



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



Tesis

Desarrollo e implementación de un sistema de adquisición de señales de ultrasonido

Para obtener el título de
Ingeniero Eléctrico y Electrónico

Presenta

Luis Albert Zavala Mondragón

Dirigida por

M. C. Jaime Fabián Vázquez de la Rosa

Ciudad Universitaria

Enero de 2014

Agradecimientos

Cada uno de mis logros forma parte de un esfuerzo colectivo. Quiero aprovechar estas líneas para expresar mi gratitud a quienes me apoyaron a conseguirlos: mi madre, Rosalba Mondragón Moreno, mi padre, Juan Zavala Negrete, y mis hermanos, de quienes tanto he aprendido.

Asimismo, le agradezco a toda la gente que participó académicamente en este proyecto, pero en especial a la Dra. Lucía Medina Gómez, quien me brindó los recursos para realizarlo, y al Dr. Jaime Fabián Vázquez de la Rosa, quien fungió como mi asesor, sin dejar de lado a los doctorandos Edgar Alonso Trueba y Carlos Armando Villagómez, ellos sentaron las bases del sistema aquí presentado y compartieron sus conocimientos conmigo.

Finalmente, quiero agradecer a SEP-CONACyT131937 por la beca que me ha otorgado; gracias a este apoyo ha sido posible finalizar el presente proyecto.

Índice

Contenido

Capítulo 1: Marco teórico	1
1.1 Introducción al ultrasonido.....	1
1.1.1 ¿Qué es el ultrasonido?.....	1
1.1.1.1 Formación de ecos.....	2
1.1.1.2 Atenuación	3
1.1.2 Configuraciones comunes para ensayos con ultrasonido	3
1.1.2.1 Pulso-eco	3
1.1.2.2 Transmisión.....	4
1.1.3 Aplicaciones del ultrasonido en ensayos no destructivos.....	4
1.1.4 Aplicaciones del ultrasonido en medicina	5
1.1.5 Transductores ultrasónicos	5
1.2 Señales.....	6
1.2.1 Señales analógicas.....	6
1.2.2 Señales digitales.....	7
1.2.3 Conversión A/D	7
1.3 Sistemas digitales	7
1.3.1 Circuitos combinacionales.....	8
1.3.2 Circuitos secuenciales.....	8
1.3.3 Registros	9
1.3.3.1 Transferencia de registros (RTL)	9
1.4 Máquinas de estados finitos	9
1.5 Dispositivos lógico-programables: El FPGA	10
1.6 Lenguajes de descripción de hardware (HDL)	12
1.6.1 VHDL	12
1.7 Buffer FIFO.....	12
1.8 Memoria SDRAM	13

Capítulo 2: Especificaciones del sistema y selección de componentes	15
2.1 Sistemas computacionales de adquisición de datos y grabadores digitales de formas de onda	16
2.2 Módulo de acondicionamiento analógico y conversión analógica digital (MAACAD) ..	16
2.2.1 Tratamiento analógico de señales	16
2.2.1.1 Amplificación.....	16
2.2.1.1.1 Amplificador de bajo ruido.....	17
2.2.1.1.2 Amplificador de ganancia programable (PGA).....	17
2.2.1.2 Filtrado	17
2.2.2 Conversión A/D	18
2.3 Módulo de almacenamiento y transmisión (MAT).....	18
2.3.1 Etapa de recepción de datos	19
2.3.2 Etapa de almacenamiento.....	19
2.3.3 Etapa de comunicación para el despliegue de resultados	20
2.3.4 Selección de los componentes	20
2.3.4.1 Selección de etapa de acondicionamiento analógico y ADC	20
2.3.4.2 Selección de etapa de almacenamiento y transmisión	20
2.3.5 Características del hardware seleccionado [16].....	21
2.3.5.1 Módulo de acondicionamiento analógico y conversión analógica digital	22
2.3.5.1.1 Analog Front End AFE5808/08A [17]	23
2.3.5.2 Selección de tarjeta de almacenamiento y transmisión	24
2.3.5.3 Adaptador FMC-ADC	27
Capítulo 3: Diseño del sistema	28
3.1 Diseño del hardware del Sistema de Captura y Almacenamiento.....	31
3.1.1 Funcionamiento general del sistema.....	31
3.1.1.1 Ruta de datos	32
3.1.1.1.1 Bloque de entrada.....	33
3.1.1.1.2 Captura de señales LVDS	33
3.1.1.1.2.1 FIFO	34
3.1.1.1.2.2 Submuestreador	35
3.1.1.1.2.3 Concatenador	36
3.1.1.1.3 Arbitraje y multiplexado	37

3.1.1.1.3.1	Arbitraje	37
3.1.1.1.4	Bloque Controlador de Memoria (MCB)	38
3.1.1.1.4.1	FIFO de escritura	39
3.1.1.1.4.2	FIFO de lectura	39
3.1.1.1.4.3	FIFO de comandos	40
3.1.1.1.4.4	Señales de reloj	40
3.1.1.1.5	Control de flujo de salida de datos	40
3.1.1.1.6	UART	41
3.1.1.1.6.1	Recepción	42
3.1.1.1.6.2	Control de acceso	43
3.1.1.2	Sistemas de control	44
3.1.1.2.1	Registros de control	44
3.1.1.2.1.1	Canales habilitados por cada disparo del pulsador	44
3.1.1.2.1.2	Número de muestras (ciclos de escritura)	44
3.1.1.2.1.3	Factor de submuestreo	45
3.1.1.2.2	Sistema de configuración de registros	45
3.1.1.2.2.1	Máquina de validación de datos	46
3.1.1.2.3	Sistema de control principal	48
3.2	Diseño de software del sistema	50
3.2.1	Configuración	50
3.2.1.1	Estructura de parámetros	51
3.2.1.2	Introducción, validación y ajuste de parámetros	51
3.2.1.3	Generación de cadenas de configuración y envío por puerto serie	51
3.2.1.3.1	Módulo de acondicionamiento analógico y conversión analógico-digital ..	51
3.2.1.3.2	Módulo almacenamiento y transmisión	53
3.2.2	Recepción y organización de datos	54
3.2.2.1	Recepción y organización de datos	55
3.2.2.2	Agrupamiento de datos	56
Capítulo 4:	Funcionamiento del sistema	57
4.1	Ejemplo de funcionamiento	57
4.1.1	Inicialización	58
4.1.2	Introducción de parámetros	58

4.1.3	Inicio de adquisición.....	59
4.2	Funcionamiento de la etapa de configuración del sistema	59
4.2.1	Software de configuración	59
4.2.1.1	Configuración del MAACAD.....	59
4.2.1.1.1	Configuración de prueba	59
4.2.1.1.2	Funciones implementadas y captura de señales de configuración	60
4.2.1.2	Configuración del MAT	62
4.2.2	Hardware de configuración	62
4.2.2.1	Verificación de datos de configuración.....	63
4.2.2.2	Configuración de registros.....	63
4.2.2.3	Registros de configuración	64
4.2.3	Almacenamiento de datos	65
4.2.3.1	Bloque de entrada	65
4.2.3.1.1	Bloque LVDS.....	65
4.2.3.1.2	FIFO LVDS	66
4.2.3.1.3	Concatenador.....	66
4.2.3.1.4	Cola de espera del multiplexor.....	67
4.2.3.2	Sistema de arbitraje y Multiplexado.....	67
4.2.3.2.2	Multiplexor	68
4.2.4	Control	68
4.2.4.1	Controlador de memoria	69
4.2.5	Proceso de lectura y envío de muestras	70
4.2.5.1	Segmentación y envío de muestras	71
	Capítulo 5: Conclusiones y trabajo futuro.....	72
	Apéndices.....	74
	Apéndice A: Comunicación UART [24]	74
	Apéndice B: Comunicación SPI	76
	Apéndice C: Nociones básicas de comunicación LVDS.....	77
	Apéndice D: API preliminar del sistema (recepción).....	79
	Bibliografía.....	85

Introducción

Los equipos de ultrasonido tienen una gran relevancia hoy en día, dada la diversidad de sus aplicaciones. Éstas abarcan el estudio de las propiedades en materiales, detección de fallas y diagnósticos en medicina, por mencionar algunas.

En el laboratorio de acústica de la Facultad de Ciencias se realizan diversos desarrollos en las áreas de ensayos no destructivos y medicina, siendo la primera la de mayor avance en la actualidad. Las plataformas de desarrollo recientes se basa en equipos de marcas como *Ultratek* o *Handy Scope*, cuyo hardware es inalterable, de modo que no es posible adaptarlos a las necesidades particulares de cada usuario, reduciendo la gama de posibles aplicaciones y desarrollo. Dichas condiciones hacen necesario el diseño de la plataforma de desarrollo aquí presentada.

A lo largo del presente trabajo se exponen los resultados del diseño de la etapa de recepción de un sistema de adquisición de señales de ultrasonido, el cual realiza un conjunto de secuencias de adquisición definidas por el usuario que son almacenadas y transmitidas a una PC en tiempo diferido para su posterior procesamiento y análisis.

La arquitectura aquí descrita se centra en el manejo de un conjunto de elementos básicos, los cuales consisten en una etapa de acondicionamiento analógico y de conversión A/D (ambas integradas en un solo circuito) precedidas de un sistema deserializador y un sistema de control basados en una arquitectura FPGA, además de sistemas de configuración, comunicación y almacenamiento, entre otros.

El diseño de circuitos de instrumentación digital basados en FPGA ofrece ventajas significativas sobre otras tecnologías similares como los ASIC en costo y flexibilidad, por ejemplo, la capacidad de expansión futura. El sistema además de ser utilizado para realizar pruebas en el laboratorio, también será modificado para formar parte de un sistema de imágenes tomográficas para ensayos no destructivos.

El escrito se encuentra dividido en cinco capítulos. El capítulo 1 proporciona una descripción general de los conceptos fundamentales utilizados a lo largo del desarrollo del dispositivo, tales como que es el ultrasonido y sus aplicaciones, señales analógicas y digitales, conversión analógico-digital, sistemas digitales y su diseño y, finalmente, una breve descripción de algunos dispositivos, tales como memorias SDRAM y FPGA.

En el segundo capítulo, se abordan las características generales con las que debe cumplir el sistema, además de la descripción de la arquitectura implementada a nivel conceptual, haciendo énfasis en los elementos principales y sus características.

Posteriormente el tercer capítulo se enfoca en una descripción detallada del desarrollo, implementación y funcionamiento de la arquitectura implementada en hardware mediante VHDL,

incluyendo todos los módulos de comunicaciones, almacenamiento, control, etc. Además se aborda el desarrollo del software de control, configuración y comunicaciones.

En el capítulo 4 se presentan los resultados obtenidos a lo largo del desarrollo del sistema y se muestran ejemplos de aplicación. Por último, en el quinto capítulo se exponen las conclusiones acerca del desarrollo del sistema.

Capítulo 1: Marco teórico

Antes de abordar la resolución del problema se presenta una breve explicación de los conceptos más importantes empleados en el desarrollo del sistema de recepción de señales de ultrasonido, así como algunos ejemplos de aplicación que justifican la elaboración del dispositivo.

1.1 Introducción al ultrasonido

Las aplicaciones del ultrasonido hoy en día son cuantiosas debido a su bajo costo, gran versatilidad, poca invasividad, buena resolución, etc. El sistema aquí presentado conforma únicamente una parte de, a su vez, un sistema de adquisición de señales, siendo el pulsador su complemento.

1.1.1 ¿Qué es el ultrasonido?

Este término se refiere al conjunto de señales acústicas cuya frecuencia se encuentra por encima de las perceptibles por el sistema de audición humano (20[Hz] hasta 20[KHz]). Conforme se incrementa la frecuencia de la señal, su longitud de onda se reduce. Esto puede apreciarse en la tabla 1.1.

Frecuencia [Hz]	Longitud de onda
100	15.4 [m]
1'000	1.54 [m]
10'000	15.4 [cm]
100'000	1.54 [cm]
1'000'000 =1 [MHz]	1.54 [mm]
10'000'000 =10 [MHz]	0.15 [mm]

Tabla 1.1 Tabla en la que se ejemplifica la reducción de longitud de onda de una onda comparada con la frecuencia de la misma calculada para tejidos suaves [1].

A continuación se presenta la expresión que relaciona a la frecuencia (f) con la longitud (λ) de onda y la velocidad de propagación (v) (asumiendo que ésta es la misma a lo largo de todo el medio):

$$v = \lambda f$$

La longitud de onda de la señal afecta en gran medida la resolución del dispositivo, ya que una mayor frecuencia de operación provee una mejor resolución axial, permitiendo la visualización de objetos más pequeños. Por otro lado, el incremento en la frecuencia de onda tiene como desventaja una mayor atenuación, con la consecuente reducción en la capacidad de penetración del haz.

Para este proyecto se emplearán principalmente transductores a dos frecuencias distintas, las cuales son de 500[kHz] y 1[MHz] para realizar las pruebas de operación. Tales transductores están disponibles en el laboratorio al momento del desarrollo de este proyecto.

1.1.1.1 Formación de ecos

Los ecos son reflexiones que se forman por diferencias de impedancia acústica (Z_A) entre dos medios, dando como resultado una onda reflejada y otra transmitida. Éstas son dependientes de dos factores principales: la rigidez (k) y la densidad del medio (ρ). La relación entre rigidez, densidad e impedancia acústica del medio puede reducirse a las siguientes expresiones:

$$Z_A = \sqrt{\rho k}$$

$$Z_A = \rho v$$

La figura 1.1 ejemplifica la reflexión y transmisión existente durante la propagación de una onda entre dos medios con distinta impedancia acústica.

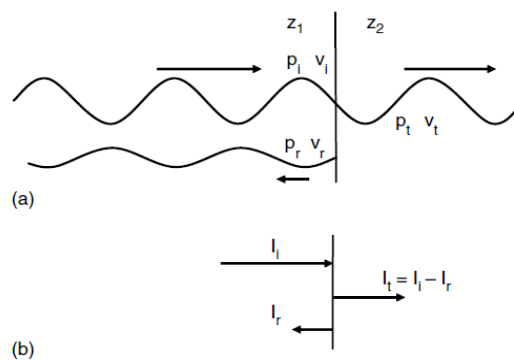


Figura 1.1. (a) Muestra la presión reflejada y transmitida en la interfase de dos medios. Asimismo ilustra la relación entre intensidad incidente, reflejada y transmitida [2].

La reflexión de la señal puede llevarse a cabo en cualquiera de los modos presentados a continuación:

- **Especular:** Se presenta en objetivos grandes y libres de irregularidades. En este tipo de superficie el ángulo de incidencia es igual al reflejado (ver fig. 1.2 (a)).
- **Dispersa (scattering):** Ocurre cuando el objeto en el que incide la onda es pequeño en comparación con la longitud de onda. La onda incidente es dispersada en un gran ángulo, como si se tratase de una fuente puntual (ver fig. 1.2 (b)).
- **Difusa:** Cuando la superficie tiene irregularidades comparables con la longitud de onda de la señal (ver figura 1.2 (c)), hablamos de una reflexión de este tipo.

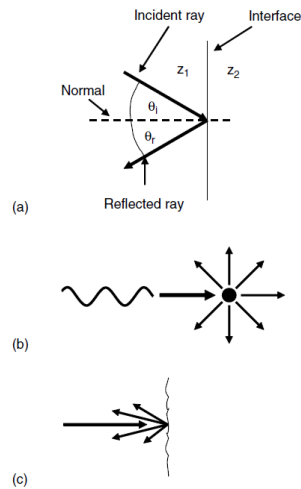


Figura 1.2 Tipos posibles de reflexión de ondas: (a) Especular, (b) Dispersa, (c) Difusa [2].

1.1.1.2 Atenuación

Se manifiesta como una disminución en la intensidad de la señal durante su propagación. Se debe principalmente al fenómeno denominado absorción, en el cual la energía de la señal se convierte en un incremento de temperatura en el medio. Cabe mencionar que este efecto depende de la frecuencia y que el decaimiento de la señal se presenta en forma exponencial a lo largo de su trayecto.

1.1.2 Configuraciones comunes para ensayos con ultrasonido

1.1.2.1 Pulso-eco

Se trata de la configuración más empleada en la medicina, debido a que, en contraste con la técnica de transmisión, en ésta solo es necesario tener acceso a una cara del objeto de estudio.

El sistema consta de un sistema pulsador, un receptor y un transductor común a ambos sistemas y su funcionamiento se basa en los ecos producidos por cambios de medio a lo largo de la trayectoria de propagación de la onda –descrito previamente en 1.1.1– como se muestra en la figura 1.3.

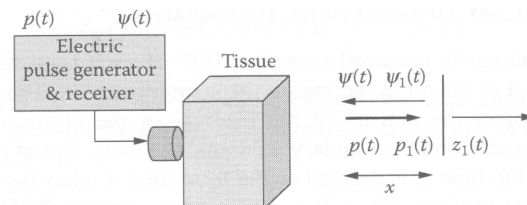


Figura 1.3 Configuración general del sistema de adquisición de señales por configuración pulso-eco [1].

1.1.2.2 Transmisión

Es un método alternativo al de pulso-eco. En él se emplea la atenuación de la señal para obtener información acerca del medio de propagación. En esta configuración es necesario el uso de dos transductores: uno para la emisión de la señal y el otro para su recepción. La configuración básica se muestra en la fig. 1.4.

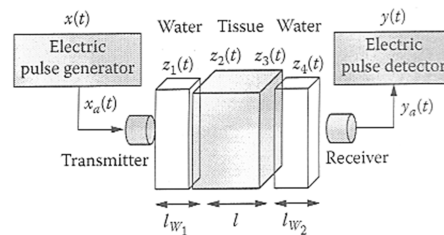


Figura 1.4 Configuración general del sistema de adquisición de señales por configuración transmisión/atenuación [1].

La principal ventaja de la técnica de transmisión sobre la técnica de pulso-eco es que la señal transmitida sólo atraviesa la trayectoria de estudio una vez hasta llegar al sistema de detección, lo que se traduce en una mayor sensibilidad y profundidad de penetración. Esto, a su vez, permite la aplicación de señales con una menor potencia.

1.1.3 Aplicaciones del ultrasonido en ensayos no destructivos

Se trata de sistemas de determinación de fallas y defectos en materiales sin causarles daño. Este tipo de sistemas es de gran relevancia en la industria, ya que permite conocer la integridad de diversos materiales, la evaluación de riesgos, la detección de fisuras, entre otras aplicaciones.

Entre los distintos sistemas con esta función se encuentran: inspección visual, rayos X, corrientes de Foucault, inspección magnética, penetración de líquidos y ultrasonido. En este último caso, la forma de detectar fallas o grietas se puede llevar a cabo de varias formas, que incluyen desde el monitoreo de los ecos generados por el paso de la señal de ultrasonido (ver fig. 1.5) y la atenuación en la transmisión de la señal, hasta métodos de generación de imágenes tales como tomografía.

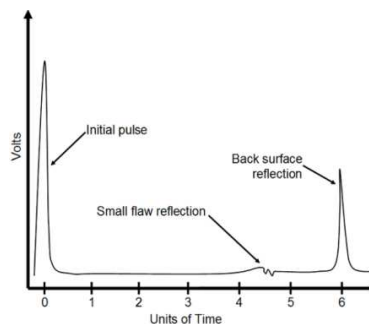


Figura 1.5 Ejemplo de ecografía de un material para detección de grietas mediante ensayos no destructivos en el que se muestra la detección de una falla [3].

1.1.4 Aplicaciones del ultrasonido en medicina

Las aplicaciones médicas del ultrasonido son cuantiosas e importantes, principalmente por la gran cantidad de información que aportan acerca de los tejidos blandos, lo que conlleva importantes ventajas con respecto a otros equipos de imagenología comunes como es el caso de los rayos X.

Dentro de la imagenología médica, algunas áreas donde hay aplicaciones comunes del ultrasonido son:

- Cardiología
- Ginecología
- Cirugía
- Urología
- Oftalmología
- Etc.

Las frecuencias de operación comunes para las aplicaciones médicas se encuentran entre 1 y 100[MHz] [1], por lo cual en imágenes es posible obtener resoluciones axiales del orden de milímetros. Mientras que la resolución radial depende del diámetro del haz de ultrasonido.

Comúnmente se recurre a la formación de imágenes como son las señales en modo A y las imágenes en modo B (formadas a partir de un conjunto de señales en modo A), como se muestra en las figuras 1.6 (a) y (b), respectivamente.

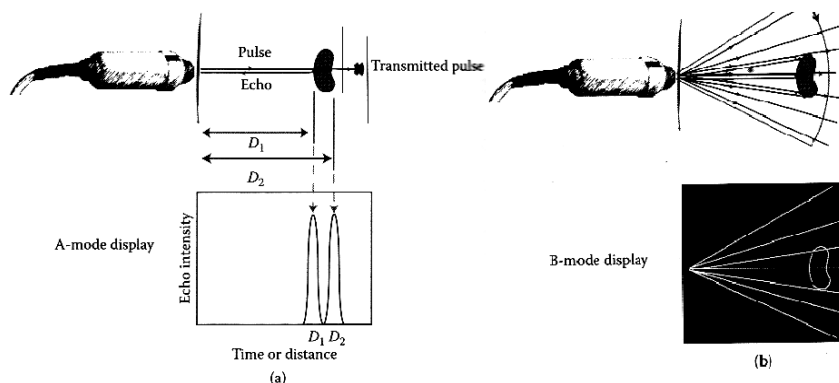


Figura 1.6 (a) Muestra una ecografía en modo A, con la cual es posible detectar las distintas interfaces entre tejidos a lo largo de la trayectoria de propagación de la señal y también calcular distancias mediante el tiempo de vuelo de la señal. Por otra parte, (b) ejemplifica la formación de una imagen en modo B a partir de un conjunto de señales en modo A [2].

1.1.5 Transductores ultrasónicos

Se trata de transductores electroacústicos cuyo ancho de banda comprende a las ondas de ultrasonido. Poseen varias características particulares dependiendo de su construcción y mecanismos de transducción.

En el laboratorio de acústica se dispone de transductores piezoeléctricos de inmersión similares a los mostrados en la figura 1.7.



Figura 1.7 Transductores de inmersión usados en ensayos no destructivos.

Los parámetros importantes en un transductor ultrasónico no enfocado se resumen en la tabla 1.2.

Características importantes	Definición
Campo cercano del transductor	Región comprendida entre la cara del transductor y el punto sobre el eje normal al transductor en el que se encuentra el punto de máxima presión emitida.
Frecuencia pico	Frecuencia de máxima amplitud para una señal de pulso-eco.
Ancho de banda	Rango de frecuencias que tienen una pérdida de potencia menor a 6dB que la frecuencia central (para configuración pulso-eco).
Frecuencia central	Media aritmética de las frecuencias que componen el BW.
Duración de pulso	Tiempo de duración de la respuesta a una excitación dada.
Impedancia de entrada	Impedancia eléctrica presentada por el transductor.
Sensibilidad	Es una función que relaciona el voltaje con el que el transductor es excitado con respecto al voltaje recibido (para configuración pulso-eco). Por ello es una función del medio en el cual es realizado el ensayo.
Relación señal a ruido	Relación que existe entre la señal recibida y el ruido ambiental.

Tabla 1.2 Características importantes de los transductores ultrasónicos.

1.2 Señales

1.2.1 Señales analógicas

Son funciones cuyo valor de amplitud es continuo y está definido para cualquier tiempo dado. Algunos ejemplos de este tipo de señales son voltaje, corriente eléctrica, señales de audio y fuerzas o señales bioeléctricas, tales como las electrocardiográficas.

En este proyecto las señales continuas a trabajar son las proporcionadas por los transductores de ultrasonido.

1.2.2 Señales digitales

Se trata de señales $x[n]$ definidas únicamente como instantes discretos de tiempo ($t = t_n$ tal que $n \in \mathbb{Z}$), cuyos valores de amplitud pertenecen a un conjunto finito de elementos ($\{a_1, a_2, a_3 \dots a_N\}$ tal que $x(t_n) = x[n] = a_i$ donde $1 \leq i \leq N$).

Una señal binaria es aquella cuyos valores de amplitud únicamente pueden tomar valores de $x[n] \in \{0,1\}$ para $n \in \mathbb{Z}$. Éste tipo de señales es ampliamente utilizado en los sistemas computacionales. Más adelante se retomará este tema.

1.2.3 Conversión A/D

Para poder obtener y analizar las muestras de voltaje provenientes de los transductores es necesaria una etapa de conversión analógico-digital. En la implementación del dispositivo esta etapa se encuentra embebida dentro del dispositivo denominado como *Analog Front End* del cual se hablará más adelante. Los procesos involucrados en esta etapa son los siguientes [4] (figura 1.8):

- **Muestreo.** En esta etapa se convierte la señal continua en el tiempo en una señal discreta mediante la toma de muestras a intervalos regulares. Dicho intervalo es conocido como periodo de muestreo.
- **Cuantización.** La amplitud de la señal se convierte en un conjunto finito de valores truncando o redondeando dichos valores, lo cual introduce el llamado error de cuantización.
- **Codificación.** Los valores de amplitud de la señal cuantizada se codifican en una cadena binaria de n bits.

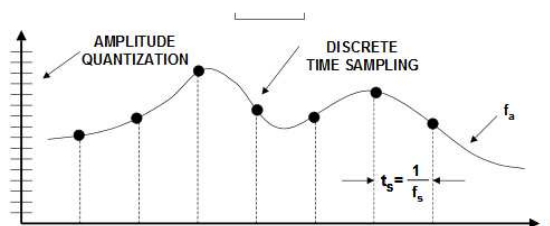


Figura 1.8 Representación del proceso de muestreo y cuantización. El eje t representa los instantes de muestreo. En el eje ordenado se muestran los diversos valores con los que se puede cuantizar la señal muestreada [5].

1.3 Sistemas digitales

Se trata de sistemas encargados de manipular elementos discretos de información representada en forma binaria, cada uno de esos sistemas se encuentra constituido por un conjunto de módulos digitales. En éstos las señales sólo emplean dos valores discretos de amplitud, es decir, son sistemas de representación binarios.

1.3.1 Circuitos combinacionales

Son sistemas digitales cuya salida depende únicamente del estado actual, que está determinado por la combinación lógica de las entradas. Este tipo de sistemas comúnmente se encuentran como dispositivos de mediana escala de integración en forma de compuertas lógicas o VLSI,¹ como celdas estándar. En el diseño de la arquitectura de este proyecto se emplean algunos circuitos combinacionales, tales como multiplexores, y operaciones básicas con compuertas lógicas, entre otros.

1.3.2 Circuitos secuenciales

Se trata de circuitos que además de circuitos combinacionales integran elementos de memoria en una trayectoria de realimentación, de este modo el valor binario de la salida depende tanto de las entradas como de la información almacenada en los elementos de memoria del circuito (*Flip-Flops latches*).

Los circuitos secuenciales **síncronos** (vea figura 1.9) son aquellos que dependen de un dispositivo llamado generador de reloj que permite que puedan ejecutar sus tareas al mismo tiempo. Estos circuitos son los más empleados en la práctica debido a su gran estabilidad y sencillez en su diseño. El arreglo lógico programable empleado (FPGA) pertenece a esta clasificación.

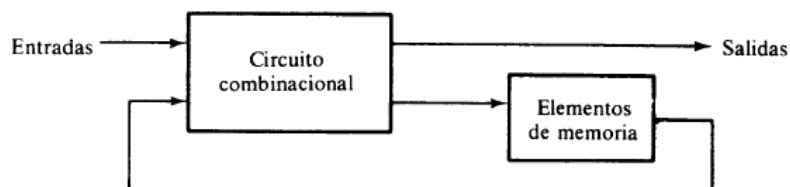


Figura 1.9 Esquema general de un circuito secuencial síncrono [6].

Los circuitos secuenciales **asíncronos** no dependen de los pulsos de un reloj para ejecutar sus funciones lógicas y los cambios de estado se realizan únicamente cuando hay un cambio en las entradas. Su temporización es más compleja que la de los circuitos síncronos y para su diseño es necesaria una mayor atención con el fin de evitar estados inestables en el sistema, por lo que generalmente se encuentran restringidos a circuitos pequeños. Las ventajas de este tipo de sistemas son el costo y su capacidad de operar a altas velocidades.

¹ VLSI (Very High Integration Scale, en inglés), "Escala de integración muy alta".

1.3.3 Registros

Un registro comprende un conjunto de flip-flops, cada uno de los cuales almacena un bit de información. Estos dispositivos, además, pueden tener un conjunto de compuertas que permitan realizar algunas operaciones fundamentales, por ejemplo: cargar datos, contar, sumar, restar y realizar corrimientos.

1.3.3.1 Transferencia de registros (RTL)

La mayoría de los circuitos digitales se pueden dividir en dos partes: la ruta de datos, la cual lleva a cabo procesamiento de datos, y la unidad de control, que determina la secuencia de esas operaciones.

