

UNIVERSIDAD DE SONORA
DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE INVESTIGACIÓN EN FÍSICA
LICENCIATURA EN TECNOLOGÍA ELECTRÓNICA

**SISTEMA DE ADQUISICIÓN DE DATOS DE UNA CCD PARA
UN PROCESADOR DE SEÑALES DIGITALES**

TESIS

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN TECNOLOGÍA ELECTRÓNICA**

**PRESENTA:
GERARDO GUTIÉRREZ HEREDIA**

**DIRECTORES DE TESIS
DR. LUIS ALFREDO GONZÁLEZ LOPEZ
DR. FERNANDO MENDOZA HERNÁNDEZ**

Hermosillo, Sonora, octubre de 2005.

Universidad de Sonora

Repositorio Institucional UNISON



**"El saber de mis hijos
hará mi grandeza"**



Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

Con mucho amor dedico este trabajo

...a mi **Padre** y a mi **Madre**

Agradecimientos...

...a los maestros de la Universidad De Sonora que me ayudaron con sus enseñanzas en especial a mi Padre y a los maestros de la carrera de Electrónica: Luis González, Fernando Mendoza, Benito Noriega, Alicia Vera, Raúl Castillo, Sara Loreli.

A mis asesores de Tesis Dres. Luis González y Fernando Mendoza.

El apoyo brindado por CONACYT con la beca del proyecto 44071-A1 “Investigación y desarrollo de sistemas óptico-digitales para el procesamiento de información tridimensional”.

A mi maravillosa familia, por su apoyo incondicional durante toda mi vida de estudiante, en especial a mi padre, madre, a mis hermanos Rosa y Toño.

Con mucho cariño a Cinthia M. por su gran apoyo durante todos estos años.

A los 5 conejillos de India: Nacho C, Benja H, Marcos A. y Kano C.

Amigos que estuvieron conmigo durante esta etapa, Diana, Alina, Ale L, Lupita L, Jazz, Lupita P, Ely, Beto G, a la gente de computación, electrónica y en especial a Janneth R.

Y a la inolvidable pandilla de los Nerds, los mejores amigos: Jazz, Kano, Neto, Caleb, Zepi, Main, Norman, Hugo, Eliseo y a la gran ayuda en este trabajo de mi amigo Adrian.

INDICE

1. INTRODUCCIÓN	1
1.1. Organización del Documento	2
2. CÁMARA CCD	3
2.1. Chip CCD	3
2.2. Registro de transferencia de datos	5
2.3. Dalsa CA-D8	5
3. DISPOSITIVOS RECONFIGURABLES Y PROCESADORES DE SEÑALES DIGITALES	8
3.1. Introducción	8
3.2. Dispositivos lógicos reconfigurables	9
PLD	9
CPLD	10
FPGA	11
XC2S200E	13
3.3. Procesadores de señales digitales	17
Clasificación	21
TMS32C6711	21
Arquitectura del C6711	22
3.4. Conclusiones	23
4. SISTEMA DE ADQUISICIÓN DE DATOS	25
4.1. Diagrama general	25
4.2. LVDS	26
4.3. Memoria FIFO	28
4.5. Recepción de la imagen a la memoria (CCD-FIFO)	30
4.6. Transferencia de imagen al DSP (FIFO-DSP)	31
4.7. PCB	33

5.	PRUEBAS ELECTRICAS Y RESULTADOS	34
5.1	Convertidores LVDS-TTL y TTL-LVDS	35
5.2	Señales de la cámara CCD	38
5.3	Señales de control por el FPGA	42
5.4	Memoria FIFO (Simulación de lectura del DSP)	47
6.	CONCLUSIONES	52
	Bibliografía	53
	Apéndice A. <i>PERFILOMETRIA</i>	55
	Apéndice B. <i>EMIF</i>	58
	Apéndice C. PCB	62
	Apéndice D. <i>Código VHDL</i>	66

LISTA DE FIGURAS

Figura 2.1. Fotografía de un sensor CCD.	4
Figura 2.2. Almacenamiento de fotones en cada pixel.	4
Figura 2.3. Representación de la transferencia de información del registro.	5
Figura 2.4. Cámara Dalsa CA-D8.	6
Figura 2.5. Señales de control de la cámara CA-D8.	7
Figura 3.1 Representación de una suma lógica dentro de un PLD.	10
Figura 3.2. Representación interna de un FPGAs.	12
Figura 3.3. Diagrama de interconexiones programables con celdas básicas.	12
Figura 3.4. Diagrama General del XC2S200E.	14
Figura 3.5. Diagrama de una placa de una celda básica.	16
Figura 3.6. Diagrama interno de una celda básica.	17
Figura 3.7. Arquitectura de procesadores de computadoras.	19
Figura 3.8. Diagrama de bloque del TMS320C671x.	22
Figura 4.1. Diagrama general del Sistema de Adquisición.	26
Figura 4.2. Diagrama de un circuito de comunicación LVDS.	27
Figura 4.3. EMI en líneas de transmisión.	27
Figura 4.4. Grafica de frecuencia contra corriente de protocolos.	28
Figura 4.4.1 Bloques de funcionamiento de la memoria FIFO.	29
Figura 4.5. Terminales de conexión de cámara CA-D8 de DALSA.	30
Figura 4.6. Terminales de conexión de memoria FIFO.	31
Figura 4.7. Terminales de conexión del EMIF.	32
Figura 5.1. Sistema de conversión TTL-LVDS- TTL (DS90C031- DS90C032).	35
Figura 5.2. Salida positiva del LVDS en alto y negativa en bajo.	36
Figura 5.3. Salida positiva del LVDS en bajo y negativa en alto.	36
Figura 5.4. Diferencial de señales cuadradas emitidas por un generador de señales.	37
Figura 5.5. Diferencial de señales cuadradas por separado.	37
Figura 5.6. Recepción por un LVDS de una señal cuadrada.	38
Figura 5.7. Señal de STROBE del CCD.	39
Figura 5.8. Señales de FVAL (Superior) y LVAL (Inferior) del CCD.	39
Figura 5.9. Señales de FVAL (Superior) y LVAL (Inferior) del CCD.	40
Figura 5.10. Señales de Datos D1 y D2 del CCD.	41
Figura 5.11. Señales de Datos D3 y D4 del CCD.	41
Figura 5.12. Señales de Datos D7 y D8 del CCD.	42
Figura 5.13. Ejemplo de señal EXSYNC dada por el FPGA.	43
Figura 5.14. Señal EXSYNC (FPGA) después de la conversión a LVDS.	44
Figura 5.15 Señal FVAL con retraso.	45
Figura 5.16. Pulso de RESET dado por el FPGA.	45
Figura 5.17. Señales de EXSYNC (Superior) y RESET (Inferior).	46
Figura 5.18 Diagrama a bloques del código de VHDL.	47
Figura 5.19. Señales de Datos Q1 y Q2 de la FIFO.	48
Figura 5.20. Señales de Datos Q3 y Q4 de la FIFO.	48
Figura 5.21. Señales de Datos Q7 y Q8 de la FIFO.	49
Figura 5.22. Señal de HF de la FIFO (inferior) señal de RESET del FPGA de 50 KHz (superior).	50
Figura 5.23. Señal de AF/AE de la FIFO (inferior) señal de RESET del FPGA de 12KHz (superior).	50
Figura 5.24. Señal de AF/AE de la FIFO (inferior) señal de RESET del FPGA de 6KHz (superior).	51
Figura A-1. Figura tomada por una CCD con el patrón de franjas.	55
Figura A-2. Información de una línea de datos.	56
Figura A-3. Información de una línea de datos en dominio de la frecuencia.	56
Figura A-4. Figura reconstruida en tres dimensiones.	57
Figura B.1. Diagrama general del DSP.	58
Figura B.2 Terminales de entrada/salida de la EMIF del DSP.	60

<i>Figura C.1. Diseño esquemático del PCB</i>	63
<i>Figura C.2. Diseño de Layout del PCB.</i>	64
<i>Figura C.3. Anverso del Layout del PCB (lado de componentes).</i>	64
<i>Figura C.4. Reverso del Layout del PCB.</i>	65

LISTA DE TABLAS

<i>Tabla 3.1. Familia de FPGAs Spartan2E</i>	15
<i>Tabla 3.2. Familias más importantes de DSP</i>	21
<i>Tabla 3.3. Características principales del TMS320C6711</i>	23
<i>Tabla 4.1. Terminales de Conexión del EMIF</i>	32
<i>Tabla B.1. Registros de la EMIF del DSP.</i>	59
<i>Tabla B.2. De Terminales de la EMIF del DSP.</i>	60

Capítulo 1. INTRODUCCIÓN

La medición de contornos o figuras en tres dimensiones, es muy utilizada en el reconocimiento de rostros, en la industria para verificación de calidad de materiales, visión robótica, entre otras aplicaciones. A esta medición se le conoce como Perfilometría (Profilometry)

Existen varios procesos para la reconstrucción de imágenes en tercera dimensión como métodos ópticos. Un método muy comúnmente utilizado es el que se llevan a cabo por medio de procesamiento de señales o de imágenes utilizando cámaras de CCD.

Un método para la reconstrucción de imágenes en tres dimensiones es la "Perfilometría por Transformada de Fourier" o FTP (Fourier-Transform Profilometry) empleando la técnica de "Contornos de Moire", la cual al proyectarse un patrón de franjas de luz sobre un objeto, analiza las deformaciones de las franjas siendo la información o los datos que se deben de procesar para llevar a cabo la reconstrucción. [Apéndice A] [10].

Para llevar a cabo el procesamiento de imágenes se utiliza comúnmente una computadora personal con una interfaz (*Framegrabber*) para una cámara digital. La desventaja de utilizar una computadora de uso general es que el sistema operativo utiliza capacidad de cómputo en otras tareas prolongando así el tiempo necesario para el procesamiento de las imágenes e imposibilitando su procesamiento en tiempo real.

La propuesta del proyecto 44071-A de CONACYT, donde se incluye este trabajo de tesis es desarrollar un sistema que lleve a cabo la reconstrucción de un objeto en tres dimensiones empleando sistema de co-procesamiento independiente de una computadora. Para ello se emplearán dispositivos como DSPs (Digital Signal Processors) y un FPGAs (Field Programmable Gate Array). El propósito de esta tesis es desarrollar una interfaz o sistema de adquisición de datos entre una cámara CCD (Charge Coupled Device) DALSA CA-D8 y el DSP TMS320C6711 mediante el control de una memoria FIFO (First In First Out).

El control de dispositivos se lleva a cabo con un FPGA, enviando señales de sincronización o de inicio a otros dispositivos como la cámara CCD y la memoria FIFO, para que finalmente el DSP lea la información y la procese.

EL empleo de un DSP aumenta la eficiencia el sistema de reconstrucción debido a que el DSP es un procesador específicamente para el procesamiento señales, siendo mas rápido comparado con uno de uso general en estos trabajos. Mediante el EMIF (External Memory Interface) se realiza la lectura de datos sin hacer uso de direccionamiento como lo requieren las memorias RAM. Esta es la razón por la que en esta interfaz se utiliza una memoria FIFO.

1.1 ORGANIZACIÓN DEL DOCUMENTO

Esta tesis esta organizada primeramente en su capítulo 2 por la descripción general de las características de una cámara CCD digital y las señales que emplea la cámara usada en este trabajo. El capítulo 3 muestra una descripción general de los dispositivos digitales reconfigurables y los procesadores digitales de señales; específicamente se muestran las características de los dispositivos XC2S200E y TMS320C6711 empleados para el desarrollo del proyecto general, considerando para esta tesis los periféricos necesarios. El desarrollo del sistema de adquisición se describe en el capítulo 4, comenzando por una revisión de las características del protocolo EIA-644 y dispositivos de conversión LVDS a TTL y TTL a LVDS. Así también se muestran las características de la memoria FIFO empleada (SN74ACT7807) y los procesos de lectura y escritura de datos. En este capítulo también se describe la interfaz resultante mediante un circuito impreso (PCB) que alberga los convertidores LVDS a TTL y TTL a LVDS y memoria FIFO. Para comprobar la ejecución de nuestra interfaz se muestran resultados de pruebas eléctricas experimentales de cada dispositivo en el capítulo 5. Finalmente, el capítulo 6 presenta las conclusiones y comentarios de trabajo futuro.

Capítulo 2. CÁMARA CCD

Para extraer la información tridimensional del contorno de un objeto por medio de la perfilometría por transformada de Fourier, se requiere proyectar un patrón de franjas. Este patrón bidimensional deformado por el objeto se adquiere a través de una cámara que contiene un CCD (Charge Coupled Device) o Dispositivo de Acople de Carga, el cual consta de una matriz de sensores (píxeles) que almacenan luz para conformar un arreglo denominado *frame*. Como ejemplo, en las cámaras fotográficas digitales este arreglo hace las veces de la película donde se registra la luz que significa una imagen.

Cada píxel de la cámara CCD convierte en electrones los fotones que recibe. La cantidad de electrones almacenados en cada píxel es proporcional a la intensidad de luz que reciben. Las cargas almacenadas en cada píxel son enviadas a un registro de corrimiento que almacena la intensidad de corriente de una línea de píxeles. Este registro, a su vez, reenvía dicha información a otro circuito electrónico el cual pueda almacenar la información. Este mecanismo es el que utilizan la mayoría de las cámaras digitales de fotografía o de video.

Una cámara convencional obtiene una fotografía plasmando la luz en una película fotosensible, la cantidad de luz reflejada en esta es controlada por un mecanismo llamado obturador que se abre y cierra en un tiempo determinado, controlando la cantidad de fotones que llegan a la película. Si éste permanece abierto demasiado la fotografía resultaría en una imagen blanca.

Una cámara CCD es muy similar a la convencional, con la diferencia que reemplaza la película por un sensor electrónico de luz. La "fotografía" captada se conoce como *frame*. Asimismo, como en una cámara convencional, donde se controla el tiempo de exposición mediante un obturador, también se puede manipular el tiempo de exposición del CCD con un control electrónico para matizar la imagen. [1][2][3]

2.1. CHIP CCD

El CCD es un sensor compuesto por una matriz de capacitores MOS uniformemente espaciados que atrapa la luz de la imagen que se desea fotografiar y, mediante el efecto fotoeléctrico, convierte los fotones capturados en cada capacitor o pixel en carga eléctrica. La cantidad de carga eléctrica almacenada en cada pixel depende de la intensidad de luz en dicho punto tomada de la imagen. La transferencia y lectura de las cargas almacenadas en cada pixel se realiza mediante fases de reloj sincronizadas que se conectan a los pixeles del CCD. En la Fig. 2.1 se muestra un lente con un sensor de CCD en el interior. En la Fig. 2.2 se tiene un ejemplo de cómo recibe y almacena cada pixel la intensidad de luz.

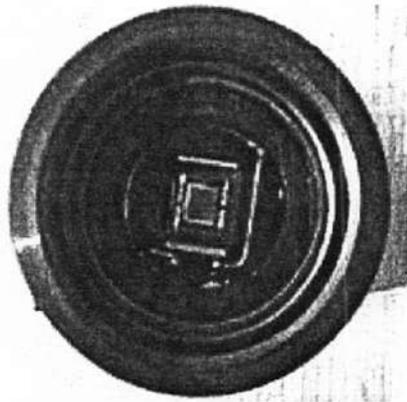


Figura 2.1. Fotografía de un sensor CCD

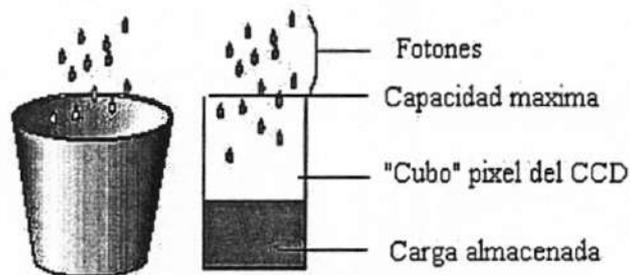


Figura 2.2. Almacenamiento de fotones en cada pixel.

La información almacenada en cada pixel, es decir, las cargas almacenadas, se convierten en niveles de voltaje y posteriormente se digitalizan usando un convertidor analógico-digital (A-D). El convertidor A-D empleado maneja los datos electrónicos en

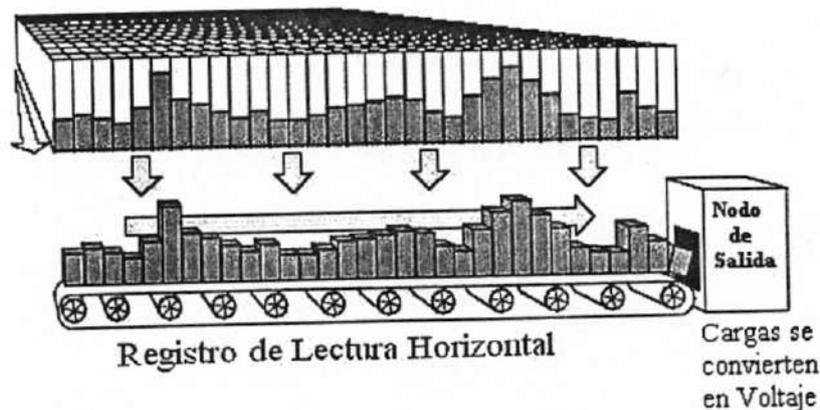
valores de 8 bits, es decir, representa la intensidad de luz de cada pixel con un número entre 0 y 255, reflejando los tonos de gris. [1][2][3].

2.2 REGISTRO DE TRANSFERENCIA DE DATOS

La imagen capturada en el CCD debe ser enviada a una memoria o dispositivo de almacenamiento externo. Para esto, el registro de transferencia de datos se encarga de transferir la información desde el CCD.

Para leer la imagen capturada se efectúa un barrido de la matriz de pixeles línea por línea y se va almacenando una a una en el registro de lectura horizontal, el cual envía los datos de cada pixel al exterior. Al leerse una línea en forma horizontal, se continúa con la siguiente hasta el final del *frame*, ver la Fig. 2.3. [12].

512 líneas de 512 pixeles



2.3. Representación de la transferencia de información del registro.

2.3. DALSA CA-D8

Para la recepción de la imagen del sistema se utiliza una cámara CCD, de la empresa DALSA, modelo CA-D8, (Fig. 2.4). Algunas características de ésta CCD son:

- 512V x 512H pixeles.
- Salida de 25 MHz (Velocidad de 77 frames/segundo).
- Señales de datos y control en formato EIA-644 (LVDS).
- 8 bits de datos.



Figura 2.4. Cámara Dalsa CA-D8

Los controles principales de la cámara permiten manipular la exposición del sensor del CCD y el envío de *frames*. La cámara CCD posee un obturador controlado electrónicamente para determinar el tiempo de exposición de los sensores a la luz que llenará la matriz. A partir de este punto se espera la señal de petición de un control externo para leer la información de la cámara y transferirla a una memoria.

La cámara también maneja señales de banderas para que el usuario o sistema que esté utilizándola, tenga control sobre los datos enviados por la cámara. Estas banderas son las de FVAL (*Frame Valid*) y LVAL (*Line Valid*). La bandera LVAL es utilizada para dar a conocer que la CCD está enviando una línea de píxeles, durante el envío la señal permanece en alto ocurriendo una transición en cada cambio de línea. De igual manera, FVAL se encuentra en alto mientras se envía un *frame*.

Otra señal utilizada por la cámara, que también es enviada al bus de salida, es la de reloj, también conocida como STROBE, y trabaja a una frecuencia de 25 MHz. Las señales de control de la cámara son las que manipulan el inicio de envío de un *frame* y la exposición de la CCD. Estas señales son EXSYNC y PRIN, ambas de uso opcional.

La señal de EXSYNC es utilizada para controlar las lecturas o transferencias de *frame*. Si esta señal se tiene en bajo, los datos de salida de la cámara estarán operando en el modo "*Maximum Frame Rate*" o modo de trabajo libre. Cuando la señal de

EXSYNC es controlada por el cambio flanco de subida, indica la lectura de un *frame* y de ésta manera, puede ser determinada la frecuencia de lectura. La duración mínima de ésta señal debe ser de 100 ns, en alto o en bajo.

La señal PRIN puede ser utilizada para reducir el tiempo de exposición efectiva para el reseteo de los pixeles (drenado de la carga acumulada) en una imagen entre cada lectura de *frame* por cada flanco de subida de EXSYNC. PRIN se activa en bajo y comienza la exposición del CCD cuando hay un flanco de subida. Si ésta señal está desconectada o se encuentra en alto, se maximiza el tiempo de integración o exposición del CCD. Si se encuentra en un estado bajo el CCD no se expone por lo que no c tiene ninguna imagen. El pulso de mínimo de PRIN debe de ser por lo menos de 7.5 μ s. Durante el periodo de transferencia del *frame* la cámara ignora las entradas que se tenga en PRIN.

