

# DISEÑO Y CONSTRUCCIÓN DE UN CONTROLADOR LÓGICO PROGRAMABLE BASADO EN UN NUCLEO RECONFIGURABLE

*Snaider Carrillo L., Agenor Polo Z., Mario Esmeral P*

Grupo de Robótica y Sistemas Inteligentes  
Fundación Universidad del Norte, Barranquilla – Colombia  
Tel: +575 3509509, web: <http://www.uninorte.edu.co>

[scarrillo@uninorte.edu.co](mailto:scarrillo@uninorte.edu.co), [apolo@uninorte.edu.co](mailto:apolo@uninorte.edu.co), [mesmeral@uninorte.edu.co](mailto:mesmeral@uninorte.edu.co)

## RESUMEN

El propósito del presente documento es mostrar el procedimiento seguido para el diseño y construcción de un Controlador Lógico Programable (PLC), compatible con el lenguaje de programación Lista de Instrucciones (IL) según la norma IEC 61131-3 y con capacidad de ser programado de forma local a través de la interfaz RS-232 y de forma remota a través de una red de área local mediante el estándar IEEE 802.3.

El Controlador Lógico Programable presentado en este documento, a diferencia de los PLC convencionales, cuenta con el núcleo de sistema descrito a manera de microprocesador empotrado haciendo uso del lenguaje de descripción de hardware VHDL y sintetizado en un dispositivo de lógica programable FPGA. Para la interfaz de usuario se implementó el software OPENPCS 2004, el cual contempla en un 100% los lenguajes de programación establecidos en la norma IEC 61131-3.

Finalmente se muestran comparaciones del desempeño del Controlador Lógico Programable construido frente a un controlador disponible de manera comercial para uso en un entorno industrial.

## 1. INTRODUCCIÓN

De acuerdo con la definición de la NEMA (*National Electrical Manufacturers Association*) un Controlador Lógico Programable es: "Un dispositivo electrónico operado digitalmente, que usa una memoria programable para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas para controlar, a través de módulos de entrada/salida digitales o analógicos, varios tipos de máquinas o procesos" [1].

El controlador Lógico Programable (PLC) ha progresado rápidamente desde su inserción en el campo

industrial en la década de los 60 [2]. Originalmente fue desarrollado como un dispositivo de control secuencial, para reemplazar a los circuitos electromecánicos basados en relés y desde entonces es usado ampliamente en la automatización de procesos industriales.

Al mismo tiempo que la automatización avanza a pasos agigantados, los procesos a controlar en la planta por medio del PLC se vuelven mucho más complejos [3]. Esto causa que los programas que hay que implementar demanden mayores recursos hardware del sistema y el "scan time" se convierta en un factor clave a la hora de diseñar un PLC; sin olvidar que hay que manejar un mayor número de puertos de entrada/salida y diferentes tipo de protocolos de comunicación que permitan la interacción del PLC con otro tipo de dispositivos.

Una manera de satisfacer estos requerimientos es recurrir al uso de herramientas HDLs, las cuales han facilitado la construcción de sistemas empotrados reconfigurables, que encuentran su aplicación en campos específicos como las comunicaciones y la automatización industrial [2,4].

Además de la reconfigurabilidad, los HDLs y los dispositivos de lógica programable (CPLDs, FPGAs), han permitido reducir los costos de construcción de prototipos y salida al mercado del producto final; estimulando de esta manera la iniciativa de aquellos grupos de investigación y desarrollo que poseen recursos económicos limitados.

El Grupo de Robótica Móvil y Sistemas Inteligentes de la Universidad del Norte ha construido un microprocesador empotrado de propósito específico, que sirve como núcleo de procesamiento para un Controlador Lógico Programable (PLC), compatible con el lenguaje de programación IL (*Instructions List*) según la norma IEC 61131-3 [5]; el cual fue implementado dentro de una FPGA mediante el lenguaje de descripción de hardware VHDL [6,7]; además se cuenta con un módulo de red compatible con el estándar IEEE 802.3, que contiene además de forma

hardware la pila de protocolos TCP/IP, y es utilizado para programar el dispositivo a través de una LAN.

