

DISEÑO DE UN GENE DIGITAL EN FPGA Y MATLAB CON APLICACIONES EN ROBÓTICA MÓVIL

J. Prieto⁺, O. Ramos⁺, A. Delgado^{*}

⁺Estudiantes, ^{*}Profesor Titular
{jprietor, odramosc}@unal.edu.co, adelgado@ieee.org

Departamento de Ingeniería Eléctrica y Electrónica
Universidad Nacional de Colombia, Bogotá

RESUMEN

El presente artículo ilustra el funcionamiento del gene digital, a partir de la emulación de su contraparte biológica. Se explica detalladamente la implementación con la ayuda de herramientas de programación como el software MATLAB y Xilinx. Finalmente se mencionan aplicaciones donde puede utilizarse el gene digital en arreglos programables de computas (FPGAs).

1. INTRODUCCIÓN

La necesidad de presentar una constante innovación en los métodos de diseño de mecanismos robóticos, con la finalidad de obtener mejores y más rápidas respuestas en su interacción con el medio que los rodea, ha sido una constante en la historia de la robótica móvil.

Resultados de esta búsqueda han planteado la importancia de emular el comportamiento biológico en todas sus etapas. Dentro de las características emuladas, que es el foco principal del presente documento, encontramos el gene biológico.

En esencia, los genes son segmentos de ADN transcritos en mRNA. Su composición se basa en cadenas de tres moléculas conocidas como nucleótidos. Desde el punto de vista de la emulación electrónica, encontramos varias posibilidades de programación e implementación del gene biológico, conocido como gene digital.

Entre ellas, encontramos algoritmos realizados en MATLAB, y diseños de hardware en Xilinx.

Este tipo de sistemas emulados, es muy útil en aplicaciones que requieren de procesamiento paralelo basado en la ejecución de acciones a partir de condiciones propias del mecanismo y del medio que lo rodea. Como en la navegación de un robot, entre otros.

El objetivo principal de nuestro trabajo, es plantear algunos métodos de diseño e implementación del gene digital. Adicionalmente, realizar propuestas de aplicación viables del modelo, con énfasis en la robótica.

En el presente artículo, realizaremos una explicación más detallada de los aspectos mencionados. Para ello, realizamos una división en secciones, en la segunda se discute la estructura de registros del gene digital. En la sección tres detallamos los procedimientos para implementar el gene digital en MATLAB y en el

ambiente Xilinx. En la sección tres, proponemos dos aplicaciones que pueden ser realizadas usando el gene digital, enfatizando el proceso que debe ser utilizado, y los criterios de diseño que deben establecerse. Por último, presentamos las conclusiones formuladas durante la realización del gene digital.

2. GENE DIGITAL

Basados en los aspectos mencionados, se plantea el gene digital como una configuración de registros de forma tal que se logre asociar una acción a un comportamiento determinado en la entrada del sistema, ver figura 1. El robot incorpora en un solo registro binario, las lecturas de todos sus sensores (detectores, sensores digitales y análogos). Con el registro de sensores u se obtiene, de manera paralela, el registro de la acción requerida y [1].

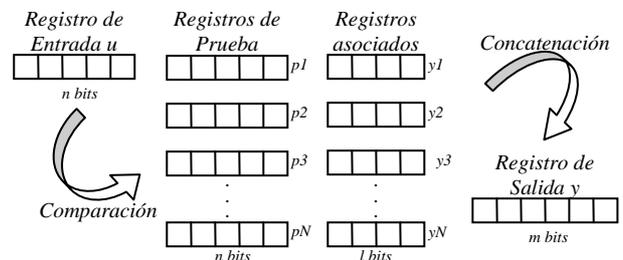


Figura 1. Diagrama de registros en el gene digital.

Hay cuatro secciones de trabajo: (i) un registro de entrada (u); (ii) una serie de registros de prueba (p_i); (iii) registros asociados a los anteriores (y_i); (iv) un registro de salida (y).

La operación consiste en realizar comparaciones, en paralelo, entre el registro de entrada y cada uno de los registros de prueba. A partir de esta comparación y según alguna restricción, el registro asociado correspondiente, es o no concatenado al registro de salida. Como método para realizar la comparación, se propone el uso de la distancia de Hamming, en la que se mide el número de bits diferentes entre los registros.

Para determinar la condición de concatenación, definimos un parámetro llamado umbral de Hamming, si la distancia es menor al umbral, se concatenará el registro asociado correspondiente.

2.1 Programación en MATLAB

La etapa inicial para aplicar el gene digital, fue el desarrollo de un algoritmo en MATLAB, figura 2, el cual llevará a cabo de forma secuencial, las etapas que conforman la estructura del gene. De esta manera podemos comprender mejor su funcionamiento, ventajas y debilidades, a partir de una observación de la tarea que éste realiza.