La Fig. 2.5 muestra una grafica de tiempo de los pulsos y la forma de funcionamiento de la CCD. [12].

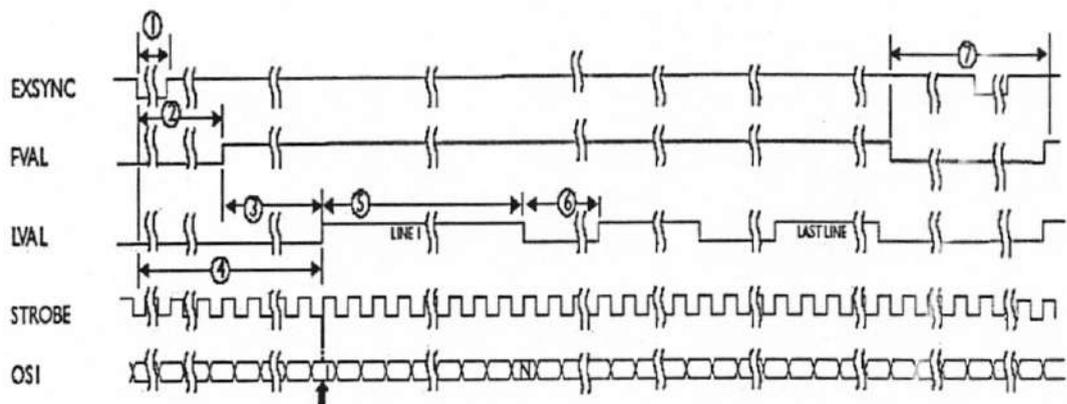


Figura 2.5. Señales de control de la cámara CA-D8.

Capítulo 3. DISPOSITIVOS RECONFIGURABLES Y PROCESADORES DE SEÑALES DIGITALES

3.1. INTRODUCCION

Algunos dispositivos programables como los FPGAs (Field Programmable Gate Array) y PLDs (Programmable Logic Device) utilizan estructuras programables básicas para circuitos mas complejos como contadores, registros, memorias o incluso un procesador. Por lo tanto, se puede desarrollar un control digital para algún sistema en específico usando esta tecnología. La ventaja de este dispositivo es que solo basta hacer una programación adecuada para que se lleve a cabo la función o tarea deseada, y de esta manera, se evita la necesidad de desarrollar un circuito digital que realice el mismo fin.

Debido a que este sistema de adquisición de señales, o tarjeta de interfaz, demanda un control que tenga la organización de todos los dispositivos (señales de control de CCD, lectura y escritura de dispositivos de memoria, DSP, etc.), se ve la necesidad de utilizar un FPGA, haciendo mas fácil el desarrollo del control digital del sistema por su facilidades de manejo y programación.

Otro dispositivo que también es utilizado en el desarrollo de este sistema es un DSP (Digital Signal Processor), el cual procesara los datos adquiridos mediante nuestra interfase a una velocidad alta en comparación con un procesador convencional.

Existe una gran variedad de procesadores, cada uno con diferentes aplicaciones y distintas maneras de llevar acabo el trabajo. Por ejemplo, un procesador convencional, como el de una computadora personal, realiza cálculos con las instrucciones de operación matemática más básicas, como la suma, el control de dispositivos y periféricos. Para este caso, y debido a la necesidad de un procesamiento de señales con operaciones matemáticas frecuentes, se requiere de un procesador que tenga como tarea exclusiva el tratamiento de los datos en tiempo real (con el cual se puedan realizar operaciones complicadas como multiplicación, división, implementación de filtros digitales, etc.).

3.2 DISPOSITIVOS LOGICOS RECONFIGURABLES

Estos dispositivos operan mediante la programación eléctrica de sus celdas básicas e interconexiones. Después de la creación de los PLDs, y gracias a las memorias monolíticas, fueron introducidos los FPLAs (Field Programmable Logic Array) que dieron origen a los primeros PAL (Programmable Array Logia) en 1978. No fue sino hasta 1980 cuando fueron complementados los PLDs para la creación de los CPLDs (*Complex PLD*) con estructuras básicas que consistían en arreglos de PALs. [3][7][8]

En base a las arquitecturas de los dispositivos lógicos programables actuales se consideran las siguientes categorías:

3.2.1. PLDs

Los PLDs son ASICs (Applications Specific Integrated Circuits) configurables o programables por el usuario para realizar una determinada función lógica. Estos dispositivos consisten de una matriz o arreglo de compuertas AND seguida de otra matriz de compuertas OR mediante las cuales se pueden realizar las operaciones lógicas deseadas.

Debido a que los PLDs están situados en una zona intermedia en donde pueden sustituir cientos o miles de CIs (Circuitos Integrados) o realizar operaciones muy básicas para un FPGA, son muy utilizados ya que su precio es bastante bajo en comparación a la realización de una placa PCB (Printed Circuit Board), que contenga todos los CIs necesarios, o un FPGA.

Un ejemplo del funcionamiento del PLD es la suma de señales lógicas, Fig. 3.1, en donde se hace una suma de las señales A, A', B, B', C y C', obteniendo como resultado la señal F; en donde $F = AB' + A'B$.

También existen varios tipos de PLDs dentro de los cuales se pueden nombrar algunos como:

- FPLA
- PAL
- PROM (Programmable Read Only Memory)

- CPLD

Dentro de esta serie de tipos de PLDs varían: el número de compuertas internas, velocidades, conexiones fijas entre compuertas internas, etc.

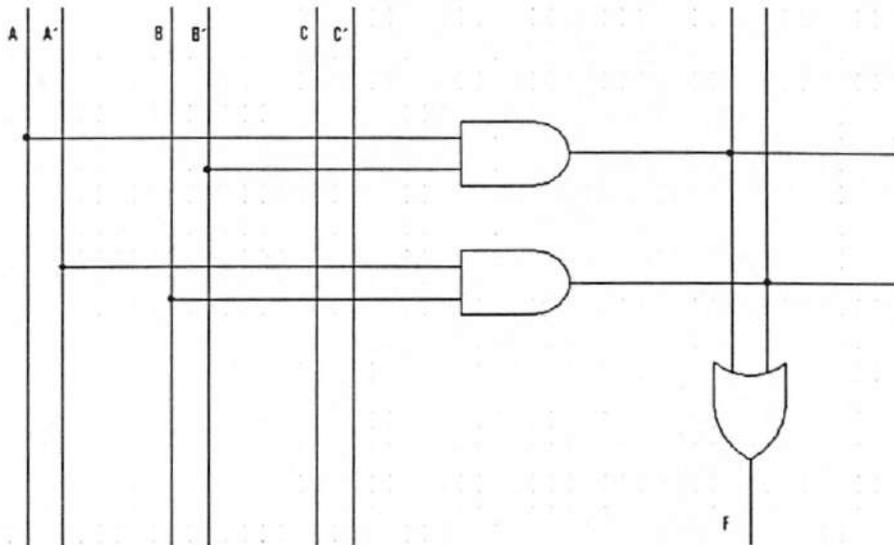


Figura 3.1 Representación de una suma lógica dentro de un PLD

3.2.2. CPLDs

Estos dispositivos son más complejos que los PLDs y ello se debe a que permiten implementar sistemas más completos con un menor espacio de ocupación; tienen bloques lógicos internos como celdas básicas contienen a un PLD.

Debido a que los CPLDs son tan complejos y se asemejan tanto a los FPGA, los CPLDs a veces son considerados dentro de esa categoría. La arquitectura de los CPLDs consiste en:

- **Bloques Lógicos**, similares a un PLD debido a que se componen de compuertas básicas OR y AND, las cuales son utilizadas para realizar operaciones lógicas como suma de productos.
- **Matrices o Multiplexores**, para las interconexiones de los bloques lógicos, ya sean entradas y/o salidas de bloque.
- **Celdas de Entrada/Salida**, se utilizan para conectar los bloques lógicos a las entradas y salidas del CPLD.

3.2.3. FPGAs

Un FPGA es un dispositivo de lógica programable, y como se mencionó anteriormente utiliza la arquitectura de arreglos de compuertas y la forma de programación de los PLDs. Internamente este dispositivo está compuesto de bloques lógicos configurables o CLBs (Cell Logic Basic) parecidos a los PLDs dentro de los CPLDs con la diferencia que la interconexión de estos se realiza mediante un campo programable, el cual rodea a todos los CLBs.

La estructura de los FPGAs consta de 3 componentes importantes (Fig. 3.2), que mediante la programación se interconectan para realizar una función específica:

- **CLB**, llevan a cabo las operaciones lógicas, y están compuestos de flip-flops, multiplexores y uno o más bloques de control que pueden ser un generador de funciones de entrada, control de reloj, reset, etc.
- **IOB (In/Out Block)**, se encargan de interconectar los CLBs con el mundo exterior, es decir, son las entradas y salidas del FPGA. Cada terminal puede recibir o transmitir datos u operar de forma bidireccional. Estos bloques contienen buffers de entrada y salida para tener una mejor compatibilidad con las tecnologías TTL (Transistor-Transistor Logic) y CMOS (Complementary Metal Oxide Semiconductor)
- **Interconexiones Programables**, Fig. 3.3, llevan a cabo la implementación del diseño debido a que conectan entre sí los CLBs y los CLBs con los IOBs. Estas interconexiones están hechas por rejillas de dos caras de segmentos de metal, PIPs (Programmable Interconnect Points) y bloques de switches.

Mediante estos 3 componentes el FPGA lleva a cabo su programación, reservando únicamente los CLBs que se van a utilizar para el diseño, su interconexión entre estos y los IOBs.

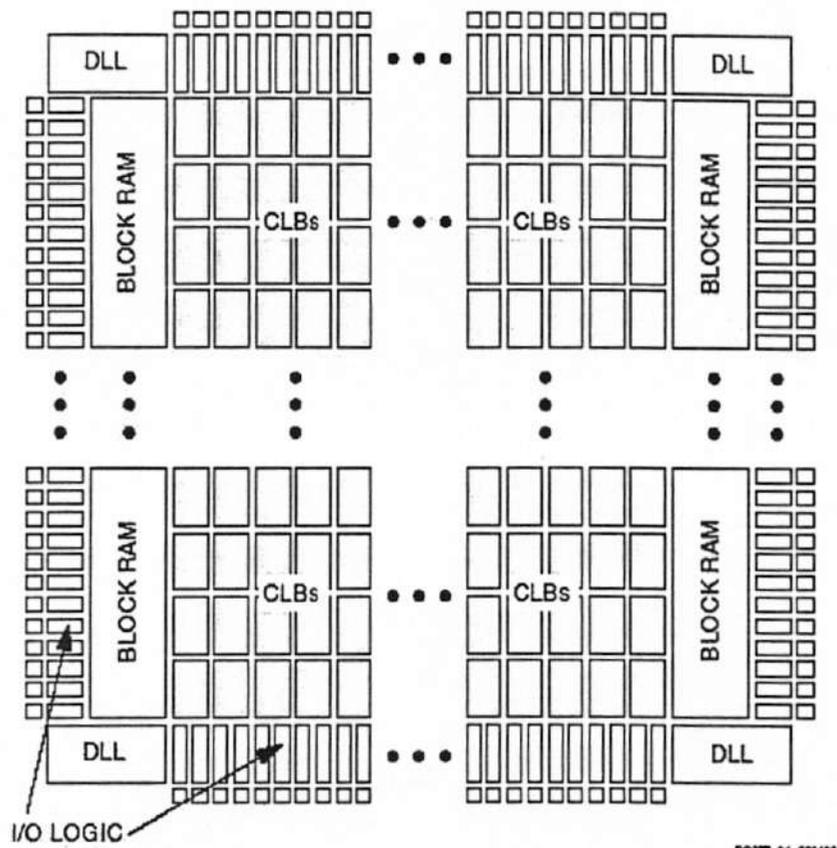


Figura 3.2. Representación interna de un FPGAs

D9077_01_002102

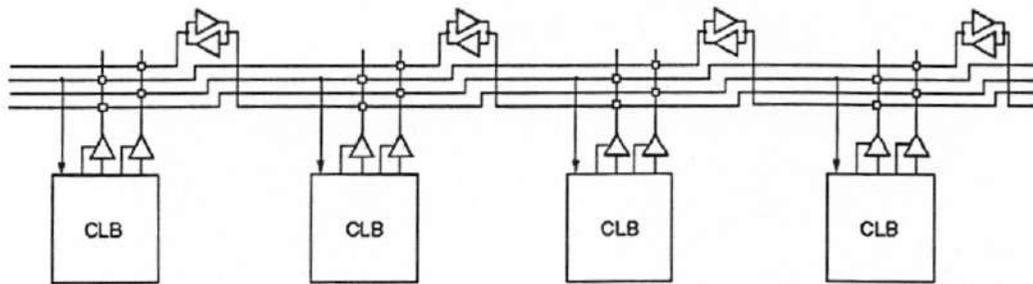


Figura 3.3. Diagrama de interconexiones programables con celdas básicas

El tamaño, número de bloques, estructura y conectividad de las conexiones varía dependiendo de la arquitectura del FPGA. Estas diferencias son manejadas con técnicas de programación y objetivos de aplicaciones de los componentes. [4][5][6].

3.2.3.1. VENTAJAS DE UN FPGA

- Puede ser modificado eléctricamente y estas modificaciones toman milisegundos.
- No se necesitan de otros componentes para realizar diseños, lo cual hace económica su implementación.
- La implementación de los diseños es rápida, debido a que solo se carga la programación por el usuario.
- La revisión de un diseño, se lleva a cabo por simulaciones evitando pérdidas y la manufactura de diseños mal realizados.

3.2.3.2. DESVENTAJAS DE UN FPGA

- La información programada es volátil, es decir, se pierde al interrumpir la alimentación eléctrica.
- La velocidad de conmutación es más lenta que los ASIC (Application Specific Integrated Circuit).
- En comparación al DSP, requieren de tiempo considerable para el desarrollo de un sistema.
- Algunas operaciones necesitan diseños digitales complejos cuando un procesador las realiza con una sola simple ejecución de instrucciones.

3.2.4. FPGA XC2S200E DE XILINX

En este trabajo el FPGA tiene la tarea de generar las señales de control y sincronía de la cámara CCD. Con estas señales se puede tener un control de la duración entre cuadros de imagen (*frames*) y el tiempo de exposición de los píxeles de la cámara, de manera que el DSP deberá leer los datos de la memoria FIFO (First In First Out) que se emplearan en el sistema de adquisición y procesamiento de información obtenida.

Los pasos, según las tareas del FPGA en el presente trabajo, se describen las funciones a desempeñar:

1. Esperar la señal dada por el usuario para el inicio o arranque de todo el sistema.
2. Resetear o borrar los datos de la memoria FIFO.
3. Abrir o exponer los píxeles del CCD de la cámara para que se tome una imagen.
4. Mandar el pulso de inicio de descarga de *frame* a la cámara CCD.

5. Terminar o cerrar la exposición de los píxeles.
6. Volver a esperar otro pulso señal de inicio del usuario.

Otra tarea del FPGA es acondicionar algunas señales empleando un circuito básico de compuertas lógicas requerido para el trabajo del DSP. Para las pruebas de este sistema se utilizó el FPGA XC2S200E de Xilinx Corporation, Fig. 3.4, integrado en la tarjeta D2-SB de Digilent Incorporated. [15] [16] [17].

3.2.4.1. CARACTERÍSTICAS GENERALES DEL XC2S200E EN D2-SB

- FPGA XC2S200E de 200K compuertas.
- 143 Terminales I/O divididos en 3 puertos (A, B y C).
- Socket JTAG para la programación de la memoria flash.
- Oscilador de 50 MHz y un socket para un segundo oscilador.
- JTAG para la programación del FPGA.

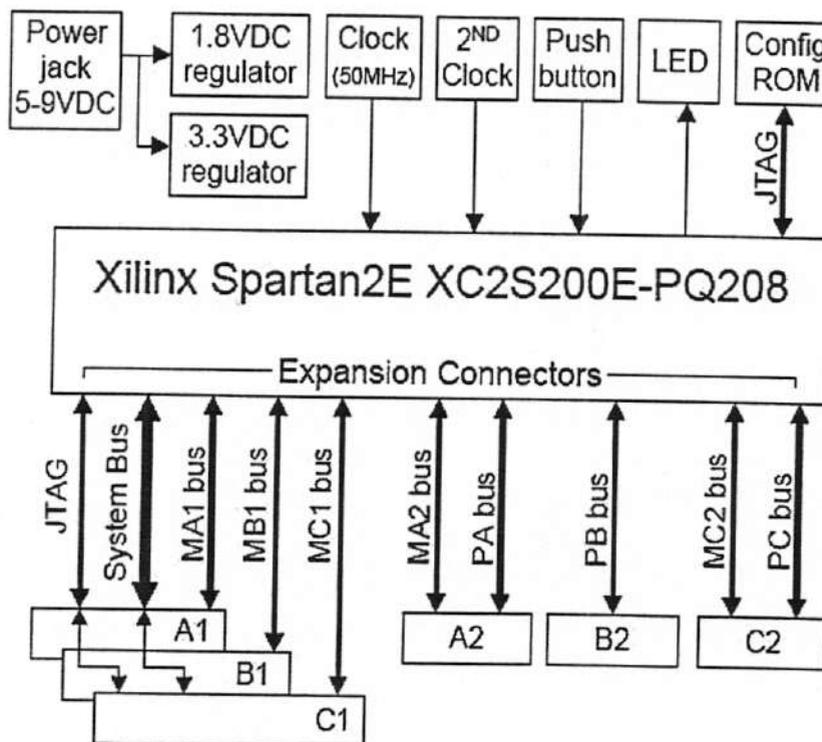


Figura 3.4. Diagrama General del XC2S200E

El chip XC2S200E pertenece a la familia de FPGA *Spartan2E*. En la Tabla 3.1 se muestra a los siete miembros de esta familia de FPGA y muestra la densidad de compuertas de cada uno. Estos chip tienen como características, densidades de compuertas que van desde 50,000 hasta 600,000 las cuales pueden operar a velocidades arriba de 200 MHz.

Dispositivo	Celdas lógicas	Rango de compuertas	Arreglo de CLB	Numero total de CLB	I/O disponibles	I/O de par diferencial	Bits de RAM	Bloques de bits de RAM
XC2S50E	1,728	23,000 – 50,000	16 x 24	384	182	83	24,576	32K
XC2S100E	2,700	37,000 – 100,000	20 x 30	600	202	86	38,400	40K
XC2S150E	3,888	52,000 – 150,000	24 x 36	864	265	114	55,296	48K
XC2S200E	5,292	71,000 – 200,000	28 x 42	1,176	289	120	75,264	56K
XC2S300E	6,912	93,000 – 300,000	32 x 48	1,536	329	120	98,304	64K
XC2S400E	10,800	145,000 – 400,000	40 x 60	2,400	410	172	153,600	160K
XC2S600E	15,552	210,000 – 600,000	48 x 72	3,456	514	205	221,184	288K

Tabla 3.1. Familia de FPGAs *Spartan2E*

Una de las propiedades que tiene la programación de los FPGA es permitir el diseño de actualizaciones sin que haya reemplazos de hardware. Otras características que tiene la familia *Spartan2E* son:

- Están basadas en arquitecturas Virtex-E.
- Reprogramación ilimitada.
- Tecnología Micron 0.15 avanzada.
- Select RAM
 - 16 Bit/LUT de RAM distribuida.
 - Puerto dual de bloque de RAM de 4 KBit configurable.
 - Interfase rápida para RAM externa.
- Varios *registros/latches* con enable, set y reset.
- Cuatro DLLs para el control de reloj.
- 19 High-Performances para estándares de interfase.
 - LVTTTL
 - LVCMOS
 - HSTL
 - SSTL
 - AGP
 - CTT
 - GTL

- LVDS
- LVPECL
- Mapeo, colocación y ruteo; completo y automático

3.2.4.2. CLB del XC2S200E

El CLB es la celda básica del FPGA la cual en el caso del XC2S200E, esta compuesta de un generador de funciones de 4 entradas, circuitos lógicos de acarreo y control, y dispositivos de almacenamiento.

En la Fig. 3.5 se muestra una placa que contiene cada una de las celdas básicas. Los generadores de funciones están implementados por tablas de referencia LUT (*Look-Up Tables*) de cuatro entradas. Estos bloques además de operar como generadores de funciones pueden trabajar como bloques de RAM síncrona de 16 x 1-bit cada LUT. Los dispositivos de almacenamiento pueden ser configurados como flip-flops tipo D.

Los multiplexores MUXF5 de la Fig. 3.6 combinan la salida de los dos generadores de funciones, y el multiplexor MUXF6 combina a los dos *slices* o circuitos que contienen la celda básica.

