

**DESARROLLO DE UNA ESTACIÓN CENTRAL DE UN SISTEMA SCADA A
TRAVÉS DE UNA FPGA.**

ANDRÉS MARÍN HERNÁNDEZ

**UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA
MEDELLÍN**

2009

**DESARROLLO DE UNA ESTACIÓN CENTRAL DE UN SISTEMA SCADA A
TRAVÉS DE UNA FPGA.**

ANDRÉS MARÍN HERNÁNDEZ

Trabajo de grado para optar al título de Ingeniero Electrónico

Director

JUAN CARLOS VÉLEZ GALLEGO

Ingeniero Electrónico con MSC(c) en Ingeniería de sistemas

UNIVERSIDAD PONTIFICIA BOLIVARIANA

FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA

MEDELLÍN

2009

Nota de aceptación

Presidente del jurado

Jurado

Jurado

Medellín, Octubre 21 de 2009.

DEDICATORIA

Este trabajo está dedicado a mi madre quien realizó un enorme esfuerzo para que yo terminara mi carrera con éxito y naciera un nuevo ingeniero.

AGRADECIMIENTOS

El autor expresa sus agradecimientos a Juan Carlos Vélez, Fabio Guzmán, ambos por su ayuda en la consecución del presente trabajo de grado, a la empresa Variadores S.A. por facilitar los equipos pertinentes y su asesoría con los mismos, a la Universidad Pontificia Bolivariana y el equipo de docentes que contribuyeron en la formación profesional durante toda la carrera, a los compañeros y amigos de la universidad quienes hicieron del arduo estudio de la ingeniería una situación mucho más agradable y llevadera.

CONTENIDO

INTRODUCCIÓN	14
OBJETIVOS	16
1. SISTEMA SCADA	17
2. FPGA	22
2.1. DESCRIPCIÓN	22
2.2. PRINCIPIO DE FUNCIONAMIENTO	24
2.2.1. CLB	25
2.2.2. IOB	27
2.2.3. BLOCKRAM	29
2.2.4. Multiplicadores dedicados	30
2.2.5. DCM	32
2.3. RECURSOS UTILIZADOS	32
2.3.1. Microblaze	32
2.3.2. Xilkernel	35
3. VARIADOR DE FRECUENCIA	38
3.1. DESCRIPCIÓN	38
3.1.1 Motor de Inducción	39
3.2. FUNCIONAMIENTO DEL VARIADOR DE FRECUENCIA	40
3.3. INTERFAZ ELECTRÓNICA	42

3.4. PROTOCOLO MODBUS	43
3.4.1. Trama para lectura de registros	45
3.4.2. Trama de respuesta	47
3.4.3. Trama de escritura de registros	48
4. CONEXIÓN FPGA A INTERNET	49
4.1. Modelo OSI	49
4.1.1. Capa física	50
4.1.2. Capa de enlace	51
4.1.3. Capa de red	51
4.1.4. Capa de transporte	52
4.1.5. Capa de sesión	52
4.1.6. Capa de presentación	53
4.1.7. Capa de aplicación	53
4.2. Lwip	54
4.3. Procedimiento para la configuración de la estación central como servidor	55
4.3.1. spi_usr_app.c	56
4.3.2. Echo.c	59
5. ENTORNO GRÁFICO	63
5.1. FUNCIONAMIENTO DEL SOFTWARE	64
5.2. VISUALIZACIÓN DE DATOS	66
6. INTEGRACIÓN DEL SISTEMA	67

7. PUESTA EN FUNCIONAMIENTO	70
7.1. PRECAUCIONES CON EL VARIADOR	70
7.2. PRECAUCIONES CON LA RED FÍSICA	72
7.3. PRECAUCIONES CON CONEXIÓN A INTERNET	73
7.4. MONITOREO Y ANÁLISIS	74
7.5. CONTROL REMOTO	77
8. CONCLUSIONES	79
BIBLIOGRAFÍA	82
LISTA DE ANEXOS	86

LISTA DE FIGURAS

Figura 1. Esquema general de un sistema SCADA comúnmente visto.	23
Figura 2. Pantalla SCADA de EMABESA.	23
Figura 3. Scada para cámaras frigoríficas.	24
Figura 4. Tarjeta de desarrollo FPGA Spartan 3E.	26
Figura 5. Arquitectura general de una FPGA Spartan 3E.	27
Figura 6. Esquema general de un CLB.	28
Figura 7. Esquema general de los Slices.	29
Figura 8. Esquema general de las LUTs.	29
Figura 9. Esquema general de los multiplexores.	30
Figura 10. Esquema general de los bloques IOB.	31
Figura 11. Esquema general de los bloques de memoria RAM de puerto simple.	32
Figura 12. Esquema general de los bloques de memoria RAM de puerto doble.	33
Figura 13. Esquema general de los multiplicadores dedicados.	34
Figura 14. Esquema general de los bloques DCM.	35
Figura 15. Diagrama de bloques del procesador Microblaze.	37
Figura 16. Módulos del Xilkernel.	40
Figura 17. Variador Yaskawa modelo CMIR-F7U.	41

Figura 18. Motor de inducción	42
Figura 19. Señales de frecuencia y amplitud con PWM.	44
Figura 20. Diagrama de la codificación NRZ.	47
Figura 21. Programa para calcular el CRC-16.	49
Figura 22. Formato de trama para leer datos de registro del variador Yaskawa.	50
Figura 23. Formato de trama de respuesta del variador ante la solicitud de lectura de datos de los registros.	50
Figura 24. Formato de trama para escribir en los registros del variador Yaskawa.	51
Figura 25. Modelo OSI.	53
Figura 26. Arquitectura del sistema utilizando Ethernetlite.	57
Figura 27. Diagrama de bloques general del programa spi_usr_app.c	61
Figura 28.a. Diagrama de bloques general del programa Echo.c.	64
Figura 28.b. Diagrama de bloques general del programa Echo.c.	65
Figura 29. Página principal del entorno gráfico para el control de la estación central.	66
Figura 30. Diagrama de bloques general del programa de control de la estación central SCADA.	67
Figura 31. Gráfica de adquisición de valores de frecuencia de salida del variador Yaskawa.	69
Figura 32. Ejemplo de implementación de la FPGA en un proceso industrial.	70
Figura 33. Esquema general montaje trabajo de grado.	71
Figura 34. Esquema de conexión de un variador Yaskawa F7 a una red Modbus.	74
Figura 35. Esquema de conexión de varios variadores Yaskawa a una red Modbus.	75

Figura 36. Gráfica de adquisición de valores de frecuencia de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.	77
Figura 37. Gráfica de adquisición de valores de corriente de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.	78
Figura 38. Gráfica de adquisición de valores de velocidad de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009	78
Figura 39. Gráfica de adquisición de valores de voltaje de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.	79
Figura 40. Selección de la frecuencia de referencia para el variador Yaskawa desde el Software de control de la estación central del sistema SCADA.	81

GLOSARIO

Automatización industrial: Se refiere al uso de dispositivos presentes en la industria que realizan una secuencia de tareas específicas, minimizando la mano de obra.

Estación central: Es el nodo principal de un sistema de control (para el caso particular) el cual se encarga de manipular, monitorear las variables del proceso industrial y organizar la información para luego ser interpretada.

FPGA: (Field Programmable Gate Array) Se refiere a un dispositivo con la capacidad de realizar conexiones internas entre sus recursos mediante la programación de un usuario para implementar funciones deseadas con diversos grados de complejidad.

Protocolo: Se refiere a un conjunto de reglas y normas las cuales son concebidas por las entidades que se relacionan con el tema en cuestión.

Sistema embebido: Sistema o dispositivo que integra subsistemas con funciones específicas con el fin de implementar una aplicación mayor.

Sistema SCADA: (Supervisory Control and Data Acquisition) Se refiere a un sistema de control y monitoreo central que se implementa generalmente a un proceso industrial el cual puede contener diferentes etapas.

Variador de Frecuencia: Es un equipo electrónico que regula la frecuencia del voltaje aplicado a un motor, logrando con ello modificar su velocidad; conservando la relación de frecuencia y voltaje para no saturar el flujo magnético del rotor del motor.

SIGLAS

ALU: Arithmetic Logic Unit

ASCII: American Standard Code of Information Interchange

BRAM: Block random access memory

BSB: Base System Builder

CLB: Configurable Logic Blocks

CMOS: Complementary Metal Oxide Semiconductor

CRC: Cyclical Redundancy Check

DCM: Digital Clock Managers

DCR: Device Control Register

DFS: Digital Frequency Synthesizer

DLL: Delay-Locked Loop

EDK: Embedded Development Kit

EMAC: Ethernet Media Access Control

FPGA: Field Programmable Gate Array

HTTP: Hypertext Transfer Protocol

IOB: Input/Output Blocks

LMB: Local Memory Bus

LUT: Look-Up Table

LWIP: lightweight Internet Protocol

MFS: Memory file system

MPMC: Multi-Port Memory Controller

NRZ: Non Return to Zero

OPB: On-chip Peripheral Bus

OPC: OLE for Process Control

OSI: Open System Interconnection

PLB: Processor Local Bus

PLC: Programmable Logic Controller

PS: Phase shift

PWM: Pulse Width Modulation

RAM: random access memory

RISC: Reduced Instruction Set Computer

ROM: read-only memory

RTS: Request To Send

RTU: Remote Transmission Unit

SCADA: Supervisory Control And Data Acquisition

SDRAM: Synchronous Dynamic Random Access Memory

TCP: Transmission Control Protocol

UTP: Unshielded Twisted Pair

VHDL: Acrónimo que representa la combinación de VHSIC es el acrónimo de (Very High Speed Integrated Circuit) y HDL (Hardware Description Language)

RESUMEN

Los sistemas SCADA se implementan principalmente en ambientes industriales, en donde el constante monitoreo de las variables que intervienen en el proceso de la producción se hace indispensable, con el fin de conocer la respuesta de los diferentes dispositivos como motores, actuadores y sensores, entre otros y poder realizar ya sea una acción preventiva o correctiva si la situación lo amerita, según lo visualizado en el sistema SCADA.

El sistema embebido FPGA es una herramienta que gracias a sus capacidades puede albergar un sistema SCADA, facilitando su funcionamiento como estación central de un proceso industrial, en donde se permita controlar y monitorear las variables presentes en dicho proceso.

PALABRAS CLAVES: SCADA; FPGA; PROCESO; AUTOMATIZACIÓN INDUSTRIAL; SISTEMA EMBEBIDO; ESTACIÓN CENTRAL; CONTROLADORES LÓGICOS; MODBUS; ETHERNET; VARIADOR DE FRECUENCIA.

INTRODUCCIÓN

Los sistemas SCADA comenzaron a tener lugar en los procesos de producción industriales de mediana y baja escala. Inicialmente el control y la adquisición de datos se han hecho a través de redes de PLC capaces de comunicarse con interfaces gráficas desarrolladas en computadores. Para las empresas, la implementación de un sistema SCADA implica inversiones financieras de gran magnitud, situación que influye necesariamente en condicionamientos tecnológicos, administrativos y operativos; debido a estas altas inversiones requeridas, se hace necesaria la concepción de soluciones tecnológicas de alto desempeño a bajos costos capaz de funcionar como estación central en la implementación de un sistema de control industrial.

Las tarjetas de desarrollo FPGA tienen gran aplicabilidad, ya que permiten integrar en un solo dispositivo diversas aplicaciones de control, supervisión, adquisición y visualización de procesos industriales a un bajo costo comparado con las prestaciones que se pueden obtener de la misma.

En la actualidad las FPGA están siendo utilizadas en la robótica, procesamiento de señales, telemetría, telemedicina, vehículos inteligentes, aplicaciones militares, entre otros. Esto se debe a su versatilidad en manejo y posibilidad de aplicación, adicionalmente se es posible la comunicación por diversos puertos y protocolos, lo cual facilita su monitoreo y incrementa sus posibilidades de aplicación.

A nivel industrial la FPGA no tiene mucha acogida en Colombia, contrario a otro país Latino Americano como Argentina en donde se ha emprendido desde el 2003 una campaña para sensibilizar a las empresas sobre el uso de las nuevas tecnologías (entre esas se encuentra la FPGA) para automatizar sus procesos industriales, mejorando campos como la electrónica de potencia, alarmas, comunicaciones, control de instalaciones eléctricas automatización de procesos, etc. Convirtiendo a la FPGA

una solución tecnológica muy viable ya que la inversión económica es menor en comparación con otras opciones en el mercado como los PLCs y es una tecnología que puede adoptar tanto las grandes empresas como las PYMES gracias al fácil acceso que se tiene a esta tecnología.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un sistema embebido capaz de cumplir las funciones básicas de una estación central en la arquitectura de un sistema Scada.

OBJETIVOS ESPECÍFICOS

- Desarrollar un sistema de visualización que permita un mejor seguimiento de los eventos relevantes que se presenten en el sistema.
- Desarrollar los protocolos de comunicación necesarios para implementar un sistema de control y adquisición de datos.
- Implementar en una FPGA un sistema de procesamiento con algunas de las funciones básicas de un controlador, para así aplicar sus funciones básicas y ser usado con los recursos del laboratorio de ingeniería.
- Desarrollar una interfaz electrónica que permita adecuar señales de potencia con la FPGA.

1. SISTEMA SCADA

El SCADA es un sistema de Control supervisor y adquisición de datos que se implementa en un Hardware, que en general son computadores; trabaja en conjunto con controladores lógicos programables, los más utilizados en la industria son los PLC's, sin embargo no se limitan sólo a estos dispositivos.

Los controladores lógicos son los encargados de efectuar la acción de control sobre los actuadores del proceso industrial con base a unos parámetros definidos, es decir que la acción de control que se realiza depende de variables del mismo proceso industrial, el controlador tiene la capacidad de captar señales provenientes de diferentes sensores y si se cumple determinada secuencia lógica previamente programada, el controlador realiza un acción sobre los actuadores.

Los controladores lógicos por si solos son suficientes para realizar la acción de control necesaria para el manejo del proceso industrial, sin embargo presentan varias limitantes, como lo son la arquitectura distribuida, comunicación restringida a la empresa o recinto, control de flujo de operaciones en modo de lazo cerrado estrictamente por el mismo controlador, entre otras limitantes. El sistema SCADA permite mitigar estas limitantes debido a que presenta las siguientes características.

- Sistema centralizado: permite la conexión de múltiples controladores lógicos a una estación central, permitiendo monitorear todos los controladores desde un mismo punto.
- Presenta una mejoría en la comunicación: con los controladores lógicos, al tener un sistema centralizado, se puede hacer uso de la Internet para tener una

red de controladores distribuidos geográficamente, con la posibilidad de acceder a estos desde cualquier parte del mundo.

- Lazos de control cerrados por el operador: el operador tiene la posibilidad de intervenir en las acciones de los controladores, en caso de alguna eventualidad no esperada u programada en el controlador de forma previa.
- Adquisición de datos: Se puede llevar un historial sobre los eventos más relevantes que se presenten en proceso, con el fin de ser almacenados y analizados en un tiempo posterior.
- Generación de reportes: Al poseer la capacidad de almacenamiento de eventos relevantes, se pueden programar señales que indiquen cuando se debe hacer un mantenimiento ya sea preventivo o correctivo (según la situación lo amerite) de algún actuador.
- Conexión con otro software: con el poder de convergencia que presenta el sistema SCADA se puede implementar una aplicación que permita la conexión con algún software especializado en caso tal de presentarse algún evento en uno de los controladores lógicos si así la situación lo amerita.
- Entorno gráfico: los datos adquiridos por el sistema SCADA se pueden presentar de forma gráfica, presentando así de una forma más amable para el operador los acontecimientos del proceso.

A continuación se explica la configuración básica de un sistema SCADA visto generalmente en la industria Colombiana.

La arquitectura consta de un servidor el cual contiene un software que posibilita la comunicación entre el PLC y el Software para control y monitoreo de sistemas SCADA el cual es propio de cada marca constructora de PLC; el software que posibilita el puente de comunicación mencionado se denomina OPC, también se cuenta con terminales denominadas clientes, las cuales representan la interfaz proceso - operario, desde ésta terminal el operario puede visualizar todas las variables programadas, observar el proceso, monitorear alarmas, sacar reportes, tomar decisiones para realizar mantenimiento preventivo o correctivo, entre otras funciones.

Se cuenta con un dispositivo que permite la interconexión de elementos para formar la red, generalmente es un enrutador, dependiendo de los módulos de comunicación que posean los PLC's conectados a éste.

Los controladores PLC son los que contienen y modifican las variables que incurren en el proceso, estas variables son las monitoreadas desde el software SCADA, el tipo de comunicación que se haga con el servidor depende de los módulos que estén asociados al PLC, estos módulos varían de acuerdo a las necesidades y topología que se tenga en el proceso y van desde módulos con conexiones seriales con los estándares más utilizados (RS232C, RS485) hasta conexiones Ethernet inalámbricas.

Por últimos los dispositivos a conectar son todos aquellos elementos que entran en contacto directo con la producción como lo son motores, sensores, electroválvulas, alarmas, bocinas, indicadores, entre otros; estos elementos necesitan conectarse a través de otros dispositivos con el fin de otorgar mayor seguridad al dispositivo de control PLC, para esto se recurre a elementos como Relés de estado sólido, contactores y elementos adicionales de protección contra corriente.

Figura 1. Esquema general de un sistema SCADA comúnmente visto.