Las operaciones de transferencia de registros de los sistemas digitales se encuentran especificadas por el conjunto de registros del sistema, las operaciones realizadas sobre los datos almacenados y el control que supervisa la secuencia de operaciones en el sistema. En la tabla 1.3 se muestran los operadores RTL en VHDL, el cual es un lenguaje de descripción de hardware. Sobre este último se ahondará más adelante.

Operación	VHDL
Asignación combinacional	<= (concurrente)
Transferencia de registros	<= (concurrente)
Adición	+
Substracción	-
AND	And
OR	Or
XOR	Xor
NOT	Not
Corrimiento a la derecha	Sll
Corrimiento a la izquierda	Srl
Vectores/Registros	A(X downto Y)
Concatenación	&

Tabla 1.3 Operaciones a nivel RTL [6].

1.4 Máquinas de estados finitos

Se trata de un modelo de circuito secuencial en el que el sistema pasa por un número finito de estados. Dichas transiciones dependen del estado interno y/o una entrada externa. Generalmente su tarea consiste en controlar un sistema digital mayor enviando señales del estado del sistema o de sus transiciones. A este tipo de circuito se le conoce como máquina de estado con ruta de datos.

Dependiendo de los elementos que componen la lógica de salida, se reconocen dos máquinas de estado principales:

- **Máquina Mealy:** en ésta las salidas dependen tanto de las entradas como del estado presente.
- **Máquina Moore:** las salidas dependen únicamente del estado presente.

1.5 Dispositivos lógico-programables: El FPGA

Se trata de un conjunto de arreglos lógicos y elementos de ruteo (matrices de interruptores), los cuales son capaces de implementar funciones booleanas combinacionales y/o secuenciales. Además de los elementos antes mencionados, los circuitos modernos contienen bloques que realizan funciones especializadas de las que se hablará más adelante.

Los elementos fundamentales que soportan el funcionamiento del dispositivo son las denominadas LUT (*Look Up Table*). Éstas pueden ser vistas como tablas de verdad. En su arquitectura más básica, el dispositivo está formado por una memoria de N elementos y un multiplexor de N a 1 señales (ver figura 1.10). Asimismo, éste es capaz de implementar cualquier función con N entradas simplemente mediante la programación de los bloques de memoria SRAM (también hay algunos dispositivos basados en FLASH o fusibles). El dispositivo elegido es un *Spartan 6*, que tiene celdas LUT basadas en SRAM con entradas de 6 bits; de esta manera, ofrece la capacidad de emplear lógica más compleja que las generaciones previas (arquitecturas basadas en LUT de 4 bits) y reduce los retardos por interconexión de celdas, pero a costa de tener LUT más lentas.

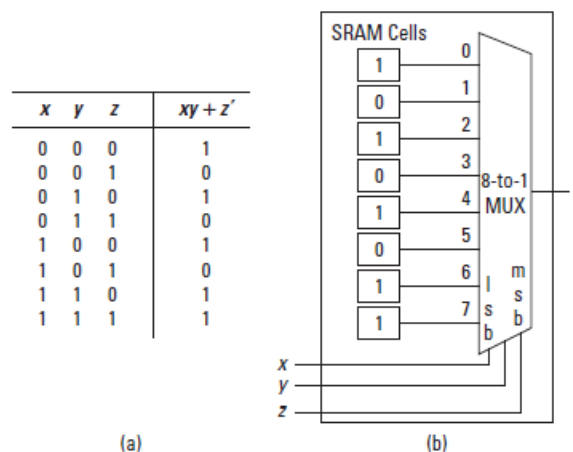


Figura 1.10 (b) Estructura simplificada de una LUT. En ésta se muestra la implementación de una función lógica combinacional representada en (a) [7].

Además de los elementos de la LUT son necesarios elementos adicionales para poder llevar a cabo circuitos secuenciales, de tal forma que éstas tienen un flip-flop D asociado. Como resultado, este par de elementos conforma la estructura básica de una celda lógica, que es el bloque de construcción más pequeño del dispositivo (figura 1.11). En el caso de los dispositivos de Xilinx, el siguiente elemento de construcción se conforma por 4 celdas lógicas agrupadas y se le llama *slice*.

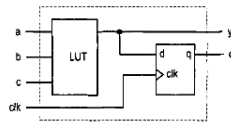


Figura 1.11 Estructura simplificada de una celda lógica, la cual es el elemento básico para implementar funciones en el dispositivo [8]

La interconexión de las celdas lógicas es muy importante para alcanzar bajos retardos de propagación. Existen diversas metodologías para lograr dicha tarea, las cuales tienen diferencias significativas y cambian en gran medida dependiendo del fabricante del dispositivo. Una estructura general se aprecia en la figura 1.12. Cabe mencionar que ésta no coincide con los dispositivos modernos; sin embargo, es una buena aproximación para entender el funcionamiento del dispositivo.

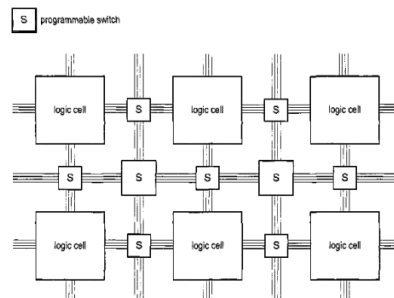


Figura 1.12 Estructura simplificada de un FPGA con arquitectura tipo isla. [8]

Además de las LUT y las matrices de interconexión los FPGA se componen de bloques de entrada/salida (con soporte para múltiples estándares) y módulos con funciones de propósito específico como: comunicaciones, memoria RAM embebida, control de memorias externas, procesadores, bloques orientados a procesamiento digital de señales y circuitería de manipulación para relojes, tales como los DCM (*Digital Clock Managers*) o PLL (*Phased Locked Loop*).

Los FPGA abarcan un gran rango de aplicaciones; debido a su naturaleza permite implementar prácticamente cualquier circuito digital; las tareas en las que se destaca son aquellas en las que es necesario manejar grandes anchos de banda, ya que los circuitos digitales descritos en él pueden ser totalmente independientes entre sí, permitiendo realizar varias tareas en forma paralela.

Considerando las características mencionadas, se decidió utilizar tecnología FPGA para esta aplicación, porque, especialmente, permite manejar grandes cantidades de información, además de poseer numerosas posibilidades de configuración para puertos de entrada/salida – para el control de los dispositivos del sistema es necesaria una gran cantidad de señales con distintos estándares– y, por último, su capacidad de reconfigurarse posibilita la adaptación del dispositivo para aplicaciones específicas, lo cual permite un desarrollo futuro sobre la misma plataforma.

La tarea que desempeñará este dispositivo es la de ayudar a realizar el almacenamiento de información, sirviendo como puente entre las muestras del conversor analógico-digital y una memoria RAM. Por otro lado en conjunto con un puente UART-USB, llevará acabo la transmisión de la información contenida en buffer hacia la PC.

1.6 Lenguajes de descripción de hardware (HDL)

Se trata de lenguajes de alto nivel que permiten describir el circuito a implementar en el hardware. Éstos se convierten en las listas que contienen las uniones para implementar circuitos internamente mediante herramientas de síntesis (como XST incluida en la plataforma *ISE Design Suite* de Xilinx). Además, también permiten realizar simulaciones para verificar el comportamiento del diseño. Existen diversos lenguajes de descripción de hardware. Para esta aplicación en particular se escogió VHDL.

1.6.1 VHDL

VHDL significa lenguaje de descripción de hardware VHSIC² (*VHSIC Hardware Description Language*, en inglés). En este lenguaje hay dos maneras principales de realizar las descripciones: la estructural, en la que el circuito se encuentra descrito en términos de otras entidades (ya sean compuertas u otros bloques más elaborados), y la de comportamiento (*behavioral*), en la que se describen las relaciones de las salidas con las entradas.

1.7 Buffer FIFO

FIFO es el acrónimo de la estructura de datos denominada *First In First Out*, lo cual significa que se realiza un almacenamiento de datos donde el primer dato entrante será, asimismo, el primero que salga de la estructura cuando éstos sean leídos, como se muestra en la figura 1.12.

Esta estructura es común en los diseños de procesamiento y manipulación de datos en FPGA, ya que permite realizar un almacenamiento temporal en cadenas de datos, además de ser útil en tareas como sincronización de dominios de reloj, entre otras aplicaciones.

² VHSIC significa “Circuito Integrado de Muy de Alta Velocidad” (*Very High Speed Integrated Circuit*, en inglés).

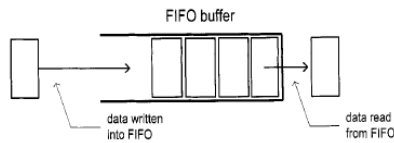


Figura 1.12 Representación del funcionamiento de un buffer FIFO, en el cual se muestra la ruta de datos de entrada-salida [8]

1.8 Memoria SDRAM

La Memoria Síncrona de Acceso Aleatorio (*Synchronous Dynamic Random Access Memory*, por sus siglas en inglés) es un subconjunto de las memorias RAM, las cuales engloban diversas tecnologías. Una de las características más importantes de este tipo de tecnología es que su operación, a diferencia de las tecnologías que le precedieron, es síncrona con respecto al reloj del sistema que la opera.

Arquitectura interna

Su arquitectura se encuentra basada en un conjunto de celdas que conforman arreglos matriciales. Cada celda es capaz de almacenar un bit de información y se encuentra constituida por un capacitor transistor MOSFET y uno que controla la carga del primero (ver figura 1.13). Esta estructura tiene la gran desventaja de que el capacitor se descarga con el paso del tiempo, por lo cual debe ser refrescado (recargado) periódicamente. Por otro lado, la simplicidad de esta estructura es bastante útil, ya que ha posibilitado la generación de memorias de alta densidad empleadas actualmente.

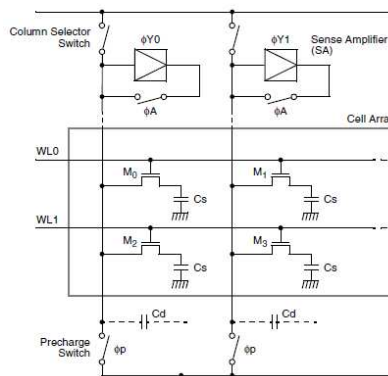


Figura 1.13 Estructura simplificada de una celda de memoria de una memoria SDRAM. WL0 y WL1 representan dos filas adyacentes, cada una de ellas es una palabra de 2 bit [9].

Además de las celdas de memoria se tiene un conjunto de circuitos de direccionamiento (decodificadores y multiplexores), amplificadores de sentido (amplificadores de prefetch), buffers de triestado y latches de salida/entrada. La estructura completa se muestra en la figura 1.14. Cabe mencionar que en la memoria mostrada sólo se podría acceder a un bit a la vez; las celdas de memoria actuales contienen múltiples capacitores, de modo que a cada localidad le corresponden palabras completas en vez de bits únicos.

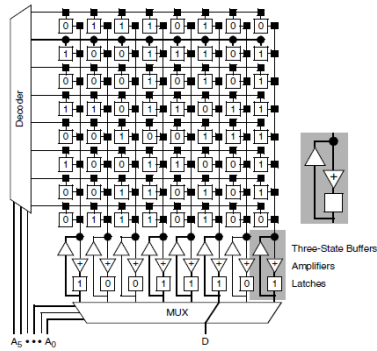


Figura 1.14 Estructura simplificada de un arreglo de memoria SDRAM de 64 bits [9].

En el diseño del dispositivo se considera este tipo de almacenamiento, dado que, como se mostrará más adelante, el protocolo USB en su modalidad full-speed es insuficiente para enviar la información capturada en tiempo real, por lo que se requiere un almacenamiento temporal para no perder dicha información.

Capítulo 2: Especificaciones del sistema y selección de componentes

Los sistemas de ultrasonido se encuentran divididos en tres etapas principales: la de transmisión, la de recepción y procesamiento. Cada una de ellas enfrenta retos particulares. A continuación se tratan las características generales que debe cumplir la etapa de recepción, una representación general de ésta se muestra en la figura 2.1.

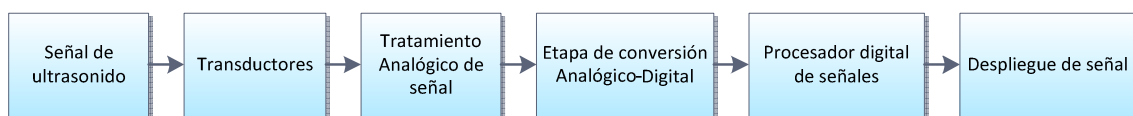


Figura 2.1 Diagrama de bloques de la etapa de recepción de señales en un sistema de ultrasonido.

En el Laboratorio de Acústica de la Facultad de Ciencias se requiere un sistema de emisión-recepción de señales de ultrasonido, que debe ser controlado mediante una interfaz computacional. El fin de la creación del mismo es tener un equipo que pueda ser empleado para fines de investigación en el área de ensayos no destructivos y posiblemente en aplicaciones que involucren tejidos biológicos.

Adicionalmente el sistema debe ser lo suficientemente flexible para poder ser reutilizado en la creación de un sistema tomográfico de imágenes empleado en las mismas áreas de investigación que abarca el laboratorio.

En este escrito se propone una arquitectura desarrollada para la etapa de recepción del sistema de adquisición antes mencionado. Cabe mencionar que el bloque de emisión se encuentra en desarrollo en este momento.

La etapa de recepción pretende entregar al usuario los datos adquiridos, para que éste por medio de herramientas computacionales pueda llevar a cabo el procesamiento digital de las señales. Los requisitos fijados para el sistema deben ser los siguientes:

- Interfaz computacional de fácil uso
- Capacidad para utilizar diversos tipos de transductores
- Captura de hasta 8 señales en forma simultánea
- Registro de múltiples eventos (pulsos)
- Ajuste de parámetros básicos del sistema, tales como: ganancia, etapas de filtrado analógico, número de canales activos, duración de las muestras y sincronía con la etapa transmisora

2.1 Sistemas computacionales de adquisición de datos y grabadores digitales de formas de onda

Los sistemas grabadores de formas de onda (*Waveform Recorders*) son los únicos útiles para registrar señales cuyas frecuencias son superiores a 80[KHz] [10], además de proveer la posibilidad de procesamiento computarizado y sincronización con procesos externos [11] (que es el pulsador en este caso). Los bloques básicos de construcción de un sistema de este tipo son:

- Multiplexado (opcional)
- Amplificación
- Etapa de conversión A/D
- Interfaz computacional o salidas analógicas

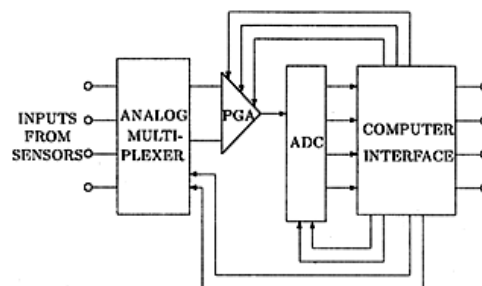


Figura.2.1 Representación general de un sistema computacional de adquisición de datos.

2.2 Módulo de acondicionamiento analógico y conversión analógica digital (MAACAD)

El acondicionamiento analógico de la señal es de gran relevancia, ya que permite realizar una adquisición óptima (baja en ruido y distorsión). Las etapas de acondicionamiento analógico a cubrir son las de amplificación y filtrado. Una limitante adicional, debido a la naturaleza de señal mezclada del circuito, es que éste debe ser lo más inmune posible al ruido digital [12].

2.2.1 Tratamiento analógico de señales

2.2.1.1 Amplificación

Debido a que las señales en estudio son de baja amplitud es necesaria su amplificación antes de la etapa de conversión analógica digital para disminuir el error de cuantización. La amplificación debe tener bajo ruido y baja distorsión, por lo que es conveniente su realización en dos etapas: la primera corresponde a un amplificador de bajo ruido, que generalmente es la primera etapa en las cadenas de acondicionamiento de señales en sistemas de instrumentación. Después se agrega una etapa adicional de amplificación que actúe previamente a la etapa de conversión analógico-digital.

2.2.1.1.1 Amplificador de bajo ruido

Esta primera etapa de amplificación responde a la necesidad de amplificar las señales provenientes de los sensores de ultrasonido (atenuadas debido a la interacción con el medio y acentuadas con la profundidad de penetración). Un factor adicional a considerar es la gran variabilidad en la amplitud (causada por los cambios de densidad, que pueden ser suaves o abruptos).

Los parámetros antes descritos hacen necesaria una amplificación con un alto SNR³ para que el ruido generado por etapas posteriores tenga un impacto pequeño en la cantidad de ruido del sistema [13], además debe tener un amplio rango dinámico⁴ que ayude a evitar saturación cuando se tengan ecos de mayor amplitud.

2.2.1.1.2 Amplificador de ganancia programable (PGA)

Los amplificadores de ganancia programable son comunes en los sistemas computacionales de adquisición de datos. Éstos son controlados digitalmente. La ventaja de incluirlos en la cadena de acondicionamiento radica en que habilitan el sistema para recibir señales cuyas órdenes de magnitud puedan variar significativamente (mV a V, por ejemplo), evitando la necesidad de incluir circuitos de conversión A/D de gran resolución para disminuir el ruido de cuantización en señales pequeñas que incrementarían el costo del sistema y disminuirían su velocidad de operación [14].

2.2.1.2 Filtrado

En esta etapa se permite pasar un conjunto de frecuencias de interés (banda de paso) y se suprimen las que no lo son (banda de supresión). Este fenómeno puede apreciarse en la figura 2.2.

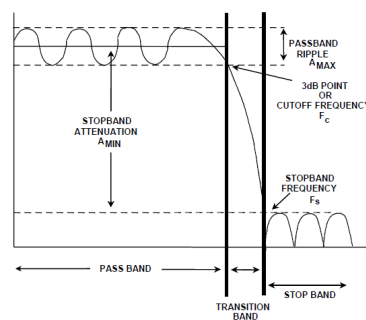


Figura 2.2 Parámetros importantes de un filtro. En este caso, un filtro pasa bajas con respuesta tipo Chebyshev. La región de la izquierda corresponde a la banda de paso, en la cual hay atenuación mínima; posteriormente, al centro se aprecia la banda de transición, en la cual aumenta rápidamente la atenuación, y finalmente, a la derecha se observa la banda de supresión, en la cual la atenuación es máxima [15].

³ SNR (*Signal Noise Ratio*, en inglés) significa Relación Señal a Ruido, la cual establece la relación de potencia que tiene la señal de interés con respecto al ruido ambiental.

⁴ Por rango dinámico se entiende la relación entre la mínima señal de entrada que produce una salida útil (determinada por la cantidad de ruido y el CMRR) y el valor máximo de amplitud que puede tenerse en la entrada sin saturar el amplificador.

En esta aplicación es necesario un conjunto de filtros que permitan realizar dos tareas fundamentales:

- Limitar en banda la señal de entrada para evitar aliasing⁵ en la etapa de conversión A/D mediante un filtro pasa bajas.
- Suprimir ruido de baja frecuencia para incrementar SNR.

2.2.2 Conversión A/D

Como se mencionó en la sección 1.2.3, en la etapa de conversión analógico-digital las señales analógicas provenientes de los circuitos digitales se vuelven una versión discreta (en tiempo y amplitud) equivalente.

Una de las características de esta etapa es que se presenta una resolución suficientemente grande para permitir un error de cuantización pequeño; generalmente, en la mayoría de los circuitos de ultrasonido, ésta varía entre 12 y 14 bit, aunque en algunos equipos anteriores se encontraba hasta en 10 bit.

Otra característica que debe poseer esta etapa es tener una frecuencia de muestreo relativamente alta, ya que los transductores de ultrasonido empleados en el laboratorio manejan frecuencias que oscilan entre 300[KHz] y 5[MHz]. Por lo que, basándose en el teorema de Nyquist, la frecuencia de muestreo mínima que debe tener el sistema está dictada por la ecuación 2.1, en la que se establece la relación entre la frecuencia de muestreo en un proceso de conversión analógico-digital y el contenido de frecuencia de la señal para evitar el fenómeno de aliasing, en el cual se pierde información importante de la señal.

$$f_s > 2 f_{max}$$

Dada la relación anterior, se concluye que teóricamente la frecuencia de muestreo mínima que debe de tener la etapa de conversión A/D debe ser de al menos 10[MSPS⁶].

2.3 Módulo de almacenamiento y transmisión (MAT)

Para transmitir la información recibida desde el conversor A/D hacia el dispositivo de despliegue de señal es necesario calcular la tasa de transferencia de datos (*bitrate*) para determinar si se dispone del ancho de banda necesario. Con el anterior fin se calcula la siguiente relación:

$$bitrate_{ADC} = (No. bits codificados en ADC)(f_{muestreo})(No. canales activos)$$

⁵ Se trata de un fenómeno en el que no es posible reconstruir la señal analógica a partir de su versión discreta debido a un muestreo con una frecuencia insuficientemente grande (descrita por Nyquist).

⁶ De las siglas en inglés de **Mega Samples Per Second**, en español se traduce como “millones de muestras por segundo”.

Si se considerara utilizar un transductor de 5[MHz], la frecuencia de muestreo al menos debería ser de 10[MHz], y si se utilizara un solo canal, el canal debería de tener capacidad para enviar hasta 100[Mb/S],⁷ lo cual no es posible dado que la especificación para USB 2.0 para USB *High Speed* es de tan solo 12[Mb/s]. Esto establece una limitante importante, ya que no es posible transmitir las señales en tiempo real.⁸

2.3.1 Etapa de recepción de datos

La etapa de recepción de datos tiene que interactuar con el formato de salida del ADC (ya sea paralelo o en serie) y convertirlo en un formato que sea apropiado para almacenamiento. Ésta debe de ser capaz de recibir la información de hasta 8 canales activos al mismo tiempo, además de poder manejar los estándares de salida del conversor, que generalmente se encuentra en formato LVDS.

Considerando los puntos mencionados anteriormente, se aprecia que la arquitectura FPGA es particularmente apropiada para esta aplicación, ya que puede ejecutar una gran cantidad de tareas en forma simultánea, tiene frecuencias de operación relativamente altas (entre decenas de MHz y 1 GHz aproximadamente) y manejan múltiples estándares de entrada/salida.

2.3.2 Etapa de almacenamiento

Debe tener una capacidad de almacenamiento mínima para varios disparos. Como tal puede fungir una memoria volátil externa o una embebida en el FPGA. Su elección se muestra a partir de hacer ciertas suposiciones acerca del funcionamiento del dispositivo. Por ejemplo, si se va a capturar una señal que se propaga en agua con una velocidad (v) de 1540 [m/s] en modo pulso-eco, se tiene el objeto reflector a una distancia (d) de 30[cm] y el conversor A/D está muestreando ($f_{muestreo}$) a 20 [MSPS] con 16 bit de resolución y se desean los 8 canales activos, lo primero que debe hacerse es calcular la cantidad de muestras:

$$\text{Resolución muestra por unidad de longitud} = \frac{f_{muestreo} [Sa]}{v [m]}$$

Para lograr abarcar toda la longitud (d) del ensayo el número de muestras necesarias es:

$$\text{Muestras necesarias para abarcar } d = (\text{Resolución muestra por unidad de longitud})(d)[Sa]$$

Lo que nos lleva a que es necesaria una memoria mínima de 608[Kbits] por cada disparo, por lo que, considerando la pequeña capacidad de almacenamiento disponible en un FPGA (usualmente se encuentra en el orden de Mb), es necesario agregar un dispositivo de almacenamiento externo temporal.

⁷ Mb/s es la notación para “Mega bits por segundo”, y expresa la tasa de transferencia de datos en un sistema digital.

⁸Un sistema en tiempo real se refiere a un sistema que es capaz de analizar una señal, antes de recibir una nueva.

2.3.3 Etapa de comunicación para el despliegue de resultados

Para alcanzar la máxima versatilidad del sistema de forma que sea posible utilizarlo en cualquier computadora es necesario utilizar un protocolo de comunicación que sea de amplio uso en las computadoras, como USB 2.0, el cual es soportado por todas las computadoras actuales y por la mayoría de los puentes de comunicación USB 2.0 (full speed) a UART⁹; asimismo, es más sencillo de implementar, lo que disminuye el tiempo de diseño. Existen también otras herramientas, como los emuladores de puertos virtuales COM.

2.3.4 Selección de los componentes

La tarea de selección de componentes es importante para una implementación que cumpla con todos los requerimientos impuestos por el personal del laboratorio, que sea reutilizable y que al mismo tiempo requiera un tiempo de diseño lo más breve posible.

2.3.4.1 Selección de etapa de acondicionamiento analógico y ADC

En los equipos modernos de ultrasonido, los componentes de la etapa de acondicionamiento analógico y de conversión analógico-digital, se encuentran integrados en un solo circuito VLSI, conocido como *Analog Front End*. Estos componentes pueden, adicionalmente, tener más funciones: pueden ser atenuadores controlados por voltaje¹⁰ y circuitos digitales de procesamiento de señales, lo que facilita el diseño y mejora el desempeño.

Con el fin de disminuir el tiempo de diseño y la complejidad del mismo se recurrió a módulos empleados para la evaluación y el desarrollo de circuitos ofrecidos por compañías de circuitos integrados, tales como Texas Instruments y Analog Devices.

Para esta tarea se escogió el circuito AFE5808A Texas Instruments, que destaca por tener uno de los mejores rendimientos en términos de ganancia, además de bajo ruido y control, en comparación con otros dispositivos de la misma familia y los fabricados por Analog Devices. Asimismo, el circuito en mención ofrece un módulo de evaluación compatible con el estándar de conexión FMC¹¹ mediante una tarjeta adaptadora; éste también se emplea en el proyecto. Su modelo es el EVM-AFE5808A.

2.3.4.2 Selección de etapa de almacenamiento y transmisión

⁹ UART (Universa Asynchronous Reciever/Transmitter) se refiere a un protocolo de comunicación asíncrona, ampliamente utilizado en comunicaciones digitales.

¹⁰ Empleados en ultrasonido para un proceso de compensación de ganancia con respecto al tiempo (también conocido como TGC o Time Gain Compensation).

¹¹ FMC (FPGA Mezzanine Card) es un estándar de interconexión creado por las organizaciones ANSI y VITA para interconexión de diversos módulos de radiofrecuencia, ADC, video, etc. Con las tarjetas de desarrollo de Xilinx, éste establece un conector específico de alta velocidad, así como la asignación de pines para las señales, relojes y voltajes de alimentación.

Para esta etapa se buscó y seleccionó una tarjeta de desarrollo que cumpliera con el estándar FMC, ya que hay una tarjeta que adapta los conectores de las tarjetas de evaluaciones de ADC de alta velocidad. La tarjeta de desarrollo adicionalmente debe poseer una memoria SDRAM que sea capaz de operar al menos a $400[\text{MT/s}]^{12}$ con un bus de 16 bits. Una característica adicional que debe cumplir es tener un puente UART-USB instalado, un emulador JTAG en la tarjeta y, por último, tener instalada una memoria FLASH u otro tipo de memoria no volátil para almacenar la configuración del dispositivo.

Dadas las especificaciones antes descritas se seleccionó una tarjeta SP605, la cual tiene instalado un FPGA modelo SPARTAN 6, la cual cumple con las características antes mencionadas, además de algunas otras que la hacen útil para desarrollos futuros en el laboratorio. Estas características se abordarán con mayor detalle en las próximas secciones.

2.3.5 Características del hardware seleccionado [16]

Como se mencionó previamente el hardware se encuentra integrado por dos elementos principales adicionales al equipo de cómputo (figuras 2.3 y 2.4): el Módulo de evaluación EVM-AFE5808A y la tarjeta de desarrollo de FPGA SP605. A continuación, se dará una breve introducción a las características de cada uno de ellos.

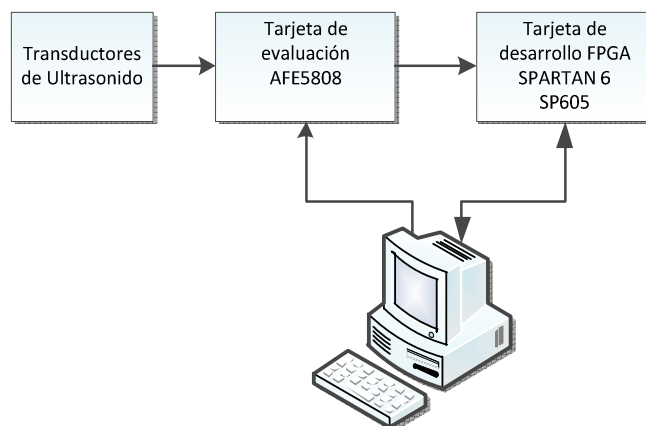


Figura 2.3 Cadena de señal de los componentes del sistema propuesto.

¹² Mega transferencias por segundo se refiere al número de palabras del ancho del bus que es capaz de enviar o recibir un dispositivo de almacenamiento.

