

Lógica Difusa en FPGA

José A. Olivas, Yazmin Maldonado, Oscar Montiel, Roberto Sepúlveda

Fuzzy logic in an FPGA

Recibido: abril 10, 2007
Aceptado: junio 01, 2007

Palabras clave: FPGA, lógica difusa, lenguaje de programación

Abstract:

Este va a ser el abstract el cual debe ser en ingles.

Keywords: FPGA, Fuzzy logic, programming language

LOS sistemas difusos han generado un impacto apreciable dentro del contexto académico e industrial. Con el desarrollo de los dispositivos lógicos programables y las metodologías de diseño asociadas, como es el caso de los lenguajes de descripción de hardware, es posible construir sistemas computacionales complejos que integran en un solo circuito integrado (CI) una gran cantidad de funciones; destacando dentro de éstos, los CPLD (Complex Programmable Logic Device), ASIC (Application-Specific Integrated Circuit) y FPGA (Field Programmable Gate Array). Los principales inconvenientes asociados con la realización microelectrónica de sistemas difusos provienen del elevado costo en términos económicos y en tiempo de desarrollo que conlleva el diseño y fabricación de un CI. En particular, el uso de arquitecturas específicas de procesamiento implementadas sobre FPGA's proporciona una excelente relación "costo-rendimiento" y un ciclo de desarrollo extremadamente corto [5].

En la actualidad algunas universidades se encuentran investigando sobre lógica difusa en FPGA como son: Universidad de Alberta, Canadá, Universidad de Colombia, Colombia; y en la República Mexicana tenemos al Centro de Investigación y Desarrollo de Tecnología Digital del IPN (CITEDI), Instituto Tecnológico de Tijuana (ITT), Universidad Autónoma de Baja California (UABC), México, etc.

En este artículo se explican algunas características de lógica difusa y de la plataforma FPGA, así como algunos lenguajes de programación para FPGA; se mencionan también algunas aplicaciones de lógica difusa en FPGA y su auge dentro de las investigaciones actuales; se presentan algunas herramientas que facilitan el trabajo para esta tecnología.

SISTEMA DIFUSO

Un Sistema difuso consta de tres etapas: Fuzzificación, Reglas difusas y Defuzzificación, como se puede observar en la Figura 1 [2].

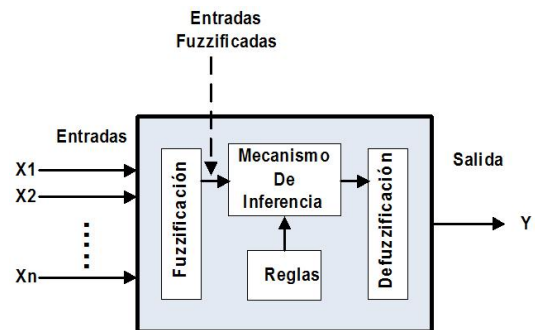


Figura 1. Estructura básica de un sistema de control difuso.

Fuzzificación

Es la interpretación de los datos de entrada por el sistema de control difuso y tiene dos componentes; la variable lingüística y la función de membresía. La variable lingüística es aquella cuyos valores son palabras en lugar de números, por ejemplo:

La estatura es una variable lingüística, si sus valores en lugar de ser numéricos son lingüísticos: alto, bajo, muy alto, muy bajo, no muy alto, no muy bajo.

La función de membresía (FM) mapea cada punto del espacio de entrada a un valor de membresía (o grado de mem-

bresía) entre 0 y 1. Generalmente, el experto define cada FM en cuanto a tipo y parámetros siguiendo ciertas reglas; algunos tipos comunes son: triangular como en la Figura 2, gaussiana, trapezoidal, etc., [2].

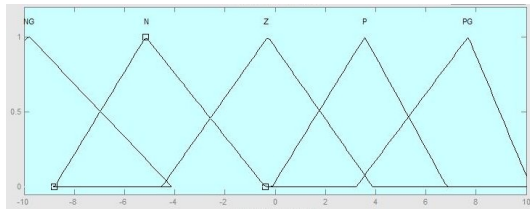


Figura 2. Funciones de membresía triangulares.

Las funciones de membresía están relacionadas con las variables lingüísticas, es decir si se tiene el ejemplo de la estatura, el número de funciones de membresía serán el número de términos lingüísticos.