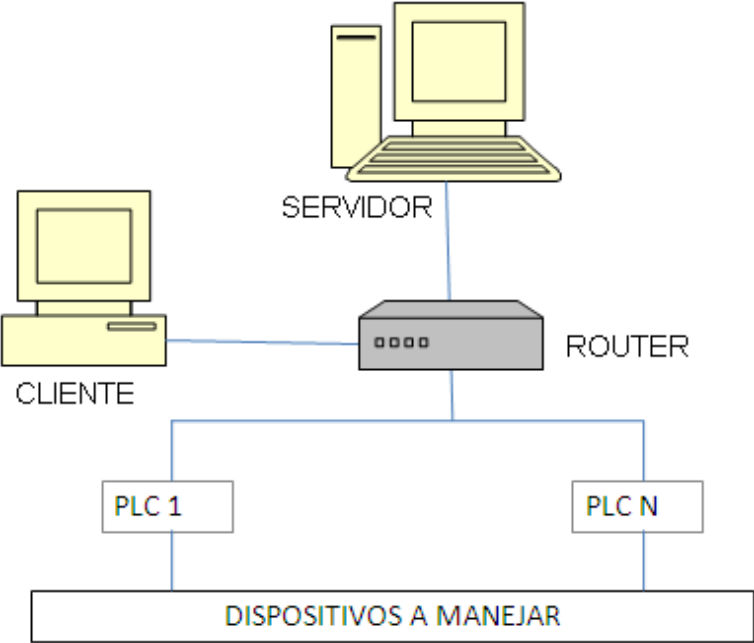
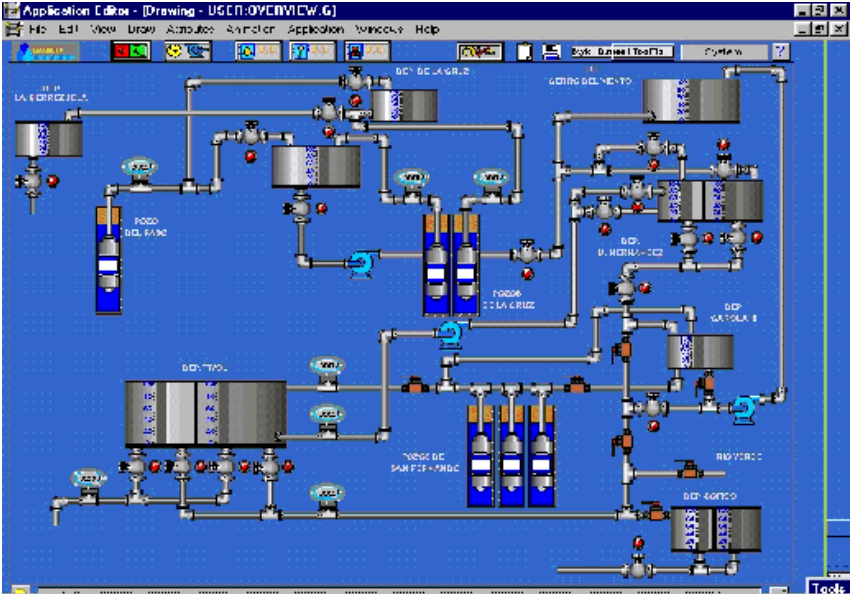


Figura 2. Pantalla SCADA de EMABESA.



Fuente: EMABESA [en línea] <Disponible en:
<http://dymael.galeon.com/automatizaciones/scadas/image002.gif>> [consulta: 25 Jul. 2009]

Figura 3. SCADA para cámaras frigoríficas.

Registros de Datos

REGISTRO DE DATOS DE LA INSTALACION

Sala	Nombre de la Sala	Humed	Tempe	Alam.	Sala	Nombre de la Sala	Humed	Tempe	Alam.
Sala n.1	Salazon	0	0	●	Sala n.13	S. Embutidos 1	0	0	●
Sala n.2	C. Polivalente	0	0	●	Sala n.14	S. Jamones 2	0	0	●
Sala n.3	C. Recepción	0	0	●	Sala n.15	S. Jamones 3	0	0	●
Sala n.4	C. Masas	0	0	●	Sala n.16	S. Embutidos 2	0	0	●
Sala n.5	C. Tripas	0	0	●	Sala n.17	Twin 1	0	0	●
Sala n.6	Produc. Terminad	0	0	●	Sala n.18	Twin 2	0	0	●
Sala n.7	Prepar. Jamones	0	0	●	Sala n.19	C. Salazon 2	0	0	●
Sala n.8	Recep. Jamones	0	0	●	Sala n.20	C. Oreo 2	0	0	●
Sala n.9	Obrador	0	0	●	Sala n.21	Tunel de -20	0	0	●
Sala n.10	C. Oreo	0	0	●	Sala n.22	Tunel de -40	0	0	●
Sala n.11	Condic. Exterio	0	0	●	Sala n.23	C. Recepción 2	0	0	●
Sala n.12	S. de Jamones 1	0	0	●	Sala n.24	P. Terminados 2	0	0	●

Salida Ver Secaderos Quitar Alarma Sonora

Fuente: MECOFI [en línea] <Disponible en: <http://usuarios.lycos.es/mecofi/hpbimg/registros.gif>> [consulta: 25 Jul.2009]

Las gráficas anteriores son ejemplos de Software SCADA comúnmente vistos en las empresas que poseen dicha tecnología, como se puede observar el Software SCADA no tiene una forma definida, depende estrictamente de las necesidades a visualizar y del criterio del programador para elegir la mejor forma de representar los datos solicitados por la empresa, entendibles por los operadores.

2. FPGA

Para el presente trabajo de grado se ha de utilizar la FPGA cuya referencia es la Spartan 3E como estación central del sistema SCADA

2.1. DESCRIPCIÓN

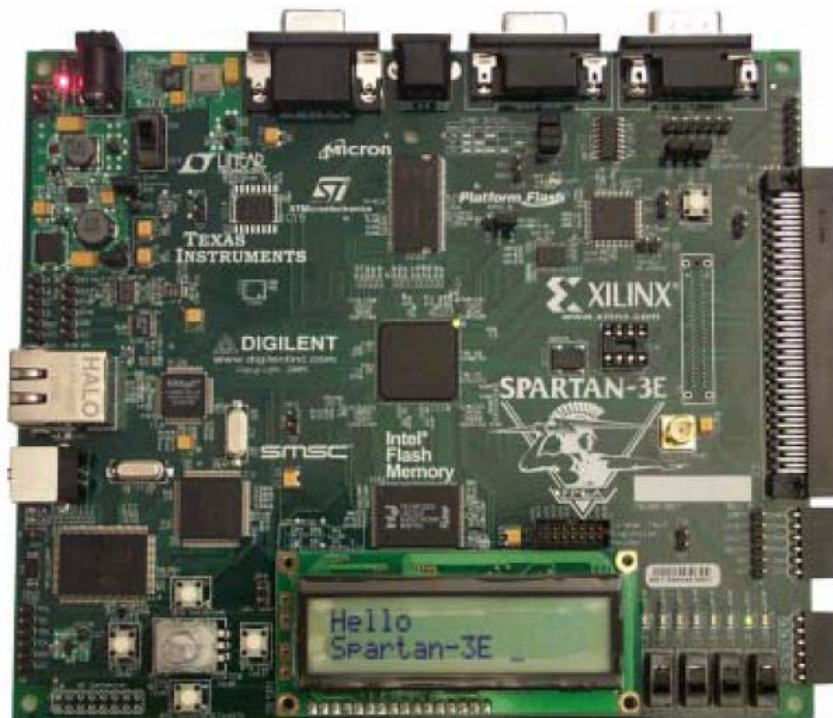
Una FPGA es un dispositivo semiconductor basado en tecnología CMOS, este dispositivo cuenta con bloques lógicos los cuales son programables y se pueden interconectar entre sí a gusto del programador con el fin de realizar una serie de funciones para llevar a cabo una tarea.

Las ventajas más sobresalientes de las FPGA's con las siguientes:

- Dispositivos reprogramables: Cuentan con la posibilidad de ser programadas tantas veces como sea necesario, lo cual convierte a este dispositivo en una herramienta para el desarrollo de hardware, capaz de estar sometido a constante mejoramiento propuesto por el diseñador.
- Bajo costo: su costo a pocas cantidades es muy bajo en comparación con otros dispositivos de lógica programable como los PLC, ya que estos requieren por lo general módulos adicionales como el de comunicaciones, análogo/digital, digital/análogo, entre otros según los requisitos del proceso, lo cual involucra una mayor inversión, lo que no sucede con las FPGA's debido a que estas tienen embebido una series de elementos que los PLC's no traen por si solos.

- Velocidad de ejecución: el diseño que se hace en la FPGA es un diseño de Hardware, por ende, las tareas programadas se ejecutan de forma paralela, lo cual aumenta la velocidad de ejecución en comparación con otros controladores lógicos que actúan de forma serial.
- Complejidad de los diseños: el dispositivo puede albergar diseños tan complejos como procesadores, y dependiendo del procesador implementado, se pueden implementar varios procesadores a la vez y aumentar la velocidad de procesamiento al paralelizar los diferentes procesos.

Figura 4. Tarjeta de desarrollo FPGA Spartan 3E.

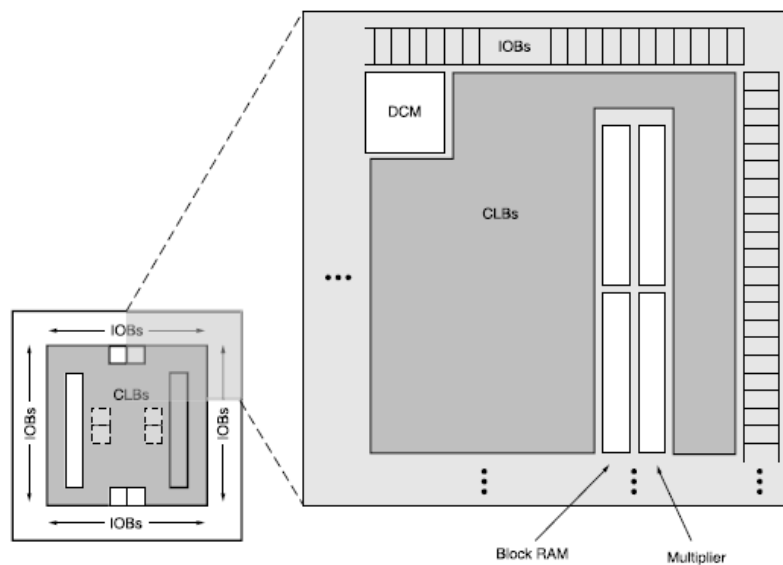


Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf > [consulta: 25 Jun.2009] p.11.

2.2. PRINCIPIO DE FUNCIONAMIENTO

Los elementos pertenecientes a la FPGA se interconectan para conducir las señales por los bloques necesarios con el fin de desarrollar la tarea programada, estas conexiones se hacen automáticamente, dependiendo de los recursos utilizados.

Figura 5. Arquitectura general de una FPGA Spartan 3E.



Fuente: Xilinx® [en línea] <Disponible en:

http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.4.

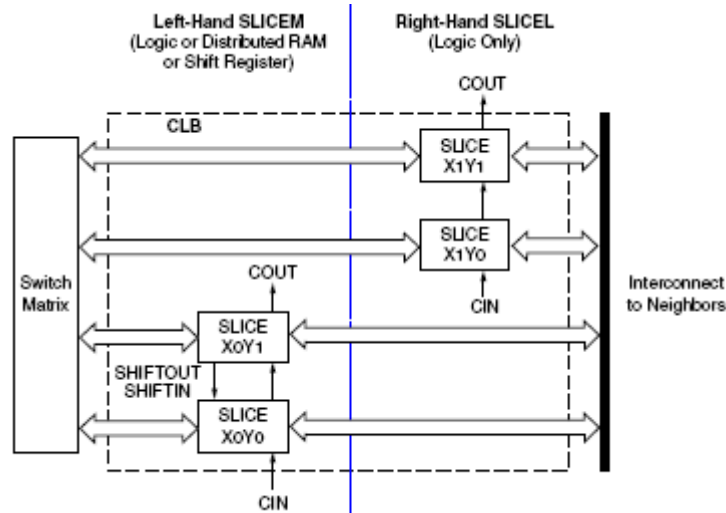
La arquitectura de la FPGA se puede dividir en 5 bloques a los cuales se les hace una breve descripción a continuación.

2.2.1. CLB

Estos bloques constituyen el principal recurso lógico con que la FPGA cuenta para implementar el Hardware programado por el diseñador, ya sean circuitos síncronos o combinacionales.

Cada CLB se compone de 4 Slices; a su vez, un Slice contiene 2 generadores de funciones lógicas (LUTs) los cuales están encargados de implementar las funciones lógicas; 2 elementos de almacenamiento los cuales pueden ser configurados como Flip Flop tipo D o como Latch sensible a nivel, estos son utilizados para sincronizar los datos procesados con las señales de reloj; 2 multiplexores los cuales pueden ser combinados con las LUTs para realizar operaciones lógicas de mayor complejidad.

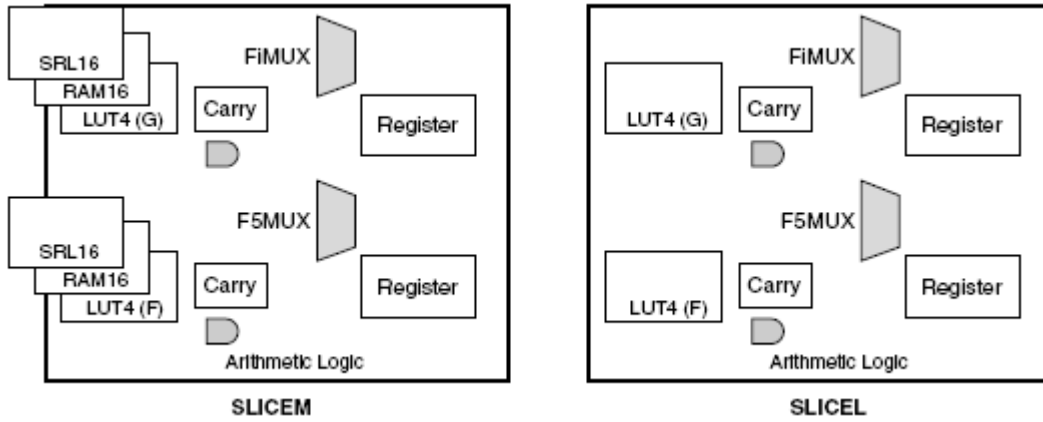
Figura 6. Esquema general de un CLB.



Fuente: Xilinx® [en línea] <Disponible en:

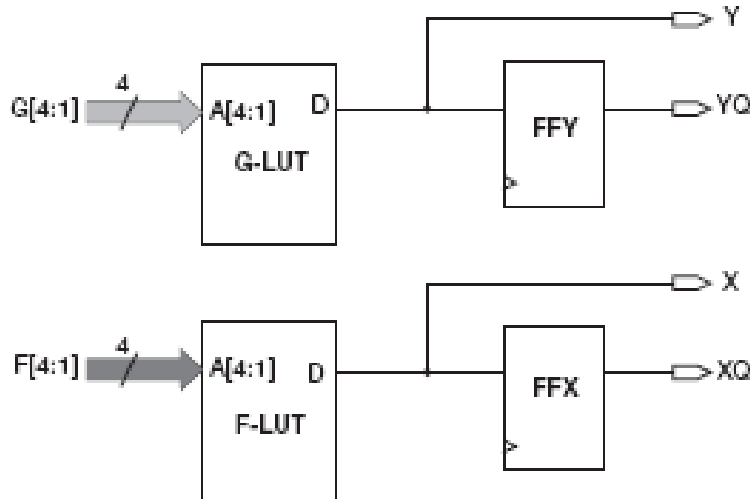
http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf [consulta: 28 Jun.2009] p.24.

Figura 7. Esquema general de los Slices.



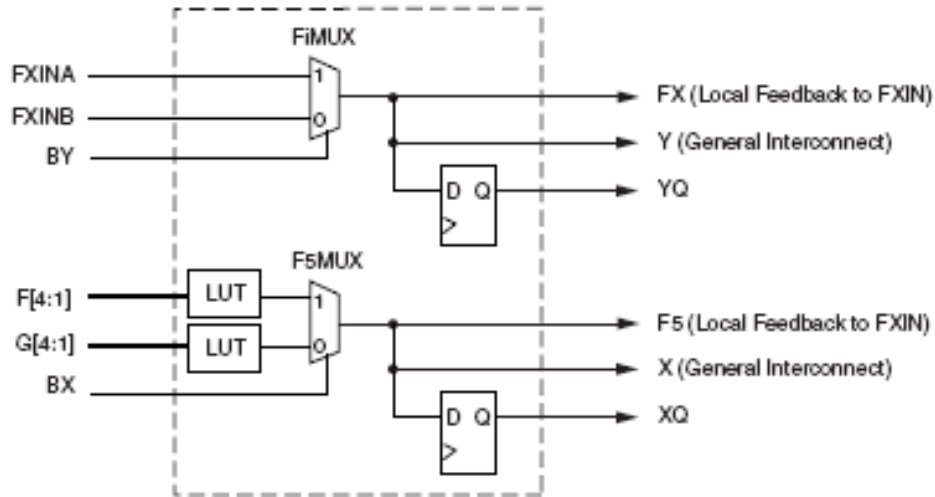
Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.24.

Figura 8. Esquema general de las LUTs.



Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.27.

Figura 9. Esquema general de los multiplexores.



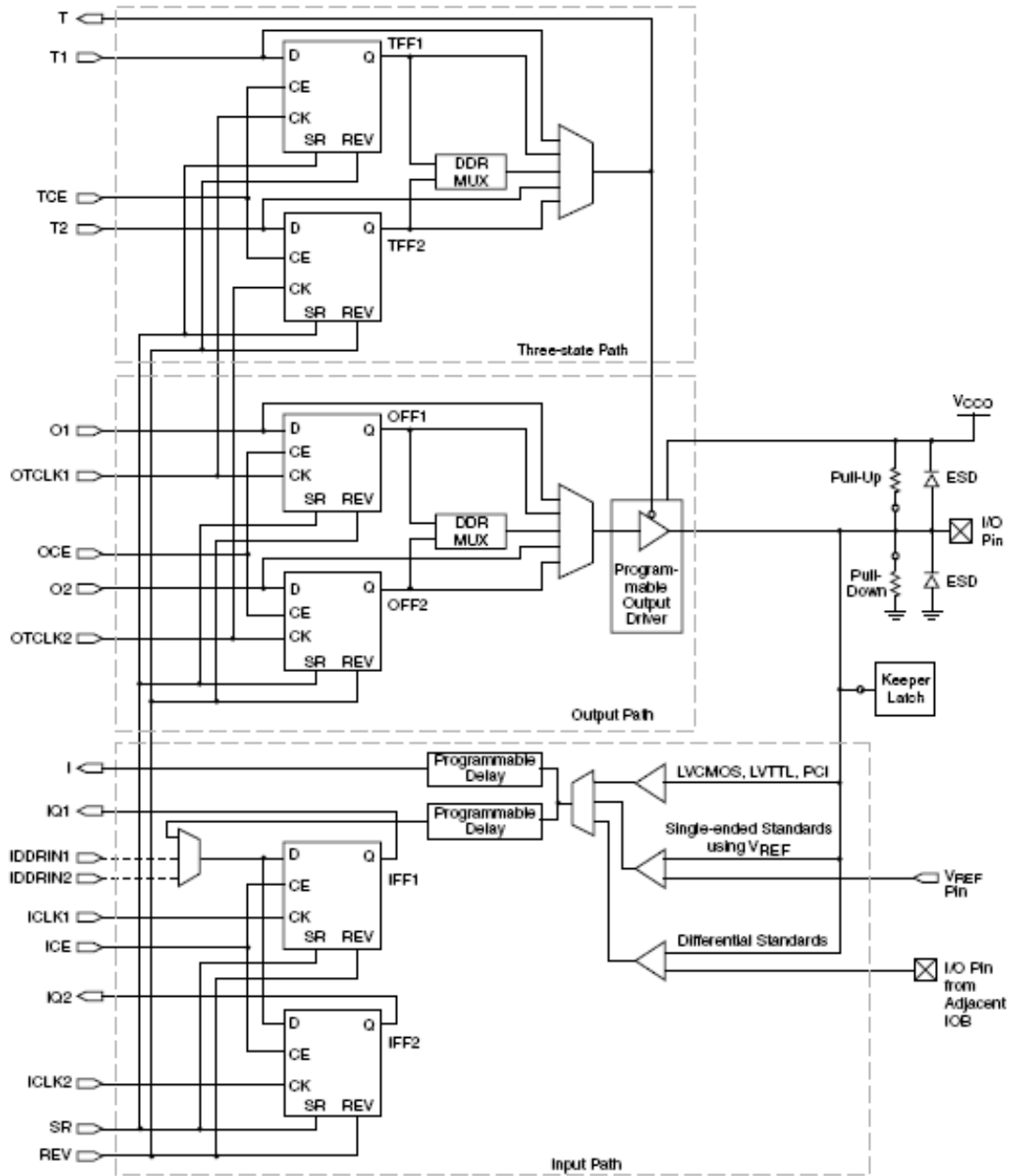
Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.27.

