

PROYECTO FINAL DE
INGENIERÍA ELECTRÓNICA

Sistema de adquisición de señales
eléctricas basado en SoC FPGA



UNIVERSIDAD NACIONAL
de MAR DEL PLATA
.....

Autor: Jordi Tadeo Tribó

Director: Dr. Ing. Carlos Arturo Gayoso

Co-Director: Ing. Lucas Rabioglio



RINFI se desarrolla en forma conjunta entre el INTEMA y la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

PROYECTO FINAL DE
INGENIERÍA ELECTRÓNICA

Sistema de adquisición de señales
eléctricas basado en SoC FPGA



UNIVERSIDAD NACIONAL
de MAR DEL PLATA
.....

Autor: Jordi Tadeo Tribó

Director: Dr. Ing. Carlos Arturo Gayoso

Co-Director: Ing. Lucas Rabioglio

Resumen

En el presente proyecto se lleva a cabo el diseño e implementación de un sistema de adquisición de señales basado en arquitectura SoC FPGA. El mismo cuenta con dos canales de entrada analógica de tensión, que operan a una frecuencia de muestreo de 1 MSa/s cada uno. Los datos son enviados en tiempo real a través de una conexión internet y visualizados de forma remota en un software que actúa de cliente.

El elemento central del sistema consiste en una plataforma de desarrollo Terasic DE0-Nano-SoC, la cual integra un SoC FPGA Cyclone V de Intel, 1 GiB de memoria RAM DDR3 y conectividad Gigabit Ethernet. Por un lado, en el sector de lógica programable, se implementa la arquitectura de adquisición y procesamiento de los datos. Por otro lado, el software que se ejecuta en el procesador envía los datos a través de internet y recibe los comandos de configuración.

Así mismo, se diseña un front-end, que consiste en una placa encargada del acondicionamiento analógico de ambos canales y su conversión al dominio digital. Esta etapa brinda flexibilidad al sistema, permitiendo capturar señales en diversos rangos de tensiones y de diversas amplitudes. El control del front-end se lleva a cabo de forma remota, a través del software de visualización. Con el objetivo de validar el diseño teórico de esta etapa, se fabrica un prototipo del front-end y se llevan a cabo diversas mediciones durante su funcionamiento.

El último componente que conforma el sistema es el cliente de visualización, el cual consiste en un software multiplataforma, capaz de ejecutarse en sistemas operativos Windows, Mac OS y Linux. La interfaz gráfica del cliente permite ajustar los diversos parámetros de adquisición y controlar tanto el servidor de adquisición como el front-end. Por otro lado, se desarrollan e implementan diversas funcionalidades que favorecen la visualización de los datos adquiridos y su posterior análisis.

Índice

1 Introducción.....	1
1.1 Objetivos.....	2
1.2 Especificación de requerimientos.....	2
1.2.1 Ámbito del sistema.....	3
1.2.2 Descripción general.....	3
1.2.2.1 Perspectiva del producto.....	3
1.2.2.2 Funciones del producto.....	3
1.2.2.3 Características del usuario.....	4
1.2.2.4 Restricciones.....	4
1.2.2.5 Suposiciones y dependencias.....	4
1.2.3 Requisitos específicos.....	4
1.2.3.1 Requerimientos funcionales.....	5
1.2.3.2 Requerimientos no funcionales.....	8
1.2.3.3 Interfaces externas.....	9
1.3 Descripción general del sistema.....	9
1.3.1 Front-end analógico.....	10
1.3.2 Placa de desarrollo.....	11
1.3.3 Cliente.....	11
1.4 Análisis de mercado.....	11
1.5 Organización del documento.....	16
2 Arquitectura SoC FPGA.....	17
2.1 Lógica programable.....	18
2.1.1 Módulos de lógica adaptativa.....	19
2.1.2 Bloques M10K.....	20
2.1.3 Bloques DSP.....	20
2.1.4 Bloques I/O.....	21
2.1.5 Lazos de enganche de fase.....	22
2.1.6 Controlador de memoria.....	23
2.2 Sistema de procesamiento.....	23
2.2.1 Procesador y caché.....	24
2.2.2 Controladores de memoria Flash.....	25
2.2.3 Controlador de memoria RAM.....	25
2.2.4 Memoria integrada.....	25
2.2.5 Periféricos de comunicación.....	26
2.2.6 Periféricos generales.....	26
2.2.7 Interfaz con la FPGA.....	27
2.3 Interconexión del sistema.....	27
2.3.1 Diagrama en bloques del sistema.....	27
2.3.1.1 Procesador.....	28
2.3.1.2 FPGA.....	29
2.3.1.3 Periféricos maestros.....	30
2.3.1.4 Periféricos esclavos.....	30
2.3.2 Mapa de memoria.....	30
2.4 Placa de desarrollo.....	31
2.4.1 Elementos generales.....	32
2.4.2 Elementos conectados a la FPGA.....	33
2.4.3 Elementos conectados al HPS.....	34
3 Front-end analógico.....	35
3.1 Diagrama en bloques.....	35

3.2 Ganancias requeridas.....	37
3.3 Diagrama esquemático simplificado.....	38
3.4 Diseño de etapas analógicas.....	40
3.4.1 Elección de amplificadores.....	40
3.4.2 Acople de alterna.....	40
3.4.3 Atenuador seleccionable.....	42
3.4.4 Protección.....	45
3.4.5 Sumador de offset.....	47
3.4.6 Atenuador programable.....	52
3.4.7 Ganancia programable.....	54
3.4.8 Buffer ADC.....	56
3.4.9 ADC.....	57
3.4.10 Generación de offset.....	59
3.5 Transferencia total.....	61
3.6 Simulaciones.....	65
3.7 Interfaz con la FPGA.....	68
3.8 Alimentación en placa principal.....	71
3.9 Alimentación en placa secundaria.....	73
3.10 Diseño de PCBs.....	78
4 Arquitectura de adquisición.....	82
4.1 Metodología de diseño.....	82
4.1.1 Integración del sistema.....	84
4.1.2 Síntesis de alto nivel.....	87
4.1.3 Verificación de lógica programable.....	89
4.1.4 Análisis temporal.....	90
4.2 Descripción de la arquitectura.....	91
4.2.1 Diagrama en bloques.....	92
4.2.2 Adquisición.....	93
4.2.3 Procesamiento.....	94
4.3 Implementación de cada etapa.....	96
4.3.1 Generación de relojes.....	96
4.3.2 Interfaz con el front-end.....	97
4.3.3 Interfaz ADC.....	99
4.3.4 Sistema de disparo.....	103
4.3.5 DMA de adquisición.....	107
4.3.6 DMA SRC y DMA SINK.....	111
4.3.7 Procesamiento.....	115
4.3.8 HPS.....	121
4.4 Síntesis del diseño.....	123
4.4.1 Recursos utilizados en la FPGA.....	123
4.4.2 Análisis temporal.....	125
5 Software.....	127
5.1 Linux embebido.....	127
5.2 Herramientas de desarrollo y depuración.....	130
5.3 Algoritmo de calibración.....	132
5.3.1 Offset.....	133
5.3.2 Ganancia.....	134
5.3.3 Aplicación de los factores.....	135
5.4 Servidor.....	136
5.4.1 Comunicación con el hardware.....	136
5.4.2 Comunicación con el cliente.....	138
5.4.3 Diagrama de flujo.....	139
5.5 Cliente.....	141
5.5.1 Área de visualización.....	142
5.5.2 Barra de herramientas.....	147

5.5.3 Panel de ajustes.....	148
5.5.3.1 Ajustes de disparo.....	148
5.5.3.2 Ajustes verticales.....	150
5.5.3.3 Ajustes horizontales.....	151
5.5.3.4 Ajustes de visualización.....	152
6 Validación.....	155
6.1 Armado de prototipo.....	155
6.1.1 Fabricación de PCBs.....	155
6.1.2 Pedido de componentes.....	157
6.1.3 Armado de placas.....	158
6.1.4 Errores de diseño físico de PCB.....	159
6.2 Mediciones.....	160
6.2.1 Instrumental utilizado.....	160
6.2.2 Medición de fuente lineal.....	162
6.2.3 Calibración de offset.....	163
6.2.4 Calibración de ganancia.....	165
6.2.5 Etapa de generación de offset.....	166
6.2.6 Entrada.....	168
6.2.7 Acoplamiento de alterna.....	169
6.2.8 Atenuador compensado.....	170
6.2.9 Amplificador sumador de offset.....	172
6.2.10 Potenciómetro digital y buffer de PGA.....	173
6.2.11 PGA y buffer de ADC.....	177
6.2.12 Protección de entrada.....	178
6.2.13 Ancho de banda total.....	179
6.2.14 Análisis de ruido.....	180
6.3 Pruebas integrales.....	181
6.3.1 Ajustes verticales y resolución.....	182
6.3.2 Acople de alterna y offset de voltaje.....	185
6.3.3 Algoritmos de procesamiento y persistencia.....	187
6.3.4 Profundidad de memoria y modo zoom.....	192
6.3.5 Modos de visualización de dos canales.....	194
6.3.6 Modo zoom.....	197
7 Conclusiones.....	199
7.1 Objetivos alcanzados.....	199
7.2 Líneas de desarrollo futuras.....	199
Bibliografía.....	203

Índice de figuras

Figura 1.1: Diagrama en bloques general del sistema de adquisición.....	10
Figura 1.2: Chasis cRIO-9063 de National Instruments.....	12
Figura 1.3: Entorno de desarrollo LabVIEW de National Instruments.....	13
Figura 1.4: Digilent OpenScope MZ.....	14
Figura 1.5: Red Pitaya STEMLab 125-14.....	14
Figura 2.1: Placa de desarrollo Terasic DE0-Nano-SoC.....	17
Figura 2.2: Arquitectura SoC FPGA Cyclone V.....	18
Figura 2.3: Arquitectura interna de ALM en Cyclone V.....	19
Figura 2.4: Ejemplo de modo de operación de bloque DSP en Cyclone V.....	21
Figura 2.5: Bancos de entrada y salida en Cyclone V SE A4 (vista superior).....	22
Figura 2.6: Vista de procesador y periféricos en Cyclone V.....	23
Figura 2.7: Diagrama en bloques de conexiones en Cyclone V.....	28
Figura 2.8: Vista de memoria (a) Matriz L3 (b) Procesador (c) Puente FPGA-to-SDRAM.....	31
Figura 2.9: Componentes de la placa de desarrollo Terasic DE0-Nano-SoC.....	32
Figura 3.1: Diagrama en bloques del front-end analógico.....	36
Figura 3.2: Acople de alterna y atenuador seleccionable.....	38
Figura 3.3: Protección, sumador de offset y potenciómetro digital.....	39
Figura 3.4: Ganancia programable, amplificador operacional de salida y ADC.....	39
Figura 3.5: Generación del voltaje de offset.....	39
Figura 3.6: Circuito equivalente para el análisis del acople de alterna.....	41
Figura 3.7: Redes de atenuación de entrada propuestas inicialmente.....	42
Figura 3.8: Efecto de la capacidad parásita en el atenuador de entrada.....	43
Figura 3.9: (a) Divisor resistivo (b) Divisor capacitivo (c) Atenuador compensado.....	43
Figura 3.10: (a) Atenuador de 0,9 (b) Atenuador de 0,1.....	44
Figura 3.11: Circuito equivalente para el análisis de la protección.....	46
Figura 3.12: Circuito sumador de offset (buffer de entrada).....	47
Figura 3.13: Respuesta en frecuencia OPA2197 a lazo abierto.....	48
Figura 3.14: Transferencia a lazo abierto OPA2197 con $C_L = 15$ pF.....	48
Figura 3.15: Respuesta al escalón inicial.....	49
Figura 3.16: Efecto de la capacidad de entrada en la primera red propuesta.....	50
Figura 3.17: Red de realimentación mejorada.....	50
Figura 3.18: Respuesta al escalón con la segunda red de realimentación propuesta.....	51
Figura 3.19: Respuesta en frecuencia final con la segunda red de realimentación propuesta.....	52
Figura 3.20: Potenciómetro digital y buffer de PGA.....	53
Figura 3.21: Circuito equivalente de potenciómetro digital (50%).....	54
Figura 3.22: (a) Respuesta en frecuencia del PGA (b) Diagrama en bloques simplificado.....	55
Figura 3.23: Buffer ADC (traslado de nivel).....	56
Figura 3.24: Diagrama en bloques ADCS7476.....	58
Figura 3.25: Diagrama temporal de la interfaz del convertor ADCS7476.....	58
Figura 3.26: Circuito de generación de offset.....	59
Figura 3.27: Red de realimentación del amplificador de offset.....	60
Figura 3.28: Simulación de respuesta en frecuencia de las escalas.....	65
Figura 3.29: Señal de entrada para simulación temporal de offset.....	66
Figura 3.30: Señal de salida para simulación temporal de offset.....	67
Figura 3.31: Señal de entrada para simulación temporal de acople de alterna.....	67
Figura 3.32: Señal de salida para simulación temporal de acople de alterna.....	67
Figura 3.33: Diagrama de pines de conector interfaz en PCB de acondicionamiento.....	69
Figura 3.34: Traslado de niveles de tensión lógicos.....	69
Figura 3.35: Diagrama esquemático y layout de placa adaptadora.....	70

Figura 3.36: Buffer de línea de datos de ADC.....	71
Figura 3.37: Diodos de protección entradas digitales de ADC.....	71
Figura 3.38: Regulación lineal y filtrado (± 5 V y 3,3 V).....	72
Figura 3.39: Regulación lineal y filtrado (-3,3 V).....	72
Figura 3.40: Referencias de tensión (5 V y 3,3 V).....	73
Figura 3.41: PSRR y CMRR vs frecuencia (OPA2197).....	74
Figura 3.42: Transformador 220 V a 12 V externo.....	75
Figura 3.43: Diagrama esquemático de fuente lineal bipolar.....	75
Figura 3.44: Circuito equivalente de etapa de rectificado y filtrado.....	75
Figura 3.45: Recta de descarga (rojo) y entrada rectificada (azul).....	76
Figura 3.46: Simulación de la fuente lineal bajo corriente de carga constante.....	78
Figura 3.47: PCB principal vista superior.....	79
Figura 3.48: PCB principal tercera capa (alimentación).....	80
Figura 3.49: PCB principal vista inferior (pistas digitales).....	81
Figura 3.50: Fuente de alimentación vista superior.....	81
Figura 4.1: Módulo ADC PMOD AD1 (Digilent).....	83
Figura 4.2: Software Quartus Prime Lite 18.1 de Intel FPGA.....	84
Figura 4.3: Platform Designer (Quartus Prime Lite).....	85
Figura 4.4: Protocolos Avalon-MM y Avalon-ST.....	86
Figura 4.5: Diseño mediante diagrama en bloques (Simulink).....	87
Figura 4.6: Diseño de máquinas de estados en Stateflow (Simulink).....	88
Figura 4.7: Simulación funcional en Modelsim de código VHDL (Quartus).....	89
Figura 4.8: Analizador lógico SignalTap mostrando buses Avalon-ST (Quartus).....	90
Figura 4.9: Herramienta de análisis temporal Timing Analyzer (Quartus).....	91
Figura 4.10: Diagrama en bloques de arquitectura de adquisición.....	92
Figura 4.11: Eventos que forman parte de un ciclo de adquisición.....	93
Figura 4.12: Buffers de memoria para captura de 8 muestras con offset de 6 muestras.....	96
Figura 4.13: Conversión de valor del ADC en voltaje en punto fijo 32 bits.....	100
Figura 4.14: Relación entre el evento de disparo y la base de tiempo.....	104
Figura 4.15: Máquina de estado de disparo para flanco ascendente (Simulink).....	105
Figura 4.16: Simulación de sistema de disparo (Simulink).....	106
Figura 4.17: Diagrama en bloques DMA adquisición.....	108
Figura 4.18: Diagrama en bloques DMA SRC.....	111
Figura 4.19: Diagrama en bloques DMA SINK.....	112
Figura 4.20: Buffers de adquisición implementados en memoria RAM.....	112
Figura 4.21: Decimación normal con factor $D=50$ (Simulink).....	117
Figura 4.22: Respuesta de filtro de promediado móvil con $D=5$	117
Figura 4.23: Efecto de aliasing en decimación con $D=5$	118
Figura 4.24: Decimación de alta resolución con factor $D=50$ (Simulink).....	119
Figura 4.25: Decimación de detección de pico con factor $D=50$ (Simulink).....	120
Figura 4.26: Uso porcentual de lógica digital por bloque.....	125
Figura 5.1: Secuencia de arranque para el caso de sistema operativo (Linux).....	128
Figura 5.2: Layout de los componentes de software en la tarjeta SD.....	129
Figura 5.3: Entorno de desarrollo integrado Eclipse IDE.....	130
Figura 5.4: Depuración remota a través de conexión Ethernet (Eclipse).....	131
Figura 5.5: Editor de texto Visual Studio Code de Microsoft.....	132
Figura 5.6: Conversión de datos realiza en el módulo de interfaz de ADC.....	135
Figura 5.7: Ejecución del script del servidor a través de una terminal SSH.....	139
Figura 5.8: Diagrama de flujo del proceso de adquisición.....	140
Figura 5.9: Interfaz gráfica del cliente de visualización.....	142
Figura 5.10: Elementos gráficos que componen el área de visualización.....	143
Figura 5.11: Modo de visualización multi-plot.....	145
Figura 5.12: Modo de visualización single-plot.....	146
Figura 5.13: Funcionalidad de zoom (detalle temporal).....	147
Figura 5.14: Íconos de la barra de herramientas.....	148

Figura 5.15: Panel de ajustes de disparo.....	149
Figura 5.16: Panel de ajustes verticales.....	150
Figura 5.17: Panel de ajustes horizontales y de adquisición.....	151
Figura 5.18: Panel de ajustes de visualización.....	153
Figura 6.1: Placa principal.....	156
Figura 6.2: Placa adaptadora y fuente de alimentación.....	156
Figura 6.3: Fuente de alimentación finalizada.....	158
Figura 6.4: PCB principal finalizado.....	159
Figura 6.5: Conector BNC, vista superior e inferior.....	160
Figura 6.6: Osciloscopio Owon XDS3202E utilizado en las mediciones.....	161
Figura 6.7: Fuente de laboratorio UNI-T UTP3305 utilizada para alimentar el front-end.....	161
Figura 6.8: Forma de onda a la salida del transformador sin carga.....	162
Figura 6.9: Forma de onda a la salida de la fuente de alimentación I_L de 0,5 A.....	163
Figura 6.10: Salida del DAC ante un escalón de 4,5 V_{PP} centrado en 2,5 V.....	166
Figura 6.11: Salida del amplificador de offset ante un escalón de 2,25 V_{PP} centrado en 2,5 V....	167
Figura 6.12: Escalón de 3,3 V_{PP} de amplitud utilizado como señal de prueba.....	168
Figura 6.13: Respuesta del acople de alterna ante un escalón positivo de 3,3 V_{PP}	169
Figura 6.14: Salida del atenuador de 0,1 ante pulso digital de 3,3 V_{PP}	170
Figura 6.15: Salida del primer amplificador operacional ante pulso digital de 3,3 V_{PP}	171
Figura 6.16: Salida del primer amplificador para los ajustes del atenuador de 0,1 veces.....	171
Figura 6.17: Detalle temporal de la salida del primer amplificador ante un pulso de 3,3 V_{PP}	172
Figura 6.18: Salida del potenciómetro digital ante un pulso de 3,3 V_{PP} (K_P 0,67).....	174
Figura 6.19: Salida del buffer del PGA ante un pulso de 3,3 V_{PP}	175
Figura 6.20: (a) Circuito equivalente de potenciómetro digital (b) Equivalente de Thévenin.....	176
Figura 6.21: Forma de onda final ante un pulso de 3,3 V_{PP} en la escapa 0,2 V/div.....	177
Figura 6.22: Efecto de muestreo y retención de ADC en arquitectura SAR.....	178
Figura 6.23: Protección ante un pulso de 25 V y una selección del atenuador de 0,9.....	179
Figura 6.24: Señal cuadrada de 1 kHz observada en escala de 5 V/div.....	183
Figura 6.25: Señal cuadrada de 1 kHz observada en escala de 50 mV/div.....	184
Figura 6.26: Señal cuadrada de 1 kHz observada en escala de 10 mV/div (offset restado).....	185
Figura 6.27: Señal cuadrada de 10 Hz y 5 V_{PP} observada bajo acople de alterna.....	186
Figura 6.28: Señal cuadrada de 10 Hz y 5 V_{PP} (offset restado).....	187
Figura 6.29: Ruido blanco Gaussiano de 0,5 V_{RMS} sin decimación.....	188
Figura 6.30: Ruido blanco Gaussiano de 0,5 V_{RMS} decimación normal ($D=100$).....	189
Figura 6.31: Ruido blanco Gaussiano de 0,5 V_{RMS} algoritmo de detección de pico ($D=100$).....	190
Figura 6.32: Ruido blanco Gaussiano de 0,5 V_{RMS} algoritmo de alta resolución ($D=100$).....	191
Figura 6.33: Ruido blanco Gaussiano de 0,5 V_{RMS} promediado de 128 <i>capturas</i> ($D=100$).....	192
Figura 6.34: Modo zoom aplicado a senoidal de 5 kHz en base de tiempo de 1 ms/div.....	193
Figura 6.35: Modo zoom aplicado a senoidal de 5 kHz en base de tiempo de 0,1 s/div.....	194
Figura 6.36: Visualización de ambos canales en un mismo gráfico (single-plot).....	195
Figura 6.37: Visualización de ambos canales en gráficos separados (multi-plot).....	196
Figura 6.38: Visualización de ambos canales en modo single-plot con zoom activado.....	197

Índice de tablas

Tabla 1.1: Requerimientos funcionales del sistema de adquisición.....	7
Tabla 1.2: Requerimientos no funcionales del sistema de adquisición.....	8
Tabla 1.3: Comparación del dispositivo con las alternativas comerciales directas existentes.....	15
Tabla 3.1: Ganancias requeridas para cada escala y resolución en ADC.....	37
Tabla 3.2: Parámetros del PGA para cada ajuste de ganancia.....	55
Tabla 3.3: Configuración del front-end para cada escala.....	62
Tabla 3.4: Rangos de voltaje en todas las etapas para cada escala.....	63
Tabla 3.5: Rango y resolución del DAC para cada escala.....	64
Tabla 3.6: Anchos de banda y ganancias obtenidos por simulación.....	66
Tabla 4.1: Configuración PLL fraccional.....	97
Tabla 4.2: Configuración bloque SPI maestro (potenciómetro).....	97
Tabla 4.3: Configuración bloque SPI maestro (DAC).....	98
Tabla 4.4: Configuración puerto GPIO front-end.....	98
Tabla 4.5: Distribución de bits puerto GPIO front-end.....	98
Tabla 4.6: Conexiones para bloque SPI maestro (potenciómetro).....	99
Tabla 4.7: Conexiones bloque SPI maestro (DAC).....	99
Tabla 4.8: Conexiones puerto GPIO front-end.....	99
Tabla 4.9: Valores de ganancia para cada ajuste de escala vertical.....	101
Tabla 4.10: Entradas y salidas del bloque de conversión a punto fijo (Simulink).....	101
Tabla 4.11: Registros implementados en interfaz Avalon-MM de control.....	102
Tabla 4.12: Conexiones para bloque de interfaz de ADC (Platform Designer).....	102
Tabla 4.13: Conexiones para bloque de disparo (Platform Designer).....	103
Tabla 4.14: Entradas y salidas del bloque de disparo (Simulink).....	107
Tabla 4.15: Registros implementados en interfaz Avalon-MM de control.....	107
Tabla 4.16: Configuración bloque DMA adquisición.....	109
Tabla 4.17: Memoria requerida para cada escala horizontal.....	110
Tabla 4.18: Conexiones para bloque DMA adquisición (Platform Designer).....	110
Tabla 4.19: Configuración bloque DMA SRC.....	114
Tabla 4.20: Configuración bloque DMA SINK.....	114
Tabla 4.21: Conexiones para bloque DMA SRC (Platform Designer).....	114
Tabla 4.22: Conexiones para bloque DMA SINK (Platform Designer).....	115
Tabla 4.23: Factores de decimación implementados para la base de tiempo de 1 s/div.....	116
Tabla 4.24: Conexiones para bloque de procesamiento (Platform Designer).....	116
Tabla 4.25: Entradas y salidas del bloque de procesamiento (Simulink).....	120
Tabla 4.26: Registros implementados en interfaz Avalon-MM de control.....	121
Tabla 4.27: Parámetros de puentes entre el HPS y la FPGA.....	122
Tabla 4.28: Vista de memoria del procesador (lightweight HPS-to-FPGA).....	123
Tabla 4.29: Recursos utilizados en la FPGA para cada bloque.....	124
Tabla 4.30: Slack de setup y hold para el diseño final (Timing Analyzer).....	126
Tabla 4.31: Frecuencias máximas para el diseño final (Timing Analyzer).....	126
Tabla 5.1: Parámetros utilizados en la calibración de ganancia.....	134
Tabla 5.2: Funciones de la biblioteca asociadas a la arquitectura de adquisición.....	137
Tabla 5.3: Funciones de la biblioteca asociadas al control del front-end analógico.....	137
Tabla 5.4: Funciones principales de interfaz de la biblioteca.....	138
Tabla 6.1: Componentes, cantidad y precio para la placa principal.....	157
Tabla 6.2: Componentes, cantidad y precio para la fuente de alimentación.....	158
Tabla 6.3: Mediciones de offset antes y después de la calibración para cada escala.....	164
Tabla 6.4: Parámetros de calibración del DAC y ADC para cada escala.....	164
Tabla 6.5: Errores de ganancia obtenidos en la calibración para cada escala.....	165

Tabla 6.6: Anchos de banda obtenidos para cada escala.....	180
Tabla 6.7: Ruido en el ADC y referido a la entrada para cada escala.....	181

Acrónimos y siglas

- AC:** Alternating Current (Corriente Alterna)
- ACP:** Accelerator Coherency Port (Puerto de Coherencia del Acelerador)
- ADC:** Analog to Digital Converter (Conversor Analógico Digital)
- ALM:** Adaptive Logic Module (Módulo Lógico Adaptativo)
- BGA:** Ball Grid Array (Matriz de Rejilla de Bolas)
- BSP:** Board Support Package (Paquete de Soporte de Placa)
- CMRR:** Common Mode Rejection Ratio (Factor de Rechazo de Modo Común)
- CPLD:** Complex Programmable Logic Device (Dispositivo Lógico Programable Complejo)
- CS:** Chip Select (Selección de Chip)
- CSR:** Control and Status Register (Registro de Estado y Control)
- DAC:** Digital to Analog Converter (Conversor Digital Analógico)
- DC:** Direct Current (Corriente Directa)
- DDR:** Double Data Rate (Doble Velocidad de Datos)
- DMA:** Direct Memory Access (Acceso Directo a Memoria)
- DNL:** Differential Non Linearity (No Linealidad Diferencial)
- DPDT:** Double Pole Double Throw (Polo Doble Vía Doble)
- DSP:** Digital Signal Processing (Procesamiento Digital de Señales)
- ECC:** Error Correction Code (Código de Corrección de Errores)
- EEPROM:** Electrically Erasable Programmable (ROM Programable y Borrable Eléctricamente)
- ESR:** Equivalent Series Resistance (Resistencia Serie Equivalente)
- FIFO:** First In First Out (Primero en Entrar Primero en Salir)
- FIR:** Finite Impulse Response (Respuesta Finita al Impulso)
- FPGA:** Field Programmable Gate Array (Arreglo de Puertas Programable en Campo)
- GPIO:** General Purpose Input/Output (Entrada/Salida de Propósito General)
- HDL:** Hardware Description Language (Lenguaje de Descripción de Hardware)
- HLS:** High Level Synthesis (Síntesis de Alto Nivel)
- HPS:** Hard Processor System (Sistema de Procesador Embebido)
- IC:** Integrated Circuit (Circuito Integrado)
- IDE:** Integrated Development Environment (Entorno de Desarrollo Integrado)

INL: Integral Non Linearity (No Linealidad Integral)

IOE: Input/Output Element (Elemento de Entrada/Salida)

IP: Internet Protocol (Protocolo de Internet)

LAB: Logic Array Block (Bloque de Arreglos Lógicos)

LED: Light Emitting Diode (Diodo Emisor de Luz)

LSB: Least Significant Bit (Bit Menos Significativo)

LVC MOS: Low Voltage Complementary Metal Oxide Semiconductor (Semiconductor Complementario de Óxido Metálico de Bajo Voltaje)

LVDS: Low Voltage Differential Signaling (Señalización Diferencial de Bajo Voltaje)

LVTTTL: Low Voltage Transistor Transistor Logic (Lógica Transistor a Transistor de Bajo Voltaje)

MAC: Media Access Control (Control de Acceso al Medio)

MBR: Master Boot Record (Registro de Arranque Principal)

MCL: Multi Level Cell (Celda de Niveles Múltiples)

MII: Media Independent Interface (Interfaz Independiente del Medio)

MLAB: Memory Logic Array Block (Bloque de Arreglos Lógicos de Memoria)

MMC: MultiMedia Card (Tarjeta Multimedia)

MMU: Memory Management Unit (Unidad de Gestión de Memoria)

MOSFET: Metal Oxide Semiconductor Field Effect Transistor (Transistor de Efecto Campo Metal Óxido Semiconductor)

MSB: Most Significant Bit (Bit Más Significativo)

ONFI: Open NAND Flash Interface (Interfaz NAND Flash Abierta)

PCB: Printed Circuit Board (Placa de Circuito Impreso)

PCI: Peripheral Component Interconnect (Interconexión de Componentes Periféricos)

PGA: Programmable Gain Amplifier (Amplificador de Ganancia Programable)

PLL: Phase Locked Loop (Lazo de Seguimiento de Fase)

PSRR: Power Supply Rejection Ratio (Factor de Rechazo a Fuente de Alimentación)

PWM: Pulse Width Modulation (Modulación de Ancho de Pulso)

QSPI: Quad Serial Peripheral Interface (Interfaz de Periféricos Serie Cuádruple)

RAM: Random Access Memory (Memoria de Acceso Aleatorio)

ROM: Read Only Memory (Memoria de Solo Lectura)

RTL: Resistor Transistor Logic (Lógica de Resistencia Transistor)

SAR: Successive Approximation Register (Registro de Aproximaciones Sucesivas)

SCU: Snoop Control Unit (Unidad de Control Snoop)

SDRAM: Synchronous Dynamic Random Access Memory (Memoria de Acceso Aleatorio Sincrónica Dinámica)

SIMD: Single Instruction, Multiple Data (Instrucción Simple Datos Múltiples)

SLC: Single Level Cell (Celda de Nivel Simple)

SMD: Surface Mount Device (Dispositivo de Montaje Superficial)

SNR: Signal to Noise Ratio (Relación Señal Ruido)

SPST: Single Pole Double Throw (Polo Simple Vía Simple)

SRAM: Synchronous Dynamic Random Access Memory (Memoria de Acceso Aleatorio Sincrónica)

SoC: System on Chip (Sistema en un Chip)

TCP: Transmission Control Protocol (Protocolo de Control de Transmisión)

THD: Total Harmonic Distortion (Distorsión Armónica Total)

UART: Universal Asynchronous Receiver Transmitter (Receptor Transmisor Universal Asincrónico)

USB: Universal Serial Bus (Bus Serie Universal)

Capítulo 1

Introducción

Los avances tecnológicos de las últimas décadas han favorecido la integración y reducción de costos de los componentes electrónicos. Hoy en día, existen nuevas arquitecturas que combinan sectores de lógica digital programable con procesadores de propósito general. Esta evolución, representa un cambio de paradigma en el diseño de sistemas de adquisición y procesamiento digital de señales.

Por un lado, las arquitecturas existentes basadas en procesadores o microcontroladores permiten implementar de manera sencilla funcionalidades relacionadas al ámbito del software. Entre ellas, se puede destacar el almacenamiento permanente de datos, la conectividad a través de internet, el diseño de interfaces gráficas, etc. Sin embargo, estos dispositivos imponen un límite en la frecuencia de operación, los tiempos de respuesta y la cantidad de procesamiento disponible.

Por otro lado, los sistemas basados en FPGA o DSP facilitan el desarrollo de hardware dedicado al control de adquisición, procesamiento de datos y almacenamiento en memoria. Esto se asocia a un aumento de la frecuencia de operación, evitando problemas de latencia, pérdida de datos, etc. Sin embargo, se hace prácticamente imposible incorporar funcionalidades propias de un sistema embebido.

El diseño de un dispositivo de adquisición con los beneficios asociados a ambas plataformas, tradicionalmente, requería el uso de una FPGA controlada por un procesador. Esta arquitectura resulta compleja y presenta varias dificultades, dado que se debe transferir una gran cantidad de datos en tiempo real entre ambos chips.

En base al análisis de las problemáticas existentes, se concluye que la nueva arquitectura basada en SoC FPGA resulta ideal para implementar un sistema que cumpla con todas las características demandadas hoy en día. Con el fin de aprovechar la arquitectura, se diseña e implementa un sistema de adquisición, procesamiento y streaming a través de internet en tiempo real.

Para lograr este objetivo, se hace uso de una plataforma de desarrollo que cuenta con un SoC FPGA, memoria RAM y conectividad Ethernet. En el sector de lógica programable, se implementa el control de flujo de adquisición y los bloques de procesamiento de la señal requeridos. Por otro lado, en el procesador se desarrolla un servidor TCP/IP, encargado de recibir comandos y enviar los datos. La placa se conecta mediante internet a una PC, que actúa como cliente, permitiendo visualizar los datos en tiempo real y configurar los diversos parámetros del dispositivo.

El sistema de adquisición cuenta con dos canales de entrada, que operan a una frecuencia de muestreo de 1 MSa/s. Adicionalmente, se desarrolla e implementa una etapa analógica, encargada del acondicionamiento de las señales de cada canal y de la conversión analógica digital.

1.1 Objetivos

Los objetivos y lineamientos que sirven de base para el desarrollo del presente proyecto son los siguientes:

- Demostrar la factibilidad del diseño de un sistema de adquisición en tiempo real basado en la arquitectura SoC FPGA.
- Implementar un sistema modular y abierto, que facilite modificaciones futuras asociadas a diversos parámetros de diseño. De esta forma, se busca facilitar la adaptación del dispositivo a diversas necesidades y proyectos futuros.
- Permitir la fácil incorporación de bloques de procesamiento digital en las diferentes etapas, sirviendo el proyecto como una plataforma para el desarrollo de productos que requieran aceleración por hardware de algoritmos DSP.
- Proveer un front-end de acondicionamiento flexible, que permita analizar y visualizar señales en diferentes rangos y escalas de voltaje.
- Hacer uso de un modelo cliente-servidor, a fin de poder utilizar el sistema en situaciones de instrumentación remota.
- Brindar una serie de utilidades y controles que faciliten el análisis y visualización de las señales adquiridas y los efectos del procesamiento aplicado.
- Facilitar el almacenamiento de los datos adquiridos, antes y luego de las diferentes etapas de procesamiento.
- Proveer una arquitectura flexible, que permita aumentar la frecuencia de muestreo y memoria de adquisición.
- Reducir los costos de diseño, implementación y mantenimiento de proyectos futuros que requieran de las funcionalidades mencionadas.

1.2 Especificación de requerimientos

El propósito de este apartado es establecer los requerimientos funcionales y no funcionales del sistema a desarrollar. Es decir, se pretende caracterizar el comportamiento del producto y sus características básicas de funcionalidad. De esta forma, la especificación sirve como base para todas las etapas de desarrollo del proyecto.

Los lineamientos generales seguidos en esta sección son los establecidos en la norma IEEE 830-1998 [1]. Teniendo en cuenta lo mencionado, el documento es de gran utilidad para todas las personas involucradas en el proyecto, ya que brinda un marco de trabajo básico y condiciona el desarrollo del mismo.

La especificación cuenta con tres secciones principales. En el apartado 1.2.1 se realiza una introducción básica del dispositivo a desarrollar y se fija el alcance del proyecto. Por otro lado, en la sección 1.2.2 se brinda una descripción general del dispositivo, su relación con otros productos y las características propias de los usuarios. Finalmente, en el apartado 1.2.3, se describen de forma detallada los requerimientos funcionales y no funcionales que debe satisfacer el sistema.

1.2.1 Ámbito del sistema

El producto a desarrollar consiste en un sistema de adquisición de señales eléctricas basado en arquitectura SoC FPGA. El mismo debe ser capaz de adquirir datos en al menos dos canales diferentes, procesarlos y enviarlos a través de internet. Las muestras son recibidas por el cliente y graficadas en tiempo real. Así mismo, el cliente debe implementar una interfaz gráfica que permita controlar el dispositivo remotamente. Dicha interfaz debe contar con las comodidades propias de visualización de señales asociadas a un osciloscopio digital.

El objetivo básico del producto es brindar una plataforma abierta, modular y flexible. Por lo que se espera que el sistema sea una base para proyectos futuros que involucren la adquisición y procesamiento en tiempo real de señales. De esta forma, las empresas se pueden enfocar en las partes específicas que generan valor agregado a sus productos y no en el diseño básico y común a todo sistema de adquisición.

De los lineamientos planteados, se deduce que el sistema debe ser modular y fácilmente modificable. Así mismo, con el objetivo de adaptarlo a diferentes proyectos, debe ser capaz de adquirir señales de diversas amplitudes y rangos de tensiones. Por último, resulta pertinente aclarar que el dispositivo no se plantea como una alternativa de bajo costo a un osciloscopio comercial.

1.2.2 Descripción general

1.2.2.1 Perspectiva del producto

El equipo consiste de dos módulos o placas, el primero tiene como funcionalidad acondicionar las señales analógicas y convertirlas al dominio digital. La segunda etapa, basada en arquitectura SoC FPGA, es la encargada de controlar la adquisición, almacenar las muestras y enviarlas a través de internet. Los datos se visualizan en tiempo real a través de un software cliente, el cual permite configurar los diversos parámetros de adquisición.

1.2.2.2 Funciones del producto

El sistema debe ser capaz de:

- Permitir adquirir apropiadamente señales de diversas amplitudes y rangos de voltajes.
- Contar con al menos dos canales de entrada.
- Permitir ajustar la cantidad de tiempo durante el cual se adquieren datos, la frecuencia equivalente de muestreo y el tamaño de memoria.
- Proveer un sistema de disparo básico que permita controlar la captura de datos.
- Enviar los datos a través de internet y visualizarlos con el menor retardo posible.
- Permitir aplicar algoritmos de procesamiento digital de señales básicos, tales como promediado de datos, decimación y filtrado.

- Brindar la posibilidad de controlar todas las funcionalidades desde la interfaz gráfica del cliente, de forma remota.

1.2.2.3 Características del usuario

Al tratarse de una plataforma que permite acelerar el desarrollo e implementación de sistemas de adquisición y procesamiento digital de señales, los usuarios que empleen el dispositivo deben ser altamente especializados y formados en dichas áreas. Algunos sectores de la electrónica que se pueden beneficiar de este desarrollo son: instrumentación electrónica, sensado remoto, equipos médicos y sistemas de control, entre otros.

1.2.2.4 Restricciones

En este apartado se listan algunas restricciones impuestas al desarrollo del dispositivo, las mismas se enumeran a continuación:

- El sistema debe estar basado en la placa de desarrollo DE0-Nano-SoC del fabricante Terasic. Esta plataforma cuenta con un SoC FPGA Cyclone V de Intel, 1 GiB de memoria RAM DDR3 y un puerto Gigabit Ethernet. Se elige este dispositivo por un tema de disponibilidad y costos, no se deben sobrepasar sus capacidades.
- El cliente de visualización debe ser multiplataforma, capaz de ejecutarse en los tres sistemas operativos de escritorio principales: Windows, Linux y Mac OS.
- La interfaz de usuario debe ser simple e intuitiva, los controles se deben agrupar y disponer de manera que faciliten el uso del sistema.
- El desarrollo debe ser abierto y modular, facilitando la incorporación de algoritmos de procesamiento y la modificación de los parámetros de diseño.

1.2.2.5 Suposiciones y dependencias

Se supone que todos los requerimientos listados en la especificación son estables en el tiempo. Es decir, no están sujetos a modificaciones durante la fase de desarrollo e implementación técnica. Tampoco existen dependencias que desencadenen una modificación del diseño del proyecto ante un cambio externo. Por otra parte, el dispositivo debe cumplir con todos los requisitos establecidos para lograr el nivel de funcionalidad, confiabilidad y seguridad deseados.

1.2.3 Requisitos específicos

En base al marco introductorio brindado en los apartados anteriores, se enumeran los diversos requerimientos del sistema identificados durante la definición del producto. Cada requerimiento cuenta con un número de referencia, que permite la trazabilidad durante la fase de desarrollo, y un nivel de prioridad, que denota el grado de importancia del mismo. Todos los requisitos representan necesidades reales, su interpretación no es ambigua y su cumplimiento es verificable.

1.2.3.1 Requerimientos funcionales

Esta categoría abarca los requerimientos asociados a funcionalidades con las que debe contar el sistema. Su cumplimiento, durante la fase de desarrollo del dispositivo, es de vital importancia para el usuario final.

Identificación	RF01
Nombre	Modelo cliente-servidor
Descripción	Las muestras adquiridas deben ser enviadas en tiempo real a través de internet a un cliente. Este último permite la visualización de los datos y el control del sistema.
Prioridad	Alta

Identificación	RF02
Nombre	Control de flujo
Descripción	El sistema debe ser capaz de capturar los datos de manera continua, actualizando la visualización con la información más reciente. Adicionalmente, se debe brindar la posibilidad de realizar una única captura ante un evento de disparo.
Prioridad	Alta

Identificación	RF03
Nombre	Arquitectura multicanal
Descripción	El sistema debe contar con dos canales de entrada, capaces de operar de manera simultánea. La arquitectura de cada canal debe ser funcionalmente idéntica, por lo que se deben incorporar etapas de acondicionamiento y memoria de adquisición independientes.
Prioridad	Alta

Identificación	RF04
Nombre	Acople de continua y alterna
Descripción	Se debe incorporar un filtro pasa-altos analógico que permita eliminar el valor medio de la señal a la entrada. Su activación, mediante el software cliente, debe ser independiente para cada canal. La frecuencia debe ser de 5 Hz asociada a una constante de tiempo de 30 ms.
Prioridad	Media

Identificación	RF05
Nombre	Escala vertical
Descripción	Se debe poder ajustar la escala vertical empleada en la visualización de los datos, entre 10 mV/div y 5 V/div, en pasos de 1-2-5. El área de visualización debe contar con 10 divisiones verticales, asociado a un rango de entrada máximo de ± 25 V. El ajuste de escala debe ser llevado a cabo de forma analógica, mediante algún mecanismo de ganancia y/o atenuación variable.
Prioridad	Alta

Identificación	RF06
Nombre	Offset vertical
Descripción	El sistema debe brindar la posibilidad de sumar a la señal de entrada un nivel de tensión de offset, de forma analógica. El rango de selección del mismo debe ser continuo y acorde a la escala vertical seleccionada. El nivel sumado debe ser independiente para cada canal.
Prioridad	Alta

Identificación	RF07
Nombre	Posición vertical
Descripción	El gráfico de ambos canales debe contar con un selector que permita ajustar la posición vertical del mismo, en un rango continuo de ± 5 divisiones, para todas las escalas verticales y de forma independiente para cada canal.
Prioridad	Alta

Identificación	RF08
Nombre	Base de tiempo
Descripción	El periodo temporal asociado a un ciclo de captura completo debe ser ajustable en el rango 10 μ s/div a 1 s/div, en pasos de 1-2-5. El área de visualización debe contar con 10 divisiones horizontales, asociadas a un periodo máximo de captura de 10 segundos.
Prioridad	Alta

Identificación	RF09
Nombre	Frecuencia de muestreo
Descripción	La velocidad de muestreo debe ser, al menos, de 1 MSa/s por canal, independientemente de la cantidad de canales activados y la base de tiempo seleccionada.
Prioridad	Alta

Identificación	RF10
Nombre	Memoria de adquisición
Descripción	Con el objetivo de cumplir con el RF08 y el RF09, el sistema debe ser capaz de almacenar en memoria un máximo de 10 millones de muestras por canal.
Prioridad	Alta

Identificación	RF11
Nombre	Sistema de disparo
Descripción	El ciclo de captura se debe poder sincronizar con un evento de disparo presente en alguno de los canales de adquisición. El sistema debe ser capaz de detectar flancos ascendentes y/o descendentes, alrededor de un nivel de tensión ajustable, en el canal elegido como fuente de disparo. Se debe incorporar algún mecanismo que brinde inmunidad ante el ruido, evitando falsos disparos. El ajuste de nivel de disparo debe ser continuo y abarcar el rango de la escala vertical seleccionada.
Prioridad	Alta

Identificación	RF12
Nombre	Posición horizontal
Descripción	La posición en pantalla del instante temporal asociado al evento de disparo debe ser ajustable. El rango de ajuste debe ser porcentual y continuo, entre 0% y 100%.
Prioridad	Alta

Identificación	RF13
Nombre	Profundidad de memoria
Descripción	El sistema debe implementar algoritmos de decimación que permitan reducir la cantidad de datos a ser enviados y graficados en el cliente. De esta manera, se brinda un ajuste de profundidad de memoria y frecuencia de muestreo equivalente. Se deben implementar todos los factores de decimación que den como resultado una cantidad de muestras par y mayor a 100. El procesamiento de los datos debe ser aplicado en la lógica programable.
Prioridad	Alta

Identificación	RF14
Nombre	Algoritmo de decimación normal
Descripción	Sea D el factor de decimación asociado al RF13, se debe incorporar un algoritmo capaz de seleccionar la primer muestra de cada intervalo de decimación y descartar las D-1 muestras restantes.
Prioridad	Alta

Identificación	RF15
Nombre	Algoritmo de decimación de alta resolución
Descripción	Sea D el factor de decimación asociado al RF13, se debe incorporar un algoritmo capaz de realizar un promedio de las D muestras que componen cada intervalo de decimación.
Prioridad	Media

Identificación	RF16
Nombre	Algoritmo de decimación de detección de pico
Descripción	Sea D el factor de decimación asociado al RF13, se debe incorporar un algoritmo capaz de seleccionar el máximo de un intervalo de decimación y luego el mínimo del siguiente intervalo. Este proceso se debe repetir de manera alternada.
Prioridad	Media

Identificación	RF17
Nombre	Área de visualización
Descripción	El área de visualización, en el software cliente, debe contar con los siguientes elementos gráficos: grillas verticales y horizontales, escalas numéricas verticales y horizontales, marcadores de posición verticales y horizontales e indicador de nivel de disparo.
Prioridad	Media

Identificación	RF18
Nombre	Visualización single-plot y multi-plot
Descripción	Se debe brindar la posibilidad de graficar ambos canales de adquisición de manera superpuesta (single-plot), o si el usuario lo desea, en áreas separadas (multi-plot).
Prioridad	Media

Identificación	RF19
Nombre	Modo Zoom
Descripción	Se debe incorporar una modalidad de Zoom, que permita analizar en detalle una fracción del periodo de tiempo asociado a una captura completa.
Prioridad	Baja

Identificación	RF20
Nombre	Persistencia
Descripción	El cliente debe permitir graficar de manera superpuesta los últimos N ciclos de captura completos. Los valores posibles para N deben ser: 2, 4, 8, 16, 32, 64 y 128.
Prioridad	Baja

Identificación	RF21
Nombre	Promediado
Descripción	El cliente debe contar con una funcionalidad que permita graficar el promedio de los últimos N ciclos de captura completos. Los valores posibles para la cantidad de promediado N deben ser: 2, 4, 8, 16, 32, 64 y 128.
Prioridad	Media

Identificación	RF22
Nombre	Ajustes de visualización
Descripción	El área de visualización en el cliente debe ser configurable, de acuerdo a las necesidades del usuario. Se deben incorporar controles que permitan ajustar la visibilidad de los diversos elementos gráficos, analizados en el RF17.
Prioridad	Baja

Identificación	RF23
Nombre	Calibración automática
Descripción	Es deseable la incorporación de los mecanismos que le permitan al usuario calibrar el sistema de manera automática. Es decir, sin recurrir a la ejecución de pasos manuales o la necesidad de emplear instrumentos externos.
Prioridad	Baja

Tabla 1.1: Requerimientos funcionales del sistema de adquisición

1.2.3.2 Requerimientos no funcionales

En este apartado se agrupan otro tipos de requisitos, no relacionados explícitamente a la funcionalidad directa del sistema de adquisición. Los mismos se asocian a características generales que favorecen o complementan la usabilidad del dispositivo.

Identificación	RNF01
Nombre	Protección de entrada
Descripción	Teniendo en cuenta el máximo rango de entrada, de ± 25 V, el sistema debe ser capaz de soportar tensiones en el rango ± 50 V, a lo largo de un periodo arbitrario de tiempo.
Prioridad	Alta

Identificación	RNF02
Nombre	Velocidad de respuesta
Descripción	A fin de mejorar la usabilidad del sistema, el tiempo de respuesta entre un cambio de configuración y el inicio de una nueva adquisición debe ser menor a 0,5 segundos.
Prioridad	Alta

Identificación	RNF03
Nombre	Velocidad de actualización
Descripción	El tiempo entre dos adquisiciones no debe ser mayor a 50 ms. Este valor permite garantizar la fluidez de visualización de los datos en el modo de adquisición continua.
Prioridad	Media

Identificación	RNF04
Nombre	Portabilidad
Descripción	El sistema de adquisición, en su totalidad, debe tener un tamaño menor a (20 cm x 20 cm x 20 cm), asegurando la portabilidad.
Prioridad	Baja

Identificación	RNF05
Nombre	Precisión
Descripción	El error relativo en el voltaje medido no debe ser mayor a un 1% bajo ninguna configuración de escala vertical. La resolución de los conversores analógico digital a utilizar debe ser de 12 bits.
Prioridad	Alta

Identificación	RNF06
Nombre	Respuesta en frecuencia
Descripción	La etapa de acondicionamiento analógica debe tener una respuesta en frecuencia plana y con un ancho de banda mayor a 500 kHz, sin importar la selección de escala vertical.
Prioridad	Alta

Identificación	RNF07
Nombre	Impedancia de entrada
Descripción	La impedancia de las entradas analógicas debe ser de 1 M Ω en paralelo con una carga capacitiva menor a 20 pF. Este requerimiento reduce el efecto del instrumento por sobre el funcionamiento del dispositivo medido.
Prioridad	Alta

Tabla 1.2: Requerimientos no funcionales del sistema de adquisición

1.2.3.3 Interfaces externas

El equipo debe contar con un puerto Ethernet que le brinde comunicación a través de internet, permitiendo controlar el sistema de forma remota. De lo mencionado, se deduce que la computadora que actúa de cliente debe contar también con conectividad de red. La comunicación puede ser punto a punto, o ambos dispositivos pueden pertenecer a una misma red TCP/IP. Por otro lado, las entradas de las señales a adquirir, para ambos canales, deben emplear conectores de tipo BNC.

1.3 Descripción general del sistema

En el apartado 1.2, dedicado a los requerimientos, se caracteriza a grandes rasgos la funcionalidad esperada del producto. Esta sección, tiene como objetivo brindar algunos detalles técnicos sobre la implementación del sistema. Teniendo en cuenta las funcionalidades y requerimientos del sistema de adquisición, en la figura 1.1 se presenta un diagrama en bloques con la arquitectura básica del dispositivo. En la misma, se describe a grandes rasgos los componentes elementales del sistema de adquisición y su funcionalidad asociada.

La arquitectura propuesta sirve como base y lineamiento general para el diseño real del sistema, llevado a cabo en capítulos posteriores. En dicha fase de desarrollo técnico, se determinan todos los parámetros asociados a cada una de las etapas. Es decir, se brindan los modelos matemáticos, diagramas en bloques, diagramas esquemáticos y algoritmos de programación necesarios para la construcción del dispositivo.

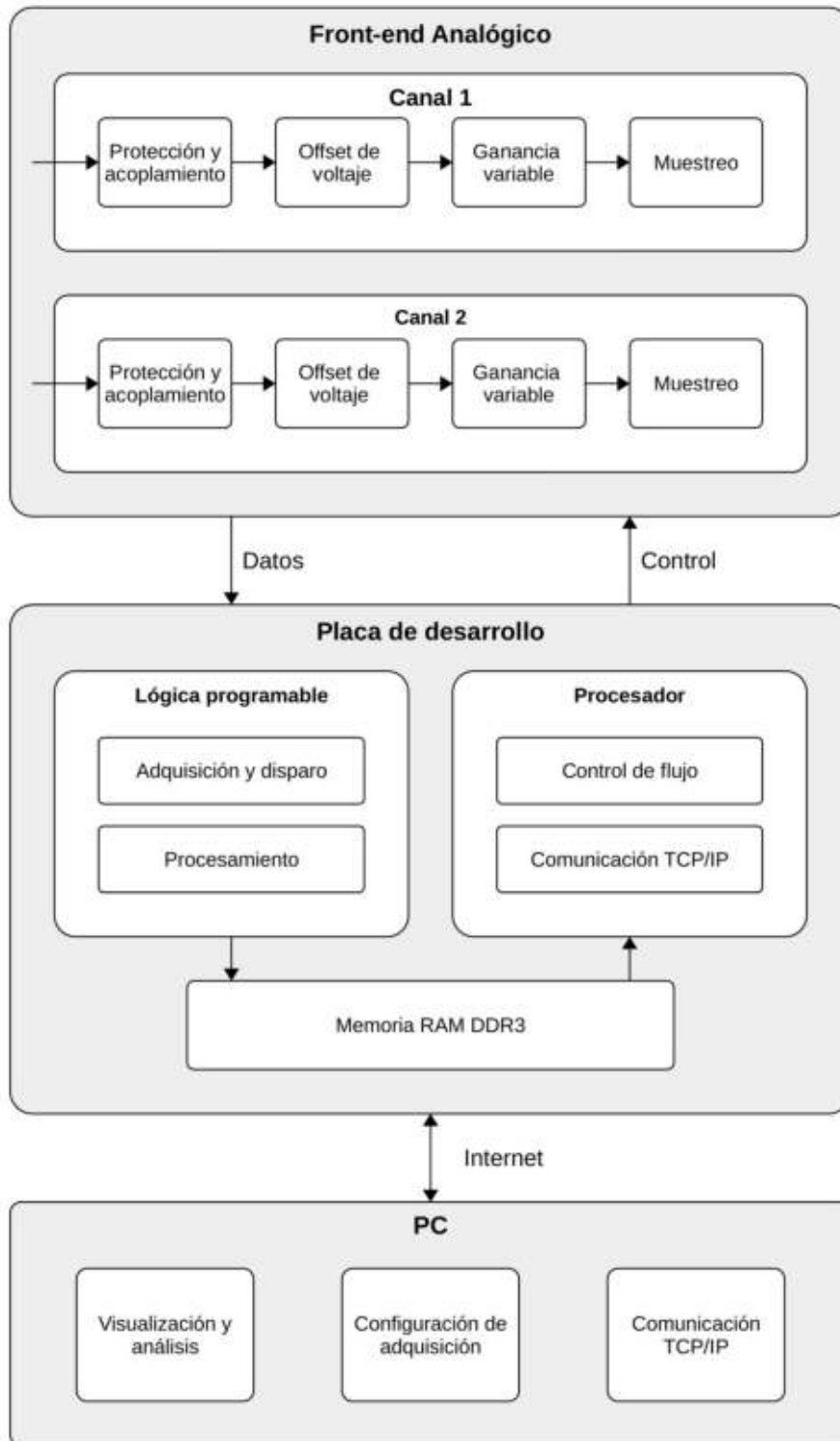


Figura 1.1: Diagrama en bloques general del sistema de adquisición

1.3.1 Front-end analógico

Todo sistema de adquisición debe contar con una etapa analógica previa a la digitalización de los datos. Esto se debe a la necesidad de acondicionar la señal de entrada y llevarla al rango fijo en el

que operan los convertidores analógico digital. Del análisis de los requerimientos se deduce que el diseño del front-end debe contar con un filtro de acople en alterna y protección de entrada. Para poder cumplir con el RF06, se debe incorporar algún mecanismo que permita sumar un voltaje ajustable a la señal de entrada. Por otro lado, a fin de implementar las diversas escalas verticales, se debe incluir un bloque de atenuación y/o ganancia programable. Estas cuatro etapas funcionales pueden ser observadas en el diagrama en bloques de la figura 1.1. La comunicación entre el front-end analógico, a implementar en un PCB aparte, y el SoC FPGA, consiste en el flujo de datos de los ADC y las señales de control necesarias.

1.3.2 Placa de desarrollo

La plataforma de desarrollo, basada en arquitectura SoC FPGA, debe ser capaz de controlar la adquisición y enviar los datos a través de internet. Las diversas funcionalidades asociadas a este elemento del sistema se deben implementar en la lógica programable y en el software ejecutado por el procesador. Por otro lado, el flujo de datos en tiempo real debe ser almacenado en la memoria RAM principal del sistema.

De acuerdo a la figura 1.1, el diseño a implementar en la lógica programable tiene como función recibir las muestras de los convertidores y almacenarlas en memoria. De forma adicional, se deben incorporar los mecanismos que permitan detectar los eventos de disparo y controlar el proceso de adquisición. Por otro lado, los algoritmos de procesamiento, asociados al RF13, también deben ser implementados en el sector de lógica programable.

El procesador, que corre un sistema operativo Linux, tiene como objetivo leer los datos adquiridos y procesados para enviarlos a través de internet. A fin de cumplir con este objetivo, se debe implementar un servidor TCP/IP basado en la conectividad Ethernet de la placa de desarrollo. Adicionalmente, el servidor debe recibir los comandos de configuración desde el cliente y controlar los diversos bloques de la FPGA y el front-end analógico.

1.3.3 Cliente

El elemento más importante del sistema de adquisición, de cara a la interacción con el usuario, es el software cliente. El programa a desarrollar se ejecuta en una PC de propósito general, conectada a través de internet a la placa de desarrollo. El cliente debe contar con una interfaz gráfica que permita cumplir con los requerimientos de visualización RF17 a RF22. Adicionalmente, el cliente debe contar con los controles asociados a los diversos ajustes del sistema de adquisición. Estos parámetros, son enviados en forma de comandos al servidor, ya que influyen en la configuración de las diferentes etapas de acondicionamiento y adquisición.