Inferencia o Reglas difusas

Una vez que se han definido las variables lingüísticas, sus términos y las funciones de membresía que se van a utilizar, la serie de pasos que describen la solución del problema se traducen a reglas de la forma IF-THEN [1], es decir

IF x is A THEN y is B

Donde A y B son los valores lingüísticos de las variables de entrada y salida respectivamente.

Defuzzificación

Después de computar las reglas difusas y evaluar las variables difusas, se espera que en lugar de tener un conjunto difuso, la respuesta sea un valor real. Existen varios métodos que se pueden aplicar, sin embargo el más común es el centro de gravedad, el cual obtiene el centro de masa de las funciones de membresías, y a la salida proporciona un dato. Es uno de los más utilizados porque su salida es muy representativa.

Aplicaciones de Sistemas Difusos

Hoy en día la lógica difusa se ha hecho acreedora de la confianza en las investigaciones y aplicaciones, ya que ha dado muy buenos resultados, tales como:

- Productos al consumidor
- Lavadoras
- Hornos de Microondas
- Cámaras de video
- Televisiones
- Traductores
- Sistemas
- Automóviles
- Trenes
- Elevadores
- Controles de tráfico

Existen también algunas aplicaciones para remplazar a un operador humano por medio de reglas difusas, por ejemplo:

- Péndulo invertido de Yamakawua
- Reactor nuclear (Hitachi),
- Carro de Sugeno
- Robot de Hirota
- Producción de etanol (Filev)

Hoy en día la lógica difusa se ha hecho acreedora de la confianza en las investigaciones y aplicaciones, ya que ha dado muy buenos resultados.

A continuación se muestra una tabla donde se indican las principales compañías Japonesas que emplean lógica difusa.

Tabla 1. Compañías de Japón que emplean lógica difusa.

Compañías	Aplicaciones
Canon	Cámaras fotográficas
Casio	Control de Humedad Temperatura
Daldan	Plantas Gas Refrescante
Fuji electronic	Mezclador Químico
Hitachi	Control en Elevadores Control del subterráneo de Sendai
Idec Izumi	Crecimiento Cristalino de GaAs
IshidaInstruments	Medidores Automáticos
Mitsubishi Electric	Control en Elevadores
Nissan Motor Company	Transmisión Automática sistemas de frenado ABS
Nuclear Power Corp	Control de Potencia en Plantas Nucleares, Controladores Roboticos
Ricoh	Reconocimiento de Voz
Sanyo	Control del Diafragma en Cámaras Fotográficas
Seiko	Sistemas Expertos
Toshiba	Control de Elevadores

FPGA

Es un dispositivo semiconductor que contiene componentes lógicos programables e interconexiones programables entre ellos, ver Figura 3. La raíz histórica de los FPGA's son los dispositivos de lógica programable compleja (CPLD) de mediados de los ochenta. Su creador es Ross Freeman, cofundador de Xilinx [3]. Muchos FPGA's modernos soportan la reconfiguración parcial del sistema, permitiendo que una parte del diseño sea reprogramada, mientras las demás partes siguen funcionando. Este es el principio de la idea de la "computación reconfigurable", o los "sistemas reconfigurables". Los FPGA's son generalmente más lentos que sus contrapartes, los circuitos integrados de aplicaciones específicas (ASIC por sus siglas en inglés), no pueden soportar diseños muy complejos, y consumen más energía. Sin embargo, los FPGA tienen ciertas ventajas tales como la reducción del tiempo de introducción al mercado de productos, la habilidad de ser reprogramados después de haber salido al mercado a fin de corregir posibles errores, y reducir los costos de investigación, diseño y pruebas de un nuevo producto.

En la Figura 4 se puede apreciar un chip FPGA, su fabricante es ALTERA [4].

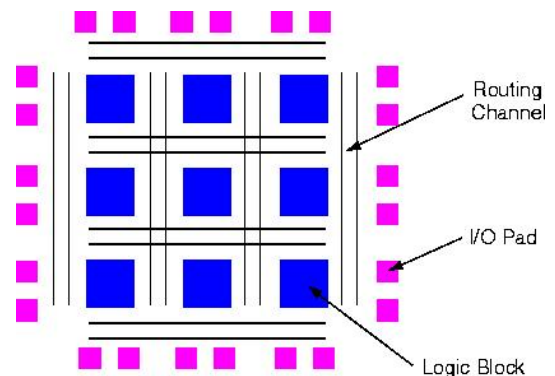


Figura 3. Estructura interna de un FPGA.