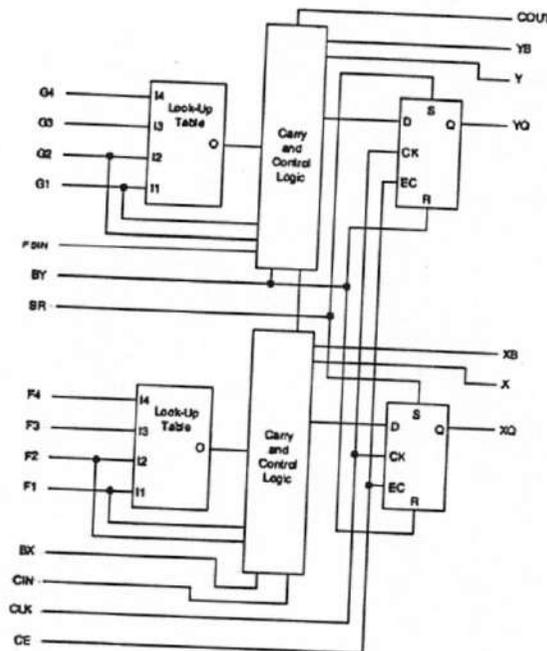


Figura 3.5. Diagrama de una placa de una celda básica

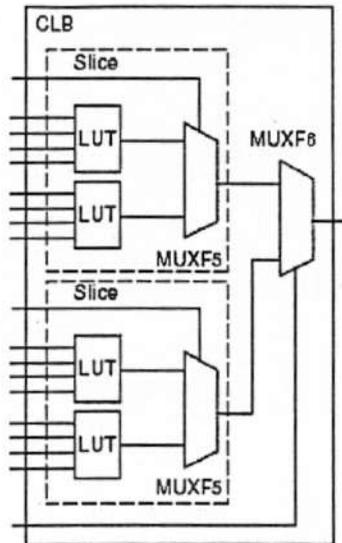


Figura 3.6. Diagrama interno de una celda básica

3.3. PROCESADORES DE SEÑALES DIGITALES

El procesador de señales digitales mejor conocido como DSP es un procesador de propósito específico que realiza secuencias de operaciones de forma más rápida que un procesador convencional. Para entender un poco más lo que es un DSP se tiene el significado de cada una de las siglas.

- *Digital*: Es el proceso de hacer discreta una señal continua para ser representada en forma numérica.
- *Signal*: Información eléctrica que puede ser utilizada para ser procesada por un circuito electrónico.
- *Processor*: Conjunto de operaciones realizadas en una secuencia preprogramada a una serie de datos

Como se menciona anteriormente, se está haciendo uso del DSP en muchos dispositivos electrónicos y se ha convertido en un elemento muy común debido a que ha sustituido las operaciones que realizan otros dispositivos de control o procesamiento electrónico, como microcontroladores o microprocesadores. Algunas implementaciones de los DSPs son:

- Telefonía
- Radares
- Televisiones
- Faxes
- Procesos de control

- Audio
- Filtros
- Procesos industriales
- Dispositivos multimedia

Los microcontroladores o microprocesadores son chips electrónicos que llevan a cabo una tarea dentro de un sistema, y como su nombre lo dice, son para tener un control del mismo o para llevar a cabo un proceso con los datos que maneja. A estos dispositivos se les programa las funciones que realizarán. El DSP puede sustituir a estos dispositivos electrónicos debido a que puede realizar las operaciones de controlador y procesador. Además tiene recursos para realizar un trabajo más eficiente que otros dispositivos de control o procesamiento donde se emplean dispositivos de uso general. Como en el caso de una computadora personal, utilizando un procesador de uso general o no específico como el DSP.

La función de los dispositivos electrónicos como los microprocesadores, microcontroladores, DSPs, etc.; es llevar a cabo desde simples procesos de sumas de datos, operaciones lógicas programando compuertas y demás; hasta programas complejos realizados en lenguaje de alto nivel (como los empleados en las computadoras personales), filtros digitales, controles PID (Proporcional Integral Derivativo) o procesos matemáticos. Para que un procesador lleve a cabo alguna de estas operaciones requiere de información que no está digitalizada o en forma discreta, de ahí la necesidad de digitalizarla por medio de un convertidor Analógico-Digital, después el controlador o procesador se encargará de realizar las operaciones lógicas y/o aritméticas según sea programado.

El DSP, funcionan en base a una unidad de tiempo y ésta es obtenida de un oscilador o reloj, el cual da la base de tiempo al procesador. Los procesadores utilizan este reloj para sincronizar el trabajo, es decir, operan en base a cada ciclo de reloj (por ello una parte de la eficiencia de los procesadores es medida en base a ésta velocidad). Esta unidad de tiempo, marca el ritmo al que se realizará un proceso.

Los procesadores convencionales utilizan la arquitectura Von Neumann la cual emplea un único bloque para la memoria de programa y datos, los DSP utilizan la

arquitectura Harvard en la cual existen dos bloques independientes, uno para cada área, Fig. 3.7.

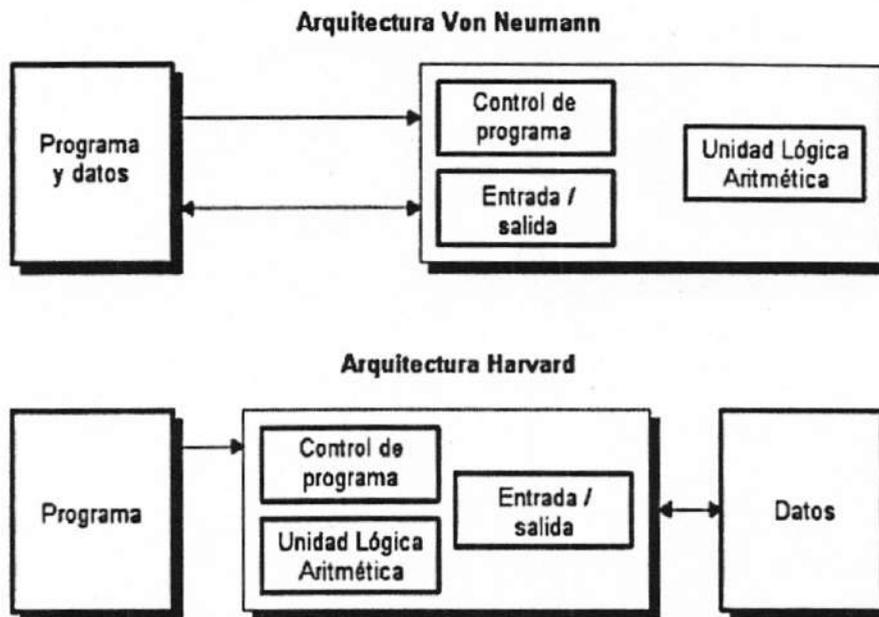


Figura 3.7. Arquitectura de procesadores de computadoras

El procesador convencional realiza una operación cada vez que hay una conmutación de reloj. Es importante conocer ésta forma de trabajo debido a que el DSP funciona de una manera más rápida en cuanto a ciclos de reloj. Los procesos o formas de operación que un procesador convencional realiza son por ejemplo:

- Leer una dirección apuntada por el programa o tarea que se esta realizando.
- Decodificar la instrucción
- Ejecutar la instrucción

En la ejecución de la instrucción se dan los siguientes pasos:

- Leer los datos de la memoria
- Operar con ellos
- Dejarlos en RAM

En la lectura de información mientras el bus esta ocupado leyendo un dato, no puede ser utilizado por otra unidad de decodificación; en el caso del DSP, se utilizan 3 buses, uno para el programa y dos de datos, lo cual permite al DSP leer una instrucción y 2 operandos.

Un microprocesador es un dispositivo de propósito general el cual opera con bloques grandes de software, por tanto, no son comúnmente utilizados en la computación de tiempo real y trabajan mas lentamente, seleccionando una acción, esperando hasta terminar el trabajo y luego pasar a la siguiente instrucción dada por el usuario. Por otro lado el DSP es utilizado como un tipo de controlador “incrustado” o un procesador dentro de otro dispositivo que es empleado solo para realizar un grupo de tareas en particular. En este caso el DSP es un asistente del procesador de propósito general.

El procesamiento de señales es una técnica que convierte la información del mundo real (analógicas) en datos digitales para su análisis posterior, convirtiendo las señales analógicas a números, facilitando el análisis y simplificando el aislar, analizar y modificar dichas señales.

El DSP después de terminar el procesamiento puede convertir la señal digital a analógica con un mejoramiento de la misma, debido a que puede filtrar ruidos, quitar interferencia, amplificar frecuencias y suprimir otras, encriptar la información o analizar ondas muy complejas dentro de cada componente espectral.

Una herramienta muy importante que tiene el DSP es la ejecución de varias operaciones en un solo ciclo de reloj, debido a que está diseñado con circuitos adicionales a los de un procesador convencional, el cual tendría que hacer estas tareas en forma de programa. El ejemplo mas común es la operación MAC (Multiply and Accumulate), esta instrucción como su nombre lo dice efectúa una multiplicación y acumulación, llevadas a cabo en un solo ciclo de reloj, por medio de compuertas lógicas que están conectadas a uno o mas registros del DSP. [14] [18].

3.3.1. CLASIFICACION

Las empresas más importantes que desarrollan DSPs (*Analog Device, Texas Instrument y Motorola*) los tienen divididos en familias, las más importantes se muestran en la Tabla 3.2. Dentro de estas familias existen algunos DSPs basados en arquitecturas específicas como distinción de cada familia.

Analog Device	Texas Instrument	Motorola
Blackfin	C6000	56300
TigerSharc	C5000	56800
Sharc	C2000	56800E
	OMAP	MSC8100

Tabla 3.2. Familias más importantes de DSP

Algunas de las arquitecturas más importantes son la Static Superscalar, Micro Signal Architecture (MSA), RISC, Harvard, VelociTI VLIW (Very Long Instruction Word).

El DSP TMS32C6711 usado en la presente interfaz de adquisición de datos se basa en la arquitectura VLIW. [18]

3.3.2. DSP TMS32C6711 DE TEXAS INSTRUMENT

La tarjeta de Texas Instrument modelo DSKC6711, que contiene un DSP TMS32C6711, se utilizará para el procesamiento de los datos obtenidos a partir de la interface desarrollada en este trabajo, con el propósito de alimentar al DSP con la información adquirida por la CCD.

El TMS320C6711 pertenece a la familia de procesadores de punto flotante TI C6x. Esta basado en arquitectura avanzada de "Palabras-de Instrucciones-Muy- Largas, o VLIW según sus siglas en inglés, y es capaz de ejecutar hasta ocho instrucciones por ciclo. El procesador está diseñado para que la eficiencia de un compilador de C sea tomada en consideración asumiendo que el lenguaje C sea utilizado para la programación del DSP. De allí, es posible que un programa escrito en C pueda correr a

un 80%-90% de velocidad de procesamiento comparado con el escrito en código ensamblador. La Fig. 3.8 ilustra las características principales del procesador.

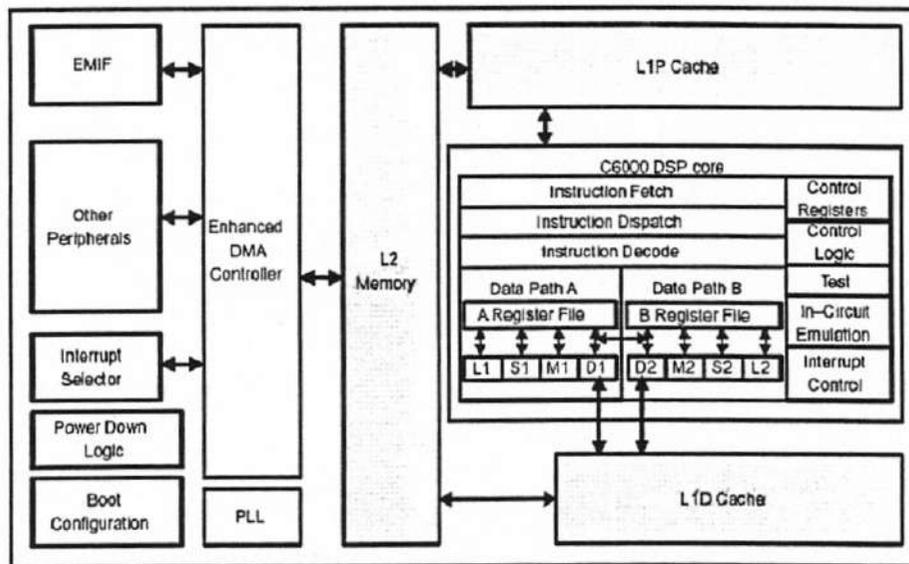


Figura 3.8. Diagrama de bloque del TMS320C671x

El C6711 tiene dos memorias caché internas de 4 kB nivel 1: el caché de programa (L1P) y el caché de datos (L1D); y una interna de 64 kB de nivel 2 (L2) para el almacenamiento de datos/programas. Hay ocho unidades funcionales, las cuales son seis unidades aritméticas lógicas (ALU) y dos unidades multiplicadoras. [14]

3.3.2.1. ARQUITECTURA DEL C6711

- Dos puertos seriales multicanales con buffer (McBSPs), los cuales pueden manejar hasta 128 canales y proveen una interfaz directa para los dispositivos externos de los estándares de la industria,
- Una Memoria de Acceso Directo (DMA),
- Dos contadores de 32 bits de propósito general, los cuales pueden ser usados para medir el tiempo y contar eventos, para interrumpir el CPU, o mandar eventos de sincronización a la DMA,
- Un puerto de interfaz de 16 bits (HPI), el cual es un puerto paralelo para comunicarse con una PC,
- Una interface de memoria externa de 32 bits (EMIF), la cual soporta una interface independiente a una dirección de programa, programa de datos de 256 bits, dos direcciones de datos de 32 bits, dos direcciones de datos de 64 bits, dos datos de 64 bits, y dos datos de almacenamiento de 54 bits.

Las ocho unidades funcionales están divididas en dos rutas de datos, A y B, como se muestra en la Tabla 3.3. La unidad (.L) es para las operaciones aritméticas y lógicas. La unidad (.S) es para las brechas, la manipulación de bits y las operaciones aritméticas. La unidad (.M) es para las operaciones lógicas y aritméticas. La Unidad (.D) es para la carga y almacenamiento y operaciones aritméticas. Estas cuatro unidades están en ambas rutas de datos A y B. Las unidades (.L) y (.S) son ALUs de punto flotante/fijo y las unidades (.D) son ALUs de punto fijo. Las unidades (.M) son multiplicadores de punto flotante/fijo.

Tiempo de Instrucción de ciclo	6.67 ns (Velocidad de reloj: 150 MHz)
Máximo de operaciones	900 MFLOPS (Millions of Floating Point Operations per Second), con 6 unidades funcionales
Máximo de instrucciones	1,200 MIPS (Millions Instruction per Second), 8 instrucciones/ciclo
Soporte de datos	8 / 16 / 32 bits, operaciones de punto fijo. 32 / 64 bits, operaciones de punto flotante.
Espacio de direcciones	4 GB
Memoria Interna	Nivel 1 cache de programa 4kB Nivel 1 cache de datos 4kB Nivel 2 cache de RAM 64 kB
Registro de propósito general	2 sistemas de 32 bits x 16
Unidades funcionales	ALU Unidad "multipler"
Voltaje de alimentación	3.3 V (I/O), 1.8 V (interno)

Tabla 3.3. Características principales del TMS320C6711

Se describe el uso del DSP en el capítulo 4 pese a no ser el objetivo de ésta tesis, más por ser requerido para entender la arquitectura que emplea la memoria externa EMIF, que a su vez obtiene los datos del CCD. [14]

3.4. CONCLUSIONES

Los dispositivos programables son utilizados para optimizar el desarrollo del trabajo de adquisición de señales.

El FPGA lleva a cabo el control del sistema, es decir, provee las señales de control para el CCD y escritura sobre el DSP, obteniendo un ahorro de operaciones, que un

procesador convencional utilizaría para controlar todo esto y llevar a cabo los procesos de perfilometría.

Por otro lado también se tiene que el desempeño del DSP optimiza el trabajo de perfilometría gracias a ser un procesador de uso específico.

En el capítulo 3.3 se menciona que el uso del DSP en este trabajo solo se emplea en la interconexión con el sistema de adquisición de datos. Éste enlace se hace directamente con la memoria FIFO (First In First Out) por medio de la tarjeta de PCB. Solo cabe mencionar que para la lectura de la información almacenada en dicha memoria no hace falta un direccionamiento como en la memoria convencional, solamente un pin por el cual se tenga el reloj de lectura proporcionado por el DSP y 8 pines adicionales de datos para la transferencia de la información.

Capítulo 4. SISTEMA DE ADQUISICIÓN DE DATOS

El sistema de adquisición de datos está formado por procesos que llevan a cabo el proceso de adquisición de imágenes empleado en el trabajo de perfilometría, cuya función es tomar la fotografía de una figura, haciendo incidir en ella un patrón de franjas de luz, cuyos datos serán procesados para la reconstrucción de la misma en tres dimensiones.

La cámara CCD emplea un protocolo de comunicación EIA-644 (Electronic Industries Alliance) mejor conocido como LVDS (Low Voltage Differential Signaling). Para establecer la comunicación entre la CCD con el resto del sistema, y viceversa, se utilizan convertidores LVDS-TTL.

Para controlar el envío de imágenes o *frames* de la cámara se utiliza un FPGA. Las señales de información enviadas desde el CCD son almacenadas en una memoria FIFO, empleada por el DSP como memoria externa, y es utilizada para que no haya pérdida de datos. La memoria FIFO además de recibir datos desde la cámara también recibe señales del DSP para obtener la información de la imagen que será procesada.

4.1. DIAGRAMA GENERAL

El propósito del sistema desarrollado es interconectar todos los dispositivos utilizados (CCD, LVDS, FPGA, FIFO y DSP) para hacer llegar la información de la imagen al DSP y posteriormente ser procesada.

En la Fig. 4.1 se visualiza la interconexión de todos los dispositivos del sistema mediante una placa de circuitos impresos o PCB (Printed Circuit Board) en la cual internamente se encuentran la memoria FIFO y los convertidores LVDS-TTL.

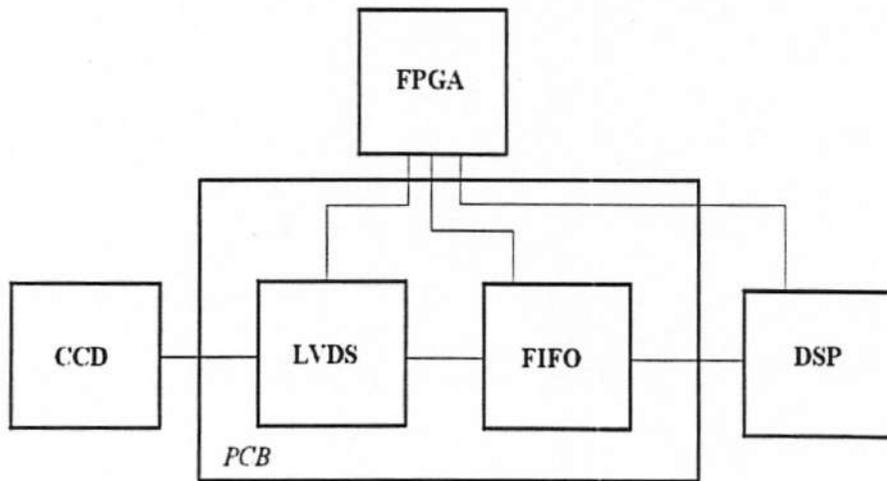


Figura 4.1. Diagrama general del Sistema de Adquisición

El funcionamiento general del sistema es:

- Inicia con una señal enviada por el FPGA a la memoria FIFO para su borrado, seguida de otra señal a la cámara para comenzar el envío de *frames*,
- Se transforman las señales enviadas y recibidas por la cámara mediante los convertidores LVDS-TTL,
- El CCD envía los datos de la imagen junto con las señales de control hacia la memoria FIFO,
- El DSP lee la información almacenada en la memoria FIFO.

4.2. PROTOCOLO DE COMUNICACIÓN EIA-644 (LVDS)

Se emplea en el sistema un convertidor de LVDS a TTL (Transistor-Transistor Logic) debido a que la cámara de CCD utiliza el protocolo de comunicación estándar EIA-644-1995 de la Electronic Industries Alliance.

En la comunicación por medio de transmisores y receptores LVDS se utilizan dos pins de información diferencial, es decir, cuando el transmisor recibe una señal enviará su mismo estado lógico por el pin positivo (+) y el estado inverso por el pin negativo (-). En la Fig. 4.2 se ve un arreglo de un transmisor-receptor de LVDS.

