une architecture d'ensemble, les différents blocs analogiques et numériques sont définis et spécifiés. Ensuite les étudiants se répartissent le travail par sous ensemble, en binômes. Ce projet se termine en juin/juillet, et les blocs validés sont envoyés en fabrication au CMP (3), en technologie AMS CMOS 0,35 μm . Cette puce RadioFréquence et numérique sera testée en début de 3^{ème} année l'aide d'un testeur RF dédié pour la partie analogique. Afin de tester la partie RF, l'école Phelma a acquis un testeur RF sous pointes, qui permet de tester les circuits analogiques nus et en boîtier.

Il serait logique d'utiliser un testeur numérique pour un test complet. La différence entre test analogique et numérique réside essentiellement dans l'aspect génération de vecteurs de test (ATPG) et l'application de ces vecteurs. L'intérêt pédagogique d'utiliser un vrai testeur de composants (ATE) n'est pas justifié en regard du coût de celui-ci. Le test de la partie numérique fera appel à une architecture matérielle et logicielle implémentée sur une carte FPGA associée à un processeur, solution nettement plus abordable en terme de coût.

3. **Développement du testeur numérique. Projet “Etude de cas d'implantation de système embarqués”, SLE 3A Phelma-Ensimag:**

La décision prise est donc de réaliser un testeur numérique capable d'appliquer des vecteurs générés par un ATPG afin de tester un circuit intégré numérique. Le développement du testeur numérique sur FPGA est confié un groupe d'élèves 3A de l'option SLE dans le cadre d'un projet étalé sur 4 mois. La réalisation fait appel à une architecture déployée sur une carte FPGA Xilinx zc706(5), équipée d'un circuit FPGA associé à un processeur ARM9 dual core. Cette carte zc706 supporte une carte fille sur son connecteur Fmc. Et c'est sur cette carte fille qu'est installé le circuit à tester, monté en boîtier DIL par les services du CMP. Le principe est de faire appel à la souplesse de la partie logicielle assurée par le processeur ARM pour piloter les tests, et au FPGA Zynq pour l'aspect application des vecteurs et l'interfaçage avec le circuit sous test. Les compétences requises pour ce projet correspondaient parfaitement aux étudiants d'une autre filière de Phelma intitulée SLE (Système et Logiciel Embarqué).

La filière SLE est partagée avec l'école Ensimag (sciences de l'informatique) (6) de G-INP. Ces élèves sont destinés à maîtriser simultanément les aspects logiciels et matériels d'un système embarqué. Ce sujet de “Testeur numérique simplifié sur FPGA” cible pleinement le programme pédagogique de leur projet de 3^{ème} année SLE. Les 3 élèves ingénieurs SLE vont disposer de 4 mois pour définir, spécifier et développer ce testeur numérique simplifié sur FPGA. Une première version sera livrée pour les Travaux Pratiques de test des apprentis CSI3A. La deuxième version améliorée sera livrée par les étudiants dans la foulée.

Les objectifs du projet testeur sur FPGA :

- mise en œuvre d'une IP VHDL pour l'application des vecteurs,
- l'IP doit être configurable au niveau du nombre de broches, de leur direction et de leur position sur le connecteur du circuit sous test (DUT),
- utilisation du processeur ARM pour la partie contrôle du testeur,

- la gestion des vecteurs de tests doit être réalisée en logiciel,
- obtention d'un prototype fonctionnel en Janvier.

En option :

- IP VHDL configurable depuis le logiciel,
- définition des vecteurs depuis un format texte standardisé (STIL ou autre),
- mise en œuvre de la partie logicielle sur une architecture Linux embarquée,
- optimisation du temps de test.

Les étudiants ont proposé en retour cette architecture suivante (figure 1).