2.2.2. IOB

Son bloques que permiten la conexión entre un pin externo y la lógica interna de la FPGA; este bloque permite 3 tipos de configuración, las cuales son la ruta de entrada que captura una señal externa puesta a un pin determinado, para ser introducida a la lógica interna de la FPGA; la ruta de salida la cual lleva una señal interna hacia el exterior por medio de los pines de propósito general; y la ruta tri-estado el cual configura el pin de propósito general en alta impedancia.

El IOB también contiene flip-flops que pueden usarse como buffers de entrada y salida, resistencias que se pueden configurar como Pull-Up o Pull-Down con el fin de establecer niveles altos o bajos y un control de "Slew rate"

Figura 10. Esquema general de los bloques IOB.

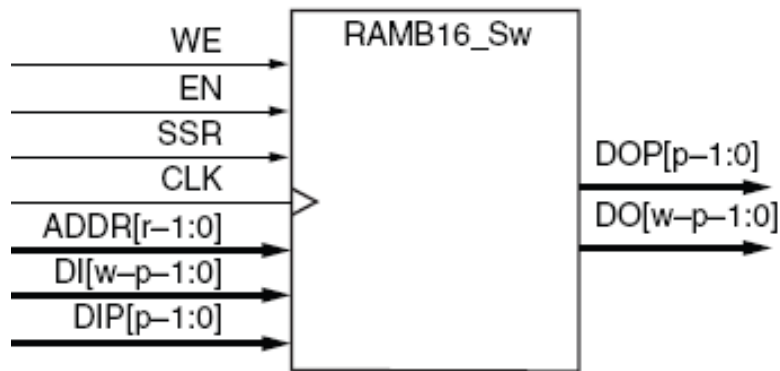


Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.11.

2.2.3. BLOCKRAM

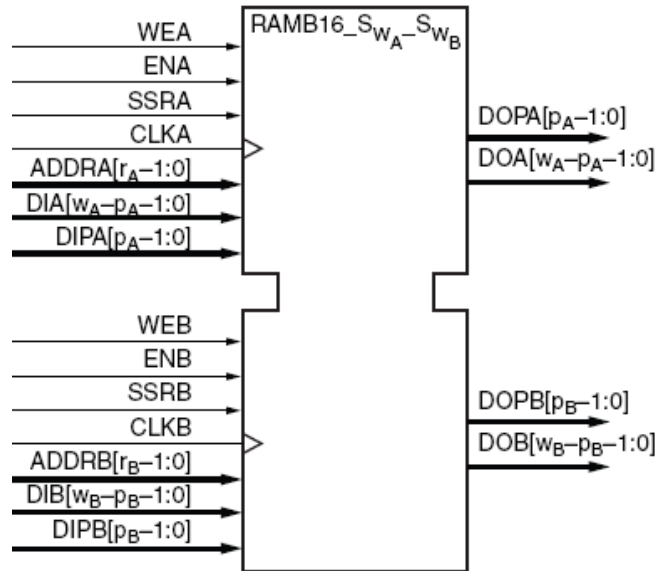
Son bloques dedicados de memoria que en total suman 20, cada uno con capacidad de 18 Kbits de memoria configurable, con el fin de implementar memorias con grandes anchos en su bus de datos o aumentar las direcciones de memoria según se requiera en el diseño pueden ser configurables como memorias de puerto simple o puerto doble, dependiendo de las necesidades del programador.

Figura 11. Esquema general de los bloques de memoria RAM de puerto simple.



Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.39.

Figura 12. Esquema general de los bloques de memoria RAM de puerto doble.



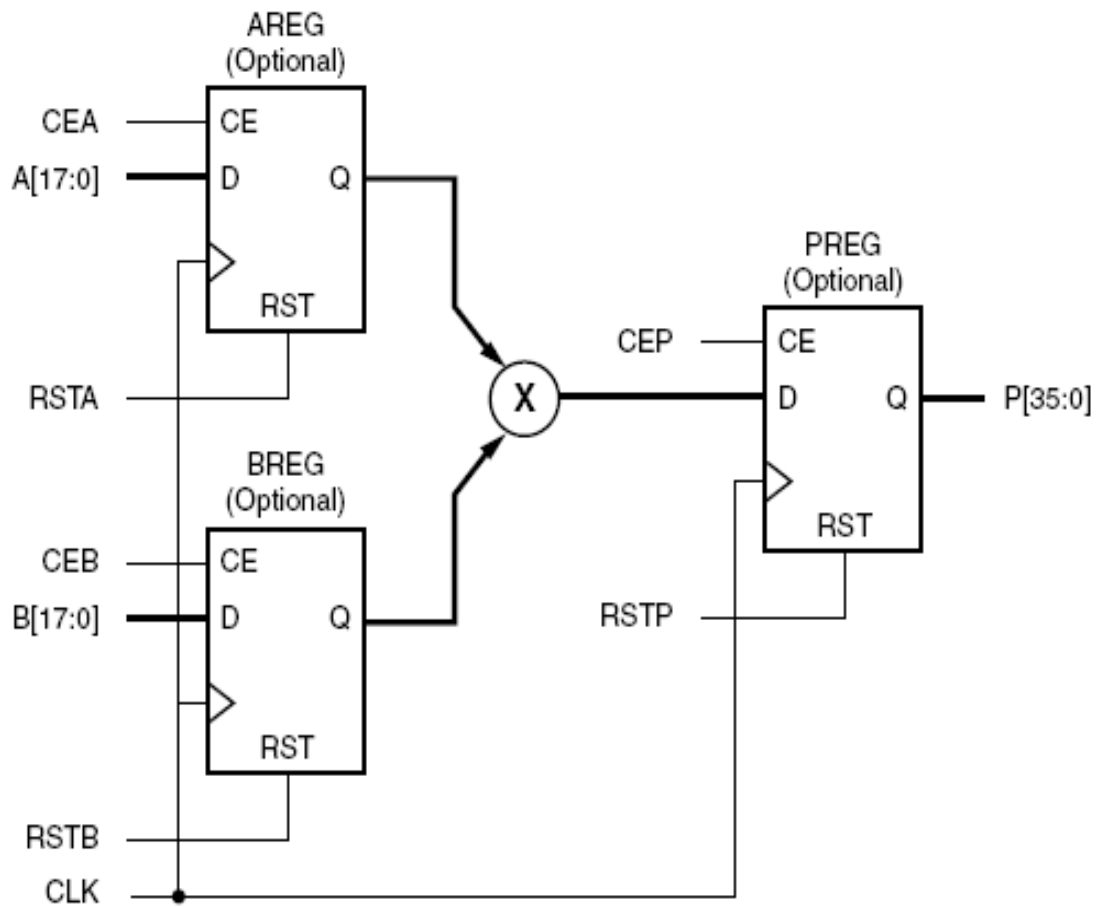
Fuente: Xilinx® [en línea] <Disponible en:

http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf > [consulta: 28 Jun.2009] p.39.

2.2.4. Multiplicadores dedicados

Son bloques embebidos que aceptan como argumento datos de hasta 18 bits para entregar resultados con una longitud de hasta 36 bits, permitiendo operaciones con datos signados. Cada bloque de memoria RAM tiene una conexión directa con el multiplicador, con el fin de manejar los datos de forma más eficiente.

Figura 13. Esquema general de los multiplicadores dedicados.



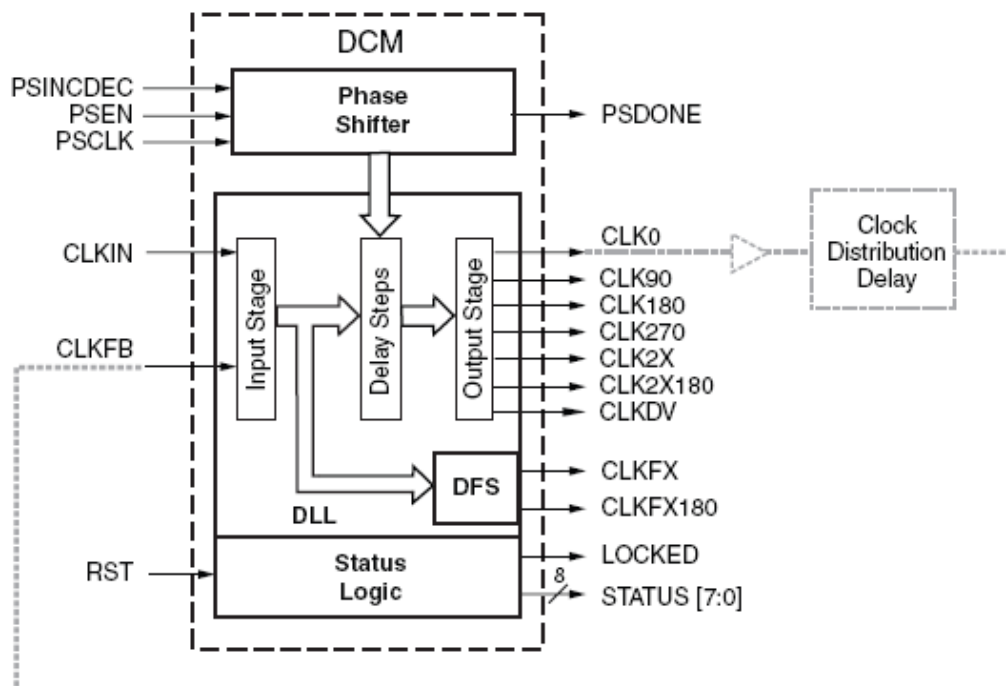
Fuente: Xilinx® [en línea] <Disponible en:
http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [consulta: 28 Jun.2009] p.45.

2.2.5. DCM

Son bloques dedicados al control de todos los parámetros relacionados con la frecuencia y la fase de los relojes internos de la FPGA, en total se tiene 4 bloques de estos.

Cada DCM cuenta con 4 componentes los cuales son el DLL el cual es capaz de eliminar los retrasos del reloj que se introduzca de manera externa, el DFS proporciona un valor de frecuencia a la salida estipulada por una entrada, el PS el cual controla los cambios de fase que pueden existir entre una señal entrada y la salida del DCM y el estado lógico (*Status Logic*) el cual indica el estado actual del DCM utilizando los pines de salida.

Figura 14. Esquema general de los bloques DCM



Fuente: Xilinx® [en línea] <Disponible en:

http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf [consulta: 28 Jun.2009] p.49.

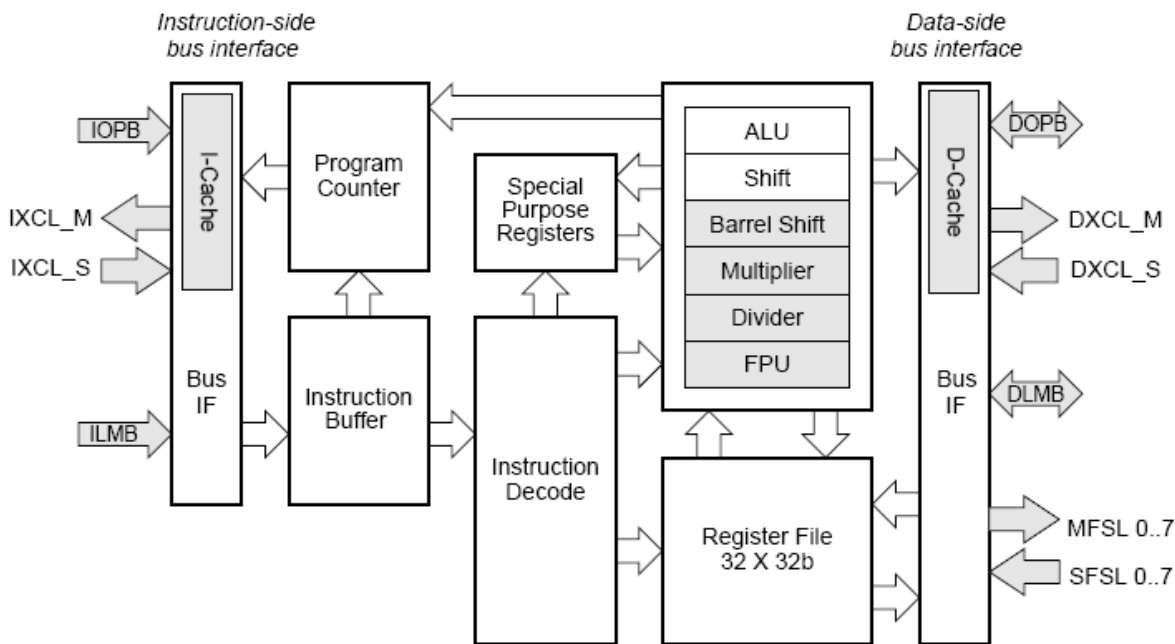
2.3. RECURSOS UTILIZADOS

Para realizar el presente proyecto se utilizan otros recursos a nivel de Software los cuales se ponen a disposición por la empresa xilinx, estos recursos de Software permiten controlar y administrar los recursos del Hardware mencionados de forma anterior, debido a que permite una interconexión entre los mismos y manipula los valores en sus registros dependiendo de una secuencia lógica impartida por el programador con el fin de llevar a cabo una tarea específica.

2.3.1. Microblaze

Es un procesador descrito por Software en lenguaje VHDL el cual se basa en la tecnología RISC de 32 bits, cuya arquitectura es de tipo Harvard y cuenta con 87 instrucciones diferentes; este procesador fue desarrollado por la empresa Xilinx para ser implementado en las familias de FPGAs Spartan y Virtex.

Figura 15. Diagrama de bloques del procesador Microblaze.



Fuente: Xilinx® [en línea] <Disponible en:

http://www.xilinx.com/ise/embedded/edk7_1docs/mb_ref_guide.pdf [consulta: 28 Jun.2009] p.11.

El procesador Microblaze tiene varias limitantes impuestas por las condiciones físicas de la FPGA, en otras palabras el espacio y la cantidad de recursos disponibles, adicionalmente las conexiones internas que debe hacer la FPGA para implementar el procesador, agregan retardos. Lo que limita la frecuencia de operación del mismo; estas son las razones por las cuales el procesador funciona bajo la arquitectura tipo RISC, donde el limitado número de instrucciones permite simplificar la unidad de descodificación¹; estas instrucciones fueron seleccionadas de tal forma que se pueda reducir el tamaño de la ALU, si el diseño requiere un poder de procesamiento mayor,

¹ AGUAYO E; GONZÁLEZ I; BOEMO E. Tutorial Xilinx MicroBlaze. España: Universidad Autónoma de Madrid, Escuela Politécnica Superior. 2005. p. 2.

de debe optar por un Hardware dedicado, diseñado con el resto de recursos que queden de la FPGA.

Debido a la limitación del procesador previamente mencionada, se implementó otras estrategias para aumentar el rendimiento del mismo, es por eso que se hace uso de la segmentación (Pipeline), con esta herramienta se posibilita realizar una instrucción por ciclo de reloj.

Microblaze dispone de la memoria distribuida que posee la FPGA, con el fin de implementar los 32 registros internos con los que fue concebido y gracias a que esta memoria cuenta con un tiempo de acceso corto, se puede habilitar las cachés, las cuales pueden ser configuradas en su mayoría por el diseñador (las cachés utilizan las BRAMs para su funcionamiento), dependiendo del tamaño de datos que se han de manejar, el uso de las cachés es necesario cuando los datos manejados se encuentren fuera de la FPGA, esto aumenta el rendimiento del procesador.

Como el procesador en cuestión obedece a la arquitectura Harvard, los datos e instrucciones son almacenados en memorias separadas, además se implementan diferentes buses para interconectar los elementos internos del mismo con el fin de aumentar su rendimiento. En Microblaze se definen 3 buses; el LMB el cual es un bus síncrono de alta velocidad utilizado para la conexión de los periféricos con los bloques de memoria interna, el bus puede ser usado tanto para datos como instrucciones, solo acepta palabras de 32 bits; el OPB que se utiliza para la conexión de periféricos (multiplexación distribuida²) cuyo tiempo de acceso es variable, acepta un tamaño de palabras variable; el DCR también utilizado para la conexión de periféricos, sin embargo la conexión es en forma de anillo y los periféricos deben tener un ancho de banda muy bajo, está diseñado para disminuir el uso de lógica interna³.

² AGUAYO E; GONZÁLEZ I; BOEMO E. Tutorial Xilinx MicroBlaze. España: Universidad Autónoma de Madrid, Escuela Politécnica Superior. 2005. p. 4.

³ AGUAYO E; GONZÁLEZ I; BOEMO E. Tutorial Xilinx MicroBlaze. España: Universidad Autónoma de Madrid, Escuela Politécnica Superior. 2005. p. 4.

Por último el procesador también maneja una serie de excepciones e interrupciones, la cual al ser activadas ejecutan una serie de instrucciones previamente especificadas al compilador, en el caso de utilizar varias interrupciones, se debe hacer uso de las prioridades de las mismas

2.3.2. Xilkernel

El Kernel se define como un Software el cual es responsable de realizar la conexión entre los diversos programas con el Hardware, gestiona los recursos del Hardware mediante llamadas al sistema⁴, restringe el acceso a los recursos del Hardware para los diversos programas.