El propósito de este trabajo es mostrar la manera en que un microprocesador puede configurarse en un PLD para aprovechar herramientas software de alto nivel existentes, reduciendo el nivel de esfuerzo requerido para satisfacer los requerimientos HW/SW de un PLC. El siguiente artículo presenta de manera breve y concisa los tópicos fundamentales concernientes al diseño y construcción del Controlador Lógico Programable basado en un núcleo reconfigurable, a saber: metodología de diseño, pruebas, resultados y conclusiones.

## 2. METODOLOGÍA DE DISEÑO

El sistema desempeña la función de un Controlador Lógico Programable (PLC), es decir, de acuerdo con un programa almacenado en memoria, efectúa el control de un proceso de interés para el usuario, con base en información leída de los puertos de entrada y del estado actual de los puertos de salida.

El usuario dispone de un software de interfaz ejecutable desde un PC, el cual le permite simular y compilar el programa de control así como también llevar a cabo la programación del dispositivo.

### 2.1 Diagrama de Bloques del Sistema

*Bloque de Interfaz de Usuario.* Este es el encargado de proporcionar al usuario las herramientas para escribir y compilar el programa de control utilizando el lenguaje lista de instrucciones, según la norma IEC 61131-3. Además este bloque también se encarga de la programación del PLC.

*Bloque de Módulos de Entrada/Salida (E/S).* Los bloques de entrada/salida, son circuitos mediante los cuales el sistema puede comunicarse con el mundo exterior. Poseen una interfaz eléctrica que se encarga de adecuar las señales de entrada y salida, de tal modo que se puedan acoplar a los dispositivos a los cuales se conectan.

*Bloque de Alimentación.* Genera las fuentes de tensión de los diversos dispositivos en el sistema. Se divide en varias etapas: una encargada de la conversión AC-DC y luego de esta múltiples etapas de DC-DC para obtener los distintos niveles de voltaje requeridos por el sistema.

*Bloque de Memoria Externa no Volátil.* La función de este bloque es la de almacenar el programa en ausencia de alimentación.

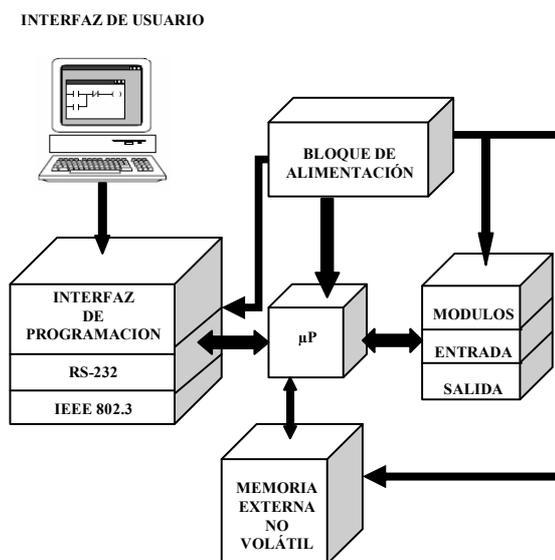


Figura 1. Representación en diagrama de bloques funcionales del PLC.

*Bloque de Interfaz de Programación.* La tarea de este bloque funcional consiste en permitir la comunicación e interacción del Controlador Lógico Programable con el dispositivo de programación (ejemplo: computadora personal (PC)), el cual se encargará de enviar el programa previamente compilado.

- *RS 232:* Se empleara como interfaz para llevar a cabo la comunicación serial del Controlador Lógico Programable con el PC, la cual permitirá la programación local del dispositivo.
- *IEEE 802.3:* Se empleara como interfaz para llevar a cabo la comunicación remota del Controlador Lógico Programable con el PC a través de una red LAN.

*Bloque Microprocesador (μP).* Este es el elemento más complejo del Controlador Lógico Programable. Se divide en dos partes principales, cada una con funciones diferentes:

- *Unidad de control:* rige el comportamiento de todos los demás componentes, y lleva el control sobre la secuencia de ejecución del programa.
- *Datath:* Se compone de cada una de las unidades funcionales de las que se vale el microprocesador para la manipulación de datos y la realización de operaciones lógicas y aritméticas.