1.4 Análisis de mercado

Una vez definidas las funcionalidades y especificaciones del sistema de adquisición, se procede a realizar un análisis de mercado. El objetivo del mismo es evaluar algunos productos comerciales existentes, se analizar sus características principales y compararlas con el dispositivo a desarrollar. De esta forma, se pretende establecer un precio de venta y por consiguiente, un costo objetivo de fabricación para el producto, asegurando la competitividad.

Una de las empresas predominantes en el diseño de software y hardware para la adquisición, control y procesamiento de datos es National Instruments. Los sistemas desarrollados por el fabricante son utilizados, principalmente, en la automatización de mediciones e instrumentación virtual en el ámbito industrial. National Instruments cuenta con hardware que permite la entrada y salida de parámetros físicos tales como tensión, corriente, temperatura, resistencia, vibración, etc.

La gama de productos está compuesta por dos tipos de dispositivos de entrada y salida [2]. Por un lado, existen sistemas controlados por una PC externa, ya sean placas de adquisición PCI/USB/Ethernet o chasis modulares. Por otro lado, el fabricante dispone de sistemas embebidos, basados en procesadores de propósito general y, en algunos casos, lógica programable. Esta última categoría cuenta con una serie de chasis modulares denominada CompactRIO, basada en arquitectura SoC FPGA, cuyo propósito y funcionalidad es similar a la del presente proyecto.



Figura 1.2: Chasis cRIO-9063 de National Instruments

En la figura 1.2 se observa el dispositivo cRIO-9063, se trata del controlador CompactRIO de menor costo a la fecha del proyecto. El mismo integra cuatro ranuras para módulos de entrada/salida, un SoC FPGA Xilinx Zynq 7020, 256 MiB de RAM, 512 MiB de memoria Flash, Gigabit Ethernet y USB 2.0 [3]. El dispositivo ejecuta un sistema operativo Linux en tiempo real y permite implementar algoritmos de control y procesamiento, tanto en el procesador como en la lógica programable.

El diseño en base a la plataforma CompactRIO, al igual que otros productos del fabricante, se lleva a cabo principalmente en el software LabVIEW. El mismo consiste en un entorno de desarrollo interactivo que permite diseñar el sistema en base a un lenguaje de programación gráfico. Cabe destacar que National Instruments cuenta con otros paquetes de software, orientados a aplicaciones de diferente índole.

Con respecto al costo de esta alternativa, a la fecha del proyecto, el controlador cRIO-9063 tiene un precio de US\$1.765. Por otro lado, a modo de ejemplo, un módulo de entrada NI-9215 [4] de 4 canales, ± 10 V, resolución de 16 bits y frecuencia de muestreo de 100 kSa/s por canal, tiene un precio de US\$685. Para el desarrollo, resulta necesario contar una licencia del paquete LabVIEW

y los módulos LabVIEW Real-Time y LabVIEW FPGA, con un costo asociado al software de US\$ 9.695. Del análisis se concluye que para contar con una plataforma de adquisición de National Instruments, basada en arquitectura SoC FPGA, resulta necesario invertir al menos US\$12.005.

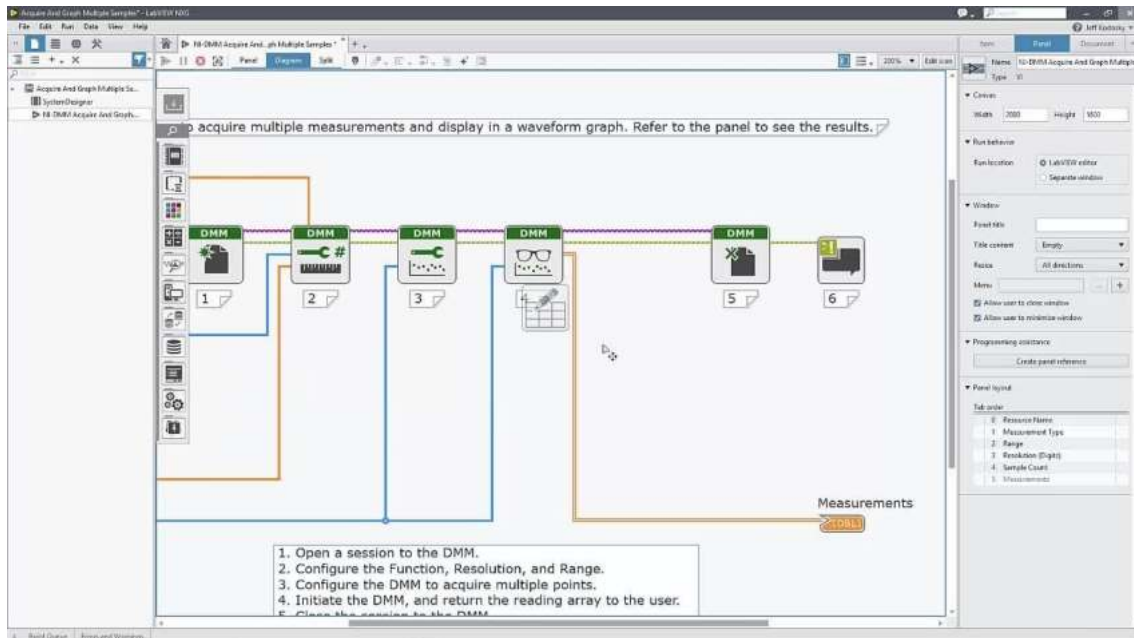


Figura 1.3: Entorno de desarrollo LabVIEW de National Instruments

La alta modularidad del hardware y la flexibilidad asociada al entorno de programación, permiten la rápida implementación de prototipos y sistemas reales de adquisición, procesamiento e instrumentación. Sin embargo, en la mayoría de los casos, el alto costo asociado a los productos resulta prohibitivo. Adicionalmente, la arquitectura cerrada y propietaria del hardware y software no permite su modificación. Este factor puede ser limitante a la hora de desarrollar un producto final con un mayor grado de personalización. Teniendo en cuenta lo mencionado, se pretende implementar un sistema que solucione las problemáticas analizadas, es decir, que represente una alternativa libre y de menor costo.

Con el objetivo de establecer un precio de venta, se analizan alternativas comerciales que pertenecen al segmento del dispositivo a desarrollar. Se seleccionan dos productos disponibles en el mercado: Digilent OpenScope MZ y Red Pitaya STEMLab 125-14, los cuales pueden ser observados en las figuras 1.4 y 1.5, respectivamente. Se trata de dispositivos destinados a la implementación de instrumentos virtuales, la adquisición y el procesamiento de datos. La placa OpenScope MZ se basa en un microcontrolador Microchip PIC32MZ y cuenta con dos entradas analógicas de 6,25 MSa/s y una salida analógica de 10 MSa/s [5]. Por otra parte, la plataforma STEMLab 125-14, se basa en un SoC FPGA Xilinx Zynq y cuenta con dos entradas y dos salidas analógicas, ambas funcionando a 125 MSa/s [6].

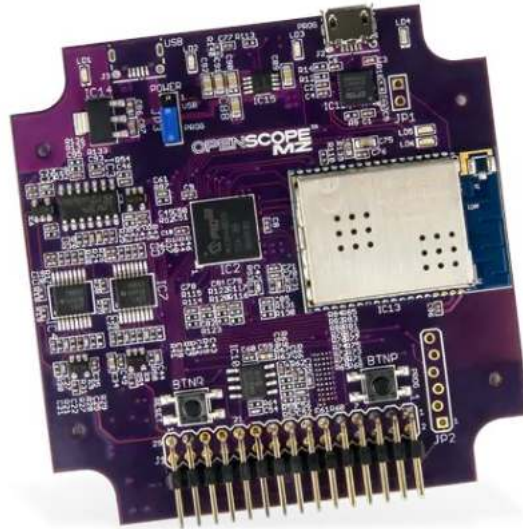


Figura 1.4: Digilent OpenScope MZ

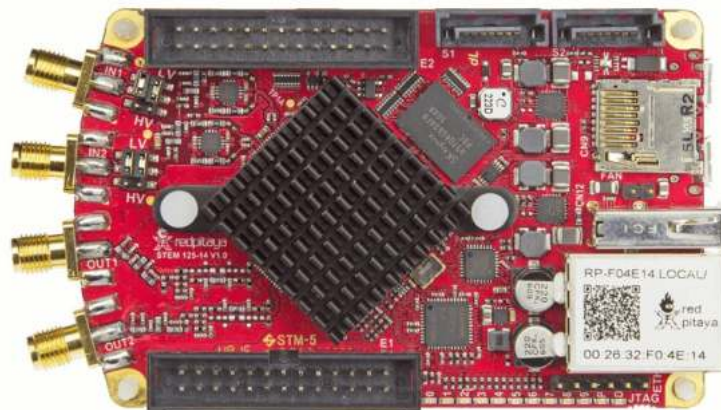


Figura 1.5: Red Pitaya STEMLab 125-14

Ambos productos cuentan con software que le permite actuar como osciloscopio y generador de funciones. Por otro lado, la placa STEMLab 125-14 ofrece otros tipos de instrumentos, entre los que destaca un analizador lógico y un medidor LCR. Estas funcionalidades son opcionales y requieren de software y hardware extra, por lo que no son tenidas en cuenta en la comparación. Existe una clara diferencia en especificaciones y arquitectura entre los dos dispositivos, siendo la placa STEMLab 125-14 un producto superior.

Estas dos alternativas se seleccionan para su análisis debido a que ambas cuentan con software y firmware de código abierto. Por otro lado, a diferencia del dispositivo a desarrollar, ninguna de las dos placas cuenta con un diseño de hardware libre, que permita su libre modificación. Cabe destacar que no se localizó en el mercado productos con especificaciones similares al proyecto que tengan un diseño de hardware libre.

	Proyecto	Diligent OpenScope MZ	Red Pitaya 125-14
Basado en	SoC FPGA Intel Cyclone V	Microcontrolador Microchip PIC32MZ	SoC FPGA Xilinx Zynq 7010
Hardware libre	Si	Solo esquemático	No
Firmware/Software libre	Si	Si	Si
Comunicación	Ethernet 1 Gbit/s	WiFi 802.11b/g – USB 2.0	Ethernet 1 Gbit/s
Memoria RAM	1 GiB DDR3	512 KiB SRAM	512 MiB DDR3
Cliente de visualización	Software multiplataforma	Interfaz Web	Interfaz Web
Entradas analógicas			
Cantidad de canales	2	2	2
Resolución ADC	12 bits	12 bits	14 bits
Frecuencia muestreo	1 MSa/s por canal	6,25 MSa/s por canal	125 MSa/s por canal
Tamaño de buffer	10 MSa por canal	32 kSa por canal	16 kSa por canal
Ancho de banda	0,5 MHz	2 MHz	40 MHz
Rango	±25 V, protección ±50 V	±20 V, protección ±40 V	±20 V, protección ±30 V
Acople alterna	5 Hz, control por software	No	No
Offset de voltaje	DAC 12 bits	PWM 8 bits filtrado	No
Ganancia variable	Si, selección por software	4 valores prefijados, selección por software	2 valores prefijados, selección manual en placa
Algoritmos de decimación acelerados por hardware	Normal, detección de pico, alta resolución	No	No
Calibración automática	Si	Si	No
Captura simple y continua	Si	Si	Si
Ajuste escala horizontal y vertical	Si	Si	Si
Ajuste posición horizontal y vertical	Si	Si	Si
Disparo por flanco	Si	Si	Si
Disparo con histéresis	Si	No	No
Selección de fuente de disparo y nivel	Si	Si	Si
Entrada auxiliar de disparo	No	Si	Si
Promediado	2 a 128 capturas	No	No
Persistencia	2 a 128 capturas	No	No
Operaciones matemáticas	No	Si	Si
Mediciones automáticas	No	Si	Si
Cursores	No	Si	Si
FFT	No	Por software, sobre buffer de adquisición de 32 kSa	Por software, sobre buffer de adquisición de 16 kSa
Salidas analógicas			
Cantidad de canales	-	1	2
Resolución DAC	-	10 bits	14 bits
Frecuencia muestreo	-	10 MSa/s por canal	125 MSa/s por canal
Tamaño de buffer	-	25 kSa por canal	16 kSa por canal
Ancho de banda	-	1 MHz	50 MHz
Rango	-	±1,5 V	±1 V
Impedancia de salida	-	No especificado	50 Ω
Formas de onda	-	Senoidal, triangular, pulso, rampa, DC	Senoidal, triangular, pulso, rampa, DC, PWM
Precio		US\$ 150	US\$ 370

Tabla 1.3: Comparación del dispositivo con las alternativas comerciales directas existentes

En la tabla 1.3 se enumeran las especificaciones del dispositivo a desarrollar y las alternativas de mercado seleccionadas. Se observa que el proyecto, al estar orientado a la adquisición, cuenta con una etapa de acondicionamiento que otorga mayor funcionalidad. Resulta de importancia el mayor rango de entrada, el acople de alterna seleccionable por software y el offset de voltaje

analógico de mayor precisión. Por otro lado, la implementación de ganancia variable, de acuerdo a la escala seleccionada, permite aprovechar mejor el rango dinámico de los ADC y obtener una mejor resolución. Esto es cierto incluso en el caso de la plataforma STEMLab 125-14, basada en un DAC de mayor precisión, ya que la misma solo cuenta con dos valores de ganancia, cuya selección se realiza mediante un jumper en la placa. Por otro lado, se considera que la frecuencia de muestreo del sistema a implementar resulta ser un factor limitante, por lo que debe ser incrementada en modificaciones futuras.

Con respecto a la adquisición y visualización de datos, los dos productos comerciales cuentan con funcionalidades similares. Se considera que, en ambos aspectos, el sistema de adquisición a implementar se encuentra a la par de las alternativas de mercado. Este no es el caso con respecto a las salidas analógicas, que no forman parte del proyecto.

Del análisis llevado a cabo se concluye que, de acuerdo a las especificaciones, el producto propuesto se ubica entre ambas alternativas analizadas. En base a lo mencionado, se establece un precio de mercado de US\$250, el cual no debe ser sobrepasado si se desea contar con cierto margen de ganancia.

1.5 Organización del documento

El presente informe consta de siete capítulos en total, los cuales describen las diferentes fases de desarrollo del proyecto. En el capítulo 2, se analiza en detalle la arquitectura interna del dispositivo SoC FPGA usado y la placa de desarrollo seleccionada. La lectura de este capítulo es opcional y se presenta a modo de referencia y complemento del resto del informe.

Se dedica el capítulo 3 a la fase de diseño técnico de la etapa de acondicionamiento analógica. En las diversas secciones internas se analizan los criterios de diseño, se justifican las elecciones de componentes y se presentan todos los diagramas esquemáticos, ecuaciones de diseño y simulaciones pertinentes.

El capítulo 4 se asocia a la arquitectura de adquisición, es decir, al diseño digital llevado a cabo en el sector de lógica programable del SoC FPGA. En el mismo, se describen los diversos bloques que componen el diseño, su implementación técnica y su rol en el ciclo de adquisición y procesamiento de los datos. Por otra parte, los componentes de software, asociados tanto al cliente como al servidor, son analizados en detalle en el capítulo 5.

La etapa de verificación del diseño teórico se lleva a cabo en el capítulo 6. En el mismo, se describe el proceso de armado de un prototipo del sistema, y se exponen las mediciones que permiten validar los cálculos y simulaciones realizadas. Adicionalmente, se llevan a cabo pruebas integrales del sistema en su totalidad, funcionando bajo condiciones reales. Finalmente, las conclusiones sobre la ejecución del proyecto se desarrollan en el capítulo 7. En el mismo, se evalúa el grado de cumplimiento de los objetivos generales y se analizan posibles mejoras a futuro.

Capítulo 2

Arquitectura SoC FPGA

El objetivo de este capítulo es presentar y caracterizar el SoC FPGA utilizado como elemento central de la arquitectura de adquisición. Estos dispositivos integran, en un mismo chip encapsulado, un procesador y un sector de lógica digital programable. Ambos subsistemas se encuentran comunicados internamente a través de diversos puentes con un elevado ancho de banda y baja latencia.

El fabricante de FPGA elegido es Intel, previamente Altera, debido a la familiaridad con las herramientas, flujo de trabajo y arquitectura de estos dispositivos [7]. La familia SoC FPGA de Intel se encuentra integrada por las series Cyclone, Stratix y Arria [8]. Estas tres gamas de productos difieren en la cantidad de lógica programable disponible, el procesador utilizado y los bloques funcionales embebidos en la FPGA. Teniendo en cuenta los requerimientos del proyecto, se decide utilizar la serie Cyclone V de menor costo.

Inicialmente se evaluó la posibilidad de integrar directamente la FPGA con el resto del sistema de adquisición en una única placa. Esto implicaría diseñar un PCB de varias capas con gran cantidad de pistas diferenciales de alta velocidad, diversos voltajes de alimentación [9] y circuitos integrados con gran cantidad de pines (por ejemplo, 672 para el caso de la FPGA). Esta opción fue descartada en una etapa temprana, ya que excede el marco de este proyecto.

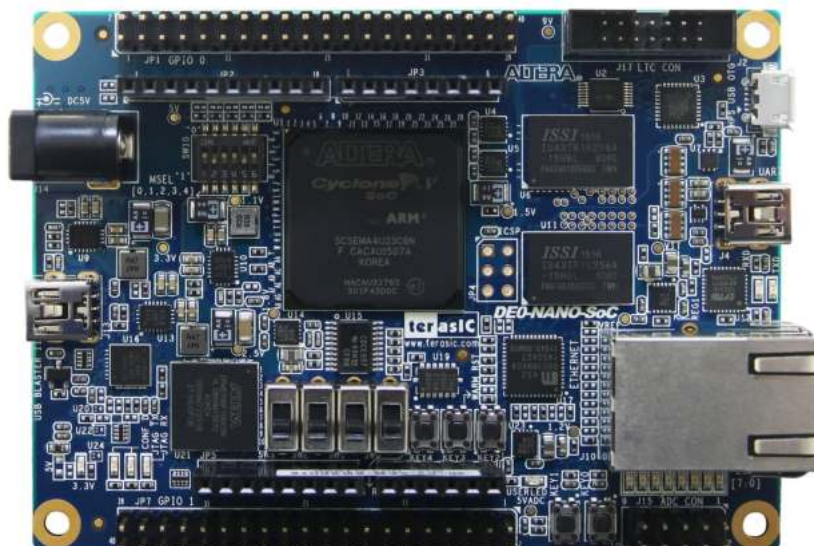


Figura 2.1: Placa de desarrollo Terasic DE0-Nano-SoC

Teniendo en cuenta lo mencionado, se decide utilizar una placa de desarrollo DE0-Nano-SoC [10] del fabricante Terasic, con base en Taiwán, la cual puede ser observada en la figura 2.1. Esta plataforma integra un SoC FPGA Cyclone V 5CSEMA4U23C6N con 1 GiB de memoria RAM DDR3 y bloques de diversa índole. El costo, a la fecha del proyecto, es de US\$ 90, siendo la alternativa más económica basada en esta arquitectura.

Este capítulo se encuentra dividido en cuatro secciones principales, las dos primeras tienen como objetivo presentar el área de lógica programable y el sistema de procesamiento del dispositivo utilizado, respectivamente. En la sección 2.3, por otro lado, se analiza la arquitectura de interconexión interna del SoC FPGA y la comunicación entre ambos subsistemas. Finalmente, se dedica el apartado 2.4 a describir la placa de desarrollo utilizada, presentando los componentes y características de mayor importancia.

2.1 Lógica programable

En este apartado se describen los componentes pertenecientes al sector de lógica programable de los dispositivos Cyclone V. Estos elementos pueden ser observados en la siguiente vista general del circuito integrado [8]:

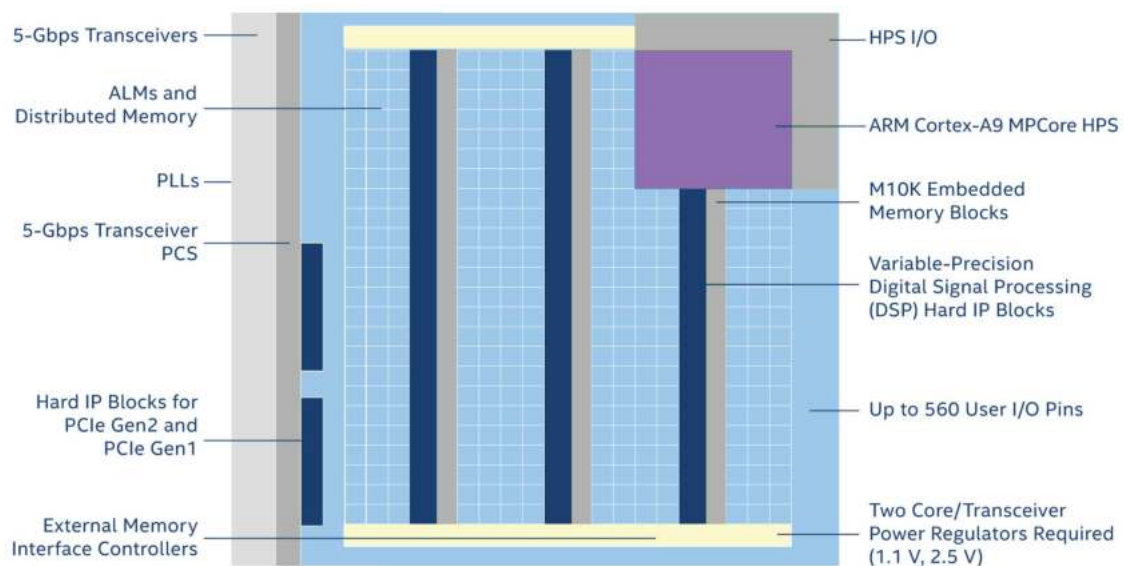


Figura 2.2: Arquitectura SoC FPGA Cyclone V

En la esquina superior derecha de la figura 2.2 se ubica el sistema de procesamiento o HPS (Hard Processor System), el cual se describe en secciones posteriores. Con respecto a la FPGA, los elementos básicos involucrados en la síntesis de lógica programable son los siguientes: ALM (lógica general), bloques M10K (memoria embebida) y bloques DSP (operaciones aritméticas).

Por otra parte, dependiendo del modelo de FPGA en particular, se incorporan bloques de hardware que brindan funcionalidades adicionales. Algunos ejemplos son: controladores de memoria externa, lazos de enganche de fase, transceptores de alta velocidad y controladores PCI Express. Con respecto a los últimos dos elementos mencionados, no son analizados, ya que no forman parte del modelo de dispositivo utilizado en particular.

2.1.1 Módulos de lógica adaptativa

En la arquitectura Cyclone V, los bloques básicos utilizados en la síntesis de lógica son denominados módulos de lógica adaptativa o ALM (Adaptive Logic Module). Este elemento es capaz de implementar lógica combinacional, lógica secuencial o actuar como memoria embebida [11]. En el diagrama de la figura 2.3 se presenta la arquitectura interna simplificada de estos bloques [12].

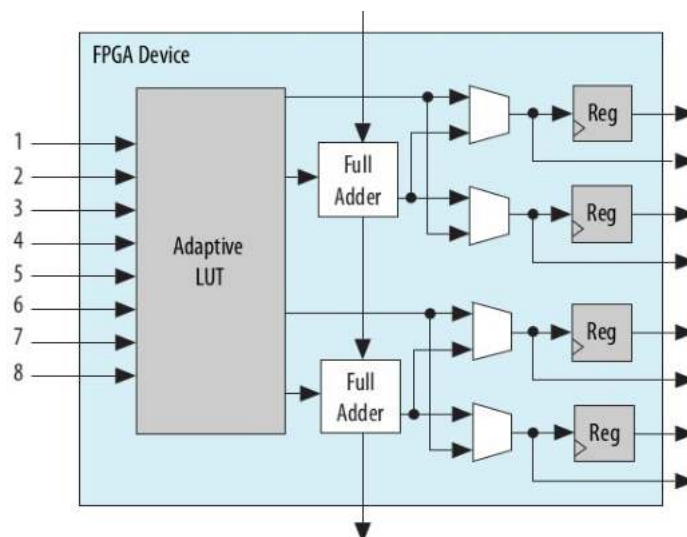


Figura 2.3: Arquitectura interna de ALM en Cyclone V

El ALM está compuesto por una tabla de búsqueda (Lookup Table) de ocho entradas, dos sumadores completos, multiplexores y cuatro registros de 1 bit cada uno. La tabla de búsqueda puede ser dividida y configurada de distintas formas, permitiendo implementar varios modos de funcionamiento. Por otro lado, los registros cuentan con funcionalidades de carga síncrona y borrado síncrono y asíncrono.

Cada grupo de 10 ALM contiguos forman un arreglo de bloques lógicos o LAB (Logic Array Block). En esta arquitectura, aproximadamente un 25% de los LAB presentes pueden actuar como elementos de memoria (MLAB). Este tipo de memoria está optimizada para implementar registros de desplazamiento en algoritmos DSP o buffers FIFO de baja capacidad. Cada ALM es capaz de almacenar 64 bits (32×2), por lo que la capacidad total de un MLAB es de 640 bits.

El modelo de FPGA utilizado cuenta con 15.880 módulos ALM, asociados a un total de 63.520 registros de 1 bit (4 por ALM) y 1.588 LAB (cada uno formado por 10 ALM). Por otro lado, 370 de

los 1.588 LAB pueden actuar como elementos de memoria, con una capacidad total de 236.800 bits o 29,6 kB (640 bits por cada bloque).

2.1.2 Bloques M10K

Los dispositivos Cyclone V cuentan con memoria embebida en la FPGA, la cual se encuentra agrupada en bloques denominados M10K [11]. Cada uno cuenta con una capacidad total de 10.240 bits, es decir 1,28 kB. Estos elementos son utilizados para generar sectores de almacenamiento de alta velocidad, como por ejemplo, RAM, ROM, buffers FIFO y registros de desplazamiento.

Entre las funcionalidades de este bloque, se destaca la posibilidad de implementar memorias de doble puerto. En este caso, cada interfaz puede utilizar un ancho de palabra distinto y operar a una frecuencia distinta. Adicionalmente, los bloques M10K soportan bits de paridad, corrección de errores (ECC) e inicialización de contenido en el encendido. Cabe destacar que el modelo de FPGA utilizado cuenta con 270 de estos elementos, es decir, un total de 345,6 kB de memoria (2.764.800 bits).

2.1.3 Bloques DSP

Uno de los elementos lógicos más importantes de la arquitectura Cyclone V son los bloques de procesamiento digital de señales o DSP (Digital Signal Processing) [11]. Se trata de componentes flexibles, capaces de adaptar su estructura interna a diversos algoritmos DSP. El modelo de FPGA utilizado cuenta con 84 de estos bloques, permitiendo cada uno implementar, entre otras, las siguientes operaciones aritméticas:

- Tres multiplicadores de 9x9 bits, con salidas independientes.
- Dos multiplicadores de 18x18 bits o 18x19 bits, con salidas independientes.
- Dos multiplicadores de 18x18 bits o 18x19 bits, cuyas salidas son sumadas (o restadas).
- Un multiplicador de 18x25, 20x24, o 27x27 bits.
- Un multiplicador de 18x18 bits o 18x19 bits, cuya salida es sumada a otra entrada de 36 bits.

A modo de ejemplo, en la figura 2.4 se puede ver un diagrama en bloques para el tercer caso presentado en la lista [11]. Es decir, dos multiplicaciones cuyas salidas son sumadas para obtener el resultado final.

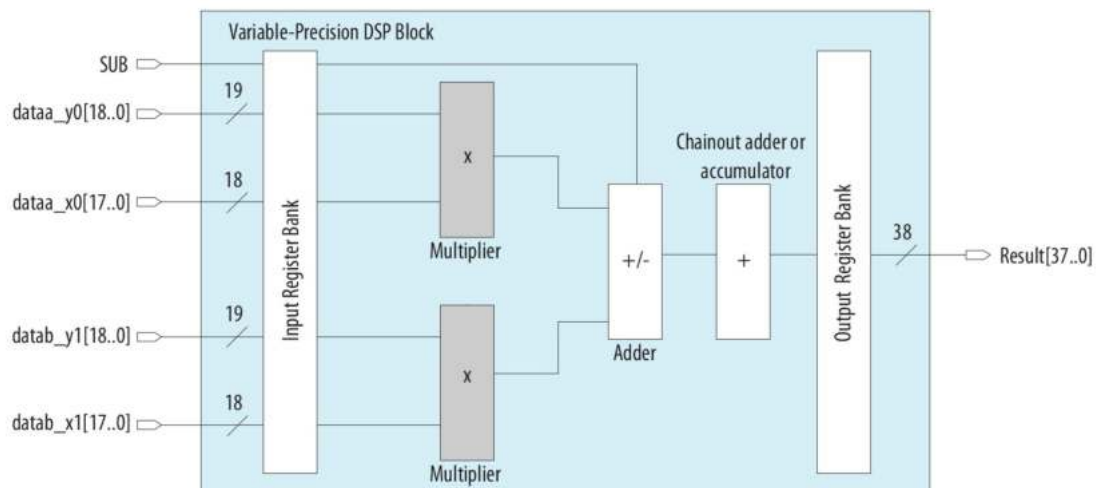


Figura 2.4: Ejemplo de modo de operación de bloque DSP en Cyclone V

Como se puede observar, las operaciones aritméticas son realizadas con precisión completa, por lo que el tamaño de palabra a la salida es de 38 bits. Este es el caso también para el resto de modos de funcionamiento presentados en la lista. Cabe destacar que los bloques DSP cuentan con otros elementos funcionales internos, asociados a funcionalidades más sofisticadas. A modo de ejemplo, el bloque integra acumuladores de salida, registros de coeficientes para la implementación de filtros FIR, etc.

2.1.4 Bloques I/O

En la arquitectura Cyclone V, la interfaz con otros componentes externos se lleva a cabo a través de elementos de entrada y salida o IOE (Input Output Elements) [11]. Estos bloques se agrupan en varios bancos diferentes, de acuerdo a sus ubicaciones en el circuito integrado. Cada banco cuenta con tensiones de alimentación independientes, por lo que se pueden utilizar diferentes niveles lógicos en cada grupo.

Los elementos de entrada y salida soportan diferentes estándares de voltajes, entre los que destacan LVCMOS y LVTTTL. Dependiendo de la alimentación del banco, se puede operar bajo las siguientes tensiones: 1,2 V, 1,5 V, 1,8 V, 2,5 V, 3,0 V y 3,3 V. El diseño de la interfaz provee gran flexibilidad, permitiendo ajustar para cada IOE parámetros como el slew rate, corriente máxima, resistencia de pull-up, salida open-drain, etc. Adicionalmente, los bloques cuentan con resistencias de terminación internas de tipo serie, paralelo o diferencial, cuyo valor puede ser programado.

Por otro lado, todas las entradas y salidas son capaces de implementar protocolos diferenciales, entre los que destaca LVDS (Low Voltage Differential Signaling). En este caso, los IOE cuentan con SERDES (serializadores deserializadores) que convierten automáticamente el flujo de datos serie a una interfaz paralelo de menor frecuencia, dentro del rango de velocidad de la FPGA.

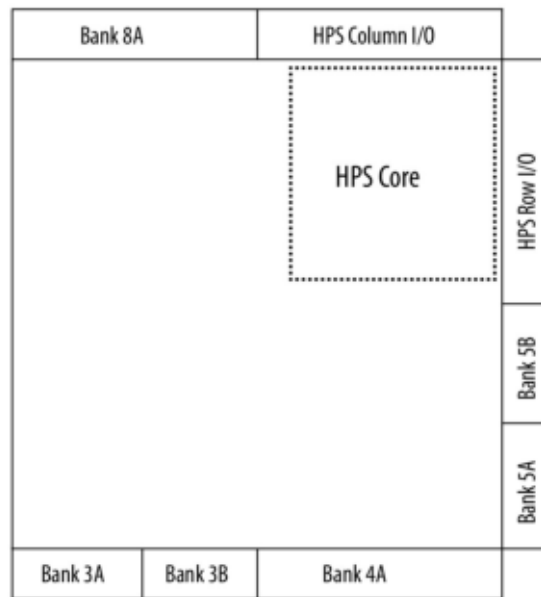


Figura 2.5: Bancos de entrada y salida en Cyclone V SE A4 (vista superior)

En la figura 2.5 se puede observar la ubicación de los bancos de entrada y salida en el modelo de dispositivo utilizado [11]. En este caso, existen 145 bloques IOE que forman parte de la lógica programable y 181 bloques IOE asociados al HPS. El primer grupo permite implementar en la FPGA un total de 32 transmisores diferenciales y 37 receptores, bajo la interfaz LVDS. El elevado grado de conectividad se asocia al empleo de un encapsulado BGA de 672 pines.

2.1.5 Lazos de enganche de fase

El sector de lógica programable en los dispositivos Cyclone V integra varios PLL, capaces de generar relojes en un amplio rango de frecuencias [11]. Esto reduce la necesidad de recurrir a osciladores externos y lógica que los sintetice, simplificando el diseño del sistema. El dispositivo utilizado en particular cuenta con cinco lazos de enganche de fase con ocho salidas cada uno.

Cada PLL emplea un preescalador de entrada N , un contador de realimentación M y varios contadores de salida C . Por ende, en este modo de operación, la frecuencia sintetizada cumple la relación $M/(N \times C)$ con la entrada. Cabe destacar que cada contador es de 9 bits, por lo que su rango total es de 512 valores.

Por otro lado, todos los PLLs son capaces de operar en modo fraccional, permitiendo obtener otras relaciones de frecuencias. En este caso, un modulador Delta-Sigma modifica ciclo a ciclo el valor del contador de realimentación M . Adicionalmente, el diseño del PLL provee, entre otras funcionalidades, la posibilidad de ajustar individualmente el retardo de fase y ciclo de trabajo para cada salida de reloj.

2.1.6 Controlador de memoria

La FPGA empleada en el diseño cuenta con dos controladores de memoria externa tipo SDRAM [11]. Uno de los controladores forma parte del área de lógica programable, mientras que el segundo pertenece al subsistema de procesamiento o HPS. En esta sección se describe el primero, el cual soporta hasta 4 GiB de memoria.

Este periférico es compatible con memorias de tipo LPDDR2, DDR2 y DDR3, con anchos de palabra entre 8 y 32 bits. El controlador cuenta con dos señales de chip select y permite implementar corrección de errores (ECC) de manera opcional. La frecuencia de operación máxima es de 333 MHz para LPDDR2 y 400 MHz para los dos protocolos restantes. Resulta pertinente aclarar que este controlador no es utilizado en el diseño. Esto se debe a que la placa de desarrollo DE0-Nano-SoC no cuenta con memoria externa SDRAM conectada al sector de lógica programable.

2.2 Sistema de procesamiento

En este apartado se describen los componentes pertenecientes al HPS (Hard Processor System) de los dispositivos Cyclone V. En la siguiente figura, se observa un esquema del HPS en su totalidad [8]:

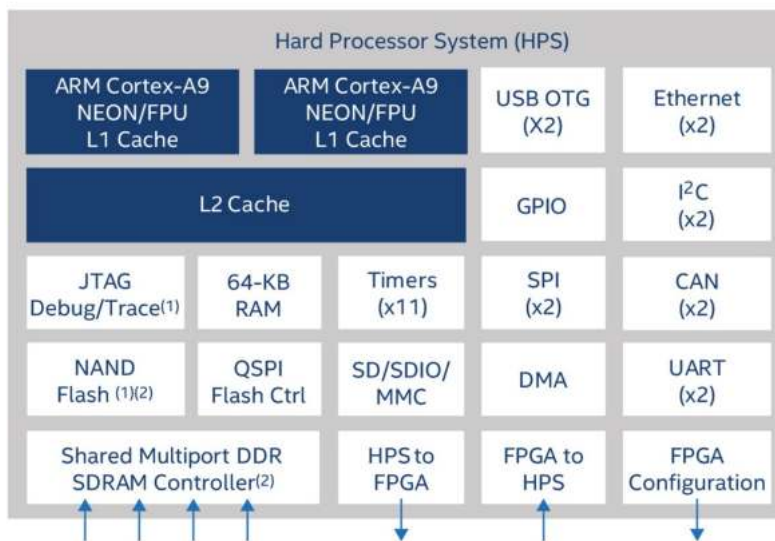


Figura 2.6: Vista de procesador y periféricos en Cyclone V

Básicamente, el sistema de procesamiento está compuesto por los siguientes grupos de elementos funcionales [13]:

- Procesador: ARM Cortex-A9 doble núcleo.
- Controladores de memoria permanente: NAND, QSPI y SD/MMC.
- Controlador de memoria principal: SDRAM.

- Memorias integradas: RAM y ROM.
- Periféricos de comunicación: Ethernet, USB, I2C, SPI, CAN, UART, etc.
- Periféricos generales: GPIO, DMA, Timers, etc.
- Interfaz con la FPGA: configuración y comunicación.

El diseño del procesador incluye una amplia variedad de periféricos de diversa índole, brindando flexibilidad al sistema y reduciendo la cantidad de componentes externos. Cabe destacar que, con el objetivo de tener un sistema de procesamiento funcional, se debe integrar el SoC FPGA en una placa con memorias externas tipo Flash (permanente) y RAM (temporal). En las siguientes secciones se profundiza en la descripción de los subsistemas más importantes que conforman el HPS.

2.2.1 Procesador y caché

El HPS cuenta con un procesador doble núcleo ARM Cortex-A9 de 32 bits, corriendo a 925 MHz. Este procesador se basa en un diseño superescalar, con una capacidad de cálculo de 2,5 DMIPS/MHz por núcleo. La microarquitectura cuenta con características típicas de procesadores de alta performance, como por ejemplo, ejecución especulativa y predicción de salto. Algunos de los componentes principales del procesador se listan a continuación [14]:

- Motor de procesamiento NEON: Se trata de un coprocesador numérico con unidad de punto flotante vectorial de doble precisión. Permite mejorar la performance en operaciones aritméticas de diversa índole. Cada núcleo cuenta con unidades independientes.
- Unidad de gestión de memoria (MMU): Este elemento, presente en ambos núcleos, permite implementar un esquema de memoria virtual mediante paginación. Este tipo de direccionamiento es utilizado en sistemas operativos complejos, como por ejemplo, Linux.
- Caché de nivel 1 (L1): Cada núcleo cuenta con un total de 64 KiB de memoria caché, separada en datos (32 KiB) e instrucciones (32 KiB). Es el tipo de memoria más rápida presente en el sistema, pero su tamaño es reducido.
- Caché de nivel 2 (L2): Existe un total de 512 KiB de caché unificada, es decir, compartida por ambos procesadores. En este caso, no existe separación entre datos e instrucciones. Esta memoria tiene conexión directa con el controlador de memoria RAM principal del HPS.
- Snoop Control Unit (SCU): Este elemento, ubicado entre ambos niveles de caché, tiene la responsabilidad de mantener la coherencia de datos entre ambos núcleos del procesador. Por ejemplo, este subsistema es capaz de copiar datos entre las memorias caché L1 de ambos núcleos.
- Controlador de interrupciones: Su funcionalidad principal es permitir a periféricos del HPS (y de la FPGA) interrumpir al procesador para informar eventos que requieran algún tipo de acción.

2.2.2 Controladores de memoria Flash

El HPS cuenta con los siguientes controladores de memoria de almacenamiento permanente [13]:

- NAND: Soporta memorias de celda simple (SLC) o múltiple (MCL), con y sin corrección de errores (ECC). Adicionalmente, este periférico incluye un motor de acceso directo a memoria interno.
- SPI: Soporta memorias de tipo NOR con anchos de buses de 1, 2 o 4 bits (Quad SPI). Es capaz de controlar hasta cuatro memorias Flash diferentes, ya que cuenta con cuatro señales de chip select.
- SD/MMC: Es utilizado con tarjetas de memoria de tipo Secure Digital (SD) y MultiMedia Card (MMC) con líneas de datos de 1, 4 o 8 bits. Es compatible con tamaños de bloques de hasta 64 KiB e incluye un controlador de acceso directo a memoria interno.

Los dispositivos Cyclone V cuentan con una serie de pines que, de acuerdo a cómo se conecten, permiten seleccionar el tipo de memoria utilizada durante el arranque. En el caso específico de la placa de desarrollo DE0-Nano-SoC, se hace uso del controlador SD/MMC, ya que cuenta con una ranura para tarjeta SD.

2.2.3 Controlador de memoria RAM

Este elemento actúa de interfaz con la memoria externa principal utilizada por el procesador y es compatible con los siguientes protocolos: DDR2, LPDDR2 y DDR3 [13]. En todos los casos, se soportan anchos de palabra de 8, 16 y 32 bits, y se pueden direccionar hasta 4 GiB de memoria.

Con el objetivo de mejorar el ancho de banda, el controlador es capaz de agrupar y reordenar los accesos a memoria, aprovechando la arquitectura interna de los módulos RAM. Opcionalmente, se brinda la posibilidad de implementar corrección de errores (ECC) y protección de acceso a memoria. Estas funcionalidades resultan de utilidad en aplicaciones que requieren altos porcentajes de confiabilidad y estabilidad.

El controlador incluye varios puertos, permitiendo que diversos dispositivos accedan a los módulos de memoria RAM. En caso de presentarse transacciones en simultáneo, el arbitraje se realiza en base a algoritmos y niveles de prioridades establecidos. Los elementos del sistema que utilizan los diversos puertos son presentados en secciones posteriores.

2.2.4 Memoria integrada

El HPS cuenta con 64 KiB de memoria ROM, en la cual se almacena el código ejecutado en el arranque del dispositivo [13]. Esta memoria viene programada de fábrica y no es modificable. Adicionalmente, el IC incorpora 64 KiB de memoria RAM de tipo estática, empleada como memoria principal en la primera fase de arranque.

2.2.5 Periféricos de comunicación

Con el objetivo de brindar conectividad al sistema de procesamiento, se incorporan periféricos de comunicación de diversa índole. Los más importantes se presentan a continuación [13]:

- Ethernet (x2): Se incluyen dos controladores MAC (capa de acceso al medio) que soportan estándares de 10/100/1000 Mbit/s. Este periférico cuenta con capacidades DMA y es capaz de comunicarse con controladores PHY (capa física) bajo los estándares MII, GMII, RGMII y SGMII.
- USB (x2): La conectividad USB consiste en dos controladores USB 2.0 OTG de Synopsys, capaces de transmitir datos a velocidades de 1, 5, 12 y 480 Mbit/s. El estándar OTG le permite al periférico actuar tanto en modo host (anfitrión) como device (dispositivo).
- I2C (x4): Se integran cuatro periféricos de comunicación de tipo I2C, que soportan velocidades de hasta 400 kbit/s con modos de direccionamiento de 7 y 10 bits. Cada controlador puede actuar como maestro o esclavo en el bus.
- UART (x2): El HPS cuenta con dos periféricos UART que pueden operar hasta 6,25 MBaud. Cada uno implementa buffers internos de transmisión y recepción de 128 bytes y soporta las señales de control de flujo.
- CAN (x2): Estos controladores están basados en el modelo Bosch D_CAN, compatible con la especificación 2.0. Ambos periféricos soportan interrupciones e identificadores de 11 y 29 bits y operan a velocidades de hasta 1 Mbit/s.
- SPI (x4): Existen dos periféricos SPI de tipo maestro y dos de tipo esclavo. Ambas modalidades pueden funcionar en modos full-duplex y half-duplex con longitudes de datos entre 4 y 16 bits. Por otro lado, cada controlador maestro soporta hasta cuatro señales de chip select.

2.2.6 Periféricos generales

En esta categoría se agrupan otros periféricos generales que, típicamente, forman parte de un sistema digital de esta clase. Entre ellos se destacan [13]:

- Timers (x4): Se integran cuatro timers de 32 bits, que son capaces de generar interrupciones periódicas. Cabe destacar que estos temporizadores cuentan con funcionalidad limitada, comparada con los módulos presentes en microcontroladores.
- Watchdog Timers (x2): Al igual que los temporizadores de propósito general, utilizan registros de 32 bits. Su objetivo es resetear el sistema de manera automática, en caso que el procesador no reinicie el temporizador periódicamente (medida de seguridad).
- Controlador DMA: El sistema cuenta con un controlador de acceso directo a memoria de ocho canales. Este mismo, es utilizado principalmente por dispositivos que no cuentan con la capacidad de actuar como maestro en el bus (I2C, UART, etc.). Otros periféricos más complejos, como por ejemplo, USB o Ethernet, pueden realizar transacciones sin recurrir a un motor DMA externo.

- Control general del sistema: En esta categoría se agrupan componentes del HPS tales como: clock manager, system manager y reset manager. Estos dispositivos se encargan de la administración y configuración general del sistema. Entre otras cosas, controlan los diversos relojes, señales de reset, modos de energía y multiplexado de pines.

2.2.7 Interfaz con la FPGA

Existen dos tipos de elementos asociados a la comunicación y control del sector de lógica programable del sistema [13]:

- Controlador (FPGA Manager): Tiene la funcionalidad principal de permitir la configuración, total o parcial, de la FPGA desde el procesador, mediante una interfaz interna de 32 bits. El controlador soporta archivos de configuración comprimidos e incluso encriptados (bajo el estándar AES).
- Puentes de comunicación: Permiten la transferencia de datos entre el sector de lógica programable y el HPS. Teniendo en cuenta la importancia de los puentes de comunicación en los dispositivos SoC FPGA, se dedica a continuación un apartado completo a la descripción de la arquitectura de buses interna.

2.3 Interconexión del sistema

Esta sección tiene como objetivo presentar la manera en que todos los componentes del sistema se comunican. Adicionalmente, se describen los mecanismos que permiten la transferencia de datos entre el sector de lógica programable (FPGA) y el procesador embebido (HPS). El diseño del sistema se basa en una matriz de interconexión, lo que permite aumentar el ancho de banda y reducir los tiempos de respuesta. El empleo de una matriz de interconexión permite el acceso en simultáneo de los dispositivos de tipo maestro a los esclavos que conforman el sistema.

2.3.1 Diagrama en bloques del sistema

En la figura 2.7, se presenta un diagrama en bloques donde se puede observar la implementación práctica de la arquitectura de buses interna, para el caso de los dispositivos Cyclone V [12]. El esquema presenta una vista general de todos los componentes y sus conexiones.

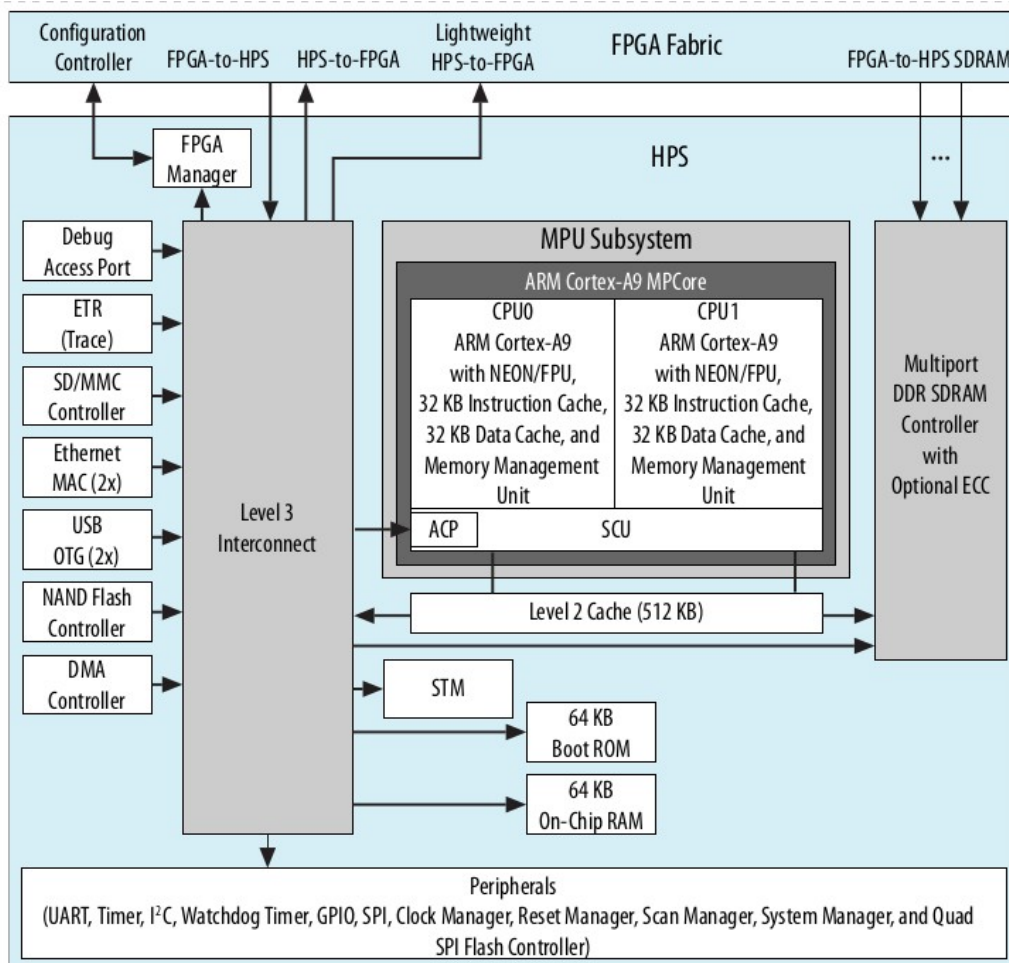


Figura 2.7: Diagrama en bloques de conexiones en Cyclone V

En la parte superior del esquema, en la figura 2.7, se ubica el área de lógica programable y sus conexiones con el resto del sistema. Por otro lado, en la parte inferior se presenta el HPS, cuyo elemento principal es la matriz de interconexión, denominada Level 3 Interconnect. Los periféricos que cuentan con capacidad de actuar como maestro se presentan a la izquierda de la matriz, mientras que en la parte inferior se puede observar el resto de periféricos esclavos de menor complejidad. Cabe destacar que el sentido de las flechas indica cuál de los dispositivos actúa como maestro y cuál como esclavo.

En las siguientes subsecciones, se describen las conexiones presentes en el esquema para cada componente principal del sistema [13]. Así mismo, se analizan los mecanismos de transferencia de datos entre la FPGA y el HPS, y los casos de usos típicos para cada puente de interconexión [15].

2.3.1.1 Procesador

El procesador cuenta con tres conexiones principales con el resto del sistema: matriz principal, controlador de memoria RAM y puerto ACP. Las primeras dos conexiones ocurren a través del controlador de memoria caché L2 y el procesador actúa como maestro. La última conexión,

consiste en un puerto tipo esclavo en el SCU utilizado por otros componentes con capacidad de actuar como maestros.

La conexión con la matriz, es típicamente utilizada por el procesador para acceder a los periféricos del HPS, la memoria embebida y la FPGA, esta última a través de los puentes HPS-to-FPGA. Por otro lado, la conexión directa con el controlador SDRAM, le brinda al procesador acceso a la memoria principal externa. El uso de un puerto dedicado en el controlador SDRAM, permite obtener un gran ancho de banda y baja latencia.

Por último, el procesador expone a la matriz de interconexión un puerto de tipo esclavo, denominado Accelerator Coherency Port (ACP). Esta interfaz es empleada por maestros que desean acceder a la memoria principal a través de la caché del procesador. La diferencia, con respecto a acceder directamente la memoria RAM, es que se obtiene una vista coherente a nivel caché de los datos. Es decir, este puerto permite acceder a los últimos datos modificados por el procesador y almacenados en caché.

2.3.1.2 FPGA

En esta sección se describen los elementos involucrados en la comunicación entre el HPS y el área de lógica programable. Como se puede observar en el esquema de la figura 2.7, existen cinco conexiones entre ambos subsistemas del dispositivo, analizadas a continuación:

- **FPGA manager:** Consiste en el puerto utilizado por el procesador para configurar la lógica programable en base a una imagen binaria almacenada en memoria permanente.
- **Lightweight HPS-to-FPGA:** Se trata de un puente de ancho fijo (32 bits) de baja velocidad y reducida latencia. Permite a los maestros, del lado del HPS, acceder a esclavos implementados en la lógica programable. Típicamente este puente es utilizado por el procesador para acceder a los registros de estado y control de componentes implementados en la FPGA.
- **HPS-to-FPGA:** Se trata de un puente de alta velocidad, cuyo ancho es configurable entre los valores de 32, 64 y 128 bits. Permite el acceso de maestros del lado del HPS a esclavos implementados en la FPGA (al igual que la versión lightweight). Típicamente, debido a su alta velocidad, este puente es utilizado para acceder a bloques de memoria implementados en la FPGA.
- **FPGA-to-HPS:** Esta interfaz permite que maestros en la FPGA se comuniquen a través de la matriz con diversos componentes del HPS de tipo esclavo. Típicamente es utilizada por lógica implementada en la FPGA que necesita acceder y controlar periféricos que forman parte del HPS. Adicionalmente, a través del puerto ACP, permite acceder a la memoria RAM de manera coherente con el procesador. Al igual que el caso anterior, se trata de una interfaz de alta velocidad cuyo ancho de palabra es configurable, bajo los mismos valores.
- **FPGA-to-SDRAM:** Consiste en cuatro puertos de lectura y cuatro de escritura (de 64 bits cada uno) que pueden ser combinados para formar varias interfaces entre la FPGA y la memoria principal. Se pueden implementar varios buses de comunicación con anchos de palabra entre 32 y 256 bits (dentro de las capacidades del controlador). Se trata de una interfaz de alta velocidad, utilizada por componentes tipo maestro en la FPGA para acceder a la memoria RAM. Este puente es el mecanismo de comunicación recomendado

para realizar grandes transferencias de datos, en los casos que no se requiere coherencia de datos con el procesador.

2.3.1.3 Periféricos maestros

En esta categoría se agrupan los elementos a la izquierda de la matriz de la figura 2.7, los cuales cuentan con la capacidad de actuar como maestro en el sistema. Ejemplos de estos dispositivos son: interfaces de comunicación de alta velocidad (USB, Ethernet) y controladores de memoria permanente (NAND, SD/MMC). Estos dispositivos integran motores de DMA internos que les permite realizar transferencias de grandes bloques de datos sin intervención del procesador.

Adicionalmente, forma parte de este grupo el controlador de acceso directo a memoria embebido en el HPS. Este dispositivo es típicamente utilizado por el sistema operativo para copiar sectores en la memoria RAM. Teniendo en cuenta lo mencionado, en caso de ejecutarse un SO, no es recomendable su uso para transferencia de datos desde o hacia la FPGA.

2.3.1.4 Periféricos esclavos

En la parte inferior de la figura 2.7, se presenta un grupo de elementos de menor complejidad, los cuales típicamente no transfieren grandes flujos de datos. Estos periféricos son generalmente accedidos y controlados a través de la matriz por el procesador, o en algunos casos la FPGA. Algunos ejemplos son: periféricos de comunicación de baja velocidad (SPI, UART, I2C), Timers, etc.

2.3.2 Mapa de memoria

Debido a la arquitectura del dispositivo, existen tres vistas diferentes del espacio de memoria en el sistema. Con el objetivo de complementar la información de la sección anterior, se presenta en la figura 2.8 los tres esquemas de memoria para los maestros en los dispositivos Cyclone V [13].

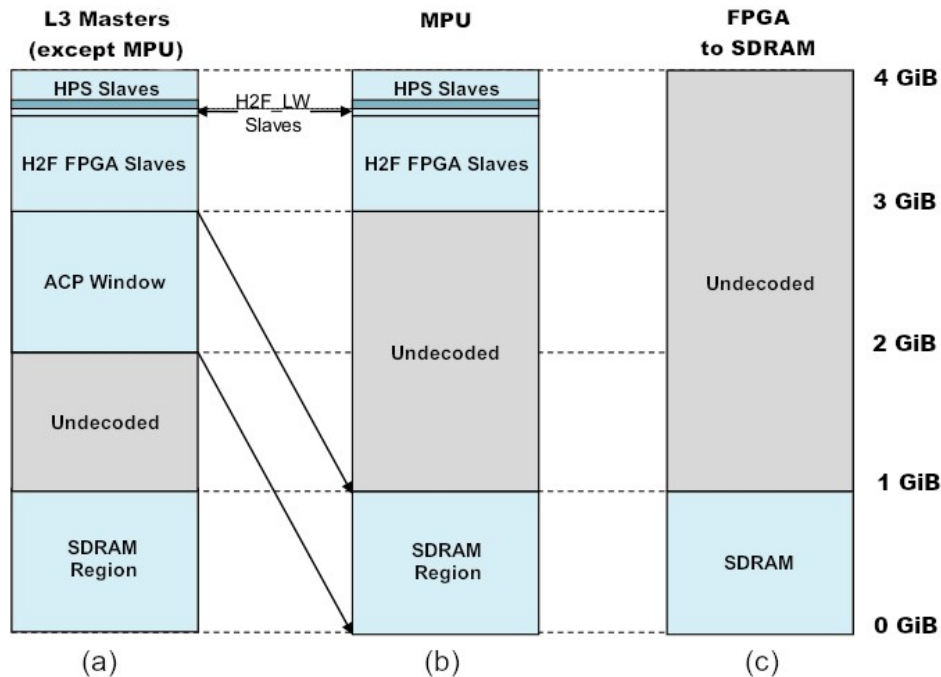


Figura 2.8: Vista de memoria (a) Matriz L3 (b) Procesador (c) Puente FPGA-to-SDRAM

En el diagrama 2.8(a), se observa la vista para el caso de todos los maestros conectados a la matriz de interconexión principal (excepto el procesador). Como fue mencionado, este grupo involucra periféricos tales como el controlador USB y Ethernet. Adicionalmente, este mapa aplica a la lógica implementada en la FPGA que acceda al HPS a través del puente FPGA-to-HPS.

En el segundo diagrama, figura 2.8(b), se observa la vista para ambos núcleos del procesador principal. Este esquema es muy similar al anterior, sin embargo, no cuenta con el sector dedicado al puerto ACP. Esto se debe a que los accesos del procesador a la memoria RAM son realizados directamente a través de los dos niveles de caché, por lo que son coherentes.

Finalmente, en el esquema de la figura 2.8(c), se presenta la vista de memoria de los componentes implementados en la FPGA conectados al puente FPGA-to-SDRAM. En este caso, se trata de dispositivos tipo maestro que acceden directamente a la memoria, de forma no coherente, a través de puertos dedicados del controlador SDRAM. Como es de esperar, el mapa está compuesto únicamente por la memoria RAM externa.

