

Un Sistema de Enrutamiento Dinámico para un Circuito Reconfigurable Bio-Inspirado

Yann Thoma^{1,3}, Eduardo Sanchez¹, Juan-Manuel Moreno Arostegui², and Gianluca Tempesti¹

¹ Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Suiza

² Universidad Politécnica de Cataluña (UPC), Barcelona, España

³ Autor responsable. E-mail: yann.thoma@epfl.ch

Abstract. En este artículo presentamos un nuevo algoritmo de enrutamiento dinámico, diseñado especialmente para un nuevo tejido electrónico llamado POEtic. Este circuito reconfigurable tiene como finalidad la simplificación del diseño de sistemas bio-inspirados para aplicaciones celulares (circuitos neuronales artificiales, por ejemplo). El circuito se compone de dos partes: una matriz bidimensional de elementos básicos, similares a las células de base de los circuitos FPGA comerciales, y una matriz bidimensional de unidades de enrutamiento, que implementan un algoritmo de enrutamiento dinámico que permite la creación de caminos de datos entre las células, sin interrumpir el funcionamiento del sistema.

1 Introducción

Los niveles de complejidad y adaptación de la vida son supremamente altos. Después de la fertilización de un óvulo por un espermatozoide, la célula inicial es capaz de dividirse recursivamente hasta crear un ser vivo completo. Luego, durante su vida, un organismo es capaz de auto-repararse en caso de agresiones externas o internas. Los seres vivos poseen además una red neuronal que les permite aprender ciertas tareas y, de esa forma, adaptarse a su entorno. Finalmente, a nivel de la población, la evolución ayuda a sobrevivir en un entorno que se modifica permanentemente. La finalidad del proyecto POEtic [7][8][10] es diseñar un nuevo circuito electrónico inspirado en los anteriores tres ejes de la vida: Filogénesis (evolución) [6], Ontogénesis (desarrollo) [9] y Epigénesis (aprendizaje) [4].

Los métodos ontogénéticos, empleados para desarrollar un circuito auto-reparable, necesitan poder modificar la funcionalidad del circuito sin interrumpir su funcionamiento. Los mecanismos epigenéticos, que utilizan generalmente redes neuronales, podrían necesitar igualmente la creación de nuevas neuronas, así como de nuevas conexiones entre neuronas, sin interrumpir la funcionalidad del circuito. Como los circuitos FPGA comerciales no permiten la auto-configuración dinámica, un nuevo circuito con estas potencialidades es esencial en nuestro proyecto.

En la próxima sección presentaremos la arquitectura general del circuito POEtic, descompuesto en dos subsistemas. En la sección 3 describiremos los elementos básicos del circuito: las moléculas. La sección 4 explica en detalle el algoritmo de enrutamiento dinámico implementado para facilitar la creación de trayectorias de datos de larga distancia dentro del circuito.

2 Arquitectura Estructural

El circuito POEtic se compone de dos partes (figura 1): el subsistema orgánico, que representa la parte funcional del circuito, y el subsistema medioambiental. Las células y, por lo tanto, los organismos se implementan en el subsistema orgánico. Este se compone de una matriz de pequeñas moléculas y de una capa de enrutamiento celular. Las moléculas son las mas pequeñas unidades de hardware programable que puedan ser configuradas por software, mientras que las unidades especializadas del enrutamiento son responsables de la comunicación intercelular. El principal papel del subsistema medioambiental es configurar las moléculas. El es igualmente responsable del proceso evolutivo y puede, de ese hecho, acceder y cambiar el estado de todas las moléculas, con el fin de evaluar el *fitness* del organismo.

Cada célula de un organismo es una colección de moléculas, bloques básicos del circuito. El tamaño y el contenido de cada célula dependen de la aplicación. Por lo tanto, el diseñador tiene que decidir para cada aplicación la configuración de moléculas necesaria para cada célula.