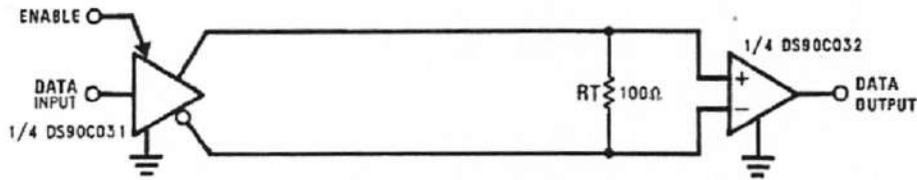


Figura 4.2. Diagrama de un circuito de comunicación LVDS

En la comunicación mediante el protocolo LVDS se utiliza una resistencia terminal de 100 ohms, la cual es empleada como una impedancia de entrada para los cambios de flujo de corriente que pasan a través de ella.

La utilización de un par de terminales de comunicación, pese a la desventaja de emplear más hilos, provee de una señal diferencial, la cual ayuda a reducir el ruido de EMI (Interferencia Electromagnética). En la Fig. 4.3 se puede ver un ejemplo, cuando el ruido afecta a un pin éste también afecta al otro, debido a que físicamente están entrelazados; debido a que el receptor no mide la intensidad de corriente que le llega a cada pin de entrada, sino la diferencia entre éstos dos, estos picos de ruido son eliminados. Al suprimir la interferencia de EMI en las señales que llegan al receptor, las transferencias pueden ser de hasta 1.5 Gb/segundo sin aumento sustancial en la disipación de energía.

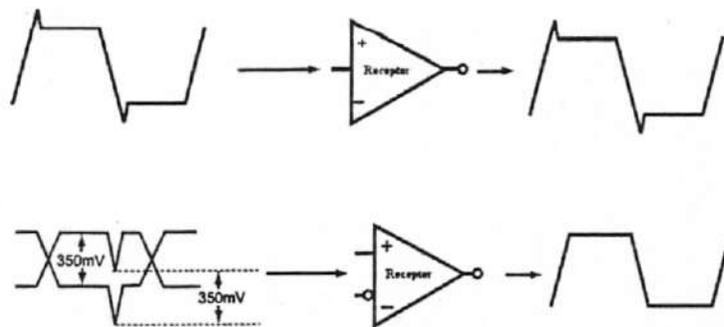


Figura 4.3. EMI en líneas de transmisión.

La ventaja en el consumo de potencia se debe a que el protocolo EIA-644 consume muy poca corriente, del orden de 3.3 mA con una diferencia de 350 mV. Cuando las señales atraviesan el resistor terminal de 100 Ω , se tiene un consumo de potencia de 1.1 mW. En comparación al protocolo GTL, que consume 40 mA con 1 V a través de la resistencia de carga y una disipación de energía de 40 mW, el protocolo

LVDS efectivamente resulta ser de bajo consumo de potencia. La Fig. 4.4 muestra una comparación de frecuencia y corriente del modo de operación de los LVDS GTL y TTL/CMOS. [13]

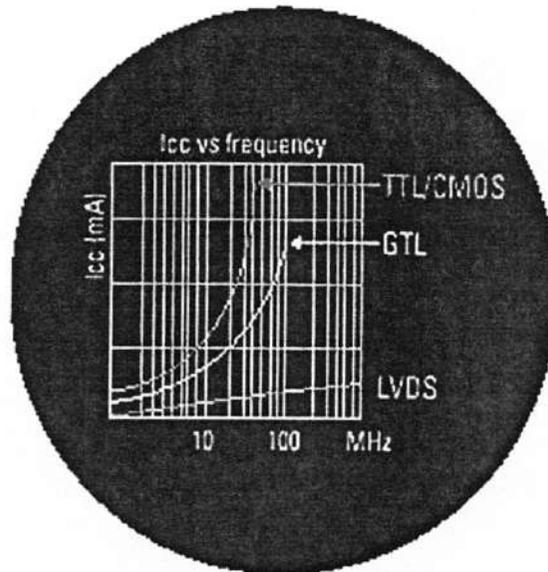


Figura 4.4. Grafica de frecuencia contra corriente de protocolos.

4.3. MEMORIA FIFO

En el sistema de adquisición de datos se utiliza una memoria FIFO como un medio de almacenamiento para evitar la pérdida de estos.

Una memoria esta compuesta por registros para almacenar información digital, para entender la memoria FIFO se tiene como base el funcionamiento de dichos registros compuestos por bloques más básicos, FLIPS-FLOPS, formados a su vez por compuertas lógicas. Un registro, entonces, es un grupo de celdas binarias, y cada una de estas almacena un bit de información.

La base de la memoria son los registros a los cuales se tiene acceso en modo de lectura o escritura. En las memorias convencionales este acceso se logra por medio de un decodificador; por ejemplo, si se hace lectura a una memoria de 1152 bits, dividida en 128 registros de 9 bits cada uno, se requieren 7 líneas de entrada, para acceder a las localidades de memoria, y 9 líneas de salida. La memoria FIFO (First In First Out)

almacena la información de manera que el primer dato almacenado será el primero en ser leído, el segundo dato almacenado será el segundo en leerse y así sucesivamente, de ahí el nombre “primero en entrar, primero en salir”.

En la Fig. 4.4.1 se muestra en forma general el funcionamiento de la memoria FIFO, la cual cuenta con sus señales de habilitación para lectura (RDCLK, RDEN1 y RDEN2) y escritura (WRTCLK, WRTEN1/DP9 y WRTEN2), los cuales habilitan las operaciones de trabajo en la memoria. También se tiene el bus de datos de la señal de 8 bits con la información a almacenar en la FIFO (D0 - D8) y el bus de datos de salida de la memoria (Q0 - Q8). Para terminar se tiene el bloque de estado la cual recibe las señales de PEN (para habilitar la programación de la memoria) y la de RESET para el borrado de la misma); y así establecer un orden por medio del cual haga uso de las banderas.

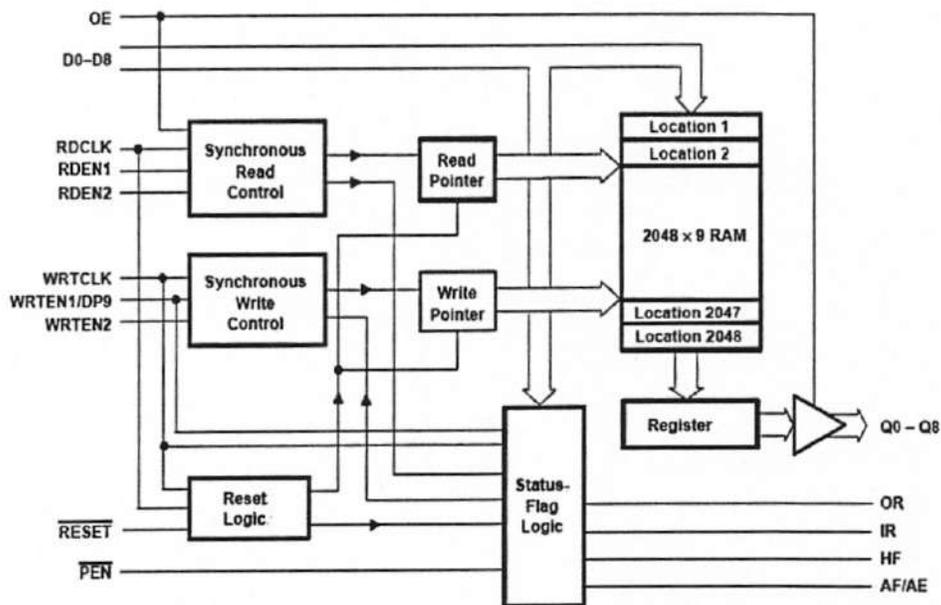


Figura 4.4.1 Bloques de funcionamiento de la memoria FIFO.

La memoria empleada en el sistema de adquisición de datos es una SN74ACT7807 de Texas Instrument, con una capacidad de 2048 x 9 bits. Como solo se utilizan 8 bits, el espacio de memoria empleado es de 2048 bytes, con una velocidad máxima de operación de 67 MHz.

La memoria tiene 3 banderas de almacenamiento:

- Para alertar antes del llenado o vaciado en su totalidad, denominada AF/AE (Almost Full/Almost Empty), y puede ser programada para modificar su capacidad de advertencia de llenado de la memoria (la terminal utilizada para programar la memoria es el PEN – Program Enable),
- Para saber que se encuentra a media capacidad, conocida como HF (Half Full), y
- Dos más para dar a conocer la disponibilidad de lectura OR (Output Ready) ó escritura IR (Input Ready).

También cuenta con una terminal de RESET utilizado para borrar toda la información no deseada o basura dentro de la memoria. Además cuenta con señales de habilitación como el RDEN1, RDEN2, WRTEN1 y el WRTEN2, que habilitan la lectura y escritura en la memoria FIFO. [11]

4.5. RECEPCION DE LA IMAGEN A LA MEMORIA (CCD-FIFO)

Una vez que se ha explicado el funcionamiento de la cámara CCD y de la memoria FIFO se hace una conexión entre los datos de salida de la cámara (ya convertidos a TTL) y datos de entrada de la memoria. La Fig. 4.5 muestra los terminales de conexión de la cámara.

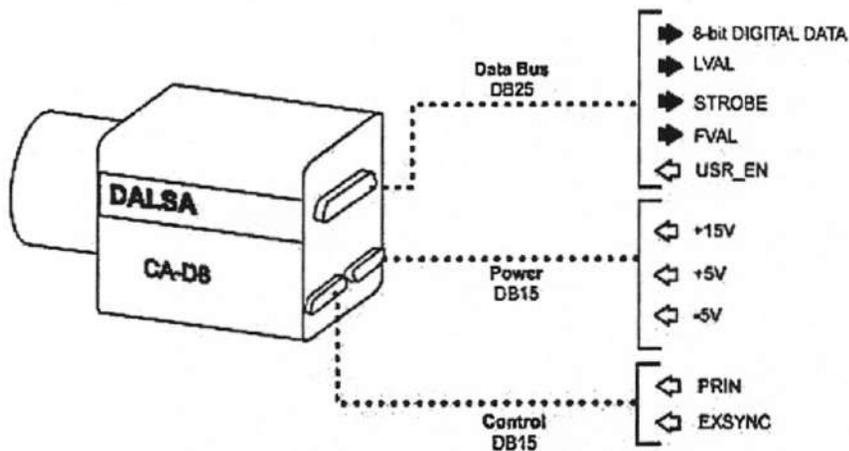


Figura 4.5. Terminales de conexión de cámara CA-D8 de DALSA.

Se conecta la señal de STROBE de la cámara con la señal WRTCLK de la FIFO para que proporcione la frecuencia en la cual se van a escribir los datos de los píxeles. En la Fig. 4.6 se visualizan los pines de conexión de la memoria.

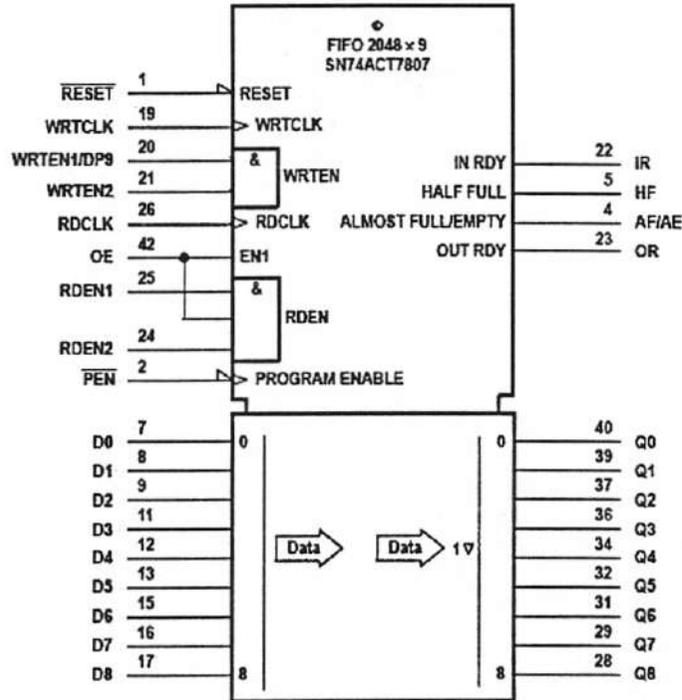


Figura 4.6. Terminales de conexión de memoria FIFO.

Otra señal necesaria de la memoria es la habilitación de escritura (WRTEH1 y WRTEH2) proporcionada por la cámara CCD. La señal LVAL habilita la escritura debido a un envío de datos.

El FPGA debe transmitir una señal de RESET de la memoria FIFO para limpiarla seguido de otra señal de EXSYNC hacia la cámara para que ésta comience el envío de *frames*.

4.6. TRANSFERENCIA DE IMAGEN AL DSP (FIFO-DSP)

El DSP accede a la memoria FIFO a través de un EMIF. Para hacer uso de un dispositivo externo el EMIF tiene señales de entrada, salida, banderas y control. En la Fig. 4.7 se muestran los terminales de la EMIF del C6711 y se describen en la Tabla 4.1.

TMS320C6211/C6711 External Memory Interface

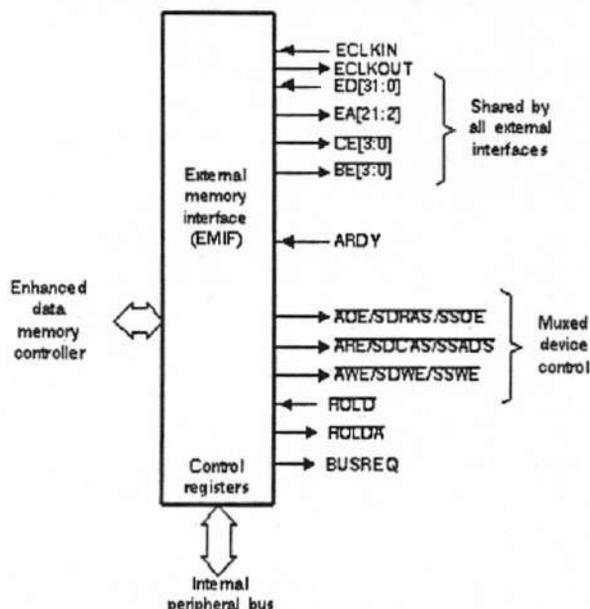


Figura 4.7. Terminales de conexión del EMIF

PIN	I/O/Z	DESCRIPCIÓN
CLKOUT1	O	Reloj de salida. Corre a la misma velocidad que el procesador.
CLKOUT2	O	Reloj de salida. Corre a la mitad de velocidad del procesador.
BUSREQ	O	Se activa en alto la señal de solicitud del bus.
ECLKOUT	O	Reloj de salida de la EMIF. Todos los pines de I/O están gobernados por el reloj de la EMIF
ED[31:0]	I/O/Z	Datos de entrada y salida (I/O). 32 bits de datos de I/O de memorias y periféricos externos.
EA[21:2]	O/Z	Salida de direccionamiento externo. Maneja bits de 21-2 de direcciones de byte.
CE0	O/Z	Se activa en bajo la selección de espacio de memoria CE0
CE1	O/Z	Se activa en bajo la selección de espacio de memoria CE1
CE2	O/Z	Se activa en bajo la selección de espacio de memoria CE2
CE3	O/Z	Se activa en bajo la selección de espacio de memoria CE3
BE[3:0]	O/Z	Se activa en bajo la habilitación de byte. Byte individuales y medias palabras (halfwords) pueden ser seleccionadas por lectura y escritura al mismo tiempo.
ARDY	I	Listo (Ready). Activado en bajo la bandera de ready para las entradas es utilizada para hacer estados de pausas para memoria y periféricos lentos.
HOLD	I	Se activa en bajo, la solicitud de espera (tres estados) del bus externo.
HOLDA	O	Se activa en bajo, la espera del reconocimiento del bus externo.

Tabla 4.1. Terminales de Conexión del EMIF

En este listado se basa la conexión del EMIF del DSP con la memoria FIFO. Se utilizarán 10 pines de los 32 bits ED [31:0] del EMIF, 8 pines para los datos de salida de la FIFO y 2 más para las banderas (HF y AF/AE); y un pin adicional (ECLKOUT) para la frecuencia de lectura de la memoria (RDCLK). [Apéndice B] [14]

4.7. PCB

Para llevar a cabo la interconexión de todos los dispositivos, se diseñó una tarjeta de circuito impreso (PCB) conteniendo la memoria FIFO, los convertidores LVDS, y tres puertos para la conexión de: la cámara CCD, el FPGA y el DSP.

Una PCB es la base física donde se crean las conexiones entre los componentes dentro de un sistema, esta hecha de un material aislante y sobre éste impresas pistas, por lo regular de cobre, las cuales conectan varios dispositivos electrónicos, de ahí el nombre Tarjeta de Circuito Impreso. Existen varios métodos para la elaboración de estas tarjetas, uno de los mas viables es la utilización de impresoras de tarjetas PCB; otro es utilizando material de presensibilizado; o bien, de manera rústica empleando serigrafía.

Para la creación de esta tarjeta se utilizó una placa cubierta de cobre y sobre ésta imprimieron las líneas de conexión, utilizando algún tipo de tinta, protegiendo el cobre debajo de ellas. La placa ya marcada, se sumergió en una solución de cloruro ferrico para remover el cobre no protegido, quedando entonces solamente las pistas de conexión.

Para los diseños esquemáticos de la placa de PCB se empleó el programa Capture CIS del paquete ORCAD versión 9.1. Para los diseños de las caras de la placa se utilizó el programa Layout Plus, incluido en el mismo paquete. El circuito esquemático y los *layout* o caras de la placa de PCB se muestran en el [Apéndice C].

Una vez teniendo hecha la placa de PCB, se pudieron realizar pruebas para detectar falsos contactos, cortos circuitos y/o fallas de diseño en las conexiones. Además se probó cada dispositivo de manera independiente. Con ellos se validó el acoplamiento entre ellos.

Capítulo 5. PRUEBAS ELECTRICAS Y RESULTADOS

En este capítulo se muestra el funcionamiento del sistema de adquisición mediante la realización pruebas a cada dispositivo por separado, es decir, se realizaron mediciones y pruebas eléctricas a la cámara CCD, convertidores LVDS-TTL y TTL-LVDS, FPGA, y FIFO. Cabe recalcar que en esta tesis no se presentan pruebas al sistema general que incluye al DSP, pero se muestra evidencia de que el sistema de adquisición de datos solo requiere de unas pocas señales del DSP para la lectura de datos y posterior procesamiento.

La primer parte de este capítulo muestra el funcionamiento de los convertidores LVDS-TTL. Después de esto se presentan los resultados de las pruebas de funcionamiento de la cámara. En este caso se analiza el envío de los píxeles y las banderas LVAL y FVAL, empleadas en la sincronía del sistema

La función del FPGA en el sistema es la de primero enviar una señal de RESET para el borrado de la información o basura de datos de la memoria FIFO. Además, se envía también un pulso de inicio a la cámara CCD para que comience la transferencia de un *frame* a la memoria.

Por ultimo se prueba la memoria FIFO, en donde el proceso de escritura y lectura de datos son analizados de manera independiente.

Para llevar a cabo las pruebas se utilizo el FPGA como generador de señales y dispositivos de medición como:

- Osciloscopio Tektronix TDS2012 de dos canales (100MHz / 1 GS/s).
- Modulo de almacenamiento de memoria y comunicación Tektronix TDS2MEM y su software (tarjeta de expansión del osciloscopio).
- Multímetro Fluye 179.

5.1. PRUEBAS A LOS CONVERTIDORES LVDS-TTL Y TTL-LVDS

Para el desarrollo de éste trabajo se utilizaron los componentes DS90C031 y DS90C032 en el sistema de conversión de TTL a LVDS y LVDS a TTL respectivamente. Las pruebas previas al empleo de estos dispositivos se llevaron a cabo mediante la implementación del circuito mostrado en la Fig. 5.1. El funcionamiento de este circuito, como un esquema de transmisor-receptor, es el siguiente: Dada una señal con niveles TTL se aplica al circuito DS90C031 para ser transformada a un formato diferencial LVDS. A su vez, esta señal es enviada a los pines de entrada del circuito DS90C032 para convertirla a una señal con niveles TTL.

