Chapitre 8

Conclusions

Information is not knowledge,
Knowledge is not wisdom,
Wisdom is not truth,
Truth is not beauty,
Beauty is not love,
Love is not music,
Music is THE BEST...

Frank ZAPPA , *Joe's Garage*

LA PRÉSENTE thèse a vu la réalisation physique d'un nouveau circuit reconfigurable, spécialement conçu pour l'implémentation d'applications bio-inspirées basées sur des systèmes cellulaires. Pour ce faire, outre la conception des éléments reconfigurables de base, nous avons défini un algorithme de routage distribué, qui offre des possibilités de création de chemins de données par le circuit lui-même. Nous allons commencer par résumer notre recherche sur le routage, et y proposer de futures explorations. Nous terminerons par l'analyse des caractéristiques du circuit POEtic, et suggérerons de potentielles améliorations, qui pourraient servir à de futurs concepteurs d'un circuit dédié aux mêmes types d'applications.

8.1 Le routage distribué

En première intention, notre algorithme de routage HIDRA fut développé spécifiquement pour le circuit POEtic. Il nous fallait un moyen de permettre aux cellules qui seraient implémentées sur ce substrat reconfigurable de modifier les connexions intercellulaires de manière autonome, ce qui n'est offert dans aucun FPGA commercial. Après nous être penchés sur un système de type anneau à jeton, nous avons opté pour la solution actuelle, qui offre plus de flexibilité.

Des unités de routage, reliées à leurs 4 voisines, ainsi qu'à un élément externe à la grille, forment une grille à deux dimensions. Elles sont composées d'un contrôleur et de cinq multiplexeurs : quatre qui sélectionnent les valeurs à envoyer dans chacune des direction, et un qui sélectionne la valeur à transmettre à l'élément externe. Le contrôleur gère les processus de routage, et est donc en charge de configurer correctement les multiplexeurs. L'algorithme est totalement distribué, et permet à des sources et destinations d'initier de nouvelles connexions à n'importe quel instant. Un identifiant leur permet alors de choisir précisément leur correspondant.

Nous avons décliné notre algorithme en trois autres variantes, HIDRA-RC, HIDRA-RT, et HIDRA-RTC, offrant des optimisations en terme de temps d'exécution du routage ou en terme de risque de congestion. Ce dernier est un point critique de nos implémentations, car aucune technique de rip-up n'a été implémentée, par souci de minimisation de l'espace requis par une unité de routage. Nous avons mené une analyse de cette congestion, afin de pouvoir comparer nos quatre algorithmes, et affirmer qu'il serait préférable de sélectionner HIDRA-RC plutôt que HIDRA, dans le cas où un tel mécanisme devait à nouveau être exploité. HIDRA-RC possède la capacité d'exploiter de manière efficace les chemins de données déjà existants, lorsque plusieurs destinations doivent être reliées à une même source. En minimisant le nombre total de multiplexeurs réquisitionnés, il est alors possible de réduire le problème de congestion.

Nos quatre variantes ont également été implémentées pour des voisinages de 3, 6, et 8, alors que le circuit POEtic est construit sur la base d'un voisinage de 4. Le code VHDL synthétisable les décrivant a servi de base à des simulations permettant de valider les implémentations, puis d'évaluer les algorithmes. Nous nous sommes particulièrement penchés sur la congestion, et avons proposé un estimateur modélisant la probabilité de congestion en fonction du nombre de chemins à créer, pour une taille de tableau, un algorithme, un voisinage, et un nombre de destinations par source donnés. Cette fonction peut donc être exploitée pour prédire si un application particulière nécessitant tel nombre de chemins à créer est implémentable sur une grille de taille donnée.

Nous avons également utilisés ces résultats pour effectuer une analyse visant à comparer les algorithmes et les voisinages. En reprenant les courbes de congestion, et puisque ces courbes sont bijectives, nous pouvons, pour un risque de congestion donné, et pour une configuration donnée (taille, voisinage, algorithme, nombre de destinations par source), évaluer le nombre de destinations correspondant. La comparaison s'est faite sur ces bases, et notre principale conclusion est qu'un voisinage de 8 est environ quatre fois plus efficace qu'un voisinage de 4, en ce qui concerne la congestion, pour un même nombre de transistors utilisés. En effet, une unité de routage à 8 voisines ne nécessite que deux fois plus de logique qu'une à 4 voisines, mais offre 2 plus de possibilités de routage que des unités à 4 voisines groupées par 4.

Une amélioration potentielle, que nous proposerions d'explorer, est le voisinage de 6 couplé à un mécanisme de routage capable de créer un arbre de Steiner de poids minimum entre une source et plusieurs destinations. Comme nous l'avons mentionné en page 69, les membranes d'eau savonneuses s'arrangent toujours en formant des angles de 120 degrés entre trois parois, et créent ainsi une membrane de taille minimale. En reprenant ce principe, il semble prometteur d'exploiter le voisinage de 6, qui offre de tels angles, pour concevoir un nouvel algorithme où les chemins peuvent se modifier lorsque plusieurs destinations sont reliées à une même source, de façon à diminuer le nombre de multiplexeurs réquisitionnés, et par voie de fait, le risque de congestion.