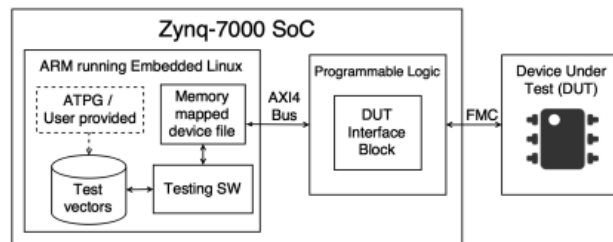


Fig.1 Première architecture du testeur numérique associant dans le même circuit FPGA de la logique programmable et un processeur ARM.

Cette solution est simple et efficace, le circuit DUT sur la droite est piloté par une IP VHDL interface installée dans le FPGA. Le programme de test étant opérationnel sur le processeur ARM (sur la gauche). Cette architecture a été mise en œuvre à l'aide de la suite logicielle Vivado à partir d'une maquette de base matérielle réalisée sans processeur ARM. Ces choix architecturaux et matériels ont permis aux étudiants de respecter les contraintes temporelles du projet et de fournir une plateforme opérationnelle fin janvier, aux caractéristiques suivantes:

- l'IP peut appliquer un vecteur à 1 fois, (temps d'application important : ~35us par vecteur), le processeur doit donc les transférer individuellement,
- IP configurable en VHDL uniquement, si bien qu'un fichier Bitstream doit être généré pour chaque DUT testé,
- le logiciel tourne sur un processeur ARM en mode "bare metal", pas de système d'exploitation,
- les vecteurs sont définis dans le logiciel, il faut recompiler pour toute modification de vecteur.

La présentation d'une première plateforme aux étudiants apprentis de CSI a permis d'affiner les caractéristiques du testeur, et d'envisager une version nettement améliorée, ci-dessous (figure 2).

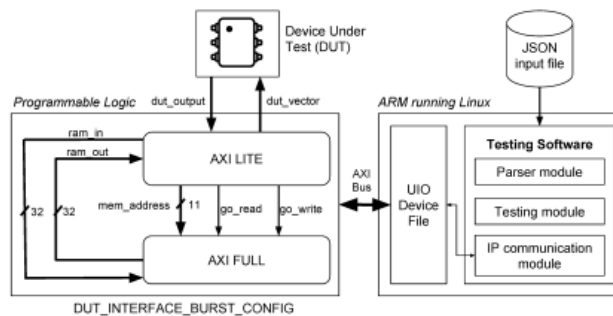


Fig.2 Architecture finale du testeur utilisant une interface AXI et implémentée sous un système Linux.

L'IP VHDL, sur la partie gauche comprend deux parties :

- l'interface DUT, nommée "AXI Lite" qui comprend les registres de contrôle et le contrôle des broches entrées/sorties,
- la mémoire de stockage des vecteurs, nommée "AXI FULL", qui utilise la BRAM FPGA pour stocker les vecteurs,

Le programme de test, dans la partie droite, tourne sur un système d'exploitation "Linux embarqué" (La distribution Peta Linux fournie par Xilinx (7)), qui inclut un "parser" pour traduire les vecteurs définis dans un fichier au format JSON(8). Ce format a été choisi de par sa popularité et la disponibilité de bibliothèques, en contraste avec STIL, qui aurait demandé un certain effort de codage, incompatible avec les délais imposés par ce projet.

Dans sa forme finale, les performances du testeur sont significativement améliorées :

- temps d'application vecteur de 100ns, profondeur de 256 vecteurs, mode Burst du bus AXI4,
- brochage du DUT configurable, toutes les entrées sorties sont connectées à des buffers bidirectionnels contrôlés par un registre interne, comme classiquement pour toutes les interfaces GPIO,
- toutes les données de l'application de test (FPGA bitstream, image boot noyau Linux, et logiciel de test) sont stockées dans la carte SDRAM « bootable » installée dans l'interface du kit FPGA zc706,
- un signal de synchronisation est fourni par l'interface pour faciliter l'analyse à l'oscilloscope.