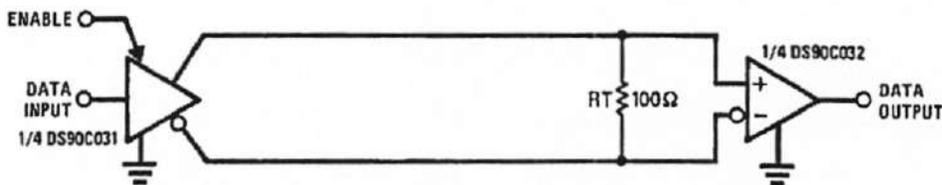


Figura 5.1. Sistema de conversión TTL-LVDS- TTL (DS90C031- DS90C032)

La primera parte de la prueba consistió en aplicar una señal TTL de estado alto en la entrada del DS90C031 y se realizó una medición en los pins de salida del mismo con una señal resultante como se muestra en la Fig. 5.2. La salida positiva del convertidor TTL a LVDS (salida de voltaje diferencial) está en estado alto y la salida negativa se encuentra en estado bajo. Al realizar un cambio de estado de alto a bajo en la entrada del circuito DS90C031 las señales de salida del mismo también son invertidas, es decir ahora el pin positivo de salida diferencial da una señal en estado bajo y el negativo en estado alto como se ve en la Fig. 5.3.

Después, se empleó un generador de señales, con una señal cuadrada con niveles TTL aplicada en la entrada del transmisor de LVDS (DS90C031), para apreciar los cambios de estados de la señal diferenciada obtenida a partir de los pins de salida del DS90C031 como se ve en la Fig. 5.4. Mientras el estado de la señal de un pin se encuentra en el flanco de bajada el estado del otro se encuentra en el flanco de subida y viceversa.

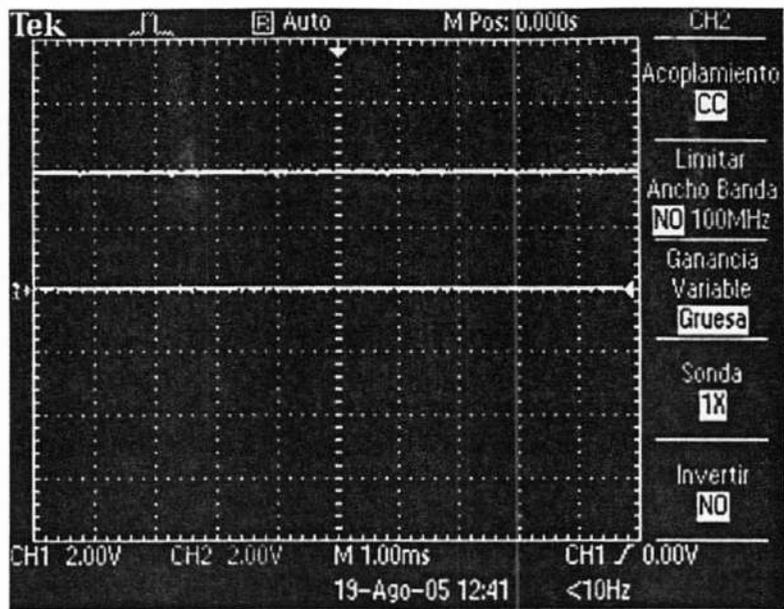


Figura 5.2. Salida positiva del LVDS en alto y negativa en bajo.

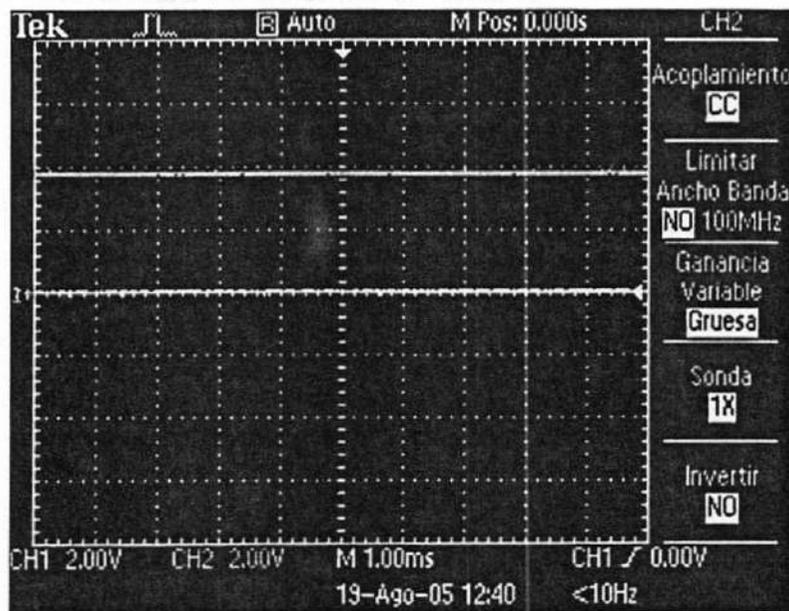


Figura 5.3. Salida positiva del LVDS en bajo y negativa en alto.

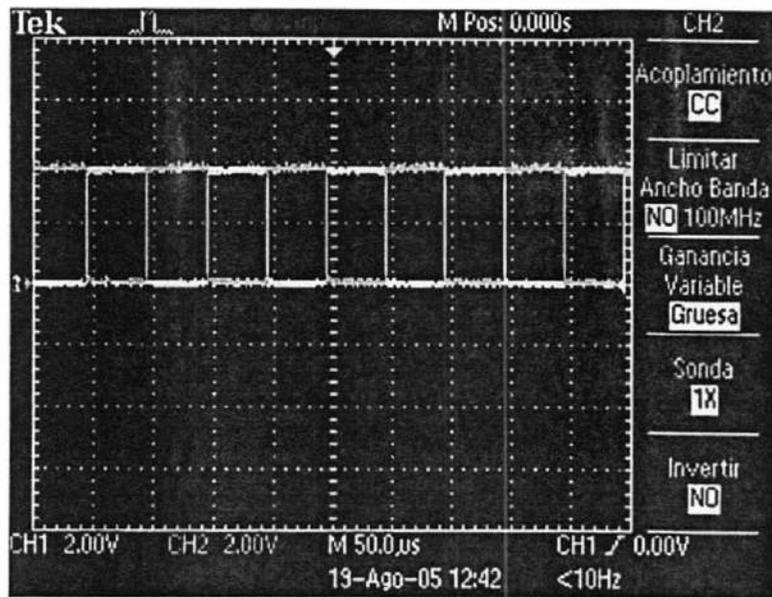


Figura 5.4. Diferencial de señales cuadradas emitidas por un generador de señales.

Para ejemplificar mejor esta prueba, en la Fig. 5.5 se aprecian la misma señal diferencial de la Fig. 5.4 pero ahora por separado.

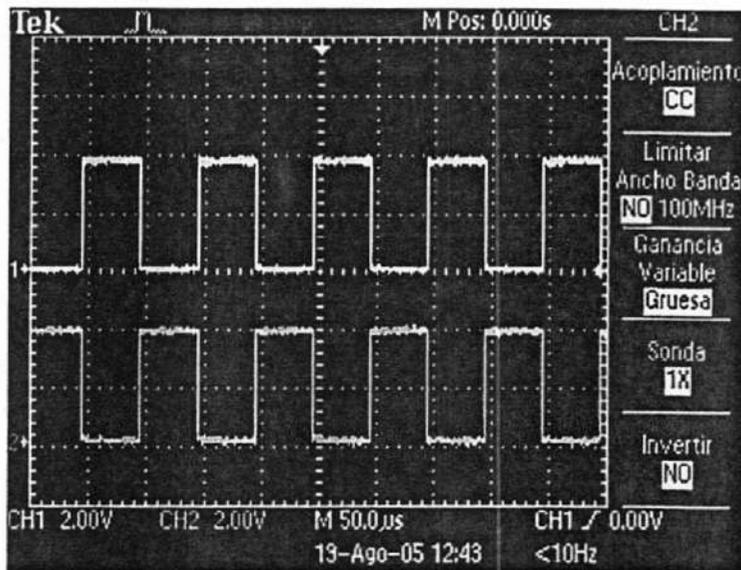


Figura 5.5. Diferencial de señales cuadradas por separado.

Otro resultado de estas pruebas se aprecia en la Fig. 5.6, donde se puede ver que al realizar un cambio en la entrada del transmisor de LVDS (DS90C031) ocurre un cambio de igual magnitud (niveles TTL) de estado en la salida del receptor (DS90C032).

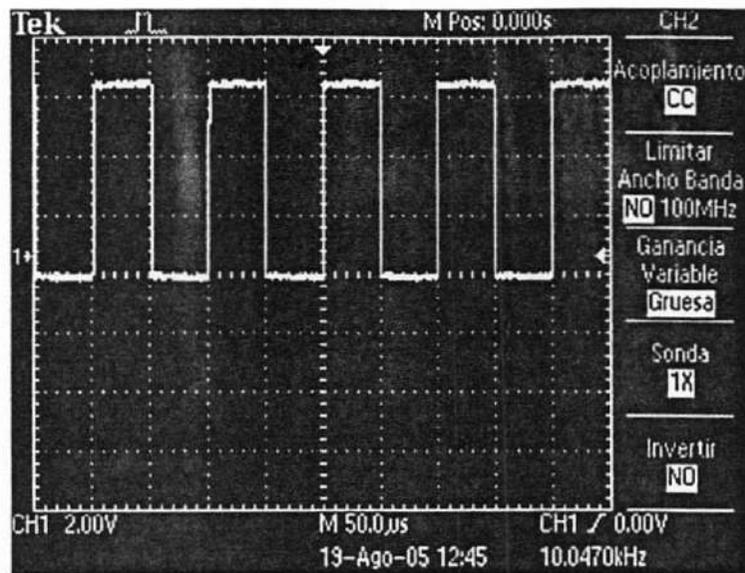


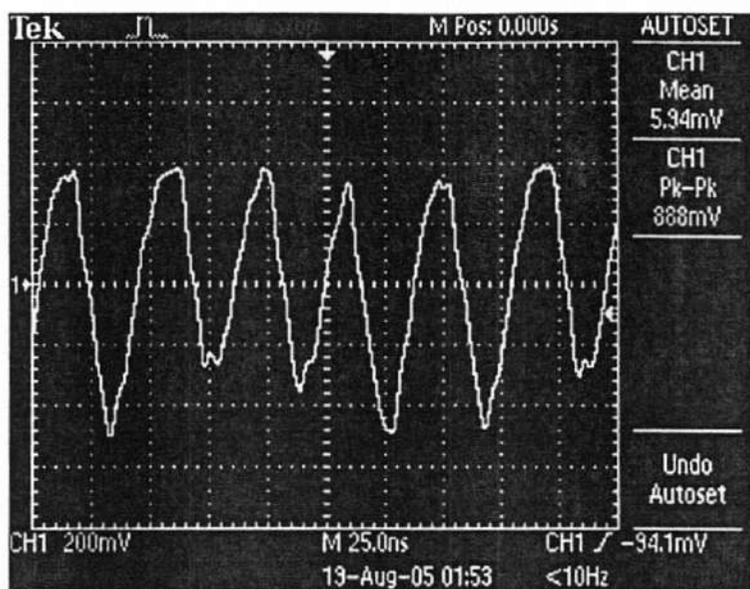
Figura 5.6. Recepción por un LVDS de una señal cuadrada.

5.2 SEÑALES A PARTIR DE LA CAMARA CCD

Las pruebas en la cámara de CCD se realizaron a manera de analizar la transferencia de información hacia el sistema; los datos enviados y recibidos fueron previamente convertidos a formato TTL mediante los convertidores LVDS a TTL.

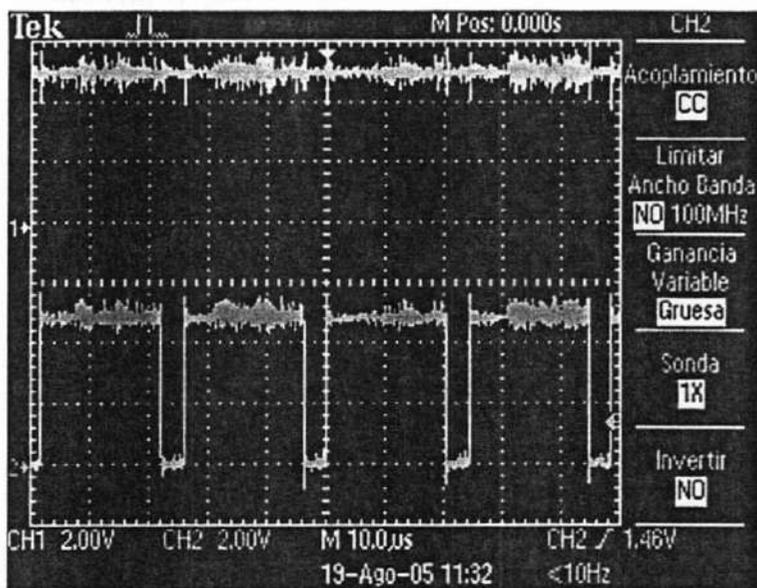
En las primeras pruebas la cámara se puso en modo de funcionamiento libre (*free run*) para que ésta operara sin la necesidad de un control dado por un dispositivo externo. Las señales de la cámara son de tres tipos: control, banderas y datos.

Las banderas son las señales LVAL (*Line Valid*), FVAL (*Frame Valid*), y STROBE (Reloj). La señal de reloj dada por la cámara es la que informa al sistema la velocidad de operación. En la Fig. 5.7 se puede apreciar la señal STROBE de 25MHz a partir de la cámara, como se puede comprobar en el manual de tal dispositivo.



5.7. Señal de STROBE del CCD.

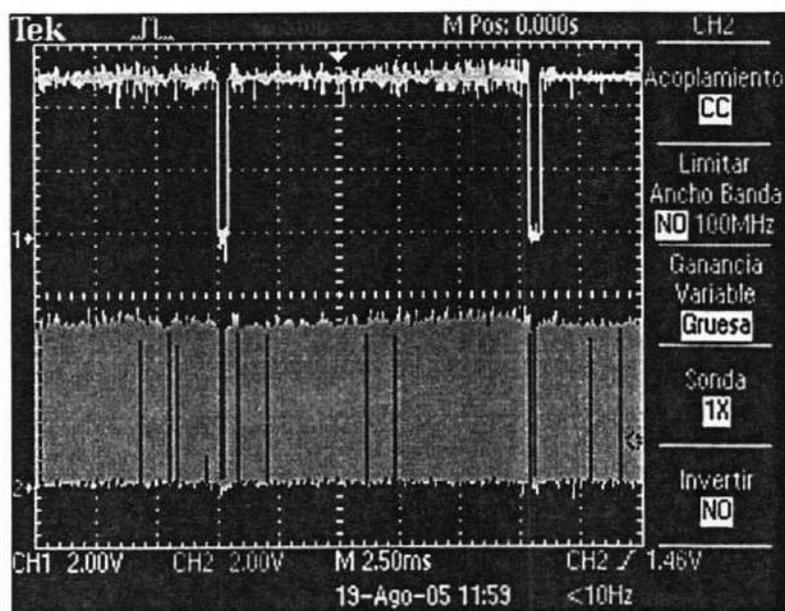
Las señales LVAL y FVAL funcionan de la siguiente manera, FVAL se habilita en alto para iniciar la transferencia de un *frame* y durante ésta, LVAL realiza 512 transiciones de estado lógico, una por cada línea enviada. La señal inferior de la Fig. 5.8 muestra la señal LVAL mientras la señal superior, FVAL, está en alto.



5.8. Señales de FVAL (Superior) y LVAL (Inferior) del CCD.

Al variar la frecuencia de medición, o escala de tiempo en el osciloscopio, se observa que las transiciones de LVAL se van juntando hasta el punto en que sus

separaciones no se distinguen. Sin embargo ahora (véase Fig. 5.9) se pueden ver las transiciones de FVAL, es decir, un cambio de estado lógico de FVAL por cada 512 de LVAL.



5.9. Señales de FVAL (Superior) y LVAL (Inferior) del CCD.

Las señales de datos de la cámara capturada por los dispositivos sensores de carga CCD se envían a la memoria FIFO para que ésta almacene la información de la imagen obtenida y la deje preparada para que el DSP las lea. Las señales de datos de la cámara D1 y D2 se muestran en las Fig. 5.10, D3 y D4 en la Fig. 5.11 y D7 y D8 en la Fig. 5.12. Las señales de alta frecuencia, como las señales de datos o la señal de STROBE, operan a 25 MHz, por ello se ven afectadas por el ruido, visualizado en las figuras o gráficas antes mostradas.

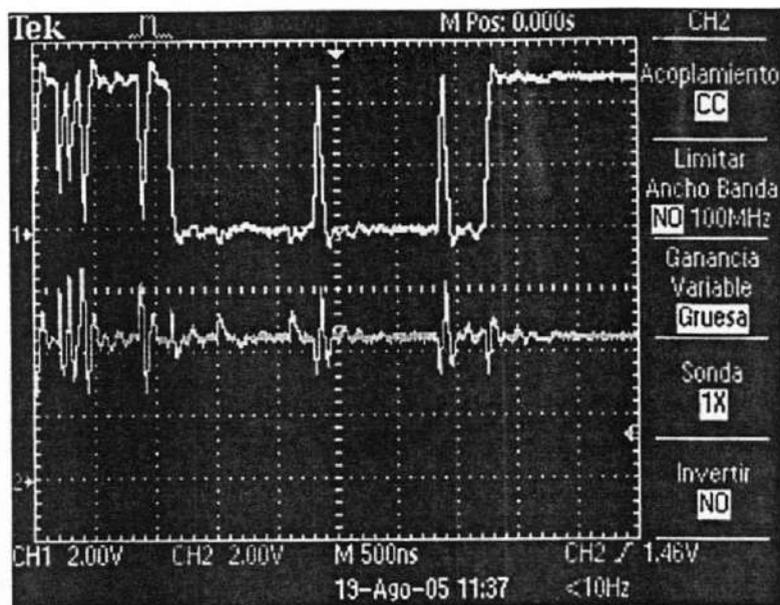


Figura 5.10. Señales de Datos D1 y D2 del CCD.

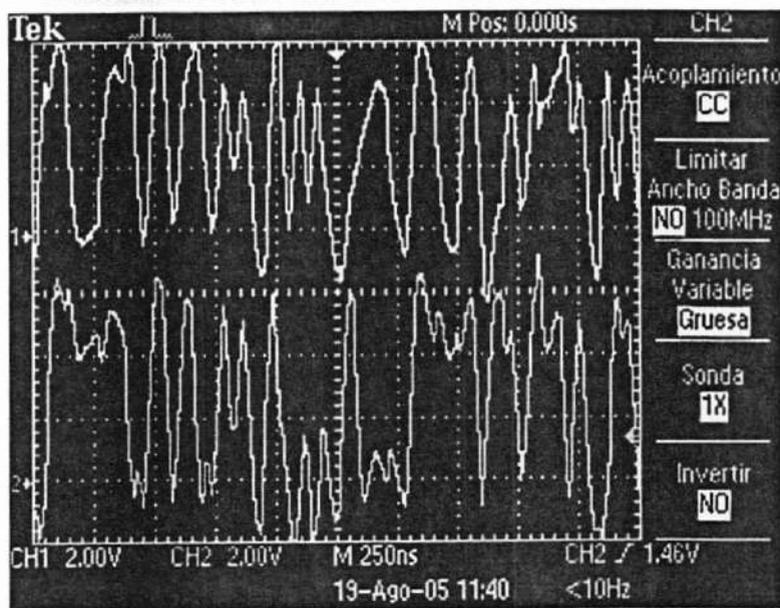


Figura 5.11. Señales de Datos D3 y D4 del CCD.

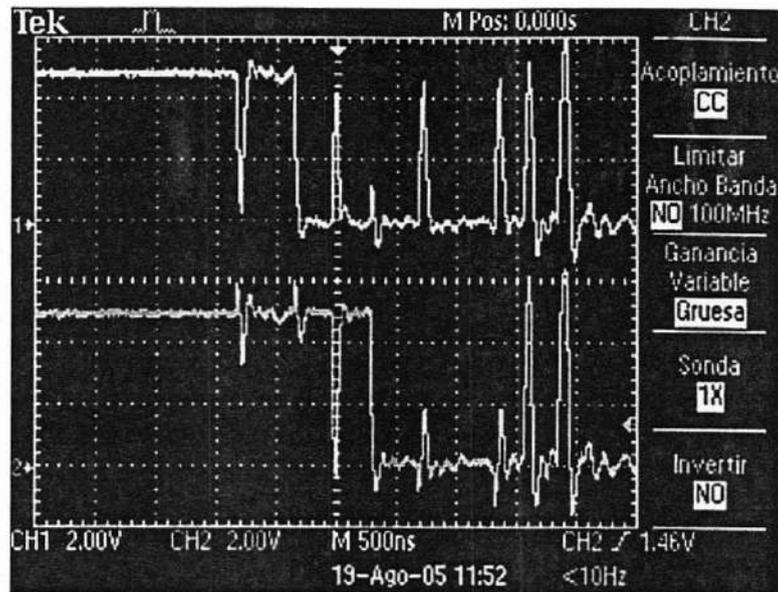


Figura 5.12. Señales de Datos D7 y D8 del CCD.

