

# Implementación en FPGA de una Lógica Dinámica Mixta (DLM) Modelo para una Planta Hidroeléctrica en el Sistema de Validación

German Ardul Muñoz Hernández (1,a), Carlos Arturo Gracios Marín (2,b), Ebhy Jaime Cuanalo Jiménez (2,c),  
Jesus Alejandro Solar Nava,(2,d) y Cesar Ávila Herrera.

(1) Benemérita Universidad Autónoma de Puebla. Av. San Claudio S/N Ciudad Universitaria, Puebla, Pue.  
MEXICO

(2) Instituto Tecnológico de Puebla. Av. Tecnológico #420 Colonia Maravillas Puebla, Pue. MEXICO

(a) gmunoz@ece.buap.mx, (b) cgracios@hotmail.com, (c) ebhy.cuanalo@gmail.com

(d) rasek\_cesar@hotmail.com

## Resumen

Este trabajo informa sobre el diseño de un campo Matriz de puertas programables (FPGA) de aplicación para ejecutar un entorno mixto de lógica dinámica (MLD). Que la aplicación representa una componente importante en el desarrollo de una coprocesador integrado por Eventos Discretos Sistemas de simulación del modelo básico.

El trabajo presentado aquí es la primera etapa de un proyecto en curso para proporcionar una simulación medio ambiente para el desempeño eficiente modelado de sistema de control en tiempo real. El objetivo de esta investigación es determinar la viabilidad del uso de hardware para acelerar la ejecución de la simulación, con el fin de reducir los tiempos de ejecución que se normalmente se encuentran en los simuladores de MLD.

## 1. Introducción

El desarrollo del software sujeto a estrictas especificaciones de funcionamiento temporal requiere de la generación, simulación y análisis de modelos adecuados de cálculo. Este tipo de software es necesario para las redes de información, sistemas de control distribuido y sistemas aeroespacial / defensa.

Con un enfoque de software de modelado de rendimiento, un diseño en particular se puede evaluar con las especificaciones antes de comprometerse con la codificación y la fase de prueba. Esto es vital en los sistemas de seguridad crítica donde el desempeño se debe garantizar a un muy alto grado de certeza. En casi todos los casos, además de la más trivial, un programa de pruebas no puede ofrecer este grado de

certeza. La actividad de diseño de software para estos sistemas puede ser en gran medida con la ayuda de herramientas de modelado de software que apoyan explícitamente una especificación temporal y la capacidad de verificación.

### 1.1 Arquitectura Lógica mixta dinámica y flujo de datos.

Varios modelos computacionales se pueden utilizar como base para la construcción de un modelo de software en tiempo real. Uno de los principales requisitos, además de apoyar la especificación de las limitaciones temporales, es que la concurrencia y la sincronización deben ser apoyadas. Un paradigma de modelado que soporta todos estos requisitos es la lógica mixta dinámica (MLD) [1]. La simulación de modelos MLD es un tema importante de investigación en tiempo real el desarrollo de sistemas [2], y uno de los problemas que ha sido claramente identificado por muchos investigadores [3], que los modelos han demostrado ser un entorno de simulación eficientes [4]. Un componente central de la ejecución de MLD eficiente es el algoritmo de descripción [1], que se describe más adelante en la Sección 5.

En general, la simulación de un modelo de un sistema de software complejo requiere importantes recursos de cómputo por lo tanto, algunos investigadores están buscando maneras de incorporar el procesamiento en paralelo a la estructura de estos simuladores. La base de este enfoque es que el sistema que se modela es siempre concordante, así como el modelo conserva la propiedad y el entorno de simulación puede explotar esta propiedad. La mayoría de los enfoques de la simulación simultánea de los modelos MLD han sido evaluadas en el esquema de cálculo como MATLAB [4], [5]. Esto ofrece una ventaja inmediata en ser