Nota: La carte zc706 a été choisie afin de permettre l'évolution du testeur vers une version pourvue d'une grande profondeur de vecteurs de test (ATPG ou SCAN), ceci grâce notamment à la taille importante de la BRAM du circuit Zynq xc045.

4. Expérimentation et utilisation du testeur :

Ainsi que nous l'avons mentionné la première mise en œuvre du TP a eu lieu avec la version de testeur simplifié aux fonctionnalités réduites. A cause de problèmes techniques en 2015, aucun bloc digital conçu par les étudiants de CSI 2A n'a pu être envoyé en fabrication. Nous avons cependant décidé de réaliser un TP d'introduction sur le test numérique avec la première version du testeur :

- dans les archives du CIME, nous avons sélectionné 4 circuits différents en boîtier DIL,
- ces circuits ont été conçus en 1994 en technologie CMOS 1,5um, par des étudiants spécialistes en microélectronique d'école d'ingénieurs 3A (circuits nommés W5, Y6, G8 et G9),
- tous les circuits assurent la même fonctionnalité (décodeur BCD-7segments en logique combinatoire), 2 circuits en logiques positive, 2 en logique négative. Brochages et boîtiers spécifiques,
- chaque circuit est disponible en 5 exemplaires. Nous avons à dessein provoqué des pannes (défauts de bonding) sur certaines puces,
- G8 présente une erreur systématique (de conception),
- nous disposons de 4 stations de test, chacune dédiée à un circuit spécifique,
- chaque station (photo ci-dessous) comporte une carte FPGA zc706 associée à une carte mezzanine pour le DUT et un oscilloscope à signaux mixtes,
- chaque plateforme de test est configurée avec un oscilloscope connecté, prêt à l'usage (figure 3).

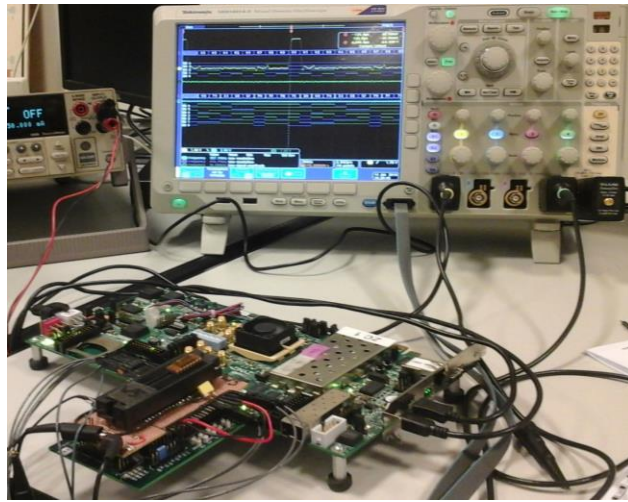


Fig.3 Plateforme de test composée de la carte FPGA, de la carte fille connectée au DUT et de l'oscilloscope à signaux mixtes.

Les étudiants répartis en binômes travaillent successivement sur les 4 plateformes. Sur un des postes, ils définissent le brochage d'un circuit. Pour chaque circuit, ils doivent entrer manuellement les vecteurs de test issus de la table de vérité des décodeurs. Ils lancent le programme de test qui effectue une comparaison des sorties obtenues avec les sorties prévues afin d'identifier les circuits bon ou défectueux (résultats du test fonctionnel en figure 4 ci-dessous).

```
w5 test begin, puce 1.  
OK: for input 0, received expected result 7E.  
OK: for input 1, received expected result 30.  
OK: for input 2, received expected result 6D.  
OK: for input 3, received expected result 79.  
OK: for input 4, received expected result 33.  
OK: for input 5, received expected result 5B.  
OK: for input 6, received expected result 5F.  
OK: for input 7, received expected result 70.  
OK: for input 8, received expected result 7F.  
OK: for input 9, received expected result 7B.  
OK: for input A, received expected result 7F.  
OK: for input B, received expected result 7F.  
OK: for input C, received expected result 7F.  
OK: for input D, received expected result 7F.  
OK: for input E, received expected result 7F.  
OK: for input F, received expected result 7F.  
FINISHED: 16 out of 16 tests were correct.
```