Con las pruebas mostradas se valida el funcionamiento de la cámara de CCD. En base a los resultados obtenidos de la CCD y los requerimientos del DSP se diseñó el sistema en conjunto con los demás dispositivos.

5.3 SEÑALES DE CONTROL CON EL FPGA

En la utilización del FPGA, debe manejar un control que sincronice a los demás dispositivos (Cámara y proceso de escritura en la FIFO); para ello es necesario conocer el sistema en su totalidad además de las características del mismo FPGA.

Al iniciar el sistema, el programa de control del FPGA, debe mandar un pulso de RESET hacia la memoria FIFO para borrar todos los datos no deseados o basura de información que pueda tener almacenada. Después debe enviar un pulso de inicio a la cámara de CCD para comenzar la transferencia de un *frame* o imagen.

Cada señal enviada por el FPGA, ya sea la señal de EXSYNC o la señal de RESET, tiene sus características como la duración de cada pulso. Por ejemplo la duración del pulso en bajo de EXSYNC debe de ser por lo menos de 100 ns. También

se entiende que se debe utilizar una señal asimétrica debido a que la duración del envío de un *frame* es mas largo que la duración del pulso en bajo de EXSYNC.

En la Fig. 5.1 (señal superior) se aprecia una señal asimétrica dada por el FPGA que puede ser enviada al CCD para controlar el envío de un *frame*. Esta señal muestra una duración de un nivel en bajo de aproximadamente 200ns y el nivel en alto puede tener una duración del orden de segundos (esto depende de la velocidad de lectura que se ejecute en la FIFO). Para efectos prácticos se redujo el tiempo del pulso positivo para apreciar dicha señal.

La señal inferior de la Fig. 5.13 muestra una onda cuadrada con una duración aproximada de 200 ns por ciclo para comparar la señal anterior.

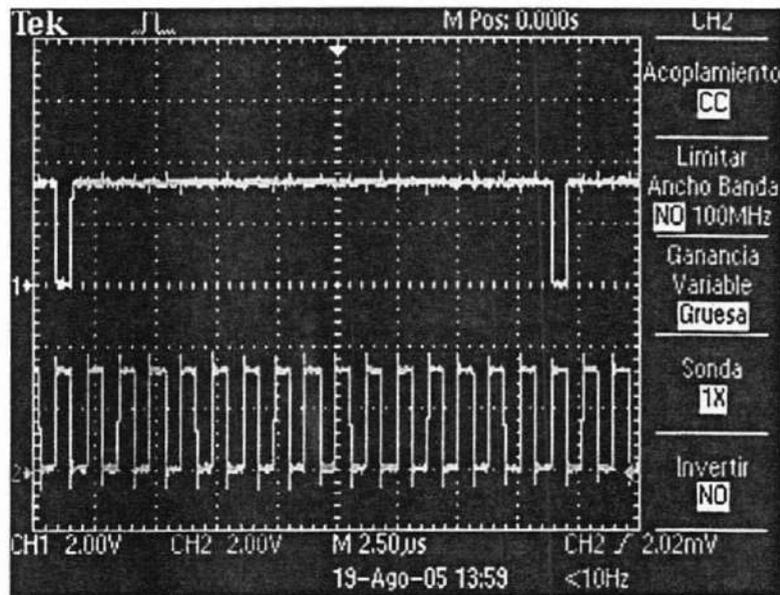


Figura 5.13. Ejemplo de señal EXSYNC dada por el FPGA.

En la Fig. 5.14 se ve la medición de voltaje diferencial del pulso negativo de la señal superior de la Fig. 5.1 (EXSYNC), generado por el FPGA, tomado después de pasar por el convertidor TTL a LVDS hacia la cámara de CCD.

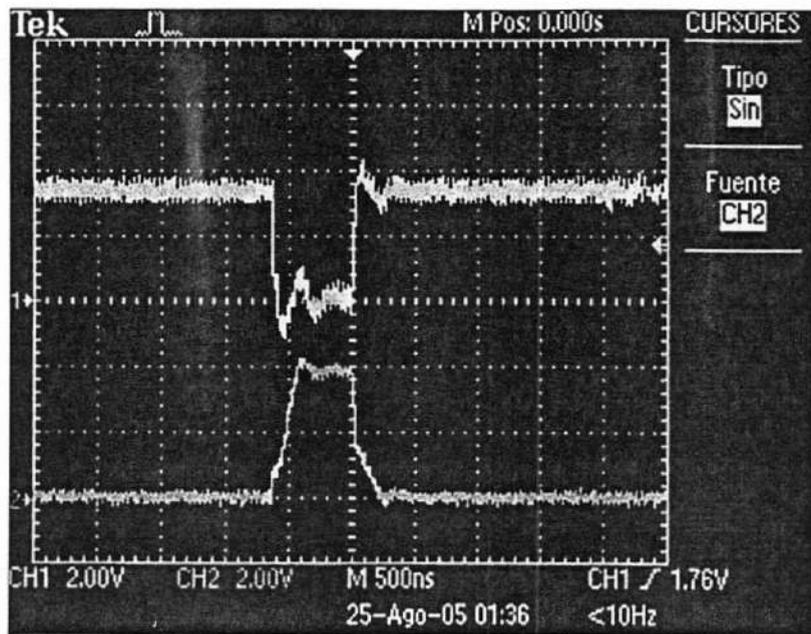


Figura 5.14. Señal EXSYNC (FPGA) después de la conversión a LVDS.

Para las pruebas de medición de señales de este trabajo de tesis, se generó una señal de EXSYNC con una duración de alrededor de 660 nanosegundos para el nivel negativo y 1.35 segundos, aproximadamente, para el nivel positivo. Como se mencionó anteriormente es una diferencia de tiempo muy grande, pero en la Fig.5.15 se puede apreciar como se ve el retraso del envío de *frame* en la señal FVAL, donde la duración del nivel positivo es la duración del envío del *frame* y la duración del nivel negativo es el tiempo de espera entre *frame* y *frame*. Se ve como la duración entre *frames* es mayor en comparación a la duración del envío de cada *frame*, con se ve en la señal superior de la Fig. 5.9

Una vez generada la señal de EXSYNC, para efectos de prueba es necesario generar la señal de RESET periódica para el borrado de la memoria FIFO. Esta señal es más compleja que EXSYNC pues la operación de borrado de la memoria depende de tres procesos operados por las siguientes señales:

- Que RESET este en alto lógico, y que su duración sea considerando,
- Que ocurran al menos cuatro transiciones del reloj de escritura o WRTCLK (dada por la señal STROBE de la cámara de CCD),
- Y que ocurran al menos cuatro transiciones del reloj de lectura o RDCLK (dada por la EMIF del DSP)

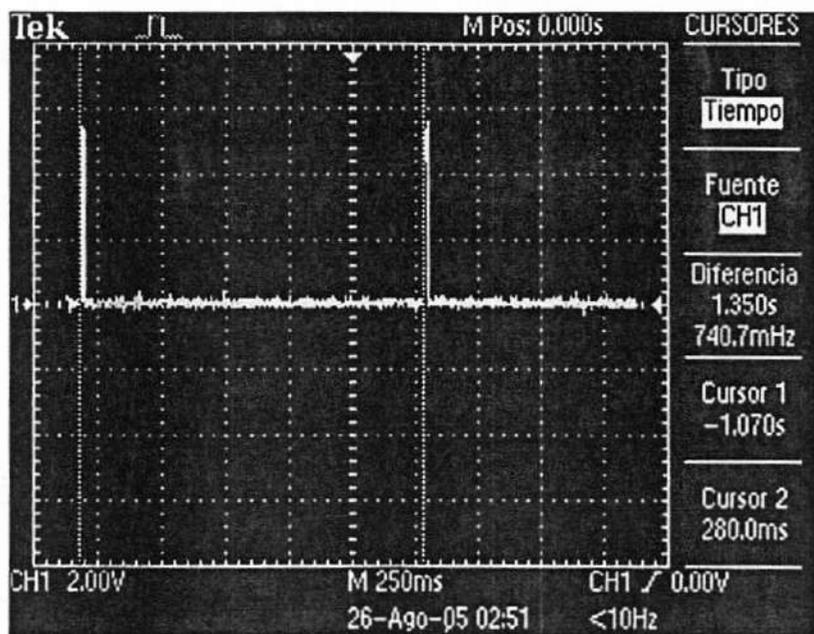


Figura 5.15 Señal FVAL con retraso.

Si la frecuencia de lectura es menor a la de escritura entonces se tendrá pérdida de datos, por ello el DSP debe operara una frecuencia de lectura mayor al de la cámara de CCD (25MHz). Como referencia, la frecuencia de la señal superior de la Fig. 5.16 es de 25MHz y la duración de la señal inferior (RESET) de la misma es de más de 4 ciclos de 25MHz (esto, si consideramos que el reloj de escritura y lectura es el mismo).

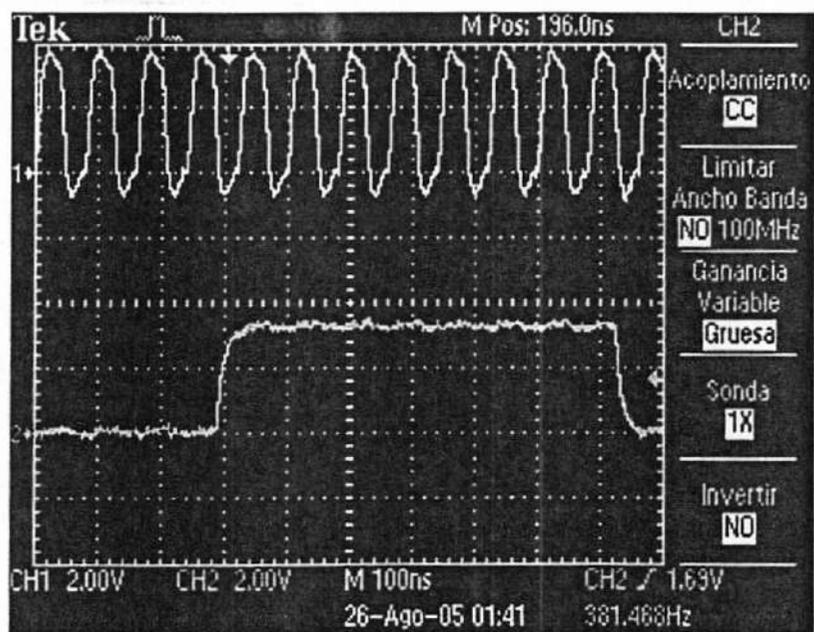
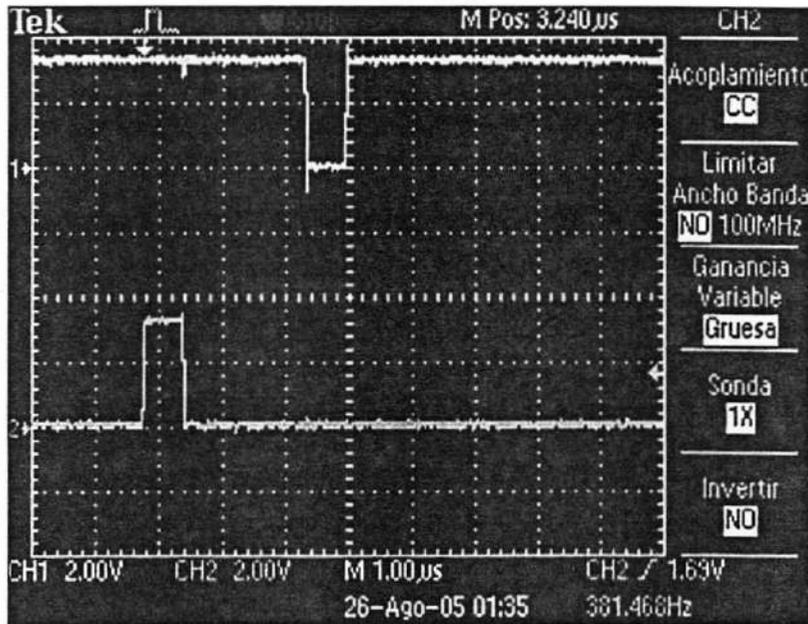


Figura 5.16. Pulso de RESET dado por el FPGA.

Una vez generadas las señales demandadas al FPGA, se generó el código implementando ambas señales para ver en forma periódica la forma en que iban a operar conforme al modo de operación del sistema. Además, como se mencionó, la señal de RESET de la FIFO debe empezar y terminar antes que comience la señal de EXSYNC como se muestra en la Fig. 5.17.



Para realizar las pruebas haciendo uso del FPGA se desarrollo un programa en lenguaje VHDL, tiendo una entrada de reloj (interno del FPGA) seguido de dos sumas de divisores de frecuencia para generar las señales de control (EXSYNC y RESET) y también la generación de señales de habilitación en estado alto, como se ilustra en la Fig. 5.18. y el código del diseño se muestra en el [Apéndice D]

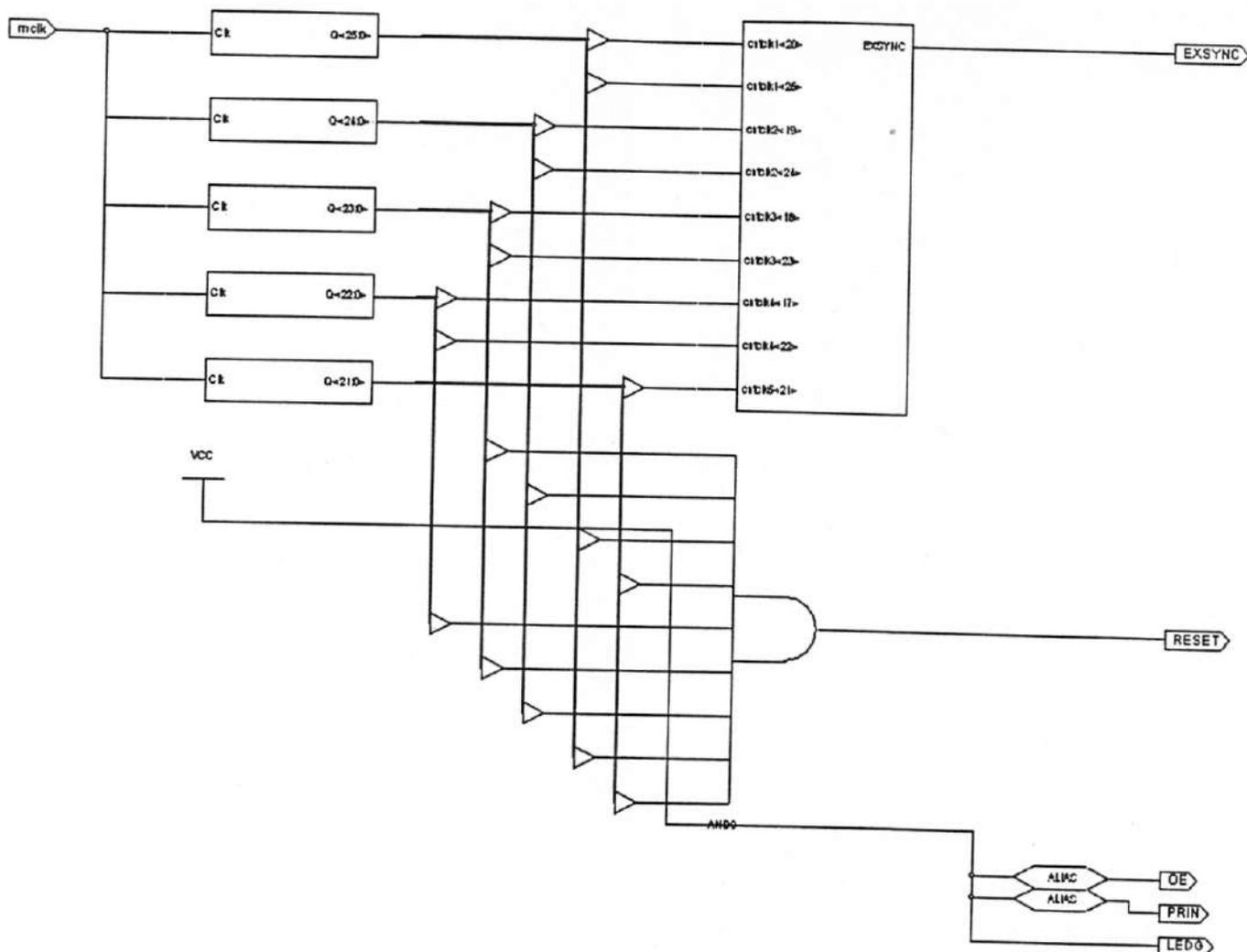


Figura 5.18 Diagrama a bloques del circuito de control con el FPGA.

5.4. MEMORIA FIFO

Para comprobar el funcionamiento de la memoria FIFO se utiliza la cámara de CCD que genera datos y su señal de escritura tal que la memoria disponga de información.

Para hacer la lectura de la memoria se necesita un reloj (RDCLK) proporcionado por el DSP, dado a que no se dispone del funcionamiento del DSP se genera una señal cuadrada de aproximadamente 200KHz, como entrada de reloj, por medio de un generador de señales. Las señales de datos de la memoria FIFO Q1 y Q2 se muestran en la Fig. 5.19, Q3 y Q4 en la Fig. 5.20 y Q7 y Q8 en la Fig. 5.21.

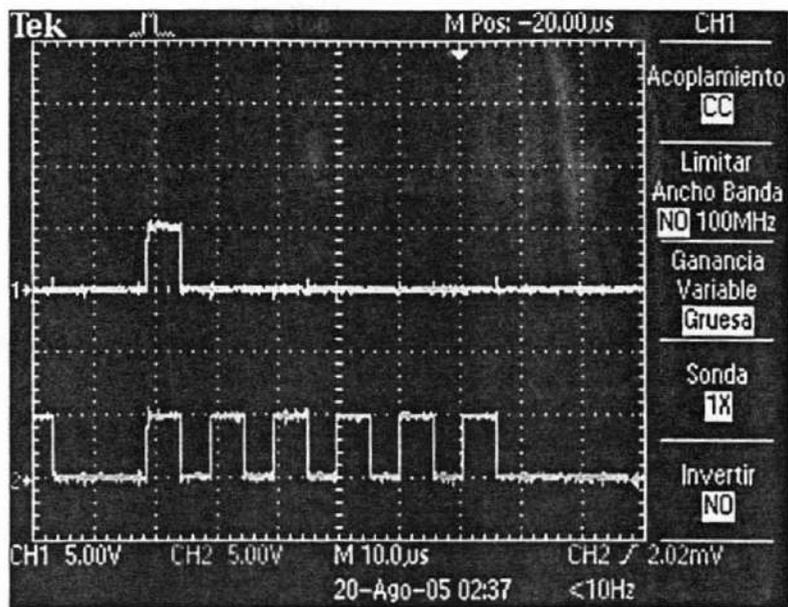


Figura 5.19. Señales de Datos Q1 y Q2 de la FIFO

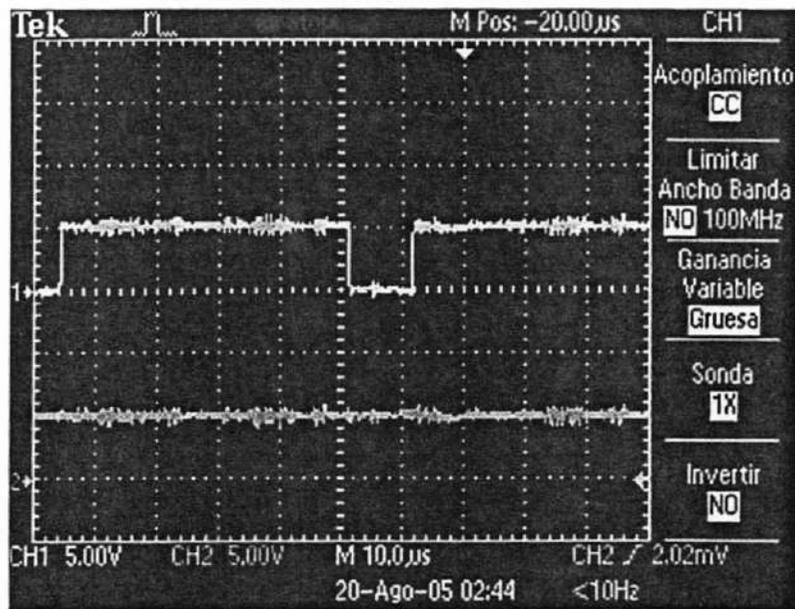


Figura 5.20. Señales de Datos Q3 y Q4 de la FIFO

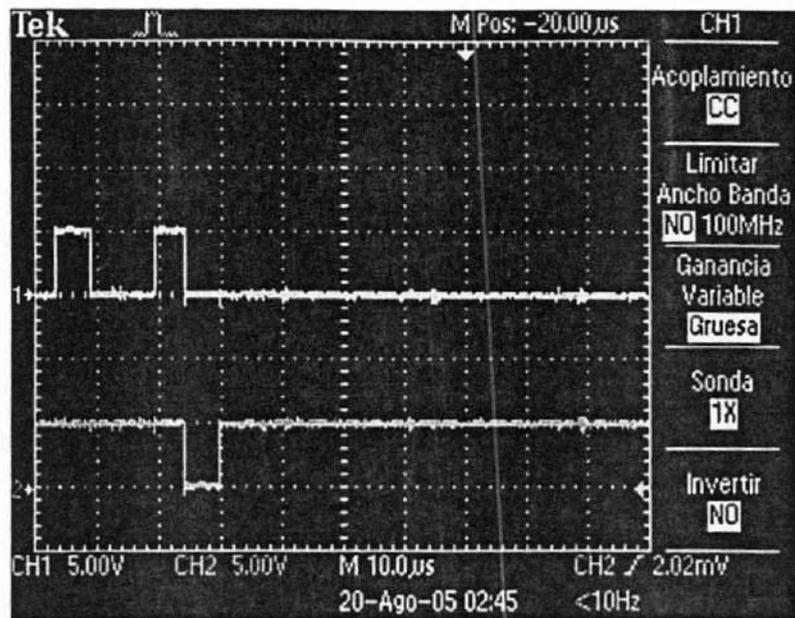


Figura 5.21. Señales de Datos Q7 y Q8 de la FIFO

Otro aspecto importante a revisar de la memoria son las banderas HF, AF/AE. Sin embargo, como éstas señales tienen una diferencia de tiempo muy grande entre su pulso negativo y positivo es difícil apreciarlas, es decir, como la diferencia de duración de pulso negativo y pulso positivo de la señal es muy grande, tampoco es posible visualizar los cambios de estados de las banderas. Para obtener evidencia de que había cambios en los estados de las banderas, entonces se generaron señales de pulso cuadrado de diferentes frecuencias.