capaz de utilizar componentes estándar para construir el sistema de simulación con los bloques funcionales. Una desventaja importante es que todos estos bloques funcionales en la plataforma de cálculo científico han evaluado en un esquema no concurrente. En consecuencia, la concurrencia inherente en el modelo no se asigna de manera eficiente para que una arquitectura distribuida. Modelos MLD también se han ejecutado en un esquema de control [6] donde la estrategia de programación común que se necesita y la interconexión de gran flexibilidad es necesaria. El uso de coprocesadores agentes de soporte de alta velocidad entre el procesador y los vínculos de comunicación un cambio de contexto de hardware eficiente, pero fundamentalmente siguen siendo un control de flujo de arquitectura RISC (aunque muy RISC), y en última instancia, sufren la misma ineficiencia en esta aplicación como el control de flujo CISC arquitecturas.

Hay una fuerte conexión entre los modelos de DLM y el paradigma de la arquitectura FPGA, ya que ambos funcionan de forma dinámica sin el uso del programa de supervisión. También tienen la ejecución de bloques de funciones descritas por los sistemas de lógica y no necesitan de gran tamaño de la memoria. Efectivamente, el modelo MLD representa un modelo de datos gráfico de flujo que se puede aumentar con la información de especificación temporal.

En teoría, si la arquitectura de flujo de datos está disponible para ejecutar el modelo MLD, a continuación, una asignación eficiente debería ser posible hacer el mejor uso de los recursos computacionales de la máquina. Esto permitiría que los modelos de complejos sistemas de tiempo real a analizar con más finas que las actuales escalas de tiempo posible, o permitir que los sistemas más complejos para ser analizados en la reducción de los tiempos.

## 1.2 Objetivos del Proyecto

El objetivo de este proyecto es explorar la implementación de la arquitectura de flujo de datos de tipo para apoyar la ejecución eficiente del modelo de lógica mixta dinámica de una planta hidroeléctrica. Este proyecto forma parte de un proyecto más amplio para desarrollar un entorno de simulación para MLD complejos sistemas de tiempo real que debe cumplir con ciertos requisitos de desempeño [7, 8]. La arquitectura propuesta debe simular el funcionamiento del sistema de modelo a través de la ejecución de un modelo MLD que

representa el sistema en los distintos niveles de abstracción [9]. También debe ser compatible con la ejecución de varios algoritmos de control de la estrategia que se puede evaluar el desempeño del modelo (utilizando los métodos tradicionales de los esquemas clásicos y de adaptación para evaluar el rendimiento de estos sistemas [4]). La arquitectura propuesta, junto con un procesador de interfaz front-end, que permiten al diseñador para interactuar con el modelo con el fin de explorar los límites de rendimiento. El diseño de la representación lógica digital del algoritmo descripción MLD es sólo una parte de este proyecto, otras cuestiones que también deben abordarse [12].

## 2 FPGA aplicaciones.

Ha habido algo de trabajo hecho anteriormente en la investigación de la implementación de modelos de MLD en forma de hardware. Los primeros trabajos fue realizado por Gracios et al [10] que se parecía a la asignación de Fuzzy Neural en tiempo real las estructuras de redes de Petri para FPGA arquitectura. Muñoz-Hernández et al [11] han hecho el trabajo más importante en la asignación del algoritmo de ruido Front-End de filtro a una plataforma FPGA. Este trabajo fue seguido por una exploración de los temas de cómo describir el filtro digital con el medio ambiente FDATool y su algoritmo de programación final en VHDL. A partir de ese trabajo fue hecho, ha habido avances significativos en la lógica digital programable, en particular en Field Programmable Gate Arrays (FPGAs), donde los números de puerta por chip se están acercando a 100.000, con velocidades de reloj de más de 100 MHz. Esto ha creado la oportunidad para arquitecturas de computación reconfigurable y la implementación hardware de numerosos complejos algoritmos que pueden beneficiarse de la aceleración de hardware flexible. Uno de los importantes beneficios que las tecnologías de FPGA y ASIC cuenta con más de la tecnología EPLD es la reprogramación lista de la lógica digital y sus patrones de interconexión.