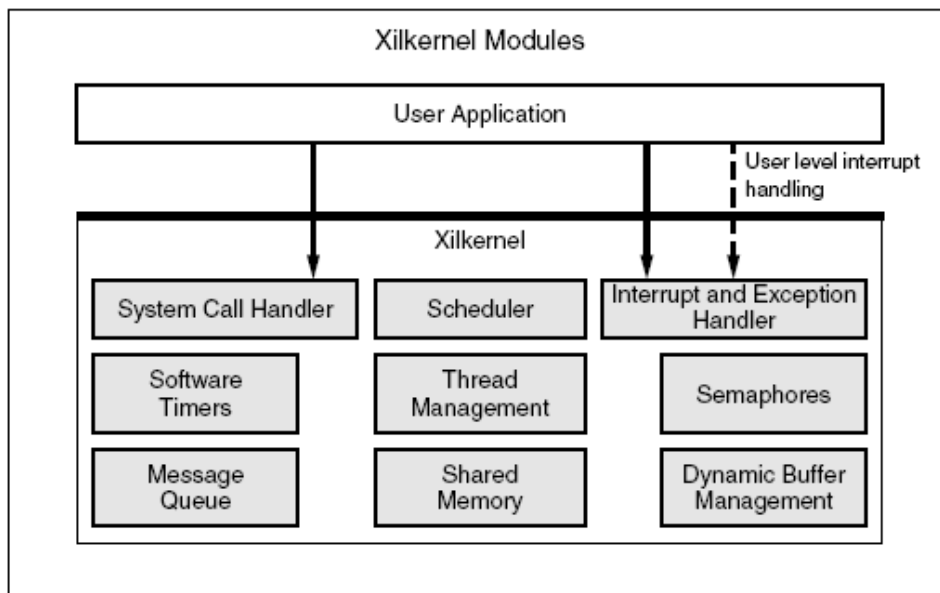
El Xilkernel es un Kernel desarrollado por **xilinx®** para ser implementado en procesadores como Microblaze, PowerPC® 405 y PowerPC 440 desarrollados por la misma empresa⁵.

El Xilkernel se caracteriza por ser pequeño y robusto, es adecuado para trabajar con servicios de alto nivel como redes, audio y video; tiene una gran capacidad de maleabilidad por parte del diseñador, permitiendo una mayor optimización en cuanto a tamaño y funcionalidad, escogiendo los módulos o recursos necesarios para el correcto funcionamiento de la aplicación; además facilita el uso de librerías aptas para el manejo de los diferentes periféricos con que cuenta la tarjeta de desarrollo Spartan.

⁴Wikimedia Foundation, Inc. Núcleo (informática) [en línea]. s.p.i. <Disponible en [http://es.wikipedia.org/wiki/N%C3%BAcleo_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica))> [Consulta: 29 Jun. 2009]

⁵Xilinx®. OS and Libraries Document Collection [en línea]. EDK 10.1 Service Pack 3. <Disponible en: http://www.xilinx.com/support/documentation/sw_manuels/edk10_oslib_rm.pdf> [Consulta: 28 Jun. 2009]. p. 62.

Figura 16. Módulos del Xilkernel.



Fuente: Xilinx® [en línea] <Disponible en: http://www.xilinx.com/support/documentation/sw_manuals/edk10_oslib_rm.pdf> [consulta: 28 Jun.2009] p.63.

3. VARIADOR DE FRECUENCIA

Para efectos prácticos, la funcionalidad del presente trabajo se demuestra con el uso de la estación de pruebas para un variador de frecuencia modelo F7 dispuesto por la empresa Variadores S.A.

3.1. DESCRIPCIÓN

El variador de frecuencia es un dispositivo electrónico el cual tiene como función regular la velocidad de rotación de un motor alimentado por corriente alterna, básicamente las maquinas de inducción, para éste propósito, el variador regula la frecuencia del voltaje incidente, logrando modificar la velocidad del motor, conservando la relación frecuencia y voltaje para no saturar el flujo magnético del motor y producir el menor ruido posible.

Figura 17. Variador Yaskawa modelo CMIR-F7U



Fuente: YASKAWA [en línea] <Disponible en: [http://www.yaskawa.com/site/dmdrive.nsf/\(DocID\)/AHUG-6H9QAE/\\$File/TM.F7.11.pdf](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf) > [consulta: 28 Ago.2009] p.1.

3.1.1 Motor de Inducción

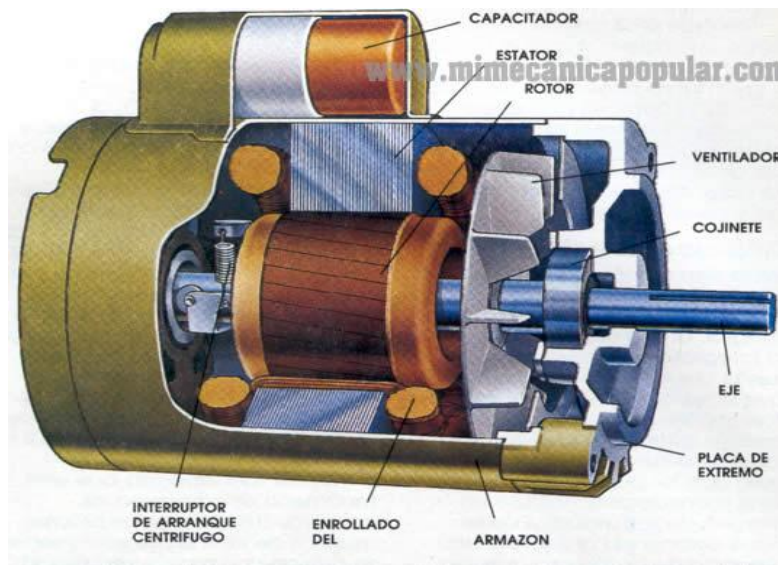
Estos motores son los más utilizados a nivel industrial por sus bondades de robustez, eficiencia y un menor costo, funcionan con corriente alterna.

El motor de inducción se compone de dos partes principales las cuales son el rotor y el estator. El estator se puede describir como un anillo magnético que posee un embobinado, el estator se encuentra fijo dentro de la carcasa protectora que lo envuelve.

Dentro del estator se encuentra el rotor, se describe como un cilindro con propiedades magnéticas debido a su embobinado, el rotor se fija al eje principal del motor.

El eje principal gira gracias al torque que produce cuando los campos magnéticos del rotor y el estator interactúan entre sí, estos campos magnéticos se producen debido a la alimentación de corriente alterna externa, en donde velocidad con que gira depende del número de polos con que cuente la máquina y la frecuencia de la alimentación, en cuanto la magnitud del voltaje, si se varia, la velocidad del motor varía poco, variar la magnitud del voltaje contribuye mas a estabilizar el campo magnético presente en el motor.

Figura 18. Motor de inducción



Fuente: Mi mecánica popular [en línea] <Disponible en: <http://www.mimecanicapopular.com/vergral.php?n=180>> [consulta: 29 Jun. 2009]

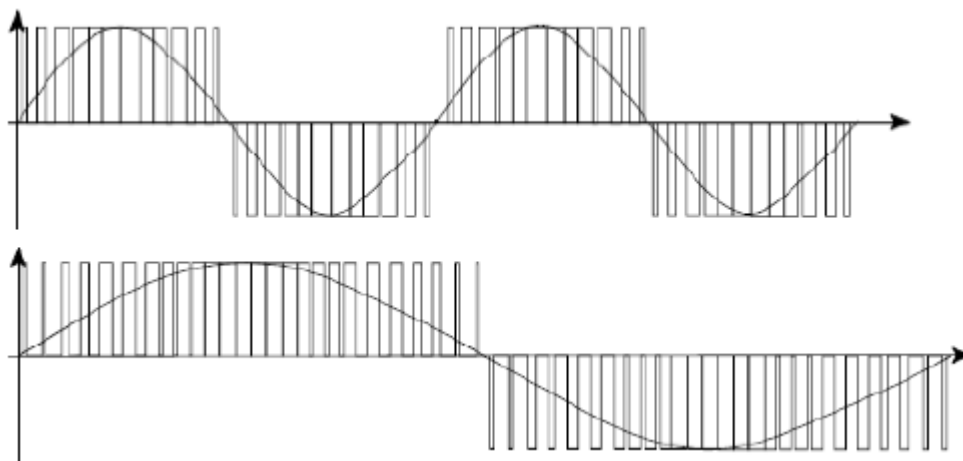
3.2. FUNCIONAMIENTO DEL VARIADOR DE FRECUENCIA

El variador de frecuencia es el más utilizado en la industria gracias a su mayor eficiencia en comparación con otros variadores existentes como los vectoriales o los

escalares, además su precio en el mercado se vuelve cada vez más favorables para el usuario. Los variadores de frecuencia se componen de dos etapas principales las cuales son:

- Etapa rectificadora: convierte la corriente alterna de la entrada del variador a corriente continua mediante un puente de diodos rectificadores, una vez se tenga la corriente continua, esta se suaviza y suaviza mediante condensadores, para luego ser tratada por la siguiente etapa.
- Etapa inversora: convierte la corriente de directa de nuevo en corriente alterna mediante el uso de diodos y transistores de potencia , obteniéndose una señal con magnitud y frecuencia diferentes; los transistores actúan de tal forma que se obtiene un tren de pulsos controlados de tensión, esta tensión varía con la frecuencia, mediante la modulación por ancho de pulso PWM.

Figura 19. Señales de frecuencia y amplitud con PWM



Fuente: YASKAWA. Manual de Instrucciones para los Variadores de Frecuencia de control Vectorial. Medellín: Variadores S.A. 2005. 174 p.

3.3. INTERFAZ ELECTRÓNICA

La tarjeta de desarrollo FPGA trae consigo un puerto de comunicación serial cuyo protocolo es el RS-232, este puerto es concebido como una herramienta de comunicación a corta distancia (máximo 10m de distancia entre dispositivos) en donde sólo interactúan dos dispositivos, en este caso la FPGA con un PC u otro dispositivo que permita la comunicación con protocolo RS-232, sin embargo los variadores de frecuencia cuentan con un puerto de comunicación configurable, ya sea a protocolo RS-422 o RS-485, por ende se debe desarrollar una interfaz electrónica que permita la interacción entre el protocolo que maneja la FPGA y el protocolo manejado por el variador, esta interfaz se refiere a un conversor RS-232 a RS-485, el plano esquemático se encuentra en el anexo B.

El protocolo seleccionado para comunicar al variador por el puerto serie es el RS-485, debido a que soporta mayores distancias, permite la conexión de hasta 31 esclavos (si todos los esclavos fuesen variadores) solo se requiere de dos hilos para la comunicación con dicho protocolo, manejando una comunicación “Half-duplex” lo cual es propicio para el manejo de los variadores por el puerto serial.

Se debe desarrollar una interfaz electrónica que permita la comunicación entre un dispositivo con protocolo RS-232 (FPGA) y otro dispositivo RS-485 (variador) la lo cual se propone el siguiente circuito, con el cual se trabajo alguna vez en la carrera.

El conversor identifica la señal de la comunicación serial con protocolo es el RS-232 cuya información maneja niveles de tensión entre -12v y 12v⁶, utilizando dichos niveles para representar los bits de información, luego de identificar dicha señal, la

⁶ Wikimedia Foundation , Inc. RS-232 [en línea]. s.p.i. <Disponible en <http://es.wikipedia.org/wiki/RS-232>> [Consulta: 17 Ago. 2009]

convierte a una señal de 5v⁷, además no representa los bits de información como niveles de tensión, sino por diferencial, es decir que dependiendo de la diferencia que exista entre las líneas “A” y “B” (pertenecientes a la comunicación RS-485) se sabe cuál es el dato que se transmitió. El conversor también tiene la posibilidad de realizar la operación inversa, el plano esquemático se encuentra en el ANEXO B.

3.4. PROTOCOLO MODBUS

Se define como un protocolo diseñado desde un principio para ser aplicado en el área de las comunicaciones a un nivel industrial, con el fin de interconectar dispositivos como PLCs, motores, actuadores, computadores, sensores y otros tantos dispositivos de entrada y/o salida que se puedan manipular en el medio industrial.

Este protocolo se sitúa en la capa 2 según el modelo OSI, su arquitectura se basa en la conexión maestro/esclavo, en donde se tiene un número variado de esclavos pero solo un maestro que facilita la coordinación entre los diferentes dispositivos conectados, para el caso concreto, un maestro puede controlar hasta 31 variadores esclavos.

Gracias al nivel de estandarización que posee este protocolo, Modbus se convierte en la herramienta más utilizada en el medio industrial, adicionalmente presenta ventajas como su naturaleza pública, es decir que cualquier persona tiene acceso a su modo de operación, es de fácil implementación en cuanto a nivel físico se refiere y la persona que lo implementa no necesita tener un vasto conocimiento del tema.

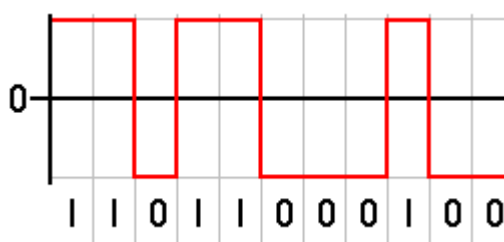
⁷Wikimedia Foundation, Inc. RS-485 [en línea]. s.p.i. <Disponible en <http://es.wikipedia.org/wiki/RS-485>> [Consulta: 17 Ago. 2009]

El tamaño de los bloques de datos manejados por este protocolo son variados dependiendo de la instrucción que se implemente, demostrando así la versatilidad del mismo, adicionalmente existen 2 vertientes por donde el protocolo Modbus tiene desarrollo, una es por Ethernet denominada Modbus/TCP y la otra se desarrolla por el puerto serial; el presente trabajo se enfoca en esta última vertiente debido a que el variador de frecuencia utiliza una conexión rs-485 para la implementación de este protocolo de comunicación.

En cuanto al protocolo Modbus con conexión serial se refiere, existen 2 tipos de comunicación las cuales son Modbus ASCII y Modbus RTU, se utilizará este último debido a que este es el tipo de comunicación que maneja el variador de frecuencia, sin embargo es ventajoso utilizar la comunicación RTU gracias a que presenta un mayor eficiencia en cuanto al manejo de los datos.

Adicionalmente la comunicación por RTU presenta características como el control de la transmisión de datos se hace mediante el conteo de caracteres, la transmisión es de carácter asíncrono, la codificación es de carácter NRZ de 8 bits de datos más los bits de arranque, paridad y parada, su topología es de tipo estrella, soporta las interfaces RS-232 Rs-422 y RS-485 y puede manejar velocidades desde 1200 y 19200 baudios y dependiendo la conexión física puede alcanzar una distancia de hasta 1200m sin la necesidad de utilizar repetidores⁸.

Figura 20. Diagrama de la codificación NRZ.



Fuente: Wikimedia Foundation, Inc. [en línea] <Disponible en: http://es.wikipedia.org/wiki/C%C3%B3digos_NRZ> [consulta: 9 Jul. 2009]

⁸ Protocolo Modbus.pdf. pág. 1

Como se mencionó anteriormente, el tamaño de la trama de la comunicación Modbus depende del tipo de información que se esté manipulando; para la comunicación con los variadores Yaskawa se identifican varios esquemas dependiendo de la función que se utilice.

Las funciones con las que se disponen son las siguientes.

Tabla 1. Comandos del protocolo Modbus para el variador Yaskawa

Función	Código
Leer múltiples registros	03H
Escribir en un solo registro	06H
Prueba de "Look-Back"	08H
Escribir múltiples registros	10H

Fuente: Xilinx® [en línea] <Disponible en:

[http://www.yaskawa.com/site/dmdrive.nsf/\(DocID\)/AHUG-6H9QAE/\\$File/TM.F7.11.pdf](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf) > [consulta: 23 Ago.2009] p.18.

El trabajo de grado se enfoca en controlar el variador y visualizar algunos de sus datos más relevantes.

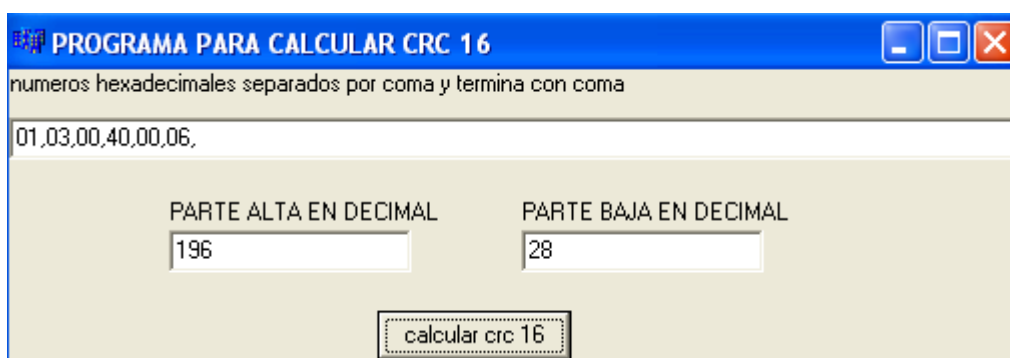
3.4.1. Trama para lectura de registros

Los registros que se van a leer son aquellos que contienen la información de la frecuencia de referencia, frecuencia de salida, corriente de salida, velocidad del motor y voltaje de AC de salida, estos datos se encuentran en los registros 0040H, 0041H,

0042H, 044H y 0045H respectivamente⁹; entre el registro 0040H y 0045H hay 6 registros, este dato se debe tener en cuenta a la hora de enviar la trama de lectura.

Por último se debe calcular el CRC 16 el cual es un método que el protocolo utiliza para verificar la validez de los datos recibidos, para saber cuál es el CRC-16 de la trama que se quiere enviar se desarrolló el Software “Calculo crc-16”

Figura 21. Programa para calcular el CRC-16.



El programa arroja como resultado los números 196 que equivale a C4H y 28 que equivale a 1C.

Ya con los datos obtenidos se puede configurar la trama necesaria para leer los registros pertinentes, la trama queda de la siguiente forma

⁹ YASKAWA. F7U Drive Parameter Access Technical Manual [en línea]. TM.F7.11. <Disponible en: [http://www.yaskawa.com/site/dmdrive.nsf/\(DocID\)/AHUG-6H9QAE/\\$File/TM.F7.11.pdf](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf)> [Consulta: 28 Ago. 2009]. p. 44.

Figura 22. Formato de trama para leer datos de registro del variador Yaskawa.

Dirección del esclavo	Función	Registro inicial a leer		Nº de registros a leer a partir del registro inicial		Cálculo CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
01H	03H	00H	40H	00H	06H	C4H	1CH

3.4.2. Trama de respuesta

Una vez enviada la trama de lectura de registros por parte del maestro (FPGA) hacia el esclavo (variador), dicho esclavo analiza la validez de la información recibida, de ser correcta la trama, el variador devuelve una trama con los datos solicitados, la trama devuelta es de la siguiente forma.

Figura 23. Formato de trama de respuesta del variador ante la solicitud de lectura de datos de los registros.

Dirección del esclavo	Función	Reg. 1		Reg. 2		Reg. 3		Reg. 4		Reg. 5		Reg. 6		CRC-16	
1 Byte	1 Byte	2Bytes		2Bytes		2Bytes		2Bytes		2Bytes		2Bytes		2Bytes	
01H	03H	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx

3.4.3. Trama de escritura de registros

Las posibilidades del protocolo Modbus no sólo se limita a leer registros del variador. También se posibilita el control del mismo, para lograr esto se debe escribir valores en los registros, dependiendo de lo que se quiera hacer, para ilustrar esta posibilidad, se le enviara desde el software diseñado en la tesis la frecuencia de referencia a la cual debe operar el variador, en otras palabras se le escribirá en el registro de frecuencia de referencia el valor al cual debe operar, este registro es el 0002H.