Fig.4 Résultats du test fonctionnel obtenues sur le port série RS232 Remerciements

Les signaux peuvent aussi être observés directement grâce aux connecteurs de la carte mezzanine, qui fournit des points de test pour tous les signaux. Les circuits sont installés et échangés très facilement grâce à un support DIL48 type ZIF, à force d'insertion nulle. Les étudiants peuvent observer les signaux analogiques ou numériques et mesurer les temps de propagation des décodeurs BCD combinatoires (figure 5).

Durant ces travaux pratiques de 2 séances de 4h, les étudiants ont le temps de tester tous les circuits. Pour le premier lot, la configuration matérielle (brochage) leur est fournie, afin qu'ils puissent se familiariser avec l'aspect logiciel de la génération et de l'application de vecteurs. Pour les 3 autres lots de circuits, ils ont également à reconfigurer et régénérer l'IP en fonction du brochage du circuit à tester. L'évaluation des comptes-rendus porte notamment sur les arguments développés lors de l'analyse des dysfonctionnements des circuits.

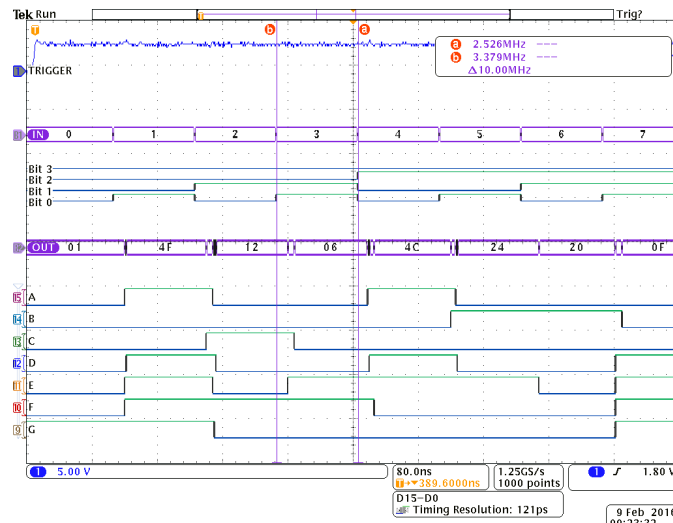


Fig.5 Nous remercions tout particulièrement le Labex Nano-Saclay (8) pour le fort soutien au projet. Nous remercions également le soutien du GIP-CNFM (10) et du projet IDEFI-FINMINA (9) dans le cadre du programme Grands Investissements d’Avenir (PIA1).

5. Evolution des travaux pratiques de test

Demander aux étudiants d’écrire eux-mêmes les vecteurs de test ne présente pas un grand intérêt pédagogique, nous avons donc rajouté une étape à ce TP. Une passerelle logicielle traduit un résultat de simulation au format « LST » généré par Modelsim en format « JSON » destiné au testeur numérique, voir figure 6. Il est également prévu de récupérer les vecteurs issus d’un générateur de vecteur de test (ATPG) pour les traduire au format « JSON ». Cette passerelle est écrite en langage Python et ne compte pas plus de 50 lignes de code. Elle a le mérite d’illustrer auprès des étudiants un exemple de l’efficacité de ce langage de programmation. Il est tout à fait envisageable de proposer aux étudiants de modifier eux-mêmes ce script pour l’adapter au circuit à tester.

La conception puis l’envoi en fabrication n’étant pas un objectif toujours atteint lors des projets CSI 2A, une solution de remplacement consiste à valider l’architecture numérique développée dans les cadre des projets de 2^{ème} année sur un FPGA installé sur une carte Xilinx de type « Breadboard » à base de Artix7, CMOD A7 35T (9), 20800 LUTs, photo en figure 7. Pour rappel un LUT de Xilinx (look up table) génère de 2 à 7 portes équivalentes selon l’architecture implémentée. Ce module remplacera dans ce cas le circuit à tester (DUT) pour les architectures de taille moyenne.