En el contexto de la asignación de una descripción del modelo MLD, la ventaja de la tecnología FPGA es que los patrones de interconexión inherente a la descripción estructural MLD potencialmente puede ser muy flexible asignada a la estructura del FPGA. Debido a la densidad de la puerta cada vez mayor cada vez hay más margen para acomodar grandes bloques de funciones en un solo chip. Además, la creciente disponibilidad comercial de múltiples sistemas FPGA admite la asignación de

los modelos MLD aún mayor a los sistemas de FPGA. La principal aplicación de una FPGA MDL sería como un coprocesador unido para apoyar el control específico de modelos de actuación y las estrategias de simulación.

### 3 MLD método de descripción

El enfoque adoptado por Muñoz-Hernández et al [7] para dividir el modelo completo de la central hidroeléctrica antes de la implementación en el ASIC es identificar tres subpartes del sistema. Esto depende de conexiones si el bloqueo de los tramos divididas o funciones, o dividir a ambos lados de los bloques o funciones. Esto llevó a la decisión de adoptar los bloques a través de las interconexiones. Se ha demostrado que este método permite la transferencia de los valores de un bloque a otro, y es compatible con el manejo de las funciones de usuario en las relaciones entre cada subsistema.

El enfoque adoptado en este trabajo está fuertemente influenciado inicialmente por la necesidad de una correspondencia directa a la FPGA. Así, el modelo MLD se divide en torno a cada subsistema, es decir, "módulo" una función local se crea para cada subsistema en el modelo original, con entrada y salida de la interfaz. Por supuesto, cada bloque aparece en varias partes de cada uno de los módulos, por lo que el modelo de resultado se divide en funciones discretas que cuando se recombinan crear el modelo original (como en el enfoque de Muñoz-Hernández et al [7]).

La ventaja de este enfoque es el funcionamiento de cualquier subsistema del módulo puede ser totalmente independiente de los demás, y la operación a nivel local es determinado con base en el número de declaraciones de entrada / salida en las definiciones de bloque. Finalmente, los tres subsistemas se obtienen son: palas, bloques hidráulicos y eléctricos. La figura 1 muestra los tres sistemas.

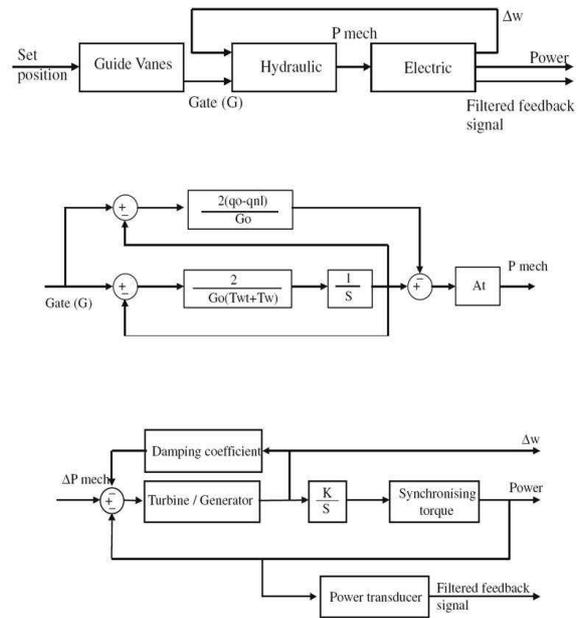


Figura 1. El subsistema de tres de la central hidroeléctrica y sus particiones.

## 2. Análisis de resultados

### 4 módulos de MLD en la FPGA

Teniendo en cuenta el método de división simple se describe en la sección anterior, el módulo de bloques funcionales extraídos del modelo de sistema MLD total de la parte hidráulica y eléctrica se muestra en la Figura 1.

Hay entradas y salidas con los enlaces correspondientes para cada bloque, pero la división final del modelo completo se hizo para obtener bloques SISO usando SIMULINK ©. Este tipo de descripción se ha aplicado y validado por Muñoz-Hernández et al [7].