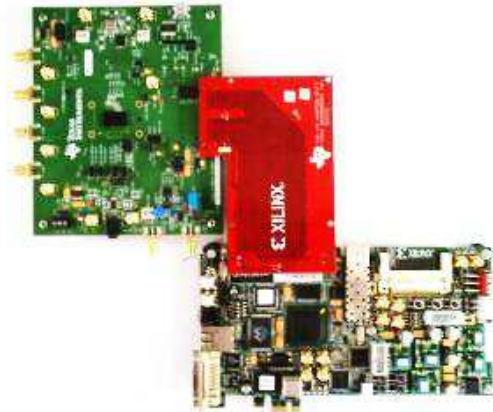


Figura 2.4 Hardware seleccionado para la implementación del sistema de adquisición de datos.

2.3.5.1 Módulo de acondicionamiento analógico y conversión analógica digital

Se trata de una tarjeta diseñada con el fin de evaluar el desempeño del circuito integrado AFE5808A (ver figura 2.5), por lo que integra un conjunto de circuitos que cumplen con diversas funciones, tales como adecuar voltajes de alimentación, establecer interfaces de comunicación, osciladores, entre otros (ver figura 2.6). Sus principales elementos se resumen a continuación:

- ❖ Reloj monopolar instalado para ADC de 40[MHz] (e), así como entrada para reloj externo.
- ❖ PLL CDCM7005 usado para generar un reloj diferencial de bajo jitter con señales CW (no utilizado en este proyecto).
- ❖ Referencias interna y externa de voltaje para ADC y atenuador controlado por voltaje.
- ❖ Circuito de interfaz analógica de 8 canales AFE5808 con salidas LVDS e interfaz de configuración SPI.
- ❖ Circuitería para comunicación USB-FIFO paralelo empleada para configuración SPI del AFE5808 y CDCM7005 mediante chip FTDI245RL.

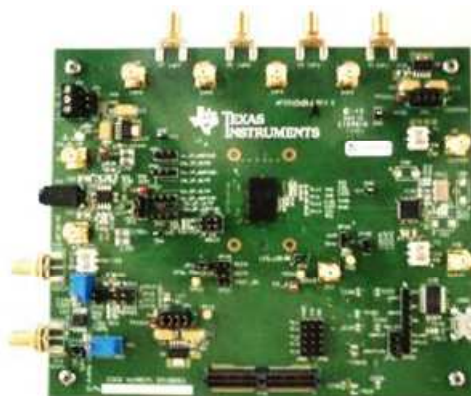


Figura 2.5 Tarjeta de evaluación del circuito AFE5808A, escogida como módulo de acondicionamiento analógico y conversión analógico-digital.

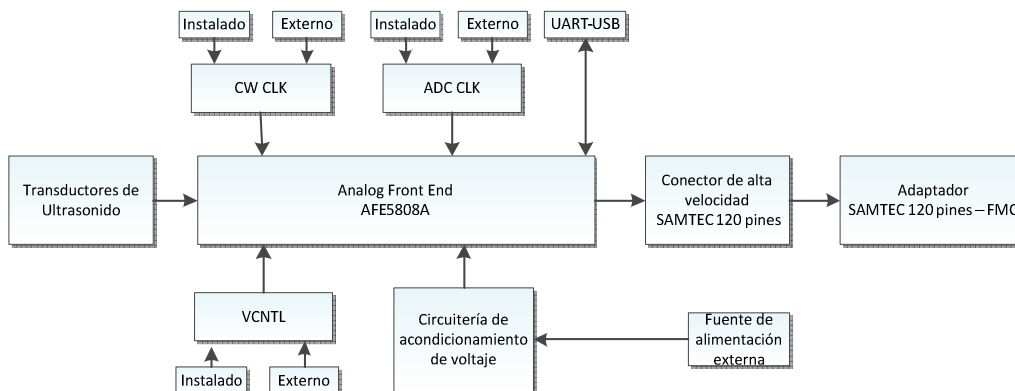


Figura 2.6 Representación a nivel de bloques del hardware del módulo de evaluación EVM-AFE5808A.

2.3.5.1.1 Analog Front End AFE5808/08A [17]

Es el núcleo del sistema de recepción. Integra las etapas de acondicionamiento analógico y conversión analógico-digital, además puede realizar algunas tareas de procesamiento digital de señales. Consta de 8 canales idénticos, cuya estructura puede apreciarse en la figura 2.7.

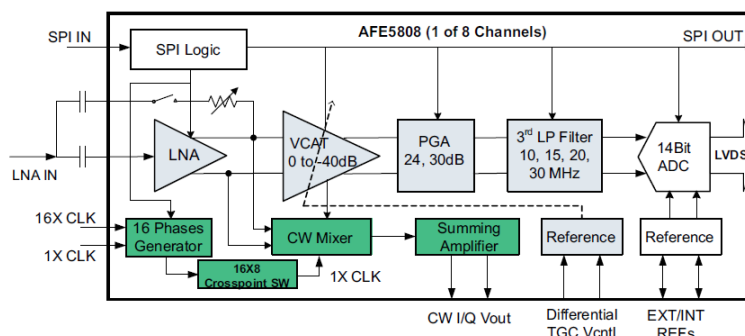


Figura 2.7 Representación por canal del AFE5808A, en ésta se muestran los principales bloques funcionales: LNA, VCAT, PGA, ADC y CW—esta última no será empleada en el desarrollo del proyecto. Asimismo se muestra la interfaz de control SPI.

A continuación se describen sus características por etapa:

❖ **LNA (Amplificador de bajo ruido):** Amplificador de bajo ruido con entrada tipo *single-ended* y salida diferencial. Las principales características de esta etapa son:

- Ganancia programable en grupo o por canal a 24/28/12dB con bajo ruido referido $(0.63/0.70/0.9 \text{ nV}/\sqrt{\text{Hz}})^{13}$.
- Rango lineal del amplificador de 1Vpp máximo referido de entrada.
- Acoplamiento capacitivo de entrada para AC.

¹³ nV/√Hz es una unidad empleada para medir la densidad espectral de ruido, establece el voltaje RMS de ruido introducido por el amplificador para un ancho de banda específico. Para calcular el ruido total es necesario multiplicar el valor especificado por el fabricante por la raíz cuadrada del ancho de banda de la señal de entrada [31].

- Circuito de corrección de offset (habilitado por el usuario). Éste se comporta como un filtro pasa altas y su frecuencia de corte se encuentra definida por $C_{\text{BYPASS}} \geq 15\mu\text{F}$ y los registros de control.
 - Terminación pasiva, activa (programable), o sin terminación. En general, la terminación activa es la recomendada por el fabricante dado que en ésta se tiene control sobre la impedancia de entrada, lo que da la capacidad de realizar acoplamientos de impedancia con distintos transductores.
- ❖ **VCAT (Atenuador controlado por voltaje)**
- Atenuación controlada por voltaje (VCNTL) con rango de operación de 0 a 40dB. Este elemento permite realizar TGC.¹⁴
- ❖ **PGA (Amplificador de Ganancia Programable)**
- Ganancia configurable a 24/30 [dB].
 - Circuitos limitadores de voltaje controlados digitalmente (mejoran la recuperación ante sobrecargas, disminuyendo desviación del ADC).
 - Circuito de corrección de offset.
 - Filtro pasa bajas antialiasing de tercer orden con frecuencias de corte de 10, 15, 30[MHz].
- ❖ **Convertor analógico digital**
- Conversión A/D con topología paralela con resolución de 14 bits, salida serie en formato LVDS a 12, 14 o 16 bits (cuando se configura a 12 bits suprime los dos LSB, cuando se configura a 16 bit agrega dos 0 adicionales en los MSB).
 - 10 y 65 MSPS. Los 8 canales se encuentran sincronizados con un solo reloj (diferencial o monopolar). Opera con referencias internas de voltaje REFP y REFM a 1.5 y 0.5V respectivamente, también puede usar una referencia externa.
 - Funciones de procesamiento digital de señales (promediado, filtrado, sustracción digital de offset).
- ❖ **Interfaz de control SPI**
- Se trata de un sistema de control que consta de un conjunto de registros que controlan las funciones del circuito. Cada palabra SPI está constituida por 3 bytes, el primer byte indica la dirección del registro y los dos restantes son la configuración a realizar.

2.3.5.2 Selección de tarjeta de almacenamiento y transmisión

Para la implementación del módulo de almacenamiento y transmisión se escogió la tarjeta de desarrollo, tarjeta SP605 (figura 2.8), que contiene un gran conjunto de elementos útiles para el diseño del sistema y otros adicionales que pueden servir para desarrollos futuros. Los más importantes se muestran a continuación (figura 2.9):

- FPGA modelo SPARTAN 6 XLS45T en encapsulado FGG484
- Conector FMC-LPC

¹⁴ TGC (Time Gain Control) es una técnica empleada en los dispositivos de imagenología por ultrasonido, en la cual se modifica la ganancia (o atenuación en este caso) conforme varía el tiempo en un disparo de ultrasonido, esto permite hacer énfasis en ciertas regiones de la imagen.

- Memoria FLASH y BPI de 8[MB] y 32[MB], respectivamente
- Sistema ACE CF (sistema de memoria FLASH) de 2[GB]
- Circuito emulador de JTAG
- 4 LEDs, 4 push buttons, 4 pines libres
- Transceptores SFP y 4 conectores SMA (Tx/Rx) GTP
- Memoria SDRAM DDR3 de 128[MB] con bus de 16 líneas y frecuencia de operación máxima de 400[MHz] (aunque las características físicas de la tarjeta limitan la frecuencia a 333[MHz])
- Puente USB UART
- Relojes instalados: LVDS a 200[MHz], LVCMOS 27[MHz], así como un conector SMA para reloj externo
- Puerto Ethernet
- Sistemas de acondicionamiento de voltaje
- PCI express
- Salida de video DVI



Figura 2.8 Tarjeta de desarrollo SP605.

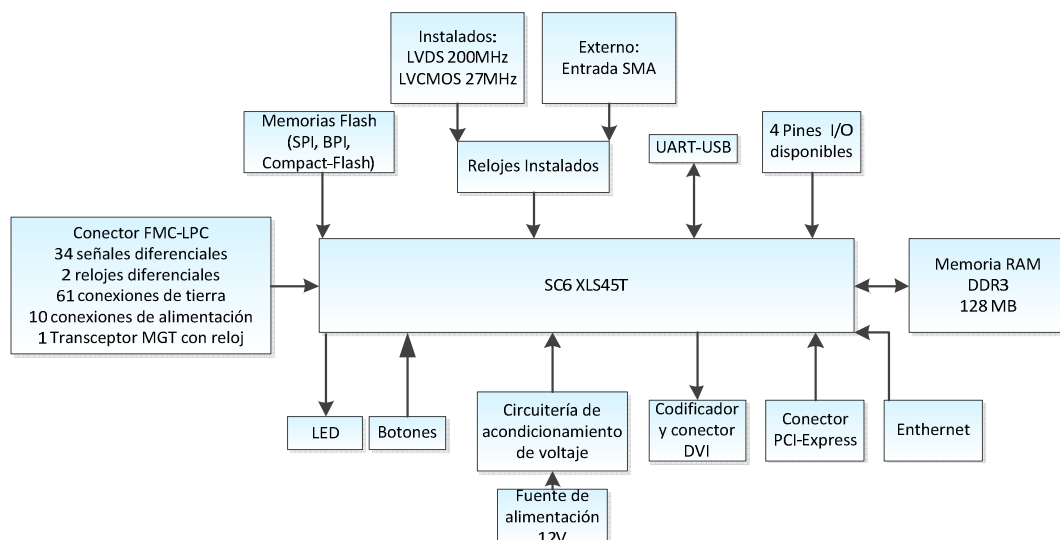


Figura 2.9 Representación a nivel de bloques hardware de la tarjeta de desarrollo de Xilinx SP605.

Las características más importantes que permiten al SC45XLS45T cumplir con los requisitos para el diseño son los siguientes:

- Capacidad para manejar los estándares de puertos necesarios para el proyecto (LVDS, LVCMOS, LVTL 1.2V) con resistencias de pull-up, pull-down o de terminación diferencial, según sea el caso.
- Los puertos LVDS cumplen con las características de transferencia adecuadas.
- Gran cantidad de celdas lógicas disponible (50K CLBs).
- Disponibilidad de PLL (4), necesarios para generar las señales de reloj del MCB.
- Capacidad hasta para 8 relojes globales.
- 2 controladores de memoria instalados con capacidad para SDR, DDR, DDR2, DDR3, LPPDR RAM.
- Bajo costo.

En la figura 2.10 se muestra una representación de la estructura interna del dispositivo, en la cual se especifica cada uno de sus componentes.

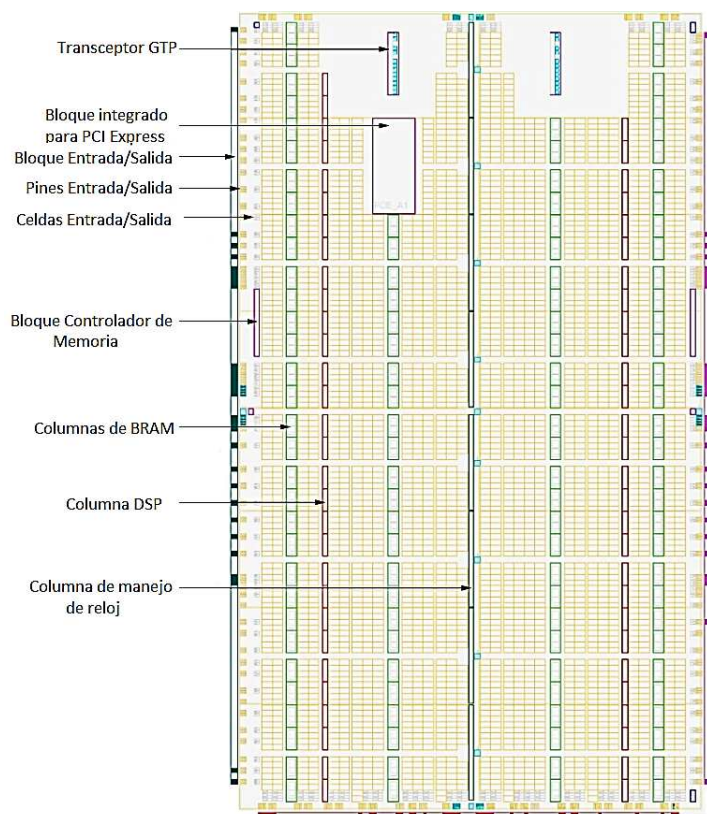


Figura 2.10 Estructura interna de los FPGA pertenecientes a la familia Spartan 6 XLST, en el esquema se pueden apreciar sus componentes principales: transceptor GTP, bancos de entrada/salida (4 en el caso del XCXLS45T), celdas de entrada/salida, bloques controladores de memoria, columnas de BRAM (*Block RAM*), celdas de procesamiento digital de señales, columnas de manejo de reloj –cada una contiene 1 PLL y 2DCM– [18].

2.3.5.3 Adaptador FMC-ADC

Como se mencionó anteriormente, el estándar FMC facilita las tareas de diseño, ya que permite una integración directa entre las tarjetas madre e hija que estén apegadas al estándar. Texas Instruments no brinda ningún módulo directo para este propósito; sin embargo, tiene a la venta una tarjeta adicional que permite adaptar cualquiera de sus circuitos convertidores analógico-digitales de alta velocidad a una tarjeta madre FMC (ver la figura 2.11).



Figura 2.11 Adaptador FMC empleado para conectar el módulo de acondicionamiento analógico conversión A/D al módulo de almacenamiento y transmisión.

Capítulo 3: Diseño del sistema

El proceso de diseño del sistema involucra las siguientes etapas de diseño tanto de software, como de hardware:

- Diseño del software de configuración del módulo de acondicionamiento analógico y CAD, además del correspondiente a la recepción y agrupamiento de datos del módulo de almacenamiento y transmisión.
- Descripción de hardware del módulo de almacenamiento y transmisión.

Antes de describir el diseño de cada bloque, es importante se hablará de la estrategia general que se tomó en cuenta para generar el sistema, lo cual ayudará a comprender el comportamiento global del sistema. Posteriormente, se hará una descripción más detallada de cada elemento.

Un esquema general de funcionamiento se describe en la figura 3.1 y 3.2, en la que se pueden distinguir cinco procesos principales: en el primero, el usuario introduce la configuración deseada para el ensayo; posteriormente, en el segundo, se configuran los módulos (MAACAD y MAT); en el tercero, se realiza el ensayo; después el MAT envía las muestras capturadas a la computadora para que ésta las organice; por último, se entregan los datos al usuario en forma de arreglos matriciales para su procesamiento.

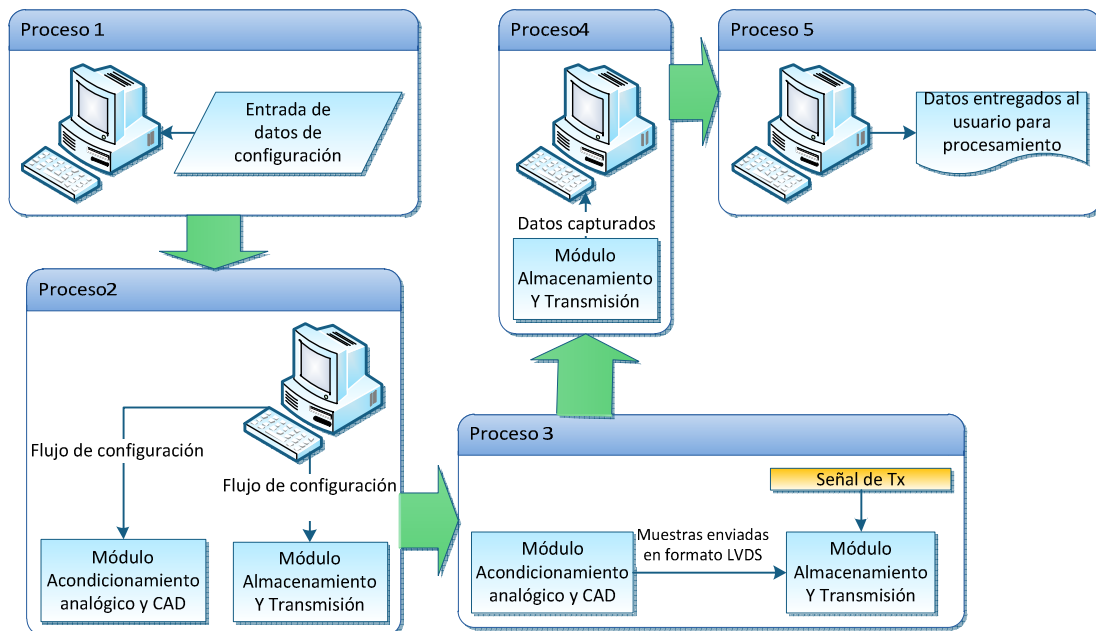


Figura 3.1 Esquema general de funcionamiento del sistema de adquisición de datos.

Una descripción de las tareas implementadas del software se muestra en la figuras 3.1 y 3.2. Éste fue llevado mediante la herramienta computacional MATLAB, dado su amplio uso entre

los integrantes del laboratorio. Este espacio ofreció un conjunto de herramientas útiles para el presente proyecto, porque facilitó la implementación de la interfaz de usuario.

- ❖ **Proceso 1.** Un conjunto de funciones implementadas en scripts crean y modifican una estructura de datos en la que se encuentran los parámetros de configuración del sistema. Asimismo, éstas deben realizar tareas de validación de datos para evitar errores de operación.
- ❖ **Proceso 2.** El programa genera las cadenas de configuración para cada uno de los dispositivos y las envía a cada uno de ellos.
- ❖ **Proceso 3.** Con la configuración definida por el usuario se lleva a cabo el ensayo, el sistema debe sincronizarse con el pulsador para esta tarea.
- ❖ **Proceso 4.** Mediante un programa de control, la PC recibe las muestras almacenadas en el MAT y las ordena.
- ❖ **Proceso 5.** Se entrega un arreglo de matrices al usuario para que pueda procesarlas.

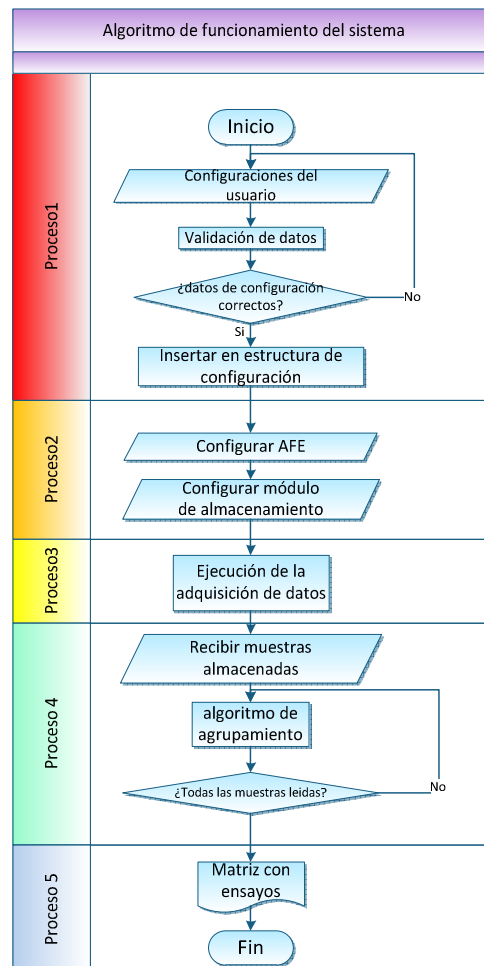


Figura 3.2 Diagrama de flujo del sistema de adquisición de señales de ultrasonido.

Almacenamiento

Un elemento especialmente importante para lograr los requisitos del sistema planteados previamente es un esquema de direccionamiento de memoria lo más eficiente posible, tanto en términos de tiempo como de desperdicio de memoria.

En el capítulo 1 se explicó que las memorias RAM tienen una estructura matricial, sin embargo, es importante especificar que las memorias modernas poseen más de una matriz; cada una de éstas es llamada banco.

Las características de la memoria empleada (MT41J64M16LA-187E) son las siguientes: 8 bancos, 128 columnas, 8192 filas y celdas de 128 bits. Con las características anteriores, se tienen aproximadamente 8 millones de localidades que permiten almacenar 67.1 [MSa].

Es importante evitar emplear un esquema de direccionamiento que implique realizar cambios consecutivos de bancos, filas o columnas, ya que cada vez que se cambia cualquiera de estos valores es necesario realizar un conjunto de procesos que agregan latencia, esto implica un ancho de banda reducido, además de una estructura de control compleja.

El esquema de direccionamiento propuesto considera que la memoria es un único arreglo unidimensional. Esto implica la menor cantidad de cambios, incrementando el ancho de banda del sistema, por lo que las direcciones están formadas de la siguiente forma (ver figura 3.3):

$$\text{Dirección}_{\text{Unidimensional}} = \text{Banco} \& \text{Fila} \& \text{Columna}$$

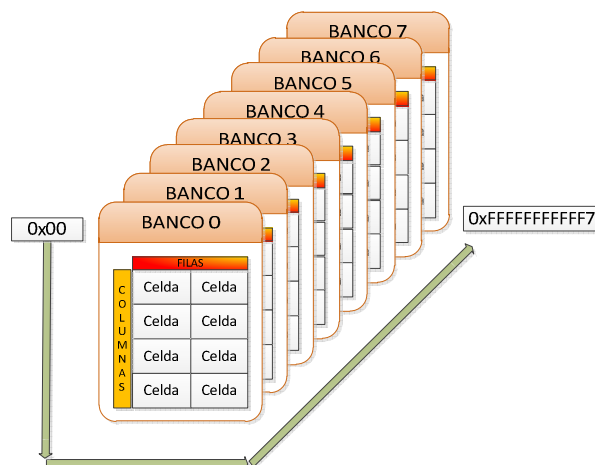


Figura 3.3 Esquema de direccionamiento de la memoria RAM.

El mapa de memoria propuesto es el mostrado en la figura 3.4. Ahí todas las muestras de los canales se almacenan en forma consecutiva, alternando canales en forma cíclica. En la figura 3.6 se muestra un caso en que se tienen los 8 canales activos; siempre el primer canal está dado por el canal 0 o el más cercano a éste, posteriormente sigue el 1 (o el más cercano) y así consecutivamente. En el caso de que se efectúe más de un disparo continuará de la misma forma,

de modo que siempre es posible calcular la dirección de cada elemento, lo que facilita la recuperación y separación de los datos.

0x0	Ch0
0x1	Ch1
0x2	Ch2
0x3	Ch3
0x4	Ch4
0x5	Ch5
0x6	Ch6
0x7	Ch7
0x8	Ch0
0x9	Ch1
0xA	Ch2
0xB	Ch3
	·
	·
	·
0xN	Ch7

Figura 3.4 Mapa de memoria y esquema para el almacenamiento de muestras.

3.1 Diseño del hardware del Sistema de Captura y Almacenamiento

Éste debe ser capaz de captar las señales provenientes de los 8 canales del *Analog Front End*. Su diseño es crítico, ya que debe organizar, ordenar y almacenar una gran cantidad de información en poco tiempo. Los bloques con los que debe cumplir el diseño son:

- Recepción LVDS por cada canal
- Organización de la información para almacenamiento
- Controlador de memoria RAM DDR3
- Transceptor UART
- Registros de configuración
- Sistema de configuración de registros
- Sistema de monitoreo y control
- Sistema de control de flujo de salida de datos

3.1.1 Funcionamiento general del sistema

El sistema de captura puede dividirse en dos secciones: por un lado, la ruta de datos involucra subsistemas que interactúan directamente con las muestras provenientes del ADC, tales como módulos de comunicación LVDS, concatenadores, multiplexores, entre otros. Éstos se realizan mediante transferencias a nivel RTL en forma de descripciones de comportamiento y máquinas de estados finitos con ruta de datos (FSMD).¹⁵ Por otro lado, el sistema de control monitorea y

¹⁵ FSMD se tratan de máquinas de estados que además de realizar tareas de control, también realizan operaciones sobre los datos, tales como almacenar, sumar, multiplicar, corrimientos, etc.

sincroniza todos los procesos involucrados en la captura mediante un conjunto de máquinas de estados finitos. Una representación del sistema completo se aprecia en la figura 3.5.

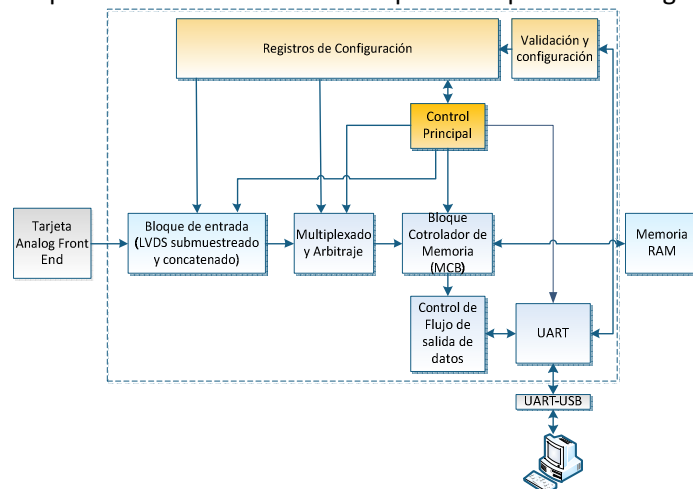


Figura 3.5. Representación de los sistemas que componen la etapa de almacenamiento y transmisión. En azul se aprecian los componentes que pertenecen a la ruta de datos. En Naranja se encuentran los elementos correspondientes a los sistemas de control.

La arquitectura propuesta se muestra en la figura 3.5. A grandes rasgos las tareas de los bloques presentes son las siguientes:

- **Bloque de entrada LVDS con Submuestreador:** recibe las muestras provenientes del ADC.
- **Concatenador:** realiza el agrupamiento de muestras por cada canal, de forma que sean compatibles con el resto del mismo.
- **Multiplexado y arbitraje:** se encarga de realizar el multiplexado y ordenamiento de datos que van a la memoria.
- **Bloque controlador de memoria:** provee una interfaz entre el sistema y la memoria RAM.
- **Memoria RAM:** se trata de un medio de almacenamiento temporal para las muestras.
- **Control de flujo de salida de datos:** segmenta las palabras provenientes de la memoria RAM en bloques compatibles con el módulo UART. Asimismo, realiza la función de control de flujo para evitar pérdidas de información al transmitir.
- **UART:** sirve como puente de comunicación con la PC. Éste realiza las tareas tanto de comunicación para configuración como de envío de las señales capturadas.
- **Registros de configuración:** éstos se encargan de almacenar las configuraciones definidas por el usuario.
- **Módulo de configuración y validación de registros:** se encarga de validar y configurar los registros de control, así como de recibir la instrucción de inicio.
- **Control principal:** sincroniza el funcionamiento de todo el sistema.

3.1.1.1 Ruta de datos

Ésta es de vital importancia, ya que su capacidad define la máxima velocidad de adquisición de datos, la cual está en función de la velocidad del sistema, del diseño de las máquinas de estados y

de la velocidad de transferencia a la memoria RAM. Una estructura simplificada de esta etapa se aprecia en color azul en la figura 3.5.