Un des points faibles de nos quatre algorithmes est la présence de liaisons combinatoires traversant le circuit entier, et potentiellement plusieurs circuits, ce qui a pour désagréable conséquence de devoir réduire la fréquence d'horloge pour les applications mettant en jeu une grille de circuits. Nous avons donc développé un cinquième algorithme, HIDRA-L, où chaque unité de routage est une machine de Moore. Toutes les connexions sont absolument locales, et donc, aucune liaison longue distance n'est présente, offrant ainsi une bien meilleure scalabilité. Toutefois, leur réalisation nécessite une quantité de logique environ deux fois supérieure à celle d'une unité de routage d'un des quatre algorithmes précédents. Cette logique supplémentaire est la principale raison qui nous a fait choisir l'algorithme HIDRA, plutôt que HIDRA-L pour la réalisation de POEtic.

La scalabilité de HIDRA-L pourrait finalement être encore améliorée si nous introduisons des techniques asynchrones, ou GALS (Globalement Asynchrone Localement Synchrone), dans nos unités de routage. Dans HIDRA-L, une horloge globale doit être distribuée à toutes les unités de routage, pour les synchroniser. Dans un système multi-circuit, cette synchronisation n'est pas chose aisée, et l'asynchronisme pourrait apporter une stabilité supplémentaire au routage distribué. Le nombre de transistors nécessaires à une telle réalisation risque toutefois fort de dépasser celui de HIDRA-L, qui est déjà conséquent.

Enfin, l'arrivée des nanotechnologies promet de nouveaux défis pour les scientifiques. Nous sommes en droit d'espérer que, dans un futur plutôt proche, nous disposerons de grilles, à deux ou trois dimensions, d'éléments nanoscopiques qui devront être capables de s'auto-configurer. Gageons que le routage de signaux dans de telles structures ne se fera pas sans peine, et peut-être que nos solutions, qui ciblent actuellement les systèmes électroniques, trouveront des applications dans ce nouveau domaine de recherche.

8.2 Conclusions POEtiques

Le premier de nos algorithmes de routage a été spécifiquement conçu dans le but d'être partie intégrante du circuit POEtic. Ce circuit, qui contient un microprocesseur couplé à un tableau d'éléments reconfigurable, le sous-système organique, offre une nouvelle plateforme dédiée à l'implémentation d'applications bio-inspirées mettant en jeu des mécanismes phylogénétiques, ontogénétiques, et/ou épigénétiques. Une de nos contributions fut la conception du sous-système organique, sous forme de fichiers VHDL, écrits en respectant une description directement réalisable en matériel.

Le microprocesseur y offre la possibilité d'exécuter des algorithmes évolutionnistes, et de tester les individus sur le substrat reconfigurable qu'est le sous-système organique. Ce dernier, basé sur des éléments de premier abord semblables aux blocs de base des FPGAs commerciaux de fabricants comme Xilinx ou Altera, offre de nouvelles caractéristiques, qui permettent à nos organismes artificiels d'évoluer, de modifier leur comportement, en se reprogrammant et en changeant la topologie du réseau intercellulaire.

Les éléments clés de la plasticité du vivant sont la capacité des cellules à modifier leur comportement en fonction de leur environnement, ainsi que leur aptitude à créer de nouvelles connexions intercellulaires, notamment dans le cas des cellules neuronales. Notre sous-système organique a donc été conçu en gardant en mémoire ces deux caractéristiques fondamentales du vivant. Nos éléments de base, les molécules, peuvent reconfigurer d'autres molécules, et le routage distribué offre à ces molécules la possibilité de créer des connexions par elles-mêmes, sans qu'une intervention extérieure au sous-système organique ne soit nécessaire.

La reconfiguration partielle offre aux cellules implémentées sur le circuit POEtic la possibilité de modifier leur comportement en changeant les bits de configuration des molécules qui les composent. Cette caractéristique peut notamment être exploitée par les mécanismes de croissance, où une cellule se différencie en chargeant la configuration des molécules de sa partie fonctionnelle en fonction de sa position dans l'organisme. L'autoréparation peut également en tirer parti, en laissant une cellule partiellement endommagée être remplacée par une autre en pleine santé, qui peut être reconfigurée.

Le routage distribué permet aux cellules de créer de nouvelles connexions durant la *vie* de l'organisme. Lors de l'exécution de mécanismes ontogénétiques, une cellule peut se connecter à une cellule totipotente, et lui envoyer des informations, comme sa position, ou le génome décrivant les fonctionnalités de l'organisme. L'organisme peut ainsi croître de manière autonome, sans qu'une intervention extérieure ne soit nécessaire. Les réseaux de neurones peuvent également tirer profit du routage, en initiant de nouvelles connexions par lesquelles la structure du réseau change.