Un módulo solo bloque, que sigue las reglas de ejecución de una función definida, se puede asignar a los concurrentes la operación digital de forma lógica descripción de dos; i. e. Comportamiento estructural y funcional. La primera descripción tiene el nombre del bloque y la definición de entrada / salida. En esta sección, los bits de resolución para cada puerto variable puede ser definida, y la segunda parte tiene la información sobre la arquitectura que es determinada por el comportamiento lógico. Mezcla de lógica dinámica (MLD) los sistemas

pueden ser descritos por ecuaciones lineales dinámicos sujetos a lineal entera mixta desigualdades. Estas desigualdades pueden implicar continuos y binarios (lógico) variables. MLD es capaz de representar una gran variedad de sistemas, por ejemplo: sistemas híbridos, las máquinas de estados finitos, limitados sistemas lineales y sistemas no lineales mediante funciones lineales a trozos para representar a las no-linealidades.

Teniendo en cuenta los supuestos antes, los módulos de primaria en la aplicación de FPGA se muestra en la figura. 2. No se muestra en la figura. 2 es el funcionamiento de la lógica de control, pero esto puede ser representado por la especificación de pseudo-(lo que representa el algoritmo de MLD) en MATLAB lenguaje en la Tabla 1.

### 5 FPGA La implementación del modelo MLD

La forma actual de la aplicación FPGA de la modelo de la planta hidroeléctrica de MLD se evalúa el uso de herramientas añadido en MAX-PLUS II de Altera Corp. se muestra la evaluación del desempeño en la figura 4 teniendo en cuenta el tiempo de retardo caminos rendimiento. El módulo de descripción MLD se muestra en la Tabla 2 y 3.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;use
IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity hydromod is
```