2.4 Placa de desarrollo

Finalmente, esta sección tiene como objetivo describir la placa de desarrollo Terasic DE0-Nano-SoC que se utiliza en el presente trabajo. A continuación, se presenta un diagrama en el que se pueden observar todos los elementos que forman parte de la plataforma [10]:

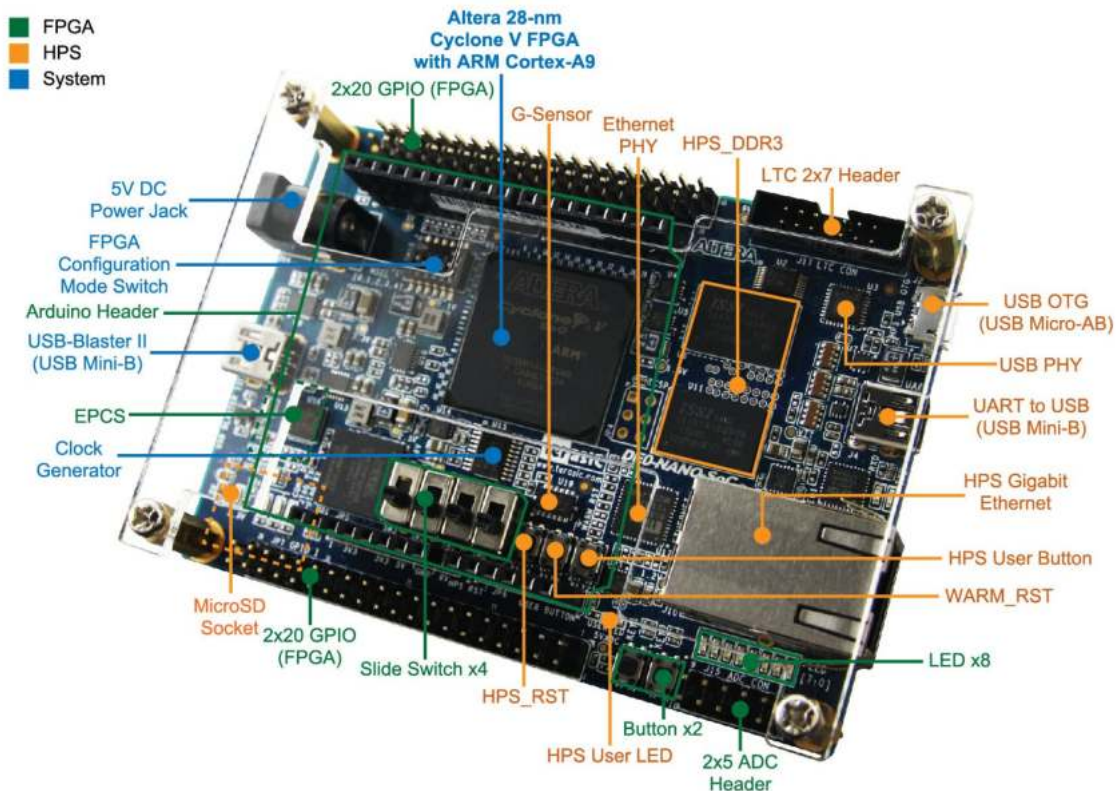


Figura 2.9: Componentes de la placa de desarrollo Terasic DE0-Nano-SoC

En la figura 2.9, se observa que los componentes se encuentran agrupados dentro de las siguientes categorías: FPGA (verde), HPS (naranja) y sistema (celeste). Los primeros dos grupos, se refieren a elementos conectados a la lógica programable o el procesador, respectivamente. El último grupo, se refiere a elementos generales que proveen funcionalidad a ambos subsistemas.

2.4.1 Elementos generales

La placa cuenta con los siguientes bloques de propósito general [10]:

- **Gestión de energía:** La alimentación es brindada por una fuente conmutada externa de 5 V y 2 A, a través de un conector tipo “barrel jack”. La placa emplea reguladores lineales y convertidores DC/DC para obtener los diversos niveles de voltaje requeridos. A modo de ejemplo, se generan tensiones de 9 V, 5 V, 3,3 V, 2,5 V y 1,5 V, entre otras.
- **Generación de reloj:** A fin de obtener las diferentes frecuencias de reloj empleadas por el SoC FPGA, la placa hace uso de un PLL CDCE937 [16] de Texas Instruments. El mismo, cuenta con una memoria EEPROM interna y se encuentra programado para generar relojes de 24 MHz, 25 MHz y 50 MHz. En particular, el sector de lógica programable cuenta con tres señales de reloj de entrada, de 50 MHz cada una.
- **Selección de modo:** La placa cuenta con botones deslizables, de tipo DIP switch, que permiten seleccionar el modo de configuración de la FPGA en el arranque. La primera opción, consiste en programar la FPGA a partir de un archivo almacenado en una memoria

Flash permanente presente en la plataforma (EPCS). La segunda opción, permite realizar la programación de la lógica programable mediante código ejecutado en el HPS.

- USB Blaster II: La plataforma incluye los circuitos necesarios para establecer una comunicación JTAG, tanto con el HPS como con la FPGA. Esta interfaz permite programar la FPGA de manera temporal y escribir la imagen de configuración en la memoria EPCS. Adicionalmente, la conexión JTAG brinda capacidades de depuración de ambos subsistemas, permitiendo ver las señales internas de la FPGA y contenidos de memoria RAM. La interfaz está compuesta por un puerto USB Mini-B y un microcontrolador USB 2.0 Cypress CY7C68013A [17].

2.4.2 Elementos conectados a la FPGA

La placa de desarrollo cuenta con los siguientes elementos conectados al sector de lógica programable [10]:

- EPCS: Se trata de una memoria Flash que puede ser utilizada para configurar la FPGA en el arranque (sin recurrir al HPS). La selección de este modo de funcionamiento se realiza mediante el switch que fue descrito en el apartado anterior.
- LEDs (x8): La placa incluye ocho LEDs de color verde conectados a pines digitales de la FPGA. Cabe destacar que son encendidos con un nivel lógico positivo (3,3 V).
- Botones pulsables (x2): Existen dos botones pulsables, conectados a la FPGA a través de un circuito integrado encargado de eliminar los rebotes. Un nivel lógico de 0 se corresponde con el caso del botón pulsado, siendo el nivel lógico 1 el estado de reposo.
- Botones deslizables (x4): Se incorporan cuatro switches de dos posiciones, conectados directamente a la FPGA sin un mecanismo de eliminación de rebotes.
- ADC: La plataforma de desarrollo integra un conversor de 12 bits, ocho canales y una frecuencia máxima de 500 kSa/s. El ADC utilizado es el modelo LTC2308 [18] de Analog Devices, conectado a la FPGA a través de una interfaz SPI de cuatro líneas.
- Headers Arduino: La placa cuenta con cinco conectores, cuya geometría y pinout es compatible con la placa Arduino Uno R3. Esta interfaz brinda 16 señales digitales conectadas a la FPGA, ocho entradas analógicas conectadas al ADC y diversos niveles de alimentación (3,3 V, 5 V, 9 V y GND).
- Header ADC: Consiste en un conector de 10 pines (2 x 5) que brinda acceso a las ocho entradas analógicas del ADC. En las dos conexiones restantes, se pueden obtener 5 V y GND.
- Puertos GPIO: La conectividad principal de la placa de desarrollo se da a través de dos conectores de 40 pines cada uno (2x20). Cada puerto cuenta con 36 líneas digitales conectadas al sector de lógica programable del SoC FPGA, así como también tensiones de 5 V, 3,3 V y dos pines de GND.

2.4.3 Elementos conectados al HPS

Finalmente, se presentan a continuación los elementos que componen el subsistema de procesamiento [10]:

- Memoria RAM: Consiste en 1 GiB de memoria DDR3 conectada al controlador SDRAM multipuerto del HPS. En esta implementación en particular, la memoria se encuentra repartida en dos módulos IS43TR16256A [19] de ISSI y funciona a una frecuencia de 400 MHz.
- Ethernet: La plataforma integra un IC que actúa de PHY (capa física), conectado a uno de los dos controladores Ethernet (acceso al medio) integrados en el HPS. El controlador utilizado es el modelo KSZ9031RNX [20] de Microchip, compatible con las velocidades de 10/100/1000 Mbit/s. La comunicación se realiza a través de un puerto RJ45, el cual integra los correspondientes LEDs de estado.
- USB OTG: La placa cuenta con un puerto USB Micro-AB que puede ser utilizado tanto en modo Host como Device (estándar USB 2.0 OTG). El puerto es acompañado por el correspondiente controlador de capa física (PHY), en este caso, un IC USB3300 [21] de Microchip.
- Ranura SD: A fin de implementar almacenamiento permanente, la placa cuenta con una ranura tipo microSD, conectada directamente al controlador SD/MMC del HPS. En este caso, la interfaz de comunicación entre la tarjeta y el HPS utiliza cuatro líneas de datos bidireccionales.
- UART: Uno de los dos periféricos de comunicación UART del HPS es accesible a través de un puerto USB Mini-B. La conversión de protocolo es realizada por un integrado FT232R [22] de FTDI. Cabe destacar que este periférico resulta de importancia, ya que es utilizado por el HPS para comunicarse con otra PC. Por ejemplo, la UART es utilizada para imprimir mensajes de diagnóstico durante el arranque del sistema operativo.
- Acelerómetro (G-Sensor): Consiste en un acelerómetro digital de tres ejes ADXL345 [23] de Analog Devices conectado al HPS a través de I2C. Este IC hace uso del primer controlador I2C integrado en el HPS (I2C0).
- Header LTC: Se trata de un conector de 2x7 pines que brinda acceso a los periféricos de comunicación I2C1 y SPIM0 (SPI Master 0) del HPS. Adicionalmente cuenta con líneas de GND, 3,3 V y 9 V.
- Botones pulsables: Consiste en un botón de propósito general conectado a uno de los puertos GPIO del HPS y dos botones de reset del sistema (cold reset y warm reset).
- LED: Consiste en un LED de color verde, controlable por el usuario a través de uno de los puertos GPIO del HPS.

Capítulo 3

Front-end analógico

El objetivo de este capítulo es presentar el diseño del front-end analógico del sistema de adquisición. Esta etapa tiene dos funcionalidades principales: acondicionar las señales a ser adquiridas y hacer la conversión analógica digital. La implementación del front-end es realizada en un PCB principal de cuatro capas y 12 cm x 8 cm. Adicionalmente, se diseña una fuente de alimentación, en una placa aparte, destinada a proveer las tensiones utilizadas por el front-end. Cabe destacar que la comunicación con el SoC FPGA se realiza a través de un cable cinta de 40 pines (2 x 20).

En la primera parte de este capítulo, se describe de manera general el front-end, es decir, se presenta un diagrama en bloques y esquemático simplificado. Luego, en los apartados 3.4 a 3.6, se profundiza en el proceso de diseño de cada etapa de acondicionamiento, presentando los diversos cálculos teóricos y parámetros involucrados. Por otro lado, se evalúan diferentes alternativas de implementación y se justifica la elección de los diversos componentes del diseño. Posteriormente, los cálculos son validados en base a las simulaciones pertinentes a cada etapa del diseño, realizadas en el software LTSpice.

En las secciones 3.7 a 3.9, se presentan los elementos del diseño que no se encuentran directamente involucrados en el acondicionamiento. Es decir, se describe la interfaz con la FPGA, las etapas de alimentación del PCB principal y el diseño de la fuente. Finalmente, el apartado 3.10 se dedica a la implementación física de ambas placas, o layout de PCB, realizado en el software libre Kicad.

3.1 Diagrama en bloques

En la figura 3.1 se presenta un diagrama con los diferentes bloques funcionales y etapas de acondicionamiento implementadas en el PCB del front-end. Cabe destacar que el diseño para ambos canales de adquisición es, a nivel funcional, idéntico.

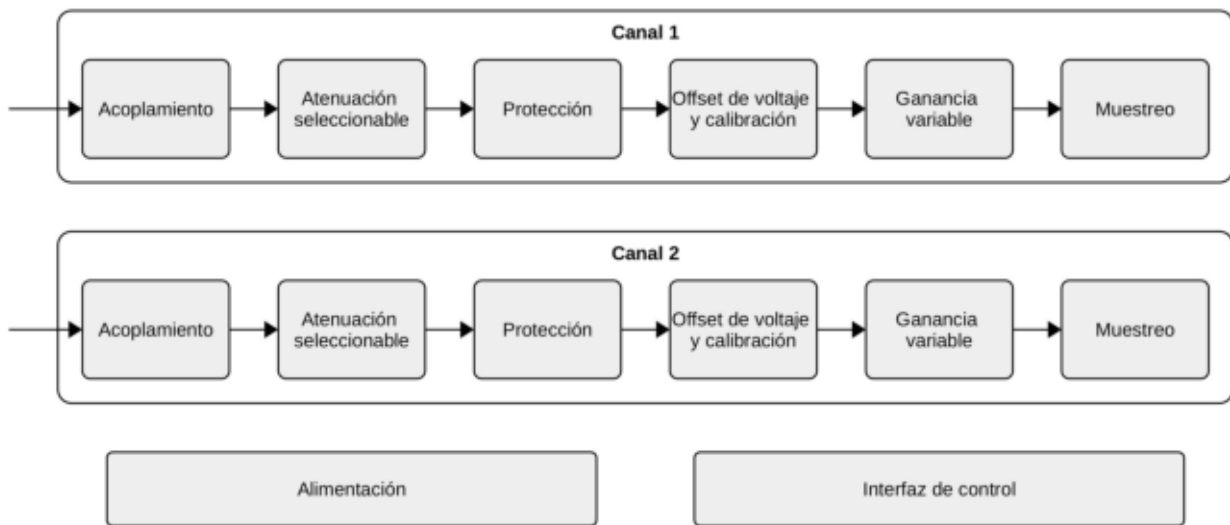


Figura 3.1: Diagrama en bloques del front-end analógico

A continuación, en base a la figura 3.1, se describe brevemente cada etapa analógica:

- **Acoplamiento:** Se brinda la posibilidad de activar un filtro pasa altos que permite acoplar en alterna las señales de entrada, cuya frecuencia de corte es de 5 Hz.
- **Atenuación seleccionable:** Con el objetivo de aumentar el rango de señales posibles a la entrada, se hace uso de dos redes de atenuación, cuyas ganancias son 0,9 y 0,1 veces. La selección del atenuador se lleva a cabo mediante un relé mecánico. El valor de 0,1 se emplea en los rangos de tensión más altos, por otro lado, el valor de 0,9 es usado en las escalas de voltaje más pequeñas.
- **Protección:** Se incorporan circuitos destinados a proteger las etapas posteriores en caso de un nivel de tensión de entrada excesivo. De acuerdo a la especificación de requerimientos, el rango de entrada se establece en ± 25 V, la red de protección es capaz de tolerar entradas en el rango ± 50 V, independientemente de la atenuación seleccionada.
- **Offset de voltaje y calibración:** Cada canal cuenta con un ajuste de offset independiente, que permite sumar a la entrada un nivel de tensión con el objetivo de desplazarla. Esta funcionalidad se implementa mediante el uso de un convertor digital analógico de 12 bits. Adicionalmente, el DAC es utilizado para realizar la calibración de las diferentes escalas verticales.
- **Ganancia variable:** Esta etapa consiste en un potenciómetro digital utilizado como atenuador variable y un amplificador de ganancia programable. La combinación de estos elementos con el atenuador de entrada, permite generar los valores de ganancia requeridos para cada escala vertical.
- **Muestreo:** Cada canal cuenta con un convertor analógico digital de 12 bits funcionando a 1 MSa/s, de acuerdo a las especificaciones de diseño. Se emplea un voltaje de escala completa de 3,3 V, por lo que todas las escalas son llevadas a un rango de $3 V_{PP}$ centrado en 1,65 V.

- Interfaz de control: Consiste en los circuitos necesarios para realizar la configuración del front-end, en base a señales digitales provenientes de la FPGA. Permite seleccionar el atenuador de entrada, activar el acople de alterna, ajustar los valores del DAC, ajustar la constante del potenciómetro y seleccionar la ganancia programable.
- Alimentación: Finalmente, en esta categoría se agrupan los elementos encargados de derivar las tensiones de alimentación necesarias para el funcionamiento del resto de las etapas. Los niveles de tensión principales a utilizar son -5 V, 5 V y 3,3 V. Adicionalmente se genera un nivel de referencia de 5 V para el DAC y 3,3 V para los ADC de ambos canales.

3.2 Ganancias requeridas

El front-end debe ser capaz de llevar el rango de voltaje de entrada de todas las escalas a un valor de $3 V_{PP}$ centrado en 1,65 V. El valor de excursión, de $3 V_{PP}$, permite contar con un margen en ambos extremos del rango de conversión, de 3,3 V. De esta forma, se busca conservar la linealidad del sistema evitando el uso de tensiones cercanas a los rails de alimentación de las diferentes etapas.

Para cada ajuste de escala, en pantalla se visualizan 10 divisiones verticales (en el software de cliente). Adicionalmente, se incorpora un ajuste de posición, que permite desplazar cinco divisiones en ambas direcciones la señal. Por lo tanto, el rango completo de entrada en cada ajuste se corresponde con 20 divisiones en total. En la tabla 3.1, se presentan las ganancias totales requeridas para el front-end.

Escala (V/div)	Cantidad divisiones	V_{IN} (V_{PP})	G_{TOT} (V/V)	V_{OUT} (V_{PP})	Resolución ADC V_{IN}
5	10	50	0,06	3	13,43 mV
2	20	40	0,075	3	10,74 mV
1	20	20	0,15	3	5,371 mV
0,5	20	10	0,3	3	2,685 mV
0,2	20	4	0,75	3	1,074 mV
0,1	20	2	1,5	3	537,1 μ V
0,05	20	1	3	3	268,5 μ V
0,02	20	0,4	7,5	3	107,4 μ V
0,01	20	0,2	15	3	53,71 μ V

Tabla 3.1: Ganancias requeridas para cada escala y resolución en ADC

Como se puede observar en la primera fila de la tabla 3.1, para el caso de la mayor escala (5 V/div), el rango completo se corresponde con 10 divisiones. Esto se debe a que la especificación de diseño del front-end establece un voltaje de entrada máximo de ± 25 V. Por lo tanto, en esta escala no es posible realizar un ajuste de posición, ya que esta posibilidad requiere un rango de entrada de ± 50 V. Adicionalmente, en la última columna de la tabla 3.1, se presenta la resolución del ADC, referida a la tensión de entrada para cada escala. Estos valores se calculan en base a la siguiente relación:

$$RES_{ADC}(V_{IN}) = \frac{V_{REF}}{2^N} \cdot (G_{TOT})^{-1} \quad (3.1)$$

El primer término en la ecuación 3.1 se corresponde con un LSB del ADC. En este caso, teniendo en cuenta la resolución de 12 bits y la tensión de referencia de 3,3 V, dicho valor es aproximadamente de 805,7 μ V. La resolución del ADC es dividida por la ganancia total del front-end, a fin de referirla al voltaje de entrada.

3.3 Diagrama esquemático simplificado

En esta sección se presenta el diagrama esquemático de los circuitos involucrados en la etapa de acondicionamiento analógico. El diagrama consiste en una versión simplificada de un solo canal, donde se omiten todos los elementos relacionados a la interfaz con la FPGA y la alimentación. En la figura 3.2, se puede ver la etapa de acoplamiento y los atenuadores seleccionables de entrada. En la figura 3.3, se observa la protección, el circuito sumador de offset, el potenciómetro digital, y el buffer hacia la siguiente etapa.

En la figura 3.4, se muestra el amplificador de ganancia programable, el desplazamiento de la señal a voltajes positivos y la conversión analógica digital. Finalmente, la etapa encargada de generar el offset de voltaje sumado a la entrada, se presenta en el diagrama de la figura 3.5. Cabe destacar que el amplificador operacional que actúa de buffer para el ADC se alimenta con una tensión unipolar de 3,3 V, el resto de los amplificadores utilizan tensiones de ± 5 V.

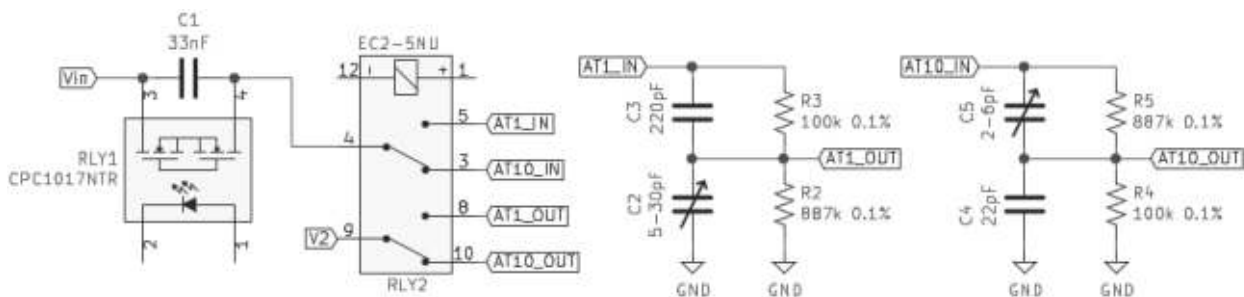


Figura 3.2: Acople de alterna y atenuador seleccionable

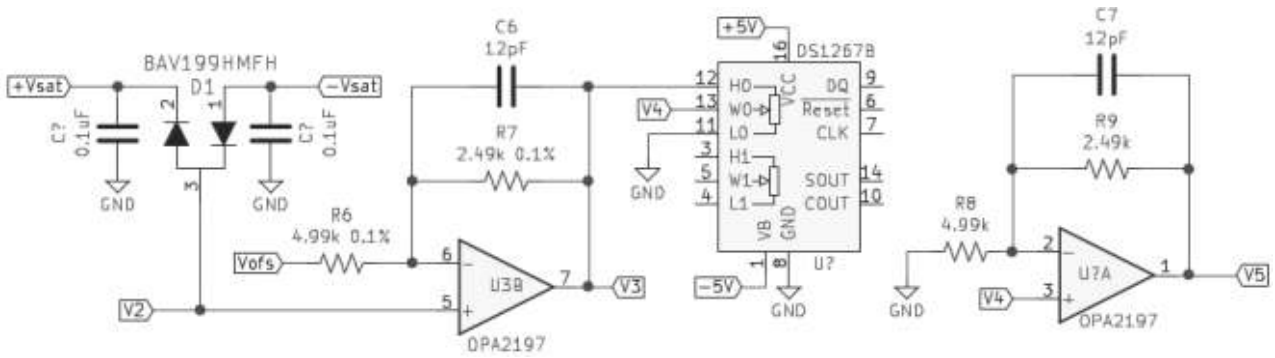


Figura 3.3: Protección, sumador de offset y potenciómetro digital

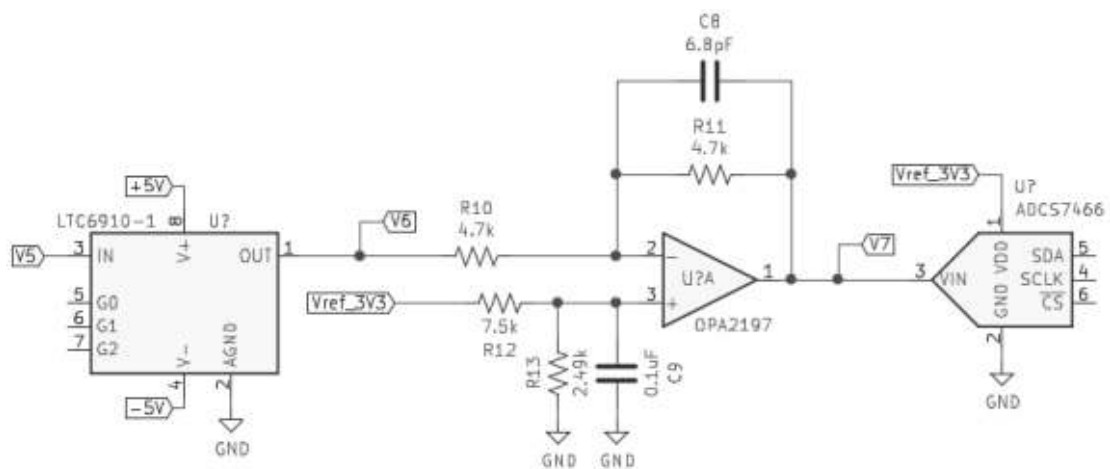


Figura 3.4: Ganancia programable, amplificador operacional de salida y ADC

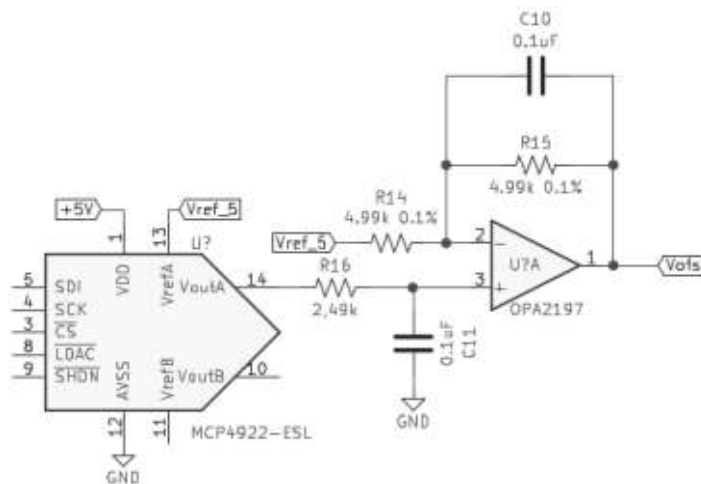


Figura 3.5: Generación del voltaje de offset

3.4 Diseño de etapas analógicas

En las diferentes subsecciones de este apartado se presentan los cálculos involucrados en el diseño de cada una de las etapas. Así mismo, se evalúan las diferentes alternativas de diseño y se justifican las elecciones realizadas. Con el objetivo de facilitar el cálculo de las transferencias parciales de cada etapa, se nombraron diversas tensiones internas (V_2 a V_6) en los diagramas esquemáticos de las figuras 3.2 a 3.5. En la sección 3.5, se hace uso de estas expresiones para calcular la transferencia total del front-end, es decir, desde V_{IN} en la figura 3.2 a V_7 en la figura 3.4.

3.4.1 Elección de amplificadores

El amplificador operacional que se utiliza en las diferentes etapas es el modelo OPA2197 [24] (versión dual) del fabricante Texas Instruments. Algunas de sus características se presentan a continuación:

- Entrada y salida rail-to-rail.
- Voltaje de alimentación bipolar de ± 18 V (máximo).
- Voltaje de offset de ± 100 μ V (máximo).
- Corriente de bias de ± 20 pA (máxima).
- Producto ganancia ancho de banda de 10 MHz.

Los factores determinantes en la elección del amplificador son su bajo offset de voltaje y corriente de bias. Dado que el circuito de acondicionamiento cuenta con varias etapas, los desajustes de cada amplificador operacional se propagan y amplifican. Es deseable, entonces, contar con amplificadores de precisión que permitan minimizar el offset total.

Las entradas y salidas rail-to-rail del amplificador operacional mejoran la calidad de las señales, reduciendo las distorsiones al aproximarse a las tensiones de alimentación y aumentando el rango dinámico. Las características mencionadas lo convierten en un amplificador flexible, lo que permite su utilización en todas las etapas del front-end, a modo de reducir la complejidad del diseño.

Cabe destacar que se consideró la elección de amplificadores de mayor ancho de banda. Estos últimos cuentan con mayores valores de consumo, offset de tensión y offset de corriente. Así mismo, un mayor ancho de banda aumenta el riesgo de inestabilidad, debido al efecto de capacidades parásitas a mayores frecuencias. Un valor de 10 MHz resulta suficiente para lograr la respuesta en frecuencia deseada para todo el sistema.

3.4.2 Acople de alterna

En base al diagrama esquemático de la red de entrada, en la figura 3.2, se puede plantear un circuito equivalente que permita calcular la respuesta en frecuencia del acople de alterna:

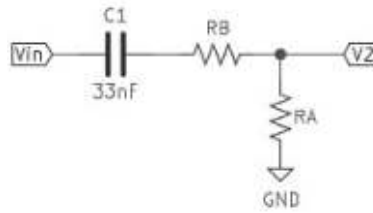


Figura 3.6: Circuito equivalente para el análisis del acople de alterna

El circuito está compuesto por la entrada, el capacitor de acople C_1 y la red de atenuación. Se emplea una denominación genérica para las resistencias del atenuador (R_A y R_B), dado que su valor depende de la red seleccionada. Adicionalmente, los capacitores que forman parte de las redes de atenuación, en la figura 3.2, no son tenidos en cuenta, debido a que actúan a mayor frecuencia. En caso de que se desactive el acople de alterna, el capacitor C_1 es “puenteado” mediante un relé de estado sólido. Resolviendo el circuito de la figura 3.6, se obtiene:

$$\frac{V_2}{V_{IN}}(S) = \frac{R_A \cdot C_1 \cdot S}{1 + (R_A + R_B) \cdot C_1 \cdot S} \quad (3.2)$$

$$A_\infty = \frac{R_A}{R_A + R_B} \quad (3.3)$$

$$f_P = \frac{1}{2 \cdot \pi \cdot (R_A + R_B) \cdot C_1} \quad (3.4)$$

Se trata de una red pasa altos con una ganancia en alta frecuencia de valor A_∞ y una frecuencia de polo f_P . Este último parámetro es calculado a continuación:

$$f_P = \frac{1}{2 \cdot \pi \cdot (987 \text{ k}\Omega) \cdot (33 \text{ nF})} = 4,88 \text{ Hz} \quad (3.5)$$

En el cálculo de la frecuencia de corte se tiene en cuenta que, independientemente del atenuador seleccionado, la suma de las resistencias es la misma (987 k Ω). La frecuencia elegida, de 5 Hz, se asocia con una constante de tiempo de 33 ms, por lo que el sistema tiene una velocidad de respuesta aceptable. Así mismo, esta ubicación del polo permite realizar mediciones de tensión a la frecuencia de línea de 50 Hz, con un error de amplitud menor al 10%.

La activación y desactivación del acople se realiza mediante un relé de estado sólido, en paralelo con el capacitor de acople C_1 . Se hace uso de este tipo de relé debido a que es menos voluminoso y más confiable que su contraparte electromecánica. El dispositivo elegido es el modelo CPC1017NTR [25] de IXYS, cuyas características más importantes se presentan a continuación:

- Switch SPST (normalmente abierto) basado en MOSFETs.
- Corriente de LED de 1 mA (típica) y 50 mA (máxima), caída de 1,2 V (típica).

- Tiempo de conmutación de 10 ms (máximo) bajo una corriente de LED de 5 mA.
- Aislación de entrada a salida de 1500 V_{RMS}.
- Voltaje de bloqueo de 60 V (en ambas direcciones).
- Corriente continua de 100 mA (máxima) y pulsada de 350 mA (máxima t=10 ms).
- Resistencia de switch de 16 Ω (máxima) bajo corriente de 100 mA.
- Capacidad de salida de 25 pF (típica) a una frecuencia de 1 MHz.

En la elección se considera principalmente el voltaje de bloqueo, debido a que debe ser mayor al rango de entrada del front-end, de ± 50 V. La corriente máxima del switch es varios órdenes de magnitud superior a la esperada en el diseño, considerando la impedancia de entrada de 1 MΩ. Con respecto a los elementos parásitos del switch, la capacidad de 25 pF queda efectivamente en paralelo con C₁ (33 nF), por lo que su valor es despreciable. Por otro lado, la resistencia de encendido de 16 Ω, no modifica considerablemente la transferencia, debido a su pequeño valor comparado con las resistencias del atenuador.

La inherente aislación de este dispositivo permite minimizar los circuitos de control necesarios para la activación del relé. En este caso, se conecta directamente el dispositivo a un pin de salida del área de lógica programable del SoC FPGA, a través de una resistencia de limitación de corriente de 499 Ω. Este valor permite obtener una corriente de 4 mA en el LED interno del relé, asegurando la correcta conmutación.

3.4.3 Atenuador seleccionable

Para el caso de las escalas de voltaje más grandes, resulta necesario contar con un mayor grado de atenuación. La solución implementada hace uso de dos redes de atenuación diferentes, cuya selección se realiza mediante un relé mecánico. La resistencia serie total para ambos atenuadores debe ser aproximadamente 1 MΩ. Esta condición fija la impedancia de entrada del sistema y la frecuencia de corte del acople de alterna. Teniendo en cuenta los requisitos de diseño, inicialmente se consideraron las siguientes dos redes:



Figura 3.7: Redes de atenuación de entrada propuestas inicialmente

Al realizar las simulaciones correspondientes, se obtuvo un comportamiento pasa bajos con una frecuencia de corte aproximada de 120 kHz. Esta respuesta se debe a la presencia de un polo, proveniente de la interacción de las resistencias del atenuador con las capacidades parásitas de los diodos de protección y el primer amplificador operacional. La situación se puede observar en el siguiente circuito equivalente:

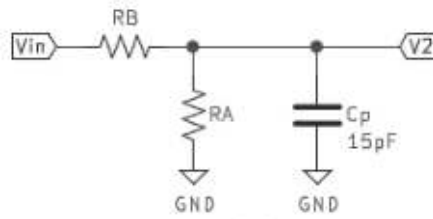


Figura 3.8: Efecto de la capacidad parásita en el atenuador de entrada

A continuación, se calcula la transferencia de la red de la figura 3.8:

$$\frac{V_2}{V_{IN}}(S) = \frac{R_A}{R_A + R_B} \cdot \frac{1}{1 + (R_A // R_B) \cdot C_P \cdot S} \quad (3.6)$$

$$A_0 = \frac{R_A}{R_A + R_B} \quad (3.7)$$

$$f_P = \frac{1}{2 \cdot \pi \cdot (R_A // R_B) \cdot C_P} \quad (3.8)$$

Es decir, se trata de una red pasa bajos con ganancia en continua A_0 y frecuencia de corte f_P . Teniendo en cuenta los valores de las resistencias del atenuador (887 k Ω y 100 k Ω) y la capacidad parásita total de 15 pF [24], [26], se procede a calcular la frecuencia de corte:

$$f_P = \frac{1}{2 \cdot \pi \cdot (90 \text{ k}\Omega) \cdot (15 \text{ pF})} = 118 \text{ kHz} \quad (3.9)$$

El valor obtenido no es aceptable para esta aplicación, por lo que, en el diseño final, se decide utilizar un atenuador compensado, cuya estructura genérica se analiza a continuación.

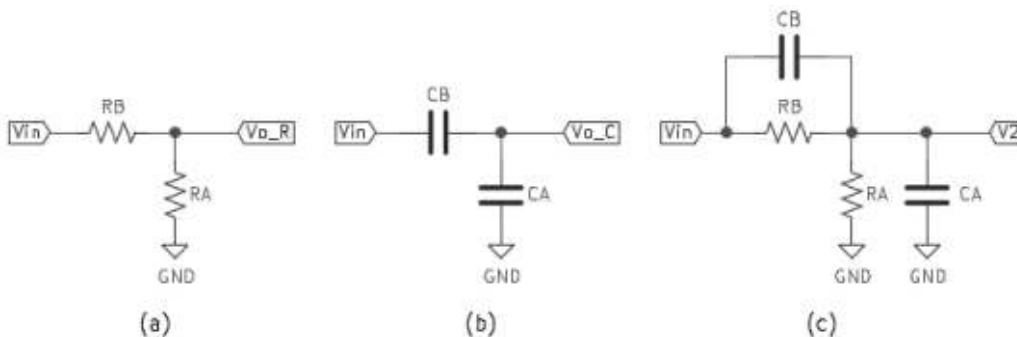


Figura 3.9: (a) Divisor resistivo (b) Divisor capacitivo (c) Atenuador compensado

Si la transferencia de los divisores de las figuras 3.9(a) y 3.9(b) es idéntica, ambas redes pueden unirse a su salida en la topología de la figura 3.9(c). En ese caso, se logra el valor de atenuación deseado para toda frecuencia, eliminando la presencia del polo. A continuación, se deduce la condición necesaria de acuerdo a lo mencionado anteriormente:

$$\frac{V_{OR}}{V_{IN}}(S) = \frac{R_A}{R_A + R_B} \quad (3.10)$$

$$\frac{V_{OC}}{V_{IN}}(S) = \frac{C_B}{C_A + C_B} \quad (3.11)$$

Igualando las ecuaciones 3.10 y 3.11 se obtiene:

$$\frac{R_A}{R_B} = \frac{C_B}{C_A} \quad (3.12)$$

Haciendo uso de la propiedad demostrada en la ecuación 3.12, se decide utilizar capacitores variables o trimmers, para lograr la relación necesaria, absorbiendo la capacidad parásita. En la figura 3.10, se presenta un circuito equivalente de los dos atenuadores en la versión final del diseño. Adicionalmente, con el objetivo de evaluar el rango de compensación, se agrega la capacidad parásita C_p .

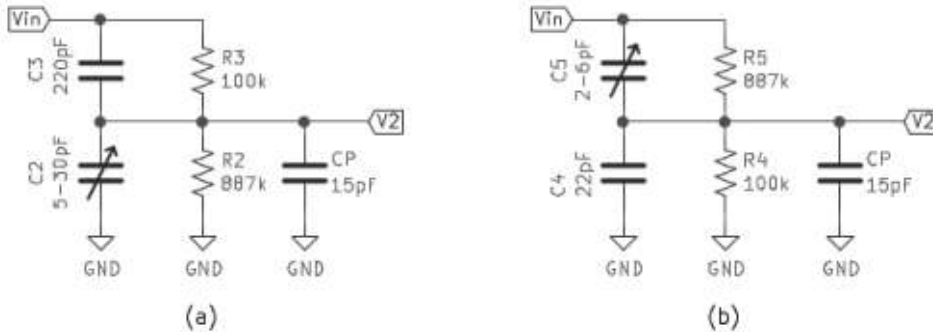


Figura 3.10: (a) Atenuador de 0,9 (b) Atenuador de 0,1

Para el caso del atenuador de 0,9 veces, de la figura 3.10(a), se debe cumplir que:

$$\frac{C_3}{C_2 + C_p} = \frac{R_2}{R_3} = 9 \quad (3.13)$$

Por lo tanto, con un valor de C_3 de 220 pF y un trimmer en el rango 5 pF a 30 pF para C_2 , se pueden compensar las capacidades parásitas en el rango de 0 pF a 19,4 pF. Por otro lado, para el caso del atenuador de 0,1 veces, de la figura 3.10(b), se debe cumplir que:

$$\frac{C_4 + C_P}{C_5} = \frac{R_5}{R_4} = 9 \quad (3.14)$$

Por lo tanto, con un valor de 22 pF para C_4 y un trimmer de 2 pF a 6 pF para C_5 , se pueden compensar las capacidades parásitas en el rango 0 pF a 32 pF.

Finalmente, con respecto a la conmutación entre ambos atenuadores, se hace uso de un relé electromecánico del tipo DPDT (Double Pole Double Throw). La conexión de este dispositivo se realiza de manera que se encuentre seleccionado el atenuador de 10 veces por defecto con el relé desenergizado. De esta forma, se asegura cierto nivel de protección de las etapas siguientes del front-end, en caso que se ingrese una señal sin estar el sistema alimentado. El relé que se utiliza es el modelo EC2-5NU [27] del fabricante Kemet, cuyas características más importantes se presentan a continuación:

- Voltaje de conmutación de 220 V_{DC} o 250 V_{AC} (máximo).
- Corriente de conmutación de 2 A (máxima).
- Resistencia de contacto de 75 mΩ (máxima).
- Expectativa de vida de 100 millones de conmutaciones (sin carga).
- Bobina de 5 V (nominal), 178 Ω, consumo nominal de 140 mW.

Como se puede observar, el dispositivo cumple con los requerimientos necesarios para la aplicación. La conmutación del relé es llevada a cabo mediante un MOSFET canal N modelo NTR3C21NZT1G [28] del fabricante ON Semiconductor. Este transistor cuenta con una tensión de ruptura de 20 V (drain a source) y una corriente máxima de 3,6 A. El dispositivo se elige debido a que asegura una resistencia de 19 mΩ con una tensión de compuerta de 3,3 V, valor de nivel lógico proveniente de la FPGA. Por otro lado, como medida de protección, se incorpora en paralelo a la bobina del relé un diodo 1N4148W-7-F [29] del fabricante Diodes Incorporated. Se trata de un diodo de silicio de rápida conmutación, con una tensión en directa de 0,71 V bajo una corriente de 1 mA, y una corriente máxima de 300 mA.

3.4.4 Protección

Dado que la tensión de alimentación del primer amplificador operacional se encuentra en el rango ± 5 V, se debe proteger la entrada no inversora del mismo contra tensiones excesivas a la entrada del sistema de adquisición. Se implementa una protección en base a dos diodos conectados a tensiones de -3,3 V y 3,3 V respectivamente.

En un principio se consideró usar diodos Schottky, ya que su bajo voltaje en directa permite conectarlos directamente a la misma alimentación que los amplificadores (± 5 V). Sin embargo, este tipo de dispositivo cuenta con una corriente de fuga inversa considerablemente mayor a los diodos comunes. Teniendo en cuenta que el atenuador de entrada emplea resistencias de valores elevados, la corriente de fuga de los diodos Schottky generaría una distorsión considerable en la señal de entrada [30].

En base al análisis realizado, se decide utilizar diodos dobles de silicio de baja corriente inversa BAV199HMFH [26] de ROHM Semiconductor. Las características más importantes del dispositivo se describen a continuación:

- Corriente de fuga de 5 nA (máxima) bajo un voltaje en inversa de 75 V.
- Tensión de directa de 0,9 V (típica) bajo una corriente de 1 mA.
- Corriente máxima en directa de 215 mA.
- Tensión en inversa máxima de 100 V.

El nivel de tensión máximo posible en la entrada no inversora del amplificador operacional es aproximadamente de 4,2 V, debido a la suma de la tensión en directa del diodo (0,9 V) y la tensión de saturación (3,3 V). Este valor se encuentra en el margen de las tensiones que soporta el amplificador operacional, de entradas rail-to-rail.

A fin de obtener la corriente máxima en el diodo, se plantea el caso en que se ingresa con una tensión de 50 V, es decir, la máxima tensión de protección establecida en la especificación. La corriente en el diodo es mayor cuando se selecciona el valor de atenuación de 0,9. Esta situación se observa a continuación:

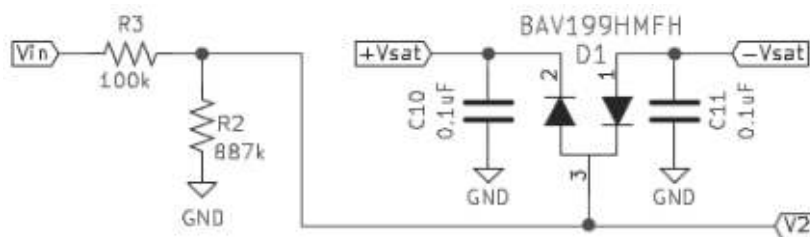


Figura 3.11: Circuito equivalente para el análisis de la protección

A partir de la figura 3.11 y realizando un equivalente de Thévenin de la entrada y el atenuador, se calcula la corriente en el diodo:

$$V_{TH} = \frac{R_2}{R_2 + R_3} \cdot V_{IN} = 45 \text{ V} \quad (3.15)$$

$$R_{TH} = R_1 // R_2 = 90 \text{ k}\Omega \quad (3.16)$$

$$I_D(\text{max}) = \frac{V_{TH} - V_{SAT} - V_D}{R_{TH}} = \frac{45 \text{ V} - 0.9 \text{ V} - 3.3 \text{ V}}{90 \text{ k}\Omega} = 0,45 \text{ mA} \quad (3.17)$$

Producto de los elevados valores de resistencia en el atenuador de entrada, el valor de corriente máximo, en la ecuación 3.17, es pequeño y se encuentra dentro de las especificaciones del diodo. Por otro lado, los capacitores C₁₀ y C₁₁ de la figura 3.11 tienen como objetivo absorber cualquier transitorio de voltaje de alta frecuencia hasta que actúe la protección.

3.4.5 Sumador de offset

El primer amplificador operacional, en la etapa de acondicionamiento, permite sumar un voltaje a la señal de entrada. Adicionalmente, actúa de buffer para la siguiente etapa y establece un límite de ancho de banda para la señal de entrada. El diseño propuesto puede ser observado en el diagrama esquemático de la figura 3.12.

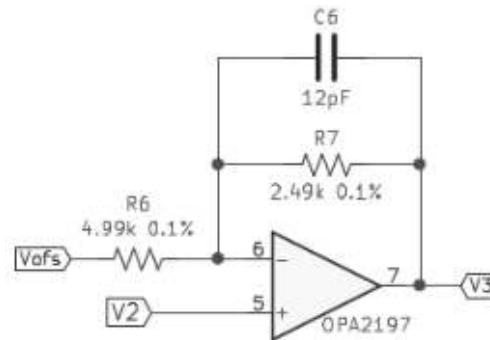


Figura 3.12: Circuito sumador de offset (buffer de entrada)

A continuación, se calcula la transferencia del circuito suponiendo el caso de un amplificador operacional ideal (y sin considerar C_6). El análisis de estabilidad y respuesta en frecuencia se realiza en la segunda parte de esta sección.

$$\frac{V_3}{V_2}(S) = 1 + \frac{R_7}{R_6} = 1,5 \quad (3.18)$$

$$\frac{V_3}{V_{OFS}}(S) = -\frac{R_7}{R_6} = -0,5 \quad (3.19)$$

A partir de las ecuaciones 3.18 y 3.19, se puede apreciar que el circuito de la figura 3.12 amplifica 1,5 veces a la señal de entrada y -0,5 veces al voltaje de offset. Estos valores de ganancias permiten obtener la transferencia total requerida para cada escala y un rango apropiado de offset (referido a la señal de entrada). Una vez obtenida la transferencia deseada a bajas frecuencias, se lleva a cabo el análisis de estabilidad y respuesta en frecuencia. En base al modelo de simulación del amplificador brindado por el fabricante [31], se obtiene en LTSpice la función de transferencia a lazo abierto mostrada en la figura 3.13.

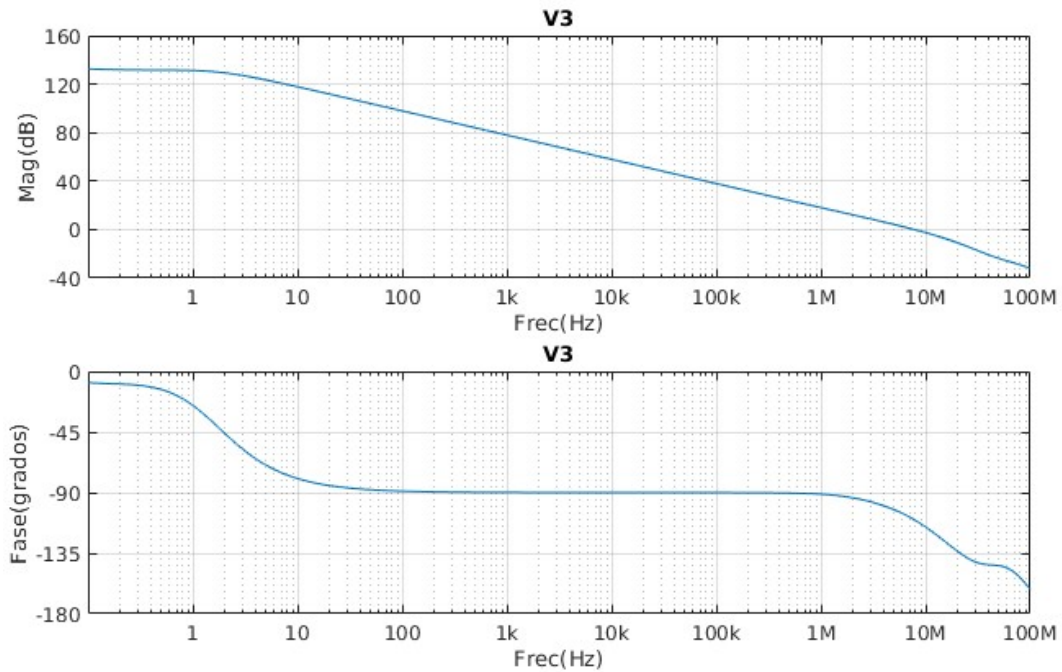


Figura 3.13: Respuesta en frecuencia OPA2197 a lazo abierto

Se puede observar que la respuesta está compuesta por una ganancia en continua de 135 dB y un polo dominante en baja frecuencia, el cual genera una caída constante (y una fase de -90°). El cruce por 0 dB ocurre a 10,4 MHz con un margen de fase de 70° , hecho que se corresponde con el producto ganancia ancho de banda especificado por el fabricante. Con el objetivo de verificar el modelo de simulación, se analiza la transferencia a lazo abierto de la hoja de datos del amplificador [24], mostrada a continuación:

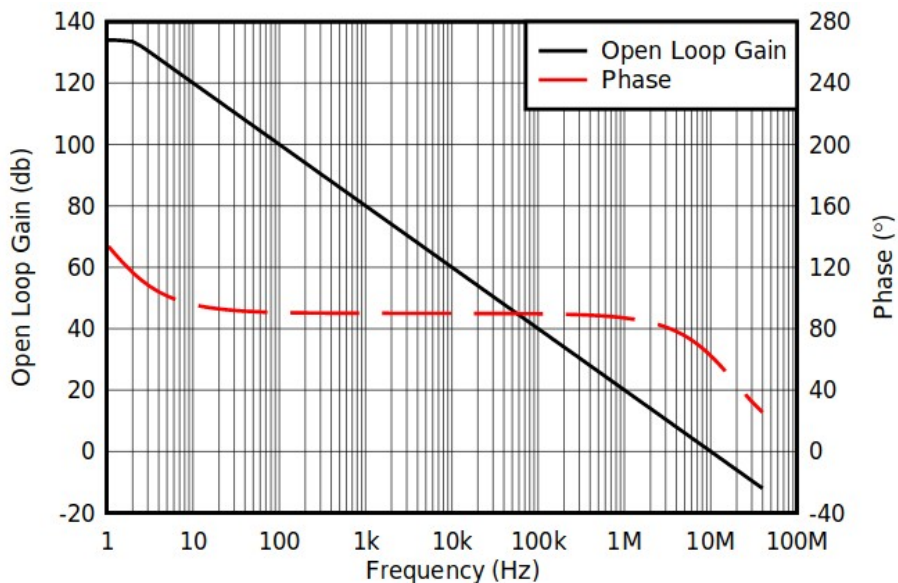


Figura 3.14: Transferencia a lazo abierto OPA2197 con $C_L = 15 \text{ pF}$

Se observa que la función transferencia de amplitud a lazo abierto obtenida por simulación, en la figura 3.13, coincide en gran medida con la proporcionada por el fabricante, en la figura 3.14. Sin embargo, en el segundo caso, el margen de fase aproximado es de 50° , es decir, 20° menor que el obtenido por simulación. Esto se debe a que dicha transferencia se especifica para una capacidad de carga de 15 pF . La impedancia de salida del amplificador, de $375\ \Omega$, produce un polo con el capacitor de carga a una frecuencia de $28,3\text{ MHz}$, agregando 20° en el cruce por 0 dB , el que no se modifica significativamente.

Del análisis realizado, se concluye que el amplificador es estable a ganancia unitaria, mejorando el margen de fase para valores mayores de ganancia. Teniendo en cuenta la ganancia a lazo cerrado deseada, inicialmente se propuso, para la etapa de la figura 3.12, una red de realimentación con un valor de R_7 de $10\text{ k}\Omega$ y R_6 de $20\text{ k}\Omega$. Al realizar las simulaciones, se obtuvo la respuesta mostrada en la figura 3.15, ante una entrada de tipo escalón unitario.

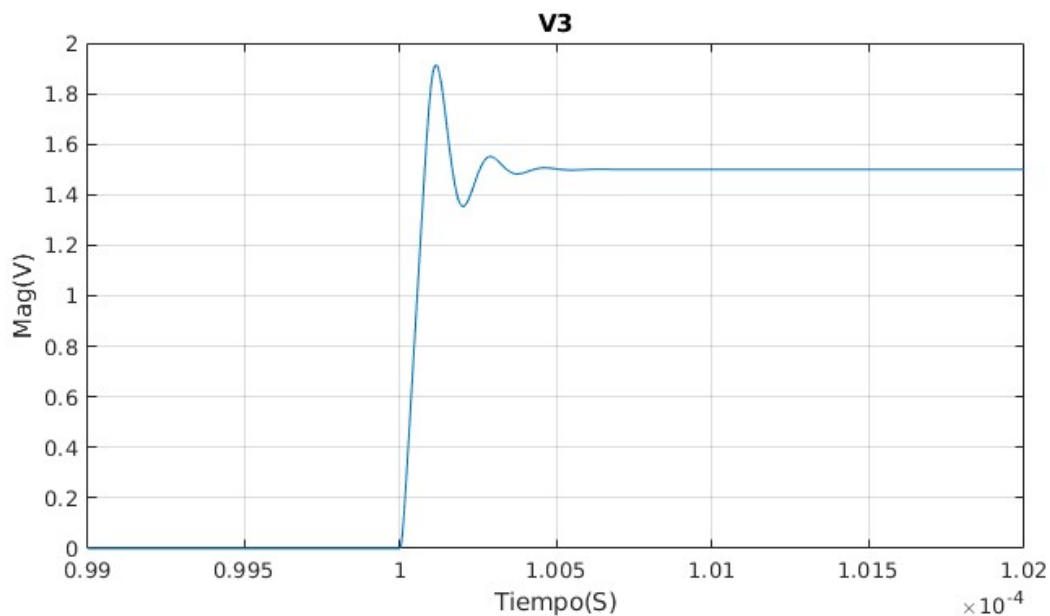


Figura 3.15: Respuesta al escalón inicial

Dado que el sobrepico observado no se corresponde con el margen de fase teórico esperado, se realizó el Bode en frecuencia de la ganancia de lazo, obteniendo un margen de fase de 30° a $3,6\text{ MHz}$. La diferencia con los valores esperados se debe a la presencia de capacidad parásita en la entrada inversora del amplificador operacional. La combinación de la capacidad parásita con las resistencias de realimentación produce un polo que agrega desfase, deteriorando la estabilidad y ancho de banda del amplificador.

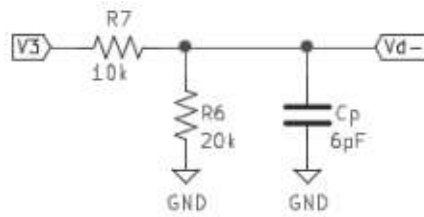


Figura 3.16: Efecto de la capacidad de entrada en la primera red propuesta

En la figura 3.16 se observa el circuito equivalente de la red de realimentación propuesta inicialmente, incorporando la presencia de la capacidad parásita. La función transferencia de la realimentación es la siguiente:

$$H(S) = \frac{R_6}{R_6 + R_7} \cdot \frac{1}{1 + (R_6 // R_7) \cdot C_p \cdot S} \quad (3.20)$$

$$f_p = \frac{1}{2 \cdot \pi \cdot (R_6 // R_7) \cdot C_p} \quad (3.21)$$

A continuación, se calcula la frecuencia del polo f_p , teniendo en cuenta que la capacidad parásita de entrada es aproximadamente 6 pF [24]:

$$f_p = \frac{1}{2 \cdot \pi \cdot (6,66 \text{ k}\Omega) \cdot (6 \text{ pF})} = 4 \text{ MHz} \quad (3.22)$$

La presencia del polo en 4 MHz produce un desfase de 42° en el cruce por 0 dB de la ganancia de lazo, que ocurre a una frecuencia aproximada de 3,6 MHz, reduciendo la estabilidad del sistema. Con el objetivo de mejorar la respuesta, en el diseño final se reducen los valores de las resistencias a 5 k Ω y 2,5 k Ω , aumentando la frecuencia del polo a 16 MHz. Adicionalmente, se agrega un capacitor en paralelo a R_7 como se puede observar en la figura 3.17.