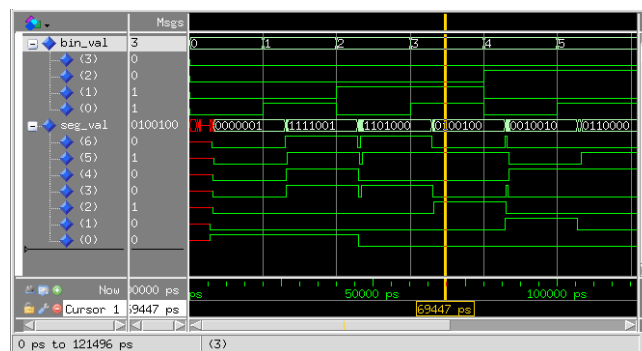


Fig.6 Simulation temporelle du décodeur, obtenue avec Modelsim

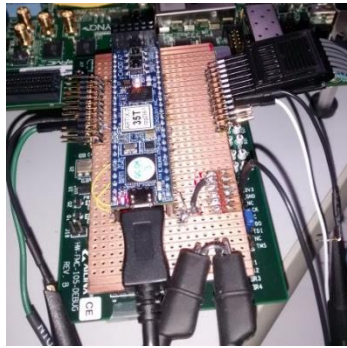


Fig.7 Module Xilinx FPGA Artix A7-35T, en format très compact, DIL 48

Nous disposons également d'un type de circuit DUT beaucoup plus modeste avec la carte à base de CPLD Xilinx CoolRunner CMOD C2 (10) 64 macrocells. Celle-ci est dévolue à l'analyse de défaillance, ceci en programmant des variations autour d'une même architecture numérique de base.

6. Conclusion

Le bilan de cette expérience de développement d'un testeur numérique simplifié à base de FPGA est très positif. Le projet a abouti grâce à une collaboration transversale entre deux domaines de compétences. D'un côté, un besoin en test pour la formation apprentissage CSI de Phelma, de l'autre, la mise en œuvre d'un projet embarqué sur une carte FPGA, pour des étudiants de filière SLE 3A de l'Ensimag. Ce type d'expérience prend tout son sens en répondant pleinement à la formation des élèves.

7. Remerciements

Cette réalisation a bénéficié du financement dans le cadre du Programme d'investissement d'avenir, IRT Nanoelec'ANR-10-AIRT-05', et du support complémentaire du GIP-CNFM et du projet IDEFI-FINMINA ANR-11-IDFI-0017. Les auteurs remercient pour leur travail et leur motivation sans faille les trois étudiants concepteurs et développeurs du testeur numérique : Felipe Dienstmann, João Redro Meireles et Matheus Schuh.

7- Références :

1. Phelma webpage, website: <http://phelma.grenoble-inp.fr/>
2. Conférence des Grandes Ecoles, website: <http://www.cge.asso.fr/en>
3. Zigbee Alliance Homepage, website: <http://www.zigbee.org/>
4. Multi-Projet Circuits (CMP), website: <http://cmp.imag.fr/>
5. Xilinx ZC706 product page, website: <https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html>
6. Ensimag Homepage, website: <http://ensimag.grenoble-inp.fr/welcome/>
8. Petalinux tools, website: <http://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html>
9. JSON homepage, website: <http://www.json.org/>
10. Xilinx Artix 7 breadboard, website: <http://store.digilentinc.com/cmod-a7-breadboardable-artix-7-fpga-module/>
11. Xilinx Cool-runner breadboard, website: <http://store.digilentinc.com/cmod-breadboardable-coolrunner-ii-cpld-module/>
- 12.
- 13.
- 14.