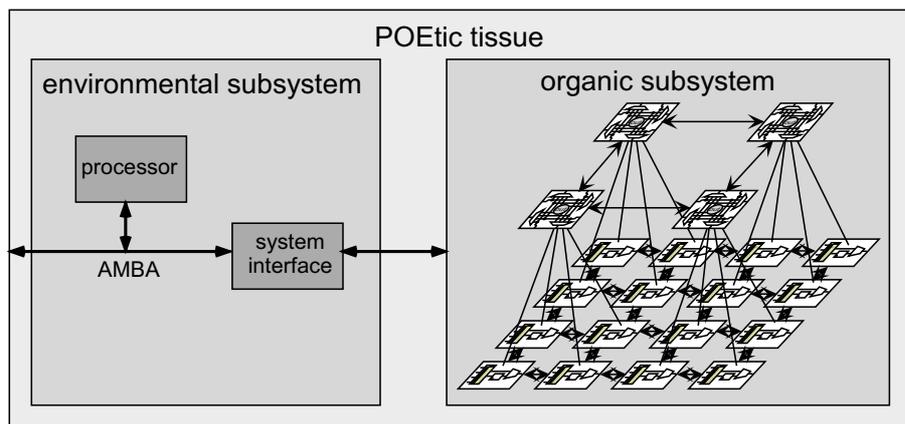


Fig. 1. El circuito POEtic, compuesto de un subsistema medioambiental y un subsistema orgánico.

2.1 Subsistema Medioambiental

El subsistema medioambiental se compone principalmente de un microcontrolador: un procesador 32 bits de tipo RISC. Su función es configurar las moléculas, ejecutar los mecanismos evolutivos y controlar las entradas/salidas del circuito. Para acelerar los procesos de la evolución, un generador de números aleatorios ha sido añadido directamente al hardware. Un bus AMBA [1] se usa para conectar el procesador con la interface sistema encargada de la comunicación entre el

procesador y el subsistema orgánico. Este bus está conectado igualmente a las patas externas del circuito para facilitar la comunicación intercircuitos, así como la utilización de RAM externa.

2.2 Subsistema orgánico

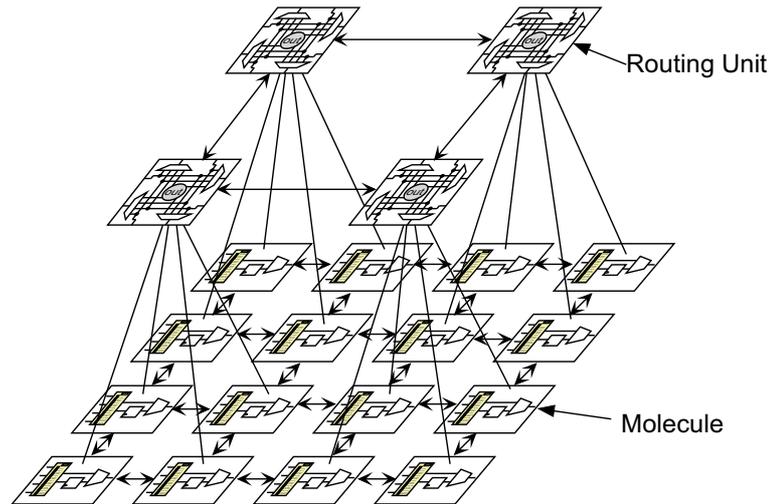


Fig. 2. El subsistema orgánico está compuesto de dos capas: las moléculas y las unidades de enrutamiento.

El subsistema orgánico está hecho con dos capas (cf. figura 2): una matriz bidimensional de elementos lógicos básicos, llamados moléculas, y una matriz bidimensional de unidades de enrutamiento. Cada molécula puede acceder a la capa de enrutamiento, utilizada para comunicaciones intercelulares. Esta segunda capa implementa un algoritmo de enrutamiento dinámico que permite la creación de caminos de datos entre las células, sin necesidad de interrumpir la ejecución de la aplicación.

3 Estructura Molecular

Como se ha presentado brevemente en la sección anterior, la primera capa del tejido POEtic es una matriz bidimensional de pequeñas unidades programables, llamadas moléculas. Cada molécula está conectada a sus 4 vecinas y a una unidad de enrutamiento (cuatro moléculas por unidad de enrutamiento), y posee una memoria look-up table de 16 bits (LUT) y un flip-flop (DFF). Esta estructura, bastante similar a aquella de los circuitos FPGA standard [2], está diseñada sin embargo para las aplicaciones POEtic: diferentes modos de funcionamiento

permiten configurar la molécula como una memoria, un comparador serie de direcciones, una célula de entrada, una célula de salida, etc. Con un total de 8 modos, estas moléculas permiten al diseñador la construcción de células capaces de comunicar entre si, de guardar un genoma, de cicatrizar, etc.