A modo de ejemplo si se quisiera que el variador operara a una frecuencia de 50 Hz, se le debe escribir en el registro 00002H el valor de 5000 (1388H) de la siguiente forma.

Figura 24. Formato de trama para escribir en los registros del variador Yaskawa.

Dirección del esclavo 1 Byte	Función 1 Byte	Registro a escribir 2 Bytes		Valor del registro 2 Bytes		Cálculo CRC16 2 Bytes	
01H	06H	00H	02H	13H	88H	25H	5CH

Si el comando fue recibido por el variador de forma exitosa, éste hace el cambio en su registro y devuelve la misma trama que el maestro envió para cambiar su registro.

4. CONEXIÓN FPGA A INTERNET

Una vez definido los periféricos a utilizar mediante el asistente BSB y haber creado el hardware, se procede con la configuración del Software, en otras palabras el sistema operativo que se encargará del manejo del Ethernet y el protocolo TCP/IP, el sistema operativo para Microblaze es Xilkernel (como ya se había hecho referencia), el cual para el correcto funcionamiento del presente proyecto, usa 2 librerías las cuales son lwip y xilmfs; esta conexión para ser entendida, se debe describir el modelo OSI sobre el cual se define la conexión de un dispositivo a Internet, para el caso concreto el dispositivo es la FPGA.

4.1. Modelo OSI

Se interpreta como un estándar de redes de comunicación a seguir el cual surge como respuesta ante la necesidad de impartir un orden ante el crecimiento desordenado en cuanto se refiere a las implementaciones de las redes que se generaban en las compañías de forma individual.

El uso de las redes tuvo una expansión exponencial, lo que comenzó a traer problemas a la hora de interconectar diferentes redes pues la tecnología utilizada era muy diversa y la comunicación era casi imposible. Es por eso que se crea el modelo OSI el cual se divide en 7 capas las cuales son:

Figura 25. Modelo OSI.



Fuente: Wikimedia Foundation, Inc. [en línea] <Disponible en: http://es.wikipedia.org/wiki/Modelo_OSI> [consulta: 20 Jul. 2009]

4.1.1. Capa física

Concierne todo lo relacionado con la parte física de las conexiones que hacen posible la comunicación entre el computador o terminal con la red, se relacionan con los cables como el cable trenzado UTP utilizado para conectar la FPGA con el Router que facilita la salida de la FPGA a Internet; conectores como el RJ45 el cual es donde va conectado el cable UTP.

La información viaja por estos medios mediante cambios de niveles en la tensión o en la corriente, dependiendo de las herramientas utilizadas y las necesidades presentes; pueden manejar topologías de punto a punto o punto a multipunto.

4.1.2. Capa de enlace

Se encargar de realizar el direccionamiento físico, es decir repartir la información correspondiente a cada elemento de la red, en caso tal de tenerse una topología multipunto; al distribuir la información, debe tener un control de flujo de la misma, lo cual abre el camino para otra función la cual es detectar errores en los paquetes de la información.

Cada dispositivo físico que necesite tener acceso a la red debe contar con una dirección MAC la cual es única para cada dispositivo.

4.1.3. Capa de red

Esta capa tiene como objetivo describir las herramientas que hacen posible que la información procedente de un origen llegue al destino, sin necesidad de existir una conexión directa. El dispositivo representante de esta función es el enrutador.

En el caso de presentarse alguna congestión en la red producida por la alta demanda en una conexión, El enrutador se encarga de realizar acciones para controlar dicha congestión y tratar que la red no colapse.

En esta etapa es donde se realiza el direccionamiento lógico (direcciones IP) y se determina la ruta que debe seguir la información para llegar a su destino.

4.1.4. Capa de transporte

La función básica es realizar un puente entre la información proveniente de las capas superiores y la capa de red para ser transferida dicha información dividiéndola en paquetes para su fácil transporte, debido a que divide los datos por paquetes, se debe encargar de que dichos paquetes lleguen de forma correcta a su destino.

La comunicación que se transporta entre los diversos dispositivos no depende de la naturaleza de la red física que se utilice en el momento, gracias a la función descrita en la capa de transporte, lo cual facilita la estandarización en la comunicación a nivel global.

4.1.5. Capa de sesión

En esta capa se establece, controla y finaliza todas las conexiones entre los usuarios finales que acceden a la comunicación, la cual puede ser un proceso, servicios o aplicaciones.

Se encarga de establecer y controlar la sesión realizada entre el origen y el destino, controlar el tiempo de acceso a una operación crítica en tal caso que 2 o más comunicaciones lo requieran, controlar puntos de verificación utilizados para reanudar una comunicación sin comenzar desde el principio en caso tal de que esta se haya interrumpido por algún motivo. Sin el uso de esta capa, no se podría garantizar la conexión entre la fuente y el destino para transmitir la información.

4.1.6. Capa de presentación

En esta capa se tiene como objetivo la representación de la información con el fin de poder entender dicha información recibida, sin importar las representaciones internas que manejen los diferentes dispositivos conectados, por ende encontramos códigos como ASCII, Unicode; sin embargo esto no sólo aplica para caracteres, también para sonidos e imágenes y otros datos en general.

Esta capa se comienza a relacionar directamente con el usuario final, debido a que ya no se tiene la necesidad de conocer como maneja la información cada máquina o la semántica que se utiliza internamente, lo cual ayuda a la estandarización de la comunicación.

4.1.7. Capa de aplicación

En esta capa se describe la posibilidad de acceso que poseen las aplicaciones (las cuales pueden ser del usuario) con los demás servicios descritos en otras capas, define los protocolos utilizados para el intercambio de datos con el fin de manejar diversas aplicaciones como correo electrónico, transferencia de datos y otras tantas que surjan con base al crecimiento tecnológico que se evidencia todos los días.

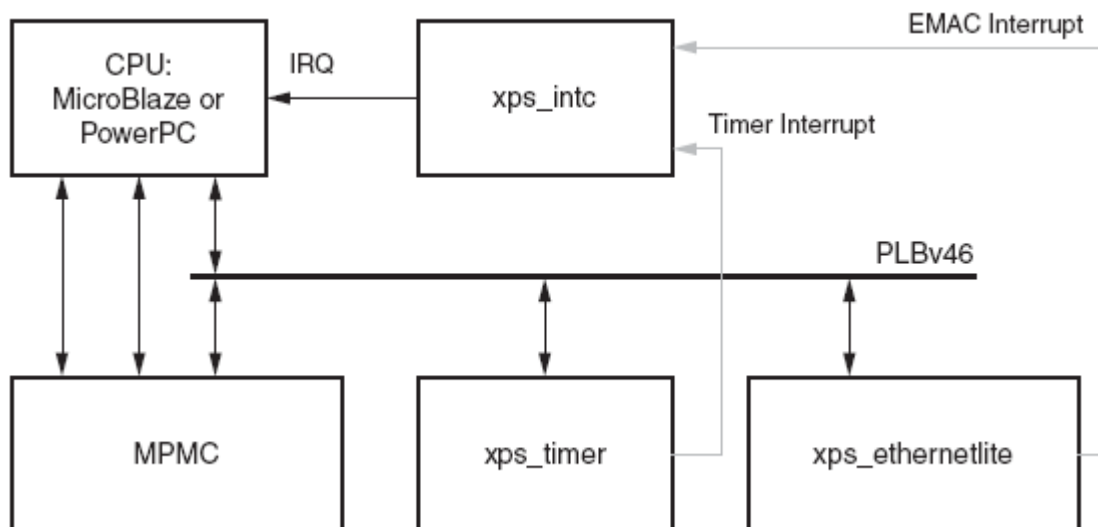
El usuario final generalmente crea una interfaz que le facilite el acceso a las aplicaciones, para ejemplificar lo dicho, se observa el acceso una página HTTP, el usuario ingresa la dirección a la cual desea acceder en la interfaz y está por debajo se encarga de realizar la petición.

4.2. Lwip

Es una librería que permite el manejo de protocolo TCP/IP mediante el uso de hilos, los cuales son muy útiles para realizar varias tareas “al mismo tiempo”, aunque los hilos comparten los mismos recursos, representa un aumento en la eficiencia y agilidad en el proceso.

Lwip controla el Ethernetlite el cual es la interfaz de conexión entre el procesador y la señal externa, el siguiente esquema ejemplifica una conexión del Ethernetlite.

Figura 26. Arquitectura del sistema utilizando Ethernetlite.



Fuente: Xilinx® [en línea] <Disponible en:

http://www.xilinx.com/support/documentation/sw_manuals/edk10_oslib_rm.pdf [consulta: 28 Jun.2009] p.165.

El timer y el modulo Ethernetlite se interconectan mediante el bus PLB, se hace necesario el uso del timer debido a que la librería Lwip requiere acceder a ciertas aplicaciones cada intervalo de tiempo para llevar a cabo su función; tanto el timer como el Ethernetlite poseen una señal de interrupción las cuales se hacen necesarias para avisar al procesador Microblaze sobre alguna operación que los mismos estén ejecutando, estas señales de interrupciones se conectan a otro módulo el cual su

función es controlar las interrupciones, darle un orden de prioridad a las mismas para que el procesador se le facilite la atención de las mismas; una vez el procesador atiende las interrupciones y procesa la información correspondiente, realiza las acciones pertinentes sobre el MPMC que representa la conexión que tiene el procesador con otros periféricos, se debe hacer la aclaración que el procesador también tiene acceso a los módulos del timer y Ethernetlite, con el fin de impartirles alguna instrucción en tal caso que así se requiera.

Al habilitar el uso de Lwip, se hace uso del procedimiento Socket API el cual es utilizado para realizar una comunicación por una red, si se quiere utilizar este procedimiento, se deben habilitar de igual forma las funcionalidades de los semáforos, mutex y yield .

Por último se debe configurar como se va a realizar el manejo de los hilos que se involucran en el proceso, es decir cuál es el hilo principal, que para el caso se llamará **main_thread**, de donde parte el proceso.

4.3. Procedimiento para la configuración de la estación central como servidor

Para el presente trabajo se deben tener en cuenta 2 archivos fundamentales cuya extensión es *.c, el primero contiene la inicialización de los recursos y el segundo contiene lo referente al servidor Echo, lo que en últimas instancias es la función para la cual se programará la FPGA y así pueda funcionar como estación central del sistema SCADA.

4.3.1. spi_usr_app.c

Contiene en su función principal la inicialización de las interrupciones del procesador, lo cual es fundamental para que Microblaze reconozca los eventos de interrupción; el procesador debe manejar dos tipos de interrupciones las cuales provienen del timer y del Ethernetlite, así que estas interrupciones se deben registrar en el control de interrupciones para que éste las administre y le suministre la información de las mismas al procesador; luego se habilita el control de las interrupciones y se define la dirección de salida de los datos.

Una vez configurado el control de interrupciones, se especifica el intervalo de tiempo sobre el cual el timer funcionará, es decir que el timer presentará una señal de interrupción en estos intervalos de tiempos programados; ya definido esto, se inicializa o se le hace un “reset” y le aclaran las interrupciones, para luego colocar en marcha los timers y habilitar las interrupciones de los mismos. Por último se inicializa el Xilkernel el cual se traduce finalmente como el sistema operativo.

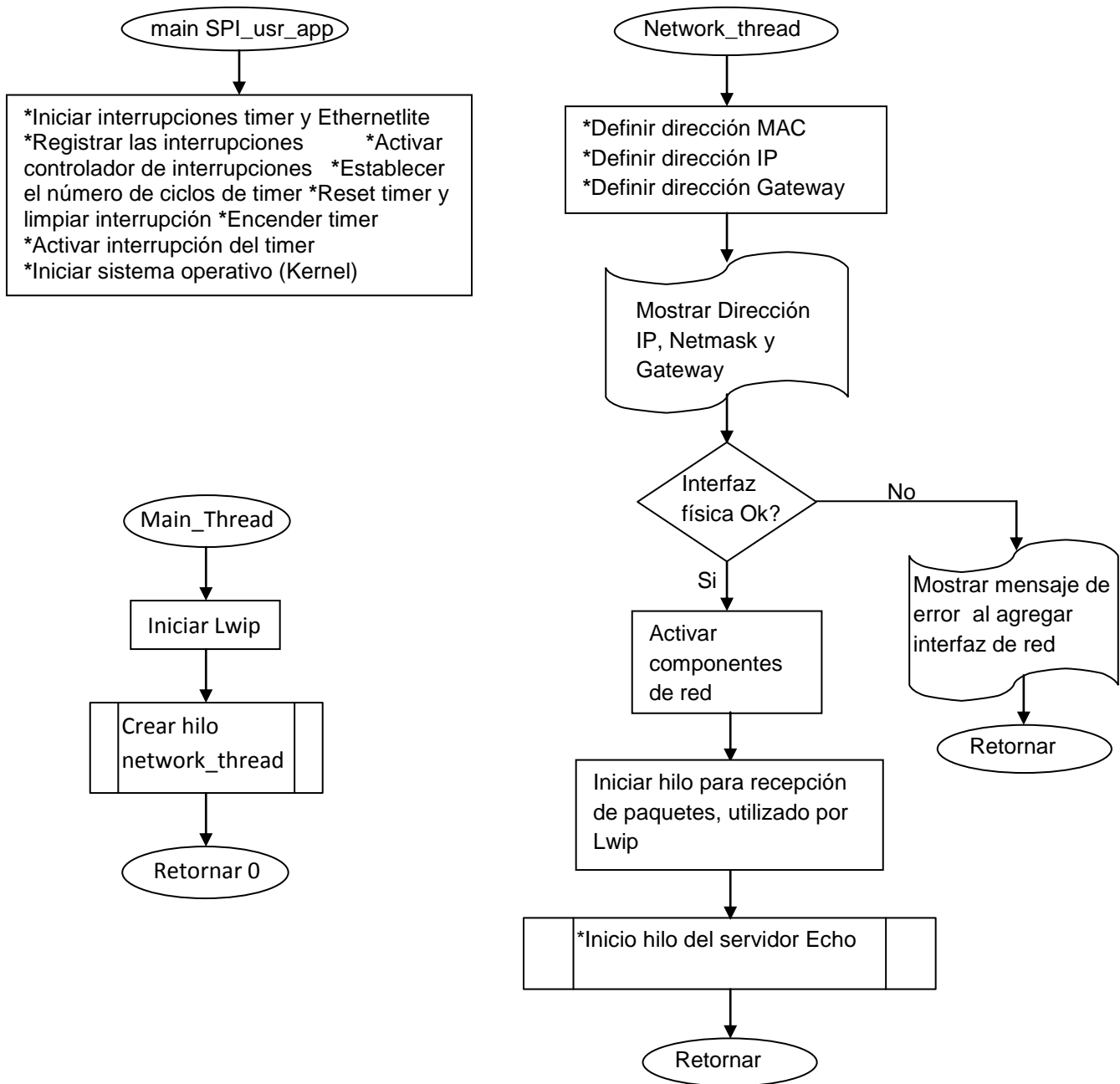
Al inicializar el sistema operativo, se crea un hilo estático denominado **main_thread**, el nombre del hilo se configuró en el asistente del sistema operativo como se mencionó de forma anterior y es de donde parte todo el proceso de conexión a Internet; el hilo estático inicializa la librería Lwip y lanza un hilo dinámico llamado **network_thread** el cual se encarga de manejar la capa física encargada de manejar el conector RJ-45 que es el conector del Ethernet y la capa de enlace encargada del direccionamiento físico y acceso a la red; esto según el modelo OSI.

El hilo **network_thread** configura la dirección MAC de la FPGA, la cual debe ser de 6 Bytes por norma, la dirección IP local, la dirección de la máscara y la dirección de Gateway o enlace, para el caso concreto se está trabajando con la versión IPV4, esto quiere decir que las direcciones IP que se manejan son de 32 bits, lo cual presenta un número limitado de direcciones disponibles en comparación con la versión IPV6, sin embargo es más que suficiente para demostrar el funcionamiento del proyecto.

Una vez definidos los parámetros de la red, se inicializa la conexión de dichos parámetros de red con la interfaz gráfica que para el caso se denomina **netif**, el cual se activa una vez se haya terminado de hacer las conexiones con los parámetros de la red.

Una vez realizada la conexión entre la capa física y la capa de enlace, se pone en marcha otro hilo el cual se encarga de iniciar la recepción de los paquetes de la información proveniente de Internet, éste hilo corre internamente y se usa para suplir las transacciones que requiera hacer el Lwip; también se encarga de correr el hilo encargado de la aplicación de servidor web llamado **web_application_thread**.

Figura 27. Diagrama de bloques general del programa spi_usr_app.c.



4.3.2. Echo.c

Una vez se ejecuta el hilo **echo_application_thread**, comienza a funcionar el archivo **Echo.c**, este hilo no requiere el protocolo HTTP; este hilo se encarga de configurar la comunicación con la red mediante una aplicación Socket API. Así que después de inicializar los dispositivos que la FPGA tenga asociados en el momento, se crea el socket para trabajar el protocolo TCP/IP. Para que las aplicaciones de sockets funcionen, se necesitan la dirección IP del servidor, el número del puerto en el cual el servidor escuchará las peticiones entrantes provenientes del cliente que para el caso será el puerto 7, la dirección IP del equipo remoto que se traduce en el cliente y el número del puerto por donde el cliente se comunicará con el servidor.

Una vez determinados los parámetros para el funcionamiento del socket, se coloca en funcionamiento la aplicación del servidor Echo; para lograr su funcionamiento el socket del servidor debe permanecer en un estado de escucha, por ende se prepara el socket para dicha función, se le indica cuando debe comenzar a escuchar y cuantos clientes puede atender a la vez, cabe anotar que entre más clientes sean escuchados al tiempo, se tendrá menos velocidad en las respuestas, debido a que los recursos que hacen posible la comunicación cliente-servidor son limitados.

Una vez el socket comience a escuchar los datos provenientes de los clientes, el socket debe aceptar dichos datos entrantes para ser analizados y entregar la respuesta adecuada.

A diferencia de un servidor Web con protocolo HTTP, el servidor Echo no requiere de un encabezado propio para funcionar como el perteneciente al HTTP, se encarga de recibir los comandos enviados por el usuario mediante el entorno gráfico de control diseñado en el presente proyecto, se procesan dichos comando y dependiendo de cuál sea, se realiza la acción correspondiente, una vez se ejecute la acción, el servidor Echo devuelve el mismo comando recibido para indicar que la ejecución del mismo fue exitosa.

El servidor Echo se encuentra en la estación central, debido a la simplicidad del Echo, no se requiere a un acceso de memoria como el caso de un servidor Web con protocolo HTTP, solo requiere ejecutar instrucciones cuando se le necesite.