3.1.1.1.1 Bloque de entrada

La etapa de captura está compuesta por los siguientes bloques:

- Bloque de recepción LVDS
- FIFO con dominios de reloj asíncronos
- Submuestreador
- Concatenador

La cadena de la señal en esta etapa se aprecia en la figura 3.6. Posteriormente se ahondará en las características de cada bloque y en las estrategias realizadas para su implementación. Los bloques “LVDSChan” y “FIFODistRAM” se repiten por cada canal. El bloque de submuestreo “SubSampler” es único. Éste, en conjunto con el registro de canales habilitados, permite la escritura al FIFO “FIFODistRAM” de cada canal (mediante una AND), permitiendo desestimar muestras cuando el usuario así lo define mediante un registro del que se hablará más adelante.

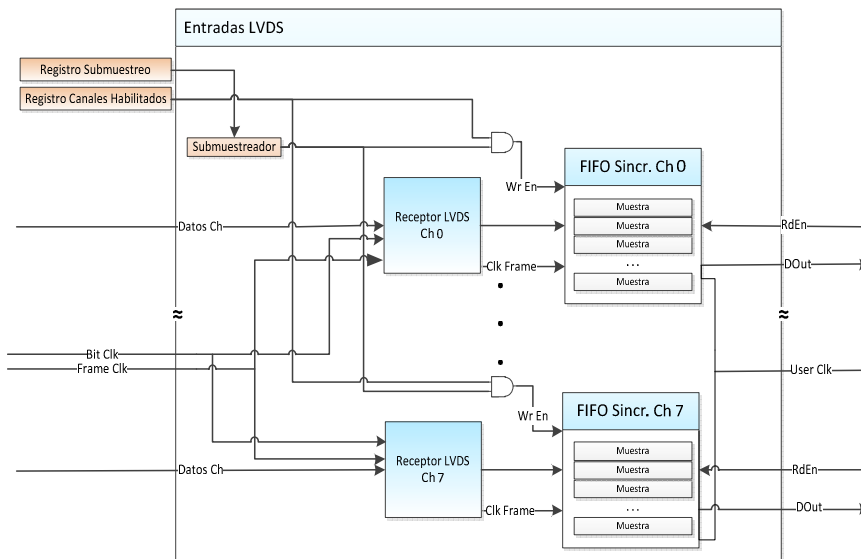


Figura 3.6 Representación esquemática de la etapa de captura y submuestreo.

3.1.1.1.2 Captura de señales LVDS

Los esquemas de configuración LVDS varían en forma importante, tanto en la topología de la red, como en la de los sistemas de reloj. El AFE5808A provee un esquema que se basa en el empleo de dos líneas de reloj adicionales a las de datos. Uno de los relojes define cada bit entrante a DDR,¹⁶ éste es llamado *Bit Clock*; el otro define el fin e inicio de cada muestra y es llamado *Frame Clock*,

¹⁶ DDRs (*Double Data Rate*) significa que el circuito secuencial se sincroniza tanto con el flanco de subido como con el flanco de bajada del reloj de referencia.

como se muestra en la figura 3.7. Para una mejor comprensión del protocolo refiérase al apéndice A.

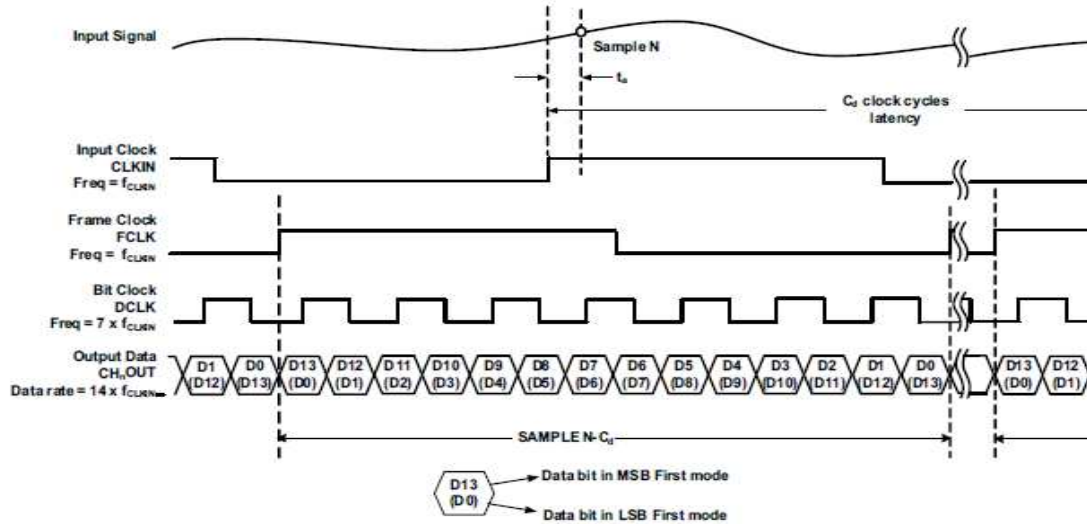


Figura 3.7. En ésta se aprecia la relación de tiempo entre la línea de datos serie y los relojes *bit* y *frame*.

Para lograr la captura de datos DDR es necesario emplear ya sea *flip-flops* que tengan soporte para esa función o un *flip-flop* para el flanco de subida y otro para el flanco de bajada. Se optó por esta última opción, ya que ésta puede ser directamente inferida de la descripción en VHDL sin la necesidad de hacer la llamada directa a una arquitectura en particular.

La forma en que funciona este bloque es mediante un registro de corrimiento que captura un bit en cada flanco del *Bit-Clock*, mientras que un proceso paralelo envía a la salida el registro capturado cada flanco de subida del *Frame-Clock*. La representación del bloque se muestra en la figura 3.5; la descripción del bloque en VHDL se encuentra en el apéndice C. Esta estructura se repite por cada canal.

3.1.1.1.2.1 FIFO

Ya que las muestras llegan en forma continua y previendo el caso de que el resto del sistema se encuentre temporalmente ocupado causando pérdidas de información, a la salida de cada bloque LVDS se agregó un buffer FIFO para almacenamiento temporal, el cual adicionalmente sirve como sincronizador de dominios de reloj para evitar metaestabilidad,¹⁷ porque el resto de la lógica necesita un reloj que opere a mayor frecuencia por razones que se expondrán más adelante.

Una estructura común para este tipo de circuitos puede verse en la figura 3.8. En ésta se aprecia que el funcionamiento de estos dispositivos se basa en la sincronía entre apuntadores de lectura y escritura con su dominio respectivo.

¹⁷ Metaestabilidad se refiere a un evento en que es imposible determinar el valor de una señal del sistema, debido a que éste se encuentra en el límite que distingue los dos niveles lógicos permitidos [19].

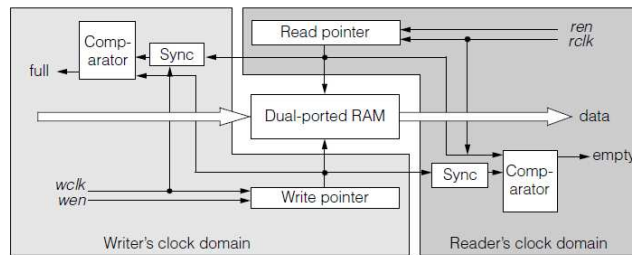


Figura 3.8 FIFO actuando como circuito de sincronización entre dos circuitos con dominios de reloj asíncronos [19].

Para evitar la complejidad de la temporización involucrada en un FIFO con dominios de reloj asíncrono (un reloj de escritura y uno de lectura), se recurrió a la herramienta *FIFO Generator*, que se encuentra integrada en el paquete *ISE design suite*.

3.1.1.1.2.2 Submuestreador

Dado que la frecuencia del reloj del módulo del ADC es fija y la frecuencia de los traductores puede variar considerablemente (entre 100[KHz] y 5[MHz]), muestrear a 40[MHz] puede ser muy costoso en términos de memoria y ancho de banda para transmisión, por lo que no es apropiado para todos los casos. Para el caso de los transductores de baja frecuencia se generaría una cantidad excesiva de información que ocuparía memoria que puede ser empleada para otros procesos o adquisiciones de mayor duración. A esto, un sistema de submuestreo es una buena alternativa; sin embargo, hay que tener en cuenta que este proceso puede introducir aliasing, ya que incluso tratándose de muestreo de señales digitales se debe de cumplir con el teorema de Nyquist.

El proceso de submuestreo por lo general está acompañado de un filtro pasa bajas en una etapa previa para evitar aliasing. La conjunción de ambos elementos conforma lo comúnmente denominado como *filtro de decimación*. La descripción de este último procedimiento no es el propósito de este proyecto, sin embargo, la arquitectura se diseñó de forma tal que sea fácil la integración de dicha etapa en caso de ser requerida en posteriores aplicaciones.

El funcionamiento de este sistema se basa en un contador. Su cuenta va de 0 hasta el factor de submuestreo (F_{Sub}). Cada vez que la cuenta llega a F_{Sub} se permite la entrada de una muestra al sistema mediante la señal de control wr_en (habilitar escritura), lo que produce una frecuencia de muestreo efectiva de:

$$f_{Efectiva\ de\ Muestreo} = \frac{f_s}{1 + f_{Sub}}$$

3.1.1.1.2.3 Concatenador

La razón de integrar este bloque corresponde a que la palabra mínima que maneja el controlador es de 32 bits y la máxima de 128 bits. Las celdas de la memoria RAM instalada coinciden con este último valor, por lo que es conveniente manejar las palabras de dicha longitud para que el incremento en la dirección de memoria sea de 1 celda por muestra almacenada.

El circuito opera del siguiente modo: si el FIFO proveniente del sistema de recepción LVDS tiene una o más muestras almacenadas, el concatenador las almacena en la posición correspondiente, de lo contrario espera hasta que algún dato llegue. El sistema sigue operando del mismo modo hasta haber completado una palabra de 128 bits. Posteriormente la envía a un FIFO de almacenamiento temporal, esta relación puede apreciarse en la figura 3.9.

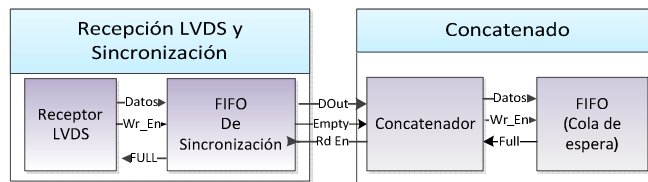


Figura 3.9. Relación entre los módulos que conforman la cadena del bloque de entrada.

Este sistema funciona mediante una FSMD (ver la figura 3.10 y la tabla 3.1.). Ésta opera en función de una bandera de entrada y de una señal de habilitación. Para aumentar la eficiencia, va precedida de un FIFO, de modo que si el sistema se encuentra temporalmente detenido, la FSMD puede continuar trabajando. La estructura de la palabra concatenada se muestra en la tabla 3.2.

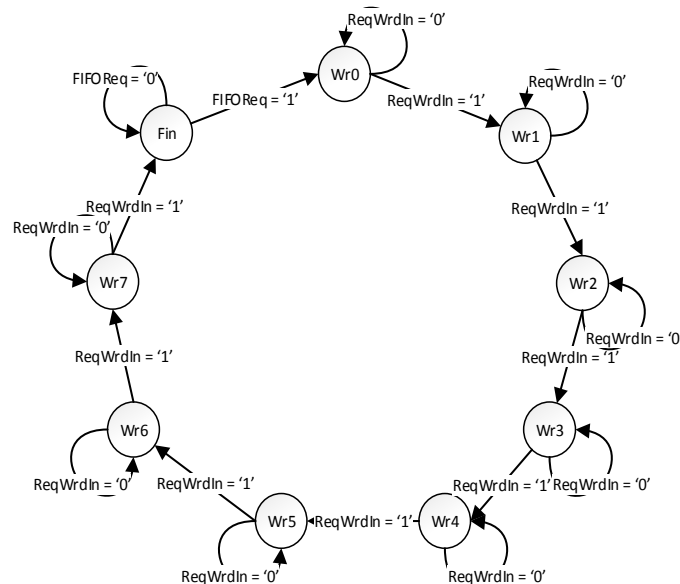


Figura 3.10 Máquina de estados de concatenado. Consta de nueve estados. En los primeros 8 almacena las palabras para la memoria RAM. En el último envía la palabra a un FIFO externo.

No. de estado	Nombre del estado	Función
1 - 8	WrX	Almacena una muestra.
9	Fin	Envía la palabra concatenada al FIFO.

Tabla 3.1 Funcionamiento de los diversos estados que componen la FSM de concatenado.

No. de muestra	8	7	6	5	4	3	2	1
Longitud	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit	16 bit
Posición	112 - 127	96 - 111	80 - 95	64 - 79	48 - 63	32 - 47	15 - 31	0 - 15

Tabla 3.2 Estructura de cada palabra concatenada.

3.1.1.1.3 Arbitraje y multiplexado

Los dispositivos de entrada y la memoria RAM conforman una red con un único bus compartido que lleva información unidireccionalmente, ésta proviene de los 8 canales de entrada con la cadena descrita anteriormente.

En general, las redes disponen de tres tipos de subsistemas para su funcionamiento: de ruteo, arbitraje y dispositivos de conmutación. En este caso, el sistema de ruteo no es necesario dado que hay un solo dispositivo de destino, o sea, la memoria RAM.

El sistema de arbitraje se encarga de asignar tiempos de acceso para evitar conflictos entre los dispositivos. En este caso todos los que se encuentren activos deben tener la misma cantidad de tiempo para enviar información, la cual será de un paquete de datos cada vez que se dé acceso. La conmutación de los dispositivos se realiza mediante un multiplexor, como se puede ver en la figura 3.11.

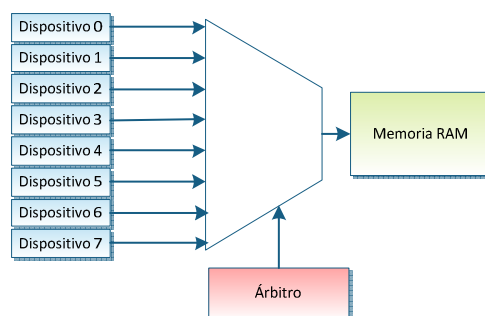


Figura 3.11 Elementos involucrados en la interconexión entre los dispositivos de entrada y el acceso a memoria.

3.1.1.1.3.1 Arbitraje

Se basa en una máquina de estados que realiza la asignación de tiempos de acceso según el registro de canales habilitados. El esquema que sigue es el denominado como Round Robin, que da acceso en forma cíclica durante un intervalo de tiempo definido (ver figura 3.12).

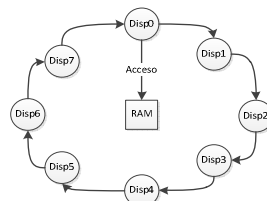


Figura 3.12 Algoritmo de arbitraje Round-Robin. En este caso, cada dispositivo habilitado envía un paquete.

El acceso a cada dispositivo es de 1 paquete por cada ciclo de acceso si es que está habilitado en el registro. De lo contrario determina la siguiente posición activa mediante un codificador con prioridad y va a ese estado, de modo que es posible brincar uno o más elementos no habilitados.

3.1.1.1.4 Bloque Controlador de Memoria (MCB)

Es un sistema que se encuentra físicamente en el dispositivo y es accesible sólo mediante la herramienta *Core Generator*, un *core* de propiedad intelectual llamado *Memory Interface Generator (MIG)*. Este sistema también provee una interfaz de usuario que configura de 1 a 6 puertos de entrada y salida, lo que simplifica la tarea de comunicación con la memoria RAM [20].

La ventaja de usar el controlador integrado implementarlo en la lógica programable es que aparte de ahorrar elementos permite alcanzar velocidades de operación de hasta 800[MT/s] [20], aunque debido a la construcción de la tarjeta sólo se pueden alcanzar 667[MT/s] [21].

El SP6XLS45T dispone de dos controladores, uno en el banco 1 y otro en el banco 3 [22]. Este último fue empleado porque la memoria RAM instalada en la tarjeta se encuentra conectada a las terminales de dicho banco. En la figura 3.13 se aprecian el controlador de memoria y sus componentes. Es importante resaltar que la distribución de los puertos (2 puertos bidireccionales y 4 unidireccionales es configurable [22]). En este caso se configura como un único puerto de 128 bit dado que ése es el tamaño de la celda de memoria RAM empleada.

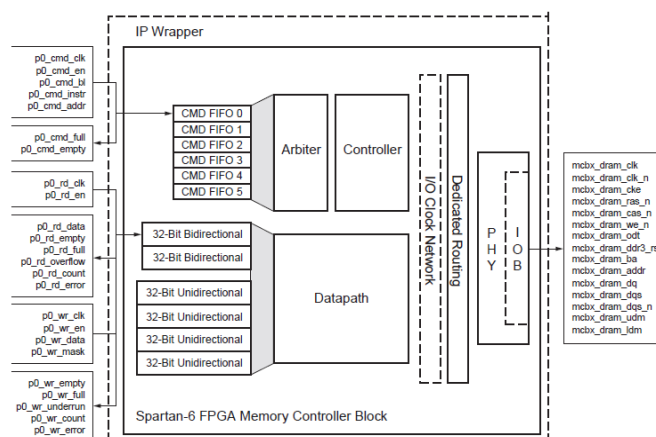


Figura 3.13 Arquitectura del controlador de memoria. Se aprecian los principales elementos que lo componen (dentro de la línea punteada), asimismo, se observan las señales de la interfaz de usuario (fuera del recuadro) [20].

La interfaz provista por el *Memory Interface Generator* consta de tres FIFOs que controlan el funcionamiento de todo el sistema y un PLL que genera las señales de reloj necesarias.

3.1.1.1.4.1 FIFO de escritura

Tiene capacidad para almacenar hasta 64 elementos. En él se guardan las muestras y la lectura se realiza automáticamente por el sistema cuando se da la instrucción en la sección de control. Las señales y un resumen de su funcionamiento se muestran en la tabla 3.3.

Función	Nombre en el core
Reloj de FIFO de escritura	p3_wr_clk
Cuenta el número de elementos almacenados. Se proporciona para diseño únicamente debido a su latencia.	p3_wr_count[6:0]
Entrada de datos a escribir en la memoria	p3_wr_data[127:0]
FIFO vacío	p3_wr_empty
FIFO lleno	p3_wr_full
FIFO vacío	p3_wr_empty
Habilitar escritura	p3_wr_en
Error de escritura	p3_wr_error
Sirve para enmascarar la escritura de datos, en este caso no se enmascaran los datos nunca.	p3_wr_mask[127:0]
Se pone en alto cuando se intenta realizar una operación de escritura y no hay suficientes datos para ejecutarla.	p3_wr_underrun

Tabla 3.3 Señales de la interfaz de usuario de escritura.

3.1.1.1.4.2 FIFO de lectura

Es muy similar a la interfaz de escritura, sin embargo, en este caso únicamente se pueden realizar operaciones de lectura en este FIFO. Las señales y un resumen de su funcionamiento se muestran en la tabla 3.4.

Función	Nombre en el core
Reloj de FIFO de lectura	p3_rd_clk
Cuenta el número de elementos almacenados. Se proporciona para diseño únicamente debido a su latencia.	p3_rd_count[6:0]
Salida de datos leídos de la memoria	p3_rd_data[127:0]
FIFO vacío	p3_rd_empty
FIFO lleno	p3_rd_full
FIFO vacío	p3_rd_empty
Habilitar lectura	p3_rd_en
Error de lectura	p3_rd_error
Se pone en alto cuando se realiza una operación de lectura y no hay suficiente espacio para almacenar datos, por lo que el FIFO se desborda.	p3_rd_overflow

Tabla 3.4 Señales de la interfaz de usuario de lectura.

3.1.1.1.4.3 FIFO de comandos

Este FIFO almacena los datos necesarios para ejecutar una instrucción de lectura o escritura. Tiene 4 elementos de profundidad. El valor de la longitud de la ráfaga se mantiene fijo para aumentar lo más posible el ancho de banda del sistema. Las señales de control de esta etapa se ilustran en la tabla 3.2. Las señales y un resumen de su funcionamiento se muestran en la tabla 3.5.

Función	Nombre de la señal ¹⁸
Dirección de escritura/lectura en memoria	p3_cmd_addr[29:0]
Longitud de la ráfaga de escritura/lectura	p3_cmd_bl[5:0]
Reloj del FIFO de comandos	p3_cmd_clk
Señal de FIFO vacío	p3_cmd_cmd_empty
Señal de error	p3_cmd_en
FIFO lleno	p3_cmd_full
FIFO vacío	p3_cmd_empty
Instrucción de comandos	p3_cmd_isntr[2:0]

Tabla 3.5 Señales de control del MCB.

3.1.1.1.4.4 Señales de reloj

Adicional a la interfaz de control, el MCB también invoca un PLL que emplea para generar las señales de reloj necesarias para que el sistema pueda operar correctamente. Los relojes que se generan son:

- Relojes de operación de la interfaz física. Operan exactamente al doble de la frecuencia establecida para la interfaz (en este caso para operar la memoria a 300[MHz] es necesario generar un reloj de 600[MHz]), son dos relojes que están desfasados 180°.
- Reloj empleado para calibrar la interfaz física. Siguiendo las recomendaciones de [20] se fijó a 100[MHz].
- Reloj de usuario. Éste es opcional y en este caso fue útil sincronizarlo para toda la lógica del sistema.

3.1.1.1.5 Control de flujo de salida de datos

Se lleva a cabo mediante una FSM que segmenta en 16 palabras de 8 bit la palabra concatenada de 128 bits proveniente de la memoria RAM en la fase de lectura. Para finalizar espera una palabra de confirmación. El diagrama de estados se aprecia en la figura 3.14. El funcionamiento de los estados se resume en la tabla 3.6

¹⁸Para aumentar el ancho de banda y reducir la latencia del sistema, las memorias RAM modernas utilizan ráfagas, de modo que se realizan varias escrituras/lecturas consecutivas a partir de la dirección de inicio.

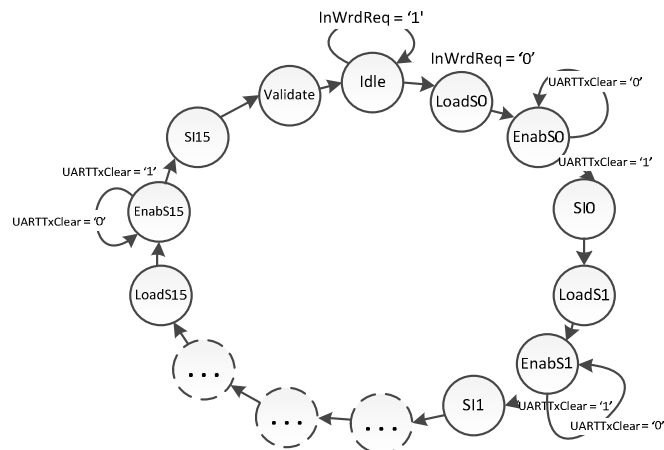


Figura 3.14 Diagrama de la máquina de estados de control de flujo de salida de datos.

No. de estado	Nombre del estado	Función
1	Idle	Se encuentra en espera de un dato para enviar.
2	LoadSX	Carga el dato de entrada para enviarlo al módulo UART (repetido 16 veces).
3	EnabSX	Verifica que el módulo UART esté disponible, de ser así envía la señal de solicitud (repetido 16 veces).
4	SIX	Retira la señal de solicitud (repetido 16 veces).
5	Validate	Espera dato de confirmación.

Tabla 3.6 Estados de la FSM de control de flujo y salida de datos.

3.1.1.1.6 UART¹⁹

Cumple dos funciones principales: en un inicio permite realizar la configuración del circuito y posteriormente debe funcionar como salida de datos. Dado que tiene dos dispositivos tratando de tener acceso al módulo es necesario contar con un sistema de multiplexado y uno de control, caso similar al acceso de la memoria RAM. Una explicación más detallada del funcionamiento se encuentra en el apéndice A. **Transmisión**

El protocolo de comunicación UART establece una secuencia específica para el envío de palabras. Inicialmente se envía un '0' que marca el principio de la palabra, posteriormente, dependiendo de la configuración, se envían entre 6 y 8 bits de datos (que es este caso). Finalmente se envía el bit de parada. Esto puede apreciarse en la figura 3.15. El funcionamiento de cada estado se resume en la tabla 3.7.

¹⁹ UART es el acrónimo para *Universal Asynchronous Receiver/Transmitter*. Se trata de un protocolo de comunicación serie asíncrona común en los sistemas de comunicaciones electrónicas.

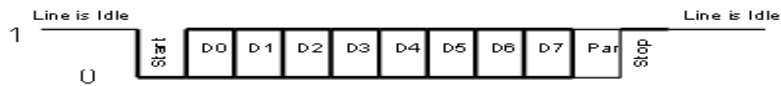


Figura 3.15 Temporización de una palabra de 7 bits enviada por UART. En el caso de este proyecto no se emplea bit de paridad [23].

Para lograr la temporización apropiada, el circuito se implementó mediante una máquina de estados cuyo funcionamiento se aprecia en la figura 3.16, que se describe a continuación:

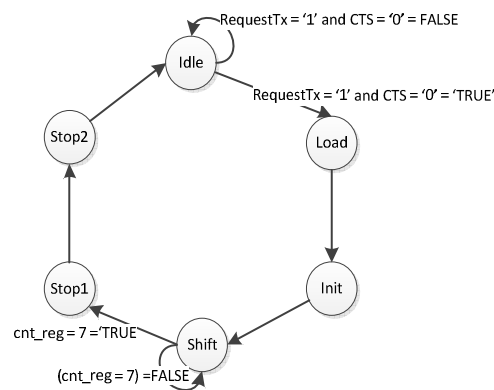


Figura 3.16 Diagrama de estados del módulo de transmisión UART.

No. de estado	Nombre del estado	Función
1	Idle	En espera de la señal de envío (<i>RequestTx</i>)
2	Load	Retardo de carga de palabra a enviar
3	Init	Envía bit de inicio.
4	Shift	Se envía un bit a la vez.
5	Stop1	Primer bit de parada
6	Stop2	Segundo bit de parada

Tabla 3.7 Funcionamiento de los estados del módulo de transmisión UART.

3.1.1.1.6.1 Recepción

El funcionamiento del sistema de recepción es muy similar al del transmisor descrito previamente. Una vez detectado el bit de inicio se realiza un desfaseamiento, de forma que sea posible realizar un muestreo de la señal cada periodo con una posibilidad mínima de errores. Finalmente una vez que se han muestreado todos los bits de la palabra el sistema realiza una pausa en forma de bits de parada (ver figura 3.16). La máquina de estados implementada se muestra en la figura 3.17; el funcionamiento de cada estado se resume en la tabla 3.7.

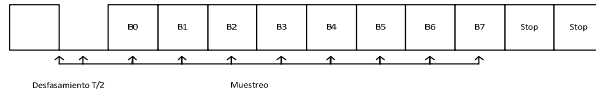


Figura 3.16 Funcionamiento de un módulo de transmisión UART en el que se aprecia el desfasamiento inicial de medio periodo, de forma que al muestrear cada periodo se tenga una posibilidad de errores menor.

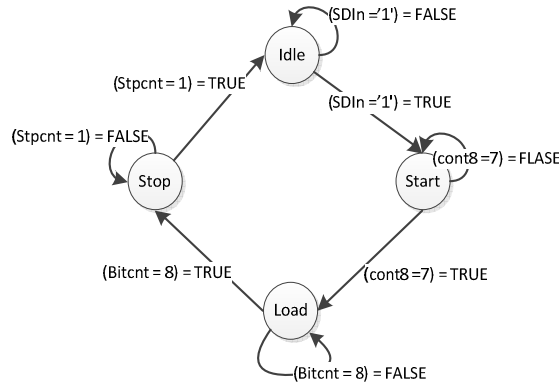


Figura 3.17 Diagrama de estados del módulo de recepción UART.

No. de estado	Nombre del estado	Función
1	Idle	Espera un bit de inicio, que es un '0'.
2	Start	Realiza un retardo de exactamente medio periodo de la señal.
3	Load	Muestra un bit cada periodo de la señal hasta que se tiene la palabra completa.
4	Stop	Espera dos periodos de reloj y regresa el sistema a "Idle".

Tabla 3.7 Funcionamiento de los estados del módulo de recepción UART.

3.1.1.1.6.2 Control de acceso

Para transmisión el control al acceso de los puertos se lleva a cabo mediante dos multiplexores. El primero conmuta la señal que avisa al puerto que hay datos en la entrada (*Request*); el segundo multiplexor se encarga de conmutar un bus de entrada con la información a transmitir. Las señales de control provienen de la FSM de control principal. La implementación fue realizada mediante el arreglo mostrado en la figura 3.18.