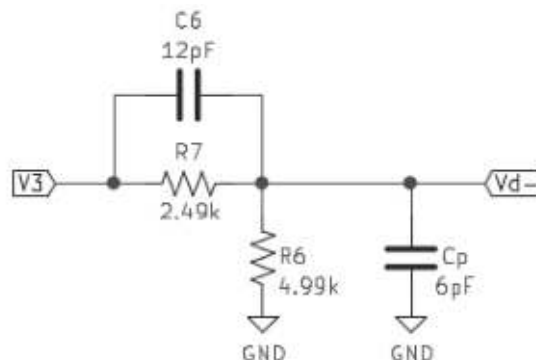


Figura 3.17: Red de realimentación mejorada

En el análisis de la etapa del atenuador compensado se demostró que, idealmente, se puede eliminar completamente la presencia del polo asociado a la capacidad parásita. En un caso real, la cancelación no es completa, sin embargo, se puede reducir el aporte de fase y mejorar la respuesta del circuito. En base a lo mencionado, se agrega el capacitor de compensación C_6 , cuyo valor se calcula a continuación:

$$\frac{R_7}{R_6} = \frac{C_p}{C_6} \quad (3.23)$$

$$C_6 = 2 \cdot C_p = 12 \text{ pF} \quad (3.24)$$

En base a la nueva red de realimentación, de la figura 3.17, se obtiene la siguiente respuesta en el tiempo y en frecuencia:

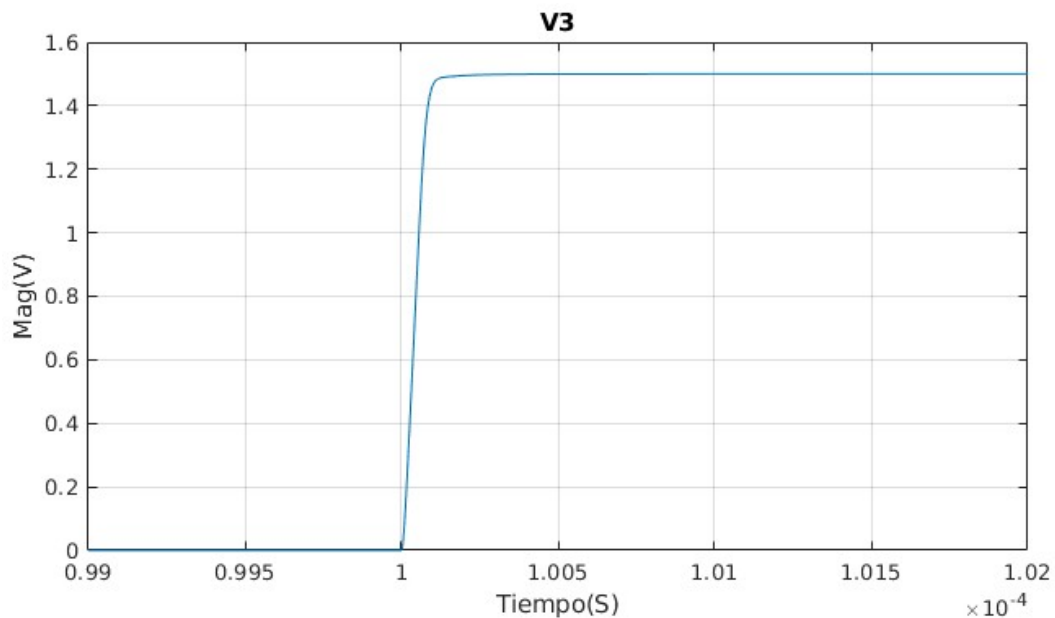


Figura 3.18: Respuesta al escalón con la segunda red de realimentación propuesta

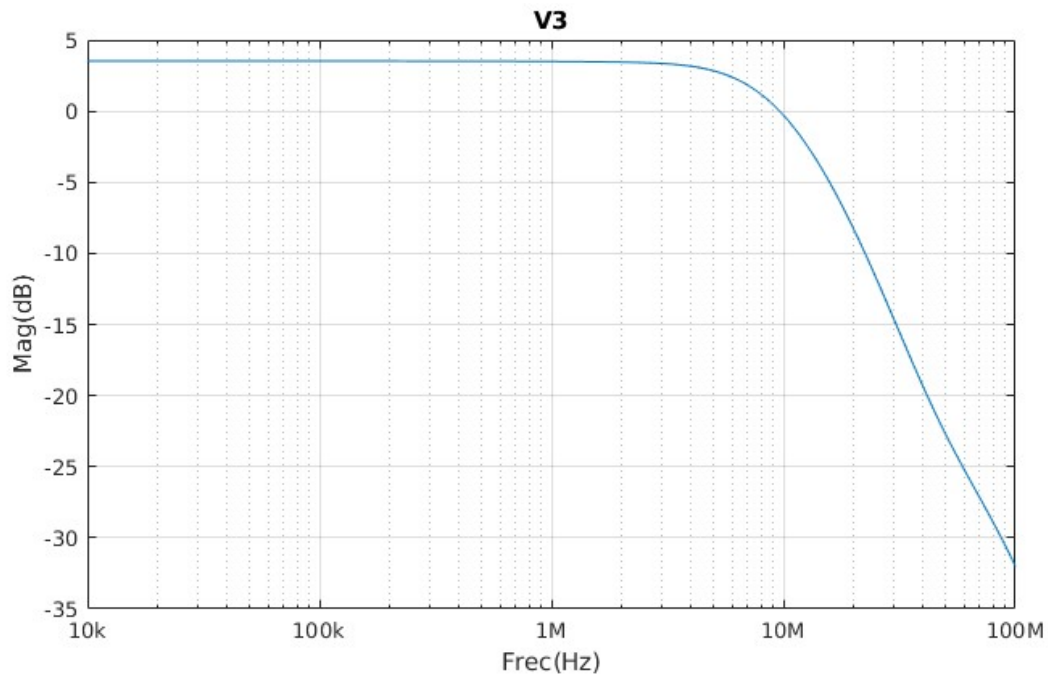


Figura 3.19: Respuesta en frecuencia final con la segunda red de realimentación propuesta

El ancho de banda final, obtenido en la simulación, es de 8,6 MHz, levemente mayor al valor teórico esperado, calculado a continuación en base al producto ganancia ancho de banda del amplificador operacional:

$$B(3\text{ dB}) = \frac{GBW}{H(0)^{-1}} = \frac{10\text{ MHz}}{1,5} = 6,66\text{ MHz} \quad (3.25)$$

3.4.6 Atenuador programable

La siguiente etapa, en el acondicionamiento de la señal, consiste en un potenciómetro digital utilizado como atenuador programable de 256 pasos. El potenciómetro es seguido por un amplificador operacional, que actúa como buffer para el amplificador de ganancia programable (siguiente etapa). Esto es necesario debido a que la impedancia de entrada del PGA no es alta y depende de la ganancia seleccionada.

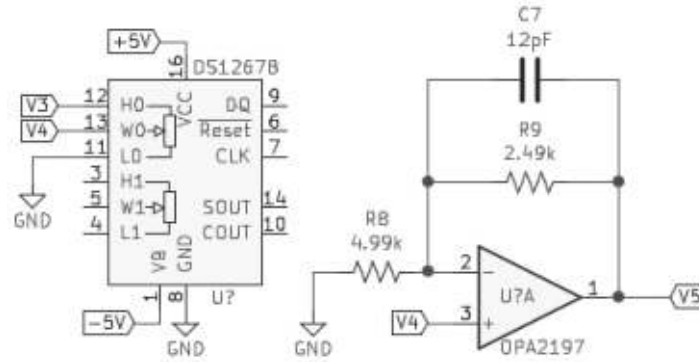


Figura 3.20: Potenciómetro digital y buffer de PGA

En la figura 3.20 se observa el diagrama esquemático simplificado de esta etapa, en el cual se omiten las conexiones del segundo canal, capacitores de desacople y elementos relacionados a la comunicación con la FPGA. En base al análisis del circuito, se obtienen las siguientes funciones de transferencia:

$$\frac{V_4}{V_3}(S) = \frac{R_{p2}}{R_{p1} + R_{p2}} = K_p \quad (3.26)$$

$$\frac{V_5}{V_4}(S) = \left(1 + \frac{R_9}{R_8}\right) = 1,5 \quad (3.27)$$

Ambas expresiones son válidas únicamente en baja frecuencia, es decir, no se considera el capacitor de compensación C_7 y el efecto de las singularidades del amplificador.

El potenciómetro digital utilizado es el modelo DS1267BS [32] de Maxim Integrated. Se trata de un potenciómetro dual de 8 bits, compartido por ambos canales del front-end. Las características principales se listan a continuación:

- Programación independiente de cada potenciómetro (interfaz SPI).
- Voltaje de alimentación bipolar de ± 5 V (máximo).
- Resistencia total de $10 \text{ k}\Omega \pm 20\%$.
- Corriente de salida de 1 mA (máxima).
- Capacidad de salida de 7 pF (típica).

De acuerdo a las especificaciones presentadas, la tolerancia en la resistencia total es bastante baja (20%), sin embargo, dado que el potenciómetro es utilizado como divisor resistivo, este valor no es de importancia. El hecho de que la corriente de salida sea limitada tampoco afecta en este caso, debido a la alta impedancia del amplificador operacional. Con el objetivo de obtener el ancho de banda del atenuador, se plantea el siguiente circuito equivalente [33]:

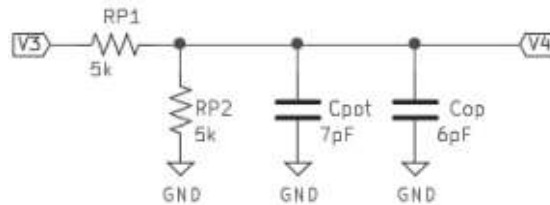


Figura 3.21: Circuito equivalente de potenciómetro digital (50%)

En el diagrama esquemático de la figura 3.21, se puede observar el potenciómetro y su capacidad de salida (C_{POT}) así como también la capacidad de entrada del amplificador operacional de la siguiente etapa (C_{OP}). En base a este circuito equivalente, se obtiene la siguiente función transferencia:

$$\frac{V_4}{V_3}(S) = \frac{R_{P2}}{R_{P1} + R_{P2}} \cdot \frac{1}{1 + (R_{P1} // R_{P2}) \cdot (C_{POT} + C_{OP}) \cdot S} \quad (3.28)$$

$$f_P = \frac{1}{2 \cdot \pi \cdot (R_{P1} // R_{P2}) \cdot (C_{POT} + C_{OP})} \quad (3.29)$$

El caso de menor frecuencia de corte se da para un ajuste del potenciómetro de 50%, en el que el paralelo de las resistencias R_{P1} y R_{P2} tiene su valor máximo (2,5 k Ω). En base a este valor y a la capacidad parásita total de 13 pF [24], [32], se obtiene un ancho de banda de 4,9 MHz, suficiente para esta aplicación.

Finalmente, con respecto al amplificador operacional, se agrega un capacitor de compensación C_7 de 12 pF en la realimentación. El análisis de estabilidad y respuesta en frecuencia es idéntico al realizado en la sección 3.4.5. Teniendo en cuenta lo mencionado, el ancho de banda a lazo cerrado del amplificador, obtenido por simulación, es de 8,6 MHz.

3.4.7 Ganancia programable

La siguiente etapa en la cadena de la señal consiste en un amplificador de ganancia programable LTC6910-1 [34] de Analog Devices. Este dispositivo, permite elegir entre ocho ganancias diferentes, en base al valor de tres líneas digitales de entrada. A continuación, se presentan algunas de sus características:

- Valores de ganancia de 0, 1, 2, 5, 10, 20, 50 y 100 veces.
- Voltaje de alimentación bipolar de ± 5 V (máximo).
- Entrada y salida rail-to-rail.
- Producto ganancia ancho de banda de 11 MHz.
- Voltaje de offset de 9 mV (máximo).

En las figuras 3.22(a) y 3.22(b), se puede observar la respuesta en frecuencia del amplificador y su diagrama en bloques, respectivamente. Cabe destacar que ambas figuras fueron obtenidas de la hoja de datos del fabricante [34].

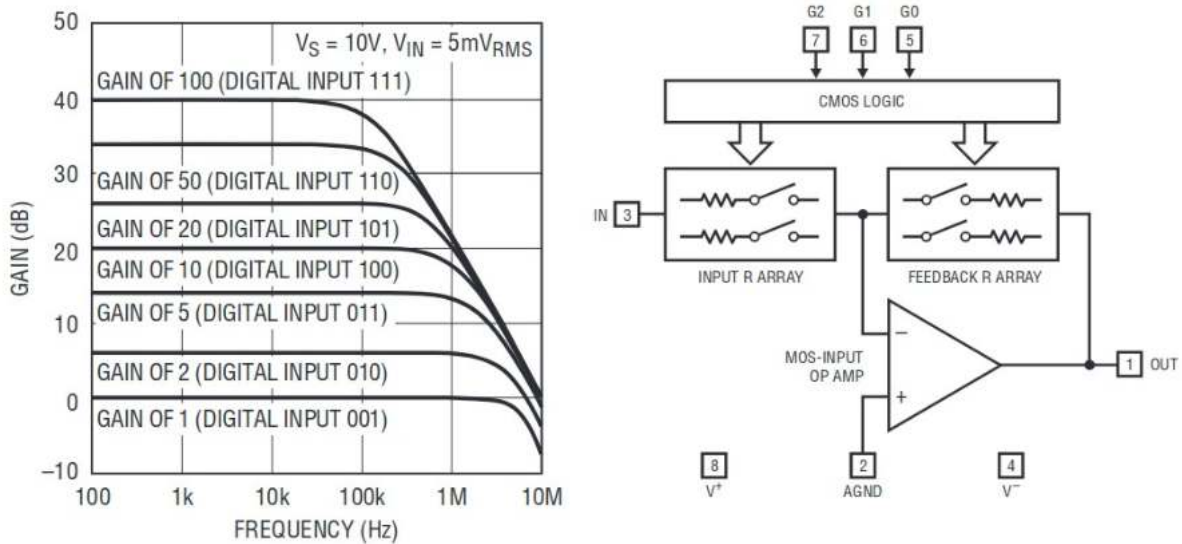


Figura 3.22: (a) Respuesta en frecuencia del PGA (b) Diagrama en bloques simplificado

El PGA está implementado en base a un amplificador operacional en modo inversor, cuya red de realimentación se ajusta de acuerdo a la ganancia. Debido a la topología, la impedancia de entrada al PGA es baja y depende de la ganancia seleccionada. Teniendo en cuenta lo mencionado, la transferencia en baja frecuencia para esta etapa se puede expresar de la siguiente manera:

$$\frac{V_6}{V_5}(S) = -G_{PGA} \quad (3.30)$$

G_{PGA} (V/V)	A2,A1,A0	Z_{IN} (k Ω)	B (3 dB) (MHz)
0	000	Inf	-
-1	001	10	11
-2	010	5	5,5
-5	011	2	2,2
-10	100	1	1,1
-20	101	1	0,5
-50	110	1	0,22
-100	111	1	0,11

Tabla 3.2: Parámetros del PGA para cada ajuste de ganancia

En la tabla 3.2 se listan los valores de ancho de banda para cada una de las ganancias del PGA, de acuerdo a la hoja de datos. En el diseño se utilizan ganancias de 1, 2, 5 y 10 veces, por lo que el ancho de banda se ubica en el rango de 1,1 MHz a 1,1 MHz. Los valores de 20, 50 y 100 veces brindan la posibilidad de aumentar la sensibilidad del sistema de adquisición a futuro, implementando escalas de menor amplitud.

3.4.8 Buffer ADC

Esta etapa consiste en un amplificador operacional, cuya funcionalidad principal es sumar un offset de 1,65 V a la señal de entrada. Dado que las etapas anteriores ajustan la ganancia de cada escala, logrando una amplitud de 3 V_{PP}, la salida del buffer se encuentra en el rango 0,15 V a 3,15 V. La topología implementada se observa en la figura 3.23, presentada a continuación:

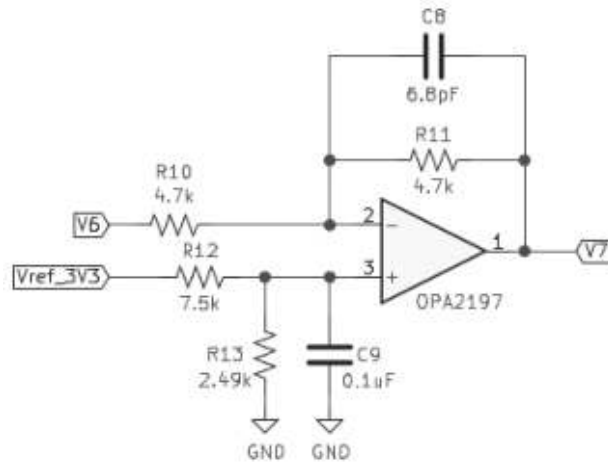


Figura 3.23: Buffer ADC (traslado de nivel)

El amplificador se alimenta con un voltaje unipolar de 3,3 V, por lo que la salida de esta etapa se encuentra en el rango del ADC. De esta manera, se evita la necesidad de recurrir a protecciones adicionales en la entrada del convertor [35]. Al igual que en secciones anteriores, se obtiene la transferencia en baja frecuencia para el circuito de la figura 3.23, considerando el amplificador como ideal:

$$\frac{V_7}{V_6}(S) = -\frac{R_{11}}{R_{10}} = -1 \quad (3.31)$$

En la transferencia de la ecuación 3.31, se puede observar que se trata de una etapa inversora de ganancia unitaria. Idealmente, la señal de entrada se encuentra en el rango $\pm 1,5$ V, es decir, 3 V_{PP}. Sin embargo, una elección de una escala pequeña con una señal de entrada no adecuada, hace que las etapas anteriores saturen, obteniendo una señal en el rango ± 5 V (a la entrada del amplificador operacional). Dado que la señal ingresa por la entrada inversora, existe inherente protección del amplificador operacional, por lo que no se requieren circuitos adicionales.

Con respecto al offset de 1,65 V, debido a que este nivel es sumado directamente a la entrada, debe ser preciso y no contener ruido o variaciones significativas. Teniendo en cuenta lo mencionado, el valor de 1,65 V se obtiene en base a la misma referencia de tensión externa de 3,3 V utilizada por los ADC. A continuación, se calcula la transferencia del circuito de la figura 3.23, para el caso del voltaje de referencia:

$$\frac{V_7}{V_{REF}}(S) = \frac{R_{13}}{R_{13} + R_{12}} \cdot \frac{1}{1 + (R_{12} // R_{13}) \cdot C_9 \cdot S} \cdot \left(1 + \frac{R_{11}}{R_{10}}\right) \quad (3.32)$$

$$f_P = \frac{1}{2 \cdot \pi \cdot (R_{12} // R_{13}) \cdot C_9} = 850 \text{ Hz} \quad (3.33)$$

$$A_0 = \frac{R_{13}}{R_{13} + R_{12}} \cdot \left(1 + \frac{R_{11}}{R_{10}}\right) = 0,5 \quad (3.34)$$

En la ecuación 3.34, se puede verificar que el amplificador operacional brinda la ganancia requerida para obtener 1,65 V a partir de la referencia de 3,3 V, es decir, 0,5 veces. Adicionalmente, se incorpora una red de filtrado pasa bajos con una frecuencia de corte de 850 Hz, de acuerdo al cálculo de la ecuación 3.33. El objetivo del filtrado es eliminar cualquier nivel de ruido o transitorio presente en la referencia de 3,3 V.

Finalmente, para el capacitor de compensación C_8 del circuito de la figura 3.23, se elige un valor de 6,8 pF destinado a extender la respuesta en frecuencia y mejorar la estabilidad del amplificador.

3.4.9 ADC

Con respecto a la etapa de muestreo, se emplean conversores analógico digital ADCS7476 [36] de Texas Instruments (uno por canal). Las características principales se presentan a continuación:

- Arquitectura tipo SAR (aproximaciones sucesivas) de 12 bits con muestreo y retención interno.
- DNL de $\pm 0,5$ LSB, INL de $\pm 0,4$ LSB (valores típicos con V_{DD} entre 2,7 V y 3,6 V).
- Error de offset $\pm 0,1$ LSB, error de ganancia $\pm 0,2$ LSB (valores típicos con V_{DD} entre 2,7 V y 3,6 V).
- SNR de 72,5 dB, THD de -80 dB (valores típicos con entrada de 100 kHz).
- Tensión de alimentación de 2,7 V a 5,25 V (utilizada como referencia para la conversión).
- Rango de escala completa de 0 V a V_{DD} (V_{REF}).
- Frecuencia de muestreo de 1 MSa/s (máxima).
- Interfaz de comunicación serie SPI de 20 MHz de tres líneas: CS, SCLK, SDATA.

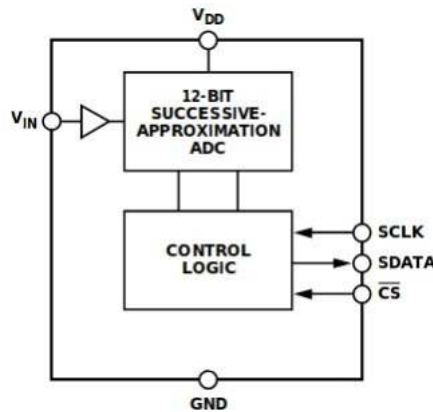


Figura 3.24: Diagrama en bloques ADCS7476

Como se observa en el diagrama en bloques de la figura 3.24, el ADC cuenta con seis conexiones: la interfaz SPI (SCLK, SDATA y CS), alimentación (V_{DD} y GND) y la entrada V_{IN} . La interfaz de comunicación sirve como temporización para el proceso de adquisición. Esta situación puede ser observada en el diagrama temporal de la interfaz SPI de los convertidores [36], presentado a continuación:

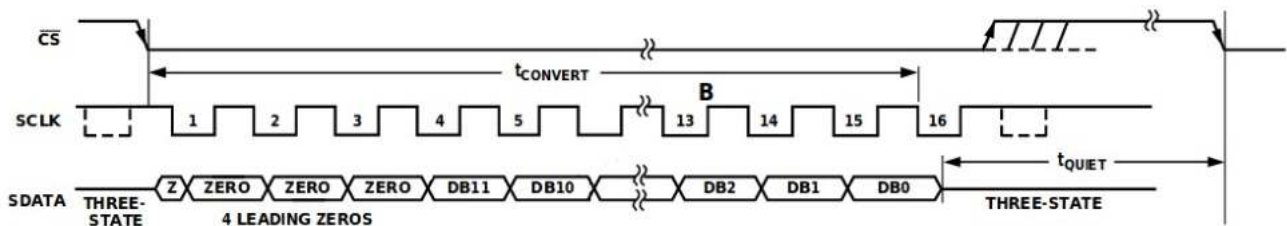


Figura 3.25: Diagrama temporal de la interfaz del convertidor ADCS7476

La señal de CS inicia el proceso de conversión de los datos, que requiere 16 flancos de reloj. Por cada muestra, se envían cuatro ceros y luego 12 bits de información, ocurriendo el cambio de la señal de datos en los flancos descendentes del reloj. Adicionalmente, es necesario un tiempo de “silencio” entre dos adquisiciones consecutivas. Para lograr la frecuencia de 1 MSa/s requerida en el diseño, se utiliza un reloj de 20 MHz. Es decir, entre cada transacción de 16 bits se presenta un tiempo de silencio correspondiente a cuatro ciclos de reloj.

El ADC es alimentado en base a una referencia de 3,3 V, el valor de tensión elegido resulta conveniente, ya que es compatible con los niveles lógicos utilizados en la FPGA [9], evitando el agregado de componentes electrónicos adicionales destinados a la conversión de niveles de tensión.

3.4.10 Generación de offset

Finalmente, se presenta la etapa encargada de generar el nivel de tensión de offset ajustable, sumado en la entrada de cada canal. El circuito está compuesto por un DAC de 12 bits, cuyo voltaje de salida varía entre 0 V y 5 V (en 4096 pasos) y un amplificador operacional, que lleva la señal a un rango de ± 5 V.

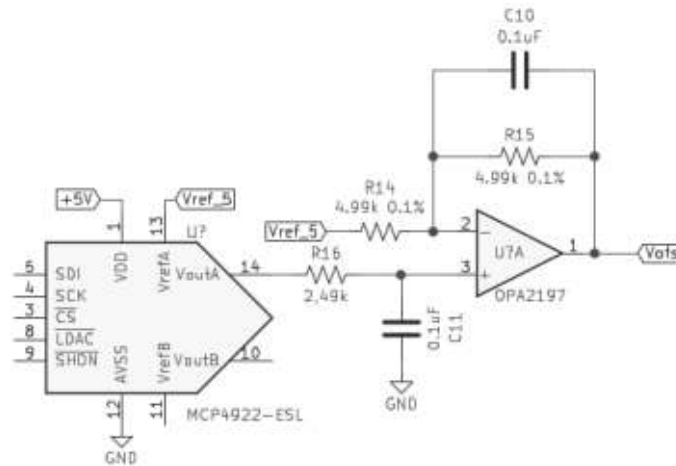


Figura 3.26: Circuito de generación de offset

El diagrama esquemático de esta etapa se observa en la figura 3.26, en la cual se omite la alimentación del amplificador, la interfaz de comunicación del DAC y las conexiones relacionadas al segundo canal de adquisición. El DAC utilizado es el modelo MCP4922-ESL [37] de Texas Instruments. Se trata de un convertor dual, por lo que permite la implementación de ambos canales en un mismo integrado. Algunas de sus características, obtenidas de la hoja de datos, son:

- Arquitectura tipo divisor resistivo.
- Resolución de 12 bits, DNL de $\pm 0,2$ LSB, INL de ± 2 LSB.
- Voltaje de alimentación de 5,5 V (máximo).
- Buffer de salida rail-to-rail con opción de ganancia 1 o 2 veces.
- Entrada de referencia externa (independiente de la alimentación).
- Interfaz de comunicación SPI de 20 MHz (máximo).

En el diseño realizado, el voltaje de alimentación es de 5 V y la tensión de escala completa es provista por una referencia de tensión externa de 5 V. La ganancia de salida es unitaria, por lo que se obtiene un voltaje en el rango 0 V a 5 V. Cabe destacar que la entrada de referencia del DAC incorpora un buffer de alta impedancia opcional, sin embargo, su activación limita el rango posible de valores de referencia. En este caso, el buffer se encuentra deshabilitado, siendo la impedancia de entrada de 165 k Ω en paralelo con 7 pF.

La salida del DAC es amplificada 2 veces, logrando la excursión de 10 V_{PP} requerida. Adicionalmente, se resta un offset de 5 V, obteniéndose el rango final de ± 5 V. Dado que el nivel

de tensión es sumado a la señal de entrada, debe ser estable y contar con la menor cantidad de ruido posible, es por eso que se realiza un filtrado pasa bajos.

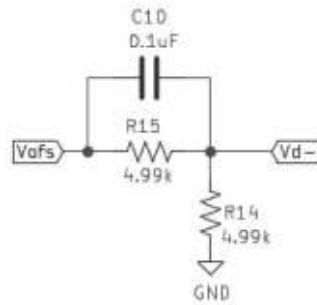


Figura 3.27: Red de realimentación del amplificador de offset

A fin de calcular la transferencia de la etapa de amplificación de la figura 3.26, se obtiene primero la transferencia de la red de realimentación del amplificador, que se observa en la figura 3.27.

$$H(S) = \frac{R_{14}}{R_{14} + R_{15}} \cdot \frac{1 + R_{15} \cdot C_{10} \cdot S}{1 + (R_{14} // R_{15}) \cdot C_{10} \cdot S} \quad (3.35)$$

$$f_z = \frac{1}{2 \cdot \pi \cdot R_{15} \cdot C_{10}} = 318 \text{ Hz} \quad (3.36)$$

$$f_p = \frac{1}{2 \cdot \pi \cdot (R_{14} // R_{15}) \cdot C_{10}} = 636 \text{ Hz} \quad (3.37)$$

La realimentación, asociada a la ecuación 3.35, tiene una ganancia en baja frecuencia dada por el divisor resistivo, un cero y luego un polo, siendo la ganancia final unitaria. Dado que las singularidades de la red de realimentación se ubican en baja frecuencia, respecto a el ancho de banda del amplificador operacional, se puede aproximar la transferencia a lazo cerrado como $1/H(S)$. Haciendo uso de esta simplificación, se obtiene la transferencia del circuito:

$$\frac{V_{OFS}(S)}{V_{DAC}} = \frac{2}{1 + R \cdot C \cdot S} \quad (3.38)$$

$$\frac{V_{OFS}(S)}{V_{REF}} = \frac{-1}{1 + R \cdot C \cdot S} \quad (3.39)$$

$$f_p = \frac{1}{2 \cdot \pi \cdot C \cdot R} = 318 \text{ Hz} \quad (3.40)$$

Donde V_{REF} representa el voltaje de referencia de 5 V y V_{DAC} representa la tensión a la salida del convertor analógico digital. Cabe destacar que en las ecuaciones 3.38, 3.39 y 3.40 se hace uso de las siguientes simplificaciones:

$$R_{14} = R_{15} = R = 4,99 K \Omega \quad (3.41)$$

$$R_{16} = R/2 = 2,49 K \Omega \quad (3.42)$$

$$C_{10} = C_{11} = C = 0,1 \mu F \quad (3.43)$$

El circuito propuesto brinda la transferencia en baja frecuencia deseada y establece un ancho de banda para ambas señales de 318 Hz. Esto permite filtrar ruido o transitorios presentes tanto en la salida del DAC como en la referencia, obteniendo una tensión de offset estable. El valor de frecuencia de corte se asocia a una constante de tiempo de 0,5 ms, asegurando un tiempo de respuesta aceptable ante un cambio en el nivel de offset.

3.5 Transferencia total

En este apartado se calcula la ganancia total asociada al front-end analógico, en base a las transferencias parciales obtenidas en las secciones anteriores. Adicionalmente, se presentan tablas con diversos parámetros relacionados a cada escala, como por ejemplo, las excursiones de tensión máxima de cada etapa, atenuador de entrada utilizado, ganancia del PGA necesaria, etc. En la segunda parte de esta sección, se calcula la transferencia total para el caso de la tensión se offset. Esto permite determinar el rango y resolución del nivel de offset para cada ajuste de escala.

A continuación, se repiten las expresiones de ganancia de cada etapa:

$$\frac{V_2}{V_{IN}} = \frac{R_A}{R_A + R_B} = At \quad (3.44)$$

$$\frac{V_3}{V_2} = 1 + \frac{R_7}{R_6} = 1,5 \quad (3.45)$$

$$\frac{V_4}{V_3} = \frac{R_{P2}}{R_{P1} + R_{P2}} = K_P \quad (3.46)$$

$$\frac{V_5}{V_4} = \left(1 + \frac{R_9}{R_8}\right) = 1,5 \quad (3.47)$$

$$\frac{V_6}{V_5} = -G_{PGA} \quad (3.48)$$

$$\frac{V_7}{V_6} = -\frac{R_{11}}{R_{10}} = -1 \quad (3.49)$$

Las transferencias parciales son válidas únicamente en baja frecuencia, es decir, se omite la variable frecuencial y cualquier tipo de dependencia y singularidad. La transferencia total se obtiene como el producto de las ecuaciones 3.44 a 3.49, y se presenta a continuación:

$$\frac{V_7}{V_{IN}} = G_{TOT} = (2,25) \cdot (At) \cdot (K_p) \cdot (G_{PGA}) \quad (3.50)$$

La configuración de cada etapa del front-end, necesaria para lograr la ganancia total de cada escala, se enumera a continuación:

Escala (V/div)	Cantidad divisiones	V_{IN} (V_{pp})	G_{TOT} (V/V)	At (V/V)	K_p	N_p	G_{PGA} (V/V)
5	10	50	0,06	0,1	0,27	68	1
2	20	40	0,075	0,1	0,33	85	1
1	20	20	0,15	0,1	0,67	171	1
0,5	20	10	0,3	0,1	0,67	171	2
0,2	20	4	0,75	0,1	0,67	171	5
0,1	20	2	1,5	0,9	0,74	190	1
0,05	20	1	3	0,9	0,74	190	2
0,02	20	0,4	7,5	0,9	0,74	190	5
0,01	20	0,2	15	0,9	0,74	190	10

Tabla 3.3: Configuración del front-end para cada escala

En la tabla 3.3, se puede observar el uso del atenuador de 0,1 veces para el caso de las cinco escalas de mayor rango y el empleo de una atenuación de 0,9 veces en las cuatro escalas menores. La presencia de un atenuador variable a la entrada permite reducir las ganancias requeridas en las etapas posteriores. Con respecto al potenciómetro digital, se utilizan constantes de atenuación K_p en el rango 0,27 a 0,74. El diseño es realizado de manera que exista un margen a ambos extremos del uso del potenciómetro.

Los valores numéricos N_p que deben ser programados en el potenciómetro, se listan en la ante última columna de la tabla 3.3. Para el caso del potenciómetro empleado en el diseño, con una resolución N de 8 bits, las constantes se obtienen en base a la siguiente expresión:

$$N_p = 2^N \cdot K_p = 256 \cdot K_p \quad (3.51)$$

En la tabla 3.4, se enumeran las tensiones pico a pico de las diferentes etapas para el caso de la máxima señal de entrada. Los rangos son calculados en base a las transferencias parciales presentadas al principio de esta sección.

Escala (V/div)	V_{IN} (V_{PP})	V_2 (V_{PP})	V_3 (V_{PP})	V_4 (V_{PP})	V_5 (V_{PP})	V_6 (V_{PP})	V_7 (V_{PP})
5	50	5	7,5	2	3	3	3
2	40	4	6	2	3	3	3
1	20	2	3	2	3	3	3
0,5	10	1	1,5	1	1,5	3	3
0,2	4	0,4	0,6	0,4	0,6	3	3
0,1	2	1,8	2,7	2	3	3	3
0,05	1	0,9	1,35	1	1,5	3	3
0,02	0,4	0,36	0,54	0,4	0,6	3	3
0,01	0,2	0,18	0,27	0,2	0,3	3	3
Máx.	-	6,6	10	10	10	10	3,3

Tabla 3.4: Rangos de voltaje en todas las etapas para cada escala

En la última fila de la tabla 3.4, se precisan los máximos valores posibles de excursión de las señales sin que ocurra distorsión por saturación. Se puede observar que los valores pico a pico se encuentran dentro del rango sin distorsión para todos los casos posibles. El máximo valor para V_2 queda prefijado por la red de protección de entrada, que actúa a partir de $\pm 3,3$ V. Teniendo en cuenta que todas las etapas intermedias (V_3 a V_6) cuentan con tensiones de alimentación de ± 5 V, el límite de excursión queda establecido en $10 V_{PP}$. Por último, la salida V_7 se ubica en un rango máximo de $3,3 V_{PP}$, prefijado por la tensión de escala completa del ADC.

Con respecto a la tensión de offset, a fin de obtener la transferencia total desde la salida del DAC (V_{DAC}) a la entrada del ADC (V_7), se hace uso de las siguientes expresiones:

$$\frac{V_{OFS}}{V_{DAC}} = \left(1 + \frac{R_{15}}{R_{14}}\right) = 2 \quad (3.52)$$

$$\frac{V_3}{V_{OFS}} = -\left(\frac{R_7}{R_6}\right) = -0.5 \quad (3.53)$$

$$\frac{V_7}{V_3} = (1,5) \cdot (K_P) \cdot (G_{PGA}) \quad (3.54)$$

Donde la última fórmula surge del producto de las expresiones 3.46 a 3.49. El valor total, se obtiene del producto de la transferencias parciales 3.52, 3.53 y 3.54 y se calcula a continuación:

$$\frac{V_7}{V_{DAC}} = -G_{DAC} = -(1,5) \cdot (K_P) \cdot (G_{PGA}) \quad (3.55)$$

Haciendo uso de la ganancia asociada al DAC, se pueden obtener diversos parámetros de utilidad relacionados a la tensión de offset. Los más importantes se listan en la tabla 3.5 a continuación:

Escala (V/div)	G_{DAC} (V/V)	Rango DAC V_7 (V)	Resolución DAC V_7 (mV)	Rango DAC V_{IN} (V)	Resolución DAC V_{IN} (mV)
5	0,4	2	0,49	33,33	8,14
2	0,5	2,5	0,61	33,33	8,14
1	1	5	1,22	33,33	8,14
0,5	2	10	2,44	33,33	8,14
0,2	5	25	6,1	33,33	8,14
0,1	1,11	5,56	1,36	3,7	0,9
0,05	2,22	11,11	2,71	3,7	0,9
0,02	5,56	27,78	6,78	3,7	0,9
0,01	11,11	55,56	13,56	3,7	0,9

Tabla 3.5: Rango y resolución del DAC para cada escala

En la primera columna se observa la ganancia desde el DAC a la entrada del ADC, obtenida en base a la ecuación 3.55. En las siguientes dos columnas se enumera el rango y resolución de la tensión del DAC en la entrada del ADC (V_7). Para el cálculo de dichos parámetros se utilizan las siguientes expresiones:

$$RNG_{DAC}(V_7) = V_{REF} \cdot G_{DAC} = (5V) \cdot G_{DAC} \quad (3.56)$$

$$RES_{DAC}(V_7) = \frac{RNG_{DAC}(V_7)}{2^N} = \frac{RNG_{DAC}(V_7)}{4096} \quad (3.57)$$

Donde V_{REF} es la tensión a escala completa del DAC, de 5 V, y N es la cantidad de bits del convertor utilizado, de 12 bits. Dado que el DAC es empleado para desplazar la señal de entrada, resulta de utilidad conocer el rango y resolución referido a la entrada V_{IN} . Ambos parámetros se listan en las últimas dos columnas de la tabla 3.5 y se calculan en base a las siguientes relaciones:

$$RNG_{DAC}(V_{IN}) = \frac{RNG_{DAC}(V_7)}{G_{TOT}} \quad (3.58)$$

$$RES_{DAC}(V_{IN}) = \frac{RNG_{DAC}(V_{IN})}{2^N} = \frac{RNG_{DAC}(V_{IN})}{4096} \quad (3.59)$$

Donde G_{TOT} es la ganancia del front-end para la señal de entrada, es decir la relación de V_7 y V_{IN} , obtenida anteriormente en esta sección. Las cuatro escalas más pequeñas cuentan con un rango de voltaje (referido a la entrada) aproximadamente 10 veces menor, lo que mejora en el mismo factor la resolución. El cambio en los valores se debe al atenuador de entrada, que modifica la ganancia de la señal antes del sumador de offset.

El diseño del front-end permite conservar una buena resolución en los pasos del nivel de offset en todos los ajustes verticales. Con el objetivo de mantener la linealidad y precisión, se establece un límite por software para el rango de offset, de ± 15 V para las primeras escalas y $\pm 1,75$ V para las últimas cuatro escalas. Cabe recordar que, para un valor del DAC de 2048 (2,5 V), el nivel de tensión sumado es nulo, obteniéndose en V_7 una tensión de 1,65 V.

3.6 Simulaciones

Con el objetivo de verificar el diseño, se llevan a cabo simulaciones de LTSpice en el tiempo y en frecuencia en base al diseño completo de la etapa analógica. En el caso de los amplificadores operacionales, se emplean los modelos de simulación provistos por el fabricante [31], mientras que para el PGA se utiliza el modelo presente en LTSpice. En la figura 3.28 se presenta la respuesta en frecuencia, con el acople de alterna habilitado, para las nueve escalas implementadas:

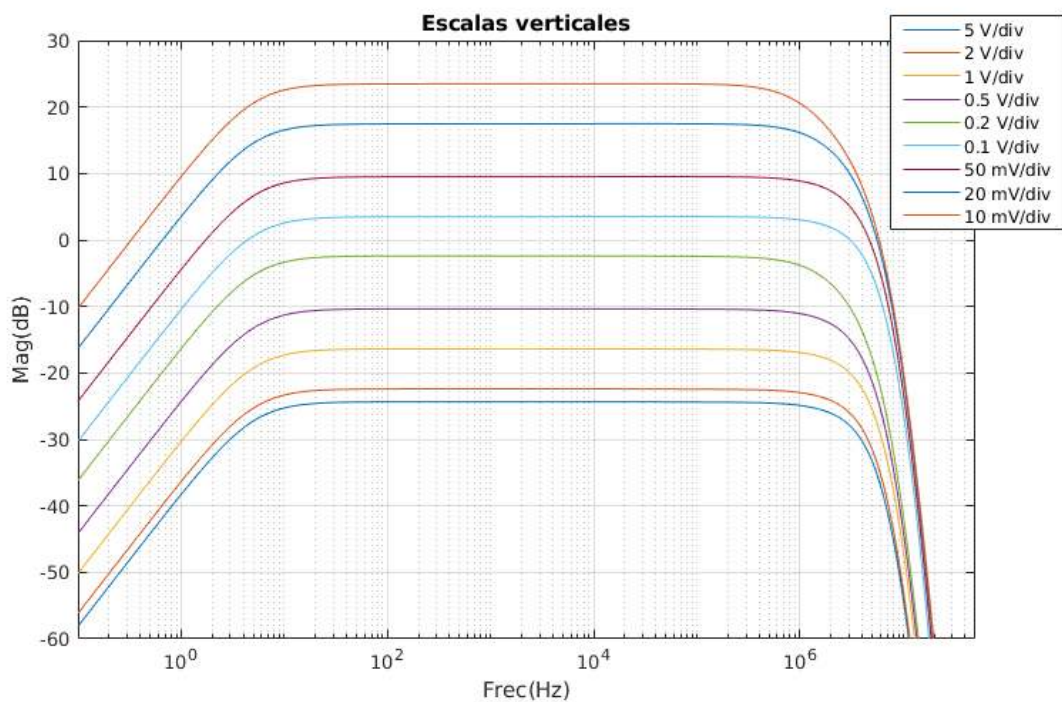


Figura 3.28: Simulación de respuesta en frecuencia de las escalas

De acuerdo a la figura 3.28, la transferencia en amplitud para cada escala cuenta con una respuesta plana en el rango frecuencial de interés. Por otro lado, los valores de ganancia y anchos de banda asociados a la simulación se enumeran en la tabla 3.6.

Escala (V/div)	B (3 dB) (MHz)	G_{TOT} (V/V)	G_{TOT} (dB)	G_{PGA} (V/V)
5	2,8	0,06	-24,44	1
2	3	0,075	-22,50	1
1	2,9	0,15	-16,48	1
0.5	2,6	0,3	-10,46	2
0.2	1,6	0,75	-2,50	5
0.1	3	1,5	3,52	1
0.05	2,4	3	9,54	2
0.02	1,7	7,5	17,50	5
0.01	1	15	23,52	10

Tabla 3.6: Anchos de banda y ganancias obtenidos por simulación

Los ganancias obtenidas concuerdan con los valores teóricos establecidos en las especificaciones de diseño, listados en la tabla 3.1 al principio del capítulo. Con respecto al ancho de banda, se observa una fuerte dependencia con la ganancia del PGA, por lo que esta etapa puede ser considerada un factor limitante. El mayor valor de ancho de banda se da para el caso de la escala de 2 V/div y es aproximadamente de 3 MHz. Por otro lado, para la escala más pequeña, de 10 mV/div, se obtiene el menor ancho de banda, de 1 MHz, asociado a una ganancia de 10 veces en el PGA.

Adicionalmente, se llevan a cabo dos simulaciones que permiten verificar el comportamiento temporal de la etapa analógica. La prueba es ejecutada para el caso de la escala de 10 mV/div, bajo una señal de entrada cuadrada de 1 kHz y 200 mV_{PP} de amplitud (máximo rango para esta escala). En la señal se introduce un valor medio de 1,5 V, el cual es restado mediante el offset de voltaje en la primera simulación y mediante el acople de alterna en la segunda simulación. Los resultados obtenidos para el primer caso se presentan a continuación:

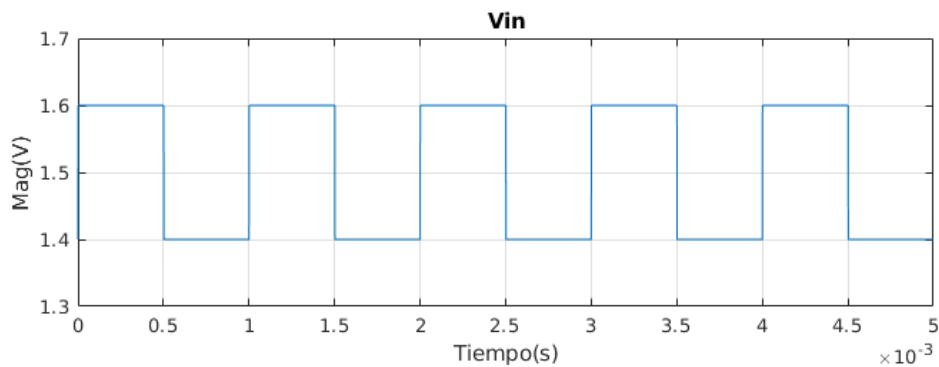


Figura 3.29: Señal de entrada para simulación temporal de offset

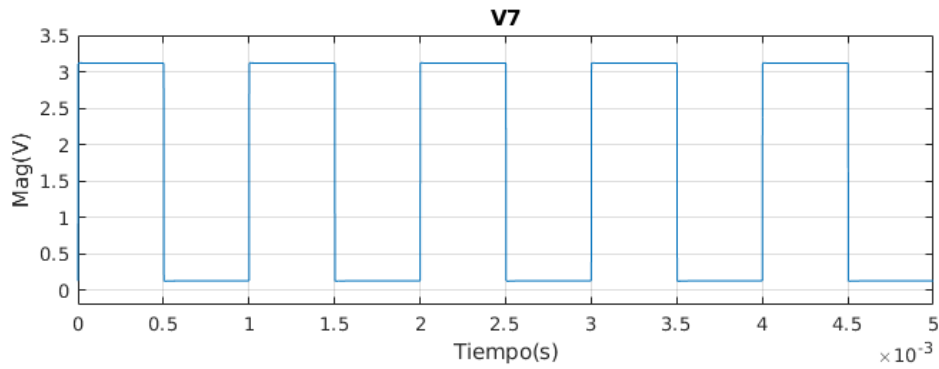


Figura 3.30: Señal de salida para simulación temporal de offset

En las formas de onda de la figuras 3.29 y 3.30 se puede denotar el correcto funcionamiento de la etapa de offset, restando el nivel de continua de la señal de entrada. La señal de salida, en la figura 3.30, no presenta sobrepicos ni oscilaciones significativas y el valor de ganancia es el esperado. La salida se encuentra centrada en 1,65 V con una amplitud de 3 V_{PP}, valores adecuados para el rango del ADC. Como fue mencionado, la prueba es repetida, en este caso recurriendo al acople de alterna:

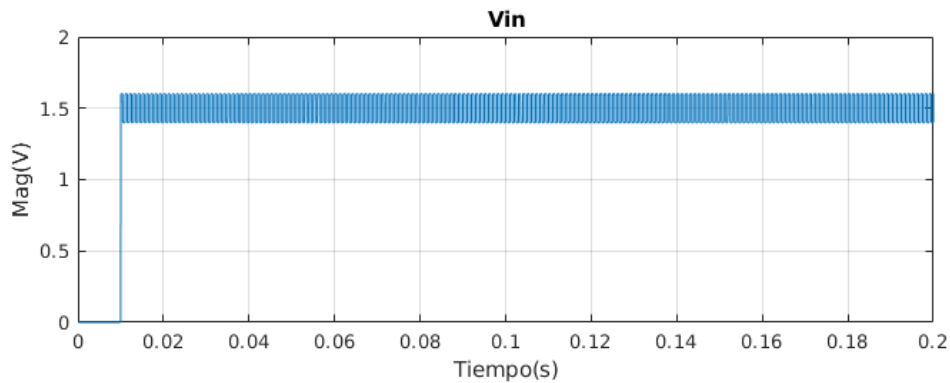


Figura 3.31: Señal de entrada para simulación temporal de acople de alterna

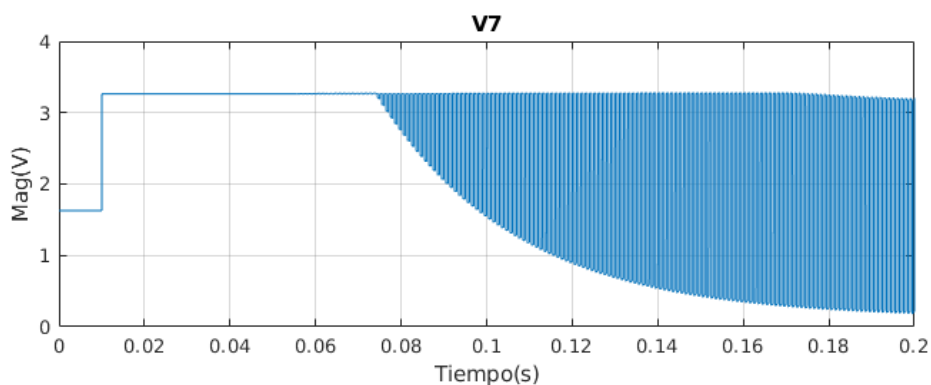


Figura 3.32: Señal de salida para simulación temporal de acople de alterna

En la figura 3.31 se presenta la forma de onda de entrada utilizada en esta prueba. La señal mantiene un valor nulo durante 10 ms, luego de lo cual existe un flanco de 1,5 V. Este tipo de entrada permite observar mejor la respuesta transitoria del sistema. Luego del escalón de 1,5 V, la señal a la salida del último amplificador se encuentra saturada al voltaje positivo de alimentación (3,3 V), de acuerdo a la figura 3.32. Esta situación se mantiene hasta que actúe el filtro de acople de alterna, luego de lo cual se obtiene la forma de onda de régimen permanente. Teniendo en cuenta la frecuencia de corte del acople de 4,8 Hz, cuya constante de tiempo asociada es de 33 ms, el retardo observado de 150 ms es el esperado.

3.7 Interfaz con la FPGA

Durante el desarrollo de las secciones anteriores, se omitieron circuitos complementarios necesarios para la comunicación con la FPGA. En esta sección, se presentan todos los detalles relacionados a la interfaz con la placa de desarrollo DE0-Nano-SoC. La comunicación se lleva a cabo a través de un cable cinta de 40 hilos con conectores de 2x20 pines. Las señales de datos necesarias son las siguientes:

- Interfaz SPI ADC (3 por canal): Cada uno de los dos conversores cuenta con líneas de CLK, CS y datos. Las dos primeras son señales de control generadas por la FPGA, la última envía las muestras desde el ADC a la FPGA.
- Interfaz SPI DAC (3 en total): Consiste en las líneas de CLK, CS y datos, provenientes de la FPGA, que realizan la configuración del DAC dual, compartido entre ambos canales.
- Interfaz SPI de potenciómetro digital (3 en total): Consiste en las líneas de CLK, CS y datos, provenientes de la FPGA, que realizan la configuración del potenciómetro dual, compartido entre ambos canales.
- Ganancia programable (3 por canal): Se trata de tres líneas digitales por canal (G2, G1 y G0) que ajustan el valor de ganancia de los PGA.
- Acople de alterna (1 por canal): Se refiere a la señal necesaria para activar el relé de estado sólido que conecta el capacitor de acople de alterna.
- Selección del atenuador (1 por canal): Consiste en una señal digital encargada de activar el relé mecánico que selecciona el atenuador de entrada.
- LED de comunicación (1 línea): se agrega un LED [38] en el PCB del front-end, controlable desde la FPGA, que permite verificar el estado de comunicación.

Las señales presentadas se pueden observar en el diagrama de pines del conector en el front-end:

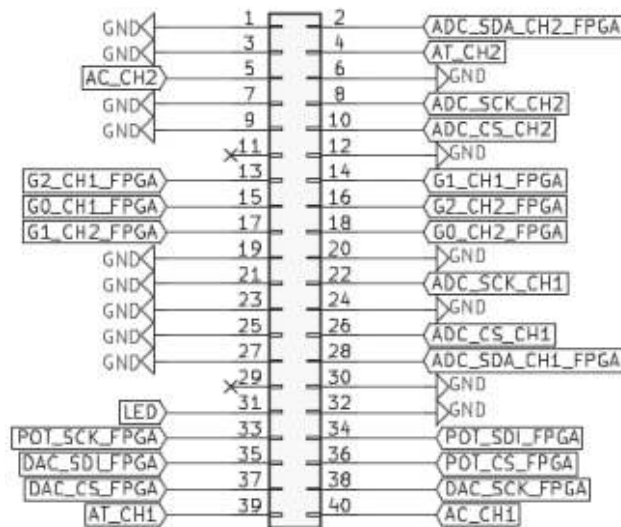


Figura 3.33: Diagrama de pines de conector interfaz en PCB de acondicionamiento

De acuerdo a la figura 3.33, se usan un total de 23 entradas y salidas digitales. Los pines 11 y 29 no se encuentran conectados, ya que brindan 3,3 V y 5 V provenientes de la FPGA y su capacidad de corriente es menor a la requerida para la alimentación del front-end. Las 15 líneas restantes conectan los planos de masa de ambas placas, reduciendo la impedancia y brindando un retorno a las señales digitales [39].

El DAC, los amplificadores de ganancia programable y el potenciómetro digital funcionan con tensiones de 5 V. Dado que la FPGA utiliza tensiones de 3,3 V, es necesario emplear circuitos integrados de traslado de niveles de tensión lógica. En la figura 3.34, se presenta la implementación circuital encargadas de esta tarea. Cabe destacar que se omiten capacitores de desacople y resistencias de pull-up y pull-down que forman parte del diseño.

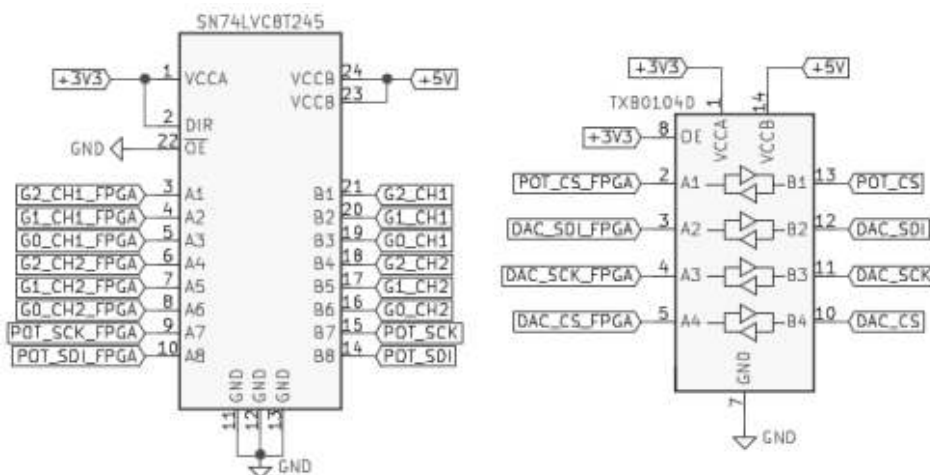


Figura 3.34: Traslado de niveles de tensión lógicos

Como se puede ver, se usan los integrados SN74LVC8T245 [40] y TXB0104D [41] de Texas Instruments, convertidores de 8 bits y 4 bits respectivamente. Cabe destacar que no se requiere ninguna característica especial para estos integrados, ya que las señales digitales involucradas son de baja frecuencia con respecto a los tiempos de propagación típicos asociados a los convertidores de niveles de tensión.

En cuanto a la comunicación entre los ADC y la FPGA, la frecuencia del bus de datos SPI es de 20 MHz, por lo que se deben tomar los recaudos necesarios para asegurar la integridad de las señales digitales. Los ADC de ambos canales se ubican lo más cerca posible del conector, minimizando el largo de las pistas digitales en el PCB.

Adicionalmente, se intercalan líneas de masa en el cable cinta entre las líneas de la interfaz SPI. Esto permite aislar las señales digitales entre sí, evitando interferencias cruzadas que degraden la comunicación [39]. Así mismo, las líneas de masa actúan de retorno para las corrientes, formando cada par una línea de transmisión.

Dado que la FPGA sólo cuenta con dos pines de masa en el conector, se incorpora al diseño una placa adaptadora, destinada a aumentar las líneas de GND e intercalar las con las señales digitales. El adaptador se coloca entre la FPGA y el conector del cable, uniendo con un plano de masa los pines marcados como GND. En la figura 3.35, se presenta el diagrama esquemático y una vista 3D del adaptador diseñado.

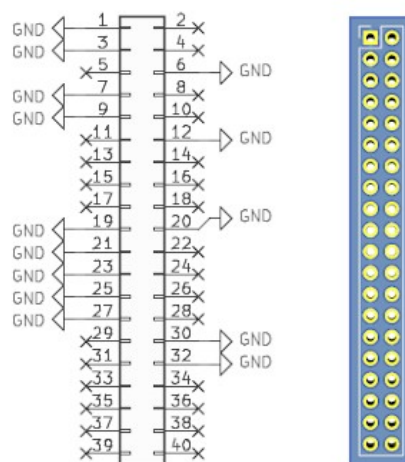


Figura 3.35: Diagrama esquemático y layout de placa adaptadora

Debido a la presencia de la capacidad distribuida del cable, se incorpora un buffer en la línea de salida de datos del ADC. Este último, debe ser capaz de entregar la corriente necesaria para generar los pulsos digitales a la frecuencia de 20 MHz. Así mismo, la incorporación del buffer busca reducir el consumo y presencia de transitorios en la alimentación del ADC, dado que el convertidor se alimenta directamente de la referencia de voltaje [42]. A continuación, en la figura 3.36, se presenta la implementación circuital asociada:

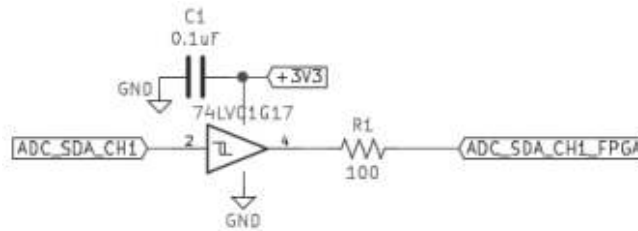


Figura 3.36: Buffer de línea de datos de ADC

Se recurre a un buffer Schmitt trigger 74LVC1G17 [43] de Texas Instruments, alimentado del rail de 3,3 V. El dispositivo es capaz de entregar corrientes de hasta ± 24 mA a su salida, por lo que resulta apropiado para esta aplicación. Un factor que se debe tener en cuenta es su retardo de propagación de entrada a salida, en este caso de 1,5 ns típico (8 ns máximo). Los datos se envían en base a un reloj de 20 MHz asociado a un periodo de 50 ns, por lo que el retardo es lo suficientemente pequeño. Adicionalmente, se agrega una resistencia de 100 Ω en serie a la salida, cuyo objetivo es reducir posibles oscilaciones en el cable debido a la inductancia y capacidad distribuida [44].

Con respecto a las señales de CLK y CS del ADC, provenientes de la FPGA, se agregan diodos de protección en la entrada del convertor, destinados a absorber cualquier tipo de oscilaciones que superen el rango 0 V a 3,3 V producidas en el cable. En la figura 3.37, se puede observar la protección del ADC, cabe aclarar que solo se muestra el circuito asociado al primer canal, siendo el del segundo canal idéntico.



Figura 3.37: Diodos de protección entradas digitales de ADC

Se emplean diodos Schottky dobles BAT54SHMFH [45] de ROHM Semiconductor, debido a su alta velocidad y bajo voltaje en directa (0,32 V a 1 mA). Las dos características mencionadas aseguran que la protección actúe antes que los diodos internos del ADC, absorbiendo la mayor parte de la energía asociada a un sobrepico en la tensión transitorio [35]. Cada diodo soporta un voltaje en inversa de 30 V y una corriente en directa de 200 mA, siendo más que suficientes para la aplicación.

3.8 Alimentación en placa principal

Teniendo en cuenta el diseño propuesto para la etapa de acondicionamiento, se requieren tensiones de alimentación de ± 5 V y 3,3 V. Del análisis de las hojas de datos de los integrados y la topología del circuito, se estima un consumo máximo de 200 mA en cada rail de alimentación. Con

lo cual se decide alimentar el front-end a través de una fuente externa, cuyo voltaje nominal sea de ± 9 V. La fuente se implementa en un PCB secundario, el cual se describe en la sección 3.9.

Para derivar los distintos voltajes requeridos, se emplean reguladores lineales en el PCB principal. Los reguladores utilizados son: UA78M05C [46] (5 V), UA78M33C [46] (3,3 V) y MC79M05C [47] (-5 V), los dos primeros de Texas Instruments y el último de ON Semiconductor. Algunas de las características principales para esta serie de reguladores lineales se listan a continuación:

- Corriente de salida de 500 mA (máxima).
- Tensión de alimentación de 25 V (máxima).
- Caída de voltaje de 2 V (típica).
- Protección térmica y de cortocircuito.
- Rechazo de ripple mínimo de 62 dB (120 Hz) bajo una corriente de carga de 300 mA.