En caso tal de que el número de clientes sobrepase el número de sockets que el servidor esté programado a escuchar, las demás peticiones quedarán en cola a la espera de que algún socket se desocupe para poder ser atendido.

El servidor Echo requiere del Software diseñado en el presente trabajo para su correcto funcionamiento, la diferencia radica en la visualización de gráficas en tiempo real y la posibilidad de crear historiales con el fin de analizar los datos de forma posterior, lo cual es una herramienta muy útil para realizar operaciones de mantenimiento ya sea preventivo o correctivo.

Figura 28.a. Diagrama de bloques general del programa Echo.c.

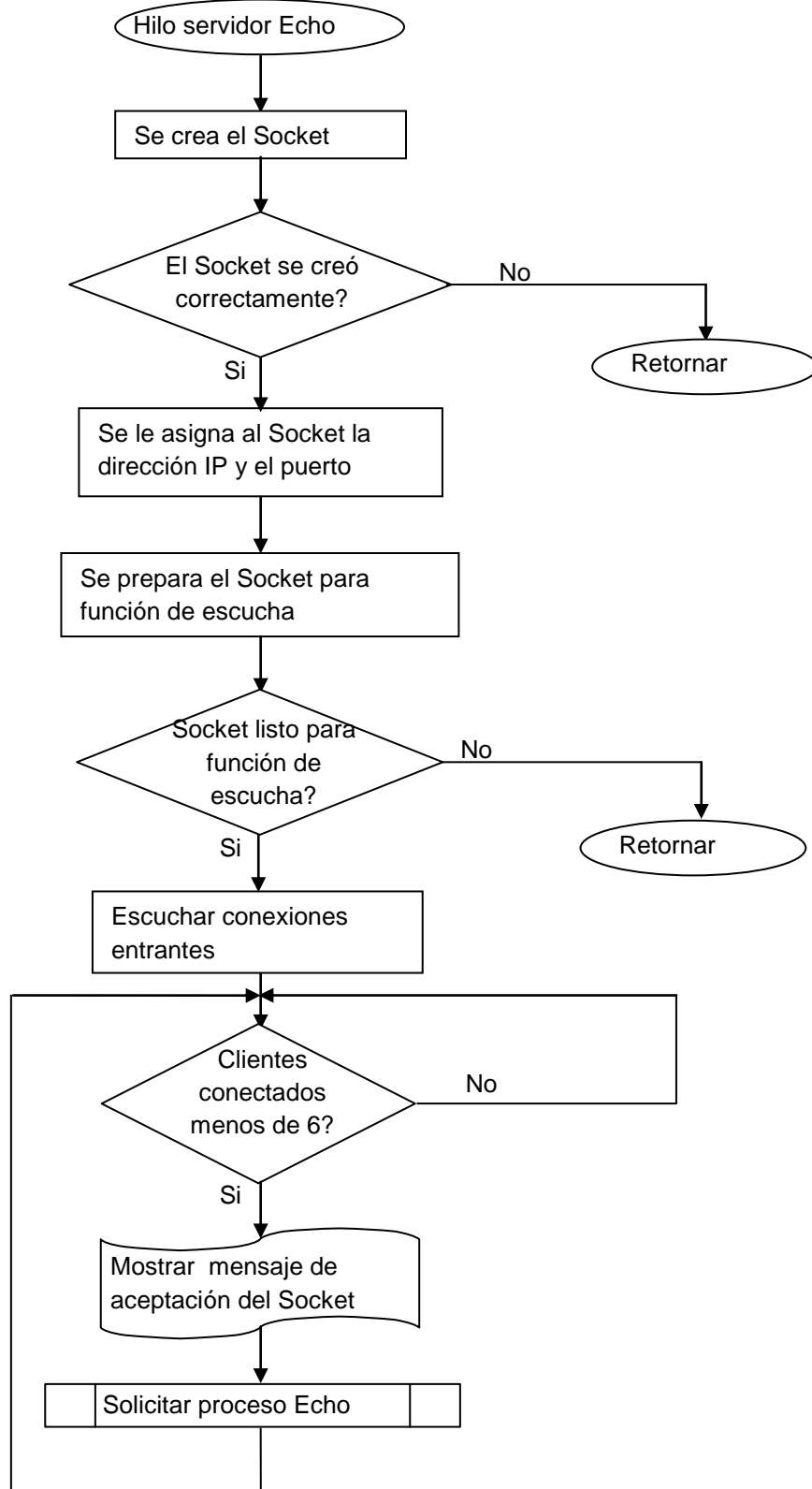
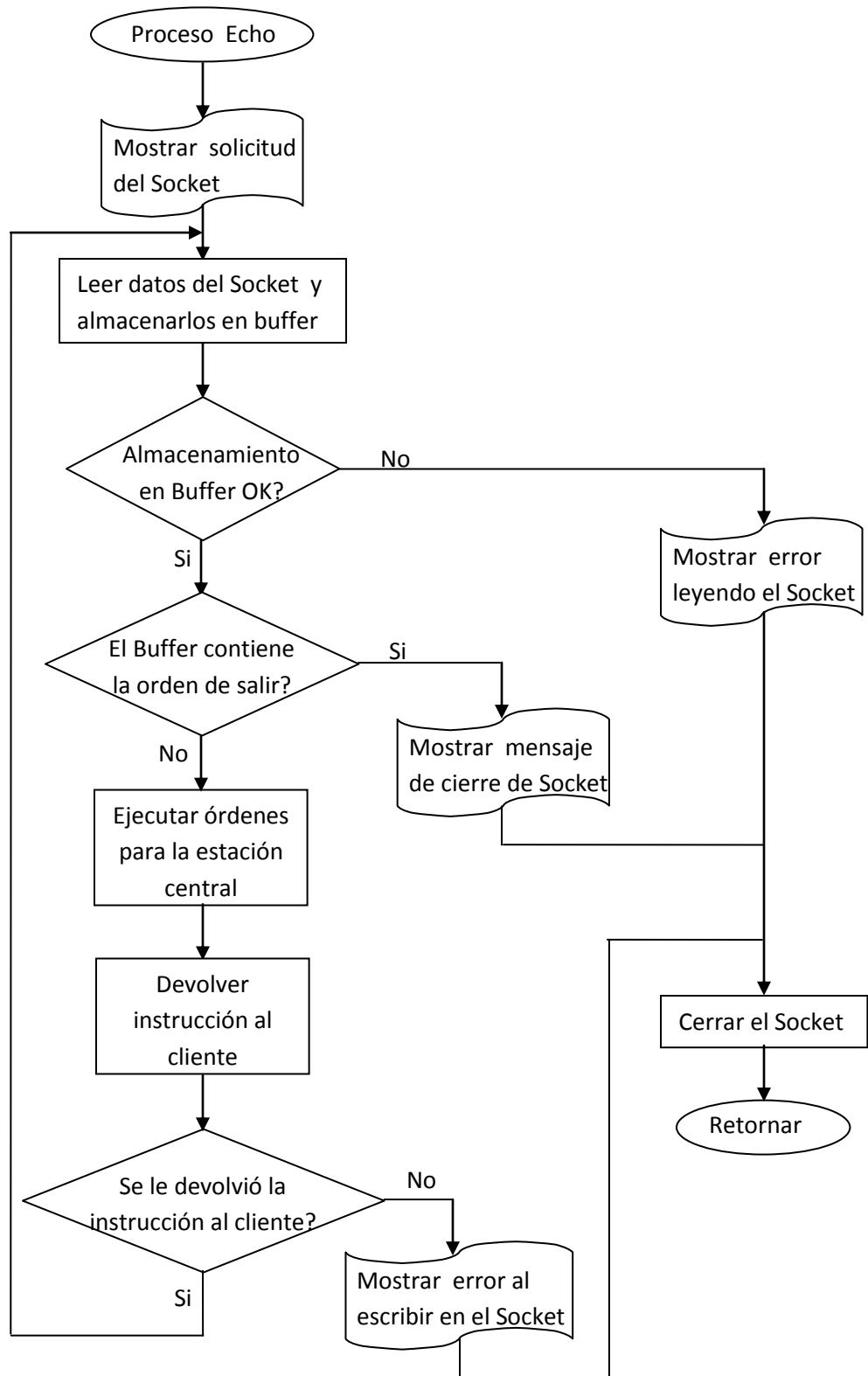


Figura 28.b. Diagrama de bloques general del programa Echo.c.

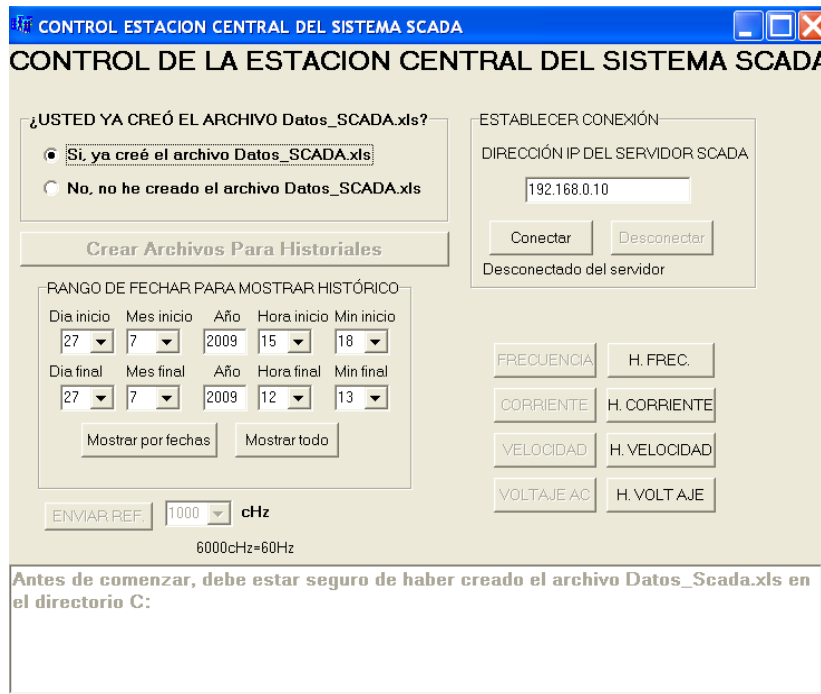


5. ENTORNO GRÁFICO

Como se explicó en los capítulos anteriores la terminal central (FPGA) contiene un servidor Echo, el cual requiere el uso del Software desarrollado en el presente trabajo para poder ser utilizado de forma apropiada.

El Software se crea bajo la aplicación **Borland C++Builder versión 6.0**, este Software se describe como un herramienta de desarrollo de aplicaciones el cual incluye una gamma de herramientas que facilitan la consecución de sus tareas, mediante el uso de los lenguajes de C y C++, apoyado en un entorno gráfico afable para el programador.¹⁰

Figura 29. Página principal del entorno gráfico para el control de la estación central.



¹⁰EMBARCADERO TECHNOLOGIES. C++ Builder. [en línea]. <Disponible en: <http://www.embarcadero.com/products/cbuilder/>> [Consulta: 21 Jul. 2009]

5.1. FUNCIONAMIENTO DEL SOFTWARE

La aplicación diseñada se basa en la programación orientada a objetos, éste método de programación facilita el desarrollo de las aplicaciones, debido a que el programador se encarga de invocar o introducir los objetos necesarios para su programa y colocarlos a interactuar entre sí para llevar a cabo la consecución de una tarea específica, una vez los objetos sean introducidos, estos por sí solos traen las librerías necesarias para su correcto funcionamiento, lo cual se traduce en una preocupación menos por parte del programador.

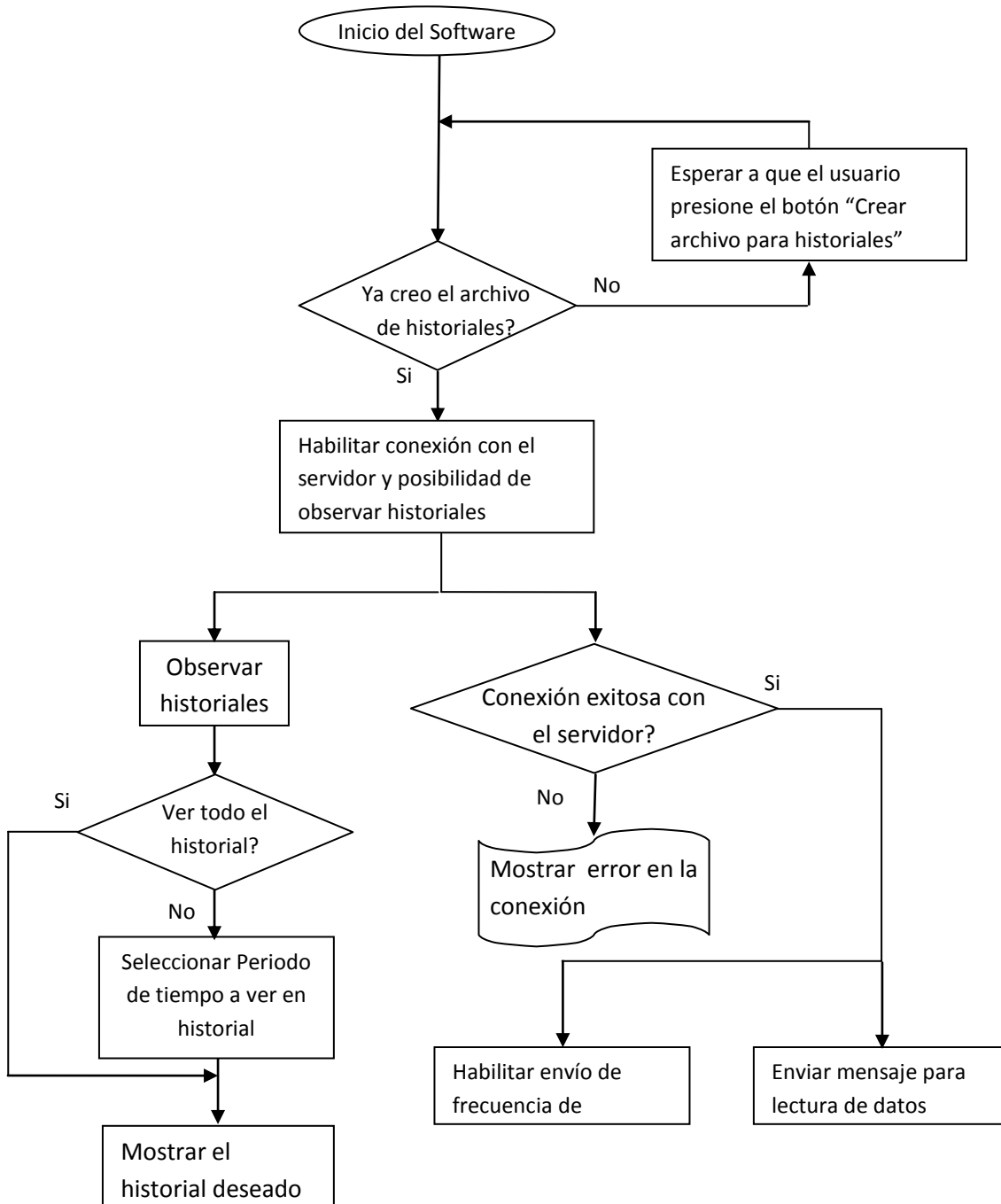
Los objetos más relevantes para lograr la adquisición de los datos de la estación central son 2. El primer objeto describe el funcionamiento de un Socket tipo cliente, el cual se encarga de establecer la conexión con el servidor Echo de la estación central y realizar las peticiones adecuadas para obtener los datos requeridos por el usuario de la interfaz, sin embargo la función del objeto Socket no se limita a la adquisición de datos, también posibilita la ejecución de comandos para que la estación central lleve a cabo y proceda con una función de control sobre el variador de frecuencia.

El segundo objeto se encarga de controlar un temporizador, cuya función es la de controlar los tiempos en que el Socket tipo cliente realiza una petición al servidor, el temporizador permite obtener los datos provenientes de la estación central de forma periódica, otorgando así un monitoreo constante, una de las principales características del sistema SCADA.

Adicionalmente la aplicación cuenta con una función que permite almacenar todos los datos que transitan por la comunicación realizada entre el cliente y el servidor con el fin de obtener historiales de eventos para ser analizados de forma posterior si así se requiere, estos datos se guardan en un archivo de Excel llamado "Datos_SCADA.xls" en el Directorio C:\, que la aplicación misma crea, se opta por guardar los datos en esta aplicación ya que es muy común en los computadores, además presenta los datos en un formato conocido, lo cual facilita su acceso y análisis.

A continuación se muestra el diagrama de flujo de forma general del Software para entender mejor su funcionamiento, no se puede olvidar que es una programación orientada a objetos, eso significa que el usuario escoge que acción se realiza mediante su interacción con el Software.

Figura 30. Diagrama de bloques general del programa de control de la estación central SCADA.

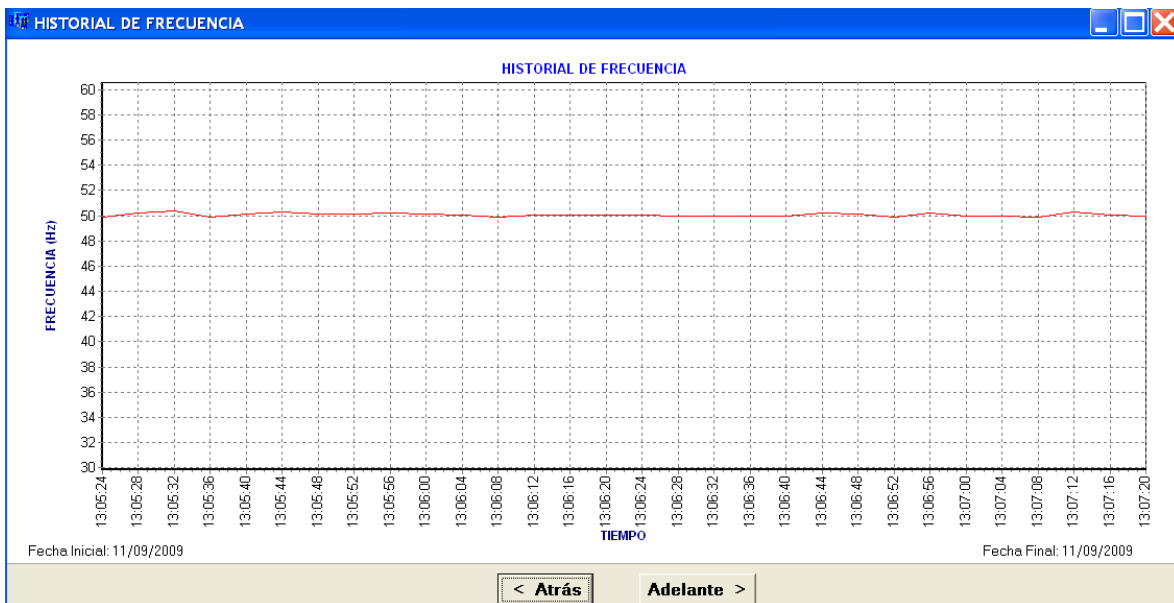


5.2. VISUALIZACIÓN DE DATOS

Los datos adquiridos mediante la comunicación cliente servidor pueden ser visualizados en tiempo real mediante gráficas de comportamiento, en donde se grafica magnitud del fenómeno medido versus tiempo, adicionalmente se tiene la posibilidad de observar datos en forma de comportamiento gráfico de sucesos anteriores, ingresando el intervalo de fecha y hora del cual se quiere observar el comportamiento.