Lenguaje de Descripción de Hardware

El diseñador cuenta con la ayuda de entornos de desarrollo especializados en el diseño de sistemas a implementarse en un FPGA. Un diseño puede ser capturado ya sea como esquemático, o haciendo uso de un lenguaje de programación especial [6]. Estos lenguajes de programación especiales

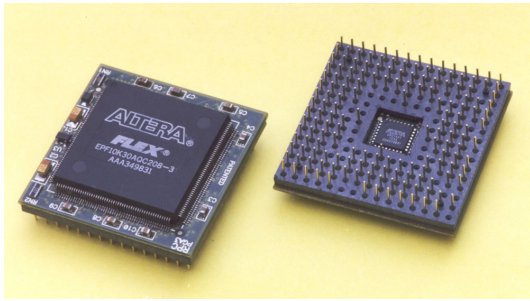


Figura 4. Chip de un FPGA, fabricante ALTERA.

son conocidos como HDL o Hardware Description Language (lenguajes de descripción de hardware). Los HDLs más utilizados son:

- VHDL
- VERILOG
- ABEL

VHDL

Del idioma Inglés Very High Description Language, esto significa que mediante él se puede describir la forma de comportarse de un circuito electrónico. El comportamiento puede ser llevado a algún dispositivo que dispondrá de sus propios componentes con los que lograr ese comportamiento deseado. VHDL es un estándar llamado IEEE 1076-1993. Un proyecto de VHDL puede contener muchos archivos. El código VHDL usualmente se encuentra en los archivos con extensión *.vhd.

VERILOG

Verilog HDL es un lenguaje descriptivo del hardware usado para diseñar y para documentar sistemas electrónicos; permite que los usuarios diseñen en los varios niveles de la abstracción. Apoya el diseño, la verificación, y la puesta en práctica de los circuitos análogos digitales.

ABEL

Por sus siglas Advanced Boolean Expression Language, permite implementar diseños lógicos en dispositivos lógicos programables. Es un lenguaje independiente del dispositivo. Es decir que se programa en la computadora y se ejecuta en el dispositivo. Este lenguaje es común utilizarlo en PLD Dispositivo Lógico Programable.

APLICACIONES DE LOGICA DIFUSA EN FPGA

A continuación se muestran algunas aplicaciones e investigaciones de lógica difusa en FPGA:

Controladores difusos

Uno de los campos de mayor desarrollo teórico y de aplicaciones de lógica difusa es el de control de procesos. En los últimos años se ha producido un incremento considerable del número de aplicaciones de control que emplean técnicas de inferencia basadas en lógica difusa, debido a que estas técnicas permiten el desarrollo de sistemas de control complejos a partir de la descripción lingüística del conocimiento de un operador experto, sin necesidad de emplear modelos matemáticos y con buenas características de robustez frente a cambios de las condiciones de operación. Los controladores difusos se emplean en los FPGA's, debido a que se aprovechan sus características, es decir, ofrece ventajas de bajo costo, espacio pequeño, incorporación de interfaces, etc.

Algunos ejemplos de control:

- Controlador difuso para frenos ABS Mediante FPGA.
- Control Difuso de Navegación de un Robot Móvil Mediante FPGA.
- Implementación de Controladores Difusos para Tráfico de Internet Basados en FPGA.
- Desarrollo de Módulos IP de Controladores Difusos para el Diseño de Sistemas Empotrados Sobre FPGA.
- Controladores Difusos PD, PI, PID Implementados en FPGA.

En la Figura 5 se muestra un diagrama a bloques de un controlador difuso implementado en un FPGA; se puede apreciar que las únicas variables que no se incluyen dentro del software del FPGA son la planta, el sensor, el actuador y las variables de entrada.