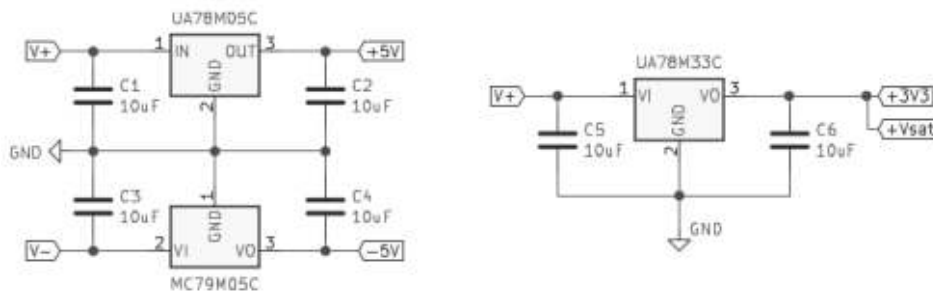


Figura 3.38: Regulación lineal y filtrado (± 5 V y 3,3 V)

En la figura 3.38 se presenta la implementación circuital de las etapas de regulación lineal. Se observa el agregado de capacitores cerámicos de $10 \mu\text{F}$, tanto a la entrada como salida, de acuerdo a las recomendaciones en las hojas de datos de los reguladores lineales [46] [47].

Como se analizó anteriormente, en la sección 3.4.4, las etapas de entrada de los canales cuentan con diodos de protección que funcionan con tensiones de 3,3 V y $-V_{\text{SAT}}$ (V_{SAT} y $-V_{\text{SAT}}$). Con el objetivo de obtener el voltaje $-V_{\text{SAT}}$, se emplea un diodo Zener de 3,3 V polarizado en inversa. Esto resulta ser una implementación más sencilla y es aceptable debido a las bajas corrientes requeridas, 0,45 mA máximo para cada canal, según la sección 3.4.4.

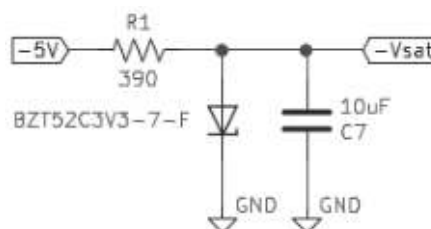


Figura 3.39: Regulación lineal y filtrado ($-3,3$ V)

El diodo que se utiliza es el modelo BZT52C3V3-7-F [48] de Diodes Incorporated, el cual cuenta con una potencia máxima de 500 mW y una tensión de Zener nominal de 3,3 V, esta última especificada a una corriente de 5 mA. El diodo se alimenta de la tensión de -5 V y se polariza con una corriente de 4,35 mA. El cálculo de este valor se presenta en la ecuación 3.60 a continuación:

$$I_{pol} = \frac{5V - 3.3V}{R_1} = 4,35mA \quad (3.60)$$

Por otro lado, para derivar las tensiones de referencia utilizadas por el DAC y los ADC, se recurre a circuitos integrados específicos para este propósito [49]. La implementación práctica se puede ver en la figura 3.40.

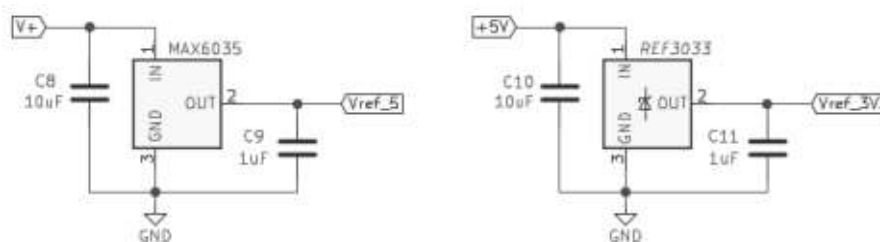


Figura 3.40: Referencias de tensión (5 V y 3,3 V)

La referencia de 5 V se obtiene del integrado MAX6035 [50] de Maxim, alimentado directamente del voltaje positivo de entrada (V_+). Esta referencia de tensión tiene una tolerancia inicial de 0,2%, un corrimiento por temperatura de 25 ppm/ $^{\circ}C$ y es capaz de entregar 10 mA. Una especificación importante es su estabilidad con carga capacitiva, en este caso 5 μF máximo, por lo que se utiliza un capacitor de desacople de 1 μF a la salida.

En el caso de la referencia de 3,3 V se utiliza un integrado REF3033 [51] de Texas Instruments, el cual cuenta con una exactitud inicial de 0,2%, un corrimiento de 50 ppm/ $^{\circ}C$ y es capaz de entregar una corriente de 25 mA. La carga capacitiva máxima que asegura estabilidad es de 10 μF , por lo que nuevamente se utiliza un capacitor de desacople de 1 μF a su salida.

3.9 Alimentación en placa secundaria

Con el objetivo de darle modularidad al sistema y permitir futuras actualizaciones, la fuente de alimentación se implementa en una placa aparte. En un principio, se evaluó la posibilidad de emplear dos fuentes conmutadas, obteniéndose de esta forma las tensiones positivas y negativas requeridas.

Dado que el front-end consiste mayoritariamente de etapas de acondicionamiento analógicas, es deseable tener tensiones estables y sin variaciones significativas. La salida de este tipo de fuente presenta ripple a la frecuencia de conmutación, cuyo valor de amplitud se encuentra en las decenas de mV [52]. Este tipo de circuito opera en rangos de frecuencia que van desde 20 kHz a 2 MHz normalmente.

A fin de evaluar el posible efecto del ripple en la etapa de acondicionamiento, se presenta un gráfico del factor de rechazo a la fuente de alimentación (PSRR) y de modo común (CMRR) de los amplificadores operacionales utilizados [24]:

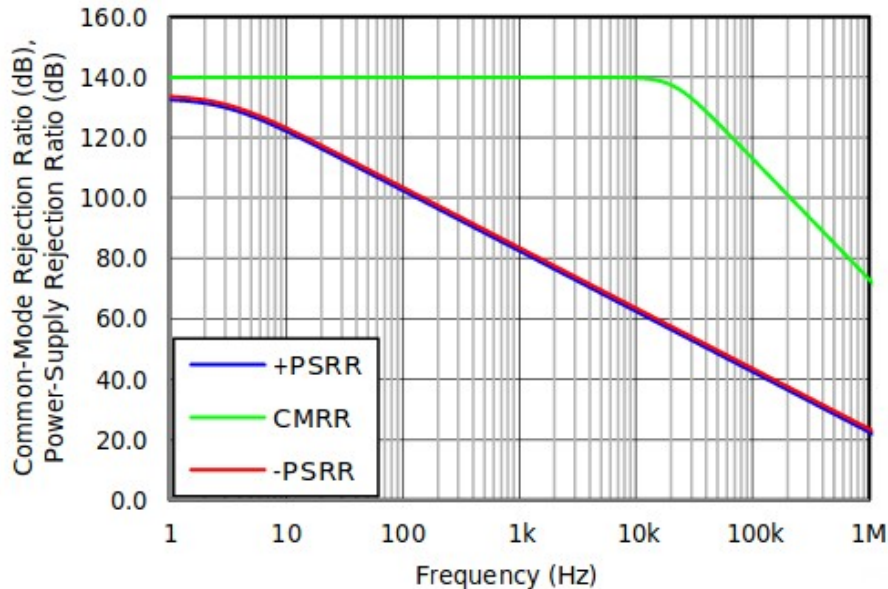


Figura 3.41: PSRR y CMRR vs frecuencia (OPA2197)

Se puede ver que, conforme aumenta la frecuencia, el rechazo disminuye. El acoplamiento de ripple en las señales analógicas adquiridas no resulta aceptable. Teniendo en cuenta el reducido consumo del front-end, se decide utilizar una fuente lineal bipolar regulada. Este tipo de fuente es sencilla de diseñar, presenta pocos componentes y es capaz de entregar los niveles de corriente requeridos.

Así mismo, debido al empleo de regulación lineal en todos los rails de alimentación, se pueden obtener los niveles de calidad requeridos. Por otro lado, la presencia de algún remanente de ripple es altamente rechazado por los amplificadores del front-end (por encontrarse a la frecuencia de línea). Las únicas desventajas de esta topología son el tamaño y la eficiencia, pero al tratarse de un circuito de acondicionamiento analógico de precisión, no son un problema significativo en este sistema.

Teniendo en cuenta las estimaciones de consumo, la fuente debe ser capaz de entregar una corriente de carga de 200 mA por cada rama (positiva y negativa). Con el objetivo de brindar la posibilidad de expandir el diseño del front-end a futuro, se diseña una fuente capaz de entregar una corriente I_L de 0,5 A por rama. Así mismo, dado que los reguladores lineales de 5 V requieren una tensión mínima de 7 V [46], [47], el voltaje en cada rama de la fuente debe ser superior a 8 V en todo momento, asegurando la correcta operación de los integrados.



Figura 3.42: Transformador 220 V a 12 V externo

La fuente diseñada hace uso de un transformador de 220 V a 12 V_{RMS} (17 V_P) externo, capaz de entregar una corriente de carga máxima de 1 A. Las etapas de rectificado y filtrado se implementan en la placa secundaria. En la figura 3.43, se presenta el diagrama esquemático del diseño de estas etapas.

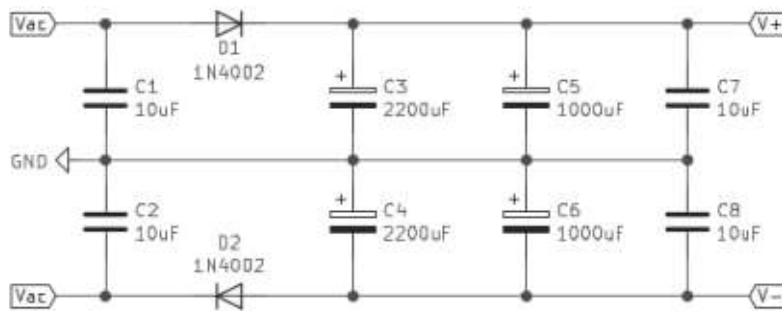


Figura 3.43: Diagrama esquemático de fuente lineal bipolar

El circuito consiste en dos diodos rectificadores de media onda seguidos por capacitores electrolíticos de filtrado. Adicionalmente, se agregan capacitores cerámicos de 10 µF, debido a su baja ESR, destinados a filtrar el ruido y mejorar la respuesta transitoria. Teniendo en cuenta que la rama positiva y negativa cuentan con las mismas especificaciones, se calculan los componentes en base al siguiente circuito equivalente:

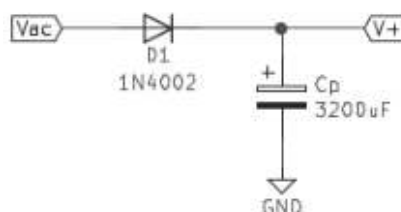


Figura 3.44: Circuito equivalente de etapa de rectificado y filtrado

En el diagrama esquemático de la figura 3.44, no se incluyen los capacitores cerámicos presentes en el diseño, debido a su bajo valor en relación a los capacitores electrolíticos. Adicionalmente, se unificaron C₃ y C₅ en un único capacitor C_P de 3200 µF. La metodología de diseño utilizada consiste en suponer una corriente de carga constante I_L de 0,5 A (valor máximo) en la salida de la

etapa de rectificado y filtrado. Esta suposición es válida debido al empleo de una etapa de regulación lineal posterior.

Con el objetivo de dimensionar la capacidad C_p necesaria para asegurar la tensión mínima, se obtiene el valor de ripple pico a pico en el voltaje a la salida. Bajo estas condiciones, la ecuación que rige el voltaje V^+ cuando el diodo no conduce es la siguiente:

$$V^+(t) = (V_{ACP} - V_D) - \frac{1}{C_p} \cdot I_L \cdot t \quad (3.61)$$

Es decir, una recta de descarga constante, cuyo valor inicial es la diferencia entre el valor pico de la señal de entrada y la caída en el diodo (considerada 1 V). La tensión de entrada se puede expresar de la siguiente forma:

$$V_{AC}(t) = V_{ACP} \cdot \cos\left(\frac{2 \cdot \pi}{T} \cdot t\right) \quad (3.62)$$

Donde T es el periodo de línea de 20 ms. En la figura 3.45, presentada a continuación, se observa la superposición de la recta de descarga con la tensión de entrada:

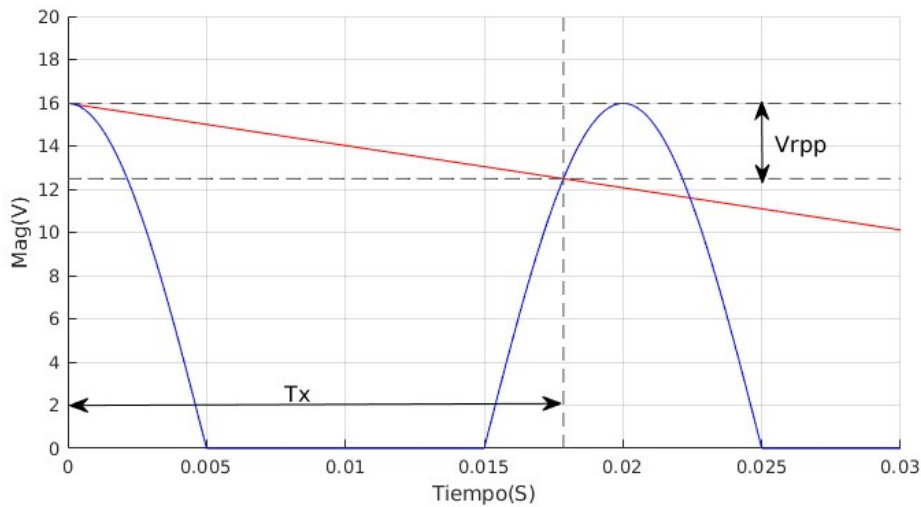


Figura 3.45: Recta de descarga (rojo) y entrada rectificada (azul)

El diodo vuelve a conducir en el instante T_x , por ende, igualando ambas curvas se puede obtener el tiempo de descarga:

$$(V_{ACP} - V_D) - \frac{1}{C_p} \cdot I_L \cdot T_x = (V_{ACP} - V_D) \cdot \cos\left(\frac{2 \cdot \pi}{T} \cdot T_x\right) \quad (3.63)$$

Por otro lado, el valor de ripple pico a pico puede ser obtenido mediante la ecuación 3.64, presentada a continuación:

$$V_{RPP} = \frac{1}{C_p} \cdot I_L \cdot T_x \quad (3.64)$$

En el cálculo se considera una tolerancia del 20% en el valor de los capacitores. El peor caso de ripple, se da para el menor valor de capacidad. Haciendo uso de las ecuaciones 3.63 y 3.64, se obtiene el valor de ripple a la salida:

$$(16V) - \frac{1}{(0,8) \cdot (3200\mu F)} \cdot (0,5A) \cdot T_x = (16V) \cdot \cos\left(\frac{2 \cdot \pi}{20ms} \cdot T_x\right) \rightarrow T_x = 17,9ms \quad (3.65)$$

$$V_{RPP} = \frac{1}{(0,8) \cdot (3200\mu F)} \cdot (0,5A) \cdot (17,9ms) = 3,5V \quad (3.66)$$

La tensión mínima de salida surge de la siguiente expresión:

$$V^+(min) = V_{ACP} - V_D - V_{RPP} \quad (3.67)$$

Considerando una variación de la tensión de línea del 20%, y haciendo uso de la ecuación 3.67, se obtiene el valor mínimo de V^+ :

$$V^+(min) = (0,8) \cdot (17V) - (1V) - (3,5V) = 9V \quad (3.68)$$

Se observa que, con el valor de capacidad elegida, la tensión de salida es siempre mayor que 8 V, por lo que se cumple el criterio de diseño. La capacidad de 3200 μF se divide en dos capacitores electrolíticos, de esta forma, se reduce la corriente de ripple en cada capacitor y la resistencia serie equivalente (ESR).

Con respecto a los rectificadores, en el diseño se emplean diodos 1N4002-T [53] del fabricante Diodes Incorporated. El dispositivo, cuyo voltaje en directa es de 1 V a 1 A, es capaz de soportar un voltaje pico repetitivo en inversa de 100 V y un voltaje RMS en inversa de 70 V. Por otro lado, el diodo puede operar bajo una corriente media máxima de 1 A, bajo una temperatura de 75°C.

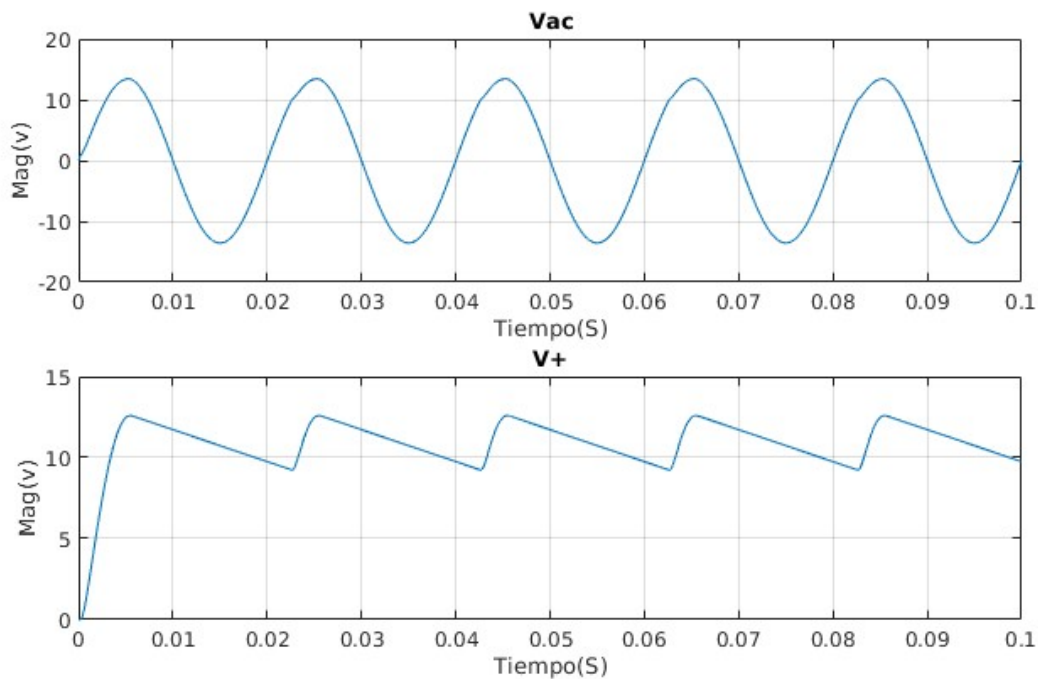


Figura 3.46: Simulación de la fuente lineal bajo corriente de carga constante

En la figura 3.46 se puede observar la verificación del diseño mediante simulación. En el primer gráfico se presenta la señal de entrada, mientras que en el segundo se puede ver la salida rectificadora y filtrada. La simulación se lleva a cabo para el peor caso de diseño, es decir: capacidad de $2560 \mu\text{F}$ (80% del valor nominal), tensión de entrada de $9,6 \text{ V}_{\text{RMS}}$ (80% del valor nominal) y corriente de salida de $0,5 \text{ A}$ (máxima carga). El valor de ripple pico a pico obtenido es de $3,4 \text{ V}$, centrado en un valor medio de $10,9 \text{ V}$. El valor mínimo a la salida es de $9,2 \text{ V}$, por lo que se cumplen las especificaciones de diseño.

3.10 Diseño de PCBs

En base al diseño teórico y elección de componentes se realiza el layout físico de ambas placas, fuente de alimentación y etapa de acondicionamiento. Se hace uso del software libre Kicad, dado que la complejidad del PCB no amerita el uso de una herramienta más compleja y de pago, como por ejemplo, Altium.

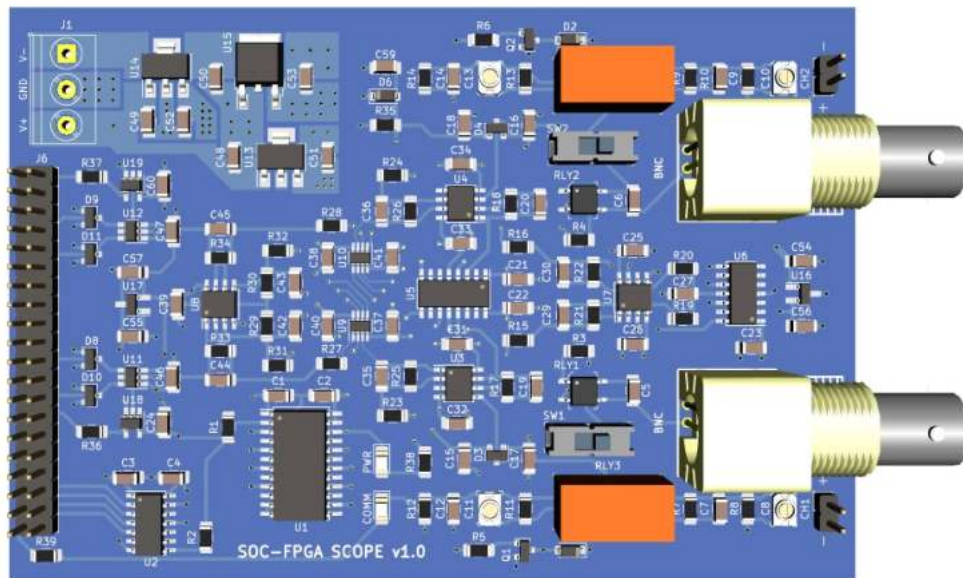


Figura 3.47: PCB principal vista superior

En la 3.47 se presenta el diseño de la placa principal, en su vista 3D superior, cuyo tamaño físico es de 12 cm x 8 cm. En la parte superior izquierda se encuentra el conector de alimentación (V_+ , V_- y GND) y los reguladores lineales. La entrada de cada canal cuenta con un conector principal tipo BNC y un conector secundario del tipo cabecera de pines. En la esquina inferior izquierda, se ubica el conector de 40 pines de la interfaz y los convertidores de niveles de tensión.

Ambos canales cuentan con un layout similar, las diversas etapas se disponen de derecha a izquierda, es decir, la entrada a la derecha y los ADC a la izquierda, en las proximidades del conector de interfaz a la FPGA. El trazado en paralelo de los canales, permite ubicar entre ambos los componentes que son compartidos. Por ejemplo, la etapa de generación de offset se encuentra entre los conectores BNC de entrada y en el centro de la placa se puede ver el potenciómetro digital dual.

Teniendo en cuenta la tecnología disponible hoy en día, y los bajos tiempos y costos de fabricación de prototipos, el diseño se implementa en un PCB de cuatro capas. Cabe destacar que no resulta imposible implementar el PCB en dos capas, sin embargo, es una tarea mucho más compleja. El uso de dos capas, no permite contar con un plano de masa libre de pistas, deteriorando la integridad de las señales y las emisiones electromagnéticas.

Las diversas señales que conforman el diseño del front-end son separadas, según su naturaleza, en cuatro grupos:

- Analógicas: Señales relacionadas a la forma de onda adquirida, que conforman las diversas etapas de acondicionamiento. Representan la mayoría de las conexiones en la placa y son susceptibles a interferencias y ruido.
- Digitales (control): Señales provenientes de la FPGA que actúan sobre los relés y amplificadores de ganancia programable, así como también los buses SPI del potenciómetro digital y DAC. Se trata de líneas digitales de baja frecuencia, por lo que su ruteo no es tan crítico. Este grupo de señales se encuentra activo en la etapa de

configuración del front-end, entre dos adquisiciones, por lo que no produce interferencias en las etapas analógicas durante el periodo de adquisición.

- Digitales (datos ADC): Se refiere al bus SPI de 20 MHz de cada convertor analógico digital. Estas pistas deben ser lo más cortas posibles, por lo que los componentes involucrados se ubican en las cercanías del conector a la FPGA. Estas señales se encuentran activas durante una adquisición por lo que se deben separar, en lo posible, de las etapas analógicas, evitando posibles interferencias.
- Alimentación: Esta categoría abarca los diferentes voltajes de alimentación requeridos por los componentes, siendo los principales ± 5 V y 3,3 V.

El ruteo de las señales analógicas se realiza en la capa superior, disponiendo los componentes de forma de minimizar la longitud de las pistas. En esta capa se ubican también los buses SPI de los ADC, para lo cual se siguen las recomendaciones de layout del fabricante [36]. La segunda capa, inmediatamente debajo de la superior, está conformada por un plano de masa, destinado a reducir lazos de corriente y mejorar la integridad de señales [54]. Cabe destacar que esta capa no cuenta con ninguna pista de otra índole.

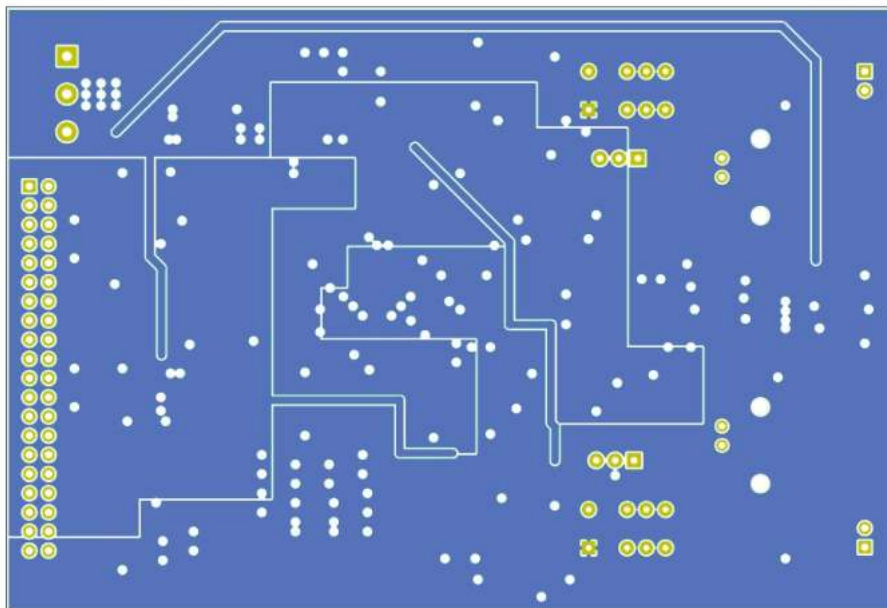


Figura 3.48: PCB principal tercera capa (alimentación)

La tercera capa es utilizada para la alimentación de los diversos elementos que componen el diseño del front-end. En este caso, se emplean tres polígonos de cobre diferentes, uno para cada tensión de alimentación (+5 V, -5 V y 3,3 V). El uso de zonas de cobre, a diferencia del trazado de pistas, permite aumentar la capacidad de corriente y aprovechar la disponibilidad de área en la capa [54].

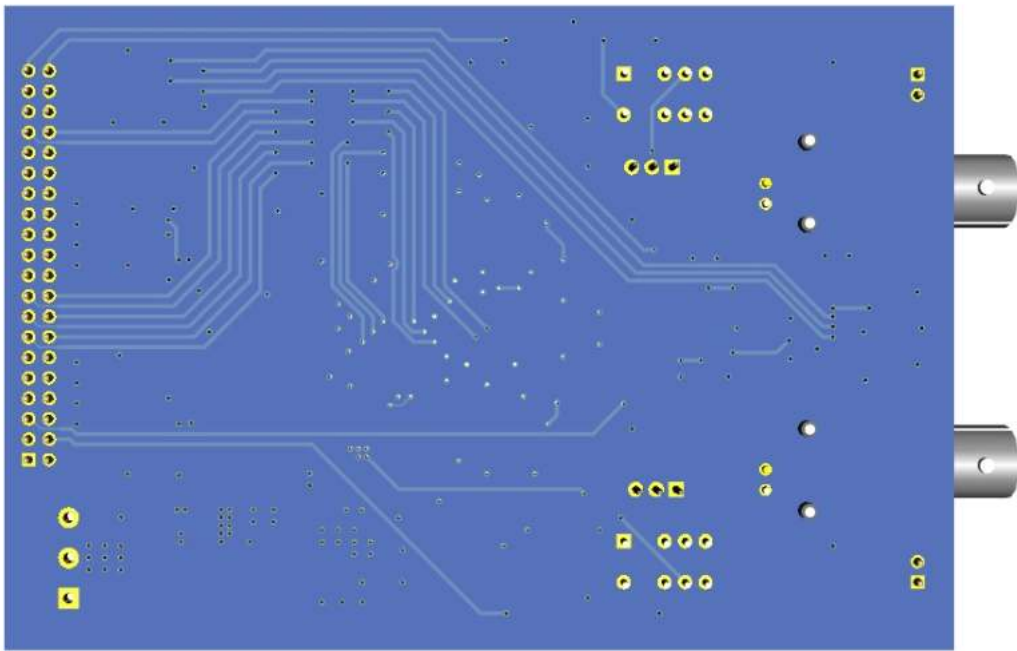


Figura 3.49: PCB principal vista inferior (pistas digitales)

En la última capa, se dispone el trazado de las líneas de control digitales provenientes de la FPGA. Dado que estas señales actúan por sobre componentes distribuidos a lo largo de toda la placa, la utilización de una capa diferente facilita el ruteo. Debido a la baja frecuencia y transporte de corriente requerido, la elevada longitud de las pistas no es un factor crítico.

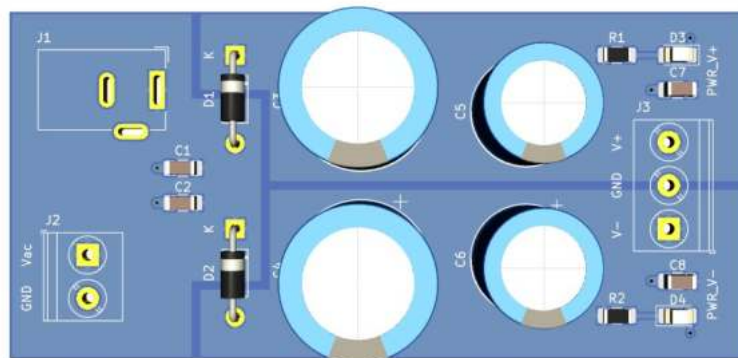


Figura 3.50: Fuente de alimentación vista superior

El diseño de la fuente lineal de alimentación consiste en un PCB de dos capas (8,5 cm x 4 cm). En este caso, el transporte de corriente es realizado en la capa superior, mediante el empleo de polígonos de cobre. Por otro lado, la capa inferior consiste en un plano de masa completo. La conexión entre la fuente y el transformador externo es realizada mediante un conector del tipo “barrel jack”. La conexión al PCB principal se realiza mediante un conector tipo “screw terminal” de tres contactos (V+, V- y GND). Adicionalmente se agrega a la entrada un conector adicional, de este mismo tipo, aumentando la flexibilidad del diseño.

Capítulo 4

Arquitectura de adquisición

El objetivo de este capítulo es describir los diversos componentes implementados en el sector de lógica programable del SoC FPGA, involucrados en la arquitectura de adquisición. Por otro lado, los elementos de software que conforman el sistema son analizados en el capítulo 5. Resulta pertinente aclarar que el diseño de la arquitectura de adquisición, el software de servidor y el cliente de visualización fue realizado de manera paralela en una primera instancia del proyecto. En una segunda etapa, se procedió a diseñar el front-end analógico e integrarlo con el resto del sistema.

La lógica implementada cuenta con diversas funcionalidades asociadas al control de flujo de la adquisición, el procesamiento de los datos y la comunicación con el front-end. Las muestras de ambos canales, provenientes de los ADC, son almacenadas de manera secuencial en un buffer circular implementado en la memoria RAM principal, compartida con el sistema operativo. El proceso de adquisición es controlado en base a diversos parámetros, destacándose los ajustes relacionados al disparo y la base de tiempo.

Adicionalmente, se implementa en la lógica programable una etapa de procesamiento, cuyo objetivo principal es reducir la cantidad de muestras mediante la aplicación de diversos algoritmos de decimación [55]. De esta forma, se brinda la posibilidad de ajustar la profundidad de memoria, es decir, la cantidad de muestras enviadas a través de internet y graficadas. Finalmente, con respecto a la comunicación con el front-end, la lógica implementada permite configurar las diferentes etapas analógicas, ajustando parámetros tales como la selección de atenuación, ganancia programable, offset, etc.

Este capítulo consta de cuatro secciones principales, en el apartado 4.1 se describen las herramientas utilizadas en el diseño, integración y verificación de la lógica programable. En el apartado 4.2 se presenta un diagrama en bloques del diseño propuesto y se describe brevemente el proceso de adquisición. Por otro lado, en la sección 4.3 se profundiza en el detalle de implementación de cada etapa involucrada en la adquisición y procesamiento. Finalmente, el último apartado es dedicado a la síntesis e implementación del diseño en la FPGA, brindando detalles sobre la utilización de los recursos del dispositivo y sobre el análisis temporal.

4.1 Metodología de diseño

El objetivo de esta sección es presentar las herramientas y metodologías utilizadas en el diseño y verificación de la arquitectura de adquisición. Dado que esta instancia de desarrollo ocurre de manera previa al diseño del front-end, se utiliza como fuente de datos un módulo ADC dual del fabricante Digilent.

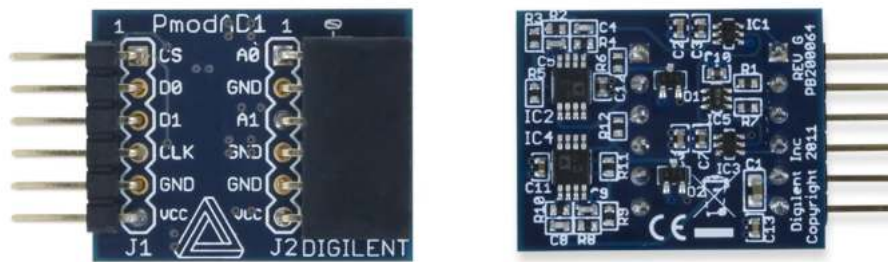


Figura 4.1: Módulo ADC PMOD AD1 (Digilent)

El modelo seleccionado, PMOD AD1 [53], se alimenta en base a 3,3 V y cuenta con dos ADC, diodos de protección, buffer de entrada y buffer de las líneas de datos. El convertor analógico digital empleado en el módulo es el modelo ADC7476A [56] de Analog Devices, funcionalmente equivalente al utilizado en el diseño del front-end, ADCS7476 de Texas Instruments. La única diferencia entre ambos ADC es el encapsulado, ya que el primero únicamente se encuentra disponible en un tamaño máximo de 2,2 mm x 1,35 mm.

Con el objetivo de simular la presencia del front-end, el módulo PMOD se conecta directamente a la FPGA, lo que permite contar con señales de entrada para ambos canales en el rango de 0 V a 3,3 V. De esta forma, se valida el diseño de la arquitectura completa de adquisición y el software involucrado, previo a la fabricación de la etapa de acondicionamiento analógica.

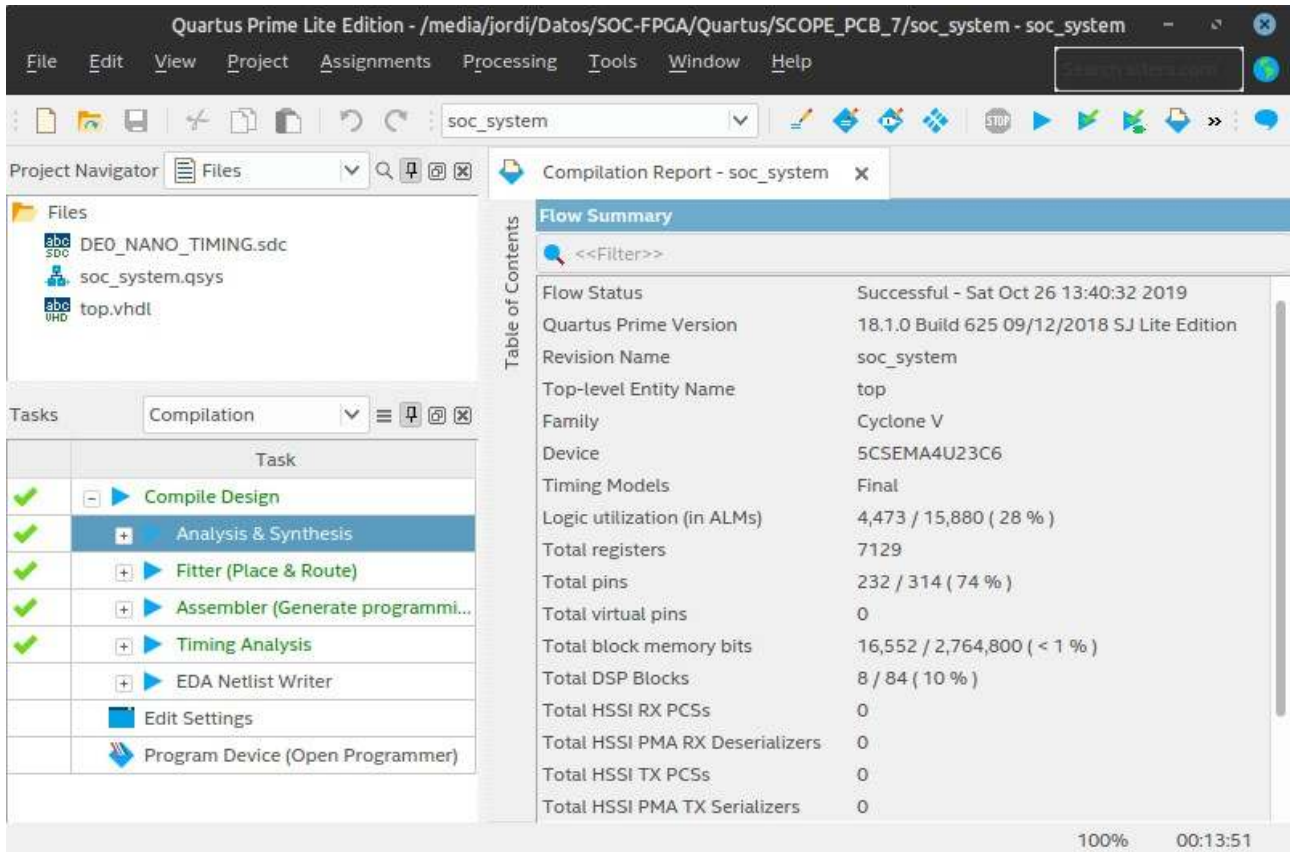


Figura 4.2: Software Quartus Prime Lite 18.1 de Intel FPGA

El diseño y síntesis de los bloques de lógica es realizado principalmente en el software Quartus Prime del fabricante de la FPGA, en su versión gratuita. En los casos que resulta posible, se utilizan componentes ya disponibles en la librería de Quartus. Por otro lado, los bloques involucrados en el procesamiento y control de adquisición, son diseñados y sintetizados en el software Simulink. Finalmente, algunos elementos de propósito específico se implementan directamente en el lenguaje de descripción de hardware VHDL.

4.1.1 Integración del sistema

La integración de los diferentes componentes de la arquitectura de adquisición se realiza mediante la herramienta Platform Designer [57] (previamente Qsys) en el software Quartus Prime. Mediante el uso de Platform Designer, se crean, configuran e interconectan los diversos bloques a ser sintetizados en la FPGA. De esta forma, se acelera el proceso de diseño y se reduce la posibilidad de errores.

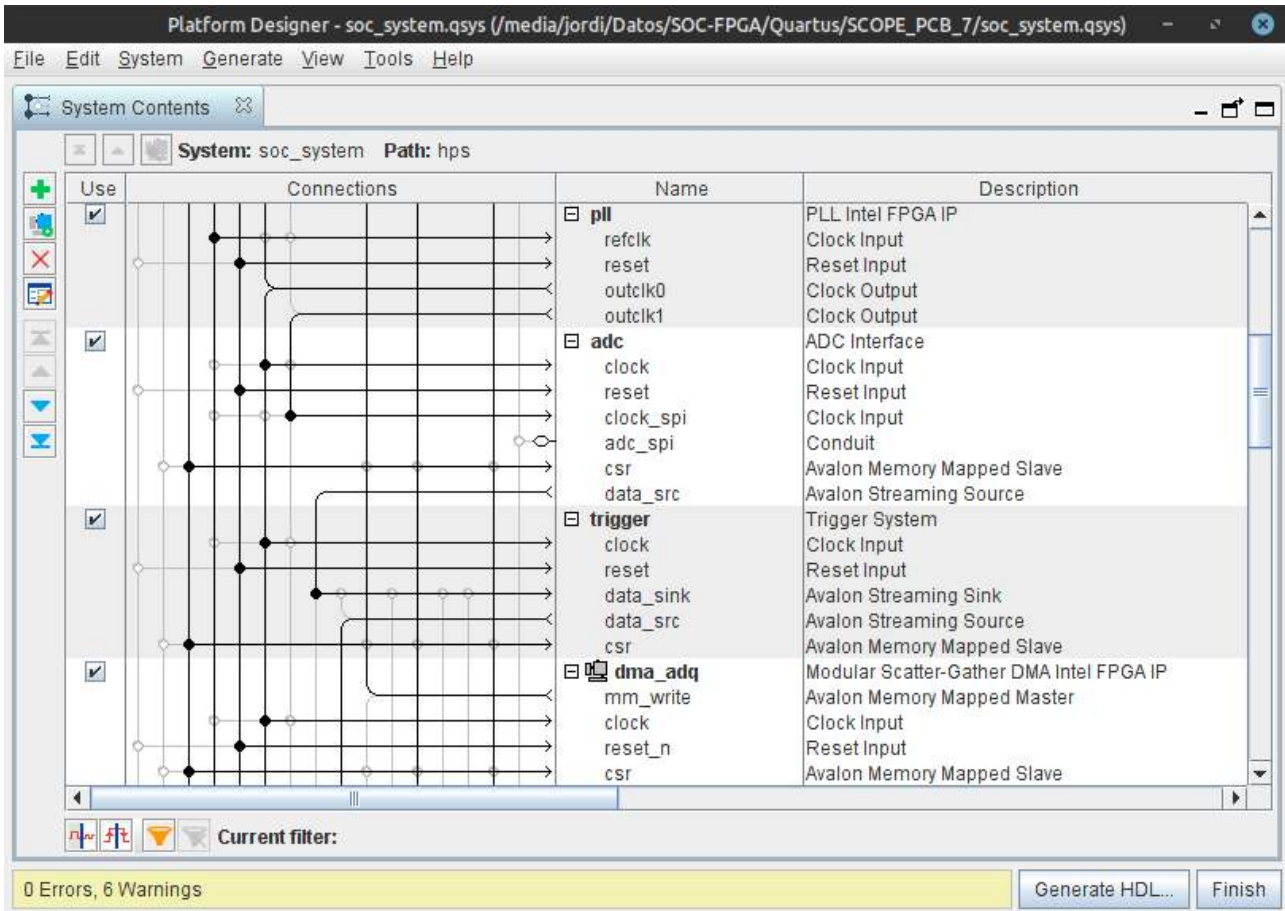


Figura 4.3: Platform Designer (Quartus Prime Lite)

Así mismo, Platform Designer permite configurar el sistema de procesamiento (HPS), habilitando los diversos puentes de comunicación y realizando su configuración. La conexión entre los bloques implementados en la FPGA y el HPS permite el acceso desde la lógica programable a la memoria RAM y los periféricos del procesador. Así mismo, el uso de los diversos puentes, analizados en el capítulo 2, permite al procesador acceder a la lógica programable. El fabricante de la FPGA pone a disposición en Platform Designer varios componentes ya diseñados [58], entre los que destacan:

- Memorias del tipo RAM, ROM y FIFO.
- Periféricos de comunicación SPI, I2C, UART, etc.
- Puertos de entrada y salida GPIO.
- Motores de acceso directo a memoria (DMA).
- Bloques de procesamiento digital de señales (DSP), procesamiento de imágenes, etc.
- Procesadores sintetizados en la FPGA (Nios II).

Adicionalmente, la herramienta brinda la posibilidad de generar componentes personalizados en base a su descripción en Verilog o VHDL, definiendo todas sus interfaces de conexión e integrándolos con el resto del sistema. Con respecto a la lógica de interconexión, existen dos

protocolos de comunicación interna entre componentes: Avalon Streaming (Avalon-ST) y Avalon Memory Mapped (Avalon-MM) [59].

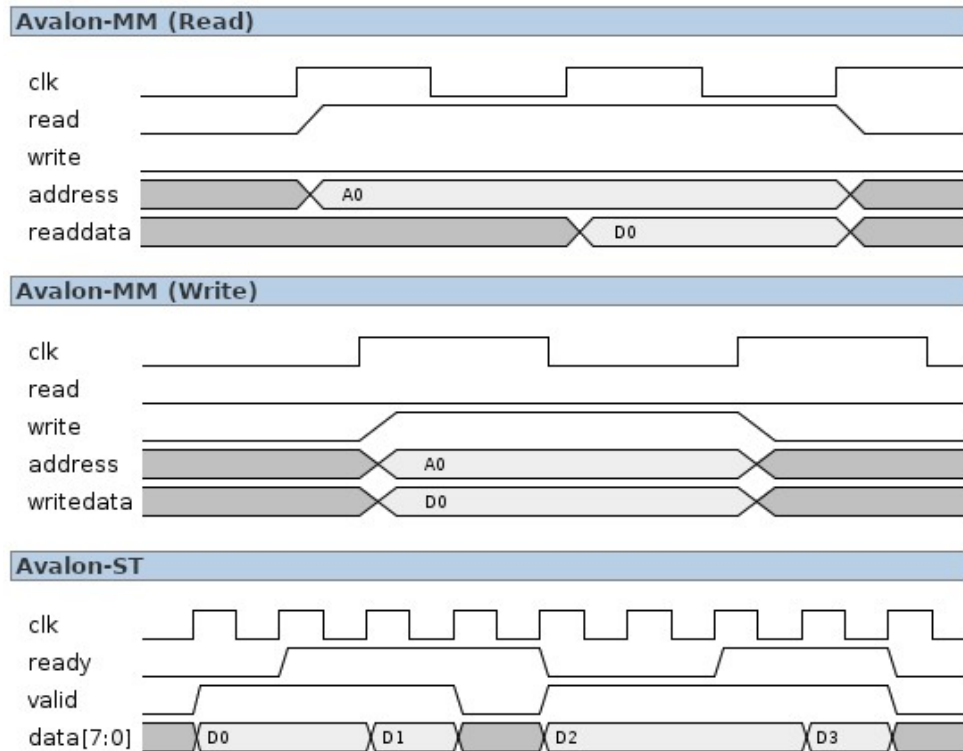


Figura 4.4: Protocolos Avalon-MM y Avalon-ST

El protocolo Avalon-ST permite la conexión directa entre la salida de datos de un bloque y la entrada del siguiente, es decir, se trata de una conexión punto a punto. Esto resulta apropiado para el caso de streaming de datos entre bloques de procesamiento en base a un reloj. El bus puede ser implementado con diferentes capacidades y parámetros de configuración. De esta forma, puede ser adaptado fácilmente a diversas aplicaciones específicas. Así mismo, el software genera la lógica necesaria para adaptar la comunicación entre dos componentes cuyas interfaces Avalon-ST difieran en alguna característica.

El protocolo Avalon-MM implementa un bus de datos en el que existen maestros y esclavos. En dicha topología, los maestros inician las transacciones de lectura o escritura y los esclavos responden. Cada esclavo expone una determinada cantidad de registros en el bus, que son seleccionados en bases a direcciones de memoria. Platform Designer se encarga de crear la matriz de interconexión entre los diferentes componentes presentes en el bus, informando la vista de memoria que tiene cada maestro por sobre los esclavos que tiene conectado. Este bus resulta apropiado para la conexión de un procesador (o bloque DMA) con diversos periféricos y sectores de memoria.

Nuevamente, al igual que para el caso del protocolo Avalon-ST, cada componente puede ser configurado con diferentes capacidades en el bus. A modo de ejemplo, se puede ajustar el ancho de palabra, ancho de direcciones, latencias de lectura y escritura, etc. Durante la síntesis, el

software genera automáticamente la lógica necesaria para la adaptación entre los diferentes componentes. Adicionalmente, en caso de que existan bloques funcionando bajo relojes diferentes, se genera la lógica necesaria para realizar el cruce de dominio de reloj de manera segura. Esto resulta de importancia para evitar fenómenos de metaestabilidad en los registros internos de la FPGA [60].

4.1.2 Síntesis de alto nivel

Como fue mencionado en el apartado anterior, en la implementación del sistema se utilizan varios componentes prediseñados y presentes en Platform Designer. Para el caso de los bloques específicos, involucrados en el procesamiento de datos y el control de flujo, se recurre a herramientas de síntesis de alto nivel (HLS). Esto permite acelerar el proceso de diseño, verificación e implementación, con respecto a un diseño en VHDL o Verilog.

El paquete de software elegido es Simulink [61], el cual permite diseñar algoritmos de forma gráfica, en base a un diagrama en bloques de las operaciones realizadas. Así mismo, Simulink cuenta con una herramienta denominada HDL Coder [62], que permite sintetizar código VHDL (o Verilog) en base al diseño realizado.

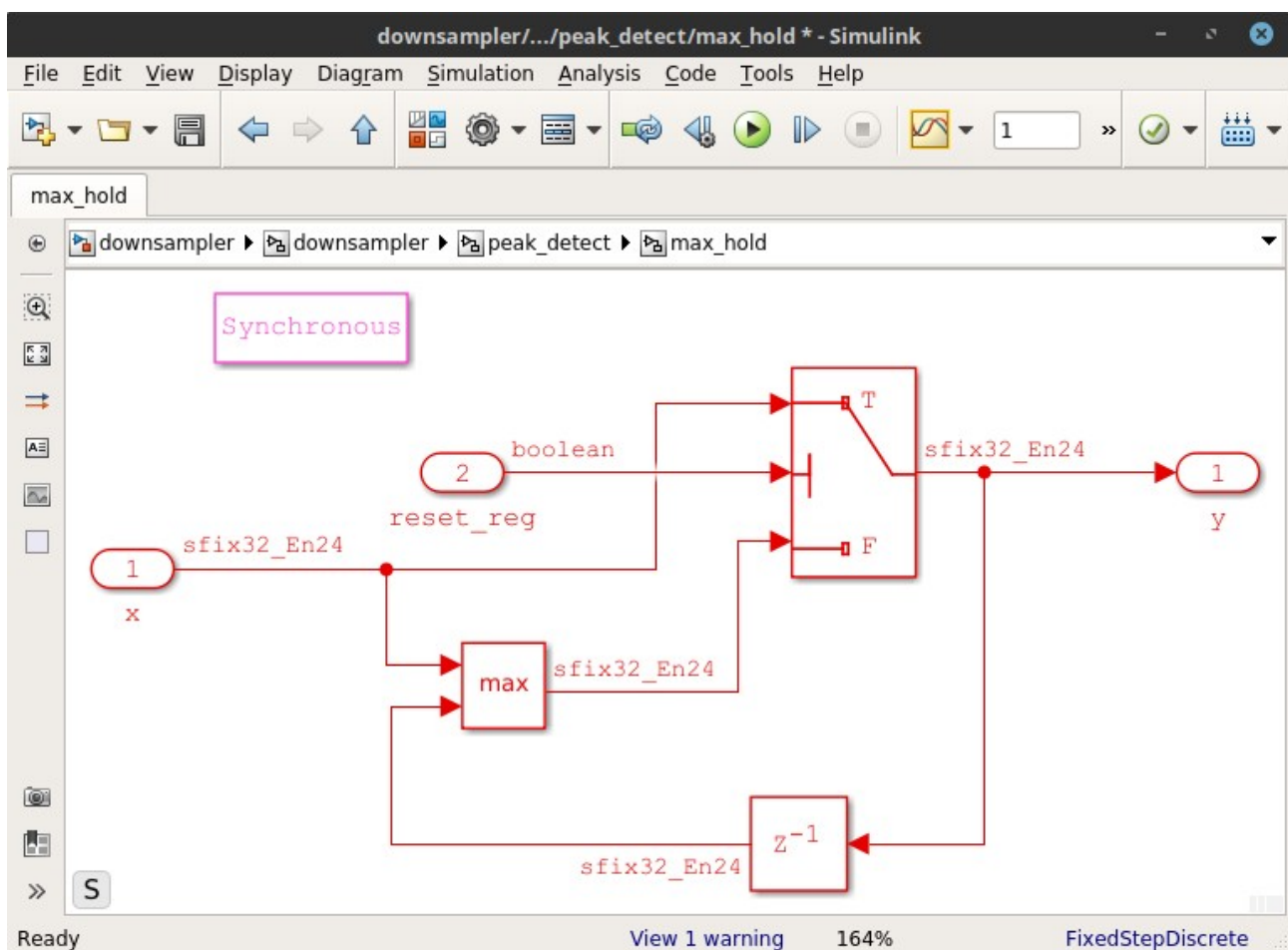


Figura 4.5: Diseño mediante diagrama en bloques (Simulink)

Una ventaja de la utilización de MATLAB, es la rápida implementación de los algoritmos con aritmética de punto fijo. El software cuenta con herramientas que permiten determinar el tamaño de la parte entera y decimal de cada señal, de acuerdo al rango y la precisión requerida. Así mismo, es capaz de propagar automáticamente los tipos de datos, en base a las operaciones realizadas. Finalmente, permite verificar el comportamiento de la implementación en punto fijo mediante simulación, evaluando la concordancia con el modelo original.

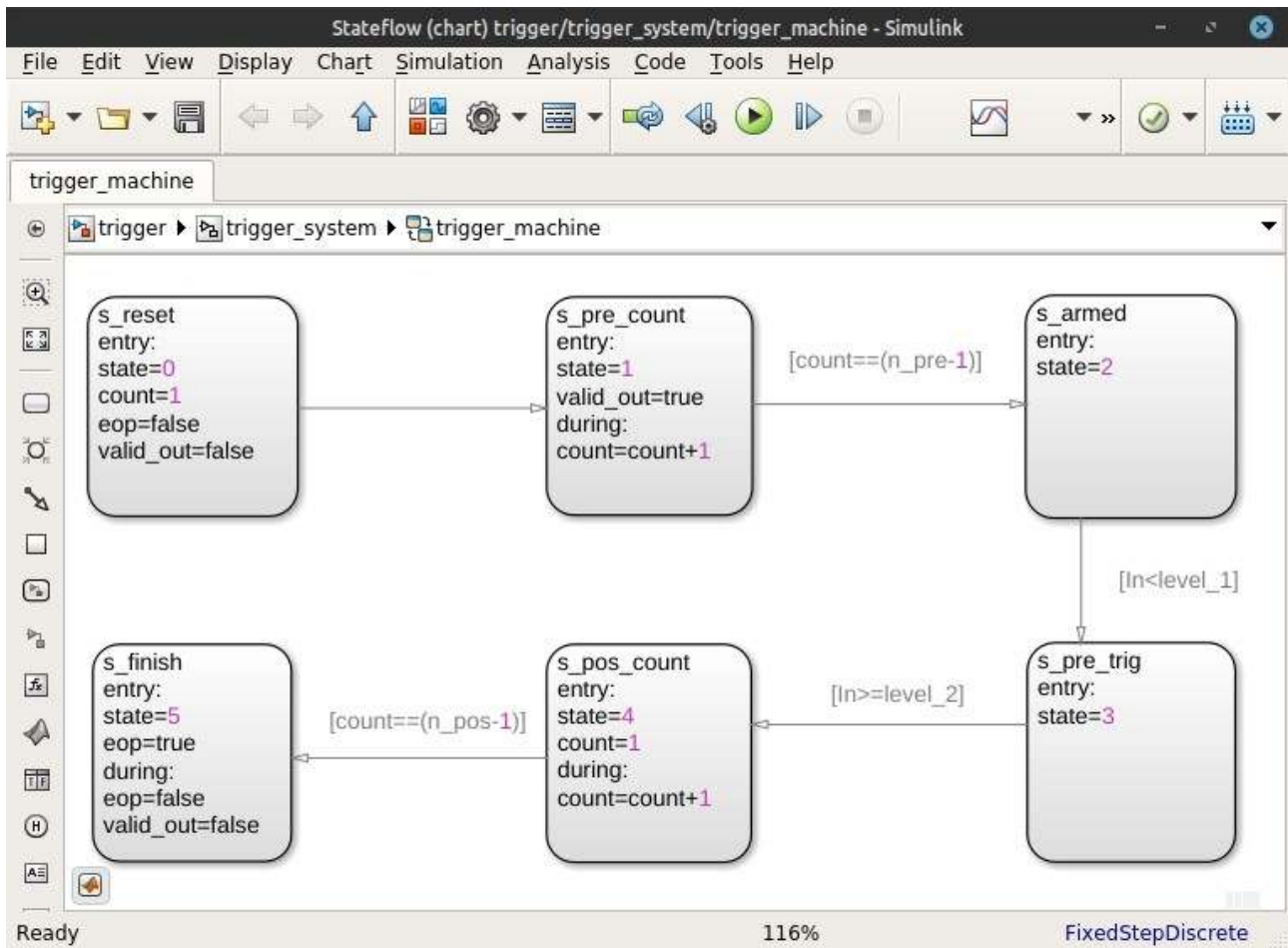


Figura 4.6: Diseño de máquinas de estados en Stateflow (Simulink)

Para los algoritmos que requieren control de flujo, se diseñaron máquinas de estado con el paquete Stateflow [63] de Simulink. Nuevamente, esto agiliza el proceso de desarrollo, ya que permite simular las transiciones entre los diferentes estados y la interacción de la máquina con el resto del sistema.

Con respecto a la verificación y validación, HDL Coder genera un modelo equivalente de Simulink, en el que se introducen los diversos registros, latencias y optimizaciones realizadas por sobre el modelo original. De esta forma, es posible visualizar gráficamente el diseño sintetizado y verificar automáticamente su concordancia numérica con el modelo original. Adicionalmente, el software es

capaz de realizar simulaciones funcionales del diseño RTL, utilizando la herramienta Modelsim de Mentor Graphics. En este caso, HDL Coder genera las señales de estímulo o testbench en base al diagrama en Simulink y compara automáticamente las salidas del simulador con los valores numéricos esperados.