Los ocho modos de operación de las moléculas son los siguientes:

- *Normal*: la LUT es una simple tabla de 16 bits.
- *Aritmético*: la LUT se divide en dos tablas de 8 bits: una para la salida de la molécula y otra para un carry. Una cadena de carry envía físicamente el carry al vecino del sur, acelerando así las operaciones aritméticas.
- *comunicación*: la LUT se divide en un registro de desplazamiento de 8 bits y una tabla de 8 bits. Este modo puede emplearse para implementar un algoritmo de enrutamiento por paquetes que no será presentado en este artículo.
- *Memoria de desplazamiento*: la LUT es tratada como un registro de desplazamiento de 16 bits. Este modo es muy útil para almacenar el genoma en cada célula. Las memorias de desplazamiento pueden conectarse en cadena para crear memorias de mayor tamaño (32 bits, 48, etc).
- *Configuración*: la molécula posee la capacidad de reconfigurar su vecina. Combinada con moléculas configuradas como memorias de desplazamiento, este modo puede emplearse para diferenciar las células. Una parte seleccionada del genoma, almacenada en las moléculas memoria, puede desplazarse para configurar las LUTs de otras moléculas (por ejemplo, para asignar pesos a sinapsis neuronales).
- *Dirección de entrada*: la LUT es un registro de desplazamiento de 16 bits, conectado a la unidad de enrutamiento. Los 16 bits representan la dirección de la célula de donde procede la información. La calidad de la molécula es el valor que llega de la capa de enrutamiento intercelular (este mecanismo sera detallado en la sección siguiente).
- *Dirección de salida*: la LUT es un registro de desplazamiento de 16 bits, conectado a la unidad de enrutamiento. Los 16 bits representan la dirección de la célula; la molécula envía el valor de una de sus entradas a la unidad de enrutamiento (este mecanismo será detallado en la sección siguiente).
- *Disparador*: la LUT es un registro de desplazamiento de 16 bits, conectado a la unidad de enrutamiento. Su tarea es suministrar un disparador cada n ciclos de reloj (donde n es el numero de bits de la dirección). Este disparador es necesario para sincronizar el algoritmo de enrutamiento.

Para tener las potencialidades de auto-reparación y crecimiento, un organismo debe ser capaz de crear nuevas células y configurarlas. El sistema de configuración de las moléculas puede ser visto como un registro de desplazamiento de 80 bits, dividido en 5 bloques: la LUT, la selección de la entrada de la LUT, la unidad de conmutación, el modo de operación y un bloque extra para todos los bits de configuración restantes. Cada bloque contiene, como se muestra en la figura 3, además de su configuración, un bit que indica si el bloque tiene que ser evitado, en los casos en que la configuración proviene de un vecino. Este bit puede ser cargado únicamente desde el microprocesador y permanece estable durante la vida del organismo.

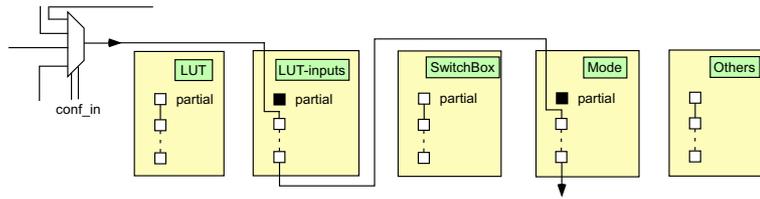


Fig. 3. Todos los bits de configuración de una molécula, divididos en 5 bloques. Los bits de configuración parcial de los bloques 2 y 4 están activos, permitiendo la reconfiguración por una molécula vecina de las entradas de la LUT y del modo de operación.

El modo especial de configuración permite que una molécula reconfigure parcialmente su vecindario, enviando los bits de configuración de una molécula a su vecina. Encadenando la configuración de moléculas vecinas es posible modificar múltiples moléculas al tiempo, permitiendo por ejemplo el cambio de los pesos sinápticos de una neurona.