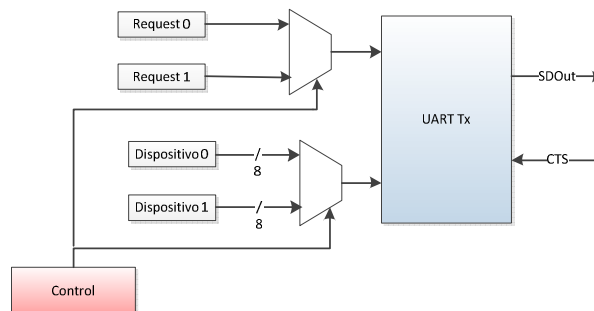


Figura 3.18 Estructura del sistema de control de acceso al módulo UART de transmisión.

3.1.1.2 Sistemas de control

El control del sistema se encuentra concentrado en una arquitectura principal, sin embargo, éste también se encuentra distribuido en las máquinas de estados finitos con ruta de datos descritas previamente. A continuación, se hablará acerca de las tareas y los elementos principales de esta estructura de control.

3.1.1.2.1 Registros de control

El sistema se encuentra controlado sobre todo por tres registros. A continuación se dará una breve descripción de las tareas de cada uno de ellos y en la figura 3.19 se muestra una representación del sistema. .

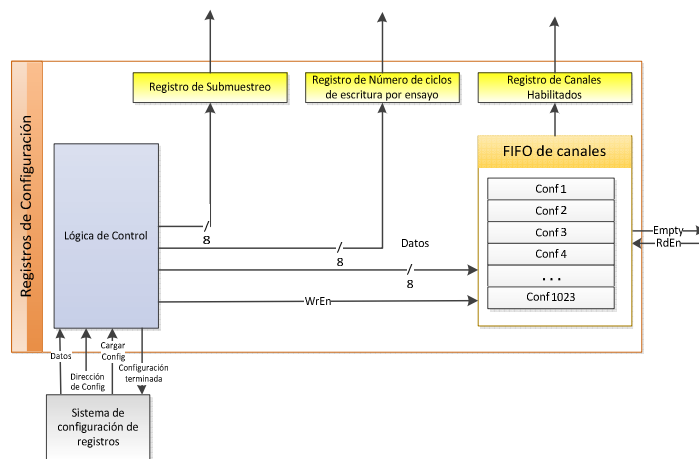


Figura 3.19 Estructura del sistema de registros.

3.1.1.2.1.1 Canales habilitados por cada disparo del pulsador.

Esto es extremadamente importante ya que es posible realizar distintas rutinas dependiendo del ensayo a realizar. Esto es posible de alcanzar mediante un FIFO que contiene canales que están habilitados para cada canal en cada secuencia de adquisición.

3.1.1.2.1.2 Número de muestras (ciclos de escritura)

Para alcanzar longitudes de hasta 60[cm] para configuración pulso-eco (establecida como la máxima por el personal encargado de este proyecto) es necesario un registro relativamente grande. Para permitir manejar un registro estándar de 8 bits se planteó que el número almacenado en ese registro equivale al número de veces que se cargará completamente el FIFO del MCB para cada canal activo.

Cada palabra que llega al FIFO del MCB contiene 8 muestras y el FIFO almacena 64 elementos, por lo que la distancia promedio de cada ciclo de escritura equivale a:

$$No. Muestras por ciclo de escritura = 512 [Muestras]$$

considerando la velocidad de propagación de los elementos comúnmente analizados en el laboratorio que se muestran en la tabla 3.8.

Medio	Velocidad de propagación [m/s]
Agua	1480
Concreto	3200 – 3600
Aluminio	6420
Hueso	3500 – 4300

Tabla 3.8 Velocidad de propagación del sonido en algunos materiales comúnmente empleados en el laboratorio de acústica de la Facultad de Ciencias.

A partir de la tabla 3.8 es posible apreciar que el medio de propagación más baja es el agua, por lo que para poder realizar las adquisiciones de la longitud establecida es necesario considerar la velocidad de propagación del agua para determinar la longitud del registro:

$$t_{ciclo\ de\ escritura} = \frac{\left(\frac{Profundidad_{FIFO}}{16[bit]}\right) (Palabra_{FIFO})}{F_{s\ Efectiva}}$$

Si la tasa de submuestreo se conserva en 0 entonces $t_{ciclo\ de\ escritura} = 12.8 [\mu s]$

$$d_{max} = (v)(t_{ciclo\ de\ escritura})(2^{longitud_{registro}} - 1)$$

lo que nos lleva a una longitud máxima de 2.41 [m] para pulso-eco y 4.8 [m] para transmisión en el caso de que se empleara un solo canal por cada disparo. La profundidad de cada muestra se disminuiría por cada canal agregado en forma inversamente proporcional al número de canales. Así por ejemplo, empleando los 8 canales en forma simultánea, la profundidad máxima de adquisición es de 63[cm] para transmisión y 31.5 [cm] para pulso-eco.

3.1.1.2.1.3 Factor de submuestreo

Este registro establece el factor de submuestreo para la etapa de entrada del sistema.

3.1.1.2.2 Sistema de configuración de registros

Es un sistema fundamental para el correcto funcionamiento del sistema. Consta de dos FSMs que trabajan en conjunto: la primera se encarga de asegurarse que las palabras recibidas por el puerto UART estén libres de errores; la segunda se encarga de que las palabras recibidas sean configuraciones válidas, posteriormente las carga al registro correspondiente.

3.1.1.2.2.1 Máquina de validación de datos

La configuración del dispositivo es una tarea crítica, ya que de ésta depende el correcto funcionamiento del circuito. La FSMD (ver figura 3.20 y tabla 3.9) de validación se encarga de asegurarse que todos los datos sean correctos por medio de un esquema redundante, donde cada dato debe ser enviado dos veces al dispositivo. Adicionalmente ésta se comunica con la máquina de configuración de registros, la cual se asegura de que sea un dato válido. Posteriormente, si es correcto, le envía a la FSMD de validación de datos una señal “ACK” (0x6 en ASCII), de lo contrario regresa una señal “NAK” (0x11 en ASCII).

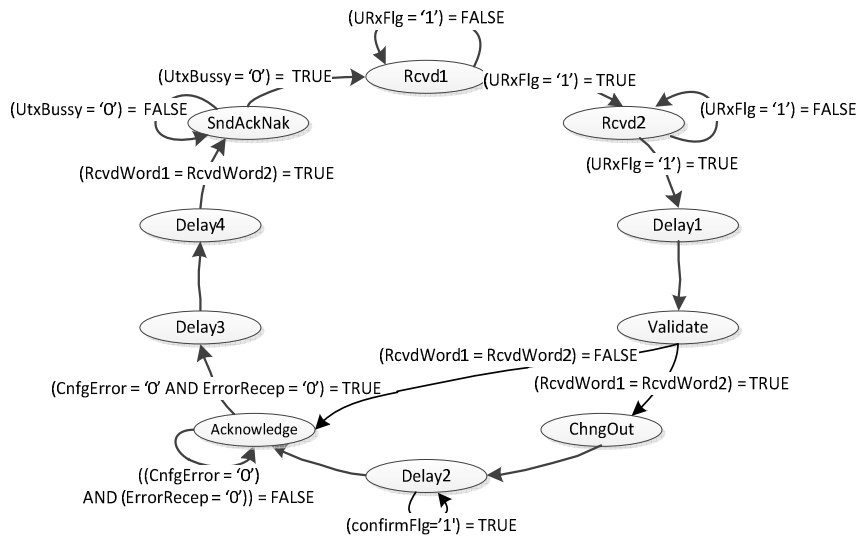


Figura 3.20 Diagrama de estados de la FSM de validación de datos.

No. de estado	Nombre del estado	Función
1	Rcvd1	Recibe la primera palabra de configuración.
2	Rcvd2	Recibe la segunda palabra de configuración.
3	Delay1	Se espera un estado para que se establezca la entrada.
4	Validare	Compara las dos palabras para verificar que sean la misma.
5	CngOut	Si las dos palabras recibidas coinciden, cambia la palabra que va al módulo de configuración de registros.
6	Delay2	Espera la respuesta de validación del sistema de configuración de registros.
7	Acknowledge	Determina la palabra de confirmación de dato válido o inválido dependiendo de la validación que se realiza en el sistema de configuración de registros.
8	Delay3	Retardo para esperar la respuesta de validación del sistema de configuración de registros.
9	Delay4	Retardo para esperar la respuesta de validación del sistema de configuración de registros.
10	SndAckNak	Envía la respuesta de confirmación y validación al usuario.

Tabla 3.9 Estados de la máquina de validación de datos.

Este sistema es crítico para el correcto funcionamiento del dispositivo, dado que configura los registros que controlan todo el sistema. Está basado en cuatro instrucciones principales, tres de las cuales corresponden a los registros del sistema y una última que corresponde al comando de inicio del sistema. Sus funciones se resumen en la tabla 3.22.

ASCII	Número (Hex)	Instrucción
DC1	0X11	Cargar canales habilitados.
DC2	0X12	Cargar un número de ciclos de escritura por ensayo.
DC3	0X13	Cargar registro del submuestreador.
DC4	0X14	Iniciar adquisición.

Figura 3.22 Comandos empleados en el sistema de configuración de registros.

Mientras no se haya completado el proceso de calibración no puede cambiar de estado la máquina de estados de inicio, posteriormente debe recibir una instrucción que corresponda a cualquiera de los comandos antes mencionados. Si la recibe, pasa a un estado de configuración en el que configura el registro correspondiente. Si el dato es inválido reinicia el proceso de configuración. El diagrama de estados del sistema se muestra en la figura 3.22 y el funcionamiento de cada estado se muestra en la tabla 3.11.

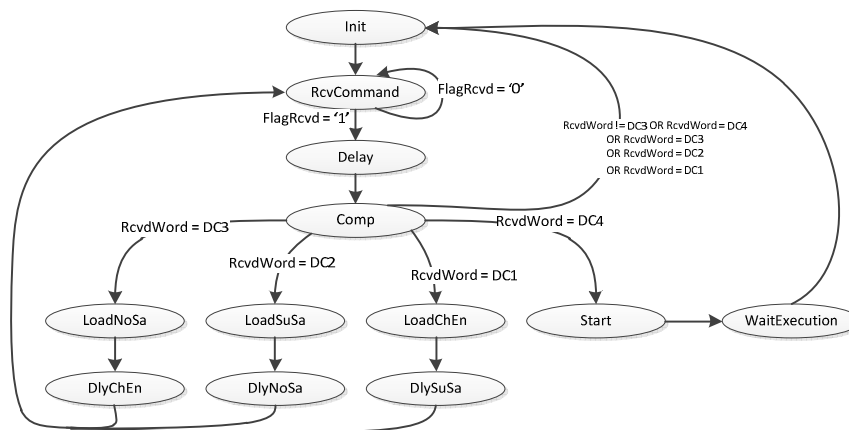


Figura 3.22 Diagrama de estados de la máquina de configuración. Se aprecia el estado de espera de comando, así como los correspondientes a la configuración de cada registro.

No. de estado	Nombre del estado	Función
1	Init	Limpiar los registros.
2	RcvCommand	Espera a que ingrese un dato de configuración.
3	Delay	Retardo para permitir establecer el dato de configuración.
4	Comp	Determina cuál es el registro a configurar o si se trata de un dato inválido.
5	LoadChEn	Espera y almacena un dato en el FIFO de canales habilitados.
6	DelyChEn	Retardo.
7	LoadSuSa	Espera y configura el registro de submuestreo.
8	DlySuSa	Retardo.

9	LoadNoSa	Espera y configura el registro del número de paquetes de 64 muestras por ciclo.
10	DlyNoSa	Retardo.
11	Start	Manda la bandera que confirma que se ha completado el proceso de configuración a la máquina de estados principal.
12	WaitExecution	Espera a que termine la ejecución de la rutina de adquisición.

Figura 3.11 Descripción de los estados de la máquina de configuración de registros.

3.1.1.2.3 Sistema de control principal

Éste se encarga de coordinar el sistema entero. Su diseño se encuentra basado en una arquitectura FSM. El diagrama de estados se muestra en la figura 3.23. El funcionamiento de cada estado se especifica en la tabla 3.12.

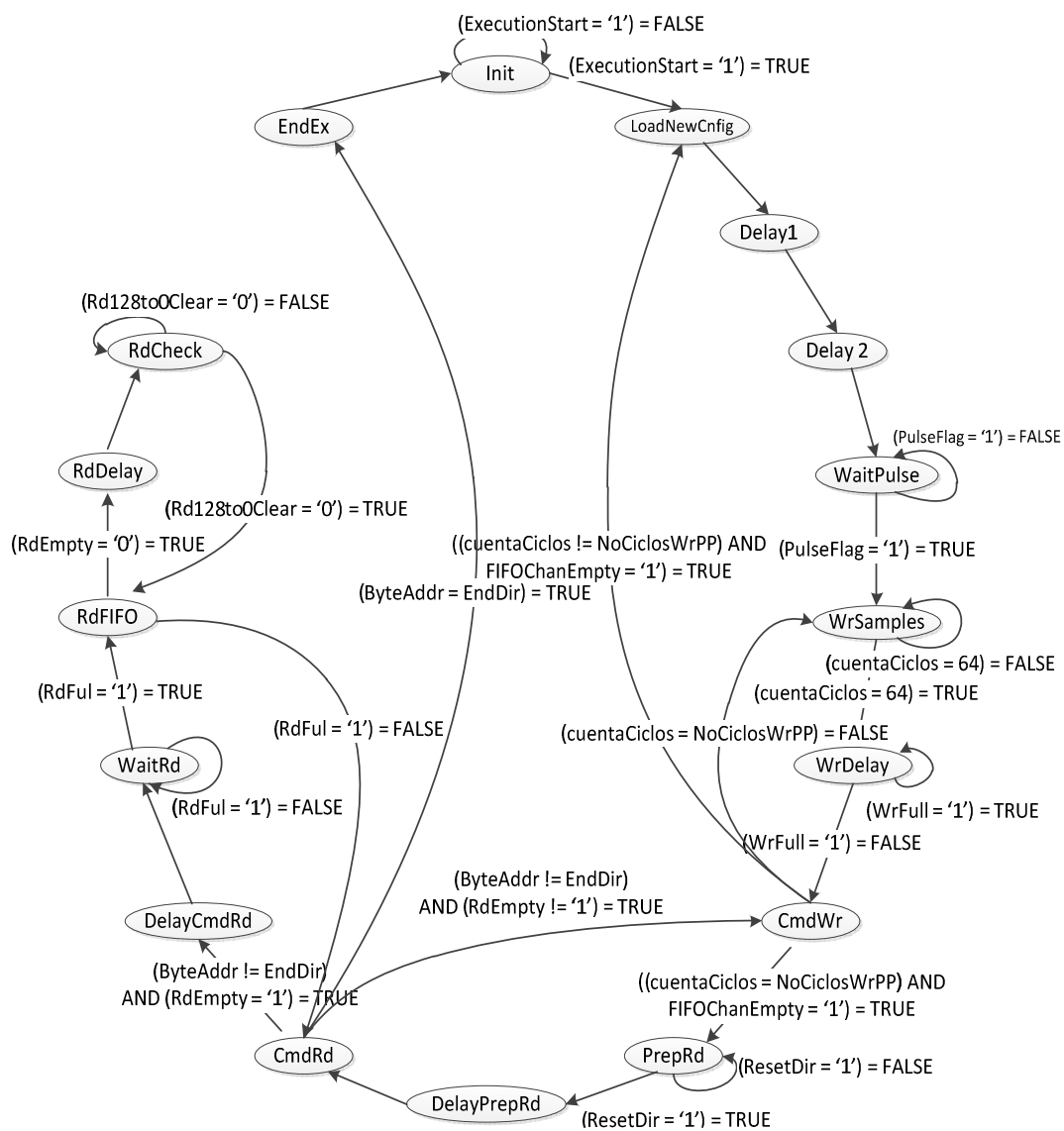


Figura 3.23 Diagrama de estados de la FSM de control.

No. de estado	Nombre del estado	Función
1	Init	Espera a que se realice la configuración del sistema, mientras mantiene en un estado de "Reset" los bloques de entrada de datos del sistema, el cual una vez completado cambia al siguiente estado.
2	LoadNewCnfig	Carga una nueva configuración al registro de canales y pasa al siguiente estado.
3	Delay1	Retardo para evitar que el valor del registro sea metaestable al inicio de la ejecución.
4	Delay2	"Delay2" mismo caso que "Delay1".
5	WatPulse	Espera por la bandera de disparo del pulsador.
6	WrSamples	Verifica cuántos elementos hay en el FIFO de escritura. Si está lleno, deshabilita la escritura, habilita el comando de escribir y va a un estado en que espera que el FIFO comience a vaciarse ("WrDelay"). Por otro lado, si es que el FIFO aún no se encuentra lleno verifica si se está escribiendo en él y de ser así incrementa la cuenta de elementos ("cuentaFIFO"), el estado no cambiaría en este caso.
7	WrDelay	Es un estado en el que se espera a que los elementos que contiene el FIFO se vacíen en la memoria RAM. Posteriormente se va al estado "CmdWr".
8	CmdWr	Verifica que se haya terminado el número de ciclos de escritura definidos por el usuario; de ser así revisa que también se haya terminado de realizar el ensayo si es que el FIFO que contiene la secuencia de canales se encuentra vacío; si esto es verdad, va al estado "PrepRd"; de lo contrario, carga una nueva configuración. Si el número de ciclos de escritura para ese ensayo no ha terminado, entonces se regresa al estado "WrSamples".
9	PrepRd	Realiza un retardo y limpia toda la ruta de datos de entrada (Recepción LVDS, FIFOs, Concatenadores).
10	DelayPrepRd	Quita el estado de "Reset" del boque de entrada y del contador de direcciones.
11	CmdRd	Habilita el comando de escritura, también verifica si se ha llegado a la última dirección en la que se escribió; de ser así, termina la ejecución yendo al estado "EndEx" o, por lo contrario, verifica si el FIFO de lectura se encuentra lleno para habilitar la escritura en memoria.
12	DelayCmdRd	Deshabilita la escritura en el FIFO de comandos y continúa al estado siguiente.
13	WaitRd	Espera a que se llene el FIFO de lectura.
14	RdFIFO	Verifica si el FIFO tiene algún elemento disponible; de ser así, lee un elemento e incrementa la dirección de lectura y el estado cambia a "RdDelay", de lo contrario continúa en el mismo estado.
15	RdDelay	Deshabilita la lectura de datos del FIFO del MCB.
16	RdCheck	Comprueba que la etapa de la etapa de control de flujo y salida de datos se encuentre libre; de ser así, va al estado "RdCheck"; de lo contrario, espera en el mismo estado.
17	EndEx	Es el fin de la ejecución. Levanta la bandera de que se terminó la ejecución, limpia las entradas y regresa al estado "Init" para esperar una nueva configuración.

Taba 3.12 Estados correspondientes a la máquina de control del sistema.

3.2 Diseño de software del sistema

El software que controla el sistema fue implementado mediante un conjunto de scripts realizados en MATLAB. Éstos realizan diversas funciones, sin embargo, a grandes rasgos se pueden clasificar dentro de las siguientes tareas:

- Configurar los módulos del sistema.
- Recibir, organizar y entregar al usuario los resultados del ensayo.

3.2.1 Configuración

La configuración del sistema es una tarea importante y consta de diversos procesos, que se explicarán más adelante. Es importante resaltar que la configuración se encuentra sujeta a las estructuras físicas de las tarjetas, que son impuestas por el diseño del fabricante, lo que impone las limitantes a considerar para el diseño de los sistemas de control.

Las tareas que corresponden a la configuración de los módulos son las siguientes (ver figura 3.24):

- Creación de estructura con parámetros de funcionamiento
- Introducción, validación y ajuste de parámetros
- Generación de cadenas de configuración
- Envío de datos por puerto serie

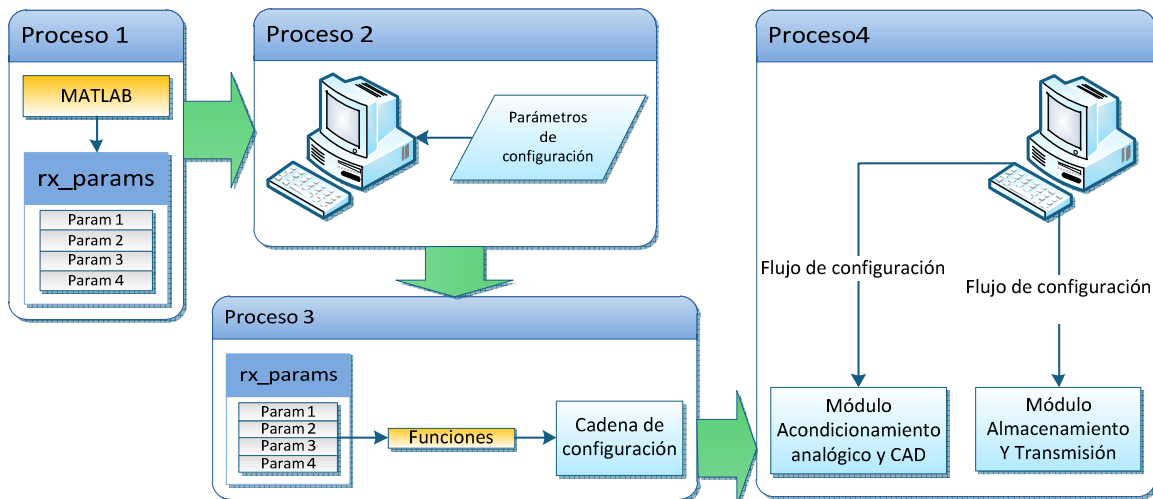


Figura 3.24 Se muestran los procesos involucrados en la configuración de los módulos que conforman el sistema de recepción.

3.2.1.1 Estructura de parámetros

Todos los parámetros de funcionamiento se encuentran encapsulados en una única estructura. Dado que en el *Analog Front End* algunos registros comparten más de una tarea, se facilita la generación de las cadenas de configuración. Ésta contiene los siguientes parámetros importantes:

- Canales habilitados (Secuencias)
- Frecuencias de corte de los filtros pasa bajas y pasa altas
- Patrones de prueba del *Analog Front End*
- Ganancias del PGA y LNA
- Limitadores de voltaje de entrada para las etapas de LNA y PGA
- Impedancia de entrada
- Tiempo de adquisición
- Frecuencia de muestreo

3.2.1.2 Introducción, validación y ajuste de parámetros

Un conjunto de funciones independientes realizan esta función –a cada parámetro le corresponde una función de configuración–, tales afectan únicamente la estructura descrita con anterioridad.

3.2.1.3 Generación de cadenas de configuración y envío por puerto serie

Las características de las estructuras de comunicación varían en forma significativa, sin embargo, los módulos de comunicaciones de ambas tarjetas están basados en puentes USB-UART, lo que permite emplear puertos virtuales COM para establecer comunicación. Esto facilita en gran medida la tarea de establecer comunicación, ya que MATLAB incluye funciones para dicho propósito.

3.2.1.3.1 Módulo de acondicionamiento analógico y conversión analógico-digital

Esta etapa se basa principalmente en una función: se encarga de agrupar las configuraciones de cada uno de los parámetros en los registros de configuración correspondientes. Adicionalmente, tiene una función específica para cada registro, los cuales generan la cadena de configuración correspondiente basada en las características del hardware de comunicación de la tarjeta.

En el caso de la tarjeta del módulo de MACAD, el puente USB-UART tiene una salida paralela, por lo que cada byte enviado al dispositivo genera 8 señales de control para dos dispositivos, así como una terminal de prueba para diagnóstico del dispositivo (ver figura 3.25).

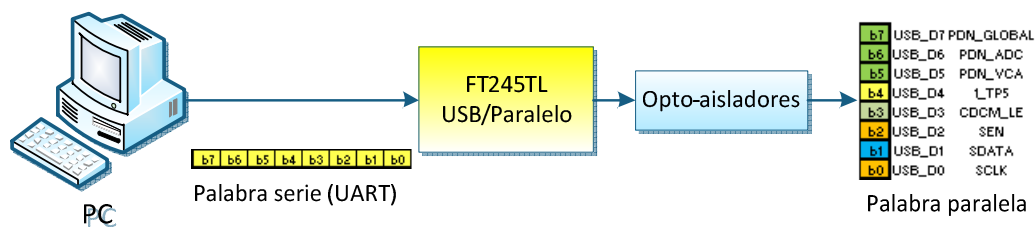


Figura 3.25 Estructura del módulo de configuración del MACAD. Éste incluye un adaptador USB-UART con salida paralela, así como un conjunto de optoaisladores para proteger al equipo de cómputo.

Las señales de configuración y control del módulo de acondicionamiento analógico y conversión analógica digital se resumen en la tabla 3.13.

Bit	Nombre	Función
b7	PDN_GLOBAL	Poner en standby el AFE5808 (Asíncrono).
b6	PDN_ADC	Apagar ADC del AFE5808 (Asíncrono).
b5	PDN_VCA	Apagar VCA del AFE5808 (Asíncrono).
b4	1_TP5	Terminal de prueba 5V
b3	CDCM_LE	Línea SPI de control del CDCM 7005 (No empleado)
b2	SEN	Habilitar envío de comandos (Send-Enable) SPI
b1	SDATA	Línea de datos SPI
b0	SCLK	Reloj SPI

Tabla 3.13 Función de las terminales de salida del módulo USB-Paralelo FT245RL instalado en la tarjeta de acondicionamiento analógico y conversión analógico digital.

La estructura de las palabras que llegan al módulo debe ser matricial (ver figura 3.26) para poder controlar los 8 dispositivos al mismo tiempo. La interfaz de control SPI del AFE5808 contempla palabras de 3 bytes: el primero es la dirección de registros y los dos restantes el contenido. Debido a que también se envía la señal de reloj es necesario duplicar todos los bits como se muestra en la figura 3.27. Una descripción del protocolo SPI se muestra en el apéndice B.

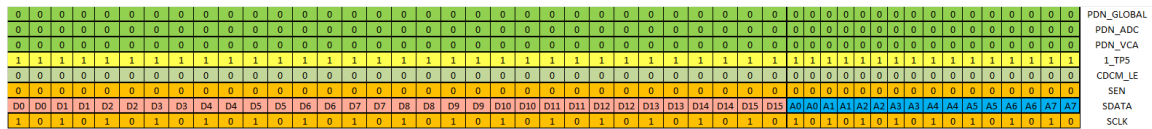


Figura 3.26 Matrices de configuración de la interfaz de control del AFE5808.

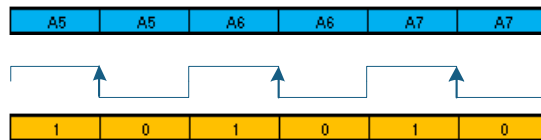


Figura 3.27 Generación de señal de reloj en la palabra de configuración para la interfaz SPI del AFE 5808A.

El árbol de llamadas para el sistema se muestra en la figura 3.28, en la que se aprecia la relación entre las funciones del sistema. En el caso de *set_reg_x*, esta función se repite por cada registro importante del sistema; no todos tienen entradas, ya que algunos de sus valores se consideraron como constantes para dar consistencia y acceso únicamente a las funciones primarias del sistema.

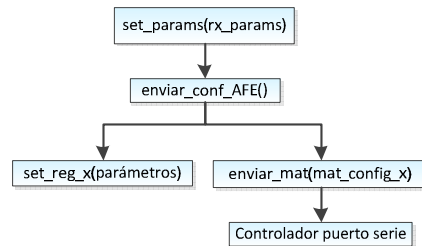


Figura 3.28 Árbol de llamadas del sistema de configuración del AFE5808A, implementado mediante funciones creadas en MATLAB.

El conjunto de funciones con las que interactuará el usuario para esta función se muestra en la figura 3.29. Cada una de la funciones alteran un parámetro en particular de la estructura; cabe mencionar que éstas están diseñadas de manera que sean lo más amigables posible con el usuario para evitar errores.

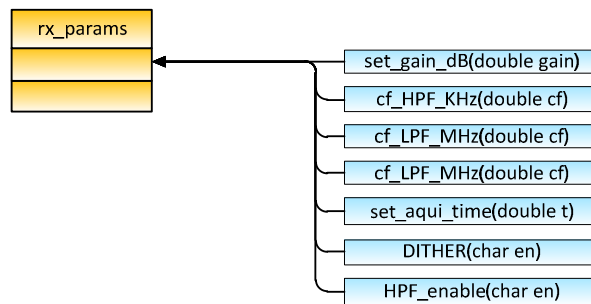


Figura 3.29 Ejemplo de la relación entre la estructura *rx_params* y las funciones que la modifican.

3.2.1.3.2 Módulo almacenamiento y transmisión

El módulo de configuración USB-UART de este módulo hace una conversión directa de USB a UART sin ninguna función adicional. Las características de operación son: tienen entre 2400 y 921600[baud] y cuentan con líneas de control de flujo (CTS, RTS). La estructura se muestra en la figura 3.30.