Finalmente, cabe aclarar que existen en el mercado otras opciones de síntesis de alto nivel (HLS), orientadas a la generación automática de un diseño RTL en base a un algoritmo en C o C++ [64]. Sin embargo, se opta por la utilización de MATLAB debido a una mayor afinidad con el flujo de trabajo y los beneficios que otorgan las diversas funcionalidades disponibles.

4.1.3 Verificación de lógica programable

Para el caso de los componentes desarrollados directamente en VHDL, resulta de utilidad realizar simulaciones funcionales del diseño RTL [65]. Para la simulación, se recurre al paquete Modelsim, presente en el software Quartus. El uso de Modelsim permite verificar el correcto comportamiento de la lógica diseñada, previo a su integración con el resto del sistema mediante Platform Designer.

A modo de ejemplo, se presenta una captura de la interfaz SPI de los convertidores analógico digital del front-end. En este caso, el objetivo de la simulación es verificar las formas requeridas para la comunicación en el bus SPI y que los datos sean capturados correctamente en la lógica programable.

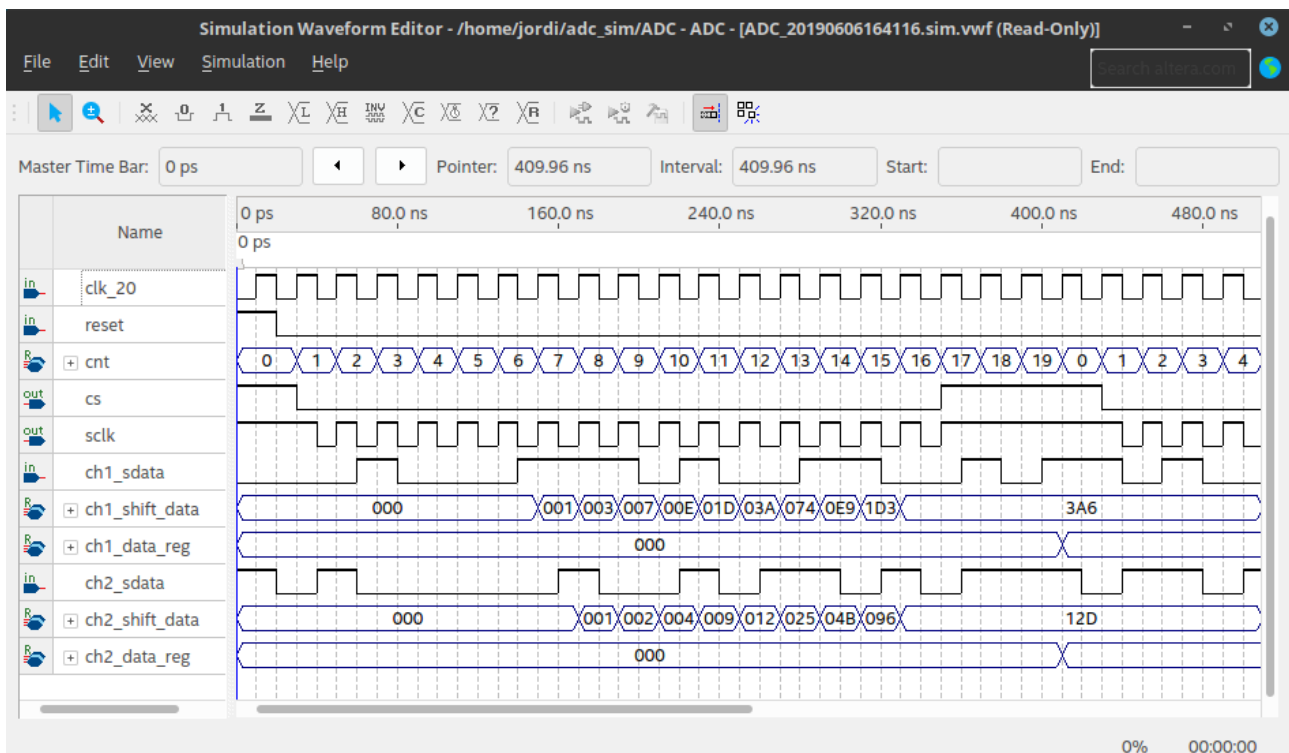


Figura 4.7: Simulación funcional en Modelsim de código VHDL (Quartus)

Por otro lado, en algunos casos resulta de utilidad ver las señales digitales internas de la FPGA para el diseño implementado y funcionando. El software de síntesis Quartus cuenta con una herramienta denominada SignalTap [66], la cual es capaz de generar la lógica necesaria para implementar un analizador lógico directamente en la FPGA. De este modo, es posible seleccionar las señales que se quieren visualizar y las condiciones de disparo, observando el flujo de datos en tiempo real. La utilización de esta herramienta durante la fase de desarrollo permite detectar errores que se presentan bajo condiciones reales, difícilmente detectables en un entorno de simulación.

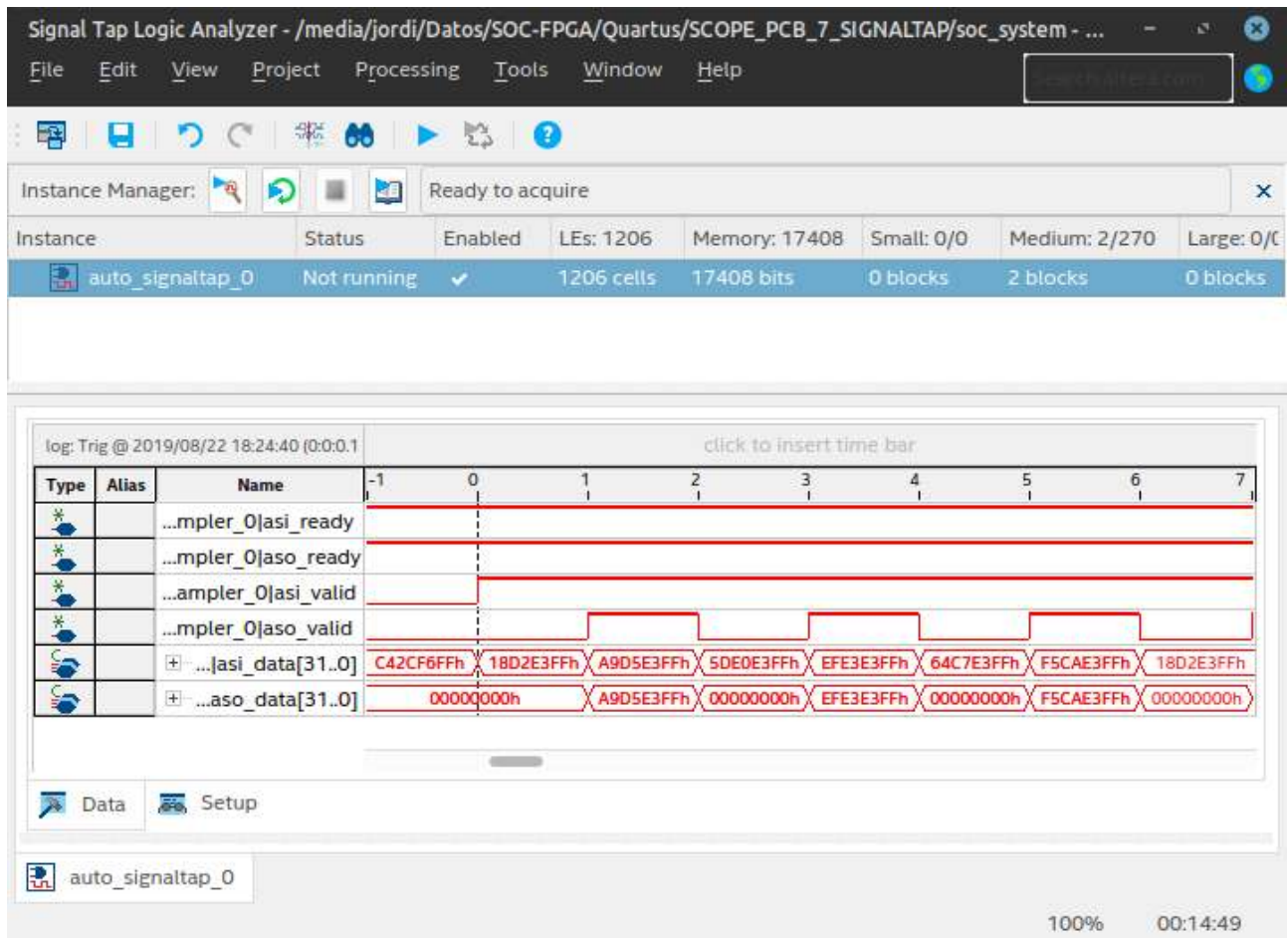


Figura 4.8: Analizador lógico SignalTap mostrando buses Avalon-ST (Quartus)

A modo de ejemplo, en la figura 4.8 se muestra una captura realizada en tiempo real de dos buses de tipo Avalon-ST, involucrados en el procesamiento de los datos adquiridos. Como se observa, la herramienta permite analizar los datos presentes en el bus y las diversas señales de control.

4.1.4 Análisis temporal

Finalmente, la última herramienta empleada durante el proceso de diseño, síntesis y verificación de la lógica programable es Timing Analyzer [67], que forma parte de Quartus. Mediante esta

utilidad es posible realizar un análisis automático de los tiempos de propagación entre los registros internos, asociados a la lógica combinacional implementada. Esto permite detectar caminos de propagación de la señal que no cumplen con los requerimientos de tiempo de setup y tiempo de hold de los registros internos. Este tipo de problemas genera corrupción de datos y fenómenos de metaestabilidad que afectan severamente el funcionamiento de la lógica programable [68].

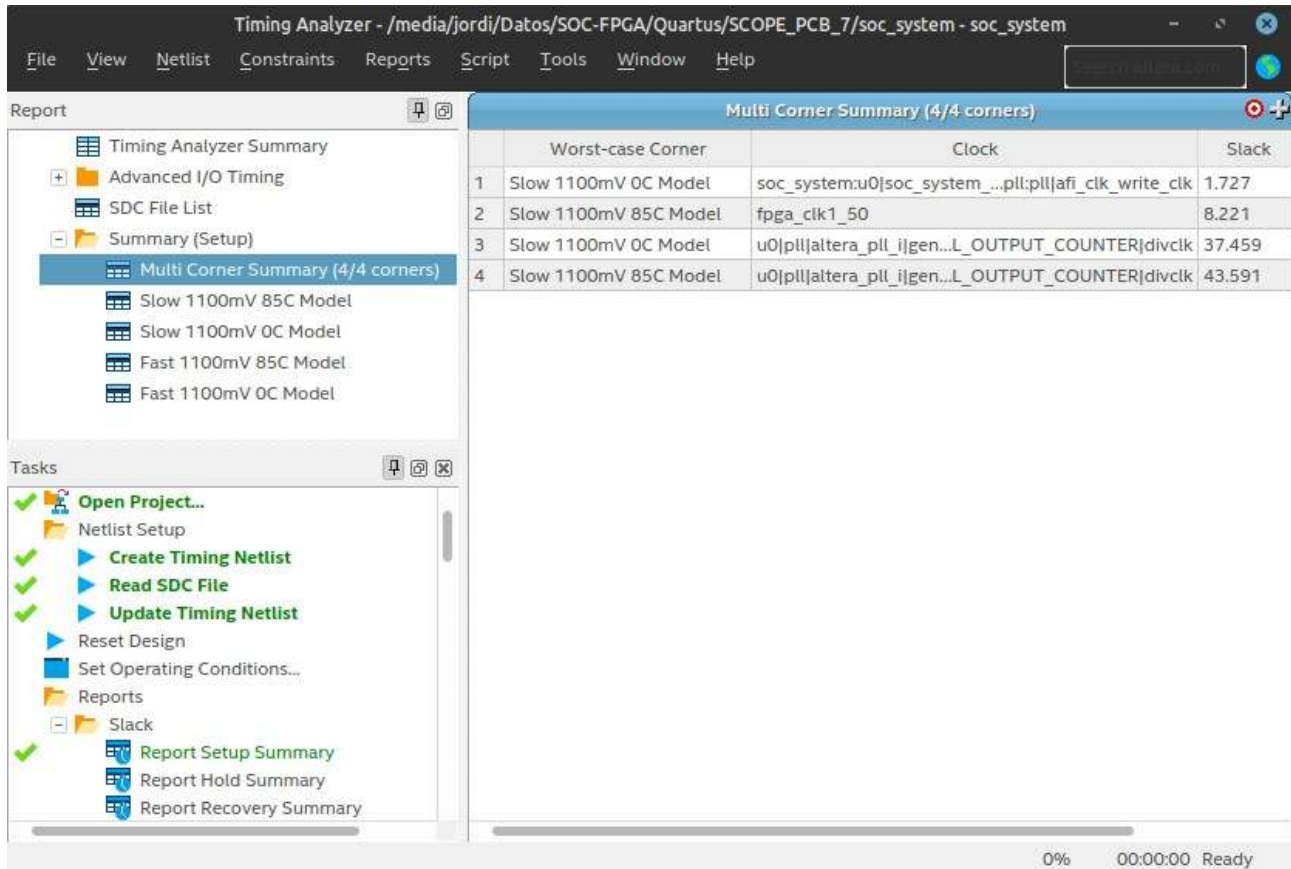


Figura 4.9: Herramienta de análisis temporal Timing Analyzer (Quartus)

Debido a su naturaleza, un problema de violación de tiempos es imposible de detectar en una simulación funcional y difícilmente detectable mediante el empleo del analizador lógico SignalTap. Teniendo en cuenta lo mencionado, el uso de la herramienta resulta de gran utilidad para optimizar el diseño y asegurar el correcto funcionamiento a la frecuencia deseada. Por otro lado, la herramienta permite hacer un estimativo de la máxima frecuencia de operación para cada uno de los relojes del sistema. Este dato resulta de utilidad a la hora de evaluar un aumento de la frecuencia de muestreo y procesamiento del sistema de adquisición en un posible trabajo futuro.

4.2 Descripción de la arquitectura

El objetivo de este apartado es brindar una descripción general del diseño para la arquitectura de adquisición. En primer lugar, se presenta un diagrama en bloques con los componentes principales que forman parte del diseño. Luego, se realiza una descripción del rol de cada bloque

en las diferentes etapas del proceso de adquisición. Por otro lado, los detalles técnicos y de implementación son analizados en la sección 4.3 en mayor profundidad.

4.2.1 Diagrama en bloques

En el diagrama de la figura 4.10, se observan los componentes principales que forman parte del sistema. Los bloques cuyo color de fondo es gris claro, son implementados en el sector de lógica programable del dispositivo. Por otro lado, se presentan en color blanco los dos componentes del HPS que resultan relevantes a la arquitectura, procesador y memoria RAM.

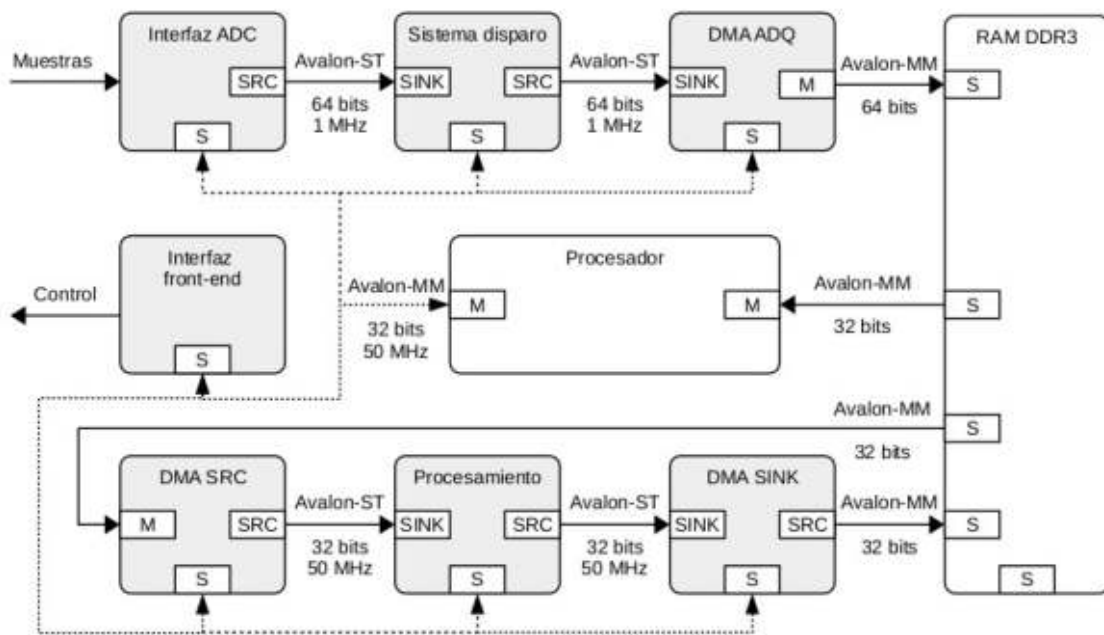


Figura 4.10: Diagrama en bloques de arquitectura de adquisición

Las conexiones entre los componentes se representan con flechas, cuyo sentido indica la dirección del flujo de datos. Adicionalmente, se especifica el tipo de interfaz, Avalon-ST o Avalon-MM, la frecuencia de operación del bus y el ancho de palabra utilizado. Cabe destacar que se hace uso de una línea de trazo sólido para las conexiones involucradas en las transferencias de los datos adquiridos y una línea de tipo punteada para las interfaces de control.

En el caso de los buses Avalon-MM, se indican los componentes que actúan como maestros y los que actúan como esclavos con las letras M y S respectivamente. Por otro lado, en las interfaces de tipo Avalon-ST, se recurre a las abreviaciones SRC y SINK para especificar el rol de cada componente en la interfaz.

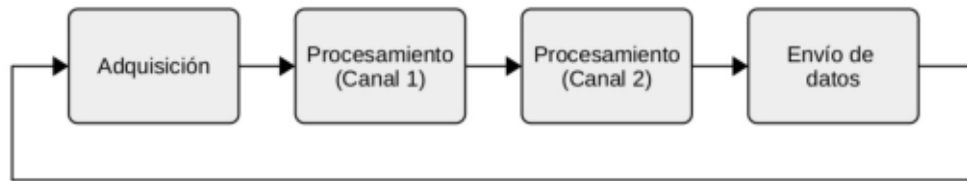


Figura 4.11: Eventos que forman parte de un ciclo de adquisición

Un ciclo de funcionamiento completo del sistema consiste de tres fases secuenciales: adquisición, procesamiento y envío de datos. Debido al tipo de arquitectura, entre dos adquisiciones consecutivas existe un retardo asociado al procesamiento y envío de las muestras. Durante este periodo, el sistema no adquiere nuevos datos, por lo que existe cierto grado de pérdida de información [69].

Los bloques involucrados en la adquisición inicial son la interfaz del ADC, el sistema de disparo y el motor de acceso directo a memoria de adquisición (DMA ADQ). Estos componentes pueden ser observados en la parte superior del diagrama en bloques principal de la figura 4.10. Luego de la adquisición inicial, los datos en memoria son leídos y procesados en la lógica programable de manera individual para cada uno de los canales. Durante esta fase del ciclo, los elementos involucrados son los siguientes: DMA SRC, bloque de procesamiento y DMA SINK, ubicados en la parte inferior de la figura 4.10. Finalmente, los datos procesados son leídos de la memoria RAM por el procesador y enviados a través de internet al software cliente.

De manera adicional, existe un bloque implementado en la lógica programable cuya funcionalidad es controlar y configurar los diferentes elementos del front-end analógico. Este componente se observa a la izquierda del diagrama principal de la figura 4.10 y no se encuentra relacionado directamente al ciclo de adquisición y procesamiento de datos.

4.2.2 Adquisición

En este apartado se realiza una descripción básica del ciclo de adquisición inicial y los requerimientos asociados a esta etapa. A continuación, se presenta un detalle de las funcionalidades de cada uno de los tres componentes involucrados:

- **Interfaz del ADC:** Se encarga de generar las señales de control del bus SPI de los convertidores de ambos canales, necesarias para realizar el muestreo a 1 MSa/s. Los datos de 12 bits provenientes del ADC son escalados y modificados de manera que representen el valor de voltaje a la entrada del front-end, en un formato de punto fijo de 32 bits. Las muestras de ambos canales son concatenadas en una interfaz Avalon-ST de 64 bits a la velocidad de muestreo, es decir, 1 MHz.
- **DMA de adquisición:** Almacena de manera secuencial los datos de ambos canales en un buffer circular en memoria RAM, cuyo tamaño depende de la base de tiempo seleccionada. El llenado del buffer se repite de manera cíclica a la espera de un evento de disparo. Este bloque recibe los datos a través de una interfaz de tipo streaming y cuenta a su salida con

una conexión directa al controlador SDRAM, en la que actúa como Avalon-MM maestro (64 bits 1 MHz).

- Lógica de disparo: Analiza el flujo de datos provenientes de la interfaz del ADC, en tiempo real, en búsqueda de un evento de disparo exitoso, de acuerdo a la configuración realizada en el software. La cantidad de muestras almacenadas antes y después del disparo dependen del ajuste de posición horizontal y la base de tiempo. En base a estos parámetros, el sistema de disparo detiene, en el momento adecuado, el flujo de datos hacia el motor DMA de adquisición. Cabe destacar que este componente no modifica los datos provenientes de la interfaz del ADC, solamente actúa sobre las señales de control del bus de salida a fin de controlar el motor DMA de adquisición.

4.2.3 Procesamiento

Luego de la adquisición inicial, se realiza el procesamiento y transferencia de los datos. Este proceso ocurre entre dos adquisiciones consecutivas y se repite para cada canal habilitado. Existen cuatro funcionalidades principales asociadas a esta etapa:

- Separación de canales: Como resultado de la adquisición inicial, las muestras de ambos canales, pertenecientes a una base de tiempo completa, quedan almacenadas en el buffer circular de manera intercalada, en bloques de 64 bits (32 bits por canal). Durante la etapa de procesamiento, los canales son leídos y transferidos a sectores de memoria separados.
- Eliminación de offset: Debido a que no existe correlación entre el evento de disparo externo y la secuencia de llenado del buffer circular, el proceso de adquisición se detiene en una posición de memoria arbitraria. Como consecuencia de esta situación, existe un offset en la ubicación de la primera muestra a nivel temporal, en relación a la primera posición del buffer circular. Durante la fase de procesamiento, este offset es tenido en cuenta con el objetivo de transferir a un nuevo sector de memoria la base de tiempo completa en el orden temporal correcto.
- Ajuste de profundidad de memoria: El proceso de adquisición inicial ocurre a la frecuencia de muestreo nominal del sistema, es decir 1 MSa/s. Se incorpora en el sistema un bloque de procesamiento, encargado de realizar la decimación de los datos. De este modo, se brinda la posibilidad de reducir la cantidad de muestras correspondientes a un ciclo de adquisición completo. Esto permite disminuir el ancho de banda de los datos transferidos a través de internet y las exigencias del cliente de visualización, mejorando el tiempo de respuesta y velocidad del sistema.
- Procesamiento de los datos: Finalmente, esta etapa brinda la posibilidad de aplicar procesamiento digital de señales acelerado por hardware, es decir, realizado en lógica programable. En el diseño propuesto se incorporan tres algoritmos de decimación diferentes: normal, alta resolución y detección de pico, los cuales son analizados en detalle en la sección 4.3.7. Cabe destacar que el diseño de la arquitectura del sistema permite la fácil implementación de otros tipos de algoritmos DSP en una instancia futura.

Para poder cumplir con los cuatros objetivos mencionados, se realiza un diseño para esta etapa en base a dos motores DMA y un bloque DSP de diseño específico. A continuación, se presenta la funcionalidad asociada a cada componente involucrado:

1. DMA SRC: Se encarga de realizar la lectura de los datos de cada canal desde el buffer circular de adquisición. Los datos son leídos de manera secuencial desde la primera a la última muestra de la base de tiempo, teniendo en cuenta el offset en el buffer. El acceso a memoria se realiza mediante un puerto dedicado en el controlador SDRAM, a través del puente FPGA-to-SDRAM. A la salida del bloque DMA SRC, se genera un flujo de datos bajo una interfaz Avalon-ST de 32 bits, ancho de palabra correspondiente a una muestra.
2. Procesamiento: Este bloque realiza las operaciones aritméticas necesarias para implementar el algoritmo de decimación seleccionado. Cuenta a su entrada y salida con interfaces Avalon-ST de 32 bits, funcionando bajo un reloj de 50 MHz, es decir, la velocidad de procesamiento. Cabe destacar que este componente actúa sobre las señales de control del bus Avalon-ST de salida, de manera que sólo se almacenen en memoria las muestras decimadas.
3. DMA SINK: Finalmente, el segundo motor de acceso directo a memoria se encarga de recibir las muestras procesadas y almacenarlas de manera secuencial en un sector de memoria separado para cada canal. Este bloque, cuenta con una interfaz de entrada del tipo Avalon-ST de 32 bits funcionando a 50 MHz y genera el acceso a memoria a través de una interfaz Avalon-MM maestro de 32 bits. Esta última, se encuentra conectada directamente a un puerto dedicado del controlador de memoria RAM, a través del puente FPGA-to-SDRAM.

Con el objetivo de complementar gráficamente la información brindada, en la figura 4.12 se presenta un esquema que permite visualizar el proceso de transferencia de datos y decimación a nivel de memoria. Se plantea un ejemplo en el que ambos canales se encuentran habilitados, se adquieren 8 muestras y la adquisición es detenida luego de que se almacenen 6 muestras en el buffer circular. Con respecto al procesamiento, se aplica un factor de decimación de 2, por lo que se obtiene una profundidad de memoria final de 4 muestras por canal.

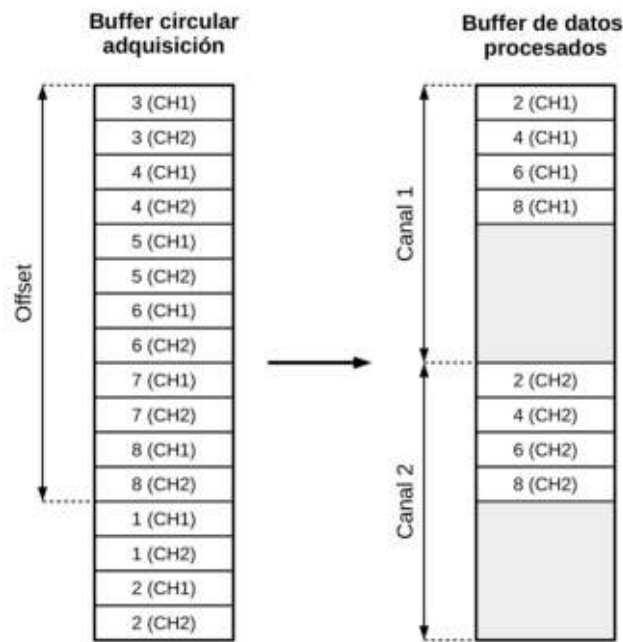


Figura 4.12: Buffers de memoria para captura de 8 muestras con offset de 6 muestras

Lógicamente, se trata de un ejemplo ilustrativo que tiene como objetivo demostrar la manera en que los datos son procesados y almacenados. En el sistema real, la cantidad de muestras es considerablemente mayor, entre 100 muestras y 10 millones de muestras por canal, dependiendo del ajuste de base de tiempo y el factor de decimación seleccionado.

4.3 Implementación de cada etapa

Este apartado brinda los detalles técnicos asociados a cada uno de los componentes presentes en la arquitectura. Es decir, se profundiza en las consideraciones de diseño que son tenidas en cuenta y la implementación práctica realizada.

4.3.1 Generación de relojes

Existen tres dominios de reloj diferentes en el diseño: 1 MHz, 20 MHz y 50 MHz. El primer reloj se asocia a la velocidad de muestreo del sistema, de 1 MSa/s. En base a dicha frecuencia operan los bloques involucrados en la adquisición inicial, es decir, la interfaz del ADC, la lógica de disparo y el motor DMA de adquisición. Por otro lado, la interfaz de los ADC recibe un reloj secundario de 20 MHz, empleado en la interfaz SPI de los convertidores.

El resto de componentes del diseño opera bajo un reloj principal de 50 MHz, destacándose los componentes involucrados en la etapa de procesamiento y transferencia de datos. El reloj de 50 MHz proviene de un oscilador presente en la plataforma de desarrollo DE0-Nano-SoC, el cual actúa como referencia principal. Las frecuencias de 1 MHz y 20 MHz son derivadas, a partir de 50 MHz, en base al empleo de uno de los cinco PLLs que forman parte del sector de lógica

programable. El lazo de enganche de fase es instanciado e integrado en el sistema mediante la herramienta Platform Designer.

Parámetro	Configuración
Modo	Fraccional
Referencia	50 MHz
Salida con estado de bloqueo	Deshabilitado
Numero de relojes	2
Frecuencia de salida 0	1 MHz
Ciclo de trabajo salida 0	50,00 %
Fase de salida 0	0 ps
Frecuencia de salida 1	20 MHz
Ciclo de trabajo salida 1	50,00 %
Fase de salida 1	0 ps

Tabla 4.1: Configuración PLL fraccional

En la tabla 4.1 se listan los parámetros de configuración más relevantes para el PLL. Se puede observar que los dos relojes derivados cuentan con un ciclo de trabajo del 50% y se encuentran en fase con la referencia. De esta manera, se simplifica el diseño del sistema y los cruces de dominio de reloj. Por otro lado, debido a las relaciones entre los tres valores de frecuencia, el PLL es configurado para operar en el modo fraccional.

4.3.2 Interfaz con el front-end

Con el objetivo de controlar y configurar los diversos componentes que forman parte del front-end analógico, se incorporan dos controladores SPI maestros [70] y un puerto GPIO [71]. El objetivo de estos bloques es comunicarse con los siguientes elementos:

- Conversor digital analógico (SPI).
- Potenciómetro digital (SPI).
- Acople de alterna (GPIO).
- Atenuación (GPIO).
- Ganancia programable (GPIO).

Para la implementación de la interfaz, se recurre a bloques ya diseñados por el fabricante y que forman parte de las librerías de Platform Designer. A continuación, se presentan los parámetros de configuración más importantes para los tres componentes:

Parámetro	Configuración
Tipo	Master
Numero de salidas de CS	1
Frecuencia reloj SPI	50 kHz
Ancho de palabra	17 bits (MSB primero)
Polaridad reloj	0
Fase reloj	0

Tabla 4.2: Configuración bloque SPI maestro (potenciómetro)

Parámetro	Configuración
Tipo	Master
Numero de salidas de CS	1
Frecuencia reloj SPI	50 kHz
Ancho de palabra	16 bits (MSB primero)
Polaridad reloj	0
Fase reloj	0

Tabla 4.3: Configuración bloque SPI maestro (DAC)

Parámetro	Configuración
Ancho de palabra	12 bits
Dirección	Salida
Valor en reset	0x000
Configuración de bits individual	Deshabilitada

Tabla 4.4: Configuración puerto GPIO front-end

Con respecto a los periféricos de comunicación SPI, se utiliza una frecuencia relativamente baja, de 50 kHz. De esta forma, se evitan problemas relacionados a la integridad de las señales y se relajan las condiciones de diseño del PCB analógico. Para ambos bloques, se ajusta el tamaño de palabra, orden de bits, polaridad y fase de acuerdo a las hojas de datos de los componentes [32], [37]. Por otro lado, con respecto a la configuración del resto del front-end, se emplea un puerto GPIO de 12 bits, cuya asignación se presenta a continuación:

Bit puerto	11	10	9	8	7	6	5	4	3	2	1	0
Función	-	LED	AT	AC	AT	AC	G2	G1	G0	G2	G1	G0
Canal	-	-	CH2		CH1		CH2			CH1		

Tabla 4.5: Distribución de bits puerto GPIO front-end

Como se puede observar en la tabla 4.5, los primeros seis bits, GPIO[5] a GPIO[0], son dedicados a los amplificadores de ganancia programable de ambos canales. Los siguientes cuatro bits, GPIO[9] a GPIO[6], actúan por sobre el acople de alterna y la selección del atenuador de entrada. Por último, el bit GPIO[10] permite controlar el LED integrado en el PCB del front-end, utilizado para indicar el estado de la comunicación.

Luego de instanciar y configurar los componentes, se procede a integrarlos en el sistema. Los tres bloques se conectan al reloj de 50 MHz y reset principal proveniente del procesador. Por otro lado, las interfaces de estado y control, de tipo Avalon-MM esclavo, se conectan al puente lightweight HPS-to-FPGA. De esta forma, se permite el acceso y control de los componentes desde el procesador mediante software. El detalle de todas las conexiones se presenta en las tablas 4.6 a 4.8 a continuación:

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Interfaz SPI master	Exportada	-	-	Conector GPIO front-end (externo)
Salida de interrupción	IRQ (OUT)	-	Clock principal (1 MHz)	-

Tabla 4.6: Conexiones para bloque SPI maestro (potenciómetro)

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Interfaz SPI master	SPI (M)	-	-	Conector GPIO front-end (externo)
Salida de interrupción	IRQ (OUT)	-	Clock principal (1 MHz)	-

Tabla 4.7: Conexiones bloque SPI maestro (DAC)

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Puerto paralelo de salida	Exportada	12 bits	-	Conector GPIO front-end (externo)

Tabla 4.8: Conexiones puerto GPIO front-end

En las tres tablas de conexiones, 4.6 a 4.8, la primera columna indica el tipo de señal, es decir, si se trata de un puerto Avalon-MM, Avalon-ST, una señal de reloj, reset, etc. De corresponderse, en las siguientes dos columnas se especifica el ancho de palabra y reloj asociado a la interfaz. En último lugar, se expone la conexión que tiene la interfaz con el resto del sistema, indicando entre paréntesis el componente involucrado.

Adicionalmente, forman parte del diseño lógico tres puertos GPIO, asociados a los botones y LEDs integrados en la placa Terasic DE0-Nano-SoC. Cabe recordar que la plataforma elegida cuenta con ocho LEDs, cuatro botones deslizables y dos botones pulsables conectados al sector de lógica programable. Estos elementos no son utilizados en el diseño, sin embargo, el agregado de los puertos permite el acceso por software en futuras modificaciones.

4.3.3 Interfaz ADC

Este componente tiene como objetivo principal recibir las muestras de 12 bits de los ADC y realizar un streaming de datos. El flujo de 64 bits a la salida, generado a la frecuencia de muestreo, consiste en una representación en punto fijo de 32 bits del voltaje asociado a cada canal. La interfaz del ADC cuenta con dos etapas internas diferenciadas, analizadas a continuación:

- Interfaz SPI: Consiste en la lógica encargada de generar las señales de reloj y chip select del bus SPI de los ADC y recibir los datos de 12 bits de ambos canales. El bus de comunicación opera a una frecuencia de reloj de 20 MHz y se generan datos, de 12 bits por canal, a una frecuencia de 1 MHz. Este bloque se implementa directamente en

lenguaje de descripción de hardware VHDL, de acuerdo a los diagramas temporales de los conversores.

- Escalado de datos: Consiste en la lógica relacionada a la conversión de las muestras de 12 bits en una representación en punto fijo de 32 bits. Dado que la etapa de acondicionamiento analógica amplifica la señal y suma un offset, en este bloque se realizan las operaciones matemáticas necesarias para recuperar el valor de voltaje de entrada. El bloque de escalado es diseñado en base al software Simulink y sintetizado mediante el uso de HDL Coder.

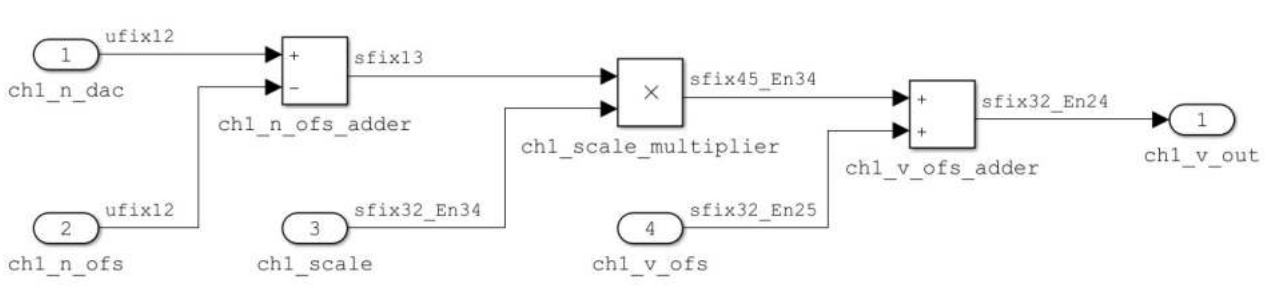


Figura 4.13: Conversión de valor del ADC en voltaje en punto fijo 32 bits

En la figura 4.13 se observan las operaciones aritméticas realizadas por sobre los valores de 12 bits de entrada. Dicho diagrama es una captura de pantalla del diseño realizado en Simulink, en la que sólo se muestra un canal. La ecuación que rige la salida es la siguiente:

$$V_{OUT} = (N_{ADC} - N_{OFS}) \cdot G_{SCALE} + V_{OFS} \quad (4.1)$$

El término N_{ADC} representa el valor de 12 bits proveniente del ADC, por otro lado, N_{OFS} es una constante cuyo valor es de 2048 para todas las escalas. La constante G_{SCALE} cuenta con un valor inicial que depende de la escala vertical seleccionada, su cálculo se deduce de la siguiente expresión:

$$G_{SCALE} = (2^N)^{-1} \cdot \frac{V_{REF}}{V_{ADC}} \cdot V_{FS} = \frac{1}{4096} \cdot \frac{3,3V}{3V} \cdot V_{FS} \quad (4.2)$$

En la ecuación 4.2, N y V_{REF} representan la resolución y tensión de referencia del ADC, 12 bits y 3,3 V respectivamente. Por otro lado, V_{ADC} es la máxima excursión a la entrada del conversor (3 V) y V_{FS} representa la excursión total a la entrada del front-end para cada escala. La ecuación de la ganancia es determinada de manera que se obtenga el voltaje real a la entrada del front-end a partir del valor numérico medido por el DAC.

Escala (V/div)	Cantidad divisiones	V_{FS} (V_{PP})	G_{SCALE}
5	10	50	1,343E-2
2	20	40	1,074E-2
1	20	20	5,371E-3
0,5	20	10	2,686E-3
0,2	20	4	1,074E-3
0,1	20	2	5,371E-4
0,05	20	1	2,686E-4
0,02	20	0,4	1,074E-4
0,01	20	0,2	5,371E-5

Tabla 4.9: Valores de ganancia para cada ajuste de escala vertical

Resulta pertinente aclarar que los algoritmos de calibración, analizados en detalle en el capítulo de software, modifican levemente los valores de las constantes N_{OFS} y G_{SCALE} . De esta forma, se logra eliminar mediante las operaciones aritméticas el error de offset y ganancia, asociado al diseño analógico del front-end.

Finalmente, con respecto al término V_{OFS} , consiste en el valor de voltaje sumado a la señal de entrada mediante el uso del DAC. Es decir, cuenta con un valor nominal de 0 V para todas las escalas y es utilizado para recuperar el voltaje sumado de manera analógica, de modo que V_{OUT} represente el valor real medido.

Con el objetivo de analizar los tipos de datos empleados en la representación de las constantes y en las operaciones aritméticas, se elabora la tabla 4.10 con las entradas y salidas del modelo desarrollado en Simulink. La misma se presenta a continuación:

Nombre	Tipo (Simulink)	Direc.	Signo	Ancho	Fracción	Mínimo	Máximo	Resolución
CH1 N DAC	ufix12	In	No	12	0	0	4095	1
CH1 N OFS	ufix12	In	No	12	0	0	4095	1
CH1 SCALE	sfix32_En34	In	Si	32	34	-0,125	0,125	5,82E-11
CH1 V OFS	sfix32_En25	In	Si	32	25	-64	64	2,98E-08
CH2 N DAC	ufix12	In	No	12	0	0	4095	1
CH2 N OFS	ufix12	In	No	12	0	0	4095	1
CH2 SCALE	sfix32_En34	In	Si	32	34	-0,125	0,125	5,82E-11
CH2 V OFS	sfix32_En25	In	Si	32	25	-64	64	2,98E-08
CH2 V OUT	sfix32_En24	Out	Si	32	24	-128	128	5,96E-08
CH1 V OUT	sfix32_En24	Out	Si	32	24	-128	128	5,96E-08

Tabla 4.10: Entradas y salidas del bloque de conversión a punto fijo (Simulink)

Debido a la precisión finita de los datos de entrada, se emplean representaciones en punto fijo dado que son más rápidas y eficientes en este tipo de hardware. El tamaño de palabra de cada muestra, de 32 bits, es el ancho de palabra nativo de la arquitectura del procesador. La parte fraccional, de 24 bits, permite tener valores en un rango de voltaje de ± 128 V con una resolución de 5,96E-8 (59,6 nV). El rango numérico total es superior al máximo esperado, esto permite aumentar el rango de voltaje de entrada en una futura modificación del sistema.

Las representaciones en punto fijo para los offsets y escalas cumplen el mismo tipo de análisis para su selección. Es decir, se determina la parte entera en base al máximo rango posible y luego se asigna la parte fraccional, de manera que se obtenga el tamaño de palabra del procesador.

Por otro lado, con el objetivo de poder configurar y controlar el componente desde el software, se implementa en el bloque una interfaz Avalon-MM esclavo de 32 bits, conectada al puente lightweight HPS-to-FPGA. De esta forma, por ejemplo, es posible establecer los valores de offset y escalado aplicados a las muestras de entrada. Los 11 registros de 32 bits implementados pueden ser observados en la tabla 4.11 a continuación:

Dirección	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
0x00	CONTROL: VALID OUT(0)							
0x04						CH1 N DAC		
0x08						CH2 N DAC		
0x0C	CH1 V							
0x10	CH2 V							
0x14						CH1 N OFS		
0x18						CH2 N OFS		
0x1C	CH1 V OFS							
0x20	CH2 V OFS							
0x24	CH1 SCALE							
0x28	CH2 SCALE							

Tabla 4.11: Registros implementados en interfaz Avalon-MM de control

Las direcciones de cada registro, en la tabla 4.11, están expresadas en cantidad de bytes, es decir, en el direccionamiento nativo del procesador. El primer registro, asociado al control, cuenta con un bit que permite habilitar o deshabilitar el flujo de datos hacia el bloque de disparo. Los siguientes cuatro registros permiten leer directamente los valores de cada canal antes y luego de la conversión a punto fijo. Estos valores son accedidos y utilizados por el algoritmo de calibración de offset y ganancia. Finalmente, los últimos seis registros permiten leer y escribir las constantes numéricas utilizadas en el bloque de conversión y escalado.

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Salida clock 1 MHz (PLL)
Clock interfaz SPI	Clock (IN)	-	-	Salida clock 20 MHz (PLL)
Reset principal	Reset (IN)	-	Clock principal (1 MHz)	HPS-to-FPGA reset (HPS)
Interfaz SPI master	Exportada	-	-	Conector GPIO front-end (externo)
Salida de datos	Avalon-ST (SRC)	64 bits	Clock principal (1 MHz)	Entrada de datos (sistema disparo)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (1 MHz)	Puente lightweight HPS-to-FPGA (HPS)

Tabla 4.12: Conexiones para bloque de interfaz de ADC (Platform Designer)

Con el objetivo de completar esta sección, se listan en la tabla 4.12 las conexiones realizadas en Platform Designer durante la integración del bloque con el resto del sistema. Cabe destacar que la mayoría de la lógica del componente opera bajo un reloj de 1 MHz, asociado a la frecuencia de muestreo. Adicionalmente, el bloque recibe un reloj de 20 MHz utilizado en la interfaz SPI de los ADC. Como fue mencionado, el flujo de salida cuenta con un ancho de 64 bits, producto de la concatenación de ambos canales en cada instante temporal.

4.3.4 Sistema de disparo

El siguiente componente involucrado en la adquisición es el sistema de disparo, encargado de analizar continuamente las muestras provenientes de la interfaz del ADC en búsqueda de un evento de disparo [72]. Este bloque actúa por sobre el motor DMA de adquisición, deteniendo en el momento adecuado la transferencia de datos a memoria RAM.

La lógica es capaz de detectar, en el canal seleccionado como fuente de disparo, flancos ascendentes y/o descendentes alrededor de un nivel ajustable. Por otro lado, se brinda la posibilidad de deshabilitar el disparo, modalidad en la que se adquieren los datos inmediatamente sin esperar algún tipo de evento. Finalmente, se provee un ajuste de histéresis, cuyo objetivo es agregar inmunidad ante el ruido en la detección de flancos, requiriendo que la señal pase por dos niveles diferentes.

La arquitectura de este bloque consiste principalmente en una máquina de estados diseñada en Stateflow, Simulink. Luego de la síntesis en HDL Coder, el componente se integra en el sistema mediante el uso de Platform Designer. Las conexiones del componente con el resto del diseño se listan en la tabla 4.13 a continuación:

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Salida clock 1 MHz (PLL)
Reset principal	Reset (IN)	-	Clock principal (1 MHz)	HPS-to-FPGA reset (HPS)
Entrada de datos	Avalon-ST (SINK)	64 bits	Clock principal (1 MHz)	Salida de datos (interfaz ADC)
Salida de datos	Avalon-ST (SRC)	64 bits	Clock principal (1 MHz)	Entrada de datos (DMA ADQ)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (1 MHz)	Puente lightweight HPS-to-FPGA (HPS)

Tabla 4.13: Conexiones para bloque de disparo (Platform Designer)

El bloque opera completamente bajo un único reloj de 1 MHz, asociado a la velocidad de muestreo. El flujo de datos, proveniente de la interfaz del ADC, se recibe mediante una interfaz Avalon-ST sink de 64 bits. El bloque analiza las muestras y actúa por sobre las señales de control del bus de salida Avalon-ST source. Adicionalmente, se implementa un puerto Avalon-MM 32 bits de tipo esclavo que permite configurar el bloque y acceder al estado de disparo.

Con el objetivo de analizar el diseño la máquina de estados, se presenta en la figura 4.14 un diagrama temporal con la relación entre el ajuste de posición horizontal, el evento de disparo, y la duración de la base de tiempo.

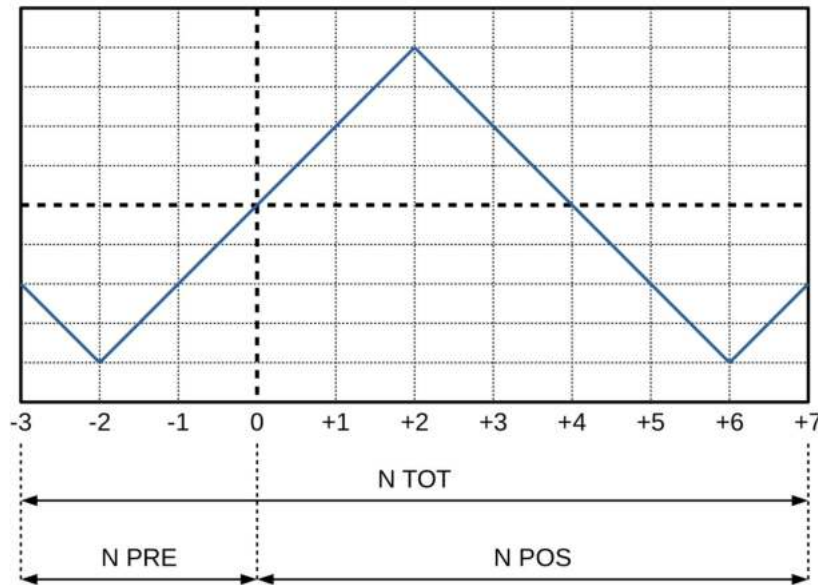


Figura 4.14: Relación entre el evento de disparo y la base de tiempo

Dado que la frecuencia de muestreo del sistema permanece constante, el número total de muestras almacenadas en el buffer circular, para cada canal, depende únicamente del ajuste de base de tiempo. Su valor puede ser calculado en base a la escala horizontal (T_{DIV}), la frecuencia de muestreo (F_S) y la cantidad de divisiones visualizadas en pantalla (N_{DIV}):

$$N_{TOT} = F_S \cdot N_{DIV} \cdot T_{DIV} = (1 \text{ MSa/s}) \cdot (10 \text{ div}) \cdot T_{DIV} \quad (4.3)$$

El ajuste de posición horizontal permite desplazar el evento de disparo a lo largo de la pantalla. Un valor de 50%, significa que el evento se encuentra en el centro de la pantalla, siendo la cantidad de muestras a ambos lados idéntica. Para otro ajuste de posición horizontal porcentual ($POS\%$), la cantidad de muestras antes (N_{PRE}) y después (N_{POS}) del disparo será:

$$N_{PRE} = N_{TOT} \cdot \frac{POS\%}{100} \quad (4.4)$$

$$N_{POS} = N_{TOT} - N_{PRE} \quad (4.5)$$

A modo de ejemplo, el caso presentado en la figura 4.14 se corresponde con una base de tiempo de 1 s/div y un ajuste de posición del 30%. Haciendo uso de las ecuaciones 4.3 a 4.5, se obtiene una profundidad de memoria de 10 millones de muestras (N_{TOT}), de las cuales 3 millones son adquiridas antes del disparo (N_{PRE}) y 7 millones luego del disparo (N_{POS}).

Del análisis anterior surge que la lógica implementada debe ser capaz de detectar el evento de disparo y detener la adquisición N_{POS} muestras después. De esta forma, en el buffer circular queda almacenado un ciclo de adquisición completo, que luego se procesa y envía al cliente. Otro detalle a tener en cuenta, es que se deben adquirir al menos N_{PRE} muestras antes de detectar un posible

evento de disparo. Es decir, en el inicio de un nuevo ciclo de adquisición, es necesario esperar que existan al menos N_{PRE} muestras en el buffer circular antes de buscar un flanco de disparo.

En la figura 4.15, se observa el diagrama de Stateflow de la máquina de estado diseñada para cumplir con los requerimientos mencionados. Cabe destacar que el modelo presentado se corresponde únicamente con el caso de disparo por flanco ascendente:

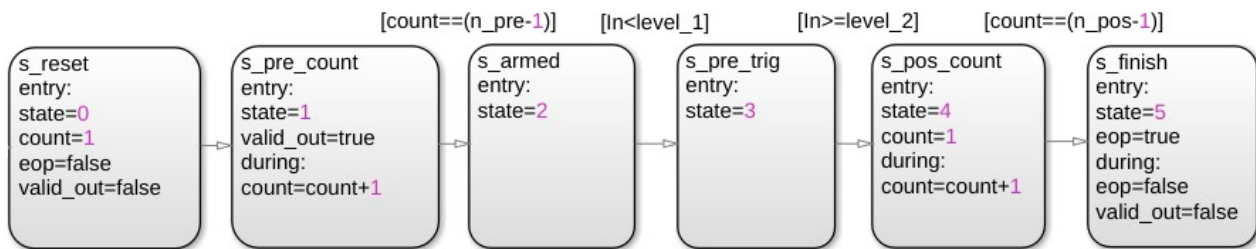


Figura 4.15: Máquina de estado de disparo para flanco ascendente (Simulink)

Debido a la funcionalidad de histéresis, existen dos niveles diferentes de disparo, nivel 1 (o inferior) y nivel 2 (o superior). Para que ocurra un disparo por flanco ascendente exitoso, la señal debe pasar de ser menor que el primer nivel, a ser mayor que el segundo nivel. De este modo, mediante el ajuste de la separación de ambos umbrales, se provee histéresis e inmunidad al ruido, lo que permite evitar falsos disparos [72].

A continuación, se describen los seis estados secuenciales de la máquina de disparo:

- **Reset:** Estado inicial a la espera de una nueva adquisición. Ante el ingreso de la primera muestra se pasa inmediatamente al siguiente estado.
- **Pre-cuenta:** El sistema de disparo permanece en este estado hasta que se almacenen en el buffer al menos N_{PRE} muestras. Ante cada nuevo valor, se incrementa un contador interno, el cual es comparado con N_{PRE} para determinar la transición al siguiente estado.
- **Armado:** Las muestras son analizadas en búsqueda de la primera condición necesaria para el disparo por flanco ascendente. Dicha condición, es que la señal se encuentre por debajo del nivel de voltaje inferior ($level_1$).
- **Pre-disparo:** Una vez ocurrida la primera condición, el sistema permanece en este estado hasta que se cumpla la segunda condición. En el caso de flanco ascendente, la señal debe superar el segundo nivel de voltaje ($level_2$).
- **Pos-cuenta:** El objetivo de este estado es asegurar que se almacenen N_{POS} muestras luego de haber ocurrido el disparo. Ante cada muestra, se incrementa un contador y al llegar al valor establecido ocurre la transición al último estado.
- **Finalización:** Al ingresar a este estado, se detiene el almacenamiento de valores en el buffer circular. La lógica de disparo queda a la espera de un reset, antes de un nuevo ciclo de adquisición.

El flujo del proceso de adquisición es controlado mediante dos señales del bus Avalon-ST de salida de datos: datos válidos (valid) y final de paquete (end of packet). El ingreso al estado pre-cuenta establece la señal valid como verdadera, iniciando el almacenamiento de los datos en el buffer circular. Por otro lado, el ingreso al último estado resetea la señal valid y genera un pulso en la señal eop, indicando la finalización de la adquisición.

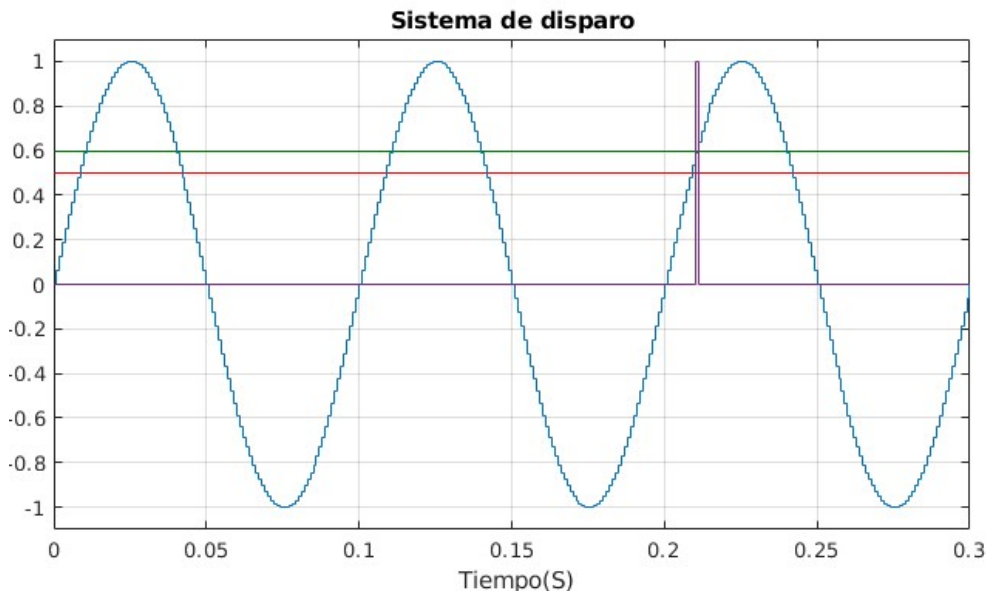


Figura 4.16: Simulación de sistema de disparo (Simulink)

En la figura 4.16, se presenta una simulación del diseño para el caso de una entrada senoidal de 100 muestras por ciclo, bajo la modalidad de disparo por flanco ascendente. El primer nivel de disparo, en color rojo, se ajusta en 0,5 V y el segundo, en color verde, en 0,6 V. Los valores de N_{PRE} y N_{POS} se establecen en 100 muestras, por lo que la adquisición consiste en dos ciclos completos de la señal. El primer flanco ascendente es ignorado debido a que todavía no existen N_{PRE} muestras en el buffer. El disparo ocurre en el segundo flanco y luego de 100 muestras (N_{POS}), se detiene la adquisición mediante un pulso en la señal eop, en color violeta.

A fin de completar la descripción del diseño realizado en Simulink, se presenta una tabla con todas las entradas y salidas del modelo:

Nombre	Tipo (Simulink)	Direc.	Signo	Ancho	Fracción	Mínimo	Máximo	Resolución
CH1 DATA IN	sfix32_En24	In	Si	32	24	-128	128	5,96E-08
CH2 DATA IN	sfix32_En24	In	Si	32	24	-128	128	5,96E-08
CH SELECT	boolean	In	No	1	0	0	1	1
MODE	ufix2	In	No	2	0	0	3	1
TRIG ENB	boolean	In	No	1	0	0	1	1
N PRE	uint32	In	No	32	0	0	2E+032	1
N POS	uint32	In	No	32	0	0	2E+032	1
LEVEL 1	sfix32_En24	In	Si	32	24	-128	128	5,96E-08
LEVEL 2	sfix32_En24	In	Si	32	24	-128	128	5,96E-08
CH1 DATA OUT	sfix32_En24	Out	Si	32	24	-128	128	5,96E-08
CH2 DATA OUT	sfix32_En24	Out	Si	32	24	-128	128	5,96E-08
VALID OUT	boolean	Out	No	1	0	0	1	1
EOP	boolean	Out	No	1	0	0	1	1
STATE	ufix3	Out	No	3	0	0	7	1

Tabla 4.14: Entradas y salidas del bloque de disparo (Simulink)

Como se puede observar en la tabla 4.14, los datos de ambos canales, y los niveles de disparo emplean la misma representación en punto fijo que la interfaz del ADC. Por otro lado, se utilizan enteros sin signo de 32 bits para las cantidades de muestras a adquirir, N_{PRE} y N_{POS} . El modelo cuenta con señales de control que permiten seleccionar la fuente de disparo (ch select), el flanco (mode) y la habilitación (trig enb). Por otro lado, las señales de salida, permiten controlar la adquisición y acceder al estado actual de la máquina de disparo.

Dirección	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
0x00	ESTADO: STATE(2-0)							
0x04	CONTROL: RESET(5) CH SELECT(4) MODE(3-1) TRIG ENB(0)							
0x08	LEVEL 1							
0x0C	LEVEL 2							
0x10	N PRE							
0x14	N POS							

Tabla 4.15: Registros implementados en interfaz Avalon-MM de control

Finalmente, en la tabla 4.15 se listan los registros que forman parte de la interfaz de estado y control conectada al procesador, de tipo Avalon-MM 32 bits esclavo. Como se puede observar, la interfaz le permite al software acceder al estado de disparo, establecer los niveles de disparo, cantidad de muestras, etc.