En principio es importante mencionar, que para facilitar el manejo de los registros, y tener un mejor control y verificación del funcionamiento del algoritmo, se usa una matriz para los registros de prueba y otra para sus registros asociados, y que por otra parte, estos últimos hacen referencia al número de la fila al que pertenece el registro de prueba correspondiente.

Primero que todo se realiza la definición del umbral de Hamming y las dimensiones con las que se desea trabajar. La cantidad de registros de prueba (N), así como su número de bits (n), y el valor del umbral, son entradas del programa.

En seguida y a partir del número de registros, se calcula el número de bits (l) necesarios para que los registros asociados puedan tomar todos los valores posibles de las filas.

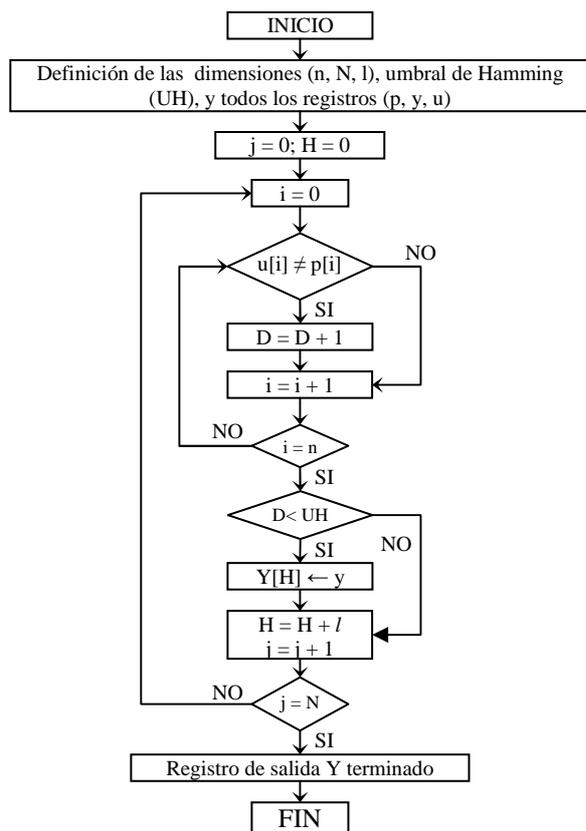


Fig 2. Diagrama de flujo del gene digital para MATLAB.

Ahora es posible definir las matrices y vectores con los cuales se va a operar. Estos son, una matriz de prueba (pi) con dimensiones $N \times n$, una matriz asociada (yi) de dimensiones $N \times l$, y un vector de salida (Y) de tamaño máximo $m = l \times N$.

Posteriormente, se establece de manera aleatoria los valores de la matriz de prueba y se llena la matriz asociada con los valores binarios de las filas respectivas. Debemos tener en cuenta que en la matriz de prueba deben estar los registros relacionados directamente con el comportamiento de la entrada a comparar, pero por motivos de practicidad, fueron llenados aleatoriamente.

Partiendo del propósito de la verificación del comportamiento del gene, se muestra la matriz de prueba junto con su asociada; en seguida se introduce el registro de entrada, teniendo presente los de prueba, y los resultados que las comparaciones producirán.

Una vez se han establecido los valores en todos los registros, se procede con la comparación entre el registro de entrada y cada uno de los registros de prueba, generando el valor de la distancia de Hamming. Para ello, se compara bit por bit, aumentando un contador (d) cada vez que estos sean diferentes.

Inmediatamente, comparamos la distancia obtenida con el umbral de Hamming, de tal manera que concatenamos el registro asociado al de prueba, siempre que la distancia sea menor a dicho valor límite.

Al terminar con todas las comparaciones y concatenaciones, se ilustra un registro de salida final (Y), cuya dimensión depende del número de registros finalmente agregados.

2.2 Programación en ambiente Xilinx

A continuación, explicaremos el proceso de programación a partir del software Xilinx. Este tipo de programa nos permite realizar implementaciones en dispositivos de arreglos de compuertas programables y llevar el concepto del gene digital a un mecanismo con procesamiento paralelo.

Un FPGA posee una gran ventaja en comparación con otros dispositivos programables como los microcontroladores, ésta se basa principalmente, en su capacidad de procesamiento paralelo, lo cual brinda mayor velocidad a un menor costo de capacidad y tiempo de programación [2].

En lo referente al gene digital, el programa se basó en tres módulos principales, los cuales explicaremos a continuación, ver figura 3.

En el primer módulo, llamado *comparación*, encontramos un registro de entrada U , el cual está conformado por cinco bits y es controlado directamente por el usuario. En esta sección se realiza la operación lógica XOR entre la entrada y cada uno de los registros de prueba definidos como constantes, los resultados de dichas operaciones, son las salidas de la etapa.