Al presionar el botón adecuado, se abre una ventana que contiene un gráfico con datos provenientes de la base de datos almacenada en el archivo de Excel, el cual se mencionó en el numeral anterior, adicionalmente el entorno gráfico cuenta con visualizadores que indican en valores numéricos el estado actual de la variable medida, para que no se presenten ambigüedades en los datos apreciados en las gráficas.

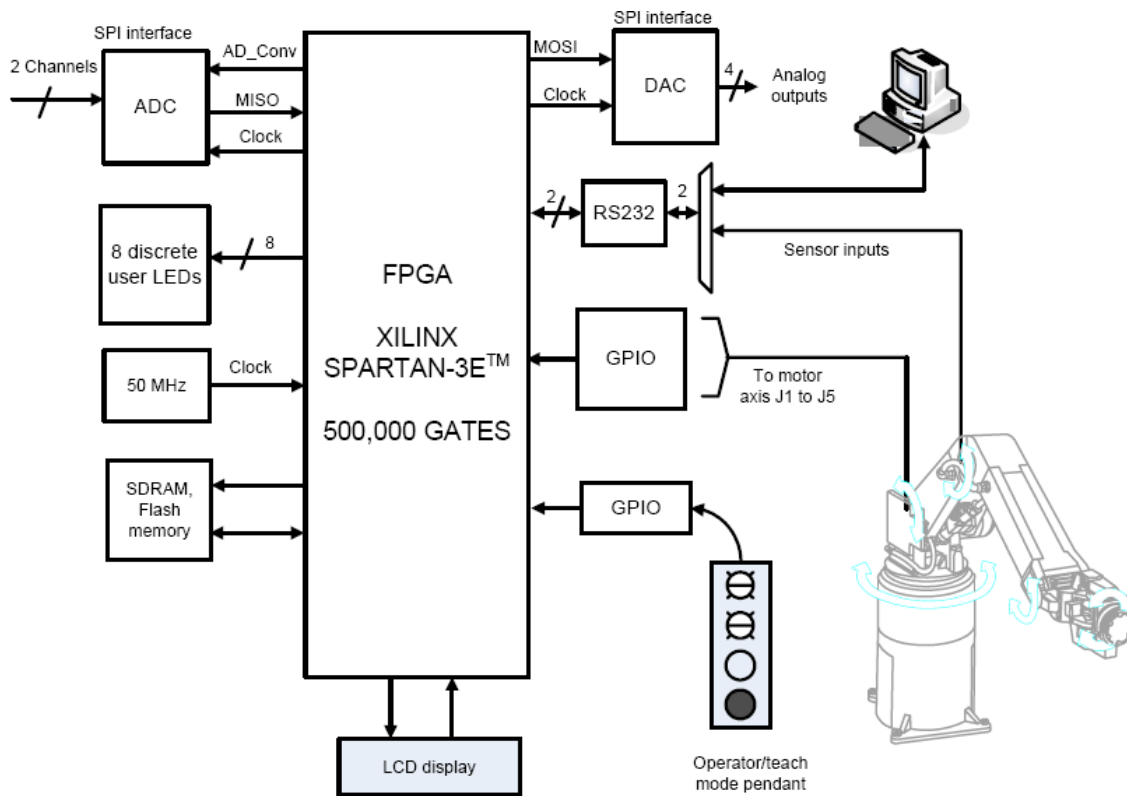
Figura 31. Gráfica de adquisición de valores de frecuencia de salida del variador Yaskawa



6. INTEGRACIÓN DEL SISTEMA

Para la realización de este proyecto se hará uso de la tarjeta de desarrollo Spartan 3E, debido a que es económica, de fácil acceso y con gran capacidad en cuando se refiere a implementar diversos periféricos en la misma. A continuación se muestra un esquema general de la implementación de una FPGA a un proceso industrial.

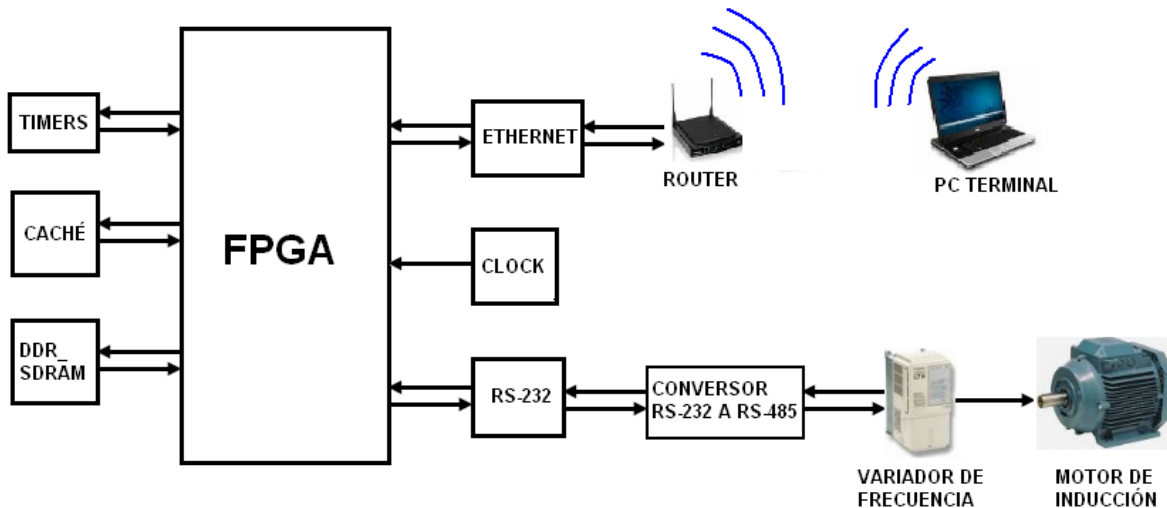
Figura 32. Ejemplo de implementación de la FPGA en un proceso industrial.



Fuente: Introduction to Embedded System Design Using Field Programmable Gate Arrays. India. p.143.

En el caso concreto del presente trabajo de grado, se implementará el siguiente esquema, mostrado de forma general, con el fin de mostrar el correcto funcionamiento del mismo.

Figura 33. Esquema general montaje trabajo de grado.



Los recursos utilizados de la FPGA son administrados por el procesador Microblaze, en el cual es necesario integrar un sistema operativo denominado Xil_kernel, el sistema operativo se hace necesario para controlar el puerto Ethernet y el protocolo TCP/IP el cual proporciona una mayor seguridad en la red, en comparación con el protocolo UDP; se desea implementar el protocolo TCP/IP debido a que uno de los objetivos del proyecto es visualizar un proceso (en este caso es el manejo de un motor de inducción) desde cualquier lugar, por ende se hace indispensable el uso de la Internet, servicio que será suministrado mediante un Router inalámbrico.

Se requiere el uso de timers o *temporizadores* los cuales indican el tiempo de interrupción que se debe manejar en el procesador para que se realicen las operaciones pertinentes como el manejo del sector referente al Ethernet; memoria caché con el fin de agilizar el acceso a algunos datos e instrucciones que el procesador requiere de forma constante debido a la utilización del módulo Ethernet y

la velocidad de procesamiento que éste necesita; el periférico DDR_SDRAM es una memoria tipo volátil que se almacena todo el sistema de archivos que contiene la FPGA, que para el caso concreto es el servidor Echo, el cual se podrá acceder desde Internet y observar los valores relevantes del estado del motor de inducción; el módulo RS-232 es el periférico que permite la conexión entre la FPGA y el variador de frecuencia, esta conexión se hace mediante un puente que para el caso concreto será un conversor RS-232 a RS-485, ya que el variador opera con el protocolo Modbus RS-485.

7. PUESTA EN FUNCIONAMIENTO

Antes de comenzar a tener comunicación con el servidor se deben realizar varias configuraciones y tener varias precauciones en cada parte del montaje para garantizar el correcto funcionamiento de todo el sistema.

7.1. PRECAUCIONES CON EL VARIADOR

La primera precaución que se debe de tener es con la configuración del variador para permitir su funcionamiento en una red Modbus, los parámetros que se deben utilizar son los siguientes:

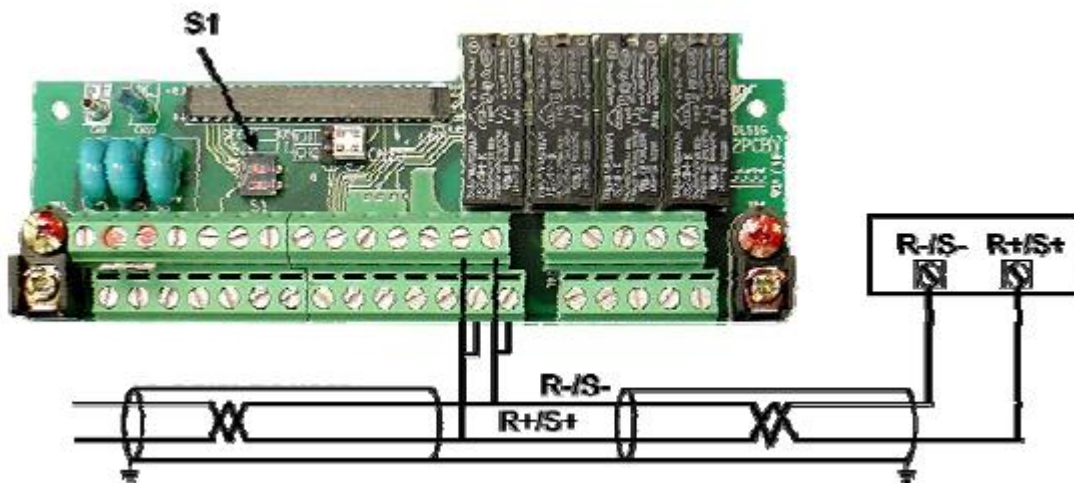
Tabla 2. Registros a configurar en el variador Yaskawa para conectar red Modbus.

Parámetro	Descripción	Valor
b1-01	Fuente de la referencia	2: la referencia se le dará desde el Software mediante la red Modbus
b1-02	Fuente de encendido	0: por seguridad, el encendido se hará desde el panel del variador
C6-02	Frecuencia de la portadora	0: genere el menor ruido (implica mayor ruido audible en el motor)
H5-01	Dirección del esclavo para la red Modbus (1~31)	1
H5-02	Velocidad (en Baudios) de la comunicación	3: velocidad de 9600 Baudios
H5-03	Paridad de la comunicación	0: Sin paridad
H5-06	Tiempo de espera de trasmisión (5~65ms)	10ms
H5-07	Habilitar/Deshabilitar el control RTS	1

Luego de realizar estos cambios se recomienda reiniciar el variador para que dichos cambios tengan efecto.

Para conectar la red Modbus se hace de la siguiente forma:

Figura 34. Esquema de conexión de un variador Yaskawa F7 a una red Modbus.



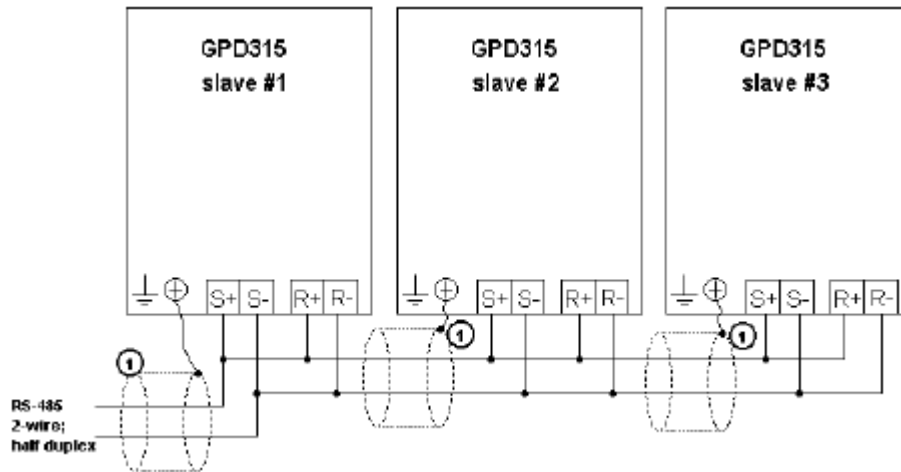
Fuente: YASKAWA [en línea] <Disponible en:
[http://www.yaskawa.com/site/dmdrive.nsf/\(DocID\)/AHUG-6H9QAE/\\$File/TM.F7.11.pdf](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf) > [consulta:
28 Ago.2009] p.10.

En el gráfico anterior se deben resaltar varias cosas:

- Para una red cuyo protocolo es RS-485, se deben de cortocircuitar R+,S+ y R-,S-, donde la línea "A" se conecta con las terminales positivas y la línea "B" con las terminales negativas.
- El cable que se conecta al variador es un cable apantallado, debido a que el variador es un dispositivo que puede introducir mucho ruido a la señal de la comunicación, el apantallamiento debe ir a tierra en ambos extremos, de lo contrario aumentan las posibilidades de ruido en la comunicación.

- En caso tal de tener más de un variador conectado a la red Modbus, al último variador se le debe mover el interruptor de resistencia terminal, lo que indica que ese es el último dispositivo en la red física, de lo contrario se crea un conflicto en la comunicación.

Figura 35. Esquema de conexión de varios variadores Yaskawa a una red Modbus.



Fuente: YASKAWA [en línea] <Disponible en:

[http://www.yaskawa.com/site/dmdrive.nsf/link2/MNEN-5JLRLD/\\$file/TM4325.pdf](http://www.yaskawa.com/site/dmdrive.nsf/link2/MNEN-5JLRLD/$file/TM4325.pdf) > [consulta: 28 Ago.2009] p.8.

7.2. PRECAUCIONES CON LA RED FÍSICA

La red física compone el medio físico por donde se comunican los variadores con la estación central (FPGA), las precauciones son las siguientes.

- El cable apantallado debe ir aterrizado como se explicó anteriormente y procurar que dicho cable esté alejado de cables de potencia que puedan ser fuente de ruido inducido.

- La distancia máxima que se puede tener entre el maestro y los esclavos es de 1200m, al sobrepasar esta distancia, la red RS-485 es inservible a no ser que se haga uso de otros elementos como repetidores.
- El conversor RS-485 a RS-232 debe estar aislado de señales de potencia y su carcasa debe estar debidamente aterrizada.
- En la comunicación RS-232 el pin de transmisión de un dispositivo debe ir con el pin de recepción del otro dispositivo y viceversa, sin embargo se debe de tener en cuenta que la FPGA, invierte de forma interna los pines de recepción y transmisión.
- La FPGA debe estar alimentada de una red de 110v estable, debido a que su tecnología CMOS lo hace un dispositivo “delicado” y de igual forma debe de estar alejada de fuentes de ruido como cables de potencia.

7.3. PRECAUCIONES CON CONEXIÓN A INTERNET

La conexión a Internet involucra el enrutador conectado a la FPGA mediante el cable Ethernet y el PC o terminal del cliente.

- El enrutador al igual que la FPGA deben tener configurados las direcciones IP, las direcciones de compuerta “GATE”, y las direcciones de las máscaras.
- La FPGA contiene una dirección IP estática lo cual evidencia la utilidad del enrutador, sin embargo el enrutador maneja direcciones públicas dinámicas para su acceso a Internet, para solucionar el problema de saber a qué dirección IP se debe conectar cada vez que se utilice el dispositivo, se opta por adquirir una dirección IP fija con el proveedor del servicio de Internet.

- El enrutador debe estar localizado en un lugar que tenga acceso a Internet al igual que el PC desde donde se va a acceder a la estación central.
- Tener conocimiento del Software que se muestra en el presente trabajo de grado.

7.4. MONITOREO Y ANÁLISIS

El monitoreo se realizó con el Software especializado, en primera instancia se capturaron los datos por un tiempo determinado operando a una frecuencia de 50 Hz, luego se procedió a observar el comportamiento del motor mediante los parámetros medidos por el variador, observando los historiales se observan los siguientes comportamientos.

Figura 36. Gráfica de adquisición de valores de frecuencia de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.

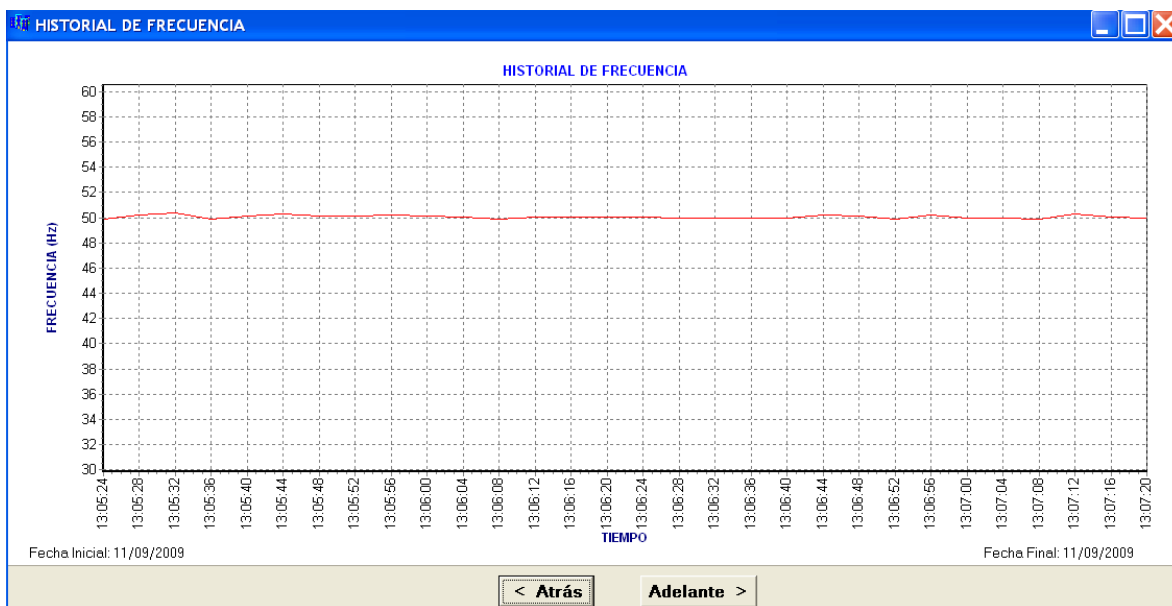


Figura 37. Gráfica de adquisición de valores de corriente de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.