4 Enrutamiento dinámico

Como ha sido presentado antes, nuestro circuito está compuesto de una matriz de moléculas, dentro de la cual se implementan las células. En un sistema multicelular, las células necesitan comunicarse entre sí: una red neuronal, por ejemplo, presenta a menudo una gran densidad de conexiones entre las neuronas. Los circuitos FPGA comerciales tienen grandes dificultades para manejar este tipo de aplicaciones, a causa de su pobre o inexistente capacidad de enrutamiento dinámico. En vista del objetivo del tejido POEtic, una atención particular ha sido dada a este problema en nuestro proyecto. Es así como una segunda capa ha sido añadida encima de las moléculas, para implementar un algoritmo de enrutamiento dinámico distribuido. Este algoritmo utiliza una versión optimizada del enrutamiento dinámico presentado por Moreno en [5], al cual le hemos añadido un control distribuido, evitando así un control global del proceso de enrutamiento.

4.1 Del Software al Hardware

Nuestro algoritmo de enrutamiento dinámico encuentra el camino más corto entre dos puntos de la capa de enrutamiento. En 1959, Dijkstra propuso un algoritmo software para encontrar el camino más corto entre dos nodos en un grafo en el que toda rama tiene una longitud positiva [3]. Si nosotros fijamos en 1 el peso de todas las ramas, el algoritmo se simplifica drásticamente. En ese caso el algoritmo se convierte en una búsqueda de tipo *breadth-first*, como sigue:

- 1: pintar de blanco todos los vertices;
- 2: pintar la fuente de gris y colocarla en la cola;
- 3: repetir

```

4:   quitar de la cola el vertice v;
5:   si v es el objetivo, hemos encontrado un camino -- salir del algoritmo;
6:   pintar v de negro;
7:   para cada vecino blanco w de v
8:     pintar w de gris;
9:     asignar v como padre de w;
10:    quitar w de la cola
11: hasta que la cola este vacia
12: si no hemos salido todavia, el objetivo no ha sido alcanzado

```

Este algoritmo actúa como un gas expandiéndose en un laberinto, de manera secuencial, un nodo a la vez. Su complejidad es $O(V+E)$, donde V es el número de vértices y E el número de bordes. Sacando ventaja del paralelismo intrínseco del hardware, es posible, basados en los mismos principios de los algoritmos de búsqueda de tipo *breadth-first*, expandir todos los nodos grises al tiempo. Esto disminuye drásticamente el tiempo necesario para encontrar el camino más corto entre dos puntos, la complejidad siendo ahora de $O(N+M)$, para una matriz $N \times M$.

Encontrar el camino más corto no es suficiente para el tejido POETic, puesto que no existe un Dios indicando quien es la unidad de enrutamiento fuente y quien el destino. Para tener un circuito capaz de realizar una auto-configuración de manera autónoma, necesitamos un mecanismo que arranque los enrutamientos. Como se explicó en la sección previa, las moléculas tienen modos especiales de acceso a la capa de enrutamiento: las moléculas de entrada o salida tienen entonces la capacidad de iniciar un enrutamiento dinámico.

4.2 Algoritmo de Enrutamiento

El sistema de enrutamiento dinámico está diseñado para conectar automáticamente las entradas y salidas de las células. Cada salida de una célula tiene un identificador único a nivel del organismo. Para cada una de sus entradas, la célula guarda el identificador de la fuente de información. Una entrada no conectada (destino) o salida (fuente) puede iniciar la creación de un camino, difundiendo su identificador, en el caso de una salida, o el identificador de su fuente, en el caso de una entrada. El camino es creado luego usando una implementación paralela del algoritmo de búsqueda de tipo *breadth-first* presentado anteriormente. Cuando todos los caminos han sido creados, el organismo puede empezar su operación y ejecutar su tarea hasta que un nuevo enrutamiento sea disparado, por ejemplo después de añadir una nueva célula o después de una auto-reparación celular.

Nuestro enfoque tiene muchas ventajas comparado a un proceso de enrutamiento estático. Primero que todo, una implementación hardware de un algoritmo de camino más corto, como el de Dijkstra, consume mucho tiempo cuando se ejecuta en un procesador, mientras que nuestra implementación paralela necesita muy pocos ciclos de reloj para finalizar una trayectoria. Segundo, cuando se crea una nueva célula, esta puede empezar un proceso de enrutamiento sin tener

que recalcular todos los caminos ya creados. Tercero, una célula tiene la posibilidad de recomenzar el proceso de enrutamiento de todo el organismo, si es necesario (por ejemplo, después de una auto-reparación). Finalmente, nuestro enfoque es totalmente distribuido, sin ningún control global sobre el proceso de enrutamiento, de tal manera que el algoritmo puede trabajar sin necesidad del microprocesador central.