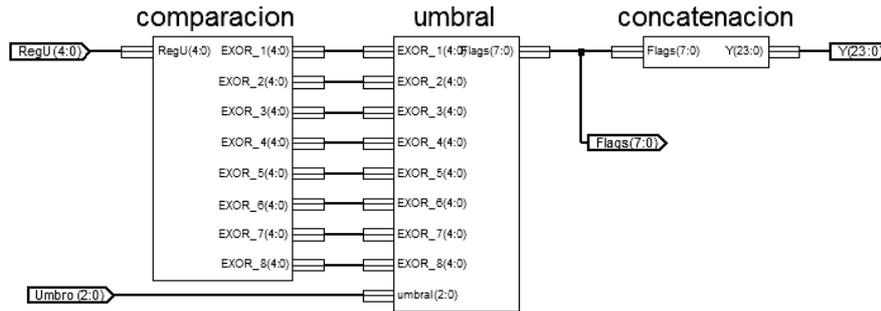


Fig. 3. Gene digital formado por los módulos de comparación, umbral y concatenación.

Posteriormente, encontramos la etapa conocida como *umbral*. Sus entradas son los resultados de las compuertas XOR y la especificación, por el diseñador, del vector de umbral. Dentro del bloque, definimos un tipo de arreglo entero que representa la conversión de cada entrada en su valor decimal, de tal manera que podamos realizar una comparación de *menor que*, entre estas entradas y el valor entero del umbral.

La condición presente, para continuar con el proceso, es un valor límite de la distancia de Hamming, entre la entrada y los registros de prueba. Si el resultado de esta medición es menor al umbral, se activa una bandera (posición de un bit, dentro del registro de salida), que junto a siete más, conformarán la salida del módulo, dicho registro fue llamado *flags*. Si el resultado no cumple la condición, la bandera permanece inactiva.

El último bloque, *concatenación*, posee un arreglo que representa la salida y que está conformado por la concatenación de los posibles registros de acción, correspondientes de acuerdo al cumplimiento de la condición. Para efectos de simulación, los registros asociados serán de tres bits, y habrá ocho debido a que poseemos igual número de registros de prueba; de tal manera que la salida será de 24 bits.

El registro anterior, conformado por los ocho registros asociados, tendrá un valor dependiente de la condición cumplida, de tal manera que si la bandera se encuentra encendida, el registro de acción estará presente; si el valor es cero, los bits del registro asociado se encontrarán en estado de alta impedancia.

En la implementación del gene digital en FPGA decidimos realizar la confirmación a partir de la revisión de los bits de *flags*, los cuales se hallaban representados por los LEDs (Diodos Emisores de Luz) en la tarjeta. Si un LED se prende quiere decir que la entrada cumple la condición de distancia de Hamming del respectivo registro de prueba.

3. APLICACIONES

Una vez hemos podido implementar de forma práctica la función del gene digital, nos dedicaremos a buscar las maneras de aplicarlo en diversos ámbitos, y así aprovechar su potencial de procesamiento en máquinas robóticas.

En primer lugar, vamos a conceptualizar el contenido de los diferentes registros, en un contexto de robótica. El registro de entrada hará referencia a la lectura de algún

sensor, en los registros de prueba habrá valores específicos con los cuales se realizarán las comparaciones. Estos valores deberán ser elegidos de forma tal que generen acciones de respuesta, dependiendo de la lectura del sensor, y con ayuda de los registros asociados.

Estos últimos, por tanto, indicarán tareas a realizar, o fragmentos de ellas, que al concatenarse con otros, formarán la acción.

Partiendo de la concepción anterior, presentamos un posible campo de acción para el gene digital.

3.1 Navegación de Robots en Áreas Limitadas

En esta sección, propondremos una aplicación que involucra el control de uno o más robots móviles dentro de un espacio limitado por el usuario.

La aplicación consiste en realizar una programación del robot, usando gene digital, en la cual los registros asociados de acción, se activen de acuerdo al cumplimiento de la regla presente y definida por el usuario. En nuestro caso, dicha regla es alejarse de uno o varios puntos en el espacio, seleccionados acorde a la necesidad del momento: por ejemplo, si el robot se encuentra sobre una mesa, es importante que no vaya a caerse de ella, de tal manera que el borde de la misma sería la *zona prohibida*, figura 4.