Co-Procesador Difuso

La ventaja principal de los procesadores difusos en hardware sobre las aplicaciones de software es su alto rendimiento. Esta ventaja es muy apreciada en aplicaciones

en tiempo real y empotradas. El diseño de la tarjeta estructural esta predefinida por la aceleración de los cálculos difusos. Una RAM ordinaria externa tiene que conectarse al coprocesador difuso y al anfitrión (host) como una regla de inferencia difusa dedicada y un medio de la base de datos difusa. La arquitectura total del sistema es bastante simple y de bajo costo. Esta arquitectura ha sido puesta en práctica sobre una plataforma de diseño de FPGA XC4005. Esto hace confiable la utilización del coprocesador difuso en usos en tiempo real y/o empotrado.

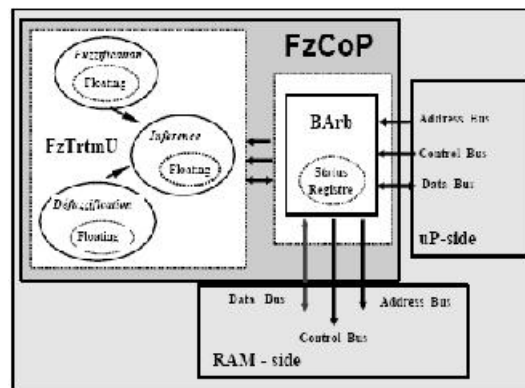


Figura 6. Arquitectura de un Fzcop.

La ventaja principal de los procesadores difusos en hardware sobre las aplicaciones de software es su alto rendimiento.

Herramientas

Algunas herramientas de las más usadas para la lógica difusa en FPGA son LABVIEW, SIMULINK de MATLAB, y XFUZZY una herramienta de CAD, a continuación se explica su funcionamiento:

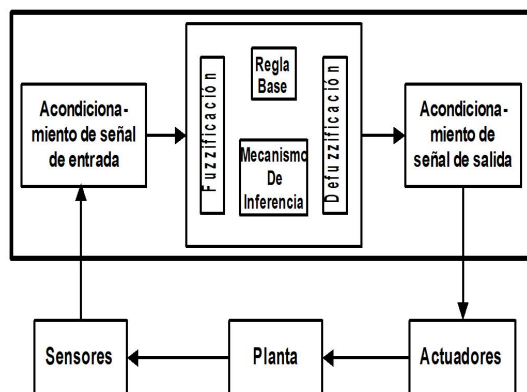


Figura 5. Diagrama a Bloques de un controlador difuso en FPGA.

El coprocesador difuso (Fzcop), Figura 6, implementado en hardware es una opción preferida en los sistemas de control embebidos.

El diseño del Fzcop es eficiente y puede ser empleado en chips FPGA baratos disponibles en el mercado. Además, no es problemático ajustar la capacidad de RAM en función de las reglas difusas y de la complejidad de la base de datos [7].

LABVIEW

Labview cuenta con una herramienta para lógica difusa de la cual se describe su funcionamiento a continuación [9]. Lo primero que se debe hacer es posicionarse en el toolbox de herramientas de la lógica difusa de LabView (ver Figura 7) que se encuentra en el menú de las herramientas del ambiente de programación de LabView. Una vez que se encuentre en el toolbox, abrir un controlador y definir las variables.

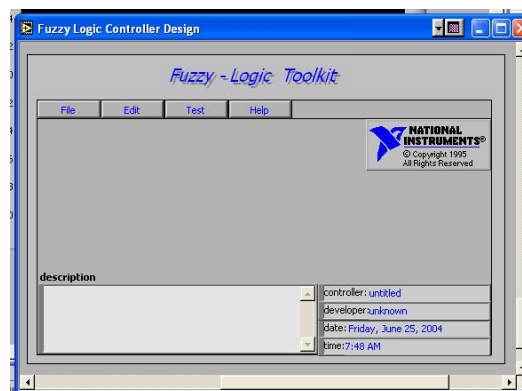


Figura 7. Ventana del toolbox de lógica difusa en LABVIEW.

En seguida se tiene que definir el modelo del sistema difuso, sus reglas difusas, y el tipo de defuzzificación, esto se hace de acuerdo a lo que necesite el usuario; ahora se necesita pasar el modelo difuso a código VHDL.