```
architecture Behavioral of hydromod is
begin
process (U,clk,Go)
-- Internal variables definition
variable yt1:std_logic_vector(7 downto 0);
variable yt2:std_logic_vector(7 downto 0);
variable yt3:std_logic_vector(7 downto 0);
variable yt4 :std_logic_vector(7 downto 0);
variable C:std_logic_vector(7 downto 0);
IF (clk'EVENT AND clk = '1') THEN
if U="0000" then tf:="00000000"; end if;
if (U="1000" or U="0100" or U="0010" or U="0001") then
if Go="11" then
C:= "00000001";--C=-0.00
d4:= "10010001";--d4=2.84
d3:= "10011010";--d3=-3.02
d2:= "01001001";--d2=1.42
d1:= "00001101";--d1=-0.25
tf:= d4*yt1-d3*yt2+d2*yt3-d1*yt4-C;
end if;
if Go="10" then
...
end process; end Behavioral;
```

La respuesta del módulo del MLD-FPGA se muestra en la figura. 4. Que representa la evaluación inicial para calcular los valores para el modelo.

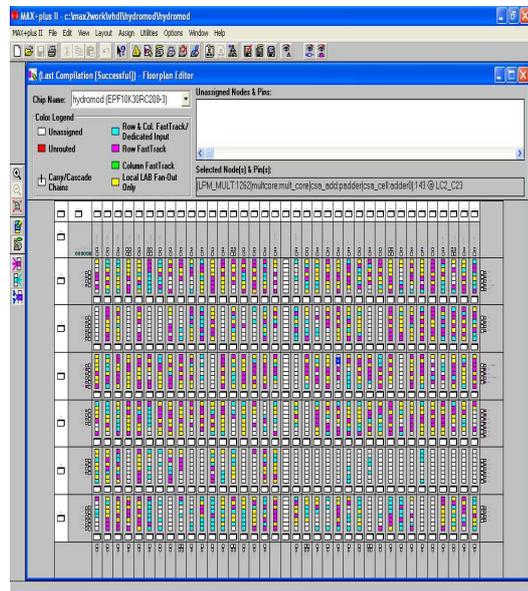


Figura 2. MLD-FPGA de aplicación del modelo de hidroeléctricas

### 6 Front-End de prueba del programa

El propósito de la evaluación del programa de front-end proporciona MAX-PLUS medio ambiente es para permitir una estructura MLD para introducir y editar gráficamente, especificando las interconexiones, las ocupaciones de bloque y las interconexiones de enlace (ver figura 3). La estructura se divide en segmentos en los tres módulos y se descarga en la FPGA para su ejecución. Al término de un ciclo de ejecución FPGA, los datos actualizados son devueltos a los bloques internos para su visualización. La ejecución sólo se detiene (además de la intervención del usuario), después de todo el tiempo de evaluación han sido evaluadas.

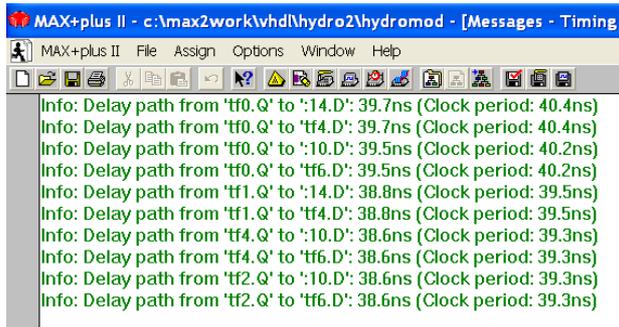
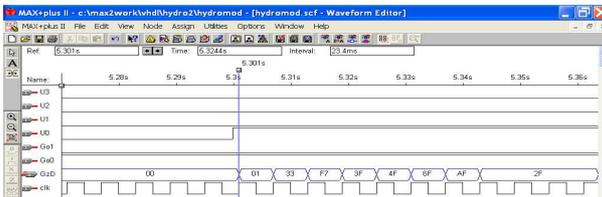


Figura 3. Caminos demora la evaluación.

### 7 Resultados

Asignación típica para la realización de una partición, el lugar y la ruta (PPR), y el% de ocupación bloques lógicos celular (CLBs) del diseño FPGA se muestran en la Tabla 4. Esto se realizó mediante el FLEX 10K FPGA de ALTERA (que cuenta con 13.000puertas lógicas). De la nota es entonces cuando se implementan tres módulos (como se muestra en la fig. 2), el FPGA es muy utilizado y el tiempo de PPR es sustancial.

Figura 4. La evaluación inicial del modelo en el FPGA.



El tiempo de ejecución en la FPGA para el modelo se muestra en la figura. 5. Estiempo de 43 ms aproximar con una operación de control FPGA. El modelo completose evalúa al mismo tiempo en el en este momento. Esta medida incluye la descarga de datos y extracción de datos pero no el tiempo de configuración de FPGA(configuración se realiza sólo una vez antes de simulación del modelo se inicia). ElFPGA seleccionado es capaz de funcionar mucho más rápido que esto, pero el uso delreloj de bus del PC y el oscilador de reloj de a bordo era conveniente para su uso en lafase inicial de prueba.

Para el desarrollo de software de la tarjeta, aunque sólo la carga de datos FPGA y los gastos de recuperación se incluyen, un módulo de modelo de tiempo de ejecución de80 ms en MATLAB se

obtiene. Con esta simple comparación, es posible demostrar la viabilidad del sistema de FPGA para evaluar el modelo de hidroeléctricas para un análisis más complejo y las aplicaciones reales. El uso final del hardware es de 58%, lo cual es suficiente para incorporar el algoritmo de control en la arquitectura mismaFPGA para el trabajo futuro.

Tabla 4. La facilidad de uso de las matrices de FPGA para sintetizar el modelo

Total dedicated input pins used:	6/6	(100%)