## 2.2. Diagrama de Flujo del Sistema

- *Inicialización de variables y señales.* Se establecen los valores iniciales de las variables, señales y bloques internos/externos al microprocesador que así lo requieran. Ejemplo: contador de programa, apuntador de pila, banderas, contadores, temporizadores, etc.
- *Configuración e inicialización del modulo 802.3.* Se configuran los parámetros del modulo 802.3, para trabajar en una red LAN y se envían los correspondientes comandos de inicialización.
- *Reserva de la cache para datos y programa.* El contenido de información almacenado en el dispositivo no volátil es copiado a la *cache* interna del núcleo.
- *Lectura de entradas al shadow register.* Se leen los valores del puerto de entrada del sistema y se almacenan en el registro imagen (*shadow register*), de tal forma que estos permanezcan estáticos durante la ejecución del programa.
- *Ejecutar programa.* Se obtiene las instrucciones de la memoria *cache*, se decodifican y se ejecutan. Aquí también se actualizan las salidas de acuerdo con el programa en cuestión.

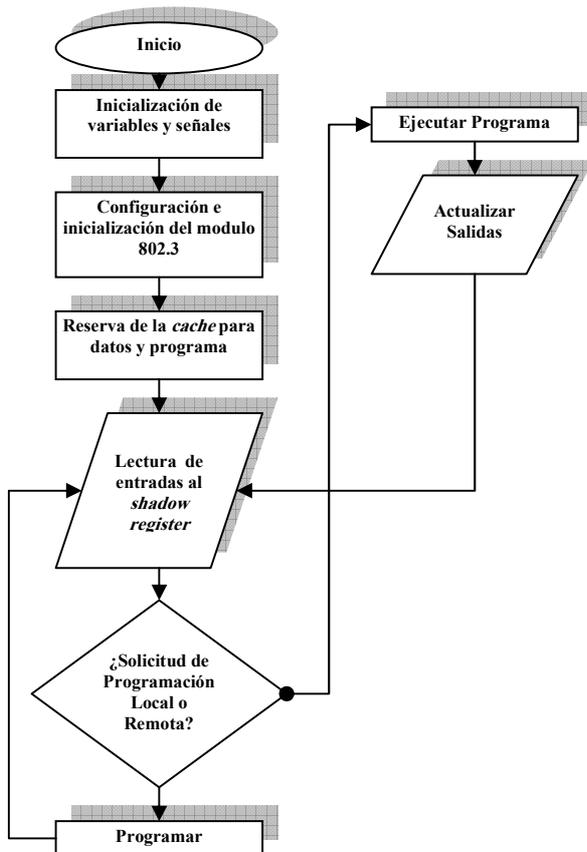


Figura 2. Diagrama de flujo del funcionamiento del PLC.

## 2.3. Núcleo del Sistema [8]

El Controlador Lógico Programable presentado en este documento, a diferencia de los PLC convencionales, cuenta con el núcleo de sistema descrito a manera de microprocesador empotrado haciendo uso del lenguaje de descripción de *hardware* VHDL y sintetizado en un dispositivo de lógica programable FPGA. Este microprocesador empotrado fue desarrollado por los autores y cumple con un subconjunto de instrucciones, tipos de variables y bloques funcionales definidas en la norma IEC61131-3. (Ver Tabla 1)

Tabla 1. Instrucciones, tipos de variables y bloques funcionales soportados por el sistema.

TIPOS DE VARIABLES		
BOOL		
INT		
TIME		
Cantidad	BLOQUES FUNCIONALES	
2	Temporizador tipo TOF	
2	Temporizador tipo TON	
4	Contadores descendentes CTD	
OPERADOR	MOD	OPERANDO
LD	N	Variable
ST	N	Variable
S		Variable Booleana
R		Variable Booleana
NOT	(	Variable Booleana
AND	N,(	Variable Booleana
OR	N,(	Variable Booleana
XOR	N,(	Variable Booleana
ADD	(	Variable TIPO ENTERO O TIME
SUB	(	Variable TIPO ENTERO O TIME
GT	(	Variable
GE	(	Variable
EQ	(	Variable
LE	(	Variable
LT	(	Variable
JMP	C, N	Etiqueta
RET		
CAL	C, N	Bloques funcionales
)		

Donde el modificador (Mod):

N => Se utiliza para la negación (*NOT*) de un operando.