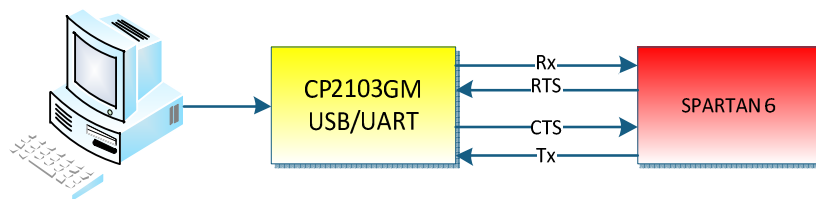


Figura 3.30 Esquema del hardware de comunicación serie del módulo de almacenamiento y transmisión.

Las tareas del módulo de comunicación del MAT son principalmente dos: configurar los registros del módulo y transmitir las muestras capturadas a la computadora. Los esquemas de transmisión son distintos para cada caso, en la figura 3.31 se muestra el esquema para la configuración. Éste es altamente redundante y se envían todos los datos dos veces. El sistema responde a ambos mensajes por medio de un mensaje de confirmación que indica si el dato es correcto y válido.

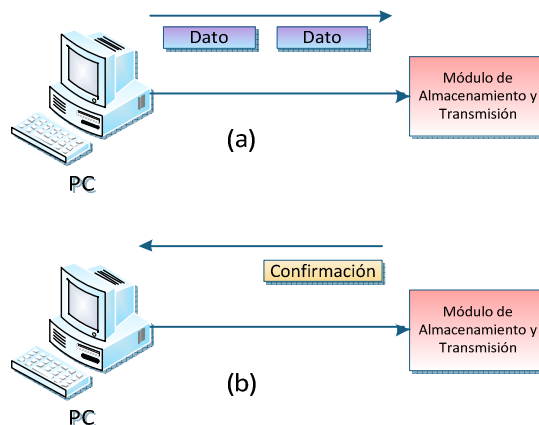


Figura 3.31 Esquema de comunicación para configurar los registros del módulo de almacenamiento y transmisión. (a) Muestra el envío de datos duplicados de la PC al módulo; por otro lado, (b) muestra el mensaje de confirmación con el que responde el módulo.

El esquema de comunicaciones empleado para la recepción (figura 3.32) de las muestras es un tanto distinto, ya que en éste el módulo envía las muestras en 16 paquetes de 8 bits, siendo ahora la PC quien regresa un mensaje de confirmación. En este caso el mensaje de confirmación surge únicamente como un control de flujo; sin embargo, el módulo está diseñado para que a futuro sea fácil integrar un sistema generador de sumas de revisión (Check Sum) para implementar un sistema de redundancia cíclica y evitar errores de transmisión de datos.

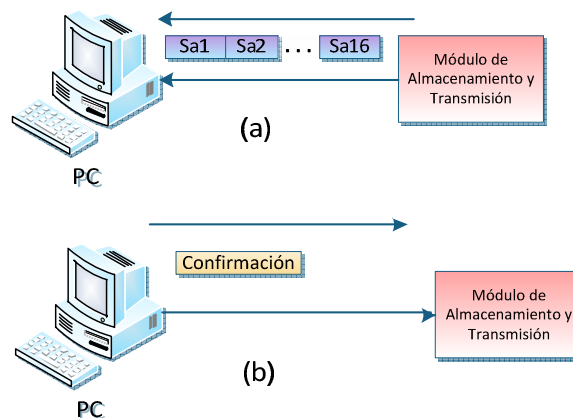


Figura 3.32 Esquema de comunicación para envío de muestras almacenadas en el MAT. (a) Al terminar el ensayo el módulo envía todas las muestras en series de 16 paquetes, posteriormente espera por el mensaje de confirmación que se envía en el proceso (b).

3.2.2 Recepción y organización de datos

En la memoria RAM del MAT, todas las muestras de los diversos canales activos y adquiridos de distintos disparos son almacenadas en forma continua, por lo que es necesario un programa que agrupe, organice y entregue las muestras adquiridas al usuario. El sistema se integra mediante un conjunto de scripts, que deben realizar las siguientes tareas:

- Recibir y controlar el flujo de datos.
- Agrupar los paquetes provenientes de UART en elementos enteros de 16 bits.

- Organizar las lecturas y convertirlas en matrices y vectores útiles para el usuario.

3.2.2.1 Recepción y organización de datos

Esta etapa responde al esquema mostrado en la figura 3.32. En ésta se reciben 16 paquetes de datos. Cada uno de los paquetes corresponde a las muestras de un solo canal, las cuales se almacenaron previamente en un formato de 8 palabras de 16 bits formando una palabra de 128 bits a partir de 8 muestras. El módulo UART maneja palabras de 8 bits, por lo que cada palabra proveniente de la memoria RAM se segmenta en 16 paquetes de 1 byte, partiendo por la mitad cada muestra como se muestra en la figura 3.33.

La transmisión de datos se realiza por bytes, comenzando por el menos significativo para terminar con el más significativo. El último proceso es la espera del MAT por un mensaje de confirmación de la PC; esta configuración aparte de permitir un control de flujo preciso, también permite que en un futuro pueda implementarse un sistema con CRC sin requerir cambios importantes en la estructura del sistema.

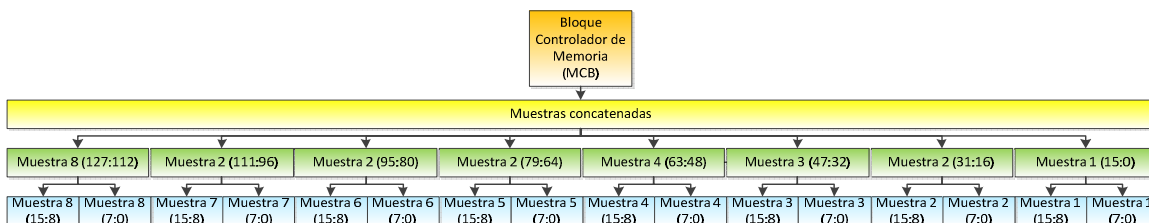


Figura 3.33 Representación del segmentado de palabras para transmisión.

La lectura de datos se realiza mediante el algoritmo presentado en la figura 3.34, en la que se recibe cada paquete de datos y se reconstruyen las muestras.

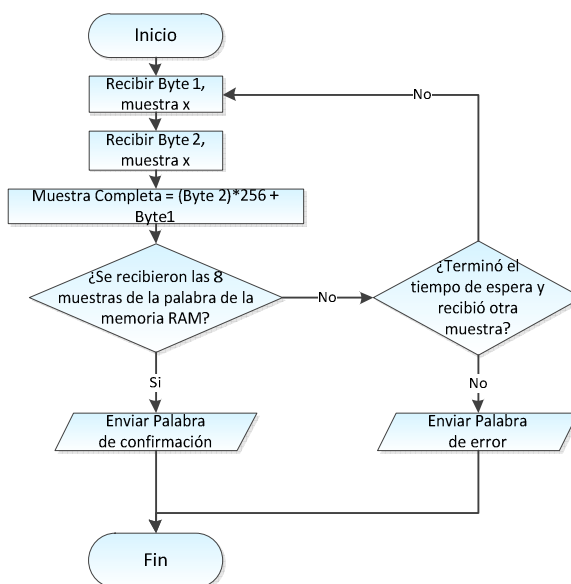


Figura 3.34 Algoritmo correspondiente a la recepción de paquetes de datos serie y reconstrucción de muestras.

3.2.2.2 Agrupamiento de datos

Se realiza mediante un algoritmo que agrupa las matrices de la forma mostrada en la figura 3.35. A continuación se da una breve explicación de las características del agrupamiento:

- A cada canal y por cada disparo se genera un renglón de una de las matrices.
- Las columnas quedan determinadas por el número de elementos establecidos por el usuario.
- El número de matrices representa el número de disparos emitidos por el pulsador.

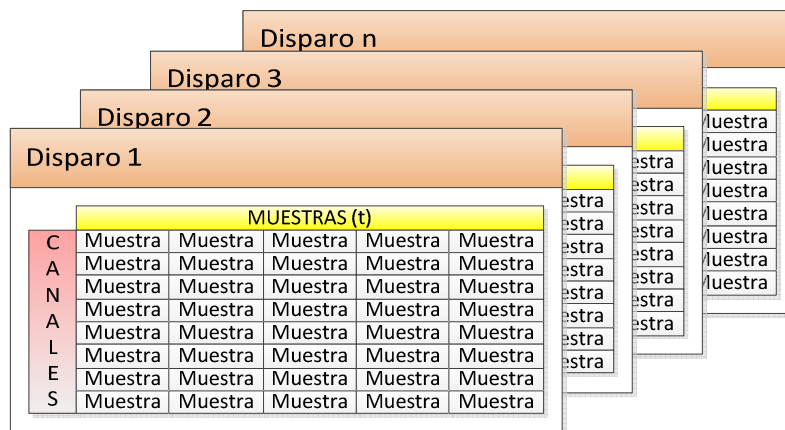


Figura 3.35 Estructura resultante del agrupamiento de datos.

Una representación del algoritmo de agrupamiento se muestra en la figura 3.36, en la que se aprecian todos los procesos involucrados en dicha tarea.

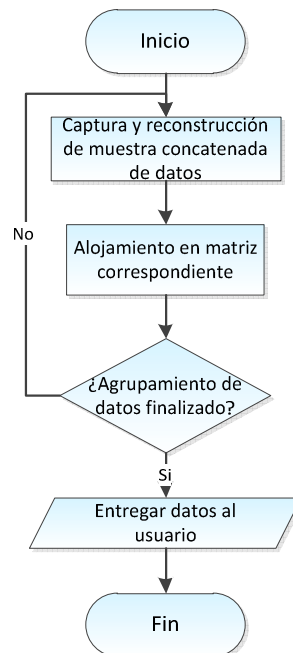


Figura 3.36 Representación del algoritmo de agrupamiento de datos.

Capítulo 4: Funcionamiento del sistema

Habiendo descrito el funcionamiento general de los elementos implicados en el funcionamiento del sistema, se dará paso a mostrar el comportamiento de los diversos componentes, tomando como base un ejemplo de operación. Se explicará el funcionamiento de cada una de las etapas involucradas para lograr una captura exitosa. La forma en que abordará el sistema es la siguiente:

- Configuración del sistema a nivel usuario, generación de las cadenas de configuración y configuración de ambos módulos.
- Procesos efectuados a nivel hardware, resaltando el comportamiento y relaciones de los diversos módulos del hardware implicados en la adquisición, almacenamiento y transmisión de las señales.
- Software implicado en la recepción, reconstrucción y agrupamiento de las muestras capturadas.

En el caso del hardware, la implementación no pudo realizarse de forma exitosa debido a problemas derivados de la temporización por problemas con los tiempos de ajuste y retención en los flip-flop, además de *skew*²⁰ en los relojes, etc. Al momento de este escrito esos problemas no han sido solucionados completamente, por lo cual terminar la implementación se incluirá en un proyecto futuro.

4.1 Ejemplo de funcionamiento

El problema planteado es el siguiente: se desea realizar la adquisición de 5 pulsos en configuración de transmisión, la profundidad deseada es de 30 cm, realizando un corrimiento en la siguiente secuencia:

Disparo	Canales activos
1	8,7,6,5
2	7,6,5,4
3	6,5,4,3
4	5,4,3,2
5	4,3,2,1

Tabla 4.1 Secuencia de ejemplo.

La adquisición se va a realizar en agua y se considera una velocidad de propagación de $v = 1540[m/s]$, lo cual supone un tiempo de adquisición de aproximadamente:

$$t_{aqui} = cv \approx 200[\mu s]$$

²⁰ *Skew* se refiere a problemas relacionados con la fase de relojes, en la cual el reloj entre dos registros se encuentra fuera de fase, de forma que se pueden producir fallas de operación.

4.1.1 Inicialización

Antes de comenzar a introducir un parámetro de configuración es necesario inicializar la estructura del sistema; el comando que realiza esta función es *init_val*, como se muestra en el siguiente ejemplo, que simula el entorno de trabajo de Matlab.

```
>> init_val
>> rx_params

rx_params =

    portAFE: 'COM3'
    portFPGA: 'COM5'
    chan: [1 2 3 4 5 6 7 8]
    cf_LPF: 15
    HPP: 'Y'
    cf_HPP: 100
    test_pattern: 7
    PGA_gain: 24
    PGA_clamp_lv: -2
    PGA_clamp_en: 'Y'
    LNA_gain: 18
    LNA_clmp: 0
    term_imp: 50
    aqui_time_us: 1.2800e+01
    f_sample_MSPs: 40
    DITHER: 'Y'
    no_pulsos: 1
    ciclos_escritura_p_ch: 1
    ciclos_escritura_totales: 8
    velocidad_popagacion: 1540
    profundidad_mm: 1.9712e+01
    factor_submuestreo: 0
```

4.1.2 Introducción de parámetros

El ajuste de parámetros se realiza de acuerdo con la API provista en el apéndice D. A continuación se muestra la introducción de la configuración para una captura exitosa. Se puede apreciar el cambio que sufre la estructura *rx_params*.

```
>> set_f_sample(40)
>> chans_enabled([8,7,6,5;7,6,5,4;6,5,4,3;5,4,3,2;4,3,2,1])
>> HPP_enable('y')
>> cut_HPF_KHz(150)
>> cut_LPF_MHz(10)
>> DITHER('n')
>> set_gain_dB(54)
>> set_aqui_time_us(200)
>> setPortFPGA('COM5')
>> setPortAFE('COM3')
>>
>> rx_params

rx_params =

    portAFE: 'COM3'
    portFPGA: 'COM5'
    chan: [5x4 double]
    cf_LPF: 10
    HPP: 'Y'
    cf_HPP: 150
    test_pattern: 7
    PGA_gain: 30
    PGA_clamp_lv: -2
    PGA_clamp_en: 'Y'
    LNA_gain: 24
    LNA_clmp: 0
    term_imp: 50
    aqui_time_us: 2.0480e+02
```

```
f_sample_MSPs: 40
DITHER: 'N'
no_pulsos: 5
ciclos_escritura_p_ch: 4
ciclos_escritura_totales: 20
velocidad_popagacion: 1540
profundidad_mm: 3.1539e+02
factor_submuestreo: 3
```

4.1.3 Inicio de adquisición

Posterior a la adquisición se realiza la captura de los datos. Para la prueba del algoritmo de agrupamiento se realizó un vector de prueba que simula ser una lectura, debido a que al momento de reportar el circuito no se logró implementar exitosamente el MAT.

```
Mat(:, :, 1) =
Columns 1 through 21
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257  257
  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514  514
  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771  771
...
```

4.2 Funcionamiento de la etapa de configuración del sistema

4.2.1 Software de configuración

Las tareas del software involucradas en la configuración del sistema son:

- Introducción de comandos por parte del usuario
- Generación de las cadenas de configuración (MAACAD)
- Configuración del MAT

4.2.1.1 Configuración del MAACAD

4.2.1.1.1 Configuración de prueba

Para corroborar el correcto funcionamiento de las funciones implementadas se recurrió al del módulo de evaluación TSW1250, que es el módulo original con el que se provee el módulo de acondicionamiento analógico y la conversión analógico-digital (MAACAD), y que permite evaluar el funcionamiento del AFE5808A (figura 4.3) y el osciloscopio de señal mezclada Tektronix MSO-2024 (figura 4.4).

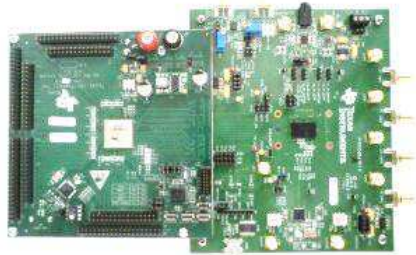


Figura 4.3 Sistema de evaluación AFE5808A.



(a)



(b)

Figura 4.4 Configuración para pruebas con analizador lógico del MSO-2024. Se emplearon las terminales de prueba mostradas en (a). El sistema de prueba completo se aprecia en (b).

4.2.1.1.2 Funciones implementadas y captura de señales de configuración

Comprende la generación de las cadenas de configuración y el envío de cada una de las palabras SPI para configurar el AFE 5808A. En total se configuraron los registros mostrados en la tabla 4.2

Número de registro	Modificable por el usuario		Función
	Sí	No	
0	x		Reset del sistema.
1	x		Habilitar/deshabilitar canales del ADC, habilitar HPF Digital (no accesible al usuario), apagar ADC completo (no accesible al usuario), LVDS output rate 1x (no accesible al usuario), reloj monopolar/diferencial (no accesible al usuario).
2	x		Habilitar/deshabilitar canales LVDS, no promediado de canales (no accesible al usuario); latencia default (no accesible al usuario); patrones de prueba.
3	x		No invertir canales (no accesible al usuario), no sustracción de offset (no accesible al usuario), no ganancia digital, factor de serialización 14x (no accesible al usuario), referencia interna ADC (no accesible al usuario).
4		x	Resolución ADC 14 bit (no accesible al usuario), formato de salida: offset binario (no accesible al usuario), LSB primero (no accesible al usuario).
10		x	Patrón de sincronización en los patrones de prueba (no accesible al usuario).

13		x	No ganancia, offset canal 1 (no accesible al usuario).
15		x	No ganancia, offset canal 2 (no accesible al usuario).
17		x	No ganancia, offset canal 3 (no accesible al usuario).
19		x	No ganancia, offset canal 4 (no accesible al usuario).
21		x	Deshabilitar HPF digital canales 1 a 4, k del filtro (no accesible al usuario).
25		x	No ganancia, offset canal 5 (no accesible al usuario).
27		x	No ganancia, offset canal 6 (no accesible al usuario).
29		x	No ganancia, offset canal 7 (no accesible al usuario).
31		x	No ganancia, offset canal 8 (no accesible al usuario).
33		x	Deshabilitar HPF digital canales 5 a 8, k del filtro (no accesible al usuario).
51	x		Registro PGA: frecuencia de corte FPB, deshabilitar supresión de offset, habilitar limitador de voltaje, ganancia.
52	x		Deshabilitar configuración de impedancia activo por canal (no accesible por el usuario), habilitar control activo de impedancia de entrada, limitador de voltaje LNA, valor de limitación de voltaje, ganancia del LNA, desactivar control de ganancia por canal del LNA (no accesible al usuario).
59	x		Frecuencia de corte FPA LNA, no atenuación digital (no accesible al usuario), deshabilitar atenuador TGC digital (no accesible al usuario), apagar sumador CW (no accesible al usuario), modo de prueba PGA (no accesible al usuario).
66	x		DITHER para disminuir ruido de cuantización.

Tabla 4.2 Funciones implementadas para la configuración del AFE5808A.

En las cadenas de configuración que recibe el MAACAD la mayoría de los elementos son constantes, con excepción de los bits correspondientes a la interfaz SPI del AFE5808A, que se puede apreciar en la figura 4.5. Los comandos mostrados corresponden a los registros de mayor relevancia en el funcionamiento del circuito; el resto de los comandos mencionados en la tabla 4.2 se configuraron para evitar problemas de funcionamiento debidos a fallas en el circuito al momento del encendido que pudiesen evitar que los registros tomen sus valores por default .

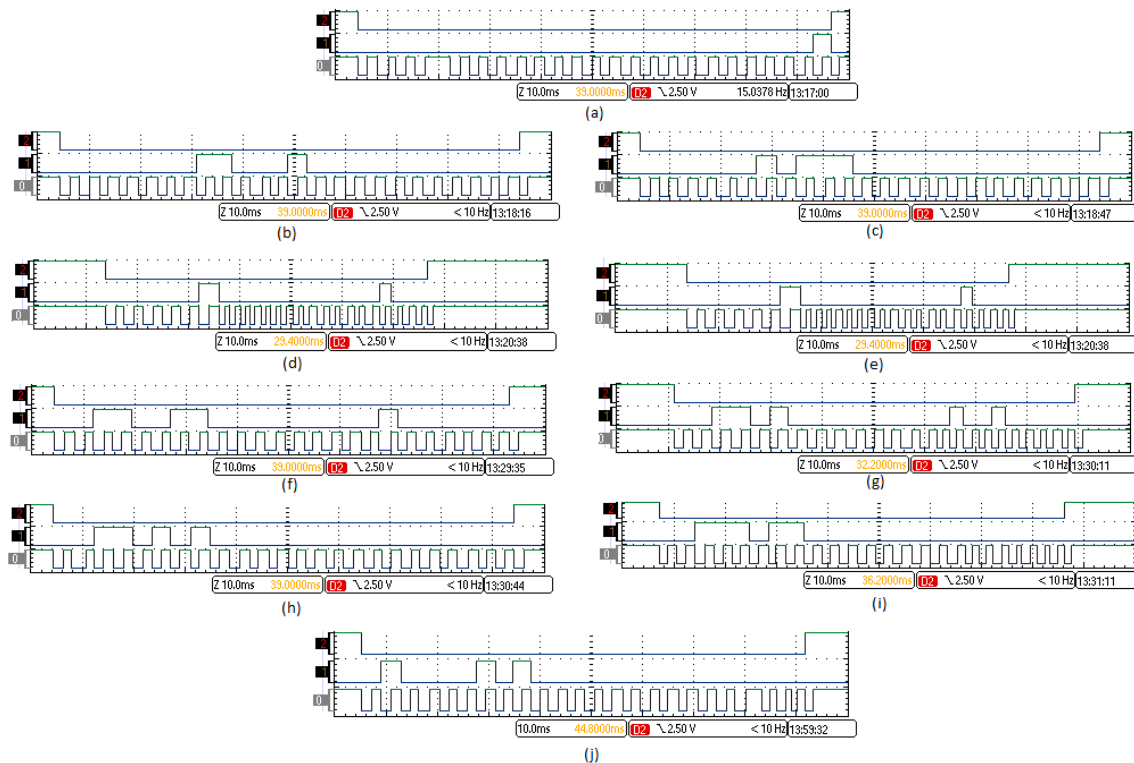


Figura 4.5 Capturas del analizador lógico. Se registró la señal incidente en el módulo de configuración del AFE5808A a través de las terminales dedicadas, con el osciloscopio de señal mezclada Tektronix MSO 2024. Las configuraciones de los registros son las siguientes: (a) registro 0 (reset), (b) registro 1, (c) registro 2, (d) registro 3, (f) registro 51, (g) registro 52, (h) registro 53, (i) registro 66.

4.2.1.2 Configuración del MAT

A diferencia de la etapa de configuración del MAACAD, no se cuenta terminales que permitan realizar la adquisición de las señales que recibe o emite el módulo; sin embargo más adelante, en la sección 4.1.2.1, se encontrarán algunas imágenes relacionadas con la simulación de la etapa de configuración desde una perspectiva del hardware, en la cual se simula el proceso de configuración.

4.2.2 Hardware de configuración

El hardware diseñado en esta etapa es el correspondiente al sistema de almacenamiento y transmisión.

Como se mencionó en el capítulo 3, el hardware de configuración está constituido por dos módulos principales:

- El primero se encarga de realizar la verificación de que los datos entrantes se recibieron correctamente, así como de enviar la confirmación de su validez a la computadora.

- El segundo verifica que las instrucciones recibidas sean correctas y se encarga de configurar los registros correspondientes.

4.2.2.1 Verificación de datos de configuración

Ésta es una de las tareas más críticas para el funcionamiento del sistema, ya que de no realizarse correctamente el sistema funcionará de una forma inesperada. Se hace énfasis en que su funcionamiento sigue un esquema redundante, como se muestra en la figura 4.6.

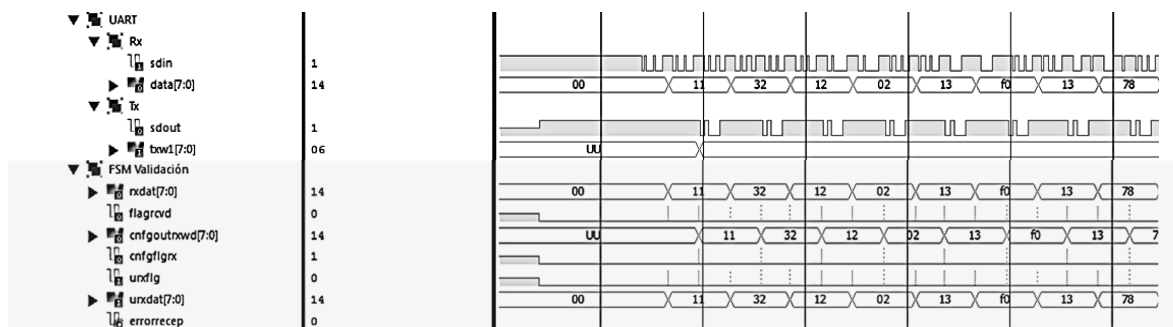


Figura 4.6 Proceso de validación de datos de configuración del MAT. La recepción de una señal redundante que llega en formato serie al puerto *sdin*. En la figura, la entrada es la correspondiente al registro “DC1” (0x11) , correspondiente al registro de número de ciclos de escritura con el valor 0x32 (50 en decimal); después se carga el registro “DC2” (0x12), correspondiente al factor de submuestreo con el valor de dos; y, finalmente, el FIFO de canales habilitados “DC3” (0x13) se carga con un 0xF0, que corresponde a los canales 7, 6, 5 y 4 activos. Para cada par de palabras entrantes el sistema responde mediante una palabra de confirmación “ACK” (0x06) para indicar que el dato de configuración es correcto mediante la señal *sdout* entre los dos marcadores finales.

4.2.2.2 Configuración de registros

Si el sistema de verificación de datos de configuración determina que los valores recibidos se recibieron correctamente y los envía a la etapa de configuración de registros (figura 4.7). Los tres procesos principales que sigue este sistema son:

- Comprobar que la palabra recibida sea válida y enviar la confirmación de la misma al sistema de verificación de datos de configuración.
- Una vez corroborada la validez de una palabra de configuración se envía al registro correspondiente.
- Si el sistema no ha concluido, la configuración mantiene la máquina de control principal en un estado de RESET. Una vez concluido este proceso se retira dicho estado y comienza la adquisición.

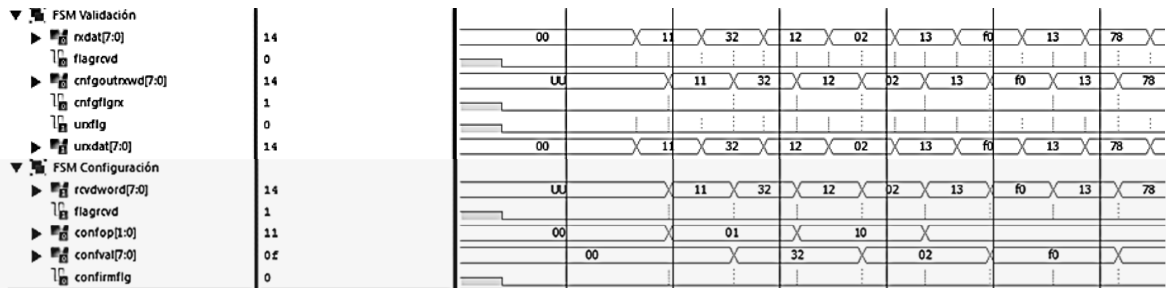


Figura 4.7 Proceso de configuración de registros. La señal *confop*, ubicada en la sección *Registros*, es la encargada de especificar el registro a configurar; *confval* es la encargada de especificar el valor de configuración de dicho registro. Finalmente, la señal *confirmflag* es la encargada de cargar el valor en el registro correspondiente.

4.2.2.3 Registros de configuración

Son los principales elementos que rigen el comportamiento del sistema, como se ve en la figura 4.8, y su funcionamiento se mantiene constante a lo largo de la ejecución de la secuencia de adquisición, con excepción del registro de canales habilitados, cuyo valor cambia para cada disparo.

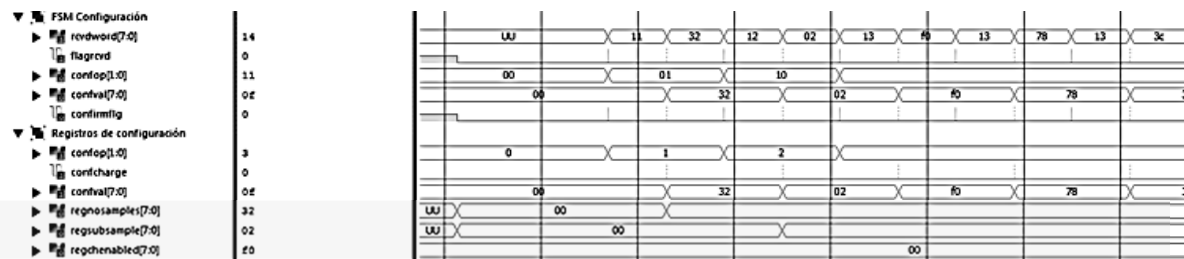


Figura 4.8 Carga de valores de los registros de configuración.

El registro de canales habilitados tiene un FIFO asociado, lo que le confiere la capacidad de cambiar los canales activos durante la ejecución de la adquisición, como se muestra en 4.9. Este proceso es controlado por el sistema central, que tiene acceso directo a la señal de lectura del FIFO de canales, el cual tiene capacidad hasta para 1024 secuencias.