El algoritmo de enrutamiento se ejecuta en tres fases:

Fase 1: Encontrando un Maestro

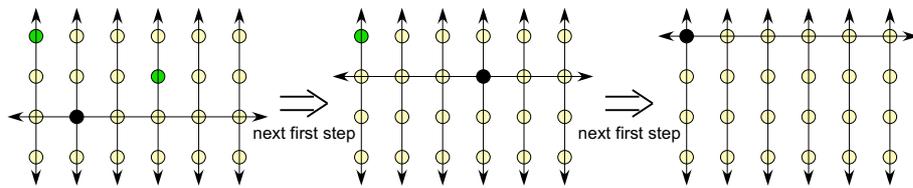


Fig. 4. Los tres primeros pasos consecutivos del algoritmo. La unidad de enrutamiento negra será el maestro, realizando entonces su enrutamiento.

En esta fase, todo destino o fuente que no este conectada a su pareja correspondiente trata de volverse maestro del proceso de enrutamiento. Un simple mecanismo de prioridad convierte en maestro a la unidad de enrutamiento más cercana a la esquina inferior izquierda, como se muestra en la figura 4. Note que no hay control global para esta prioridad, puesto que toda unidad de enrutamiento sabe si es o no el maestro. Esta fase se termina en un ciclo de reloj, puesto que la propagación de las señales es combinatoria.

Fase 2: Difundiendo la Dirección

Una vez que un maestro ha sido seleccionado, este envía su dirección, en el caso de una fuente, o la dirección de la fuente deseada, en el caso de un destino. Como se mostró en la sección 3, la dirección se guarda en una molécula conectada a la unidad de enrutamiento. Y se envía serialmente, en n ciclos de reloj, donde n es el tamaño de la dirección. El mismo camino de la primera fase se usa para difundir la dirección, como se muestra en la figura 5.

Toda unidad de enrutamiento, salvo aquella que envía la dirección, compara el valor que llega con su propia dirección (guardada en la molécula). Al final de esta fase, es decir después de n ciclos de reloj, cada unidad de enrutamiento sabe si esta implicada en el camino. En la práctica, hay una fuente única y al menos un destino.

Fase 3: Construyendo el Camino mas Corto

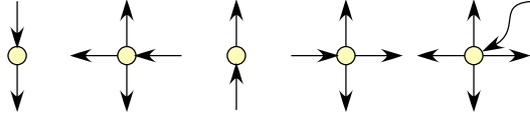


Fig. 5. La dirección de propagación de la dirección: norte \rightarrow sur || este \rightarrow sur, oeste, y norte || sur \rightarrow norte || oeste \rightarrow norte, este, y sur || unidad de enrutamiento \rightarrow norte, este, sur, y oeste.

La última fase, largamente inspirada por [5], crea el camino más corto entre la fuente seleccionada y los destinos seleccionados. Un ejemplo, con 8 fuentes y 8 destinos, se muestra en la figura 6, para una red conectada de manera muy densa.

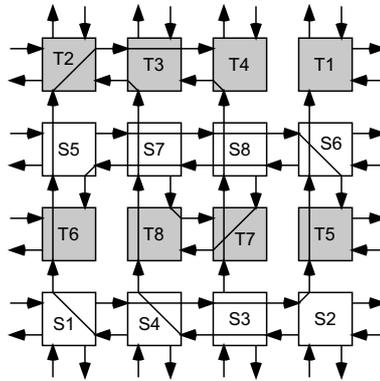


Fig. 6. Caso de test con una red densamente conectada.

Una realización paralela del algoritmo de búsqueda de tipo *breadth-first* permite a las unidades de enrutamiento encontrar el camino más corto entre una fuente y varios destinos. Empezando de la fuente, un proceso de expansión trata de encontrar destinos. Cuando se alcanza una, el camino se fija y todos los recursos de enrutamiento utilizados no estarán disponibles para las próximas iteraciones sucesivas del algoritmo.

La figura 7 muestra el desarrollo del algoritmo, construyendo un camino entre una fuente localizada en la columna 1, línea 2, y una célula destino colocada en la columna 3, fila 3. Después de tres ciclos de reloj de expansión, se llega al destino y se fija el camino, prohibiendo el uso de la misma trayectoria para un enrutamiento sucesivo.