Debido a que estas señales dependen de tres procesos como se mencionó anteriormente no se puede esperar ver una señal simétrica de las señales de estado de memoria de la FIFO. En la Fig. 5.22. se ve la señal de HF respecto a una señal de RESET generada por el FPGA cuya frecuencia es aproximadamente de 50KHz donde se ve que la señal de HF no es simétrica debido a la dependencia del borrado de la memoria.

El borrado de la memoria solo se habilita cuando la señal de RESET se encuentra en estado alto, es por esta razón por la cual las señales de bandera solo cambian cuando la señal de RESET se encuentra en el estado alto lógico.

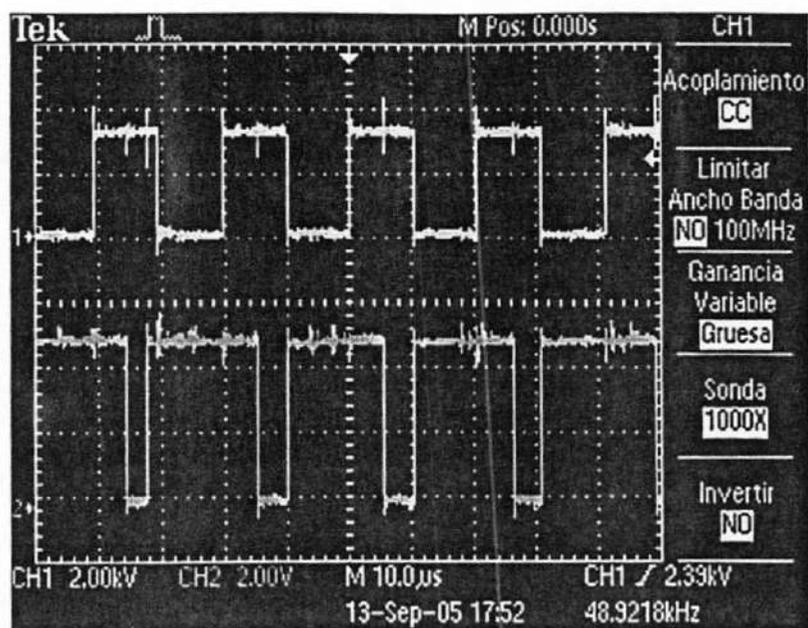


Figura 5.22. Señal de HF de la FIFO (inferior) señal de RESET del FPGA de 50 KHz (superior).

En las Fig. 5.22 y 5.23 las señales de AF/AE (inferior) se observa como cambia los estados de la memoria entre el llenado y vaciado, y el cambio de frecuencia de estos debido al cambio de frecuencia de la señal de RESET dada por el FPGA. En la Fig. 5.23 la señal de RESET tiene una frecuencia de aproximadamente 12KHz y en la Fig. 5.24 tiene una frecuencia de aproximadamente 6 KHz.

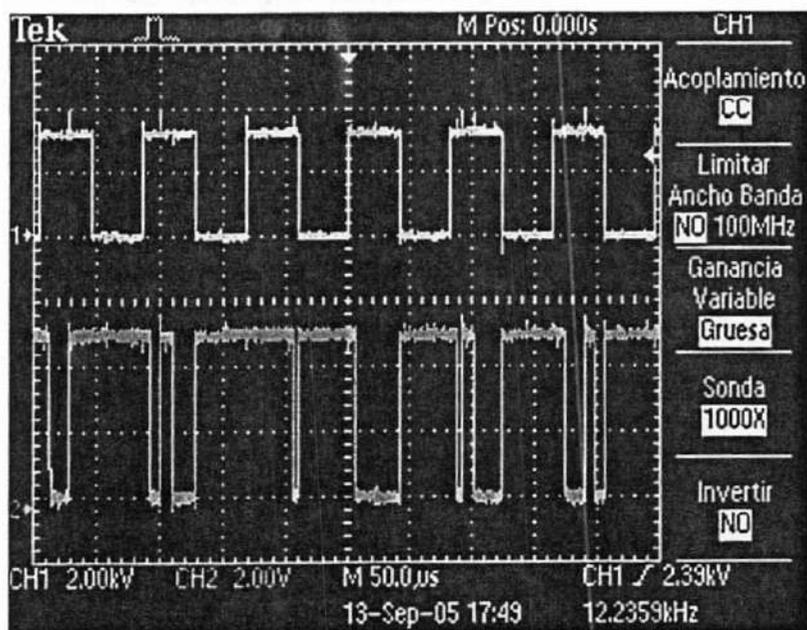


Figura 5.23. Señal de AF/AE de la FIFO (inferior) señal de RESET del FPGA de 12KHz (superior).

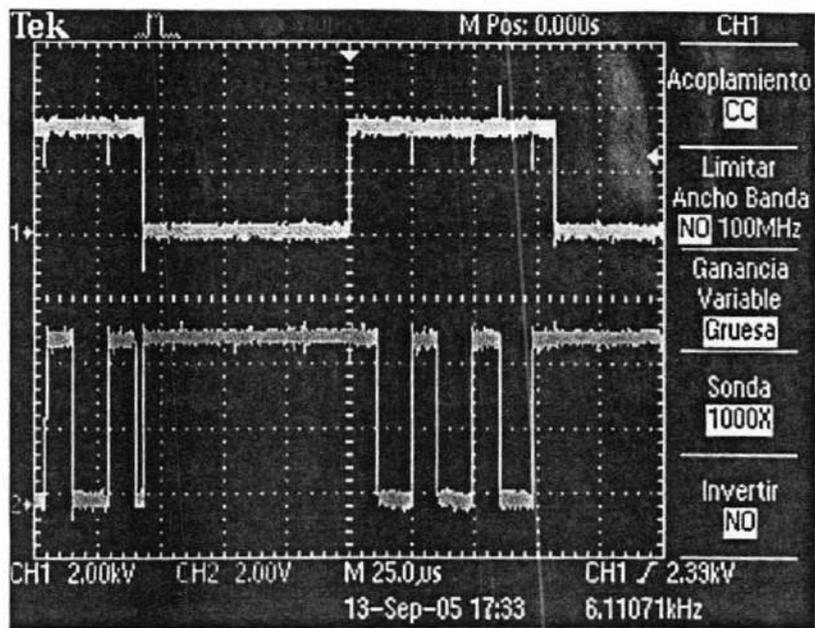


Figura 5.24. Señal de AF/AE de la FIFO (inferior) señal de RESET del FPGA de 6KHz (superior).

Capítulo 6. CONCLUSIONES

El trabajo descrito en esta tesis es parte del desarrollo de un sistema de procesamiento alternativo para la reconstrucción de imágenes en tres dimensiones empleando dispositivos de co-procesamiento. En este caso se espera desarrollar un sistema que procese datos con un DSP a partir de una cámara CCD. Por lo que se desarrolló una tarjeta interfaz para facilitarle al DSP el manejo de datos proveniente de la cámara CCD. Con esta interfaz se interconectaron todos los componentes del sistema de adquisición (FPGA, FIFO, LVDS, CCD).

Se llevaron a cabo pruebas de cada dispositivo que conforma esta interfaz y al sistema de escritura/lectura de la FIFO. Esto se realizó haciendo pruebas de las banderas de la memoria FIFO para apreciar su funcionamiento consistente en el borrado (probando las señales provistas por el FPGA) y las señales de llenado y vaciado de la memoria FIFO.

La comprobación del funcionamiento del FPGA sincronizando la memoria y la cámara, funcionó de manera correcta, resultando en un ahorro de procesos por parte del procesador y evitando la necesidad de dar direcciones para acceder a una memoria (característica de las memorias RAM). Al quitar este peso de trabajo al procesador se hace más eficiente y se procura el procesamiento en tiempo real.

También se tuvo un ahorro de circuitos de lógica y control digital del sistema, evitando la alternativa de desarrollar varios PCBs, implementando solo uno en un FPGA.

Considerando los resultados mostrados, el sistema está listo para el acoplamiento con el DSP. El trabajo posterior a este desarrollo consiste en que el usuario del DSP emplee esta interfaz para implementar los algoritmos de co-procesamiento en el sistema de reconstrucción tridimensional de objetos.

BIBLIOGRAFÍA

- [1] Scientific Imaging Technologies Inc. (SITE), "An Introduction to Scientific Imaging Charge-Coupled Devices", <http://www.site-inc.com/index2.html>, 1994.
- [2] H. Zimmermann, "Integrated Silicon Opto-electronics" Ed. Springer, 2000.
- [3] Jacob Millinan, Arvin Grable, "MICROELECTRONICS" Ed. McGRAW-HILL International Editions, 1987.
- [4] Oldfield-Dorf, "Field Programmable Gate Array" Ed. Wiley Interscience, 1995.
- [5] Stephen M. Trimberger, "Field Programmable Gate Array" Ed. Kluwer Academic Publishers, 1994.
- [6] Sudhakar Yalamanchili, "VHDL Starter's Guide" Ed. Prentice Hall, 1998.
- [7] John F. Wakerly, "DISEÑO DIGITAL Principios y Practicas" Ed. Prentice Hall, 2001.
- [8] Charles H. Roth, Jr, "Fundamentals of LOGIC DESIGN" Ed. PWS Publishing Company, 1995.
- [9] M.Morris Mano, "DISEÑO DIGITAL" Ed. Prentice Hall, 1987.
- [10] Mitsuo Takeda and Kazuhiro Mutoh, "Fourier transforms profilometry for the automatic measurement of 3-D object shapes" Applied Optics, Vol.22, No.24, 1983.
- [11] Manual de usuario "SN74ACT7807" de Texas Instrument Incorporated www.ti.com, 1991.
- [12] Manual de usuario "CA-D8" de Dalsa www.dalsa.com, 1999.

- [13] Hojas esquematicas de “DS90C031 y DS90C032” de Nacional Semiconductor www.national.com, 1998.
- [14] Manual de usuario “TMS320C6711 Code Componer Studio” Texas Instrument Incorporated www.ti.com, 2001.
- [15] Manual de usuario “D2SB” de Digilent Inc. www.digilent.com, 2002.
- [16] Hoja esquemática “D2SB” de Digilent Inc. www.digilent.com, 2002.
- [17] Hoja esquemática “Spartan-IIE 1.8V FPGA Family” de Xilinx www.xilinx.com, 2004.
- [18] Información de DSPs de Texas Instrument www.ti.com, 2005.

APÉNDICE A. PROCESO DE PERFILOMETRÍA

El término perfilometría se utiliza en la medición de un perfil o silueta de un cuerpo para obtener su representación en tres dimensiones.

En un principio el proceso de perfilometría se realizaba con la técnica de contornos de Moire llevada a cabo haciendo incidir un patrón de franjas de luz en un cuerpo y midiendo las deformaciones de las franjas.

Una posterior mejora es la técnica de Perfilometría por Transformada de Fourier o FTP (Fourier Transform Profilometry). Ésta tiene mejor resolución o sensibilidad y funciona con ayuda de métodos ópticos. Con esto se obtiene una imagen digitalizada de tres dimensiones a partir de dos imágenes de dos dimensiones, la imagen original (imagen del patrón de franjas de luz sin el cuerpo) y la imagen deformada. En la Fig. A-1. se muestra el patrón de las deformaciones hechas por el cuerpo.

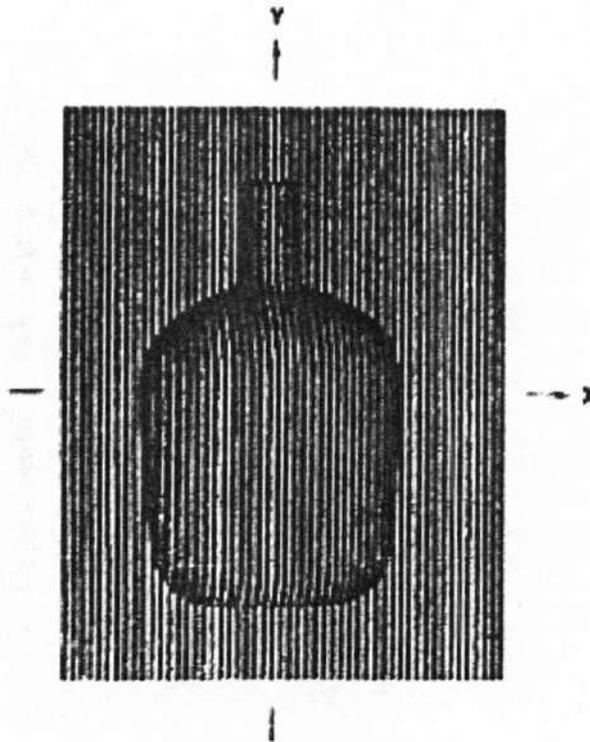


Figura A-1. Figura tomada por una CCD con el patrón de franjas.

En la Fig. A-2 se muestra las deformaciones de luz de una franja vertical en el eje

Y

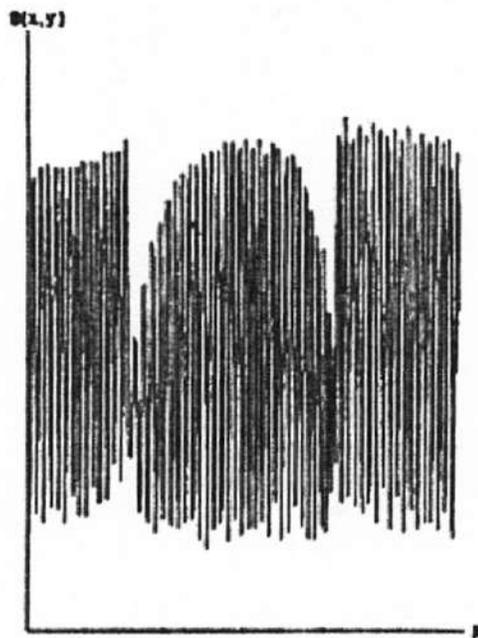


Figura A-2. Información de una línea de datos.

Los datos de cada franja son transformados al dominio de la frecuencia mediante la aplicación de la transformada de Fourier, para posteriormente analizar su espectro como se muestra en la Fig. A-3. y con ello filtrar frecuencias no deseadas.

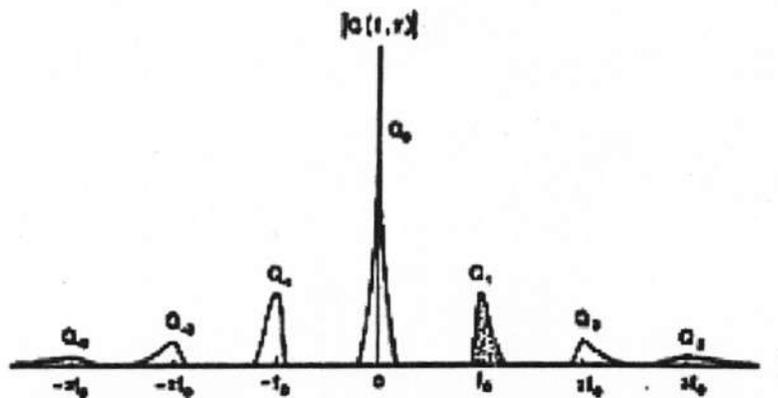


Figura A-3. Información de una línea de datos en dominio de la frecuencia.

Después que las frecuencias innecesarias son filtradas, se aplica la Transformada Inversa de Fourier, para volver la información a su representación original. En la Fig. A-4 se muestra la imagen del cuerpo ya reconstruida en tres dimensiones después de varios procesamientos. [10]

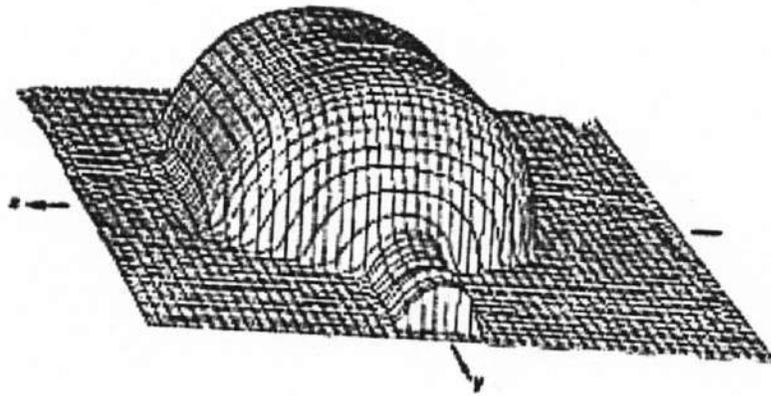


Figura A-4. Figura reconstruida en tres dimensiones.

APÉNDICE B. INTERFAZ DE MEMORIA EXTERNA DEL DSP (EMIF).

La interfase de memoria externa (EMIF) del DSP soporta una variedad de dispositivos externos a la tarjeta DSK (TMS32C6411) como:

- Synchronous-burst SRAM (SBSRAM)
- Synchronous-burst DRAM (SDRAM)
- Dispositivos asíncronos, incluyendo SRAM, ROM, and FIFOs
- Un dispositivo de memoria-compartida externo

La tarjeta DSK C6711 requiere de los servicios de un bus externo para dos necesidades:

1. Chip controlador DMA
2. Dispositivo controlador de memoria-compartida externa

En la Fig. B.1. se observa el diagrama de bloques de la tarjeta C6711 resaltando la EMIF.

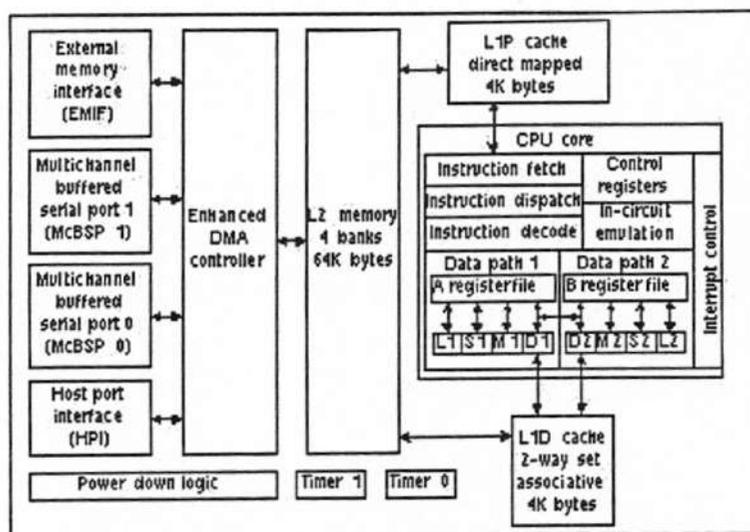


Figura B.1. Diagrama general del DSP.