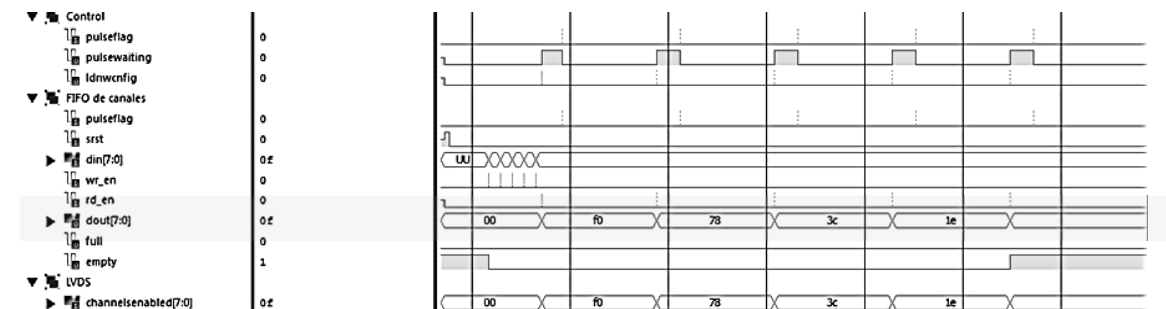


Figura 4.9 Cambio del registro de configuración de canales a lo largo de la captura. El FIFO que contiene la secuencia de canales envía un nuevo valor al registro cada vez que recibe la señal de lectura a través de la señal *loadnewconfig* comandada por la FSM de control. La máquina determina la carga de una nueva secuencia cada vez que se termina la adquisición correspondiente a cada disparo, posteriormente espera el siguiente pulso. Este ciclo se repite hasta que se hayan vaciado todos los elementos del FIFO (*empty* = '1').

4.2.3 Almacenamiento de datos

Durante la adquisición de datos, la computadora se encuentra en estado de espera y los dispositivos activos son únicamente el MAACAD y el MAT; sin embargo, la configuración del MAACAD es una constante a lo largo de toda la adquisición, debido a que su reconfiguración limitaría en gran medida la velocidad de adquisición, ya que las únicas conexiones físicas entre ambos módulos son las de las señales LVDS, no dejando ninguna libre para la configuración del módulo. Esto se soluciona manteniendo todos los canales activos y haciendo la deshabilitación únicamente en el MAT, el cual ignora las señales de los canales inactivos en cada secuencia de adquisición.

Al igual que en el capítulo 3, la secuencia en que se analizará el sistema será la misma que sigue la ruta de datos, concluyendo con las estructuras de control.

El bloque de entrada de datos sólo está activo durante la adquisición de datos, el resto del tiempo se mantiene en un estado de *reset*, también se limpia su contenido cada vez que se va a realizar un nuevo disparo para evitar errores debidos a información anterior, la cual podría ser altamente perjudicial, dado que la recuperación de los datos está basada en su posición y el orden de almacenamiento.

En la figura 4.10 se aprecian las condiciones antes descritas. Inicialmente el sistema se encuentra en espera del inicio de la adquisición; posteriormente se retira el estado de *reset* y se realiza la lectura de cada disparo.



Figura 4.10 Comportamiento del *reset* a lo largo de la espera y ejecución de las secuencias de adquisición de datos.

4.2.3.1 Bloque de entrada

Este bloque se encarga de recibir las muestras provenientes del MAACAD; cada uno de los elementos que se mostrarán en esta sección se repiten para cada canal de entrada de datos.

4.2.3.1.1 Bloque LVDS

Es el primer elemento en la cadena de adquisición de datos, su desempeño es importante para alcanzar un óptimo desempeño. Éste considera un factor de serialización 14:1 con entradas diferenciales, las cuales se convierten en monopolares mediante un búfer. El bloque se mantiene activo todo el tiempo de la adquisición, independientemente del factor de submuestreo como se ve en la figura 4.9.

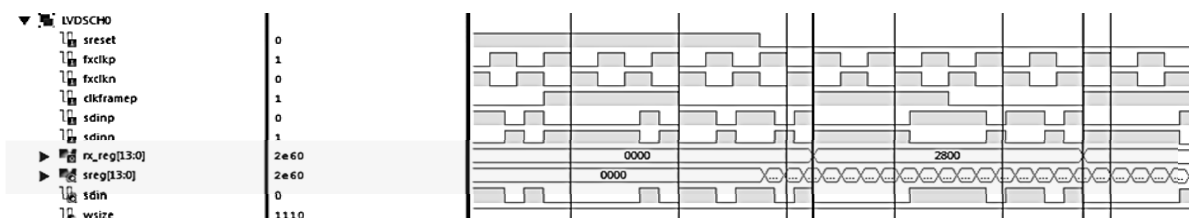


Figura 4.9 Funcionamiento de bloque de captura LVDS. Se aprecia la relación de los relojes; cada flanco de

subida de *frameclock* se actualiza el valor de salida, lo cual se ve reflejado en la señal *rx_reg*.

4.2.3.1.2 FIFO LVDS

Como se mencionó en el capítulo 3, el FIFO de sincronización maneja dos relojes distintos: por un lado, escribe a la frecuencia del *frame clock* y por el otro lee con el reloj de usuario generado con el PLL *userclk*. El sistema de submuestreo habilita o deshabilita la escritura del sistema, lo que permite un esquema de submuestreo independiente del módulo de recepción. El funcionamiento se aprecia claramente en la figura 4.10.



Figura 4.10 En cada ciclo de *frameclock* se recibe un dato nuevo; si el submuestreador determina que se puede dar acceso al sistema, entonces se habilita la escritura con *wr_en = '1'*; en otro caso dicha señal se mantiene en '0'. El sistema de concatenado es el encargado de realizar la lectura. Es importante contrastar la diferencia en las frecuencias para el sistema de escritura y el de lectura, los cuales son 40[MHz] y 200[MHz].

4.2.3.1.3 Concatenador

La concatenación de las muestras se lleva a cabo mediante una FSM, que realiza la lectura en los siguientes cuatro pasos:

- Verificación del estado del FIFO que almacena las muestras recibidas y la lectura de un elemento.
- Tiempo de espera para el establecimiento de las señales.
- Almacenamiento de la palabra en la posición correspondiente.
- Envío de la palabra concatenada a la cola de espera del multiplexor.

Cada una de estas tareas se realiza mediante un estado en particular, como se mencionó en el capítulo 3 y es posible apreciar gráficamente en la figura 4.11.

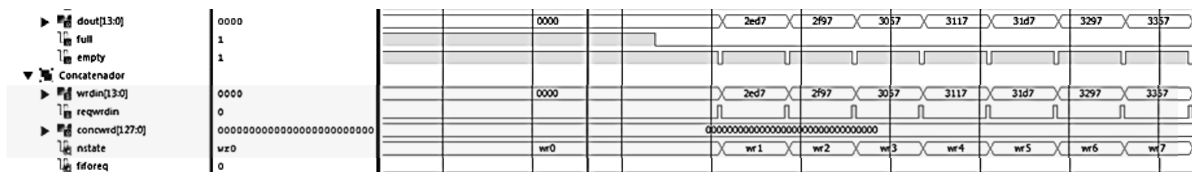


Figura 4.11 Funcionamiento del sistema de concatenado. Las señales *dout*, *full* y *empty* pertenecen al FIFO del receptor LVDS para el canal 1 (nombrado 0 en las señales internas del MAT). Se aprecia que cada vez que

empty indica que hay un dato almacenado, éste manda una señal de lectura, espera un ciclo a que se establezca la señal y lo agrega a la palabra concatenada, finalmente lo envía a la cola de espera del multiplexor cuando la señal *fiforeq* es igual a 1.

4.2.3.1.4 Cola de espera del multiplexor

Se trata de un FIFO de 128 bits con 64 bits de profundidad, almacena las muestras provenientes de cada uno de los canales. La temporización puede verse en 4.12.



Figura 4.12 Comportamiento de la cola de espera del multiplexor para el canal 1. Se aprecia el proceso de escritura, mencionado en 4.2.1.3, así como el de lectura realizados por el sistema de arbitraje mediante la señal *rd_en*.

4.2.3.2 Sistema de arbitraje y Multiplexado

4.2.3.2.1 Árbitro

El sistema de arbitraje manda las señales de lectura a la cola de cada uno de los canales a la vez. En la figura 4.13 se aprecia el comportamiento del sistema a lo largo de toda la secuencia de adquisición (la adquisición de cada uno de los 5 pulsos), posteriormente en la figura 4.14 se ve el funcionamiento detallado de la cola de espera con respecto al arbitraje y el multiplexado para cada canal.

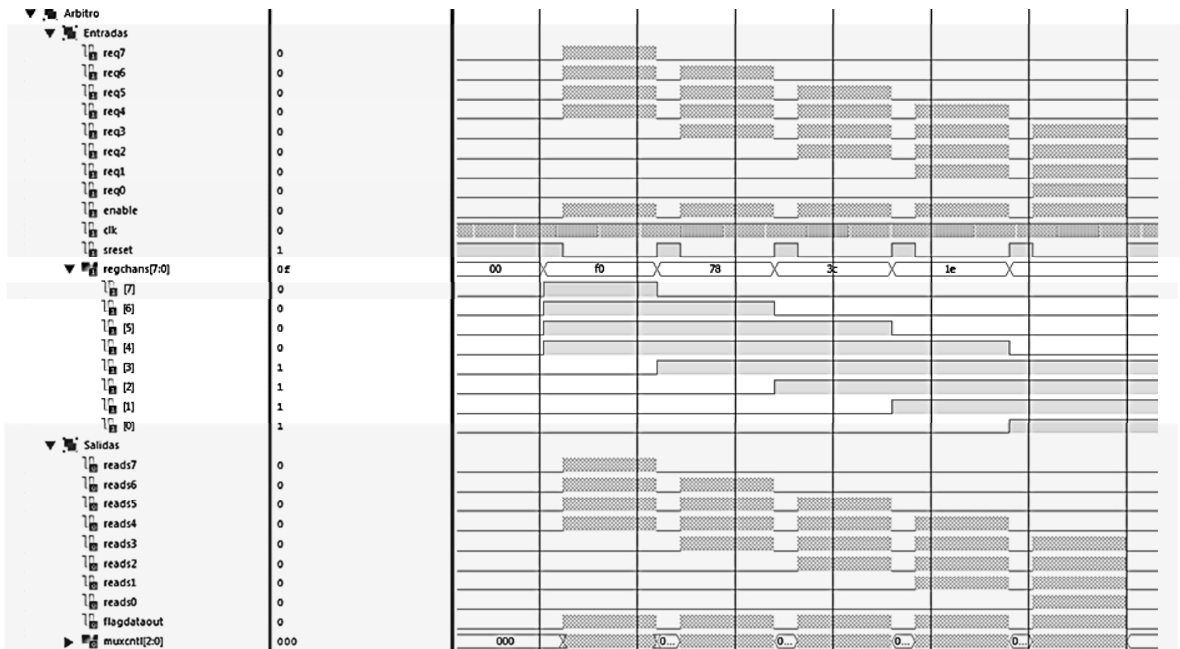


Figura 4.13 Arbitraje y multiplexado para los 8 canales a lo largo de las 5 secuencias de adquisición que siguen la secuencia de corrimiento que se muestra en el registro de canales habilitados (*regchans*) ([7,6,5,4], [6,5,4,3], [5,4,3,2], [4,3,2,1] y [3,2,1,0] respectivamente). La señal *readsX* es la señal de lectura al FIFO de

canales por cada ciclo de escritura y *mux_cnt* es la encargada de ajustar el valor del multiplexor.

El comportamiento en la cola de espera del multiplexor muestra en la figura 4.14 que la señal mediante la que los diversos canales activos piden acceso al MCB es *reqX*, la cual además permite apreciar el tiempo de espera de cada uno de los canales. Finalmente, la señal *flagdataout* indica al siguiente bloque que está enviando una señal del FIFO.

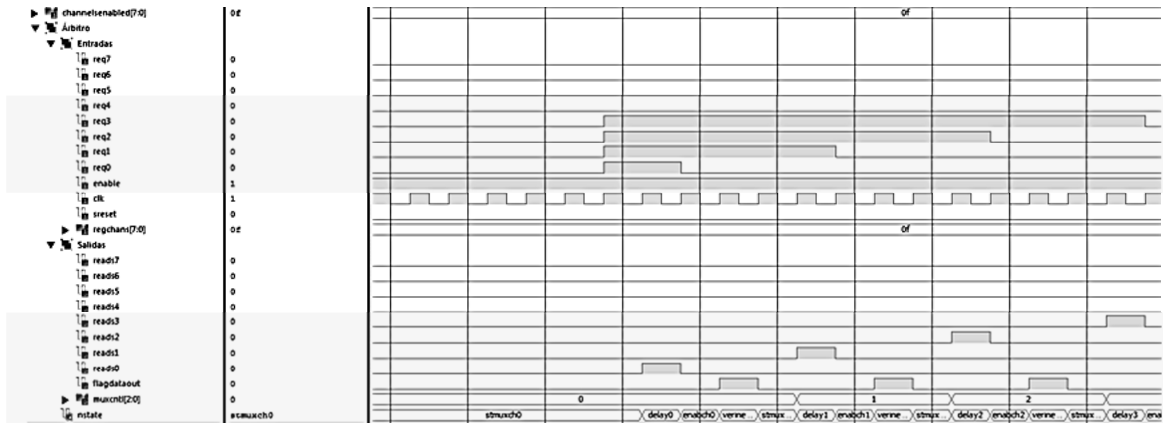


Figura 4.14 Cola de espera del multiplexor. Las señales *req0*, *1*, *2* y *3* corresponden a las banderas de acceso a la memoria. Debido a la estructura del sistema de arbitraje, el tiempo de acceso del canal con índice mayor (3 en este caso) siempre espera uno más, ya que siempre lee primero el canal con el índice más pequeño (0) y así continúa en forma ascendente, además de que todas las señales llegan exactamente al mismo tiempo.

4.2.3.2.2 Multiplexor

El sistema de multiplexado es un esclavo del sistema de arbitraje, que manipula sus señales de control, de forma que primero establece la salida, posteriormente espera un ciclo para que se estabilice y al final pide acceso al siguiente bloque. La figura 4.15 ilustra este proceso.

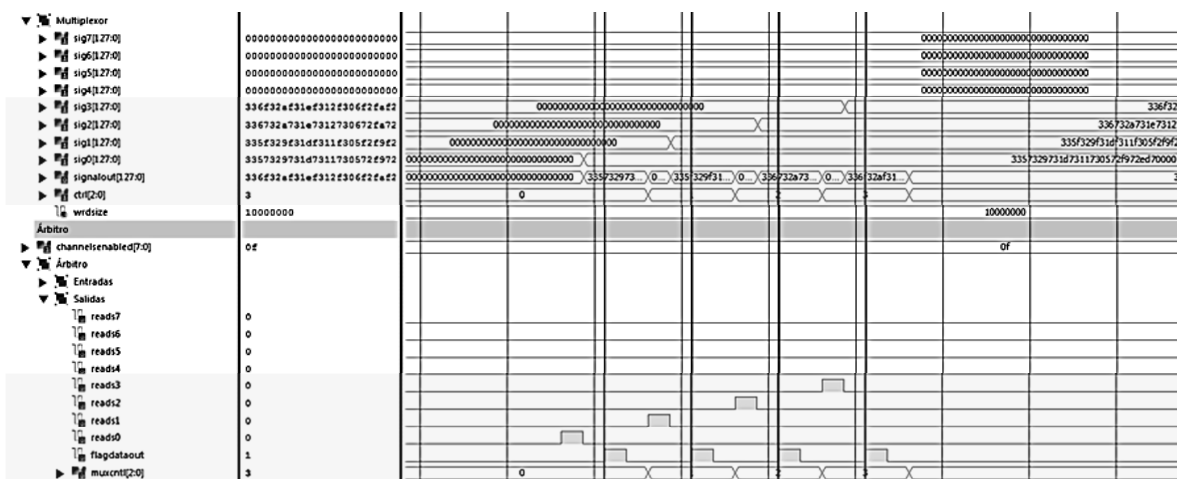


Figura 4.15 Forma de onda de las señales de entrada y salida del multiplexor, así como su relación con el sistema de arbitraje.

4.2.4 Control

El sistema de control durante la etapa de escritura cuenta el número de señales en el FIFO de escritura del MCB. Cuando se tiene la cantidad necesaria de elementos, permite deshabilitar la escritura. En la figura 4.16 se muestra una imagen en la que se presenta parte del proceso y en 4.17 se aprecia un ciclo completo en el que se llena y vacía el FIFO de escritura del MCB.

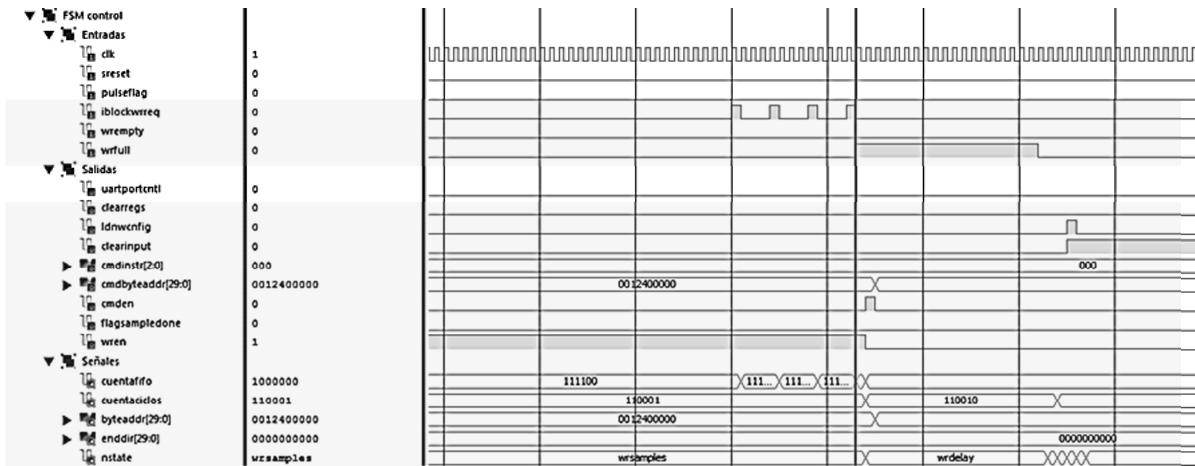


Figura 4.16 Funcionamiento del bloque de control del sistema. Se aprecian los ciclos de lectura en realizados (*iblockwrrreq*) y la ejecución del comando de escritura con *cmden* en el momento en que se encuentra lleno el FIFO (*wrfull* = '1'), asimismo se observa la carga de un nuevo valor de configuración para el registro de canales habilitados (*ldnwcfnfig* = '1').

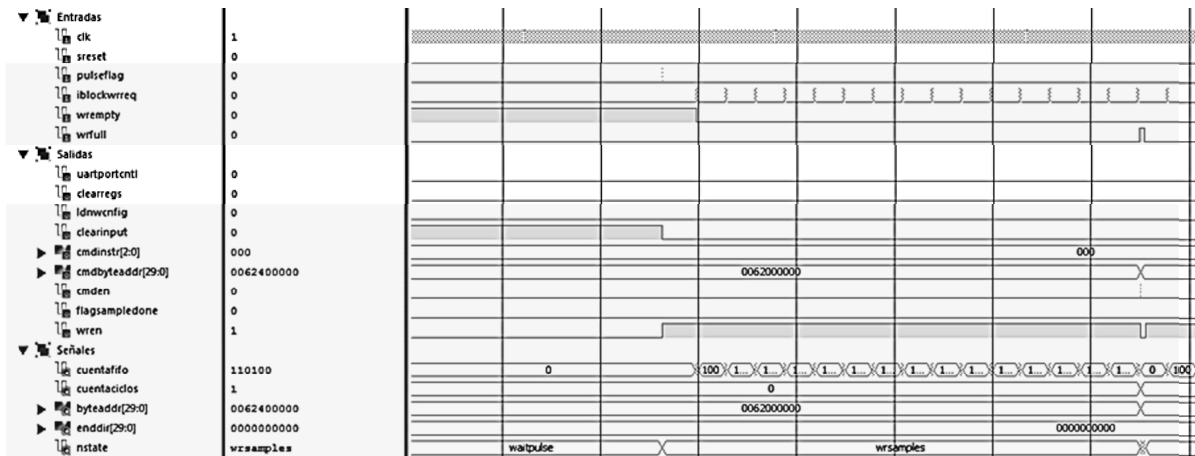


Figura 4.17 Ciclo completo de escritura-lectura del FIFO de escritura del MCB. En ésta se aprecia la escritura intermitente de los datos, ya que se realiza por pequeñas ráfagas coordinadas por el sistema de arbitraje.

4.2.4.1 Controlador de memoria

El MCB durante la lectura únicamente introduce elementos en el FIFO de escritura y se ejecuta el comando. El sistema se encarga de enviar los datos a la memoria RAM automáticamente como se muestra en la figura 4.18.



Figura 4.18 Se muestra la administración de recursos de la memoria RAM. La señal *ddrdata* es el bus de datos bidireccional de datos de la memoria; *ddraddr* es el bus de direcciones.

4.2.5 Proceso de lectura y envío de muestras

Es el último proceso realizado en el MAT, cuyos procesos pueden repetirse una o más veces dependiendo de la configuración del usuario (figura 4.19). Los pasos que sigue esta etapa son:

- Lectura de 64 elementos de la memoria RAM y su almacenamiento en el FIFO de lectura de datos del MCB
- Lectura de cada uno de los elementos del FIFO y su envío al módulo de segmentación y salida de datos
- Envío de cada elemento a la PC

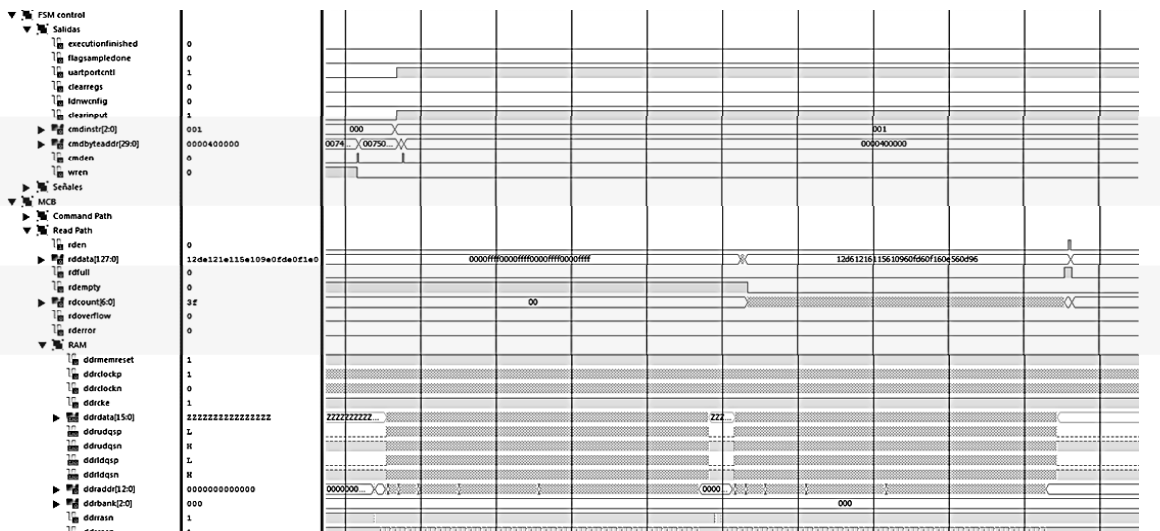


Figura 4.19 Inicio del proceso de lectura. Se aprecia que en el momento en que termina la lectura (*executionfinished* = '1') el sistema pone el bloque de entrada en un estado de *reset*. También es posible observar que después se comienza la lectura de datos de la memoria RAM hasta llenar el FIFO del MCB completamente (*rdfull* = '1'); posteriormente el sistema comienza el proceso de segmentación y envío de muestras, por lo que se lee el primer dato del FIFO (*rd_en* = '1').

4.2.5.1 Segmentación y envío de muestras

Cuando termina el proceso de captura arranca esta sección, la cual realiza tres procesos fundamentales:

- Partir la palabra entrante en 16 segmentos de 8 bits para su envío (ver figura 4.20).
- Enviar una por una cada palabra segmentada en paquetes de 16 (figura 4.21).
- Esperar una palabra de confirmación de la PC (figura 4.21).

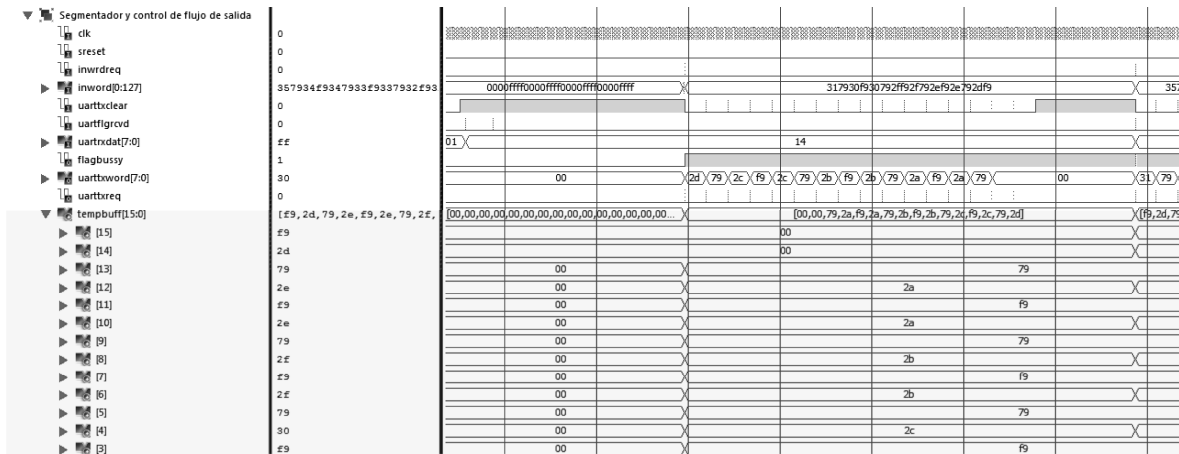


Figura 4.20 Segmentación de palabras de 128 bits en palabras de 8 bits para transmisión. Se aprecia que parte la palabra incidente de 128 bits en 16 palabras de 8 bits.

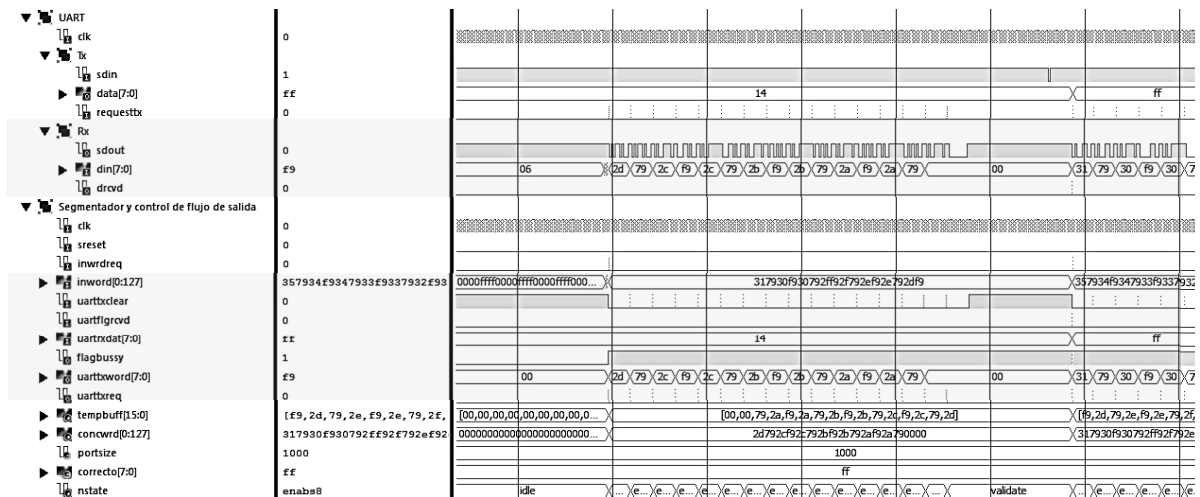


Figura 4.21 Transmisión de paquetes y recepción de señales de confirmación. Se puede observar cómo se envían todas las muestras almacenadas en la memoria a través de *sdout*; posteriormente se recibe el paquete de confirmación (0xFF) a través de *sdin*, por lo que el sistema continúa con la transmisión del siguiente paquete.

Capítulo 5: Conclusiones y trabajo futuro

A lo largo de los capítulos anteriores se presentó el planteamiento, la conceptualización y el diseño de un sistema de captura de señales de ultrasonido, que es útil para fines de investigación y desarrollo en los diversos proyectos llevados a cabo en el laboratorio de acústica de la facultad de ciencias de la UNAM.