Total I/O pins used:	9/96	( 9%)
Total logic cells used:	1208/1728	(69%)
Total embedded cells used:	0/96	( 0%)
Total EABs used:	0/6	( 0%)
Average fan-in:	3.34/4	(83%)
Total fan-in:	4039/6912	(58%)

### Referencias

[1] A. Bemporad and M. H. Morari, "Control of systems integrating logic, dynamics, and constraints." *Automatica*, vol. 35, pp. 407-427, 1999.

[2] A. Bemporad, F. D. Torrisi, and M. Morari, "Performance Analysis of Piecewise Linear Systems and Model Predictive Control Systems", *Proc Conference on Decision and Control*, Sydney, Australia, 2000.

[3] S. P. Mansoor, D. I. Jones, D. A. Bradley, F. C. Aris, and G. R. Jones, "Reproducing oscillatory behaviour of a hydroelectric power station by computer simulation", *Control Engineering practice*, vol. 8, pp. 1261-1272, 2000.

[4] German A. Munoz-Hernandez, Carlos A. Gracios-Marin, Alejandro Diaz-Sanchez, Saad P. Mansoor and Dewi I. Jones (2009). *Neural PDF Control Strategy for a Hydroelectric Station Simulator*, Automation Control - Theory and Practice, A D Rodić (Ed.), ISBN: 978-953-307-039-1, INTECH, Available from: <http://sciyo.com/articles/show/title/neural-pdf-control-strategy-for-a-hydroelectric-station-simulator>

[5] S. P. Mansoor, D. I. Jones, D. A. Bradley, F. C. Aris, and G. Jones, "Stability of a pumped storage hydropower station connected to a power system", *Proc IEEE Power Eng. Soc. Winter Meeting*, New York, 1999.

[6] C.A. Gracios-Marin, G.A. Munoz-Hernandez, A. Diaz-Sanchez, P. Nuno-de-la-Parra, J. Estevez-Carreón and Carlos Vega-Lebrúm, Recursive decision-making feedback extension (RDFE) for

fuzzy scheduling scheme applied on electrical power control generation, [International Journal of Electrical Power & Energy Systems Volume 31, Issue 6](#), Pages 237-242, July 2009

[7] Munoz-Hernandez, G.A. Jones, D.I. Fuentes-Goiz, S.I. **Modelling and Simulation of a Hydroelectric Power Station Using MLD, Electronics, Communications and Computers, 2005. CONIELECOMP 2005. Proceedings. 15th International Conference on** Issue Date: 28-02 Feb. 2005 **On page(s):**83-88 **Print ISBN:**0-7695-2283-1, IEEE **Digital Object Identifier:** [10.1109/CONIEL.2005.49](#) **Date of Current Version:** 01 august 2005

[8] Munoz-Hernández, GA, Gracios-Marin, CA, Diaz-Sanchez, A, Mansoor, SP, Jones, DI, *Implementation of a mixed logical dynamical model of a hydroelectric power station in a FPGA*, 2<sup>nd</sup> International Conference on Electronic Design, Veracruz, Mexico, 2006.

[9] CHRISTOS G. CASSANDRAS and STÉPHANE LAFORTUNE, Introduction to Discrete Event

Systems, Second Edition, Springer Science +Business Media LLC, 2008 ISBN 978-0-387-33332-8.

[10] C. A. Graciós Marín, E. Vargas Soto, A. Díaz Sánchez, Describing an IMS by a FNRTPN definition: a VHDL approach, Robotics and CIM, Vol. 21, Issue 3, Elsevier Junio 2005.

[11] Munoz-Hernandez, G.A. Tejada-Munoz, G. Fernandez-Tellez, A. Gracios-Marin, C.A. Diaz-Sanchez, A. **Implication of Electric Noise in the ACORDE Front-End Electronics of the ALICE Experiment, Electronics, Communications and Computers, 2008. CONIELECOMP 2008, 18th International Conference on** Issue Date: 3-5 March 2008 **On page(s):**148-153 **Location:**Puebla **Print ISBN:**978-0-7695-3120-5 **INSPEC Accession Number:**9896268 **Digital Object Identifier:** [10.1109/CONIELECOMP.2008.9](#) **Date of Current Version:** 12 march, 2008

[12] A FPGA Implementation of Model Predictive Control. K.V. Ling, S.P. Yue and J.M. Maciejowski. ACC 2006.”