( => Indica que la operación será postergada.

C => Indica una operación Condicional.

Variable Booleana => 8 bits

Variable Entera => 16 bits

Variable Time => 32 bits

### 2.3.1. Datapath

El microprocesador empotrado se vale del *datapath* para ejecutar cada una de las instrucciones estipuladas anteriormente, para esto se utilizan diversos bloques especializados, entre los cuales se encuentran:

Un *latch* de 16 bits, el cual almacena los valores de 16 entradas digitales al sistema. El contenido de este *latch* es refrescado antes de cada ejecución del programa.

Un multiplexor que permite seleccionar, entre entradas de distintos tamaño, uno de los operandos de la ALU. La salida del multiplexor es fija de 32 bits.

Una función *NOT* de 32 bits con habilitación. Esta es utilizada en los casos en que se requiere operar con la inversión lógica del operando.

Una ALU de 32 bits, la cual es capaz de realizar las operaciones lógicas and, or, xor; comparaciones relativas >, <, >=, <=, =; aritmética de punto fijo: suma y resta. Esta también incluye una operación *pass* o *buffer* que toma la entrada que viene del multiplexor y la coloca directamente a la salida de la ALU.

Internamente se dispone de una memoria *cache* de 8K x 8 bits para el almacenamiento tanto del programa como de los datos, distribuidos en dos memoria empotradas doble puerto de 4K x 8 bits cada una, de la siguiente forma (ver Figura 3):

Se realizó un direccionamiento de memoria haciendo uso del bit menos significativo de la dirección de 13 bits, obteniéndose como resultado final que una de las memorias quedara cargada con posiciones pares y la otra almacenara las posiciones impares tanto de programa como de datos.

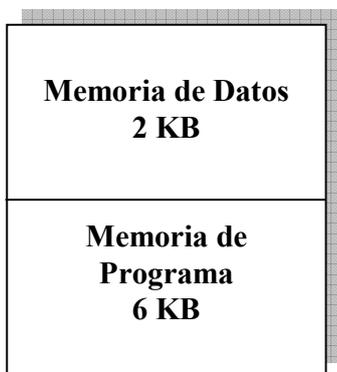


Figura 3. Mapa de memoria de la memoria *cache*.

La realización del anterior direccionamiento fue debido a que de esta manera se puede acceder a cuatro locaciones contiguas del mapa de manera simultánea.

Un registro acumulador de 32 bits que sirve para almacenar datos parciales. Este es siempre uno de los operandos de la ALU.

Una pila con profundidad de 256 palabras de 32 bits. Esta se implementó a manera de memoria empotrada usando los bits de memoria disponibles de la FPGA [9] para evitar el consumo de elementos lógicos por parte de esta. El direccionamiento de esta memoria es administrado por la unidad de control de tal manera que esta se comporta como una memoria LIFO.

Una unidad de conteo y temporización, la cual internamente está compuesta de cuatro contadores descendentes de 16 bits y cuatro temporizadores de 32 bits (2 *TON* y 2 *TOF*).

Una segunda función *NOT* con habilitación, esta vez de un bit; permite sacar al exterior la inversión lógica del *LSB* del acumulador.

Una memoria de 8 x 1 bit con salida de 8 bits funciona como registro de almacenamiento para las 8 salidas digitales del sistema. Estos 8 bits se retroalimentan al multiplexor de entrada para que puedan ser utilizados en otras operaciones si el programa así lo requiere.