El proceso siguió una metodología de modularización de sistemas, facilitando la implementación y reutilización del mismo. La estrategia empleada para la selección de los componentes fue la de buscar elementos programables VLSI, tanto en la etapa de acondicionamiento analógico y conversión analógico-digital, como en la circuitería digital auxiliar en el almacenamiento y control de flujo para transmisión de datos, acortando el tiempo de diseño y mejorando las características de operación del mismo en cuestiones relacionadas con el consumo de energía, el ruido del sistema y las características relacionadas con el control de los dispositivos.

La realización del hardware en FPGA representa una estrategia importante que posibilita la implementación de sistemas con características óptimas para la implementación de circuitos digitales que sean capaces de operar con las características necesarias en cuestiones tales como la cantidad de lógica disponible, circuitería de reloj (máxima frecuencia de operación, tiempo de propagación y circuitería de acondicionamiento tal como PLLs o DCMs) y manejo de periféricos (tanto en cuestiones relacionadas con los estándares de entrada/salida como el número de puertos disponibles); asimismo, permite la reutilización de las descripciones de hardware y del dispositivo para aplicaciones futuras.

La selección del módulo de evaluación del FPGA se basó en que fuese compatible con el estándar FMC, ya que éste permite una fácil migración entre las diversas familias de los FPGA de Xilinx que disponen módulos que cumplen con dicho estándar, lo que permite la reutilización de la tarjeta debido a que existe una gran cantidad de módulos útiles para aplicaciones de alto rendimiento.

El proceso de desarrollo de la descripción de hardware requirió del uso de las herramientas provistas por Xilinx (ISE design suite 14.5) que sigue un flujo de diseño EDA.²¹ Al momento de este escrito, la fase de implementación no se pudo concluir en forma satisfactoria, debido a problemas relacionados con violaciones de temporización que no se pudieron solucionar por falta de tiempo dentro del programa del desarrollo, ya que este tipo de problemas representa una mayor complejidad en la programación de tales dispositivos sobre otras tecnologías. Como parte de mi proyecto futuro se encuentra realizar la depuración de errores en dicha etapa.

²¹EDA (Electronic Design Automation): Estrategia de desarrollo de sistemas electrónicos, ampliamente utilizada en la industria de los dispositivos electrónicos. Ésta emplea un conjunto de herramientas de síntesis, acomodamiento e interconexión y depuración para lograr una implementación exitosa.

Las descripciones de hardware fueron realizadas en lenguaje VHDL, el cual es ampliamente utilizado por los diseñadores de hardware. Asimismo, se buscó que el diseño agrupara módulos de diseño cuyo flujo de datos fuera predecible y fácil de entender, así como fácil de integrar en cadena con otros módulos complementarios que permitan a futuro una mejora en el rendimiento y las funciones del circuito; lo anterior podría expresarse en agregar un sistema generador de sumas de chequeo (CRC) para detección de errores en el envío de las muestras o la implementación de un filtro pasa bajas digital posterior a la recepción de la etapa de recepción y previo a la etapa de su submuestreo para evitar aliasing.

Por otro lado, el empleo de un *Analog Front End* en la etapa de amplificación, filtrado y conversión A/D permitió alcanzar un sistema de bajo costo, alto rendimiento y bajo consumo (máximo 158 mW por canal, según los datos del fabricante) que ocupa un área muy pequeña, que es capaz de adquirir hasta 8 señales de forma simultánea. Cabe mencionar que estos elementos cada vez son más comunes en circuitos de instrumentación digital y abarcan una gran cantidad de aplicaciones (adquisición de señales bioeléctricas, sistemas de captura de señales para PET, CT, etc.).

El módulo de acondicionamiento analógico y CAD supuso el planteamiento de una estrategia de control a partir de los esquemas provistos por el fabricante para el análisis del hardware original y el mapa de registros especificado en la hoja de datos del *AFE5808A*. La implementación fue realizada mediante MATLAB, debido a su extendido uso por los integrantes del laboratorio.

La implementación de las funciones de configuración persiguió realizar una interfaz intuitiva que sea fácil de resumir en un API que permita al usuario programar y adquirir secuencias con un mínimo número de procesos. Para esto únicamente consideró las funciones más básicas de operación, evitando aquellas que implican configuraciones por canal o procesamiento digital de señales, siempre evitando una estructura compleja que sea confusa para el usuario. Como parte del trabajo futuro se podría proveer una interfaz gráfica que permita simplificar aún más el proceso de adquisición.

Un proceso adicional que conforma parte del proyecto futuro se trata de la implementación del módulo pulsador, el cual actualmente se encuentra en proceso de diseño, y la implementación de su correspondiente software de configuración. De este modo se tendrá un sistema completo de emisión/recepción que pueda ser empleado tanto para adquisición de señales de transmisión como pulso-eco.

Como parte de los proyectos que retomarán este trabajo se contempla la creación de un sistema de adquisición de imágenes tomográficas para aplicaciones en el área de ensayos no destructivos, el cual estará basado únicamente en sistemas de procesamiento embebido y será útil para el análisis de integridad de materiales en el campo.

Apéndices

Apéndice A: Comunicación UART [24]

Sus siglas se refieren a Receptor y Transmisor Asíncrono Universal (Universal Asynchronous Receiver and Transmitter, en inglés), y es un circuito que envía información paralela a través de una línea serie. Es común que este tipo de dispositivos se usen en conjunto con el estándar RS-232 en sus características eléctricas, mecánicas y funcionales.

El transmisor es diseñado a partir de un registro de corrimiento especial, el cual tiene carga paralela y realiza sus corrimientos a intervalos regulares, cada uno de estos últimos corresponde a un bit. El inicio de cada palabra se marca por una señal baja. El inicio de cada palabra se marca por un nivel bajo y el fin de la misma por un nivel alto. El tamaño de registro se puede encontrar entre 6 y 8 bits

Por otro lado, el receptor recibe uno a uno los bits de entrada (con una mecánica muy similar a la del transmisor) y reconstruye la palabra de entrada. La estructura de una palabra UART se muestra en la figura B.1.



Figura B.1 Estructura de una palabra enviada por UART.

Transmisor

Este módulo genera un conjunto de señales emitidas a corte a un divisor de frecuencia. Tiene la misma frecuencia de salida que el *baud rate* y realiza los procesos que se muestran a continuación:

- Mientras se encuentra en espera de un paquete de datos mantiene la línea en alto.
- Se envía el bit de inicio (nivel bajo).
- Envío de cada uno de los bits según la frecuencia delimitada por el *baud rate*.
- Corrimiento y envío de cada uno de los bits de la palabra.
- Bit de parada (línea en alto).

Receptor

Al no tener acceso a la señal de reloj del transmisor, el receptor se ve obligado a llevar a cabo un proceso de muestreo, el cual comúnmente se maneja a 16 veces el *baud rate*. Lo que permite

realizar las lecturas en el momento en que la señal ya se haya estabilizado. El esquema de funcionamiento del sistema es el siguiente:

- El sistema se encuentra en estado de espera.
- Detección de bit de inicio (línea en estado bajo).
- Desfasamiento de exactamente medio periodo antes de realizar el muestreo para asegurarse de adquirir cada bit cuando la señal se encuentre en un punto estable.
- Proceso de muestreo de la señal, en el cual cada vez que se cumple un periodo se muestrea un bit.
- Fin del proceso de adquisición y regreso al estado de espera.

Apéndice B: Comunicación SPI

Se trata de un protocolo de comunicaciones serial síncrono, full-dúplex, que permite transferir señales a alta velocidad con un bus de cuatro líneas. Se basa en un esquema de control del tipo maestro-esclavo, las señales de funcionamiento del sistema se resumen en la tabla B.1

Terminales	Función
SCLK	Reloj provisto por el maestro.
CS	<i>Chip Select</i> , marca el dispositivo con el que se está comunicando el maestro.
MOSI/SDI	Salida del Maestro Entrada del Esclavo (<i>Master Output Slave Input</i>) / Dato Entrante Serie (<i>Serial data In</i>).
MISO/SDO	Entrada del Maestro salida del Esclavo (<i>Master Input Slave Output</i>) / Salida Serie (<i>Serial Data Output</i>).

Tabla B.1 Bus de datos empleado para comunicaciones con el protocolo SPI. [25].

La forma en que se lleva a cabo la comunicación es la siguiente [26] (ver figura B.1):

- La señal CS se mantiene en un estado alto en un inicio. Cuando el maestro quiere enviar datos ésta pasa al nivel bajo, quiere decir que se van a iniciar comunicaciones con ese esclavo.
- Cuando inicia la transferencia de datos el maestro activa la línea de reloj (SCLK).
- La información se envía mediante MOSI bit por bit. Ésta es recibida por el esclavo en la terminal SDI y es recibido un bit cada flanco positivo o negativo de SCLK (depende de cada dispositivo). El maestro al mismo tiempo también puede leer datos por MISO.

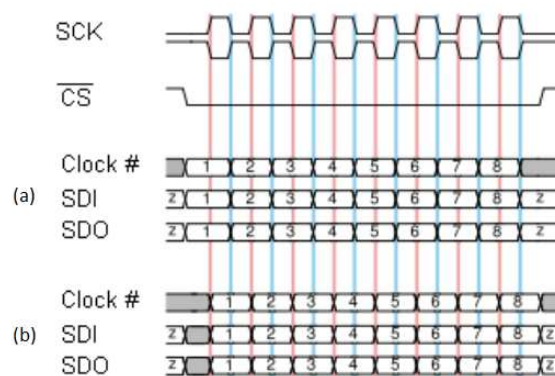


Figura B.1 Diagrama de tiempo de SPI. (a) muestra el esquema en que la SDI y SDO se encuentran en sincronía con el flanco positivo de SCLK. Por otro lado en (b) se muestra el caso alternativo en que la sincronía se realiza con el flanco negativo.

Apéndice C: Nociones básicas de comunicación LVDS

Descripción general

Se trata de un protocolo de comunicaciones digitales ampliamente utilizado en la industria para tareas como visión industrial, sistemas de medición, sistemas automotrices entre otras. Destaca por su alta velocidad de transferencia de datos, ya que según el estándar puede operar hasta 3.125[Mbps]. El swing de voltaje de operación máximo es de hasta 350[mVpp]. Con un offset característico de 1.2 y 3.3 [V] [27] y un consumo promedio de corriente aproximado de 3.5 [mA] si se tienen resistores de terminación con valor de 100[Ω] [28] (ver. figura A.1 [29]). LVDS ofrece las siguientes ventajas sobre otros protocolos de comunicación basados en señales COMOS o LVTTTL:

- Alta inmunidad al ruido
- Mayor velocidad de transferencia
- Menor consumo de energía (figura A.2)
- Más inmunidad al ruido
- Menor interferencia electromagnética (EMI)

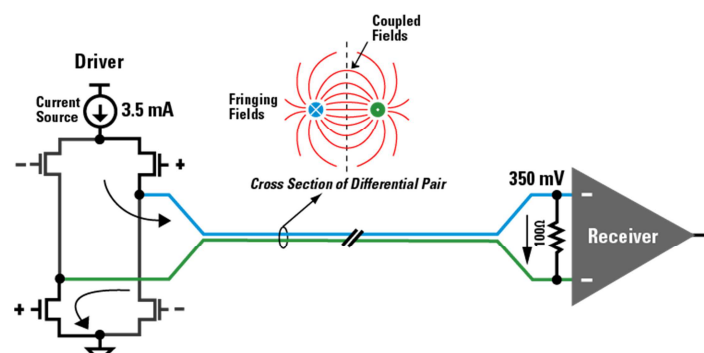


Figura A.1 Estructura de un emisor y receptor LVDS. En ésta se aprecian los diversos elementos que componen una red básica [29].

La configuración tanto en formas de redes como en esquemas de reloj varía mucho dependiendo de la aplicación y de los dispositivos. En este caso se trata únicamente de la topología de punto a punto, como se ilustra en la figura A.1. En el caso de los esquemas reloj también varía significativamente dependiendo de la aplicación. En el caso particular del AFE5808A el esquema es el mostrado en la figura A.2, donde se aprecia que en este caso se emplean dos relojes para temporización: el primero define cada bit con DDR, el segundo permite distinguir dos muestras consecutivas entre sí.

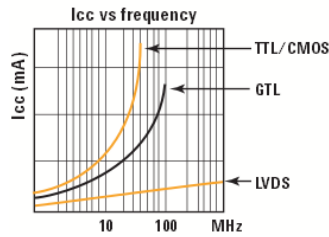


Figura A.2 Consumo de corriente contra frecuencia de operación [27].

Redes empleadas en la implementación

Este tipo de protocolo provee diversas configuraciones como las mostradas en la figura A.3; sin embargo, el circuito empleado en el proyecto (AFE5808) únicamente maneja un esquema de comunicación punto a punto.

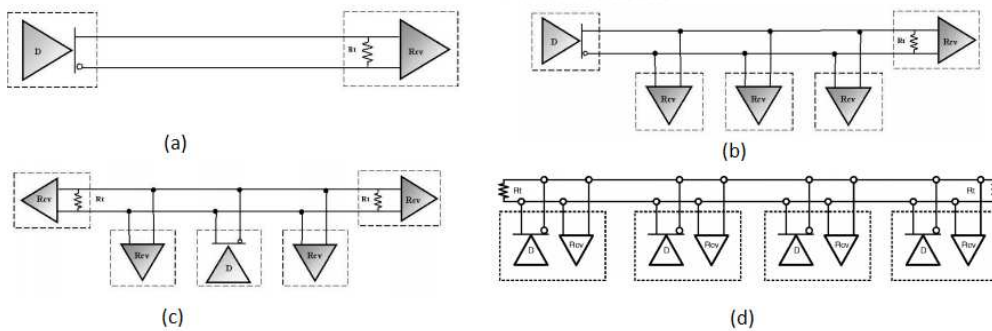


Figura A.3 Configuración de las topologías de red implementadas con el estándar LVDS. (a) Punto a punto, (b) multicáida (*Multi-drop*), (c) multicáida con repetidora en el centro (*Multi-drop center driven*), (d) multipunto (muy similar a (b), pero la repetidora puede estar en cualquier punto del bus).

Configuraciones de reloj

Los esquemas de reloj varían en forma importante; sin embargo, un esquema comúnmente encontrado en convertidores analógicos digitales es el mostrado en la figura A.4, en el cual se proveen dos relojes: uno que permite discernir una palabra de otra y uno que sirve para el muestreo y captura de la señal de datos.

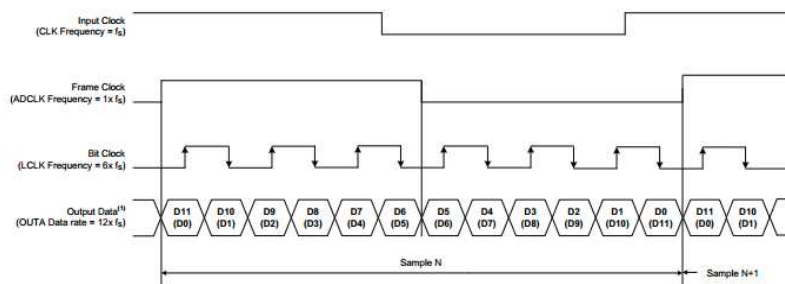


Figura A.4 Esquema de reloj empleado en los módulos de comunicación LVDS de los CAD de Texas Instruments [30].

Apéndice D: API²² preliminar del sistema (recepción)

Resumen

El sistema se conforma de un conjunto módulos de hardware, software y firmware que trabaja en conjunto con una interfaz de control descrita en MATLAB. Sus funciones incluyen la generación de secuencias de excitación/recepción con parámetros definidos por el usuario. También es útil para configuraciones que involucren pulso-eco y ultrasonido por transmisión/atenuación.

Historia

En el presente documento se muestra la información correspondiente una versión preliminar del sistema, creada entre febrero y noviembre del año 2013.

Estructura de parámetros e Inicialización del sistema

Los parámetros de recepción afectan las características de cada muestra tomada por el sistema, dichos parámetros se encuentran definidos dentro de una estructura llamada *rx_params*. Como regla general, antes de comenzar a trabajar con el sistema se debe escribir la siguiente línea de configuración *init_val*:

```
%Inicialización de la estructura de parámetros de adquisición
>> init_val
>> rx_params

rx_params =

    portAFE: 'COM3'
    portFPGA: 'COM5'
    chan: [1 2 3 4 5 6 7 8]
    cf_LPF: 15
    HPF: 'Y'
    cf_HPF: 100
    test_pattern: 7
    PGA_gain: 24
PGA_clamp_lv: -2
    PGA_clamp_en: 'Y'
LNA_gain: 18
    LNA_clmp: 0
    term_imp: 50
    aqui_time_us: 1.2800e+01
    f_sample_MSPs: 40
DITHER: 'Y'
    no_pulsos: 1
    ciclos_escritura_p_ch: 1
    ciclos_escritura_totales: 8
    velocidad_popagacion: 1540
    profundidad_mm: 1.9712e+01
```

²² API (*Application Programming Interface*): se refiere a un texto que describe las diversas funciones de un sistema para su posterior uso por un desarrollador de sistemas.

Parámetros de configuración

El comportamiento del sistema se encuentra delimitado por dichos parámetros y están contenidos en la estructura *rx_params*, como se mencionó en el punto interior. Un resumen de éstos se muestra en la tabla D.1. El sistema completo debe incluir una estructura adicional los parámetros del pulsador.

Se recomienda ampliamente que la configuración de los parámetros se lleve a cabo mediante las funciones mostradas en la tabla D.2, las cuales se encargan de validar y poner en el formato apropiado los parámetros introducidos por el usuario. Facilitando la tarea de configurar el sistema y minimizando la probabilidad de errores en la adquisición. A continuación se muestra un ejemplo de configuración:

```
>> set_f_sample(40)
>> chans_enabled([8,7,6,5;7,6,5,4;6,5,4,3;5,4,3,2;4,3,2,1])
>> HPF_enable('y')
>> cut_HPF_KHz(150)
>> cut_LPF_MHz(10)
>> DITHER('n')
>> set_gain_dB(54)
>> set_aqui_time_us(200)
>> setPortFPGA('COM5')
>> setPortAFE('COM3')
>> factor_submuestreo: 3
```

Un punto especial a tomar en cuenta es que la configuración de algunos de los parámetros podría afectar indirectamente a otros, ya que en muchas ocasiones éstos tienen una relación con ellos. Como un ejemplo se considera la velocidad de propagación y la profundidad de penetración. Se recomienda tener en cuenta estos detalles al momento de la configuración para lograr una adquisición exitosa.

Introducción de secuencias de recepción

Cada secuencia arrojará una o más matrices de elementos (dependiendo del número de secuencias definidas por el usuario), los cuales son entregados al usuario para su posterior procesamiento.

Las secuencias de canales se introducen mediante una matriz con el comando *set_chans()*, el cual se encarga de validar los parámetros de entrada, asegurando un funcionamiento apropiado. Cada fila de la matriz introducida corresponderá a una secuencia cuyas filas representan los canales y sus columnas las muestras. Cabe mencionar que la estructura antes propuesta limita el sistema a tomar muestras de la misma duración y número de canales para todos los disparos.

```
%Introducción de la secuencia de adquisiciones
set_chans([1,2,3,4,5;2,3,4,5,6;3,4,5,6,7;4,5,6,7,8]);
```


Captura de muestras

Se realiza mediante el comando *adquirir()*, el cual regresa el arreglo de matrices mencionado en el subtema D.5. Éste configura el sistema y después ejecuta los algoritmos de adquisición, reconstrucción de muestras y agrupamiento.

```
>>Mat(:, :, 1) =  
Columns 1 through 21  
0  
0 257 257 257 257 257 257 257 257 257 257 257 257 257 257 257 257 257 257 257  
257  
514 514 514 514 514 514 514 514 514 514 514 514 514 514 514 514 514 514 514 514  
514  
771 771 771 771 771 771 771 771 771 771 771 771 771 771 771 771 771 771 771 771  
771  
...
```

Parámetro	Tipo	Rango	Valor default	Observaciones
PortAFE	string	'COMX'	'COM3'	Configura el puerto COM correspondiente al MAACAD.
PortFPGA	string	'COMX'	'COM5'	Configura el puerto COM correspondiente al MAT.
chan	int	1,2,3,4,5,6,7,8	[1,2,3,4,5,6,7,8]	Matriz de 1X1 hasta 8X1024.
cf_LPF	int	10 a 30	15	Siempre activo.
HPF	char	'Y', 'N'	'Y'	Controla tanto el compensador de offset del PGA como el del LNA.
cf_HPF	int	50, 80, 100, 150	100	Por debajo de los 80[KHz] se desactiva el filtro pasa altas del PGA.
test_pattern	int	0,1,2,3,4,5,6,7	7	En 7 los patrones de prueba se encuentran desactivados.
PGA_gain	int	24 , 30	24	
PGA_clamp_lv	int	-2, 0, 2	-2	Las unidades son [dBFS] (decibeles con respecto a la escala completa).
PGA_clamp_en	char	'Y', 'N'	'Y'	Tiene respuesta de filtro.
LNA_gain	int	12, 18, 24	18	
LNA_clmp	int	0.6, 1.15, 1.5	0	Máximo de voltaje que soporta el sistema como entrada.
term_imp	int	50	50	Únicamente sirve como indicador, se incluye para posible desarrollo futuro.
aquí_time_us	int	1 a 3264	12.8	Valor mínimo de adquisición para la velocidad de propagación del sonido en agua.
f_sample_MSPs	double	0.156	40	Sólo modificar si se introduce una señal externa de reloj.
DITHER	char	'Y', 'N'	'Y'	Mejora resolución del ADC.
no_pulsos	int	1 a 1024	1	Se definen mediante la secuencia en la matriz de canales habilitados.
ciclos_escritura_p_ch	int	0 a 255	1	1 ciclo de escritura es el mínimo.
ciclos_escritura_totales	int	0 a 255	8	1 ciclo de escritura por 8 canales activos.
velocidad_propagacion	double	$0 < v \leq 12000$	1540	Como default el medio considerado es agua.
profundidad_mm	double	$0 < d < 3916$	12.8	Los 28 um son equivalentes a 512 muestras (que es el tamaño del buffer), todos los incrementos son equivalentes a este tamaño de buffer.
factor_submuestreo	int	0 a 255	0	

Tabla D.1 Elementos de la estructura rx_params.

Propiedad	Descripción	Afecta en rx_params	Comando de configuración
Canales activos	Se refiere al número de elementos habilitados para recepción. Cada uno corresponde a una secuencia.	chan	set_chans(int)
Frecuencia de corte del filtro pasa bajas	El filtro pasa bajas siempre se encuentra activo. Es configurable a 10/15/20/30MHz.	cf_LPF	cut_LPF_MHz(double)
Habilitación de supresión de ruido de baja frecuencia (filtro pasa altas)	Se tienen dos compensadores de offset: uno en el LNA y otro en el PGA, que actúan como filtro pasa altas, La habilitación de ambos es opcional. Sin embargo, para frecuencias menores que 80 MHz se mantendrá apagado en compensador del PGA cuya frecuencia de corte es fija.	HPF	HPF_enable(char)
Frecuencia de corte del filtro pasa altas	Define la frecuencia de corte del filtro pasa altas, deshabilitado si HPF='N'.	cf_HPF	cut_HPF_KHz(double)
Ganancia	Define la ganancia total del sistema, conformada por la etapa del LNA y del PGA.	PGA_gain LNA_gain	set_gain_dB(double)
Frecuencia de muestreo	Default de 40MSPs, definido por el oscilador instalado. No debe ser alterado por el usuario a menos que introduzca un reloj externo de frecuencia conocida.	f_sample	set_f_sample(double)
Tiempo de adquisición por muestra	Define el tiempo de adquisición de un ciclo excitación/recepción. Está sujeto a la memoria disponible y al número de canales activos.	aqui_time samples	set_aqui_time(double)
Profundidad de adquisición	Consecuencia de la velocidad de propagación y del tiempo de adquisición.	depth samples	set_depth(double)

DITHER		DITHER	DITHER(char)
	Activa la función de DITHER, la cual agrega ruido blanco a la señal para mejorar la resolución del sistema a costa de la alteración de la figura de ruido del sistema.		
Ajustar frecuencia de muestreo	Se trata de un sistema que permite calcular el factor de submuestreo y el cambio en la resolución temporal y espacial del sistema, ya que el buffer de datos en el MAT es fijo.	ciclos_escritura_p_ch ciclos_escritura_totales aqui_time_us profundidad_mm	set_f_sample
Ajustar velocidad de propagación	Ajusta la velocidad de propagación del sistema y, en forma indirecta, también altera la profundidad de penetración.	velocidad_popagacion profundidad_mm	set_vel_prop(double)
Configurar <i>Analog Front End</i>	Configura el módulo de acondicionamiento analógico y CAD.	---	snd_conf_AFE(estructura)
Configurar <i>FPGA</i>	Configura el módulo de almacenamiento y transmisión.	---	config_FPGA_AFE(estructura)
Recibir datos	Recibe, agrupa y entrega al usuario las matrices que contienen los datos de la adquisición.	---	leer_y_agrupar()
Realizar captura	Realiza la adquisición de datos.	---	adquirir(estructura)

Tabla D.2 Comandos de configuración correspondientes a la etapa de recepción.

Bibliografía

- [1] R. S. Kayvan Najarian, Biomedical Signal and Image Processing, USA: CRC Press, 2006.
- [2] S. A. Kane, Introduction to Physics in Modern Medicine, USA: CRC Press, 2009.
- [3] NI Analog Resource Center, "Ultrasonic Non Destructive Testing: Advanced Concepts and Applications", National Instruments, Austin, TX, 2012.
- [4] D. G. M. John G. Proakis, Digital Signal Processing: Principles, algorithms and applications, New Jersey: Prentice Hall, 1996.
- [5] W. Kester, The data conversion handbook, Norwood, MA: Analog Devices, 2004.
- [6] M. Mano, Logic and computer fundamentals, USA: Pearson, 2004.
- [7] A. G. S. Ron Sass, Embedded Systems Design With Plattform FPGAs: Principles and Practices, USA: Elsevier, 2010.
- [8] P. P. Chu, FPGA Prototyping by VHDL Examples, USA: Willey Interscience, 2008.
- [9] R. R. Paul Chang, "SDRAM and PowerQUICC II: An Introduction", Freescale Semiconductor, Austin, TX, 2005.
- [10] A. Morris, Principios de Medición en Instrumentación, México: Pearson educación, 2002.
- [11] E. O. Doebelin, Measurement Systems: Applicatios and Design, Singapur: Mc Graw Hill, 1990.
- [12] X. Xu, "Challenges and Considerations of Analog Front-ends Design for Portable Ultrasound Systems", *IEEE International Ultrasonics Symposium Proceedings*, pp. 310 - 313, 2010.
- [13] H. H. Hansen, A 33uW Sub - 3 dB Noise Figure Low Noise Amplifier for Medical Ultrasound Applications, NTNU.
- [14] T. S. Rathore, "Programable Gain Amplifiers," en *Digital Measurement Techiques*, Pangbourne, Alpha Schience.
- [15] H. Zumbalhen, Linear Circuit Design Handbook, Norwood, MA: Newness, 2007.
- [16] Texas Instuments, "AFE5808/08A (Rev E) Evaluation Module", Austin, 2011.
- [17] Texas Instruments, "Hoja de datos AFE5808A", Austin, 2012.

- [18] Xilinx, *Spartan-6 FPGA Configurable Logic Block*, San Jose, CA, 2010.
- [19] R. Ginosar, "Metastability and Synchronizers: A tutorial", *IEEE Design & Test of Computers*, pp. 23-35, 2011.
- [20] Xilinx, *Spartan-6 FPGA Memory Controller*, San Jose, CA, 2010.
- [21] Xilinx, *SP605 Hardware User Guide*, San Jose, CA, 2012.
- [22] Xilinx, *Spartan 6 Libraries Guide For HDL Designs*, San Jose, CA., 2009.
- [23] A.L.S.E., *A.L.S.E. Application Note RS232 Basics*, Paris, 209.
- [24] P. P. Chu, *FPGA Prototyping by VHDL Examples*, Hohoken, NJ: Wiley Interscience, 2008.
- [25] Intersil, "SPI Protocol and Bus Configuration of Multiple DCPs," Milpitas, CA, 2007.
- [26] Y. W. Tianxiquang Lui, "IP Design of Universal Multiple Devices SPI Interface", *IEEE International Conference on Anti-Counterfeiting*, pp. 169-172, 2011.
- [27] Agilent Technologies, "Low-Voltage Differential Signaling (LVDS)", Santa Clara, CA, 2002.
- [28] I. C. Society, *IEEE Standard for Low Voltage Differential Signaling (LVDS) for Scalable Coherent Interface (SCI)*, New York: IEEE, 1996.
- [29] Texas Instruments, "LVDS Owner's Manual", Dallas, TX, 2008.
- [30] Texas Instruments, "Understanding serial LVDS Capture in High-Speed ADCs", Dallas, TX, 2013.
- [31] Ren Mancini, "Op Amp Noise Theory and Applications" Texas Instruments, Dallas, TX, 2002.