LabVIEW de National Instruments y el módulo LabVIEW FPGA proporcionan un ambiente de desarrollo gráfico para arreglos de FPGA's, en objetivos de hardware de E/S reconfigurables. Con el módulo LabVIEW FPGA, se pueden desarrollar instrumentos virtuales FPGA en un servidor ejecutando Windows y LabVIEW compila e implementa el código de hardware. Puede crear instrumentos virtuales de FPGA embebidos que combinan acceso directo a E/S con la lógica de LabVIEW definidas por el usuario para definir hardware personalizado para aplicaciones como protocolos de comunicación digital, simulación de control en hardware y generación rápida de prototipos de control.

El sistema de desarrollo de software de E/S reconfigurable de LabVIEW incluye todas las herramientas que se necesitan para crear un sistema embebido CompactRIO-LabVIEW FPGA para sintetizar hardware personalizado en el FPGA de E/S reconfigurable, configurado por el usuario y LabVIEW Real Time para crear aplicaciones determinísticas en tiempo real.

SIMULINK DE MATLAB

Matlab es una herramienta que muchos diseñadores la utilizan para construir y probar su sistema, esta herramienta junto con sus toolboxes proporcionan los medios eficaces para las etapas de modelado y simulación. Estas herramientas proporcionan los medios de extraer esa información y llevarla a otro formato. Tal es el caso de la conversión de un modelo en simulink que nos da un formato .mdl y lo lleva a un formato de VHDL (.vhd) [8]. Esto permite al usuario desarrollar y simular un algoritmo del control numérico usando Matlab y una vez que sea completo, convierte esto al código de VHDL. Esto entonces será sintetizado en el hardware digital de la lógica difusa para ser puesta en práctica en los dispositivos tales como FPGA's y ASIC's.

Lo primero que se debe hacer es abrir el toolbox llamado FUZZY, ahí es donde se creara el controlador difuso, en seguida exportar el .fis al workspace y crear el modelo en simulink, por ultimo convertirlo a .vhd. La conversión del modelo de simulink a VHDL se muestra en la Figura 8 [11]. Los tres pasos principales para la conversión son: Modelado y simulación, Conversión de datos y síntesis, Puesta en práctica de hardware.

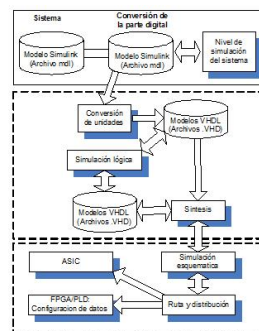


Figura 8. Rutina de la conversión de simulink a código VHDL.

En la Figura 9 se muestra un ejemplo de un retardo para convertirlo de un bloque de simulink a VHDL.

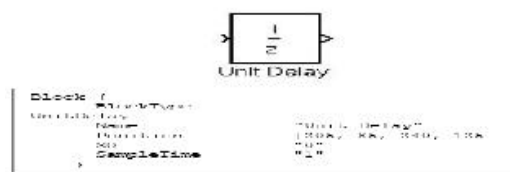


Figura 9. Unit delay model ejemplo

Figura 9. Bloque de simulink a código .vhd.

XFUZZY DE CAD

El entorno de desarrollo de sistemas difusos Xfuzzy, combina un conjunto de herramientas que facilitan las distintas etapas del proceso de diseño de sistemas de inferencia basados en lógica difusa, desde su descripción inicial hasta la implementación final. Sus principales características son la capacidad para desarrollar sistemas complejos y la flexibilidad para permitir al usuario extender el conjunto de funciones disponibles [10]. La Figura 10 muestra la ventana principal del entorno Xfuzzy.

La síntesis hardware de sistemas difusos usando el entorno Xfuzzy puede ser llevada a cabo siguiendo dos estrategias, las cuales se explicaran enseguida.