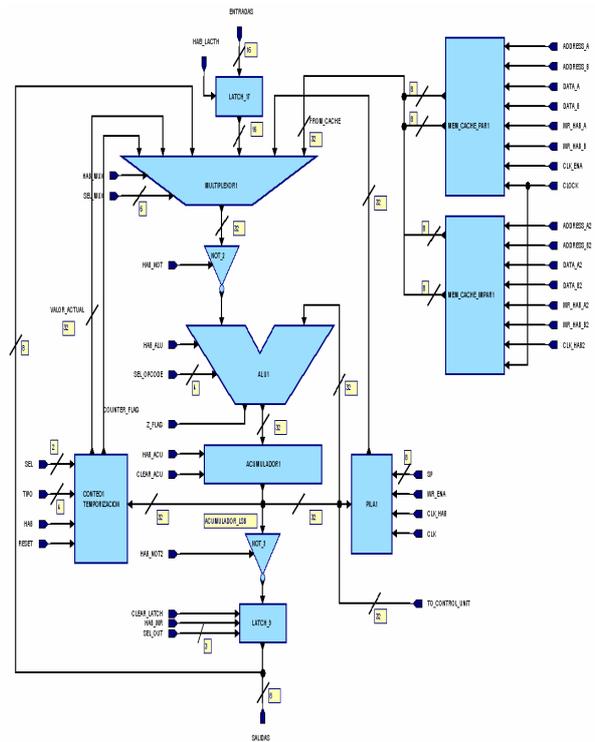


Figura 4. Diseño del Datapath implementado para la CPU del PLC.

### 2.3.2. Unidad de Control

La unidad de control es una máquina de estados finita (FSM) que gobierna cada uno de los modos de operación del microprocesador y por esta razón se considera el bloque más importante de este [10]. En la Figura 5, se muestra el diagrama de estados que describe el flujo de los modos de operación del microprocesador.

- On\_state (S\_0): Este es el estado inicial, aquí se le asignan los valores iniciales a las señales de control que así lo requieran.
- Reserva\_memoria (S\_1): En este estado se transfiere el contenido de información del dispositivo de almacenamiento no volátil hacia la memoria *cache* interna. Esto con el fin de lograr una ejecución más eficiente del programa.
- Scan\_input (S\_2): En este estado se actualizan las entradas hacia el PLC antes y después de ejecutarse el programa.
- Run (S\_3): Este estado se refiere a la ejecución propiamente dicha del programa, aquí se obtiene cada una de las instrucciones de la memoria *cache*, se decodifican y ejecutan.
- Programar\_RS232 (S\_4): En este estado se obtiene los datos provenientes del puerto serial y se almacena en el dispositivo de almacenamiento no volátil.
- Programar\_802.3 (S\_5): En este estado se obtiene los datos provenientes del puerto IEEE 802.3 y se almacena en el dispositivo de almacenamiento no volátil.

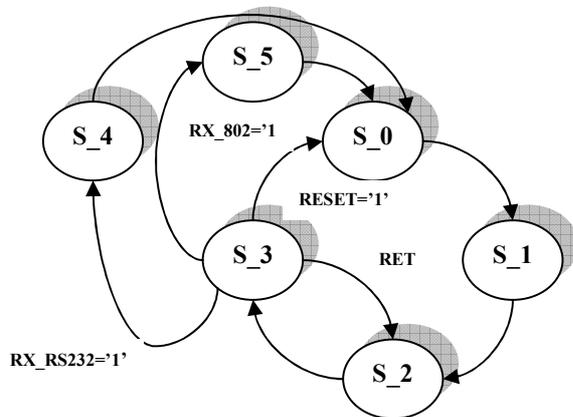


Figura 5. Diagrama de estados de la unidad de control de PLC.

### 3. SÍNTESIS E IMPLEMENTACIÓN

En esta etapa la sintetización del microprocesador se realizó sobre una FPGA Altera Cyclone EP1C6T144C6 [11], el cual opera a una frecuencia de 25MHz; además se utilizaron optoacopladores PS2845A los cuales se encargan de proporcionar la interfaz eléctrica adecuada para interactuar con el microprocesador (3.3V y 1.5V) y

los dispositivos externos al sistema (24V); una memoria AT24C32A EEPROM serial (2-wired serial) de 8K x 8 bits, la cual es la encargada de guardar una copia del programa a ejecutarse en el Controlador Lógico Programable; así como también un MAX232 y un modulo externo NM7010A [12] que incorpora la pila de protocolos TCP/IP en hardware y es compatible con el estándar IEEE 802.3 en el sub-nivel MAC y la capa física. (Ver figura 6)