5 Conclusión

En este artículo hemos presentado un nuevo circuito electrónico dedicado a la realización de aplicaciones celulares bio-inspiradas. Esta compuesto de un mi-

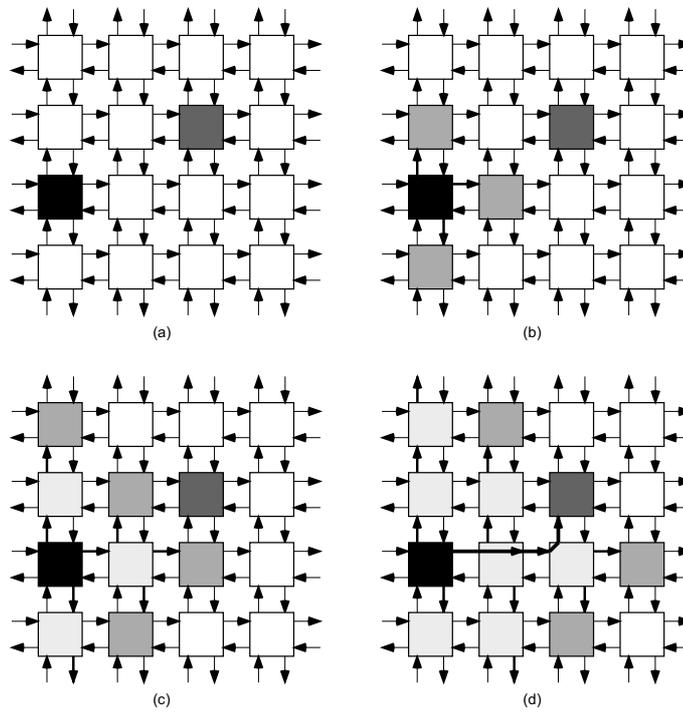


Fig. 7. Paso (a) uno, (b) dos, (c) tres y (d) cuatro del proceso de construcción del camino entre la fuente colocada en la columna 1, línea 2, y la célula destino colocada en la columna 3, línea 3.

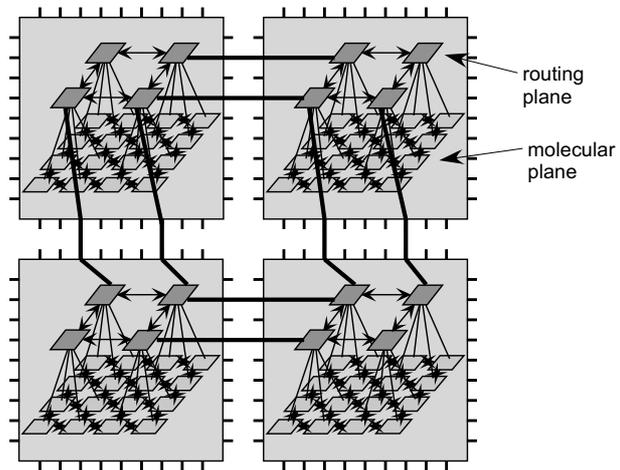


Fig. 8. Un organismo multi-chip mostrando las conexiones intercelulares.

croprocesador de tipo RISC y de dos planos: unidades funcionales y unidades de enrutamiento. El primero, una matriz bidimensional de moléculas, es similar a los circuitos FPGA standard y hace que el circuito sea lo suficientemente general para implementar cualquier sistema digital. Sin embargo, las moléculas tienen capacidades de auto-configuración que no están presentes en los circuitos FPGA comerciales y que son importantes para el crecimiento de un organismo y para la auto-reparación a nivel celular. El segundo plano es una matriz bidimensional de unidades de enrutamiento, que implementan un algoritmo de enrutamiento dinámico. Este plano se usa para la comunicación intercelular, permitiendo al tejido la creación dinámica de caminos entre células. El algoritmo es completamente distribuido, por lo cual no requiere el control del microprocesador. Aun más, su escalabilidad permite la creación de redes celulares de cualquier tamaño.

Este circuito ha sido simulado a partir de una descripción en VHDL de todo el sistema. El próximo paso del proyecto, actualmente en marcha y que debería estar terminado en la época de la conferencia, es desarrollar el layout VLSI y realizar un chip de test para validar el diseño del circuito.