La EMIF está formada por registros; el control de la EMIF y la interfase de memoria compuesta por un mapa de registros mostrados en la Tabla B.1.

Nombre del registro de la EMIF	Siglas	Dirección (Hex Byte)
Registro de control global	GBLCTL	0180 0000
CE registro de control de espacio 1	CECTL1	0180 0004
CE registro de control de espacio 0	CECTL0	0180 0008
CE registro de control de espacio 2	CECTL2	0180 0010
CE registro de control de espacio 3	CECTL3	0180 0014
Registro de control SDRAM	SDCTL	0180 0018
Registro de tiempo SDRAM	SDTIM	0180 001C
Registro de extensión SDRAM	SDEXT	0180 0020

Tabla B.1. Registros de la EMIF del DSP.

Físicamente la EMIF tiene terminales de entrada y de salida de datos, terminales de bandera y de control. En la Fig. B.2 se puede ver como esta compuesta la EMIF del C6711 y en la Tabla B.2 describe la función de cada uno de las terminales de la interfaz. La interfaz de memoria de expansión, provee una señal asíncrona del DSP a la daughterboard (tarjeta hija) AFE.

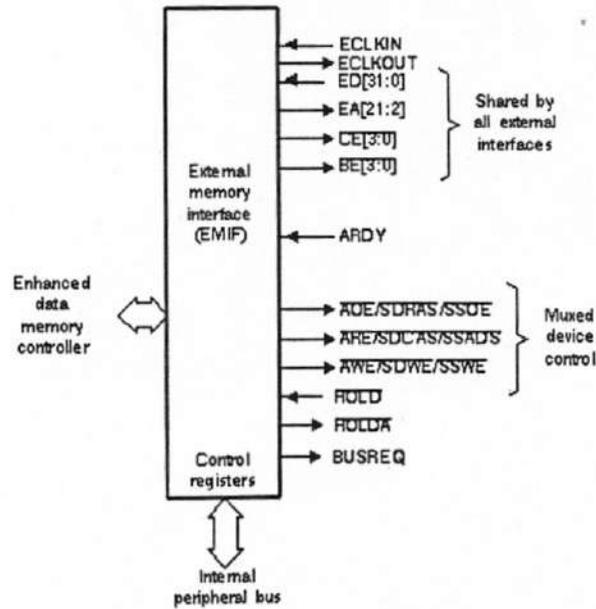


Figura B.2 Terminales de entrada/salida de la EMIF del DSP.

Pin	I/O/Z	Descripción
CLKOUT1	O	Reloj de salida. Corre a la misma velocidad que el procesador.
CLKOUT2	O	Reloj de salida. Corre a la mitad de velocidad del procesador.
BUSREQ	O	Se activa en alto la señal de solicitud del bus.
ECLKOUT	O	Reloj de salida de la EMIF. Todos los pines de I/O están gobernados por el reloj de la EMIF
ED[31:0]	I/O/Z	Datos de entrada y salida (I/O). 32 bits de datos de I/O de memorias y periféricos externos.
EA[21:2]	O/Z	Salida de direccionamiento externo. Maneja bits de 21-2 de direcciones de byte.
CE0	O/Z	Se activa en bajo la selección de espacio de memoria CE0
CE1	O/Z	Se activa en bajo la selección de espacio de memoria CE1
CE2	O/Z	Se activa en bajo la selección de espacio de memoria CE2
CE3	O/Z	Se activa en bajo la selección de espacio de memoria CE3
BE[3:0]	O/Z	Se activa en bajo la habilitación de byte. Byte individuales y medias palabras (halfwords) pueden ser seleccionadas por lectura y escritura al mismo tiempo.
ARDY	I	Listo (Ready). Activado en bajo la bandera de ready para las entradas es utilizada para hacer estados de pausas para memoria y periféricos lentos.
HOLD	I	Se activa en bajo, la solicitud de espera (tres estados) del bus externo.
HOLDA	O	Se activa en bajo, la espera del reconocimiento del bus externo.

Tabla B.2. De Terminales de la EMIF del DSP.

- 20 External Address Signal (EA [21:2]): Las 20 señales direccionadas externas del DSP están disponibles en la interfase de memoria de expansión, permitiendo más de 256 Mbytes de memoria externa para ser direccionada a cualquiera de CE2 y CE3 espacios de memoria.

- *32 External Data Signal (ED [31:0])*: Las 32 señales de datos externas están disponibles en la interfase de memoria de expansión para soportar palabras completas de 32 bits accedendo a la daughterboard.
- *CE Memory Space Enable*: Este espacio de memoria disponible, en la interfase de memoria de expansión, permite el acceso a la memoria de la daughterboard y a la memoria de dispositivos mapeados.
- *4 EMIF Control Signal*: Las señales de control de la EMIF del DSP (ARE, AWE, AOE y ARDY) proveen el control de acceso de memoria a una daughterboard.

[14]

APÉNDICE C. DISEÑO ESQUEMÁTICO Y *LAYOUTS* O CARAS DE LA TARJETA PCB

En el circuito esquemático se marcaron las conexiones de todas las terminales de los puertos y circuitos integrado. De esta manera se hace la base del PCB ya teniendo la guía de conexiones hechas en el diseño. En la Fig. C.1 se muestra dicho diseño.

Una vez que el diseño esquemático estuvo terminado, se elaboró el diseño de Layout. Para este se utilizaron dos niveles de interconexión en la placa de PCB debido a que tenía demasiadas conexiones debido a la interconexión de varios dispositivos digitales entre los que hay buses de datos de 8 bits, las señales de convertidores LVDS los cuales utilizan señales diferenciales. En la Fig. C.2 se muestra el diseño del Layout. Las Fig. C.3 y C.4 muestran el diseño de cada nivel por separado.

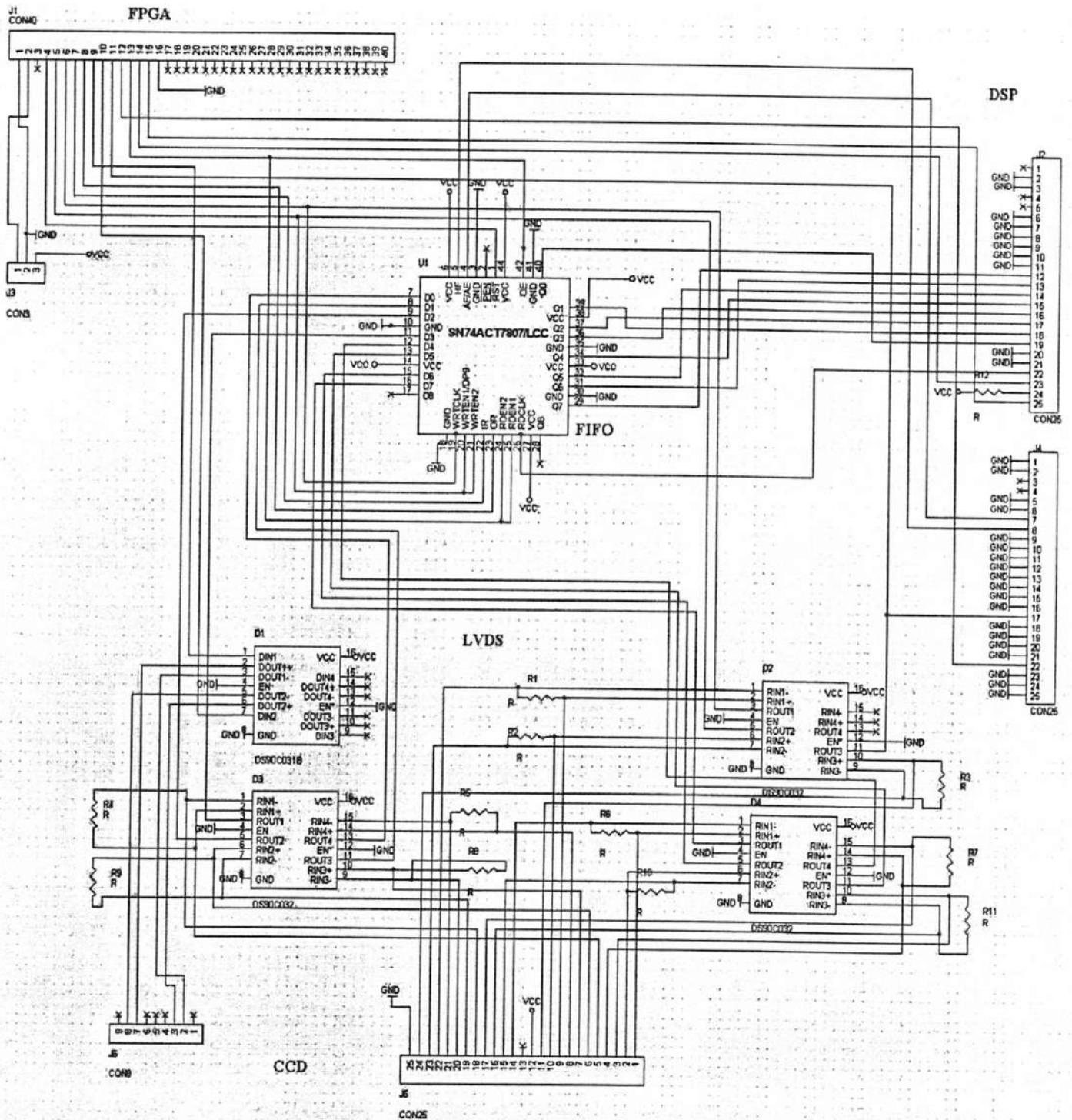


Figura C.1. Diseño esquemático del PCB

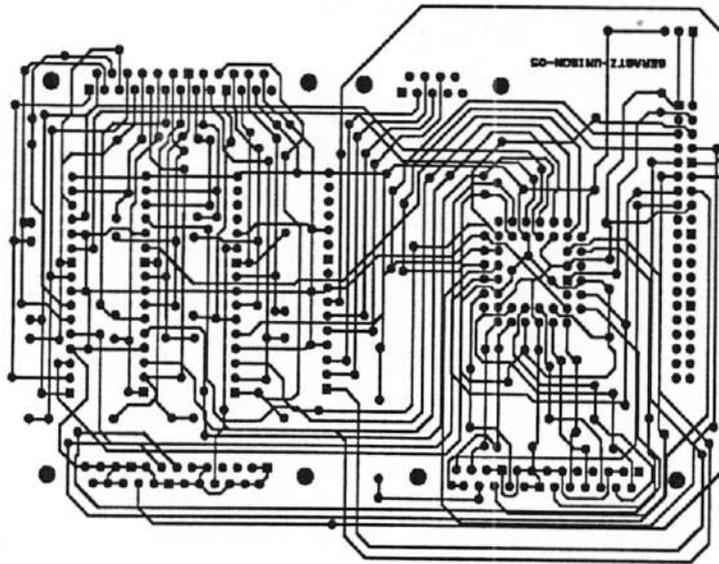


Figura C.2. Diseño de Layout del PCB.

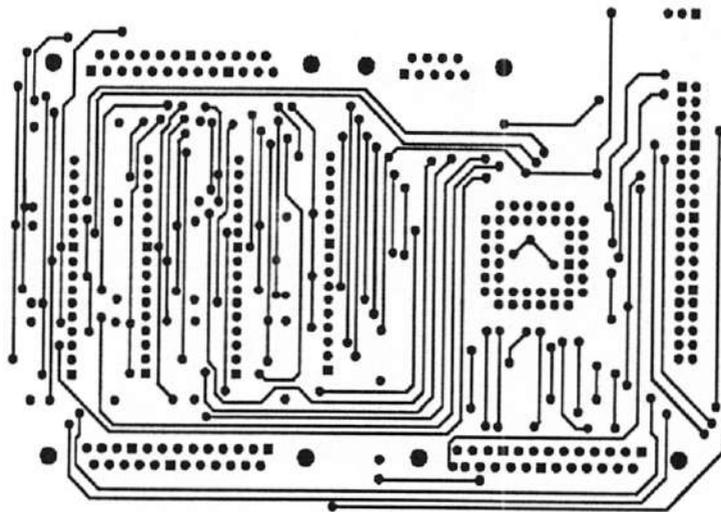


Figura C.3. Anverso del Layout del PCB (lado de componentes).

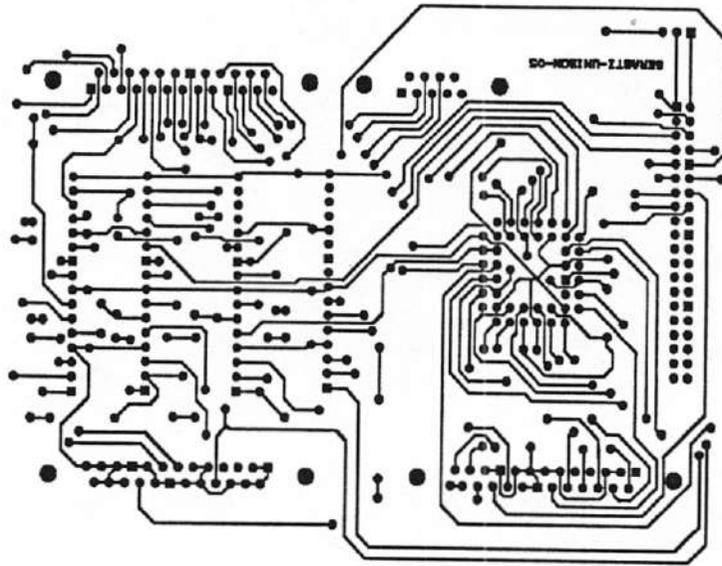


Figura C.4. Reverso del Layout del PCB.

APÉNDICE D. CODIGO DE PRUEBAS (VHDL) PARA EL FPGA

```
--Código de pruebas en la utilización del FPGA  
--en el sistema de adquisición de datos  
--Gerardo Gutiérrez H. LTE-UNISON-05
```

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

```
entity ex is generic(  
    k: integer :=26);  
    port(  
        mclk: in std_logic;  
        EXSYNC,RESET,OE,LEDG,PRIN: out std_logic);
```

```
end ex;
```

```
architecture Behavioral of ex is
```

```
    signal cntclk1 : std_logic_vector(k-1 downto 0);  
    signal cntclk2 : std_logic_vector(k-2 downto 0);  
    signal cntclk3 : std_logic_vector(k-3 downto 0);  
    signal cntclk4 : std_logic_vector(k-4 downto 0);  
    signal cntclk5 : std_logic_vector(k-5 downto 0);  
    signal cntclk6 : std_logic_vector(k-6 downto 0);  
    signal cntclk7 : std_logic_vector(k-7 downto 0);  
    signal cntclk8 : std_logic_vector(k-8 downto 0);  
    signal cntclk9 : std_logic_vector(k-9 downto 0);  
    signal cntclk10 : std_logic_vector(k-10 downto 0);  
    signal cntclk11 : std_logic_vector(k-11 downto 0);  
    signal cntclk12 : std_logic_vector(k-12 downto 0);  
    signal cntclk13 : std_logic_vector(k-13 downto 0);  
    signal cntclk14 : std_logic_vector(k-14 downto 0);  
    signal cntclk15 : std_logic_vector(k-15 downto 0);  
    signal cntclk16 : std_logic_vector(k-16 downto 0);  
    signal cntclk17 : std_logic_vector(k-17 downto 0);  
    signal cntclk18 : std_logic_vector(k-18 downto 0);  
    signal cntclk19 : std_logic_vector(k-19 downto 0);  
    signal cntclk20 : std_logic_vector(k-20 downto 0);  
    signal cntclk21 : std_logic_vector(k-21 downto 0);  
    signal cntclk25 : std_logic_vector(k-25 downto 0);  
    signal a1,b1,c1,d1,e1,f1,g1,h1,i1,j1,k1,l1,m1,n1,o1,p1,q1,r1,s1,t1,u1,x1,y1,reloj1 : std_logic;
```

```
begin
```

```
-----  
Reloja: process(mclk)  
    begin  
        if mclk = '1' and mclk'Event then  
            cntclk1 <= cntClk1 + 1;  
        end if;  
    end process;
```

```
-----  
Relojb: process(mclk)
```

```
begin
if mclk = '1' and mclk'Event then
cntclk2 <= cntClk2 + 1;
end if;
end process;
```

Relojc: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk3 <= cntClk3 + 1;
end if;
end process;
```

Relojd: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk4 <= cntClk4 + 1;
end if;
end process;
```

Reloje: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk5 <= cntClk5 + 1;
end if;
end process;
```

Relojf: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk6 <= cntClk6 + 1;
end if;
end process;
```

Relojg: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk7 <= cntClk7 + 1;
end if;
end process;
```

Relojh: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk8 <= cntClk8 + 1;
end if;
end process;
```

Reloji: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk9 <= cntClk9 + 1;
end if;
end process;
```

Relojj: process(mclk)

```
begin
if mclk = '1' and mclk'Event then
cntclk10 <= cntClk10 + 1;
end if;
```

end process;

```
Relojk: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk11 <= cntClk11 + 1;
        end if;
    end process;
```

```
Relojl: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk12 <= cntClk12 + 1;
        end if;
    end process;
```

```
Relojm: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk13 <= cntClk13 + 1;
        end if;
    end process;
```

```
Relojn: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk14 <= cntClk14 + 1;
        end if;
    end process;
```

```
Relojo: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk15 <= cntClk15 + 1;
        end if;
    end process;
```

```
Relojp: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk16 <= cntClk16 + 1;
        end if;
    end process;
```

```
Relojq: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk17 <= cntClk17 + 1;
        end if;
    end process;
```

```
Relojr: process(mclk)
    begin
        if mclk = '1' and mclk'Event then
            cntclk18 <= cntClk18 + 1;
        end if;
    end process;
```

```
Relojs: process(mclk)
    begin
```

```

if mclk = '1' and mclk'Event then
cntclk19 <= cntClk19 + 1;
end if;
end process;

```

```

-----
Relojt: process(mclk)
begin
if mclk = '1' and mclk'Event then
cntclk20 <= cntClk20 + 1;
end if;
end process;

```

```

-----
Reloju: process(mclk)
begin
if mclk = '1' and mclk'Event then
cntclk21 <= cntClk21 + 1;
end if;
end process;

```

```

-----
Relojreloj1: process(mclk)
begin
if mclk = '1' and mclk'Event then
cntclk25 <= cntClk25 + 1;
end if;
end process;

```

```

a1<= cntclk1(k-1);
b1<= cntclk2(k-2);
c1<= cntclk3(k-3);
d1<= cntclk4(k-4);
e1<= cntclk5(k-5);
f1<= cntclk1(k-6);
g1<= cntclk2(k-7);
h1<= cntclk3(k-8);
i1<= cntclk4(k-9);
j1<= cntclk5(k-10);
k1<= cntclk1(k-11);
l1<= cntclk2(k-12);
m1<= cntclk3(k-13);
n1<= cntclk4(k-14);
o1<= cntclk5(k-15);
p1<= cntclk1(k-16);
q1<= cntclk2(k-17);
r1<= cntclk3(k-18);
s1<= cntclk4(k-19);
t1<= cntclk5(k-20);
u1<= cntclk5(k-21);
--reloj1<= cntclk25(k-25);
x1<= a1 and b1 and c1 and d1 and e1 and f1 and g1 and h1 and i1 and j1 and k1 and l1
and m1 and n1 and o1 and p1 and q1 and r1 and s1 and t1 and u1;
y1<= a1 and b1 and c1 and d1 and e1 and f1 and g1 and h1 and i1 and j1 and k1 and l1
and m1 and n1 and o1 and p1 and q1 and r1 and (not s1) and t1 and u1;
--x1<=t1 and u1;
--x1<= a1 and b1 and c1 and d1 and e1 and f1 and g1 and h1 and i1 and j1 and k1 and l1
and m1 and n1 and o1 and p1;-- and q1 and r1 and s1 and t1 and u1;
--y1<= d1 and e1 and f1 and g1 and h1 and i1 and j1 and k1 and l1 and m1 and n1 and
o1 and p1 and q1 and r1 and (not s1) and t1 and u1;

```

```

--y1<= q1;
--a<=a1;
--b<=b1;
--c<=c1;
--d<=d1;
--e<=e1;
--x1<= e1 and d1 and c1;
--x1<= j1 and k1 and l1 and m1 and n1 and o1 and p1 and q1 and r1 and s1 and t1 and
ul;
and u1;

--y1<= j1 and k1 and l1 and m1 and n1 and o1 and p1 and q1 and r1 and (not s1) and t1

LEDG<='1';
RESET<= y1;
EXSYNC<= not x1;
OE<='1';
PRIN<='1';

```

```

end Behavioral;

```