4.3.5 DMA de adquisición

En este apartado se describe el motor DMA involucrado en la transferencia de los datos a memoria RAM y la implementación del buffer circular. Se emplea el componente Modular Scatter-Gather DMA Core [73] que forma parte de las librerías de Platform Designer. Este bloque se basa en una arquitectura modular adaptable y cuenta con varias funcionalidades que son aprovechadas en el diseño.

Durante la implementación del sistema en Platform Designer, el motor DMA puede ser configurado bajo tres modos de funcionamiento distintos:

- Avalon-MM a Avalon-MM: El bloque cuenta con dos puertos Avalon-MM maestro lo que le permite, por ejemplo, transferir datos entre dos sectores de memoria.
- Avalon-MM a Avalon-ST: En este caso, el componente utiliza un puerto Avalon-MM maestro para leer de manera secuencial los datos y generar un streaming a través de una interfaz Avalon-ST source.
- Avalon-ST a Avalon-MM: Bajo esta modalidad, los datos son recibidos a través de una interfaz Avalon-ST sink. Por otro lado, se implementa una interfaz Avalon-MM maestro que permite escribir de manera secuencial la información en un dispositivo tipo Avalon-MM esclavo.

En el diseño se utiliza un motor DMA configurado bajo la tercera modalidad, es decir, Avalon-ST a Avalon-MM. Bajo este modo de operación, la arquitectura interna del componente es la siguiente [73]:

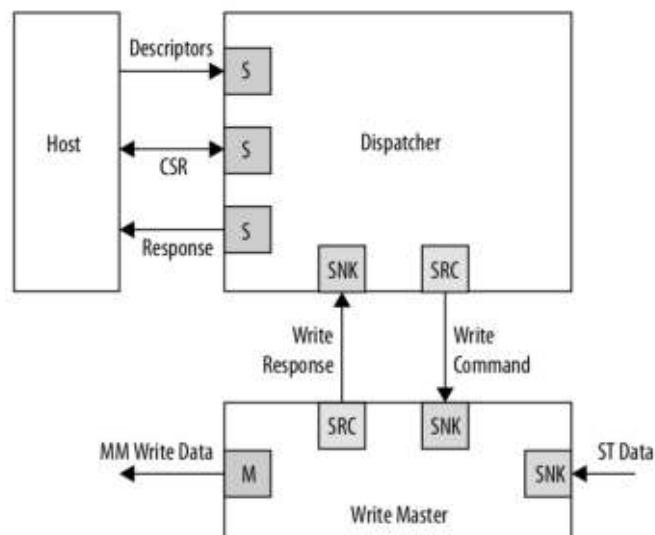


Figura 4.17: Diagrama en bloques DMA adquisición

Cada transferencia se asocia a un descriptor cuyo tamaño, 128 bits o 256 bits, depende de la configuración del bloque DMA en Platform Designer. Los descriptors cuentan con diversos campos, entre los que destacan: cantidad de bytes a transferir, dirección de memoria de lectura y dirección de memoria de escritura. El procesador es el encargado de construir los descriptors de las transacciones a realizar y escribirlos en el bloque DMA a través de un puerto Avalon-MM esclavo dedicado.

Por otro lado, en todas las configuraciones se implementa una interfaz Avalon-MM de propósito general (CSR) que permite acceder al estado del motor DMA y controlarlo. Adicionalmente, la configuración del componente brinda la posibilidad de habilitar un puerto de respuesta de tipo Avalon-MM esclavo o Avalon-ST source. Esta interfaz brinda datos sobre las transferencias realizadas, informando la cantidad de bytes transferidos, si ocurrió algún error, etc.

A continuación, se presenta una tabla con los parámetros de configuración, en Platform Designer, para el caso específico del motor DMA de adquisición:

Parámetro	Configuración
Modo	ST a MM
Ancho datos	64 bits
Tamaño FIFO datos	64 palabras
Tamaño FIFO descriptores	8 descriptores
Puerto respuesta	Habilitado (MM)
Tamaño máximo de transferencia	128 MiB
Tipo de transferencia	Palabras completas
Modo ráfaga	Habilitado (8 palabras)
Ráfaga programable	Deshabilitada
Stride addressing	Deshabilitado
Soporte de paquetes (Avalon-ST)	Habilitado
Soporte de errores (Avalon-ST)	Deshabilitado
Modulo de pre-lectura descriptores	Deshabilitado

Tabla 4.16: Configuración bloque DMA adquisición

Como se puede observar, el bloque opera bajo el modo Avalon-ST a Avalon-MM con un tamaño de palabra de 64 bits. El periférico implementa internamente memorias FIFO (First In First Out) en la interfaz de datos y de descriptores. El buffer FIFO de datos permite absorber un posible retardo o latencia en el puerto Avalon-MM maestro de escritura, en este caso, asociado al acceso de memoria RAM DDR3. De este modo, se evita una potencial pérdida de datos en el flujo de entrada.

Por otro lado, la memoria FIFO de descriptores permite almacenar en el motor DMA múltiples solicitudes de transacciones, de forma que sean ejecutadas de manera secuencial a la máxima velocidad posible. En el caso del motor DMA de ADQ, se emplea un tamaño de 64 palabras para la FIFO de datos y 8 palabras para la FIFO de descriptores.

La habilitación del puerto de respuesta, de tipo Avalon-MM, le permite al procesador acceder a la cantidad de bytes transferidos en la última transacción. Este parámetro resulta de vital importancia para la arquitectura de adquisición, debido a que se asocia al offset de datos en el buffer circular. Otra configuración importante es la habilitación del soporte de paquetes en la interfaz Avalon-ST de entrada. De esta forma, el bloque de disparo puede detener el flujo de adquisición utilizando la señal end of packet.

La interfaz Avalon-MM de escritura, conectada al controlador de memoria RAM, soporta ráfagas de datos, es decir, transferencias continuas e ininterrumpidas de información. En este caso se emplea un tamaño de 8 palabras, valor seleccionado de acuerdo al tipo de memoria RAM utilizada. Por último, la máxima cantidad de datos a transferir se relaciona con el mayor tamaño posible del buffer circular, asociado a la base de tiempo de 1 s/div. A continuación, en la tabla 4.17, se enumeran los tamaños de buffer para cada escala horizontal implementada:

Escala horizontal	Cantidad de muestras (Sa)	Memoria por canal (bytes)	Memoria total (bytes)
10 µs/div	100	400	800
20 µs/div	200	800	1,6 k
50 µs/div	500	2 k	4 k
0,1 ms/div	1 k	4 k	8 k
0,2 ms/div	2 k	8 k	16 k
0,5 ms/div	5 k	20 k	40 k
1 ms/div	10 k	40 k	80 k
2 ms/div	20 k	80 k	160 k
5 ms/div	50 k	200 k	400 k
10 ms /div	100 k	400 k	800 k
20 ms/div	200 k	800 k	1,6 M
50 ms/div	500 k	2 M	4 M
0,1 s/div	1 M	4 M	8 M
0,2 s/div	2 M	8 M	16 M
0,5 s/div	5 M	20 M	40 M
1 s/div	10 M	40 M	80 M

Tabla 4.17: Memoria requerida para cada escala horizontal

Para el cálculo, se tiene en cuenta que cada muestra está formada por 4 bytes y que el sistema cuenta con 2 canales. Teniendo en cuenta lo mencionado, se obtiene a continuación el mayor tamaño de buffer:

$$N_{TOT}(máx) = F_S \cdot N_{DIV} \cdot T_{DIV} = (1 \text{ MSa/s}) \cdot (10 \text{ div}) \cdot (1 \text{ s/div}) = 10 \text{ MSa} \quad (4.6)$$

$$M_{BUFF}(máx) = N_{CH} \cdot N_{TOT} \cdot N_{BYTES} = (2 \text{ canales}) \cdot (10 \text{ MSa}) \cdot (4 \text{ bytes/Sa}) = 80 \times 10^6 \text{ bytes} \quad (4.7)$$

Se requiere un máximo de 80.000.000 bytes (80 MB) o aproximadamente 76,3 MiB si se realiza la conversión al sistema de base 2. Por otro lado, con respecto al ancho de banda, la velocidad de flujos a memoria RAM es constante e igual a:

$$B_{RAM} = (2 \text{ canales}) \cdot (1 \text{ MSa/s}) \cdot (32 \text{ bits/Sa}) = 64 \text{ Mbit/s} \quad (4.8)$$

Es decir, durante el periodo de adquisición de datos, se transfieren 64.000.000 bits por segundo, equivalentes a 8 MB/s (7,63 MiB/s). Cabe destacar que la velocidad requerida es órdenes de magnitud menor al ancho de banda disponible para la arquitectura utilizada [74].

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Salida clock 1 MHz (PLL)
Reset principal	Reset (IN)	-	Clock principal (1 MHz)	HPS-to-FPGA reset (HPS)
Entrada de datos	Avalon-ST (SINK)	64 bits	Clock principal (1 MHz)	Salida de datos (sistema disparo)
Puerto de escritura	Avalon-MM (M)	64 bits	Clock principal (1 MHz)	Puente FPGA-to-SDRAM 0 (HPS)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (1 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Escritura de descriptores	Avalon-MM (S)	128 bits	Clock principal (1 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Puerto de respuesta	Avalon-MM (S)	32 bits	Clock principal (1 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Salida de interrupción	IRQ (OUT)	-	Clock principal (1 MHz)	-

Tabla 4.18: Conexiones para bloque DMA adquisición (Platform Designer)

Finalmente, en la tabla 4.18, se listan las conexiones en Platform Designer del bloque DMA con el resto del sistema. La interfaces de estado y control, de descriptores y de respuesta se conectan al puente lightweight HPS-to-FPGA del procesador. El bloque funciona en su totalidad bajo un único reloj de 1 MHz, asociado a la frecuencia de muestreo, el cual determina la velocidad de entrada y salida de datos. Finalmente, cabe mencionar que el motor DMA cuenta con una salida de interrupción, no utilizada en el diseño.

4.3.6 DMA SRC y DMA SINK

En esta sección se describen los dos bloques Modular Scatter-Gather DMA involucrados en la transferencia de datos durante la etapa de procesamiento. El motor DMA SRC es de tipo Avalon-MM a Avalon-ST y realiza la lectura secuencial de datos desde el buffer circular de adquisición. Por otro lado, el bloque DMA SINK es de tipo Avalon-ST a Avalon-MM y realiza la escritura de los datos procesados en RAM. El diagrama en bloques de la arquitectura de ambos componentes [73] puede ser visualizado en las figuras 4.18 y 4.19.

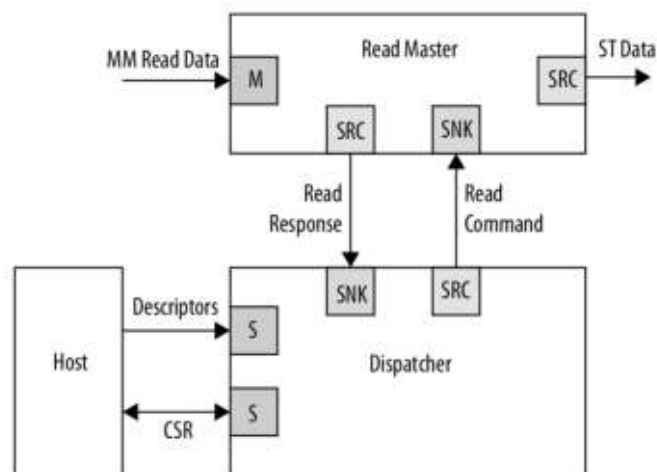


Figura 4.18: Diagrama en bloques DMA SRC

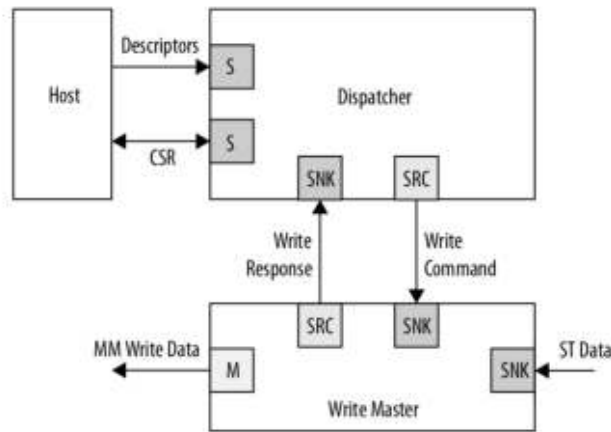


Figura 4.19: Diagrama en bloques DMA SINK

Cabe recordar que, de acuerdo a la sección 4.2.3, el procesamiento y la transferencia de datos tiene tres objetivos principales: separar las muestras de ambos canales, eliminar el offset del buffer circular y reducir la cantidad de datos a enviar y graficar. Los datos procesados de cada canal se almacenan en sectores de memoria independientes. En el esquema de la figura 4.20, se observa la distribución de la memoria del sistema.

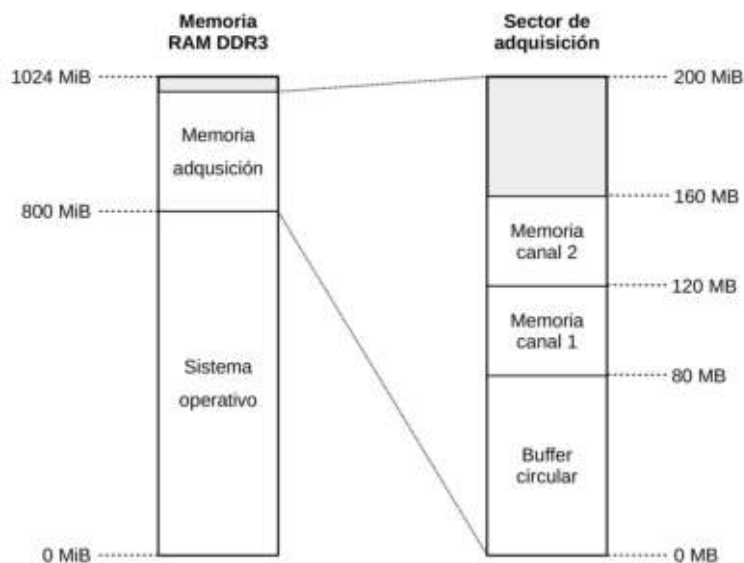


Figura 4.20: Buffers de adquisición implementados en memoria RAM

De la totalidad de 1024 MiB de memoria RAM, los primeros 800 MiB son asignados al sistema operativo, los siguientes 200 MiB conforman la memoria de adquisición, mientras que el resto (24 MiB) no son utilizados. En la parte derecha de la figura 4.20, se presenta un detalle de los 200 MiB que conforman el sector de adquisición. En este caso, se asignan 80 MB al buffer circular, cantidad obtenida en la sección 4.3.5, y 40 MB a cada canal. Este último valor, es el tamaño máximo requerido para los datos procesados, asociado al caso en que se seleccione la base de

tiempo de 1 s/div y no se aplique decimación. Su cálculo se lleva a cabo en la ecuación 4.9, a continuación:

$$M_{CH}(máx) = N_{TOT} \cdot N_{BYTES} = (10 \text{ MSa}) \cdot (4 \text{ bytes/Sa}) = 40 \times 10^6 \text{ bytes} \quad (4.9)$$

El empleo de dos motores DMA, cada uno conectado a un puerto dedicado del controlador de memoria, permite la lectura y escritura de datos en simultáneo, reduciendo el retardo entre dos adquisiciones consecutivas. Con el mismo objetivo, la transferencia de datos ocurre a una frecuencia de 50 MHz. Como consecuencia, se reduce 50 veces el tiempo de procesamiento individual de cada canal, con respecto al tiempo de adquisición. Teniendo en cuenta que los canales se procesan individualmente, en el caso que ambos estén habilitados, el tiempo de procesamiento necesario es 25 veces menor al de adquisición.

$$T_{PROC} = N_{CH} \cdot N_{TOT} \cdot (F_{CLK})^{-1} \quad (4.10)$$

En la ecuación 4.10 se obtiene el tiempo requerido para procesar los datos T_{PROC} como función del número de canales habilitados N_{CH} , el número de muestras por canal en el buffer circular N_{TOT} y la frecuencia de reloj F_{CLK} (50 MHz). Existen dos factores que determinan la máxima velocidad de procesamiento posible: el ancho de banda de la memoria RAM y la frecuencia máxima de operación del algoritmo de procesamiento. En el diseño realizado, el elemento limitante que determina la frecuencia de reloj utilizada de 50 MHz es el ancho de banda.

La máxima velocidad de acceso a memoria se da para el caso en el que el algoritmo de procesamiento no reduce la cantidad de datos, es decir, cuando se emplea un factor de decimación unitario. Bajo estas condiciones, en la ecuación 4.11, se obtiene el ancho de banda requerido.

$$B_{RAM}(máx) = (2) \cdot (50 \text{ MSa/s}) \cdot (32 \text{ bits/Sa}) = 3,2 \text{ Gbit/s} \quad (4.11)$$

El primer término de la ecuación denota la situación de lectura y escritura simultánea, la cual duplica la velocidad de transferencia. Durante el desarrollo, se utilizó la herramienta SignalTap para observar los buses Avalon-ST y Avalon-MM de ambos motores DMA. Esto permitió determinar experimentalmente que el valor de 3,2 Gbit/s (400 MB/s) asociado al reloj de 50 MHz se encuentra en el límite de ancho de banda. Es decir, al utilizar frecuencias mayores, se detectó que son necesarios varios ciclos de reloj para la lectura y/o escritura de algunos datos en memoria.

Parámetro	Configuración
Modo	MM a ST
Ancho datos	32 bits
Tamaño FIFO datos	64 palabras
Tamaño FIFO descriptores	8 descriptores
Puerto respuesta	Deshabilitado
Tamaño máximo de transferencia	128 MiB
Tipo de transferencia	Palabras completas
Modo ráfaga	Deshabilitado
Ráfaga programable	Deshabilitada
Stride addressing	Habilitado (2 palabras)
Soporte de paquetes (Avalon-ST)	Deshabilitado
Soporte de errores (Avalon-ST)	Deshabilitado
Modulo de pre-lectura descriptores	Deshabilitado

Tabla 4.19: Configuración bloque DMA SRC

Parámetro	Configuración
Modo	ST a MM
Ancho datos	32 bits
Tamaño FIFO datos	64 palabras
Tamaño FIFO descriptores	8 descriptores
Puerto respuesta	Deshabilitado
Tamaño máximo de transferencia	128 MiB
Tipo de transferencia	Palabras completas
Modo ráfaga	Habilitado (8 palabras)
Ráfaga programable	Deshabilitada
Stride addressing	Deshabilitado
Soporte de paquetes (Avalon-ST)	Deshabilitado
Soporte de errores (Avalon-ST)	Deshabilitado
Modulo de pre-lectura descriptores	Deshabilitado

Tabla 4.20: Configuración bloque DMA SINK

En las tablas 4.19 y 4.20, se listan los parámetros de configuración más importantes para ambos componentes. La habilitación del modo stride addressing para el motor DMA SRC permite ajustar el incremento en la dirección de lectura luego de cada dato transferido. De esta forma, estableciendo el valor en 8 bytes (en lugar de 4 bytes) se pueden leer del buffer circular las muestras pertenecientes a un único canal. Por otro lado, dado que las direcciones de lectura no son contiguas, esta funcionalidad es incompatible con el modo ráfaga, deshabilitado para el bloque DMA SRC.

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Puerto de lectura	Avalon-MM (M)	32 bits	Clock principal (50 MHz)	Puente FPGA-to-SDRAM 1 (HPS)
Salida de datos	Avalon-ST (SRC)	32 bits	Clock principal (50 MHz)	Entrada de datos (procesamiento)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Escritura de descriptores	Avalon-MM (S)	256 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Salida de interrupción	IRQ (OUT)	-	Clock principal (1 MHz)	-

Tabla 4.21: Conexiones para bloque DMA SRC (Platform Designer)

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Entrada de datos	Avalon-ST (SINK)	32 bits	Clock principal (50 MHz)	Salida de datos (procesamiento)
Puerto de escritura	Avalon-MM (M)	32 bits	Clock principal (50 MHz)	Puente FPGA-to-SDRAM 2 (HPS)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Escritura de descriptores	Avalon-MM (S)	128 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)
Salida de interrupción	IRQ (OUT)	-	Clock principal (50 MHz)	-

Tabla 4.22: Conexiones para bloque DMA SINK (Platform Designer)

Finalmente, en las tablas 4.21 y 4.22, se listan las conexiones en Platform Designer para ambos bloques. Al igual que el bloque DMA ADQ, el procesador accede a las interfaces de control y escritura de descriptores mediante el puente lightweight HPS-to-FPGA. Por otro lado, el bloque de procesamiento se encuentra conectado a las interfaces Avalon-ST de ambos motores DMA.

4.3.7 Procesamiento

Las muestras adquiridas de cada canal circulan por el bloque de procesamiento, en el cual se aplican diversos algoritmos de decimación. Independientemente del ajuste de escala temporal, que actúa por sobre el tamaño del buffer circular, se puede seleccionar la cantidad de muestras a ser enviadas y graficadas en cada ciclo de adquisición. Este ajuste actúa por sobre el factor de decimación D utilizado en el algoritmo, siendo la cantidad de muestras decimadas:

$$N_{DEC} = \frac{N_{TOT}}{D} \quad (4.12)$$

Con el objetivo de brindar flexibilidad al diseño, para cada base de tiempo se implementan todos los factores de decimación que den como resultado un valor de N_{DEC} par y mayor a 100 muestras. De este modo, se puede ajustar la frecuencia de muestreo equivalente y la profundidad de memoria de acuerdo a la medición a realizar [55]. A modo de ejemplo, en la tabla 4.23 se listan los valores para la base de tiempo de 1 s/div.

Cantidad de muestras (Sa)	Frecuencia de muestreo (Sa/s)	Factor de decimación	Cantidad de muestras (Sa)	Frecuencia de muestreo (Sa/s)	Factor de decimación
10 M	1 M	1	12,5k	1,25 k	800
5 M	500 k	2	10 k	1 k	1 k
2,5 M	250 k	4	8 k	800	1,25 k
2 M	200 k	5	5 k	500	2 k
1 M	100 k	10	4 k	400	2,5 k
500 k	50 k	20	2,5 k	250	4 k
400 k	40 k	25	2 k	200	5 k
250 k	25 k	40	1,6 k	160	6,25 k
200 k	20 k	50	1,25 k	125	8 k
125 k	12,5 k	80	1 k	100	10 k
100 k	10 k	100	800	80	12,5 k
80 k	8 k	125	500	50	20 k
50 k	5 k	200	400	40	25 k
40 k	4 k	250	250	25	40 k
25 k	2,5 k	400	200	20	50 k
20 k	2 k	500	100	10	100 k

Tabla 4.23: Factores de decimación implementados para la base de tiempo de 1 s/div

El bloque de procesamiento es diseñado en el software Simulink y sintetizado utilizando la herramienta HDL Coder. El componente cuenta con interfaces de tipo Avalon-ST para los datos y Avalon-MM para la configuración y control. Las conexiones en Platform Designer se presentan en la tabla 4.24 a continuación:

Descripción	Tipo	Ancho	Clock asociado	Conexión
Clock principal	Clock (IN)	-	-	Clock 50 MHz (externo)
Reset principal	Reset (IN)	-	Clock principal (50 MHz)	HPS-to-FPGA reset (HPS)
Entrada de datos	Avalon-ST (SINK)	32 bits	Clock principal (50 MHz)	Salida de datos (DMA SRC)
Salida de datos	Avalon-ST (SRC)	32 bits	Clock principal (50 MHz)	Entrada de datos (DMA SINK)
Interfaz de estado y control	Avalon-MM (S)	32 bits	Clock principal (50 MHz)	Puente lightweight HPS-to-FPGA (HPS)

Tabla 4.24: Conexiones para bloque de procesamiento (Platform Designer)

Como se puede observar en la tabla 4.24, la interfaces de entrada y salida de datos, de tipo Avalon-ST, transfieren las muestras de 32 bits a una velocidad de 50 MHz. Por otro lado, la interfaz de control es de tipo Avalon-MM esclavo de 32 bits y se encuentra conectada al procesador a través del puente lightweight HPS-to-FPGA.

El bloque cuenta con tres algoritmos distintos de decimación: normal, alta resolución y detección de pico. En el primer caso, se selecciona la primera de cada D muestras, descartando las D-1 muestras restantes. Este modo de operación es equivalente a haber muestreado a una frecuencia D veces menor. Es el método más simple e intuitivo, pero no aprovecha la información extra que se obtiene al adquirir a una mayor velocidad.

En el diagrama de la figura 4.21, se presenta un ejemplo del modelo de Simulink que se utiliza para la síntesis del algoritmo de decimación normal, aplicado a 1.000 valores de una senoidal de entrada de amplitud 20. El factor de decimación utilizado es de 20 muestras, por lo que se obtienen a la salida 20 valores.

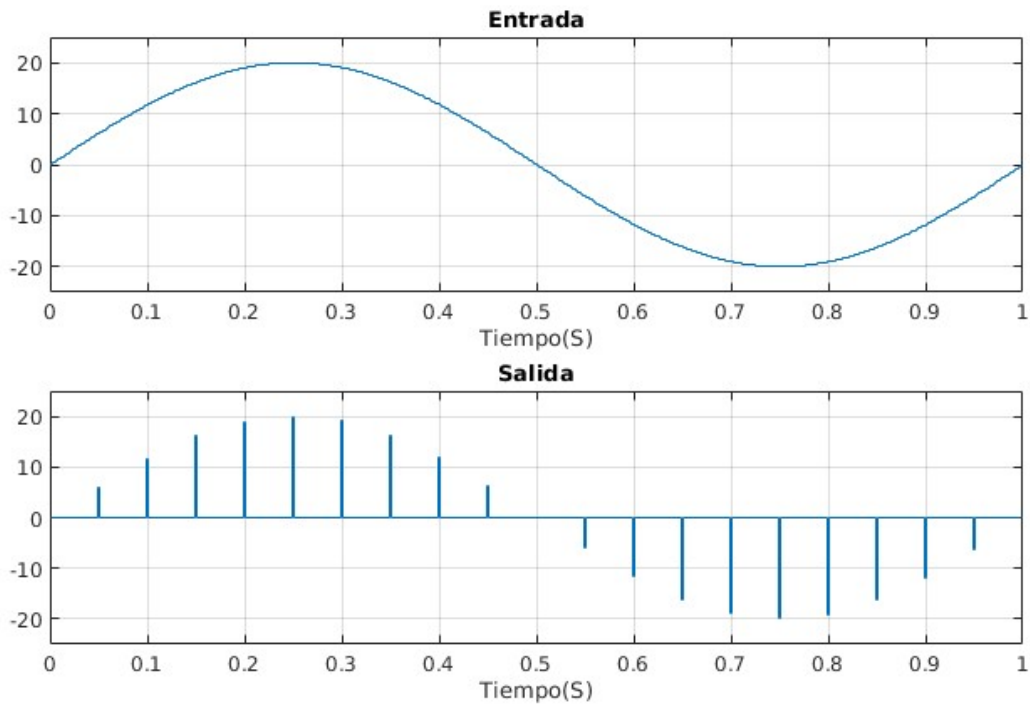


Figura 4.21: Decimación normal con factor $D=50$ (Simulink)

En el caso de alta resolución, cada valor decimado se obtiene realizando un promedio de las D muestras del intervalo original. Esta técnica, también denominada “boxcard averaging”, puede ser vista como la aplicación de un filtro de promedio móvil de largo D , seguido por una decimación del mismo factor [75] [76].

Un filtro de promedio móvil, tiene una respuesta al impulso de tipo escalón, en este caso de largo D , cuyo efecto en el tiempo es promediar las últimas D muestras. En frecuencia, tiene una respuesta del tipo sinc que puede ser vista como un filtrado pasa bajos no ideal. Con el objetivo de analizar los efectos del algoritmo, se presenta, en la figura 4.22, ambas funciones para un factor de decimación de 5.

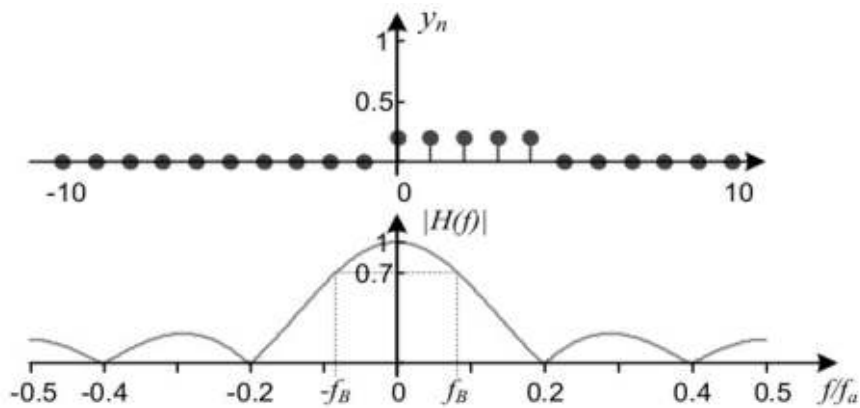


Figura 4.22: Respuesta de filtro de promediado móvil con $D=5$

En la respuesta en frecuencia, la caída del primer lóbulo ocurre en $1/D$ y el ancho de banda de 3 dB resulta ser aproximadamente $0,44/D$ (f_B en la figura 4.22). Considerando la frecuencia de muestreo de 1 MSa/s, el cruce por cero se da a 200 kHz y el ancho de banda es de 88 kHz. Dado que el filtro no es ideal, se debe asegurar que la señal de entrada no contenga contenido significativo por encima de la frecuencia de corte, de modo contrario, al realizar el proceso de decimación ocurre aliasing y distorsión. Este fenómeno se puede observar a continuación:

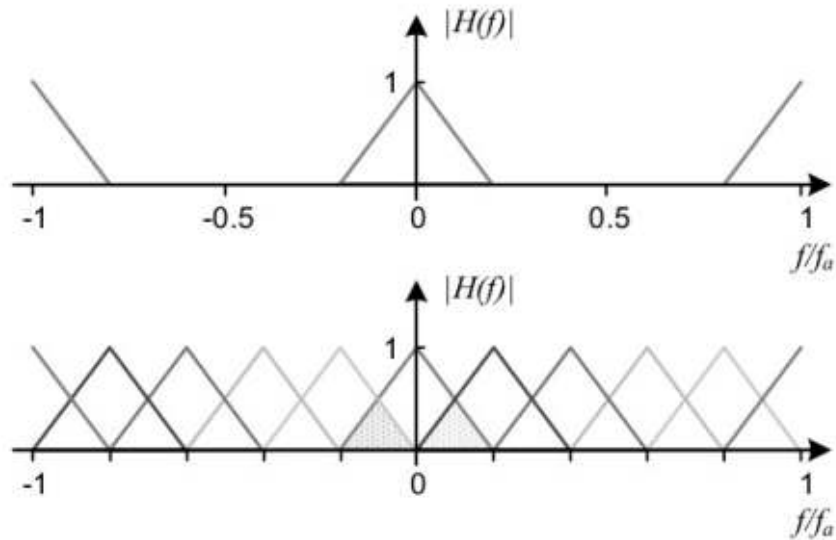


Figura 4.23: Efecto de aliasing en decimación con $D=5$

Teniendo en cuenta las limitaciones presentadas, la aplicación de filtrado digital permite reducir el ruido, tanto en la señal de entrada como en el front-end, mejorando la SNR total obtenida. Adicionalmente, se puede demostrar que la combinación de oversampling y filtrado digital reduce el ruido de cuantización debido al ADC, permitiendo aumentar la resolución del sistema [77]. Cada vez que se cuadruplica el factor D , se obtiene un bit extra de resolución, de acuerdo a la relación de la ecuación 4.13.

$$R_{BITS} = N_{ADC} + (0,5) \cdot \log_2(D) \quad (4.13)$$

Por ejemplo, muestrear a 1 MSa/s con un ADC de 12 bits y aplicar una decimación D de 256, brinda una resolución efectiva de 16 bits. Sin embargo, resulta pertinente aclarar que la fórmula 4.13 solo se asocia a la cuantización, es decir, no considera el ruido proveniente del ADC y el front-end. La resolución efectiva del sistema en su totalidad, denominada ENOB, depende de una mayor cantidad de factores.

En la figura 4.24 se observa una simulación del modelo diseñado en Simulink para el algoritmo de alta resolución, aplicado a una señal senoidal de amplitud 20 inmersa en ruido. Se utiliza un factor de decimación de 50, por sobre los 1.000 valores presentes a la entrada, por lo que se obtienen 20 muestras decimadas. El promediado se asocia a una reducción significativa del ruido, obteniendo a la salida una aproximación de la senoidal original.

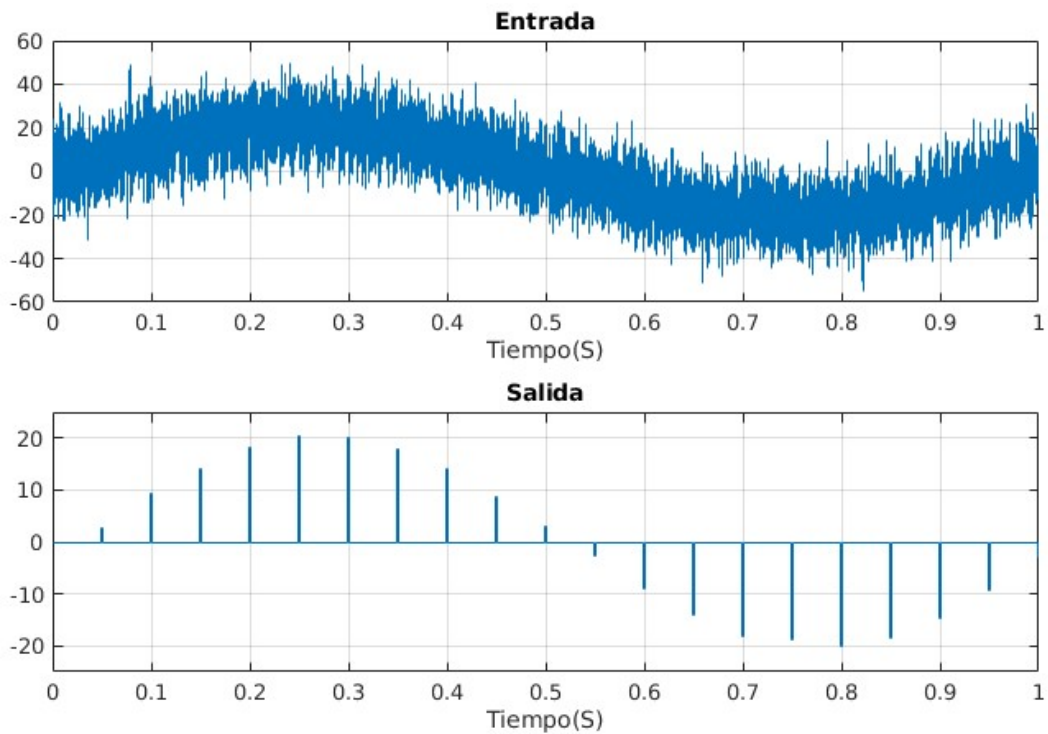


Figura 4.24: Decimación de alta resolución con factor $D=50$ (Simulink)

Con respecto al modo de detección de pico, el algoritmo selecciona el mínimo valor de un intervalo de D muestras. En el siguiente intervalo, se busca y selecciona el máximo, repitiendo el proceso de manera alternada. Este método permite obtener información sobre la presencia de transitorios o ruido entre dos muestras decimadas. Resulta de gran utilidad para casos en los que se desea observar una base de tiempo grande, haciendo uso de poca memoria y poder detectar cualquier evento anormal de alta frecuencia que pueda ocurrir entre dos muestras consecutivas.

A modo de ejemplo se presenta una simulación del algoritmo de detección de pico, aplicado a una señal de entrada del tipo ruido blanco Gaussiano limitado en banda, siendo el factor de decimación 50. En el gráfico de la figura 4.25, se observa que la señal decimada muestra claramente la presencia de ruido en la entrada.

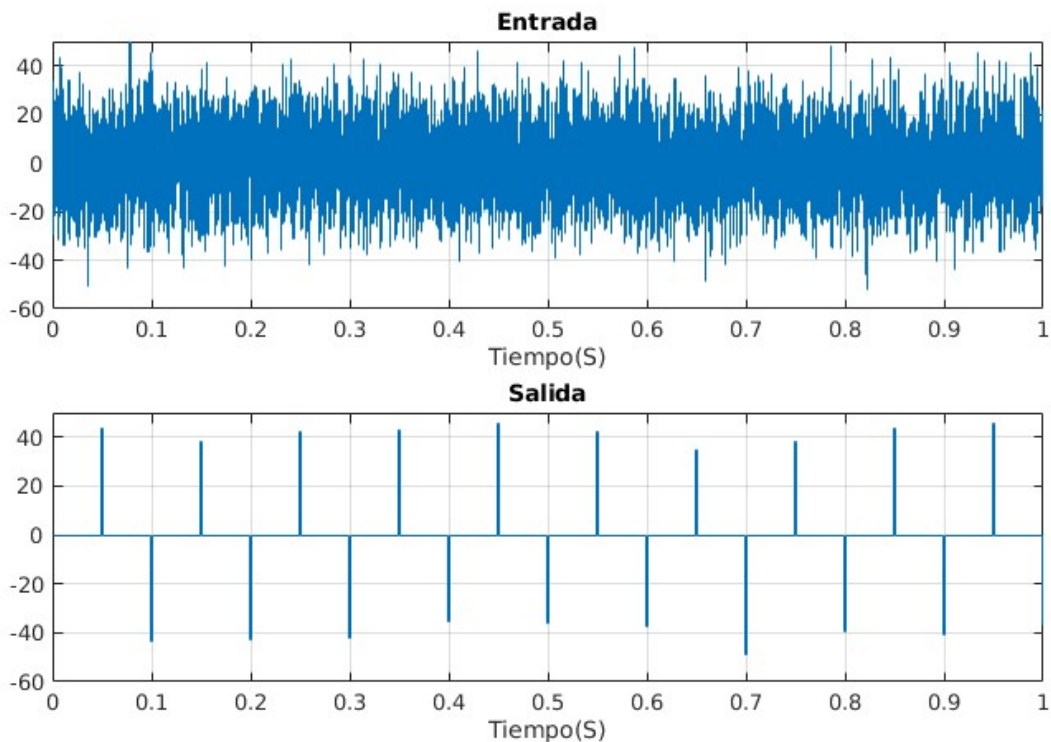


Figura 4.25: Decimación de detección de pico con factor $D=50$ (Simulink)

Los tres algoritmos se implementan principalmente en base a la utilización de registros, multiplexores y operadores aritméticos en punto fijo. Para las entradas y salidas se hace uso de la misma representación del voltaje que en el resto de componentes, es decir, 32 bits con signo y 24 bits de parte fraccional. Cabe destacar que, con el objetivo de lograr el rango y precisión requerido para cada operación, se realiza el escalado necesario de las señales internas. Las entradas y salidas del modelo de Simulink se enumeran en la tabla 4.25 a continuación:

Nombre	Tipo (Simulink)	Direc.	Signo	Ancho	Fracción	Mínimo	Máximo	Resolución
DATA IN	sfix32_En24	In	Si	32	24	-128	128	5,96E-08
D	ufix18	In	No	18	0	0	2E+018	1
D INV	ufix32_En31	In	No	32	31	0	2	4,65E-10
MODE	ufix2	In	No	2	0	0	3	1
DATA OUT	sfix32_En24	Out	Si	32	24	-128	128	5,96E-08
VALID OUT	boolean	Out	No	1	0	0	1	1

Tabla 4.25: Entradas y salidas del bloque de procesamiento (Simulink)

Se emplea una variable de 2 bits para establecer el modo de operación: normal, alta resolución o detección de pico. Así mismo, se utiliza un tamaño de palabra de 18 bits sin signo para el factor de decimación. Este tamaño es determinado en base al máximo factor de decimación posible en el sistema. Con respecto a la señal VALID OUT, su objetivo es indicar la presencia de muestras decimadas válidas a la salida, de manera que sean almacenadas por el motor DMA SINK.

Debido a la implementación del algoritmo de alta resolución, el mismo requiere de la inversa del factor D . En una primera iteración de diseño, la inversa era realizada en la lógica programable, lo

cual tenía un impacto considerable en la máxima frecuencia de operación del algoritmo y en la utilización de recursos. Esta situación fue detectada durante la síntesis del modelo y el análisis temporal. Por lo tanto, en el diseño final se realiza el recíproco directamente en el procesador, motivo por el cual el algoritmo cuenta con una variable de entrada D INV, representada en punto fijo de 32 bits.

Dirección	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0
0x00	ESTADO: RESERVADO							
0x04	CONTROL: RESET(2) MODE(1-0)							
0x08	D(17-0)							
0x0C	D INV							

Tabla 4.26: Registros implementados en interfaz Avalon-MM de control

Finalmente, en la tabla 4.26 se enumeran los registros de estado y control que forman parte del bloque de procesamiento. Como se puede observar, bajo la interfaz Avalon-MM de tipo esclavo se implementan registros para establecer los parámetros de operación. De esta forma, el procesador puede configurar el bloque previo al procesamiento de las muestras de un canal.

4.3.8 HPS

El elemento principal en un diseño basado en un dispositivo SoC FPGA es el sistema de procesamiento (Hard Processor System). A pesar de tratarse de hardware embebido en el circuito integrado, y no lógica programable, el HPS es considerado en Platform Designer como un componente. Es decir, el bloque asociado al HPS debe ser agregado en Platform Designer al sistema y se debe realizar su configuración, al igual que el resto de componentes.

Debido a la complejidad de la arquitectura del procesador y sus periféricos, el componente HPS de Platform Designer cuenta con una gran cantidad de configuraciones. La mayoría de estos parámetros se encuentran relacionados a la implementación del SoC FPGA en la plataforma de desarrollo utilizada. Teniendo en cuenta lo mencionado, se emplea como base del diseño el proyecto de referencia de Platform Designer, provisto por el fabricante Terasic. Los parámetros más importantes del HPS definidos en el modelo se presentan a continuación:

- Habilitación, modo de operación y configuración de los periféricos del HPS (UART, SPI, USB, Ethernet, USB, SD/MMC, entre otros).
- Multiplexado de entradas y salidas entre los diferentes periféricos y asignación física de pines en el circuito integrado.
- Frecuencias de reloj de cada periférico, configuración de los osciladores y PLLs internos al HPS.
- Cantidad, tipo y modo de operación de la memoria RAM, parámetros de temporización y relojes de la interfaz DDR3.

Por otro lado, la configuración relacionada a la comunicación entre el HPS y la FPGA es modificada de acuerdo a las necesidades del diseño. En la tabla 4.27 se especifican los diversos puentes de comunicación entre el HPS y la FPGA y sus parámetros de configuración.

Puente	Tipo	Ancho	Clock	Conexión
Lightweight HPS-to-FPGA	Avalon-MM (M)	32 bits	50 MHz	Múltiples
HPS-to-FPGA	Avalon-MM (M)	64 bits	50 MHz	-
FPGA-to-HPS	Avalon-MM (S)	128 bits	50 MHz	-
FPGA-to-SDRAM 0	Avalon-MM (S) (write)	64 bits	1 MHz	Puerto de escritura (DMA ADQ)
FPGA-to-SDRAM 1	Avalon-MM (S) (read)	32 bits	50 MHz	Puerto de lectura (DMA SRC)
FPGA-to-SDRAM 2	Avalon-MM (S) (write)	32 bits	50 MHz	Puerto de escritura (DMA SINK)

Tabla 4.27: Parámetros de puentes entre el HPS y la FPGA

De acuerdo a las recomendaciones del fabricante [15], se hace uso del puente lightweight HPS-to-FPGA para acceder a los registros de estado y control de los bloques implementados en la lógica programable. El puente funciona en base al reloj principal de 50 MHz, debido a que la mayoría de componentes de la FPGA operan a esa frecuencia. Dado que la interfaz del ADC, el bloque de disparo y el motor de DMA ADQ funcionan bajo una frecuencia de 1 MHz, existe un cruce de dominio de reloj entre el puerto Avalon-MM maestro del puente y el puerto Avalon-MM esclavo de los componentes. Bajo esta situación, Platform Designer genera automáticamente la lógica necesaria para el cruce de dominio, evitando problemas de violación de tiempos y metaestabilidad [60].

Con respecto a la comunicación entre la memoria RAM y la lógica programable, se emplean tres puertos dedicados en el controlador SDRAM del HPS. El primer puente es utilizado por el bloque DMAADQ para la escritura en RAM de datos, el mismo cuenta con un ancho de palabra de 64 bits y se encuentra conectado al reloj de 1 MHz. Es decir, se hace uso de la misma configuración que el puerto de escritura del motor DMA, evitando la necesidad de lógica de adaptación o de cruce de reloj.

Los otros dos puentes son utilizados para la lectura (DMA SRC) y escritura (DMA SINK) de datos durante el procesamiento. Ambos puertos cuentan con el mismo ancho de palabra y frecuencia de operación que los motores DMA. La correcta configuración del puente evita la implementación automática de lógica de adaptación o de cruce de dominio de reloj entre el puente y los componentes en la lógica programable.

Componente	Dirección inicial	Dirección final	Cantidad de bytes	Cantidad de registros
Interfaz ADC	0x0000	0x003F	64	16
Sistema de disparo	0x0040	0x005F	32	8
SPI DAC	0x0060	0x007F	32	8
SPI Potenciómetro	0x0080	0x009F	32	8
Descriptores DMA SRC	0x00A0	0x00BF	32	8
Control DMA SINK	0x00C0	0x00DF	32	8
Control DMA SRC	0x00E0	0x00FF	32	8
Control DMA ADQ	0x0100	0x011F	32	8
Sistema Procesamiento	0x0120	0x012F	16	4
Puerto GPIO	0x0130	0x013F	16	4
Puerto botones pulsables	0x0140	0x014F	16	4
Puerto botones deslizables	0x0150	0x015F	16	4
Puerto LEDs	0x0160	0x016F	16	4
Descriptores DMA SINK	0x0170	0x017F	16	4
Descriptores DMA ADQ	0x0180	0x018F	16	4
Puerto respuesta DMA ADQ	0x0190	0x0197	8	2

Tabla 4.28: Vista de memoria del procesador (lightweight HPS-to-FPGA)

En la tabla 4.28 se puede observar la vista de memoria del procesador por sobre el puente lightweight HPS-to-FPGA. Esta información es brindada por Platform Designer, de acuerdo a la lógica de interconexión implementada automáticamente. El rango de direcciones de cada componente resulta de vital importancia a la hora de acceder mediante software a los bloques diseñados. Las primeras dos columnas se corresponden con la dirección inicial y final, en hexadecimal, de los registros implementados. Estos valores están expresados en bytes y representan un offset por sobre la dirección física inicial asignada al puente lightweight HPS-to-FPGA en el mapa de memoria del procesador. Finalmente, se dedican las dos últimas columnas a la cantidad de bytes y de registros de 32 bits, para cada periférico en el bus.

4.4 Síntesis del diseño

El objetivo de este último apartado es presentar diversos datos relacionados a la síntesis del diseño digital. En primer lugar, se analizan los recursos utilizados por el diseño, de acuerdo al reporte de compilación del software Quartus. En segundo lugar, se listan los parámetros más relevantes asociados al análisis de tiempos realizado en la herramienta Timing Analyzer, perteneciente a Quartus.

4.4.1 Recursos utilizados en la FPGA

En el capítulo 2, dedicado a la arquitectura Cyclone V, se describen los diferentes tipos de elementos que componen la lógica programable: ALMs, LABs, bloques DSP, entre otros. Así mismo, para cada recurso, se especifica la cantidad disponible en la FPGA usada. La síntesis del diseño permite obtener estadísticas sobre el grado de utilización de los recursos. Los datos más relevantes, asociados al diseño final, se presentan a continuación:

- ALM: 4.473 de 15.880 (28%).
- Registros: 6.903 de 63.520 (11%).
- LAB: 675 de 1.588 (43%).

- Bits de memoria (en bloques 10K): 16.552 de 2.764.800 (<1%).
- Bloques de memoria 10K (parcial o totalmente utilizados): 42 de 270 (16%).
- Bloques DSP: 8 de 84 (10%).
- PLL: 1 de 5 (20%).

La lógica disponible es utilizada en promedio en un 18%, lo que permite el agregado de funcionalidades a futuro. El motivo más predominante para el bajo uso de recursos es el empleo de aritmética de punto fijo y el correcto escalado de las constantes. A continuación, en la tabla 4.29, se evalúa el consumo de recursos para cada bloque individual:

Bloque	ALM	Registros (bits)	Memoria (bits)	Memoria (bloques 10K)	Bloques DSP	Porcentaje
SPI Potenciómetro	68,7	227	0	0	0	1
SPI DAC	58	158	0	0	0	0,7
Puerto GPIO	4,5	14	0	0	0	0,1
Interfaz ADC	141,3	388	0	0	4	11,8
Sistema disparo	268,7	232	0	0	0	1,9
DMA ADQ	194,6	367	5.528	7	1	14,4
DMA SRC	131,3	207	2.696	5	0	6,8
DMA SINK	143,2	219	2.528	4	0	6,2
Procesamiento	247,3	244	0	0	3	9,3
Lógica interconexión	3.197,7	4.808	5.800	26	0	47,6
Otros	17,7	39	0	0	0	0,2
Total	4.473	6.903	16.552	42	8	100

Tabla 4.29: Recursos utilizados en la FPGA para cada bloque

La mayor utilización se da para la lógica de interconexión, generada automáticamente por Platform Designer. Esta categoría abarca los elementos necesarios para implementar los diversos buses Avalon-MM y Avalon-ST, la comunicación con el HPS, el cálculo de direcciones, arbitrado y priorizado de maestros en un bus, transición entre diferentes dominios de reloj, etc. [57]. Debido a la alta complejidad y funcionalidad asociada a la lógica de interconexión, el alto consumo de recursos observado en la tabla 4.29 es esperable.

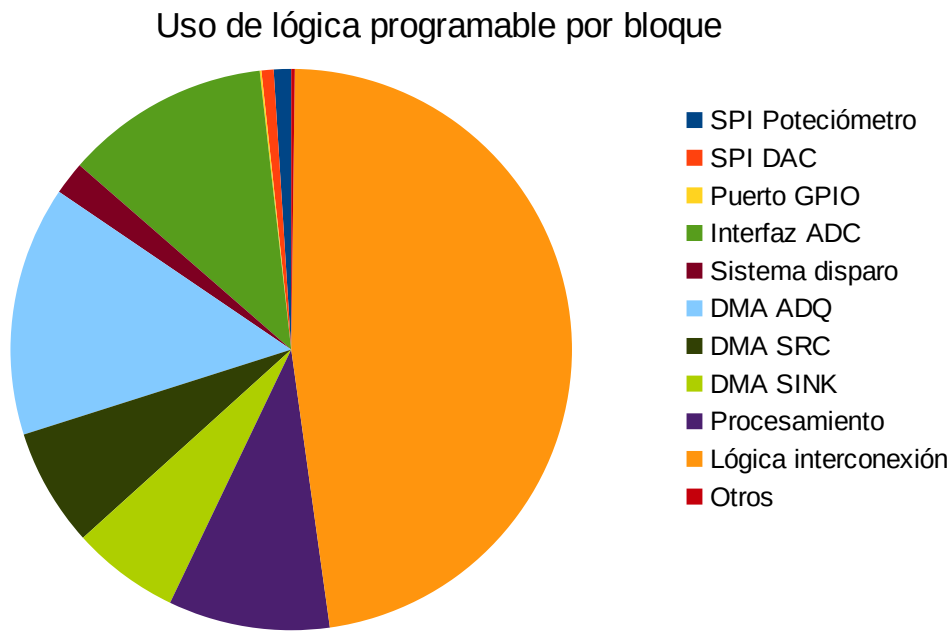


Figura 4.26: Uso porcentual de lógica digital por bloque

En la figura 4.26, se muestra el uso porcentual de cada componente del diseño. El bloque de procesamiento y la interfaz con el ADC tienen el mayor consumo, aproximadamente 10% para ambos. Los motores de acceso directo a memoria son los siguientes en cantidad de recursos. El resto de bloques, debido a su simplicidad, no realiza un uso significativo de la lógica programable.

Los resultados obtenidos indican que existe un elevado margen de recursos disponibles en el dispositivo. Esto resulta de importancia a futuro, ya que denota la posibilidad de incorporar nuevas funcionalidades al diseño. A modo de ejemplo, se pueden implementar algoritmos de procesamiento acelerados por hardware en el flujo de datos en tiempo real. Otra posibilidad se asocia a la implementación de condiciones de disparo más complejas o de operaciones matemáticas entre varios canales.

4.4.2 Análisis temporal

Otro tipo de estadísticas importantes de la síntesis del diseño se encuentran relacionadas al análisis de tiempos. Producto de la ubicación física de los recursos lógicos en la FPGA, existen retardos finitos de propagación para las numerosas señales digitales internas. Cada registro cuenta con requerimientos temporales asociados a su operación, entre los que destacan los tiempos de setup y hold de las señales de datos. El cumplimiento de estos parámetros resulta de vital importancia para asegurar el correcto funcionamiento del diseño [68], evitando metaestabilidad de registros.

La herramienta Timing Analyzer permite analizar sistemáticamente los diversos retardos de propagación internos [67], en base a la definición de los relojes y otros parámetros asociados al

diseño. Durante el proceso de diseño iterativo, el uso de Timing Analyzer permite detectar violaciones de tiempos y resolverlas en base a la información brindada por la herramienta. Por otro lado, para el caso de la síntesis final del diseño, se obtienen parámetros asociados al grado de cumplimiento de los tiempos de setup y hold.

Reloj	Frec. nominal	Slack (setup)	Modelo (setup)	Slack (hold)	Modelo (hold)
Clock principal	50 MHz	8,221 ns	Slow (85 C°)	0,129 ns	Fast (0 C°)
Clock 0 PLL	1 MHz	43,591 ns	Slow (85 C°)	0,105 ns	Fast (0 C°)
Clock 1 PLL	20 MHz	37,459 ns	Slow (0 C°)	0,172 ns	Fast (0 C°)

Tabla 4.30: Slack de setup y hold para el diseño final (Timing Analyzer)

En la tabla 4.30 se presentan los valores de margen (slack) en el cumplimiento de los tiempos de setup y hold de los registros, asociados a cada uno de los tres dominios de reloj del diseño. El software analiza la propagación de señales bajo cuatro modelos diferentes: Slow 0 C°, Slow 85 C°, Fast 0 C° y Fast 85 C°. Es decir, se analizan los cuatro extremos posibles del rango de operación del dispositivo. La herramienta reporta los peores casos posibles, es decir, los caminos de propagación con menor slack, así como también los modelos asociados a cada valor. Por otro lado, Timing Analyzer calcula la frecuencia máxima de operación para cada reloj. Estos valores se listan a continuación:

Reloj	Frec. nominal	Frec. máxima	Relación	Modelo
Clock principal	50 MHz	84,9 MHz	1,7	Slow (85 C°)
Clock 0 PLL	1 MHz	59,3 MHz	59,3	Slow (85 C°)
Clock 1 PLL	20 MHz	85,9 MHz	4,3	Slow (0 C°)

Tabla 4.31: Frecuencias máximas para el diseño final (Timing Analyzer)

De acuerdo a la tabla 4.31, existe cierto margen en el diseño para el aumento de la frecuencia, asociado a los valores de slack positivos. A modo de ejemplo, en principio, se puede elevar la velocidad de procesamiento de datos a 84,9 MSa/s, reduciendo un 41% el retardo entre dos adquisiciones. Con respecto a los relojes de 1 MHz y 20 MHz, su margen se asocia con un posible aumento de la frecuencia de muestreo, aunque esta situación requiere una mayor cantidad de modificaciones en otros elementos del diseño.

Capítulo 5

Software

Este capítulo tiene como objetivo describir los diversos elementos de software que conforman de manera directa o indirecta el proyecto. El primer apartado del capítulo se dedica a la implementación realizada del sistema operativo Linux en la placa de desarrollo. Es decir, se analiza el software de soporte, utilizado como base para el desarrollo en el SoC FPGA. Por otro lado, en la sección 5.2 se describen las herramientas y metodologías de diseño utilizadas durante la fase de desarrollo de código.

El sistema de adquisición está integrado por cuatro componentes de software diferentes: algoritmo de calibración, biblioteca de acceso al hardware, script de servidor y cliente de visualización. El algoritmo de calibración, que se describe en detalle en la sección 5.3, tiene como objetivo estimar y corregir los errores de offset y ganancia asociados al diseño analógico del front-end. En la sección 5.4 se describen los dos componentes de software ejecutados en la plataforma de desarrollo durante la adquisición.

El primer componente del servidor, que se analiza en la sección 5.4.1, consiste en una biblioteca dinámica para el sistema operativo Linux, desarrollada en lenguaje C. La biblioteca actúa como capa de abstracción del hardware, permitiendo el acceso y control de los diversos bloques funcionales implementados en lógica programable. Esta biblioteca es utilizada por el segundo elemento del servidor, analizado en la sección 5.4.2, el cual consiste en un script desarrollado en lenguaje Python, asociado a la comunicación a través de internet con el cliente de visualización. Este componente recibe los comandos con los parámetros de configuración del sistema y envía al cliente los datos adquiridos y procesados.

Finalmente, en el apartado 5.5 se analiza en detalle el cliente de visualización. El desarrollo del mismo se lleva a cabo en lenguaje Python y consiste en una interfaz gráfica multiplataforma, encargada de la visualización en tiempo real de los datos adquiridos. Cabe destacar que el cliente es el elemento más importante del sistema de adquisición, de cara a la interacción con el usuario final.

5.1 Linux embebido

El objetivo de esta sección es describir los diversos elementos de software que conforman la implementación del sistema operativo Linux en la plataforma. La mayoría de estos componentes se encuentran fuertemente relacionados al proceso de arranque del dispositivo. Este mismo, comprende desde la inicialización de los componentes de bajo nivel hasta la ejecución de los procesos del usuario en el sistema operativo.