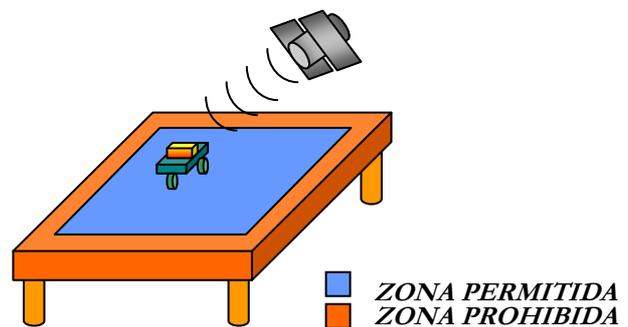


Figura 4. El robot usa un GPS y es controlado por el gene digital para evitar caer de la mesa.

Para poder realizar lo anterior, necesitaríamos de un dispositivo que nos proporcionara la posición actual del móvil, tal instrumento podría ser un GPS (*Global Positioning System*), el cual nos daría unas coordenadas, que serían codificadas en un valor binario relacionado

con el área propuesta. Dicho valor binario, sería el registro de entrada del gene digital, donde a partir de un umbral de Hamming adecuado, determinaría la acción a realizar, como seguir avanzando hacia adelante, o cambiar el sentido del desplazamiento.

En el ejemplo de la mesa, podríamos realizar una codificación correspondiente a los bordes de la misma, de manera tal, que la lectura proporcionada por el GPS generara una distancia de Hamming significativa. Siendo sencillo el planteamiento de un umbral que evitara la caída del robot.

Adicionalmente a este funcionamiento relativamente sencillo y sin ningún obstáculo para el movimiento del robot, podríamos realizar un enlace vía microondas con un computador central que dirigiera el comportamiento de las máquinas.

El computador podría establecer nuevas distancias de umbral, o asignar una codificación diferente a las lecturas del GPS, para permitirle a los robots actuar de tal manera que se alejaran del punto inicial de trabajo. Se podría añadir programación para la identificación de obstáculos, y a partir del mismo enlace de comunicaciones, proporcionarle al control central, unas lecturas que describirían el área recorrida, pudiéndose hacer un mapa computacional de la zona.

Lo anterior sería muy útil en aplicaciones de exploración de terrenos hostiles, donde se podría enviar toda una caravana de robots con la misma programación, y controlarlos paralelamente, ver figura 5.

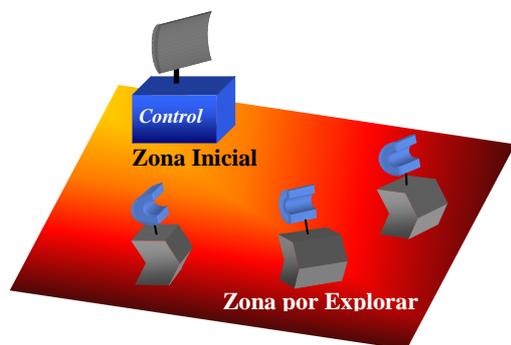


Figura 5. La caravana de robots se desplaza para explorar zonas desconocidas.

4. CONCLUSIONES

Al tomar ejemplo de los procesos biológicos presentes en la naturaleza, podemos aprovechar toda su experiencia y continua evolución, implementando nuevas y eficientes soluciones o dispositivos electrónicos, con un sin número de campos de acción.

Existe una gran variedad de procedimientos en el campo de la robótica que nos permite llevar a cabo procesos de control de mecanismos programables, de una manera simplificada y con una extensa capacidad de procesamiento. Entre ellos se encuentra el gene digital que, debido a su planteamiento paralelo, representa una alternativa viable como fundamento de funcionamiento de este tipo de aplicaciones.

Las diferentes herramientas de software y hardware nos permiten realizar pruebas de funcionamiento de los programas de una manera sencilla y eficaz. En el caso de MATLAB, plantea la posibilidad de crear una interfaz de usuario con una perfecta simulación de la operación del algoritmo. Para XILINX, nos da la posibilidad de realizar una programación en hardware, donde la simulación se realiza en condiciones reales, y donde el dispositivo programado, puede ser utilizado en un robot verdadero.

El dispositivo utilizado en el desarrollo de la programación en hardware fue una FPGA de XILINX, cuya referencia es Spartan - 3 XC3S200, con una capacidad de 200.000 compuertas.

Adicional al integrado anterior, la tarjeta posee componentes adicionales como son: una memoria ROM, ocho interruptores, ocho diodos emisores de luz (LEDs) y tres puertos de expansión de 40 pines, que pueden ser utilizados como entradas y salidas, entre otros.

5. REFERENCIAS

- [1] A. Delgado, "Digital Gene", IET Electronics Letters, En evaluación, Noviembre, 2006.
- [2] D. Vargas, D. Perlaza, A. Delgado: "Diseño e implementación de un robot móvil controlado con chip ADN en FPGA", XI Taller Iberchip, Salvador de Bahía - Brasil, Marzo, 2005.