

Proposition de stage

Génération automatique et certifiée de code embarqué pour les dispositifs médicaux implantables

Sujet

Dans le cadre de l'assistance, voire de la suppléance, aux déficiences fonctionnelles humaines par stimulation électrique fonctionnelle, l'équipe Inria CAMIN a conçu une nouvelle génération de neuroprothèses. Cette nouvelle génération de dispositifs médicaux implantables est basée sur la décentralisation de la stimulation neurale au plus près des électrodes. Dès lors, un implant embarque l'étage de génération du courant de stimulation ainsi que toute l'intelligence lui permettant d'en contrôler et surveiller l'exécution, en communiquant à travers une architecture distribuée. Dans des considérations de conception rigoureuse et de fiabilité de fonctionnement, la méthodologie outillée HILECOP a été mise au point par CAMIN [3]. Elle consiste à concevoir l'architecture numérique complexe et critique d'un dispositif médical implantable à partir d'une approche à composants, dont le comportement et l'assemblage est spécifié à l'aide des réseaux de Petri. Ce formalisme permet de réaliser une analyse formelle (logique et temporelle) du comportement de l'architecture résultante. Une fois l'architecture numérique validée, le logiciel HILECOP permet la génération automatique du circuit intégré dans les dispositifs implantables.

L'outil logiciel, et au delà la méthodologie qu'il supporte, est alors crucial dans le processus de création de cette nouvelle génération de neuroprothèses. Cette préoccupation est d'autant plus essentielle que lesdits travaux ont été transférés à la startup Inria Neurinnov qui vise à produire et commercialiser ces implants innovants; en effet, l'accès au marché impose de procéder à la certification (marquage CE) des dispositifs et donc nécessité de qualifier l'outil logiciel exploité pour leur création, à savoir le logiciel HILECOP.

Dans l'objectif de garantir la sûreté de fonctionnement du logiciel HILECOP et par là même de faciliter sa certification, un travail de vérification formelle du logiciel [2, 1] a été entamé depuis plus de 2 ans. Cette vérification formelle consiste à produire une preuve formelle d'équivalence sémantique entre la modélisation à base de réseaux de Petri et le code VHDL produit par la transformation mentionnée ci-dessus (à la manière de ce qui

a été fait dans [4] pour la certification d'un compilateur C). Cette modélisation formelle est mécanisée en utilisant l'outil d'aide à la preuve Coq [5]. Les sémantiques des deux langages cibles, à savoir le langage de modélisation de HILECOP (à base de réseaux de Petri) et le langage VHDL, ont été formalisées en Coq, ainsi que la transformation entre les deux langages. Actuellement, les travaux se focalisent sur la preuve que cette fonction de transformation préserve la sémantique, c'est-à-dire qu'un programme HILECOP s'exécute de la même manière que le programme VHDL traduit par HILECOP.

L'objectif de ce stage est de venir soutenir l'effort de preuve dans la démonstration de préservation sémantique de la fonction de génération automatique de code de HILECOP.

Compétences requises

Le candidat doit connaître la notion de preuve et avoir vu au moins un système de preuve formelle. Avoir eu une première expérience d'un outil d'aide à la preuve peut être un plus.

Remarques additionnelles

Le stage s'effectuera au sein de l'équipe MaREL du LIRMM de l'Université de Montpellier (UM), en présentiel ou en distanciel suivant les conditions sanitaires au moment du stage. L'encadrement sera réalisé par :

- Vincent Iampietro (UM, LIRMM, MaREL, Vincent.Iampietro@lirmm.fr);
- David Delahaye (UM, LIRMM, MaREL, David.Delahaye@lirmm.fr);
- David Andreu (Neurinnov, David.Andreu@lirmm.fr).

Références

- [1] V. Iampietro, D. Andreu, and D. Delahaye. Toward the Formal Verification of HILECOP : Formalization and Implementation of Synchronously Executed Petri Nets. Research report, LIRMM, Université de Montpellier, May 2020. <https://hal-lirmm.ccsd.cnrs.fr/lirmm-02611153/>.
- [2] V. Iampietro, D. Andreu, and D. Delahaye. Vers la vérification d'une méthodologie pour la conception de circuits numériques critiques. In *Approches Formelles dans l'Assistance au Développement de Logiciels (AFADL)*, pages 43–46. IRIT, June 2020. IRIT/RR—2020–05–FR, ISBN 978-2-917490-30-3.
- [3] H. Leroux, D. Andreu, and K. Godary-Dejean. Handling Exceptions in Petri Net-Based Digital Architecture : From Formalism to Implementation on FPGAs. *IEEE Transactions on Industrial Informatics*, 11(4) :897–906, Aug. 2015.
- [4] X. Leroy. A Formally Verified Compiler Back-end. *Journal of Automated Reasoning (JAR)*, 43(4) :363–446, Dec. 2009.
- [5] The Coq Development Team. *Coq, version 8.13.0*. Inria, Jan. 2021. <http://coq.inria.fr/>.