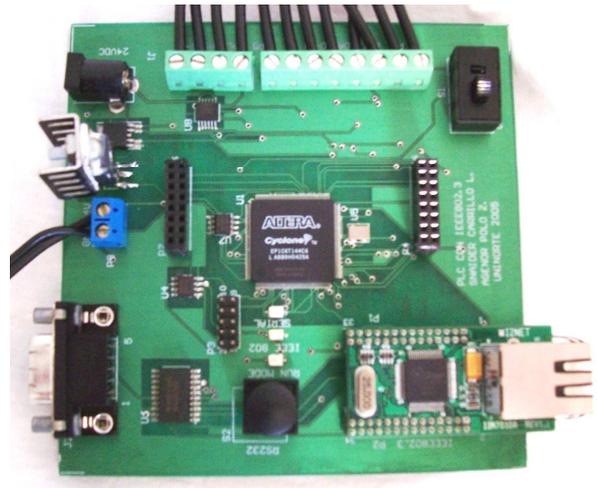


Figura 6. Plataforma final del Controlador Lógico Programable.

### 4. PRUEBAS Y RESULTADOS

Se llevó a cabo un protocolo de pruebas para validar el correcto funcionamiento del Controlador Lógico Programable. Este incluyó un banco de programas de pruebas en los cuales se cubrió cada uno de los distintos tipos de instrucciones soportadas por el sistema, esto es: instrucciones lógicas, aritméticas, comparativas, saltos condicionales y no condicionales, así como las correspondientes a los bloques especializados de conteo y temporización [13].

Para la compilación de los programas de prueba se empleó el software OPENPCS 2004 desarrollado por la empresa Infoteam [14], el cual provee soporte de 100% para cada uno de los lenguajes soportados por la norma IEC61131-3, entre los cuales se encuentra: lista de instrucciones (IL), diagrama escalera (LD), texto estructurado (ST) y bloques funcionales (FB).

Teniendo en cuenta la capacidad de E/S (16 entradas y 8 salidas), este PLC cae dentro de los controladores programables de gama baja; por esto, se han tomado como referencia un controlador de la serie SIMATIC S7 de la empresa Siemens [15] que posee características similares con el objetivo de realizar comparaciones.

Los parámetros tenidos en cuenta al momento de llevar a cabo la comparación entre los controladores fueron: tiempo de ejecución, números de contadores y temporizadores, números de puertos E/S, interfaces de programación y capacidad de memoria.

TABLA 2  
TABLA COMPARATIVA ENTRE LA CPU 215 Y EL NÚCLEO DESARROLLADO EN LA FPGA

Parámetros	Núcleo del sistema	
	CPU 215	FPGA
Tiempo de ejecución para instrucción lógica	0.8 $\mu$ s	0.2 $\mu$ s
Tiempo de ejecución para instrucción aritmética de punto fijo	1.6 $\mu$ s	0.2 $\mu$ s
Tiempo de ejecución para instrucción carga y almacenamiento	0.8 – 3.2 $\mu$ s	0.8 $\mu$ s
Numero de contadores	256	4
Numero de temporizadores	256	4
Numero de E/S	14/10	16/8
Interfaces de programación	RS-232	RS-232/ IEEE802.3
Capacidad de memoria	6.5 KB	8KB

De las cifras mostradas en la tabla 2, puede notarse que el Controlador Lógico Programable construido mejora el de la empresa Siemens en la velocidad de ejecución del programa, capacidad de memoria y en que presenta la interfaz IEEE 802.3 para la programación. En cuanto al número de bloques de conteo y temporización, el ejemplar de Siemens supera ampliamente al construido en este proyecto, pero dadas las características de la plataforma donde se implementó el PLC, es posible ampliar esta cifra eligiendo una FPGA con mayor cantidad de elementos lógicos.

## 5. CONCLUSIONES

El Controlador Lógico Programable ejecuta exitosamente cada una de las instrucciones del lenguaje *IL* planteadas como objetivo. Esto se deduce de los resultados, ya que el controlador ejecuto satisfactoriamente cada uno de los programas del banco de prueba.

Las interfaces de programación local y remota disponibles en el prototipo final, funcionan de manera satisfactoria dentro de los parámetros establecidos. El dispositivo aprobó cada una de las pruebas efectuadas a la interfaz RS-232, así como también a la IEEE 802.3

El núcleo del sistema es reconfigurable. Al estar implementado en un dispositivo de lógica programable (FPGA), el hardware puede ser modificado cambiando la descripción de este en VHDL y sintetizando nuevamente en el dispositivo.