Deux autres caractéristiques du circuit POEtic sont exploitables par les applications de matériel évolutif : l'impossibilité de créer des courts-circuits, et la connaissance des bits de configuration des molécules. Depuis la série XC6200 de Xilinx, aucun circuit reconfigurable n'a offert ces deux caractéristiques, qui sont essentielles pour la réalisation d'évolution matérielle non contrainte. Couplées au microprocesseur, elles font de POEtic une nouvelle plateforme parfaitement adaptée à de tels applications.

Afin de rendre l'exploitation de POEtic possible, nous avons développé un logiciel de conception de designs pour le sous-systèmes organique. Une interface graphique laisse le développeur configurer les molécules, et les connecter entre elles par de simples manipulations. Un outil de simulation, que nous avons également réalisé, lui permet alors de vérifier le fonctionnement de son système, en visualisant l'état des molécules et des unités de routage.

Outre le circuit POEtic, nous avons élaboré différents mécanismes ontogénétiques, qui exploitent ses caractéristiques dans le but d'implémenter des systèmes multicellulaires. L'auto-réplication partielle d'une partie du circuit a été démontrée, de même que la manière d'y réaliser des systèmes tolérants aux pannes, où une cellule endommagée peut recopier une partie de sa fonctionnalité dans une cellule saine, afin que l'organisme entier puisse continuer à effectuer sa tâche. Nous avons également montré que POEtic pouvait être utilisé comme plateforme à du matériel évolutif au niveau des portes logiques.

Le tableau 8.1 compare deux FPGAs commerciaux avec POEtic par rapport à différentes caractéristiques importantes pour l'implémentation d'applications bio-inspirées.

Alors qu'un circuit de test contenant le microprocesseur, 12 molécules et 3 unités de routage a été réalisé et validé, le circuit final vient d'être reçu de la fonderie, mais n'a pas encore pu être testé. Dans un futur proche, nous pourrons vérifier que la réalisation des mécanismes que nous avons présentés fonctionne, et dans le cas où un nouveau circuit du même type devait un jour être conçu, nous pourrions suggérer les améliorations suivantes :

- Le routage intercellulaire pourrait être revu, comme suggéré à la section précédente, afin de réduire le risque de congestion. L'implémentation présente dans le circuit POEtic utilise un voisinage de 4, qui devrait être avantageusement remplacé par un voisinage de 8.

Caractéristique	**Xilinx XC6200**	**Xilinx Virtex II Pro**	**POEtic**
Impossibilité de court-circuit	Oui	Non	Oui
Processeur dans le circuit	Non	Oui	Oui
Processeur accédant les bits de configuration	Non	Non	Oui
Détails des bits de configuration accessibles	Oui	Non	Oui
Routage dynamique	Non	Non	Oui
Auto-configuration	Non	Non	Oui

Tableau 8.1 : *Comparaison des caractéristiques utiles aux systèmes bio-inspirés, entre un XC6200, un Virtex II Pro, et le circuit POEtic.*

- Concernant la scalabilité, et étant donné que la communication inter-circuits passe par le routage distribué, nous ne saurions que conseiller l'évaluation de méthodes asynchrones, qui éviteraient de baisser la fréquence de fonctionnement du système lorsque plusieurs circuits POEtic sont reliés entre eux.
- Au niveau moléculaire, un mécanisme permettant une auto-reconfiguration totale, c'est-à-dire où tous les bits de configuration peuvent être accédés par les molécules, pourrait être mis au point. Une cellule pourrait alors entièrement s'auto-répliquer, sans qu'une partie des molécules ne doive être préparée à recevoir une configuration.
- Il serait également intéressant d'ajouter un mode opératoire moléculaire autorisant l'accès au registre de la molécule à la façon d'une mémoire RAM. L'exploitation du génome par une cellule serait alors simplifié, et ne nécessiterait pas de parcourir l'entièreté du génome par un grand registre à décalage.
- Enfin, toutes ces belles suggestions se devraient de garder à l'esprit la taille des éléments, qu'il s'agisse des molécules ou des unités de routage. Notre circuit n'a pu accueillir que 144 molécules et 36 unités de routage. Il est clair qu'avec les dernières technologies de pointe, et pour un budget plus conséquent, il aurait été possible de disposer d'un tableau reconfigurable nettement plus imposant, mais il faut rester conscient que les améliorations susmentionnées sont autant de facteurs favorisant l'explosion du nombre de transistors nécessaires.

Finalement, l'avènement des nanotechnologies nécessitera, comme nous l'avons mentionné dans la section précédente, des mécanismes d'auto-configuration des systèmes matériels. Les grilles d'éléments nanoscopiques ne pourront pas forcément être totalement configurées de l'extérieur, et une forme d'ontogenèse sera nécessaire à leur bon fonctionnement. Le contenu de cette thèse pourra dès lors apporter une petite pierre à l'édifice des systèmes auto-configurables, et qui sait, peut-être nos enfants utiliseront-ils un jour des nanomachines construites sur la base de quelques éléments de notre travail...