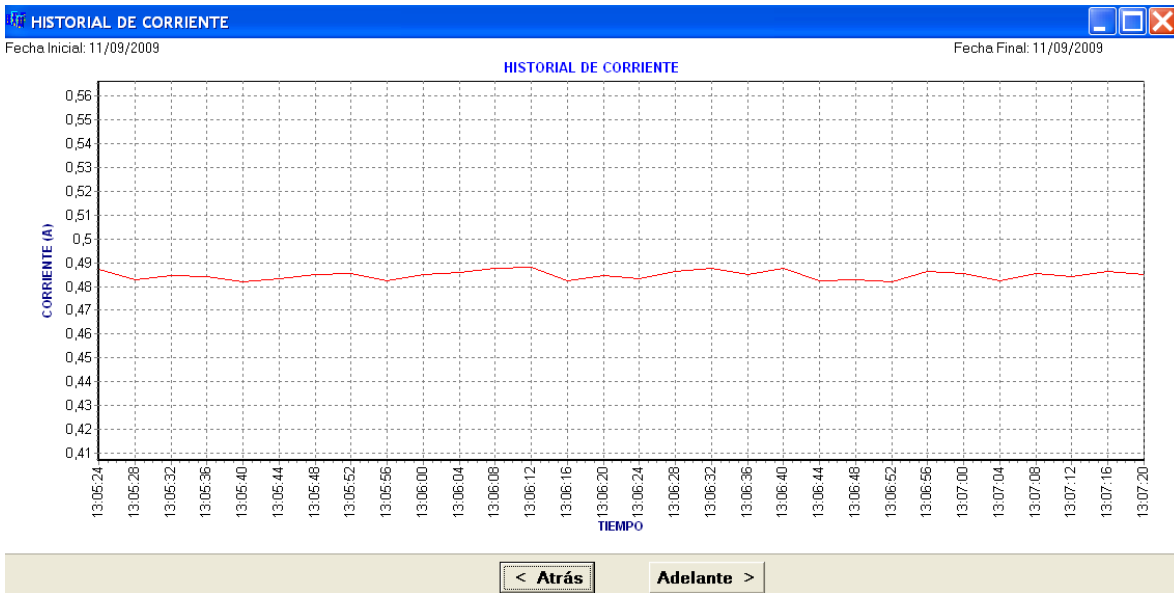


Figura 38. Gráfica de adquisición de valores de velocidad de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009

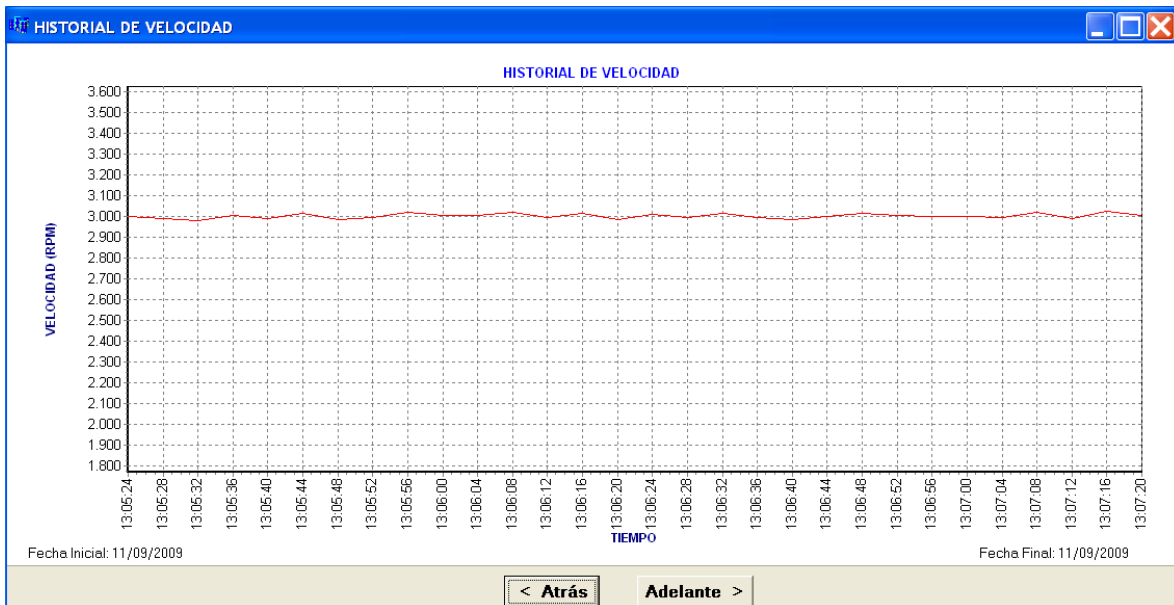
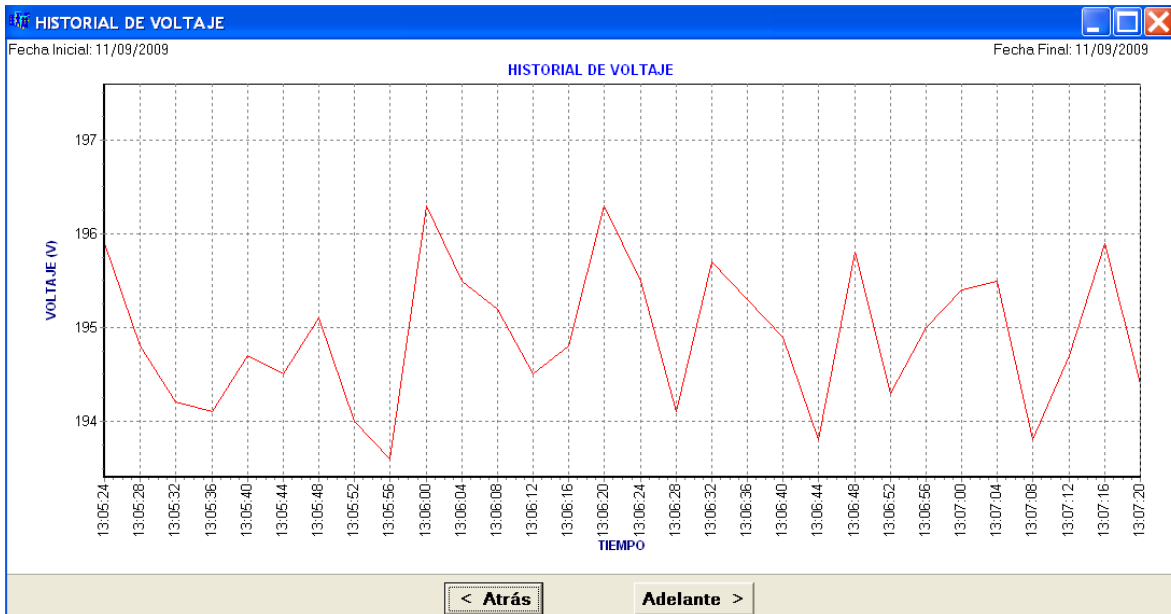


Figura 39. Gráfica de adquisición de valores de voltaje de salida del variador Yaskawa F7 tomada el 11 de Septiembre de 2009.



Las gráficas antes vistas representan los parámetros más importantes a conocer en primera instancia, el Software permite observar 30 puntos por ventana, con la posibilidad de desplazarse hacia atrás o adelante dependiendo de la necesidad para observar todos los puntos que estén incluidos en el historial y tiene la posibilidad de aumentar el rango de visión para tener observar una lectura más precisa.

Lo que se puede observar de las gráficas de comportamiento de los parámetros es que los valores medidos por el variador presentan una variación poco pronunciada, lo cual indica una confiabilidad en estos instrumentos; el parámetro que presento más variaciones fue la salida de voltaje de AC (Figura 38), esto se debe a que la señal de voltaje es la más vulnerable en cuanto a ruido en el ambiente se refiere, es por eso que las señales que se manejan generalmente en la industria son de corriente (4mA~20mA).

Por otro lado el parámetro más estable pertenece a la frecuencia de salida (Figura 35) a la cual opera el variador, lo cual le otorga más credibilidad al dispositivo llamado variador de frecuencia.

Conocer estos parámetros resulta de mucha utilidad, pues con base a estos se puede realizar una acción de control determinada para corregir algún proceso o apagado de emergencia en el caso tal que así se requiera, logrando un proceso más automatizado, lo cual disminuye la interacción de operario con las máquinas y así eliminando la posibilidad de riesgo, sin mencionar la ventaja de poder almacenar cada valor medido con el fin de posibilitar un análisis futuro si se presenta una falla y poder conocer de formas más ágiles la causa del problema.

7.5. CONTROL REMOTO

La automatización no sólo implica la visualización de variables, también implica la acción de control sobre los dispositivos que para el caso concreto es el variador de velocidad, sin embargo hay que tener mucha precaución con el control remoto, debido a que tener la posibilidad de manipular cualquier parámetro de estos dispositivos de forma remota, no significa que sea prudente manipularlos, esto con el fin de aumentar la seguridad en las estaciones de trabajo, es por eso que la opción de control en el presente trabajo de grado se limita sólo a cambiar la frecuencia de referencia para el funcionamiento del motor, solo con fines ilustrativos; en una aplicación industrial, el cambio de la frecuencia de referencia podría ser fatal para los procesos que se lleven a cabo, debido a que la frecuencia influye directamente en la velocidad del motor lo cual podría descuadrar los tiempos de operación de los procesos, por ende el cambio de la frecuencia de referencia se hace a modo ilustrativo.

Figura 40. Selección de la frecuencia de referencia para el variador Yaskawa desde el Software de control de la estación central del sistema SCADA.

¿USTED YA CREÓ EL ARCHIVO Datos_SCADA.xls?

Si, ya creé el archivo Datos_SCADA.xls

No, no he creado el archivo Datos_SCADA.xls

Crear Archivos Para Historiales

RANGO DE FECHAR PARA MOSTRAR HISTÓRICO

Día inicio	Mes inicio	Año	Hora inicio	Min inicio
27	7	2009	15	18
Día final	Mes final	Año	Hora final	Min final
27	7	2009	12	13

Mostrar por fechas Mostrar todo

ENVIAR REF. 1000 cHz

1000 cHz=60Hz

2000

3000

4000

5000

6000

8. CONCLUSIONES

- Los sistemas embebidos se presentan como una herramienta muy utilizada para desarrollar soluciones ante una situación o requerimiento específico, debido a que sólo se hace uso de los componentes necesarios para llevar a cabo la tarea en cuestión, lo cual se manifiesta en costos de producción menores, esto implica procesadores de menor capacidad, periféricos con menor velocidad en cuanto al intercambio de información se refiere y memoria de almacenamiento más reducida, sin embargo a pesar de estas limitaciones, las tareas encomendadas se pueden cumplir sin complicaciones, gracias a que los sistemas embebidos sólo tienen esa tarea específica por cumplir.
- Un sistema SCADA se hace cada vez más indispensable a nivel industrial, debido al crecimiento de la tecnología con aplicaciones autómatas para la producción, conlleva a que se implemente cada vez más esta tecnología, disminuyendo la mano de obra y ampliando los procesos de producción, lo cual dificulta la visualización del estado de las máquinas que influyen en la producción, con un sistema SCADA se puede visualizar el estado de las maquinarias sin importar la ubicación geográfica de las mismas.
- La estación central que se representa por la FPGA, la cual es el corazón del sistema SCADA, se presentó como una solución muy viable para solucionar necesidades de control a nivel industrial, debido a que dicha tarjeta contiene embebido una serie de periféricos y elementos que facilitan la acción de control sobre los actuadores presentes de la industria y la comunicación de dichas acciones con terminales remotas.

- Si se compara una FPGA con un controlador común en la industria Colombiana como lo son los PLCs, se encuentra que la FPGA presenta una ventaja monetaria claramente favorable, debido a que el procesador del PLC de por sí cuesta más que la FPGA, adicionalmente se deben comprar módulos adicionales de comunicación y de adecuación de señales (si se trabaja con señales análogas) para igualar los componentes que se traen embebidos en la FPGA, sin mencionar que el procesamiento de los PLCs son de forma serial, lo cual lentifica el proceso de control.
- La Spartan 3E al ser una tarjeta de desarrollo netamente electrónica que trabaja con tecnología CMOS, presenta una mayor susceptibilidad frente a señales de alta amplitud como las encontradas en la industria, es por eso que se debe desarrollar un sistema fuerte de protección para adecuar las señales a los valores permitidos por la tarjeta embebida, por ende es una buena técnica de control, implementar dicha acción por los puertos propios de comunicación de los actuadores siempre y cuando estos los posean, como es el caso del variador de frecuencia y su puerto de comunicación por el protocolo RS 485, así se disminuye los riesgos de daños en la FPGA.
- La versatilidad que presenta la FPGA permite implementar varios protocolos de comunicación siempre y cuando estos se conozcan, como se observó en el presente trabajo, se implementó el protocolo de comunicación del variador de frecuencia para ser controlado, sin la necesidad de tener el Software propio del variador, esto amplía las posibilidades de la FPGA como estación central de control, debido a que se pudo procesar las señales de control y respuesta del variador y se logró realizar una visualización en un entorno gráfico amigable para el usuario.

- La conexión de la FPGA a Internet presenta algunos contratiempos debido a que el proveedor local de servicios de Internet presentan una alta demanda por parte de los usuarios, lo cual implementan el concepto de IP dinámica para suplir dicha demanda, si se quiere aplicar el concepto de la FPGA como un servidor adecuado a nivel industrial, se debe optar por adquirir una IP estática por parte del proveedor del servicio.
- El presente trabajo de grado muestra que es posible manipular dispositivos robustos como son los motores, a partir de componentes electrónicos que requieren poco voltaje para su funcionamiento, indicando que la presente tesis puede ser utilizada como base para realizar prácticas en los laboratorios de la universidad utilizando los motores y variadores con los que se dispone.

BIBLIOGRAFÍA

AGUAYO E; GONZÁLEZ I; BOEMO E. Tutorial Xilinx MicroBlaze. España: Universidad Autónoma de Madrid, Escuela Politécnica Superior. 2005. 9p.

BERMUDEZ ZAPATA, Julián Alberto. Desarrollo del manual y guía para el control de máquinas de inducción por medio de variadores de frecuencia con simulación computacional. Medellín, 2009. Tesis (Ingeniero Electricista). UPB. Escuela de ingeniería Eléctrica. 84 p.

DYMAEL, S.L. Sistemas de adquisición de datos [en línea]. s.p.i. <Disponible en <http://dymael.galeon.com/automatizaciones/sisscada.htm>> [Consulta: 7 Ago. 2009]

DUBEY, Rahul. Introduction to Embedded System Design Using Field Programmable Gate Arrays. India, 2008, 153 p.

EMBARCADERO TECHNOLOGIES. C++ Builder. [en línea]. <Disponible en: <http://www.embarcadero.com/products/cbuilder/>> [Consulta: 21 Jul. 2009]

JIMÉNEZ BUENDÍA, Manuel. Protocolo Modbus. Universidad Politécnica de Cartagena, Departamento de Tecnología Electrónica- Comunicaciones Industriales. Cartagena, 10p.

Mi Mecánica Popular. Cómo trabaja un motor eléctrico [en línea]. s.p.i. <Disponible en <http://www.mimecanicapopular.com/vergral.php?n=180>> [Consulta: 29 Jun. 2009]

Medida y control. Scada para cámaras frigoríficas [en línea]. s.p.i. <Disponible en <http://usuarios.lycos.es/mecofi/>> [Consulta: 3 May. 2009]

Wikimedia Foundation , Inc. Núcleo (informática) [en línea]. s.p.i. <Disponible en [http://es.wikipedia.org/wiki/N%C3%B3cleo_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/N%C3%B3cleo_(inform%C3%A1tica))> [Consulta: 29 Jun. 2009]

Wikimedia Foundation , Inc. Códigos NRZ [en línea]. s.p.i. <Disponible en http://es.wikipedia.org/wiki/C%C3%B3digos_NRZ> [Consulta: 09 Jul. 2009]

Wikimedia Foundation , Inc. Modelo OSI [en línea]. s.p.i. <Disponible en http://es.wikipedia.org/wiki/Modelo_OSI > [Consulta: 20 Jul. 2009]

Wikimedia Foundation , Inc. RS-232 [en línea]. s.p.i. <Disponible en <http://es.wikipedia.org/wiki/RS-232>> [Consulta: 17 Ago. 2009]

Wikimedia Foundation , Inc. RS-485 [en línea]. s.p.i. <Disponible en <http://es.wikipedia.org/wiki/RS-485>> [Consulta: 17 Ago. 2009]

Xilinx®. OS and Libraries Document Collection [en línea]. EDK 10.1 Service Pack 3.

<Disponible en:

http://www.xilinx.com/support/documentation/sw_manuals/edk10_oslib_rm.pdf>

[Consulta: 28 Jun. 2009]

Xilinx®. Ds312 (v3.7). 2008. 234 p. [en línea]. <Disponible en:

http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf> [Consulta: 28

Jun. 2009]

Xilinx®. Spartan-3E Starter Kit Board User Guide [en línea]. UG230 (v1.1). 2008.

<Disponible en:

http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf> [consulta:

15 Jun. 2009]

Xilinx®. MicroBlaze Processor Reference Guide [en línea]. UG081 (v5.1). 2005.

<Disponible en: http://www.xilinx.com/ise/embedded/edk7_1docs/mb_ref_guide.pdf>

[Consulta: 28 Jun. 2009]

YASKAWA. Manual de Instrucciones para los Variadores de Frecuencia de control

Vectorial. Medellín: Variadores S.A. 2005. 174 p.

YASKAWA. F7U Drive Parameter Access Technical Manual [en línea]. TM.F/7.11.

<Disponible en: [http://www.yaskawa.com/site/dmdrive.nsf/\(DocID\)/AHUG-](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf)

[6H9QAE/\\$File/TM.F7.11.pdf](http://www.yaskawa.com/site/dmdrive.nsf/(DocID)/AHUG-6H9QAE/$File/TM.F7.11.pdf)> [Consulta: 28 Ago. 2009]

YASKAWA. GPD 315/V7 Modbus RTU Technical Manual [en línea]. TM4325.
<Disponible en: [http://www.yaskawa.com/site/dmdrive.nsf/link2/MNEN-5JLRD/\\$file/TM4325.pdf](http://www.yaskawa.com/site/dmdrive.nsf/link2/MNEN-5JLRD/$file/TM4325.pdf)> [Consulta: 28 Ago. 2009]

LISTA DE ANEXOS

ANEXO A. Anteproyecto trabajo de grado

ANEXO B. Plano conversor RS-232 a RS-485

ANEXO C. Pasos para configurar la FPGA

ANEXO D. Rutina de cálculo del CRC-16

ANEXO E. Artículo publicable

Nota: los códigos del Software de control de la estación central y el de la FPGA no se anexan, debido a que los archivos también se anexan en el trabajo de grado.