Existe redundancia en el *set* de instrucciones propio del microprocesador. Esto se debe a que el código de máquina generado por el *software* OPENPCS 2004 esta optimizado

para un microprocesador de 8 bits; asignándose distintos códigos cuando la operación implica operandos de 8, 16 y 32 bits. Debido a que la ALU implementada es capaz de manejar operandos de 32 bits dichos códigos convergen en una misma operación.

No fue necesario la implementación de la pila de protocolos TCP/IP en forma *software*, al disponer del módulo de red NM7010A, el cual incorpora de forma *hardware* dichos protocolos.

## 6. TRABAJOS FUTUROS

Analizar el comportamiento y rendimiento del Controlador Lógico Programable bajo las condiciones de un entorno industrial.

Desarrollar un compilador que aproveche las características propias del microprocesador empotrado, soportando tanto el lenguaje de programación *IL* como *Ladder*. Con esto se aspira a eliminar la redundancia en el *set* de instrucciones.

Comparar el rendimiento de la arquitectura del microprocesador empotrado mostrada en el documento, con otros microprocesadores empotrados con *Pipelined* incorporado [2].

## 7. REFERENCIAS

- [1] WRIGHT, J: "The Debate Over Which PLC Programming Language is the State of the Art", Journal of Industrial Technology on Volume 15 number 4, Agosto 1999.
- [2] Oh. Yang, "The design of a 32-bit Microprocessor for a Sequence Control using an Application Specification Integrated Circuit (ASIC)" The International Conference on Electronics, Information, AND Communications, 2004. ICEIC'04., 1st International Conference on, Volume: 1, pp.79-83, 2004.
- [3] K.H. Koo, G.S. Rho, and W.H. Kwon, "An Architecture of the RISC Processor for Programmable Controllers" Industrial Electronics, Control and Instrumentation, 1994. IECON '94. 20th International Conference on , Volume: 2 , pp.1179-1183, 1994.
- [4] J. Viñas, N. Díaz, y H. Campanella, "Modem Bandabase (Nivel Físico) Basado en el Estándar IEEE 802.11b", Internacional Conference on Reconfigurable Computing and FPGA, México, Septiembre 20-21, 2004, pp. 1-10.
- [5] K.H. John, y M. Tiegelkamp, *IEC 61131-3: Programming Industrial Automation Systems*, Springer-Verlag, Berlin, 2001
- [6] CARLSON, Steve. Introduction to HDL-Based Design Using VHDL. United States: Synopsys, Inc., 1990. 555 p.
- [7] GARCÍA, Antonio: Técnicas y tecnologías de diseño de alto nivel: VHDL 2 Ed. Bogotá: Universidad de los Andes, 2001. 170 p.

[8] S. Carrillo, M. Esmeral, y A. Polo, "Design and Implementation of an Embedded Microprocessor Compatible With IL Language in Accordance to the Norm IEC 61131-3", Internacional Conference on Reconfigurable Computing and FPGA, México, Septiembre 28-30, 2005.

[9] J. Mattos y L. Carro, "Efficient Architecture for FPGA Based Microcontrollers", Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on Volume 5, 26-29 May 2002 Page(s):V-805 - V-808 vol.5

[10] E. Hwang, *Digital Logic and Microprocessor Design with VHDL*, La Sierra University, Riverside, 2004.

[11] ALTERA Corporation: *Cyclone™ Device Handbook*. Disponible en: <http://www.altera.com/literature/lit-cyc.jsp> (8 Febrero 2005).

[12] WIZNET Corporation. *IINCHIP W3100A Technical Datasheet v1.32*. Disponible en: <http://www.iinchip.com/> (18 Mayo 2005).

[13] A. Berger, *Embedded Systems Design: An Introduction to Processes, Tools, and Techniques*, CMP Books, 2001.

[14] INFOTEAM: OPENPCS 2004. Disponible en: <http://www.infoteam.de/index.php?id=26>, (31 Mayo 2005).

[15] Catalogo de productos de Siemens. Disponible en: [http://www4.ad.siemens.de/virlib/doc\\_78/s7/S7-200\\_s.pdf](http://www4.ad.siemens.de/virlib/doc_78/s7/S7-200_s.pdf) (18 Agosto 2005).