XFTL (Síntesis mediante Tabla de búsqueda)

XFTL recibe como entrada una descripción XFL y la convierte en una tabla entrada-salida con el formato PLA de la Universidad de Berkeley. Utilizando cualquier herramienta

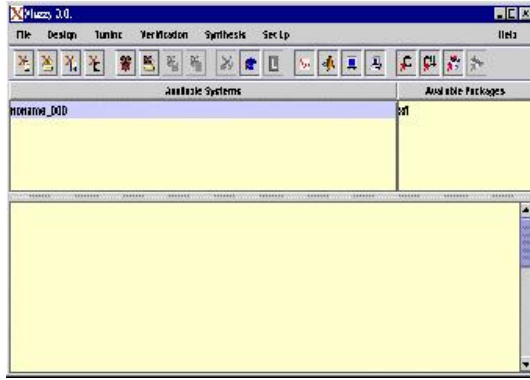


Figura 10. Ventana del entorno Xfuzzy.

de síntesis lógica compatible con dicho formato es posible minimizar la tabla y extraer las ecuaciones lógicas correspondientes. Si el usuario dispone además de las herramientas de síntesis de Synopsys y Xilinx (y el recurso xfuzzyHasSynopsys) XFTL puede automatizar los procesos de minimización de la tabla y su implementación en un FPGA.

XFVHDL (Síntesis Mediante Hardware Dedicado)

XFVHDL lee una especificación XFL y genera una descripción VHDL sintetizable basada en una arquitectura específica para sistemas difusos. Las diferentes opciones arquitecturales y el número de bits de precisión son definidos por el usuario al ejecutar XFVHDL. Utiliza una librería de celdas parametrizadas que contiene las descripciones de los bloques básicos que constituyen el sistema difuso. El código empleado en esta librería es compatible con las implementaciones de VHDL de las herramientas de Synopsys y Mentor Graphics. De acuerdo con condicionantes económicos o temporales el usuario puede elegir entre construir el sistema como un ASIC o mediante un FPGA. En este último caso XFVHDL también genera archivos script para dirigir los procesos de síntesis de Synopsys y Xilinx.

CONCLUSIONES

Una ventaja de la implementación de sistemas difusos en FPGA's es que se aprovechan sus características, es decir, ofrece ventajas de bajo costo, espacio pequeño, incorporación de interfaces, etc. El número de aplicaciones de lógica difusa en FPGA es cada vez mayor. Con la ayuda de

algunas herramientas de diseño permiten en un corto tiempo pasar de la especificación de un sistema difuso a la implementación en un FPGA lo que hace más fácil la aplicación de estos sistemas.

Bibliografía

- [1] Cirstea M.N, Khor J.G, McCormick M., *Neural and fuzzy logic control of drives and power system*, Newnes, 2002.
- [2] Montiel Oscar, Sepúlveda Roberto, Melin Patricia, Castillo Oscar, *Fundamentos de lógica difusa*, ILCSA, 2002.
- [3] Fabricante de FPGA's, consultar en www.xilinx.com, 2007.
- [4] Fabricante de FPGA's, consultar en www.atmel.com, 2007.
- [5] González Vázquez José Luís, Tesis *Análisis de controladores difusos implementados en procesadores digitales FPGA's*, Instituto Tecnológico de Tijuana, Marzo 2006.
- [6] D. Galán, C. J. Jiménez, A. Barriga, S. Sánchez Solano, "VHDL Package for Description of Fuzzy Logic Controllers", 1995.
- [7] Radoslav Raychev, Abdellatif Mtibaa, Mohamed Abid, "VHDL Modelling of a Fuzzy Co – processor Architecture", *International Conference on Computer Systems and Technologies CompSysTech*, 2005.
- [8] Distribuidor de MATLAB, disponible en www.mathworks.com, 2007.
- [9] National Instruments, distribuidor de LABVIEW disponible en www.ni.com, 2007.
- [10] XFUZZY: Herramientas de CAD para Lógica Difusa, disponible en www.imse.cnm.es/XFUZZY, 2007.
- [11] Ian A. GROUT , Modeling, simulation and synthesis: From Simulink to VHDL generated hardware, *Department of Electronic and Computer Engineering, University of Limerick, Ireland*.

Acerca del autor o autores

Oscar Montiel y Roberto Sepúlveda son Investigadores del Instituto Politécnico Nacional Centro de Investigación y Desarrollo de Tecnología Digital, Av. del Parque No. 1310, Mesa de Otay, Tijuana, B.C., México. José A. Olivas y Yazmin Maldonado son Alumnos de la Maestría en Sistemas Digitales del CITEDI. Correos-e: jym_cypress2@hotmail.com, shak_88@hotmail.com, o.montiel@ieee.org, r.sepulveda@ieee.org