Debido a razones financieras, el primer prototipo del chip POEtic tendrá únicamente el equivalente de 500'000 puertas lógicas. Esto no será suficiente para el diseño de sistemas complejos. Con ese pequeño número de moléculas, será posible implementar únicamente un organismo muy simple. Por lo tanto, hemos incluido en el diseño la posibilidad de implementar un organismo multi-chip (figure 8).

6 Agradecimientos

Este proyecto ha sido financiado por el programa Future and Emerging Technologies (IST-FET) de la Unión Europea, bajo el contrato IST-2000-28027 (PO-ETIC). La información proporcionada lo es bajo la sola responsabilidad de los autores y no refleja la opinión de la UE. La UE no es responsable del uso que pueda hacerse de los datos que aparecen en esta publicación. Los participantes suizos del proyecto están financiados por el contrato 00.0529-1 del gobierno suizo.

References

- [1] ARM: Amba Specification, rev 2.0. Advanced RISC Machines Ltd (arm). http://www.arm.com/armtech/amba_spec, 1999.
- [2] Brown, S., Francis, R., Rose, J., Vranesic, Z.: *Field Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.
- [3] Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [4] Eriksson, J., Torres, O., Villa, A. E. P.: Spiking Neural Networks for Reconfigurable POEtic Tissue. In A.M. Tyrrell, P.C. Haddow, and J. Torresen, editors, *Evolvable Systems: From Biology to Hardware. Proc. 5th Int. Conf. on Evolvable Hardware (ICES 03)*, volume 2606 of *LCNS*, pages 165–173, Berlin, 2003, Springer-Verlag.

- [5] Moreno Arostegui, J. M., Sanchez, E., Cabestany, J.: An In-system Routing Strategy for Evolvable Hardware Programmable Platforms. In *Proc. 3rd NASA/DoD Workshop on Evolvable Hardware*, pages 157–166. IEEE Computer Society Press, 2001.
- [6] Roggen, D., Floreano, D., Mattiussi, C.: A Morphogenetic System as the Phylogenetic Mechanism of the POETic Tissue. In A.M. Tyrrell, P.C. Haddow, and J. Torresen, editors, *Evolvable Systems: From Biology to Hardware. Proc. 5th Int. Conf. on Evolvable Hardware (ICES 03)*, volume 2606 of *LCNS*, pages 153–164, Berlin, 2003, Springer-Verlag.
- [7] Sanchez, E., Mange, D., Sipper, M., Tomassini, M., Perez-Uribe, A., Stauffer, A.: Phylogeny, Ontogeny, and Epigenesis: Three Sources of Biological Inspiration for Softening Hardware. In T. Higuchi, M. Iwata, and W. Liu, editors, *Evolvable Systems: From Biology to Hardware*, volume 1259 of *LCNS*, pages 33–54, Berlin, 1997. Springer-Verlag.
- [8] Sipper, M., Sanchez, E., Mange, D., Tomassini, M., Perez-Uribe, A.: A Phylogenetic, Ontogenetic, and Epigenetic View of Bio-inspired Hardware Systems. *IEEE Transactions on Evolutionary Computation*, 1:1:83–97, 1997.
- [9] Tempesti, G., Roggen, D., Sanchez, E., Thoma, Y., Canham, R., Tyrrell, A.M.: Ontogenetic Development and Fault Tolerance in the POETic Tissue. In A.M. Tyrrell, P.C. Haddow, and J. Torresen, editors, *Evolvable Systems: From Biology to Hardware. Proc. 5th Int. Conf. on Evolvable Hardware (ICES 03)*, volume 2606 of *LCNS*, pages 141–152, Berlin, 2003, Springer-Verlag.
- [10] Tyrrell, A.M., Sanchez, E., Floreano, D., Tempesti, G., Mange, D., Moreno Arostegui, J.-M., Rosenberg, J., Villa, A.E.P.: Poetic Tissue: An Integrated Architecture for Bio-inspired Hardware. In A.M. Tyrrell, P.C. Haddow, and J. Torresen, editors, *Evolvable Systems: From Biology to Hardware. Proc. 5th Int. Conf. on Evolvable Hardware (ICES 03)*, volume 2606 of *LCNS*, pages 129–140, Berlin, 2003, Springer-Verlag.