Figura 5.1: Secuencia de arranque para el caso de sistema operativo (Linux)

En base al diagrama en bloques de la figura 5.1, se describe de manera simplificada la secuencia de eventos involucrada en el arranque [78]:

1. **Boot ROM:** Luego de un reinicio, el SoC FPGA ejecuta un código no modificable provisto por el fabricante y almacenado en la memoria ROM embebida en el chip. El programa realiza la inicialización de bajo nivel, luego carga la siguiente etapa (preloader) en la memoria SRAM interna al dispositivo y procede a su ejecución. El preloader es leído de un medio de almacenamiento permanente externo, en este caso, de una tarjeta SD.
2. **Preloader:** Esta etapa realiza la inicialización de la memoria RAM DDR3 principal. Luego, copia una imagen del bootloader desde la tarjeta SD a la memoria RAM y procede a su ejecución.
3. **Bootloader:** En la implementación realizada, este componente es el encargado de programar la FPGA en base a una imagen binaria almacenada en la tarjeta SD. Luego, el bootloader copia desde la tarjeta SD a RAM el núcleo del sistema operativo y el árbol de dispositivos [79]. Este último, es una descripción de los diferentes componentes de hardware presentes en la plataforma, en un formato comprensible por el núcleo de sistema operativo.
4. **Kernel:** El núcleo de Linux realiza la mayor parte de tareas de inicialización de hardware y de software. A modo de ejemplo, carga y ejecuta los controladores de los periféricos del sistema e inicializa la memoria virtual, gestión de procesos, soporte de sistemas de archivos, pila de protocolos TCP/IP, etc. [80].
5. **Sistema de archivos raíz:** Consiste en una partición con las estructuras de archivos que forman parte del resto del sistema operativo. Luego del arranque, el núcleo monta esta partición y ejecuta el proceso inicial (`/sbin/init`), el cual invoca al resto de programas y servicios del sistema. A modo de ejemplo, se inicializa el servidor SSH, el cual permite conectarse al sistema de manera segura a través de internet y ejecutar comandos [81].

Para contar con una plataforma de desarrollo funcional, resulta necesario implementar en una tarjeta SD todos los componentes involucrados, excepto el código de la memoria ROM. Los diferentes elementos son obtenidos o compilados de acuerdo a las recomendaciones y notas de aplicación del fabricante [78], [82]. El sistema se implementa en una memoria SD de 4 GB, cuyo particionado y disposición se observa en la figura 5.2.

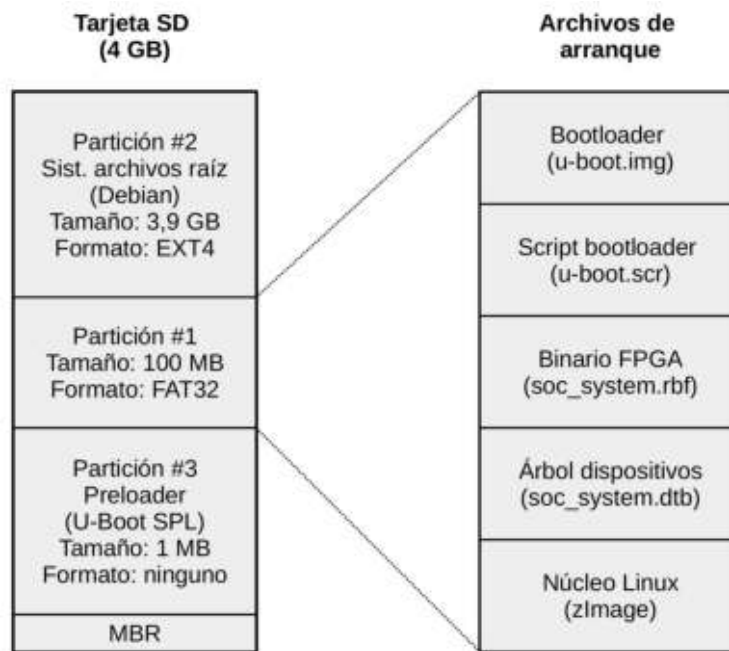


Figura 5.2: Layout de los componentes de software en la tarjeta SD

De acuerdo a los requerimientos de la arquitectura [83], el preloader se almacena en una partición de 1 MB sin formato, es decir, sin sistema de archivos. Por otro lado, el resto de elementos involucrados en el arranque se almacenan en una partición FAT32 de 100 MB. Finalmente, la mayoría del espacio de almacenamiento es dedicado al sistema de archivos raíz, con un tamaño de partición aproximado de 3,9 GB. A continuación, se analizan los componentes utilizados en el diseño y la manera en que son obtenidos:

- Imagen de configuración de la FPGA: La programación de la FPGA se realiza en base a un archivo de formato RBF (Raw Binary File), que surge de la síntesis del diseño digital en el software Quartus Prime.
- Gestores de arranque: Tanto el preloader como el bootloader son generados con la ayuda de la utilidad BSP Editor que forma parte del paquete SoC EDS [82], el entorno de desarrollo de software para los dispositivos SoC FPGA de Intel. BSP Editor se basa en el proyecto de Quartus para generar los archivos de configuración de ambos componentes. La configuración es combinada con el código fuente base (proyecto U-Boot) para compilar y generar ambos cargadores de arranque. Adicionalmente, el bootloader emplea un script con los comandos y parámetros necesarios para programar la FPGA, cargar en RAM el núcleo de Linux y proceder a su ejecución.
- Núcleo y árbol de dispositivos: Se utiliza una versión modificada por el fabricante de la FPGA del núcleo de Linux, en su versión 5.0. El código fuente es obtenido de los repositorios de Intel, configurado de acuerdo a las recomendaciones del fabricante y compilado para la arquitectura ARM del procesador [84]. En base al mismo código fuente, se genera y compila el árbol de dispositivos para la plataforma de desarrollo DE0-Nano-SoC utilizada.

- Sistema de archivos raíz: Con respecto al espacio de usuario, se decide utilizar una versión mínima de la distribución Debian, en la última versión estable a la fecha del proyecto (10.0 Buster). Esta distribución se elige debido a su gran compatibilidad con la arquitectura ARM, ya que cuenta con miles de paquetes compilados para este procesador en sus repositorios. Por otro lado, se tiene en cuenta su estabilidad y la gran disponibilidad de documentación asociada a su popularidad. La versión utilizada, que no cuenta con un entorno gráfico, consume aproximadamente 300 MB de almacenamiento y 40 MB de memoria RAM.

5.2 Herramientas de desarrollo y depuración

Esta sección tiene como objetivo describir las herramientas y metodologías de diseño utilizadas durante la fase de desarrollo de software. En el caso de los componentes de bajo nivel, escritos en lenguaje C, se emplea el entorno de desarrollo integrado Eclipse IDE (Eclipse Foundation). Se trata de una herramienta gratuita y multiplataforma, ampliamente utilizada para el desarrollo y depuración en C, C++ y Java. El editor de texto integrado en Eclipse cuenta con diversas funcionalidades de utilidad, tales como formateo automático de código, autocompletado, análisis sintáctico, entre otras. La interfaz gráfica del software puede ser observada en la figura 5.3, presentada a continuación:

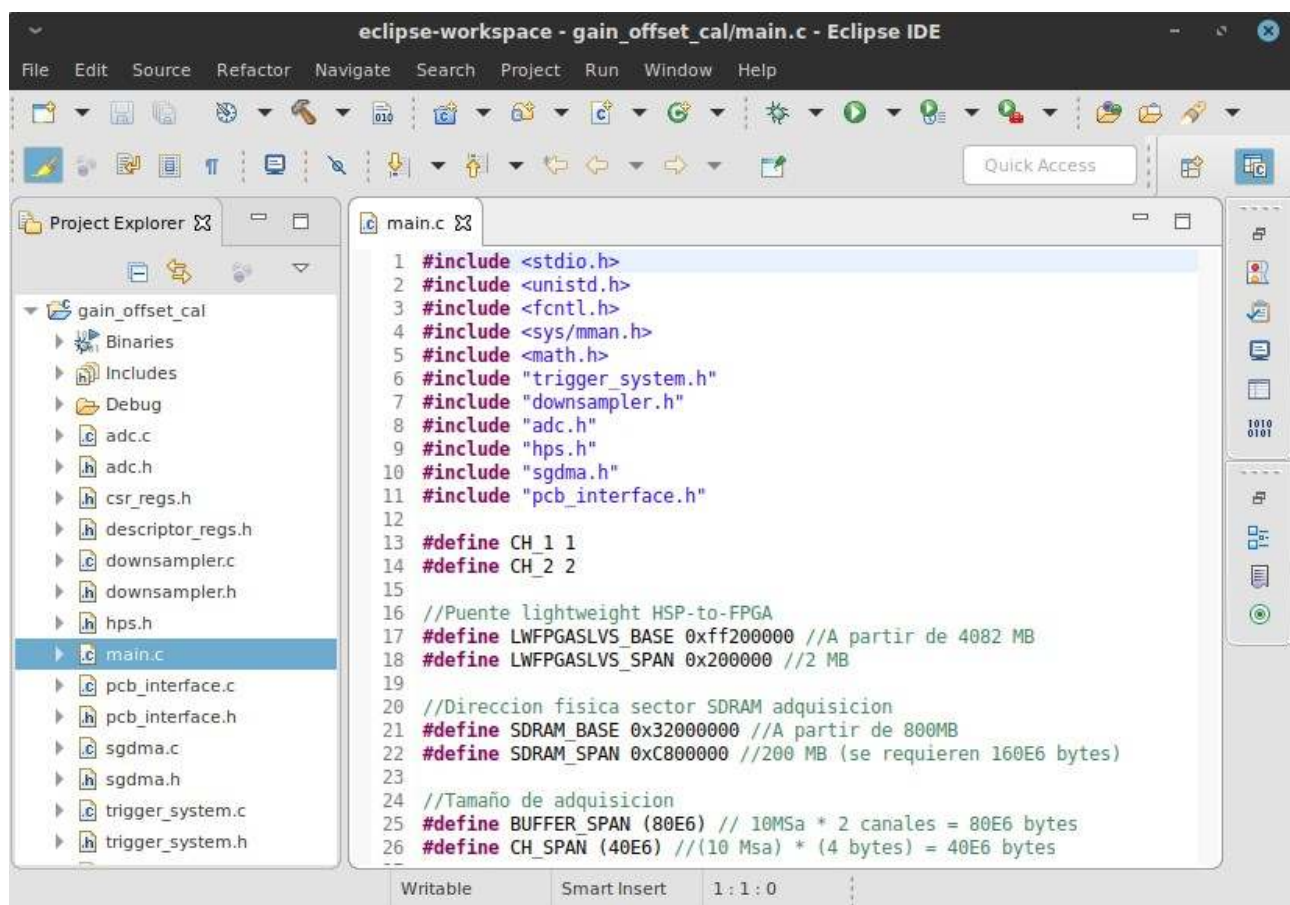


Figura 5.3: Entorno de desarrollo integrado Eclipse IDE

Dado que la arquitectura de la computadora de desarrollo, x86 64 bits, difiere de la plataforma de ejecución, ARM 32 bits, se requiere el uso de una cadena de herramientas o toolchain, que permita la compilación cruzada [85]. El paquete utilizado, Linaro toolchain, cuenta con todos los componentes necesarios para el desarrollo en C y C++ para ARM, incluyendo un compilador, preprocesador, ensamblador y linkeador basados en el proyecto GCC. Adicionalmente, Linaro provee las bibliotecas principales del sistema que permiten el desarrollo de programas Linux para ARM y cuenta con un depurador, basado en el proyecto GDB.

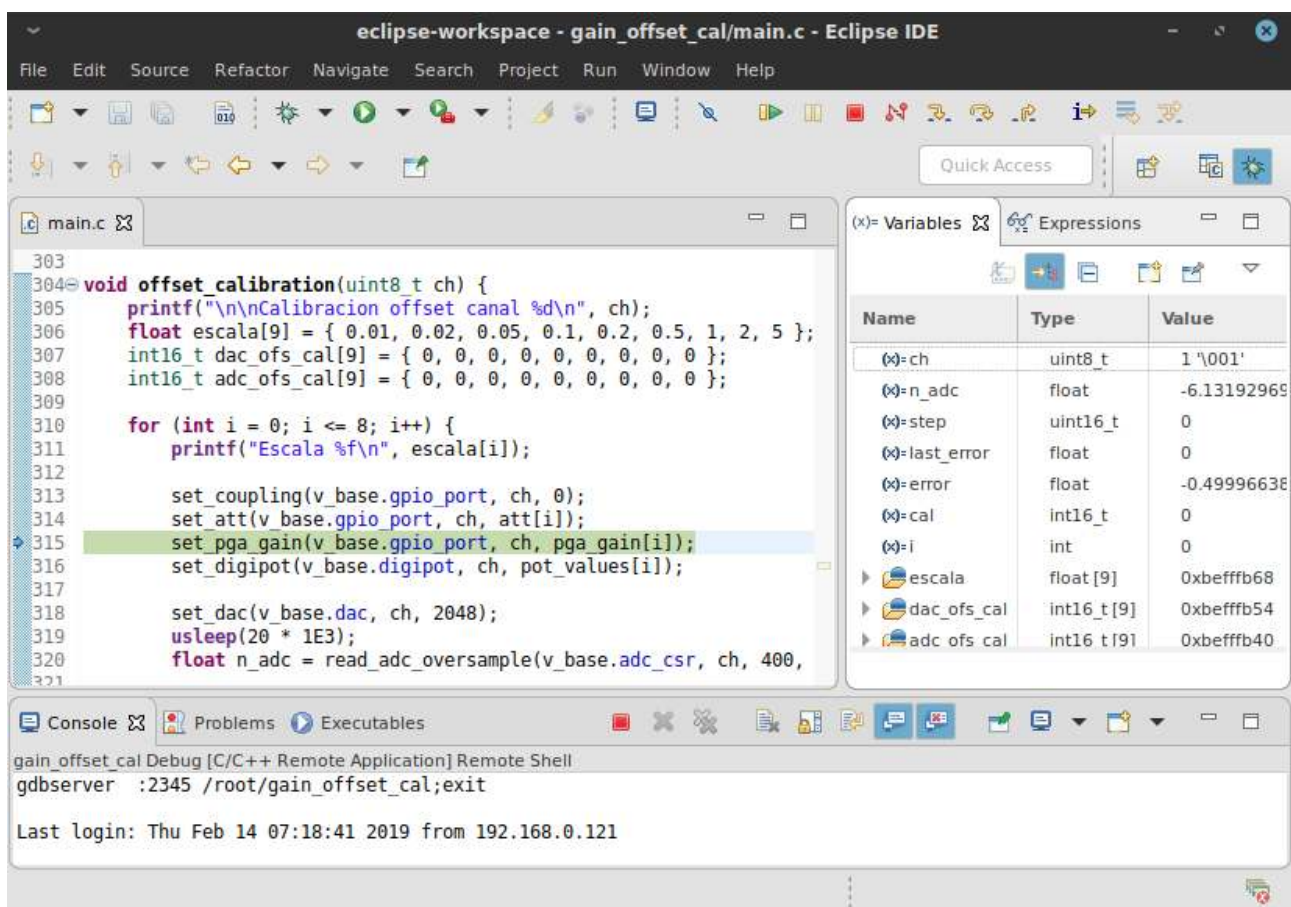


Figura 5.4: Depuración remota a través de conexión Ethernet (Eclipse)

Uno de los motivos por lo cual se utiliza el entorno Eclipse, tiene que ver con la posibilidad de automatizar la depuración remota, en base a la utilidad GDB [86]. En este caso, el programa es capaz de establecer una conexión segura, a través del protocolo SSH sobre TCP/IP, con la plataforma de desarrollo e iniciar una sesión de depuración, como puede ser observado en la figura 5.4. Esta funcionalidad permite hacer uso de la conexión Ethernet de la placa DE0-Nano-SoC para la ejecución paso a paso, ejecución condicional, visualización en tiempo real de variables y sectores de memoria, etc.

Con respecto a los componentes escritos en lenguaje Python, servidor y cliente, se utiliza el editor de texto Visual Studio Code de Microsoft. Se trata de una herramienta libre y multiplataforma,

empleada en el desarrollo en lenguajes interpretados tales como Python y JavaScript. Al igual que Eclipse, el editor ofrece análisis sintáctico, completado inteligente y auto-formateo de código, mediante el uso de diversos plugins. La interfaz gráfica de Visual Studio Code se presenta en la figura 5.5 a continuación:

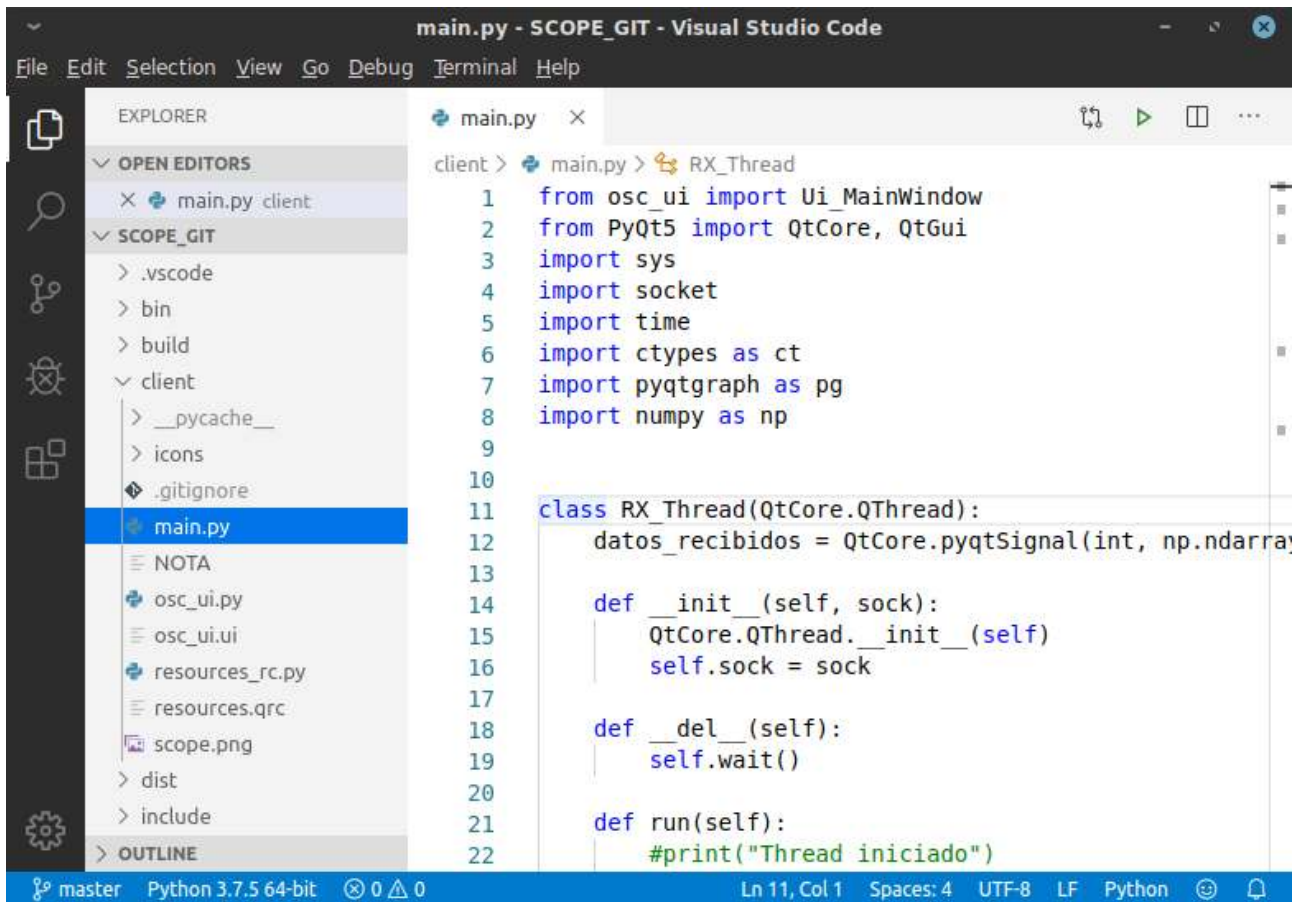


Figura 5.5: Editor de texto Visual Studio Code de Microsoft

Un factor que motiva la elección de este editor es su fuerte integración con el sistema de control de versiones de archivos Git. Esta utilidad facilita el seguimiento de cambios durante el desarrollo del código, permitiendo contar con un historial de las funcionalidades agregadas, ver las diferencias entre dos versiones, revertir cambios, etc. [87]. Estas funcionalidades fueron de utilidad, principalmente, durante la fase de desarrollo del software cliente, debido a su alto grado de complejidad.

5.3 Algoritmo de calibración

Con el objetivo de mejorar la precisión durante el proceso de adquisición, el sistema cuenta con algoritmos que permiten estimar y corregir los errores de offset y ganancia asociados al diseño analógico del front-end. El algoritmo se desarrolla en lenguaje C y consiste en un binario que, al

ser ejecutado en la placa de desarrollo, genera las constantes de calibración utilizadas por el software principal.

El diseño del front-end permite utilizar el DAC para estimar ambos errores, sin necesidad de recurrir a una señal de calibración externa. Con el objetivo de mejorar la efectividad del algoritmo, se emplean componentes analógicos de precisión en todas las etapas involucradas en la generación de la señal de calibración. A modo de ejemplo, se utilizan resistencias con tolerancia del 0,1% para el buffer de voltaje del DAC, amplificador sumador de offset y atenuadores de entrada.

El algoritmo diseñado obtiene, de manera iterativa, los parámetros de calibración de cada canal para las nueve escalas verticales implementadas. La lectura del voltaje en los ADC es realizada directamente a través de los registros implementados en la lógica programable asociada a los conversores. Con el objetivo de mejorar la medición del voltaje, se realiza un promedio de 400 valores a lo largo de dos ciclos de línea (40 ms). Por otro lado, luego de cada cambio en la configuración del front-end, el algoritmo espera 20 ms de manera que se establezca la tensión a medir. Cabe destacar que los parámetros temporales propuestos fueron analizados y verificados experimentalmente.

5.3.1 Offset

De acuerdo al diseño teórico de la etapa analógica, una constante del DAC de 2048 (2,5 V) se asocia a la situación en la que la tensión sumada a la señal de entrada es nula. Bajo estas condiciones, si no existe voltaje de entrada, se espera leer un valor numérico en el ADC de 2048 (1,65 V) para todas las escalas. La diferencia entre el valor medido en el conversor y el valor teórico esperado representa el error de offset que se busca minimizar.

Se implementa un algoritmo que busca de manera iterativa el valor numérico que, al ser programado en el DAC, minimiza el desajuste de tensión de cada escala. De esta manera, se realiza una calibración analógica inicial del sistema mediante hardware. El error remanente, es decir, la diferencia entre el valor medido en el ADC bajo estas condiciones y 2048, es eliminado mediante las operaciones aritméticas implementadas en la lógica programable.

De lo mencionado en el párrafo anterior, se deduce que el algoritmo actúa en dos fases, generando dos factores de corrección para cada escala vertical de ambos canales. Sea N_{DAC} el valor que, programado en el DAC, minimiza el offset de tensión, la primera constante de calibración se define a continuación:

$$N_{CAL\ DAC} = N_{DAC} - 2048 \quad (5.1)$$

El factor obtenido por el algoritmo representa la diferencia entre el valor ideal de 2048 y el valor que genera el menor error de offset. Debido a su definición, se trata de un número pequeño, cuyo signo depende de la dirección del desajuste. Por otro lado, sea N_{ADC} el valor numérico obtenido en el conversor analógico digital bajo estas condiciones, la segunda constante se calcula de acuerdo a:

$$N_{CAL ADC} = N_{ADC} - 2048 \quad (5.2)$$

En este caso, la constante se asocia al error absoluto remanente luego de la calibración mediante el DAC. Es decir, representa la diferencia entre el valor medido en el ADC y el valor ideal de 2048, asociado al voltaje de mitad de escala de 1,65 V. Nuevamente, se trata de un número pequeño, cuyo signo depende de la dirección del desajuste.

5.3.2 Ganancia

Una vez corregido el error absoluto, el diseño propuesto utiliza el DAC para generar una excursión a la entrada del ADC y estimar la ganancia real del front-end. De esta forma, se calcula un factor de corrección, asociado a la relación entre la ganancia medida y la teórica esperada. Este factor es aplicado en las operaciones aritméticas realizadas en la lógica programable.

Los valores de excursión del DAC utilizados permiten lograr el mayor rango dinámico posible a la entrada del ADC, mejorando la precisión de los parámetros de calibración estimados. Adicionalmente, a fin de asegurar la linealidad de las etapas, se tiene en cuenta el rango de operación de ambos conversores, limitando la excursión entre un 10% y 90% de la escala completa de ambos. En la tabla 5.1, se listan los valores de excursión utilizados para cada escala, obtenidos en base a los criterios mencionados.

Escala (V/div)	N _{pp} DAC	N _{pp} ADC	G _{TOT} (V/V)	G _{DAC} (V/V)
5	3110	1861,59	0,06	0,4
2	3110	2326,99	0,075	0,5
1	2488	3723,18	0,15	1
0,5	1244	3723,18	0,3	2
0,2	498	3726,17	0,75	5
0,1	2206	3721,73	1,5	1,11
0,05	1104	3725,11	3	2,22
0,02	440	3711,61	7,5	5,56
0,01	220	3711,61	15	11,11

Tabla 5.1: Parámetros utilizados en la calibración de ganancia

El rango pico a pico en el DAC, listado en la primera columna de la tabla 5.1, es aplicado alrededor del valor central programado en el DAC, el cual es determinado en base a la calibración de offset. Por otro lado, en la segunda columna se listan los valores teóricos esperados a la entrada del ADC, de acuerdo a la ganancia ideal entre ambos conversores (última columna). Durante la ejecución del algoritmo, las constantes de calibración de ganancia de las escalas verticales se calculan en base a la siguiente definición:

$$G_{CAL} = \frac{N_{pp ADC} (teórico)}{N_{pp ADC} (medido)} \quad (5.3)$$

Es decir, el parámetro de calibración G_{CAL} surge de la relación entre la excursión numérica en el ADC esperada y la medida bajo condiciones reales. La multiplicación de la señal de entrada por el

factor de calibración anula la variación de ganancia asociada al front-end. De acuerdo a la definición de la ecuación 5.3, un valor de G_{CAL} mayor a la unidad representa una ganancia en el front-end menor a la teórica esperada. Por otro lado, si la ganancia real es menor que la ideal, el factor será mayor a la unidad.

5.3.3 Aplicación de los factores

Esta sección representa el último apartado dedicado a la calibración y tiene como objetivo describir la manera en que las constantes son aplicadas durante el funcionamiento del sistema de adquisición. El parámetro $N_{CAL\ DAC}$, asociado al ajuste de offset por hardware, es utilizado para determinar el valor a ser programado en el DAC, de acuerdo a la siguiente expresión:

$$N_{DAC} = 2048 + N_{CAL\ DAC} + N_{VOFS} \quad (5.4)$$

El primer término de la ecuación 5.4, representa el valor teórico ideal a ser programado en el DAC, el segundo, realiza el ajuste de offset y el tercero se asocia al voltaje sumado de manera analógica a la señal de entrada. Este último término es calculado por el software del sistema de adquisición, en base al ajuste que se realiza en el cliente de visualización.

Con respecto a los otros dos parámetros obtenidos por el algoritmo, como fue mencionado, actúan sobre el módulo de interfaz de los ADC, implementado en la lógica programable. De acuerdo a lo analizado en la sección 4.3.3, este componente tiene como objetivo convertir las muestras de los ADC, de 12 bits, en un número que represente la tensión de la señal de entrada. El diagrama en bloques de las operaciones involucradas se observa a continuación:

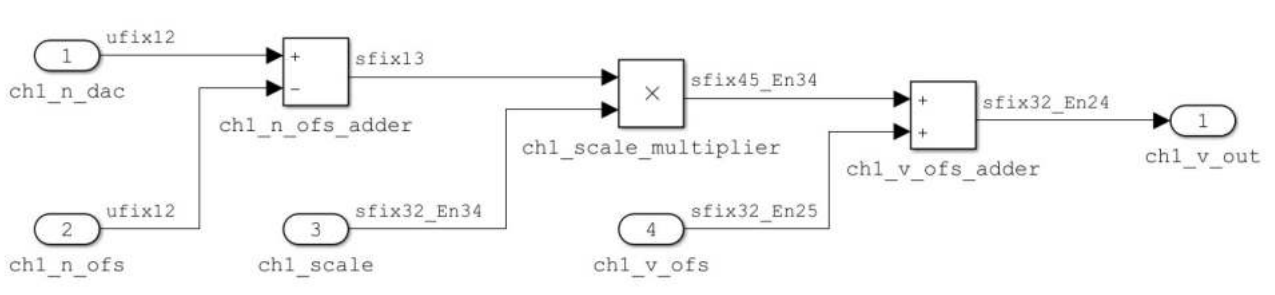


Figura 5.6: Conversión de datos realiza en el módulo de interfaz de ADC

$$V_{OUT} = (N_{DAC} - N_{OFS}) \cdot G_{SCALE} + V_{OFS} \quad (5.5)$$

Las operaciones matemáticas en la figura 5.6 pueden ser representadas mediante la ecuación 5.5. Durante el funcionamiento del sistema de adquisición, la eliminación del offset remanente es realizada en el restador de entrada. El parámetro de calibración asociado, es tenido en cuenta en el cálculo de la constante N_{OFS} , de acuerdo a la siguiente expresión:

$$N_{OFS} = 2048 + N_{CAL\ DAC} \quad (5.6)$$

Por otro lado, el ajuste de ganancia se lleva a cabo en el multiplicador, afectando el valor teórico necesario para cada escala (G_{TOT}) por el parámetro obtenido en base al algoritmo (G_{CAL}). Esta situación se observa a continuación en la ecuación 5.7:

$$G_{SCALE} = G_{CAL} \cdot G_T \quad (5.7)$$

En el capítulo 6, asociado a la verificación y validación del diseño, se analiza el funcionamiento del algoritmo bajo condiciones reales. Con el objetivo de verificar la efectividad del algoritmo, se efectúan diversas mediciones antes y después de la calibración. Adicionalmente, se listan todas las constantes obtenidas durante el ajuste de offset y ganancia.

5.4 Servidor

En este apartado se describe el software implementado en el dispositivo SoC FPGA asociado al funcionamiento del sistema de adquisición. El diseño consiste en una biblioteca dinámica, escrita en lenguaje C, relacionada al acceso y control de los componentes implementados en la lógica programable. Por otro lado, la recepción de comandos del cliente y el envío de datos a través de internet se lleva a cabo mediante un script desarrollado en Python.

Resulta pertinente aclarar que sería posible implementar el servidor en un único lenguaje. Sin embargo, el empleo de dos componentes permite simplificar el diseño y mejorar el rendimiento. Por un lado, debido a su naturaleza de bajo nivel, el lenguaje C brinda un mayor grado de control por sobre las operaciones ejecutadas y el manejo de la memoria del sistema. A modo de ejemplo, las funcionalidades asociadas al lenguaje simplifican el acceso a los buffers de adquisición y el control de los diversos componentes implementados en la lógica programable.

Por otro lado, dado que el cliente está desarrollado en Python, la utilización del mismo lenguaje en el servidor facilita la implementación del protocolo de comunicación entre ambas partes. A modo de ejemplo, resulta posible utilizar las estructuras de datos propias del lenguaje Python para la codificación de los diversos parámetros de adquisición que forman parte de los comandos enviados por el cliente.

5.4.1 Comunicación con el hardware

La comunicación con el hardware es efectuada a través de una biblioteca dinámica para el sistema operativo Linux [88], desarrollada y compilada en el entorno Eclipse. Este elemento actúa como una capa de abstracción de hardware, exponiendo al script de Python una serie de funciones que permiten controlar la adquisición a un mayor nivel de abstracción. La biblioteca consiste en un archivo binario de extensión .so (shared object), cargado en memoria durante la inicialización del script de Python.

La biblioteca accede a dos sectores en el espacio de memoria física del procesador. En primer lugar, el control de los componentes del sistema de adquisición es efectuado a través de sus registros de estado y control, implementados en el puente lightweight HPS-to-FPGA. Por otro lado, los datos adquiridos y procesados son leídos de los buffers implementados en los últimos 200 MiB de memoria RAM. Dado que el sistema operativo emplea un esquema de memoria virtual, los dos

sectores de memoria física mencionados son mapeados al espacio virtual, mediante los mecanismos provistos por el sistema operativo [89].

En la estructura interna de la biblioteca se implementan, para bloque funcional del sistema de adquisición, una serie de rutinas que permiten el acceso a las funcionalidades más relevantes del componente. En la tabla 5.2, se listan las rutinas asociadas a los tres bloques más importantes de la arquitectura de adquisición: interfaz de los ADC, sistema de disparo y bloque de procesamiento.

Nombre	Componente	Descripción
set_adc_nofs()	Interfaz ADC	Ajuste de la constante Nofs aplicada en las operaciones aritméticas de un canal en particular.
set_adc_scale()	Interfaz ADC	Ajuste del factor de escalado Gscale aplicado en las operaciones aritméticas de un canal en particular.
set_adc_vofs()	Interfaz ADC	Ajuste de la constante Vofs aplicada en las operaciones aritméticas de un canal en particular.
set_adc_state()	Interfaz ADC	Habilitación y deshabilitación del flujo de datos a la salida de la interfaz (Avalon-ST).
read_adc_data()	Interfaz ADC	Lectura de los datos numéricos asociados al ADC de un canal en particular (12 bits).
read_adc_v_data()	Interfaz ADC	Lectura de los datos numéricos de un canal luego de las operaciones aritméticas (32 bits).
trig_reset()	Sistema disparo	Reinicio de la maquina de disparo a su estado inicial (estado de reset).
set_trig()	Sistema disparo	Selección del modo de disparo (flanco ascendente, descendente, etc.) y el canal utilizado.
set_trig_level()	Sistema disparo	Ajuste de los dos umbrales de voltaje utilizados para evaluar el evento de disparo.
set_trig_n()	Sistema disparo	Ajustes de la cantidad de muestras de pre-cuenta (Npre) y de pos-cuenta (Npos).
get_trig_state()	Sistema disparo	Devuelve el estado actual de la maquina de disparo (pre-cuenta, armado, pos-cuenta, etc.).
downsampler_reset()	Procesamiento	Reinicio de la lógica interna del bloque de decimación (sistema de procesamiento).
set_downsampler()	Procesamiento	Ajuste del factor de decimación D y selección del algoritmo utilizado (normal, alta resolución, etc.).

Tabla 5.2: Funciones de la biblioteca asociadas a la arquitectura de adquisición

Para el caso de los tres bloques DMA, debido a la complejidad del componente, se utilizan las bibliotecas provistas por el fabricante del SoC FPGA. Por otro lado, el control y configuración del front-end analógico se lleva a cabo en base a las funciones listadas a continuación:

Nombre	Componente	Descripción
set_dac()	SPI DAC	Ajuste de la constante numérica del DAC asociada a un canal en particular (12 bits).
set_digipot()	SPI potenciómetro	Ajuste de la constante del potenciómetro digital asociada un canal en particular (8 bits).
set_coupling()	Puerto GPIO	Selección del acoplamiento utilizado a la entrada de un canal en particular (alterna o continua).
set_att()	Puerto GPIO	Selección del atenuador utilizado a la entrada de un canal en particular (0,1 veces o 0,9 veces).
set_pga_gain()	Puerto GPIO	Selección del valor de ganancia utilizado en el PGA de un canal en particular (1, 2, 5 o 10 veces).
set_led()	Puerto GPIO	Selección del estado del LED de comunicación del PCB del front-end analógico.

Tabla 5.3: Funciones de la biblioteca asociadas al control del front-end analógico

Como se puede observar en las tablas 5.2 y 5.3, los métodos implementados permiten acceder, de manera sencilla, a las características más importantes de cada bloque. Es decir, se brinda cierto grado de abstracción, a nivel de componente, sobre la implementación del sistema. En base a estas rutinas, se implementan las funciones que actúan de interfaz externa para la biblioteca, presentadas en la tabla 5.4.

Nombre	Descripción
system_init()	Realiza el mapeo en memoria virtual de los sectores asociados a los registros de los componentes y los buffers de adquisición. Luego, reinicia los bloques implementados en la lógica programable, llevándolos a un estado por defecto conocido.
config_update()	Recibe como argumento una estructura con todos los parámetros asociados a una nueva adquisición. Realiza la configuración del front-end y de los diversos bloques implementados en la lógica programable.
start_adquisition()	Inicia el proceso de adquisición continua en el buffer circular. La detección de los eventos de disparo y el control de flujo es realizado mediante hardware. La función no recibe ningún parámetro, debido a que el sistema es configurado previamente mediante la llamada a config_update().
get_trigger_state()	Permite acceder al estado actual de la máquina de disparo. De esta manera, se puede determinar el momento en el que finaliza la adquisición de los datos, iniciada mediante la llamada a start_adquisition().
stop_acquisition()	Detiene el proceso de adquisición inicial en curso, sin necesidad que ocurra un disparo y finalice la captura. Luego, realiza un reset de los componentes de la lógica programable involucrados. De esta manera, se puede actualizar la configuración del sistema e iniciar una nueva captura.
resample()	Realiza el procesamiento y decimación de los datos de un canal en particular, actuando sobre las señales de control de los motores DMA involucrados. Esta función es llamada luego de una adquisición inicial exitosa, para cada canal que se encuentre habilitado. Ante la finalización de la transferencia de los datos, la función devuelve un vector con las muestras del canal procesadas.

Tabla 5.4: Funciones principales de interfaz de la biblioteca

Estas rutinas generan un mayor grado de abstracción, operando por sobre todos los componentes del diseño. De esta forma, se brinda una interfaz que permite acceder a la mayoría de las funcionalidades del sistema de adquisición. En la sección 5.4.3 se analiza la manera en que las seis funciones de la tabla 5.4 son utilizadas por el script del servidor para implementar la funcionalidad requerida por el cliente.

5.4.2 Comunicación con el cliente

El segundo componente de software del servidor consiste en un script desarrollado en el lenguaje interpretado Python, en su versión 3.7 [90], cuya funcionalidad principal es actuar de interfaz con el cliente de visualización. La comunicación es realizada a través de dos sockets TCP [91] operando en puertos diferentes. El primer socket es dedicado a la recepción de los comandos de configuración enviados por el cliente. Por otro lado, el segundo socket es utilizado para el envío de los datos luego de ser adquiridos y procesados.

Así mismo, se emplea una arquitectura basada en dos hilos diferentes de ejecución [92]. El hilo principal se asocia al control del sistema de adquisición y el envío de las muestras procesadas a través del socket de datos. El segundo hilo, por otra parte, se ejecuta permanentemente en segundo plano y cumple la funcionalidad de recibir los parámetros de adquisición a través del

socket de comandos. Esta implementación permite mejorar los tiempos de respuesta del sistema ante un cambio, por parte del cliente, en la configuración de adquisición. En la figura 5.7, se puede observar el proceso de inicialización del servidor, asociado a la ejecución del script a través de una terminal SSH.



```
jordi@JORDI-PC:~$ ssh root@192.168.0.120
Linux SOC-FPGA 5.0.0 #1 SMP Sat Apr 20 17:44:44 -03 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 14 07:15:05 2019 from 192.168.0.121
root@SOC-FPGA:~# python3 server_script.py
Librería cargada correctamente
Inicialización del sistema correcta
Servidor escuchando en: 192.168.0.120
Conexión recibida desde: 192.168.0.121
Comando recibido
```

Figura 5.7: Ejecución del script del servidor a través de una terminal SSH

El acceso a la biblioteca de bajo nivel se realiza a través del módulo `ctypes` de Python. Mediante el uso de `ctypes`, se pueden invocar funciones que forman parte de una biblioteca dinámica cargada en memoria [93]. El módulo cuenta con diversos métodos, clases y atributos que permiten realizar la conversión entre las estructuras de datos del lenguaje C y las estructuras asociadas al lenguaje Python. De esta forma, se permite la conversión de objetos nativos de Python en argumentos para las funciones invocadas y obtener objetos de Python en base a los valores retornados por la función implementada en C.

Con respecto a la codificación de los datos utilizada en la comunicación a través de internet, las muestras de cada canal se agrupan para su envío en un vector de enteros de 32 bits. Se emplea, para cada valor del arreglo, la misma representación en punto fijo de 32 bits que en la lógica programable. Por otro lado, el cliente envía todos los parámetros asociados a una nueva adquisición a través de un diccionario de Python. Esta estructura de datos permite almacenar un conjunto no ordenado de pares clave-valor, en el que cada valor puede pertenecer a un tipo de dato diferente [94]. Teniendo en cuenta lo mencionado, esta clase de objeto resulta conveniente para almacenar y enviar los diversos parámetros de configuración del sistema.

5.4.3 Diagrama de flujo

Finalmente, con el objetivo de concluir la descripción del servidor, se presenta en la figura 5.8 un diagrama de flujo del proceso de adquisición. En base a este esquema se describen las interacciones entre ambos componentes del servidor y la manera en que se implementa la funcionalidad solicitada por el cliente.

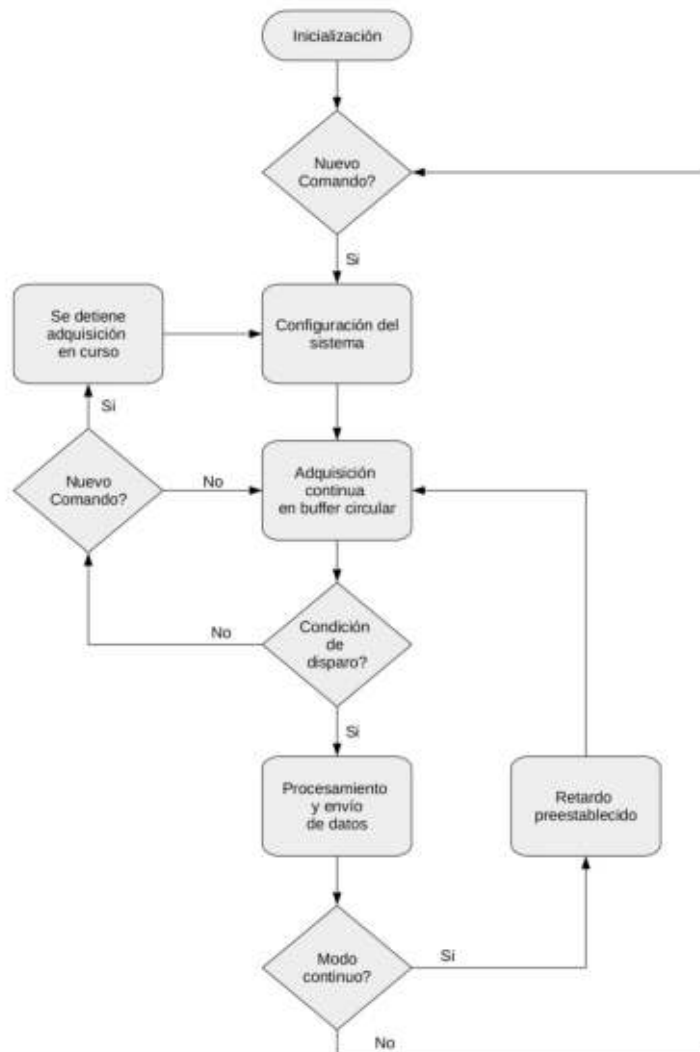


Figura 5.8: Diagrama de flujo del proceso de adquisición

El primer estado del diagrama de la figura 5.8 se asocia a la inicialización del servidor. Durante esta fase, el script carga en memoria la biblioteca dinámica y realiza el llamado a la función `system_init()`, destinada a la inicialización de bajo nivel de hardware y software. Luego, se crean y configuran los sockets TCP de datos y comandos. En esta instancia, el cliente establece la comunicación con el servidor y envía el comando inicial de adquisición. El script del servidor, por su parte, acepta la conexión y utiliza el método `config_update()` de la biblioteca para configurar el hardware del sistema de adquisición.

Un llamado a `start_acquisition()` inicia el proceso de captura continua en el buffer circular. Durante esta fase, el servidor monitorea continuamente el estado de disparo, haciendo uso de la rutina `get_trigger_state()` de la biblioteca. En el caso que ocurra un disparo exitoso y finalice la adquisición, se realiza un llamado al método `resample()` para procesar las muestras de cada canal habilitado, luego de lo cual, se envían los datos al cliente.

Por otro lado, si el cliente envía un nuevo comando antes de la finalización del ciclo actual de adquisición, se detiene la captura en curso, se modifica la configuración del sistema y se inicia una nueva captura. En este caso, las rutinas de la biblioteca involucradas en el control son: `stop_acquisition()`, `config_update()` y `start_acquisition()`.

El cliente puede solicitar que se realice una única captura de datos, modo de operación denominado “single”. Bajo esta situación, luego del envío de datos el servidor regresa a un estado de reposo, a la espera de un nuevo comando. Por otro lado, el cliente puede solicitar que se realicen múltiples capturas de manera continua. En este caso, luego de un retardo preestablecido, el servidor inicia una nueva adquisición, proceso que se repite de manera cíclica.

El retardo entre dos capturas consecutivas sólo se aplica en las escalas de tiempo más pequeñas y tiene como objetivo limitar la velocidad de adquisición a un máximo de 50 capturas por segundo. Este valor se elige experimentalmente buscando optimizar diversos parámetros asociados al comportamiento y fluidez del sistema. A modo de ejemplo, la velocidad de captura influye en el ancho de banda utilizado para el envío de datos, el consumo de recursos en el cliente y los tiempos de respuesta ante un cambio de comando.

5.5 Cliente

El elemento más importante del sistema de adquisición, de cara a la interacción con el usuario final, es el cliente de visualización. El cliente permite configurar y ajustar de manera gráfica los diversos parámetros de adquisición y realizar la visualización en tiempo real de los datos adquiridos para ambos canales. En su versión final, el cliente consiste en un script de Python, capaz de ejecutarse en los tres sistemas operativos de escritorio más importantes: Windows, Linux y Mac OS.

El desarrollo de la interfaz gráfica de usuario se basa en el framework Qt, en su última versión estable a la fecha del proyecto (5.13). El proyecto Qt consiste en un conjunto de bibliotecas y utilidades libres y gratuitas, utilizadas para el desarrollo de interfaces gráficas multiplataformas. Se trata de un framework orientado a objetos, de código abierto, cuyo lenguaje nativo es C++. El acceso a las bibliotecas del proyecto Qt por parte del lenguaje Python se lleva a cabo a través de un módulo denominado PyQt.

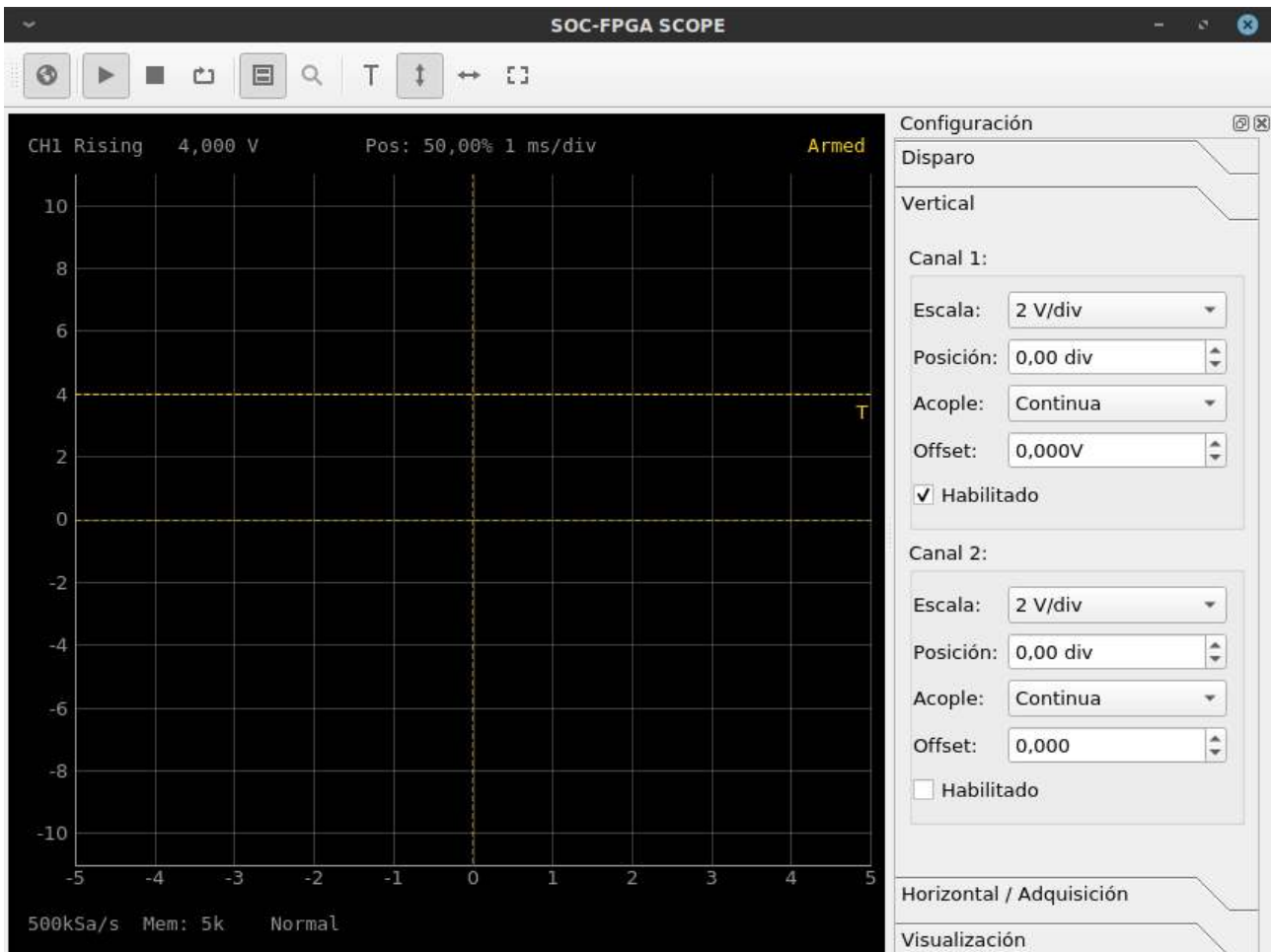


Figura 5.9: Interfaz gráfica del cliente de visualización

En la figura 5.9, se observa el diseño de la interfaz del cliente de visualización. Los diversos elementos gráficos que componen la interfaz se encuentran agrupados en tres sectores diferentes: barra de herramientas, área de visualización y panel de ajustes. La barra de herramientas, ubicada en la parte superior, está compuesta por una serie de íconos que brindan acceso a las funcionalidades o acciones más importantes. El área de visualización, ubicada en la parte inferior izquierda, cuenta con todos los elementos gráficos asociados a la visualización de las formas de onda y el estado del sistema. Por último, todos los ajustes del sistema se encuentran agrupados en las cuatro pestañas del panel de ajustes principal, ubicado en la parte inferior derecha de la ventana.

5.5.1 Área de visualización

La implementación del área de visualización se basa en la biblioteca PyQtGraph, en su última versión estable a la fecha del proyecto (0.10.0). Se trata de un módulo de Python para la generación de gráficos 2D y 3D, basado en el proyecto PyQt. Se hace uso de esta biblioteca debido a su alto grado de optimización y bajo consumo de recursos, factores de vital importancia en una aplicación de visualización en tiempo real. Por otro lado, para la representación vectorial

de los datos numéricos y las operaciones matemáticas, se emplea la biblioteca de cálculo científico NumPy.

En base a las diversas funcionalidades de PyQtGraph, se implementan un conjunto de elementos gráficos que complementan la visualización de los datos adquiridos. En la figura 5.10 se observa la disposición en el área de visualización de estos elementos, para el caso en que únicamente se encuentra habilitado un canal del sistema de adquisición.

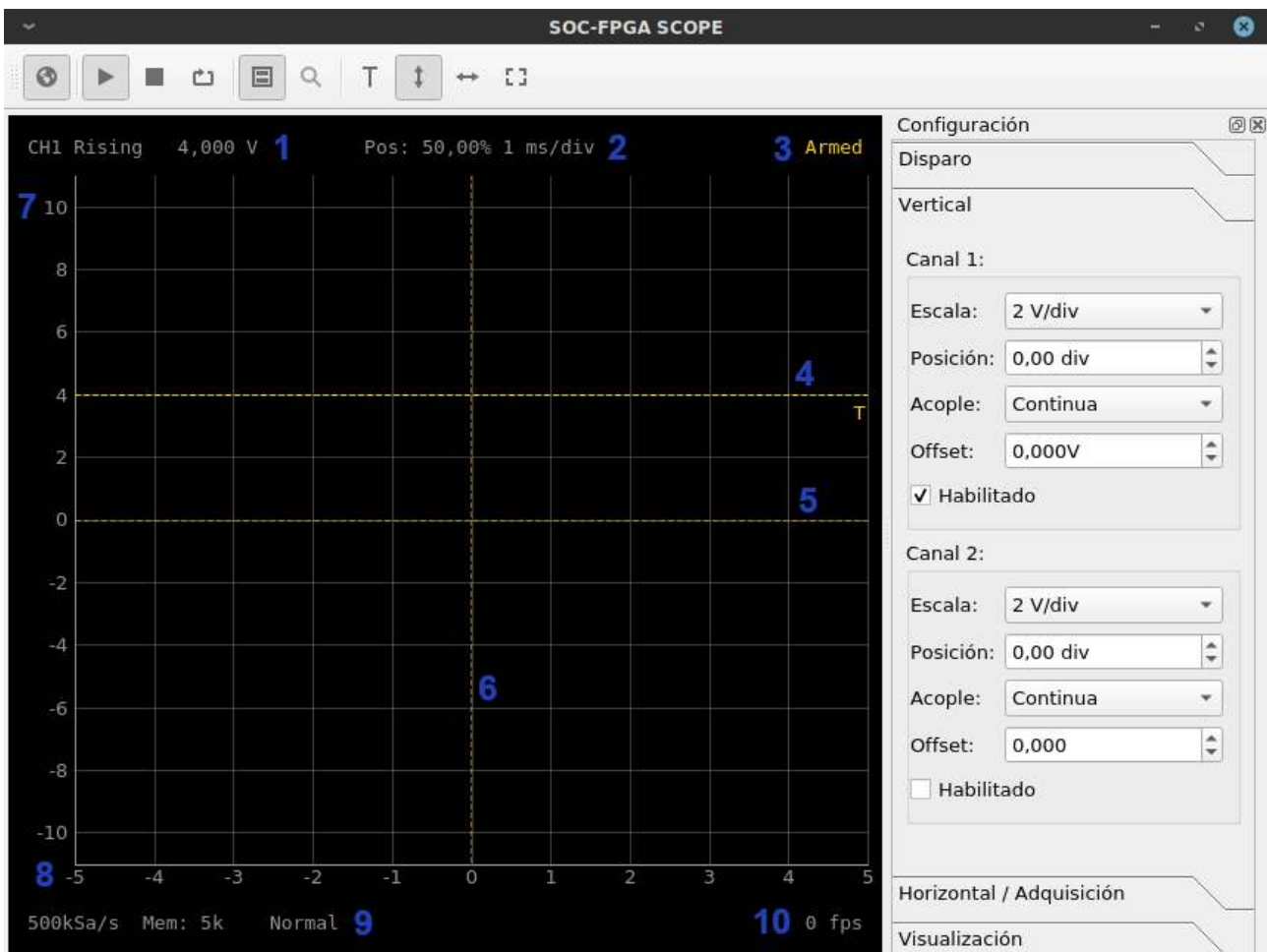


Figura 5.10: Elementos gráficos que componen el área de visualización

En base a la identificación numérica de la figura 5.10, se realiza una breve descripción de cada componente del área de visualización y su funcionalidad asociada:

1. Ajustes de disparo: Etiqueta de texto que brinda información simplificada sobre la configuración actual del sistema de disparo. Mediante su lectura, se puede identificar el canal que actúa como fuente, el modo de disparo y el nivel seleccionado.
2. Ajustes de base de tiempo: Etiqueta de texto que permite identificar la base de tiempo seleccionada y el ajuste porcentual de posición horizontal utilizado.

3. Estado de disparo: Etiqueta de texto que indica, de manera simplificada, el estado actual del sistema de adquisición. Los valores posibles son armed (armado), triggered (disparado) y stopped (detenido).
4. Marcador de nivel de disparo: Línea horizontal que indica, de manera visual, el umbral de tensión utilizado como nivel de disparo.
5. Marcador de posición vertical: Línea horizontal que indica, de manera visual, el ajuste de posición vertical del canal correspondiente.
6. Marcador de posición horizontal: Línea vertical que indica, de manera visual, el instante temporal en el que ocurre el evento de disparo.
7. Escala numérica vertical: Valores numéricos asociados al voltaje de cada una de las 10 divisiones verticales de la grilla de visualización. Estos valores dependen, principalmente, de los ajustes de escala y posición vertical del canal correspondiente.
8. Escala numérica horizontal: Valores numéricos asociados a la relación temporal de cada una de las 10 divisiones horizontales de la grilla de visualización, con respecto a la referencia brindada por el evento de disparo. Estos valores dependen de los ajustes de base de tiempo y posición horizontal utilizados.
9. Ajustes de adquisición: Etiqueta de texto que brinda información simplificada sobre la configuración actual de la etapa de procesamiento de los datos adquiridos. Mediante su lectura, se puede identificar el algoritmo de decimación aplicado, la profundidad de memoria y la frecuencia de muestreo equivalente para los datos procesados.
10. Velocidad de captura: Etiqueta de texto asociada a la cantidad de capturas realizadas por segundo. La lógica del software cliente calcula este valor en base a la frecuencia con la que se actualizan los datos del área de visualización.

Como parte de las funcionalidades relacionadas al cliente, se implementan dos modos de visualización diferentes. En el primer caso, cada canal del sistema de adquisición se visualiza en un área independiente, por lo que este modo de funcionamiento se denomina multi-plot (gráficos múltiples). En el segundo caso, ambos canales se visualizan de manera superpuesta, en una misma grilla de divisiones. Para esta configuración se utiliza la denominación single-plot, debido a la existencia de un único gráfico. Cabe destacar que, a fin de diferenciar las formas de onda y elementos gráficos pertenecientes a cada canal, se emplean colores diferentes para la representación.

La figura 5.10, utilizada para el análisis del área de visualización, se corresponde con el modo de operación multi-plot, para el caso en que se encuentra únicamente habilitado el primer canal. Bajo esta modalidad, la habilitación del segundo canal genera la siguiente disposición para el área de visualización:

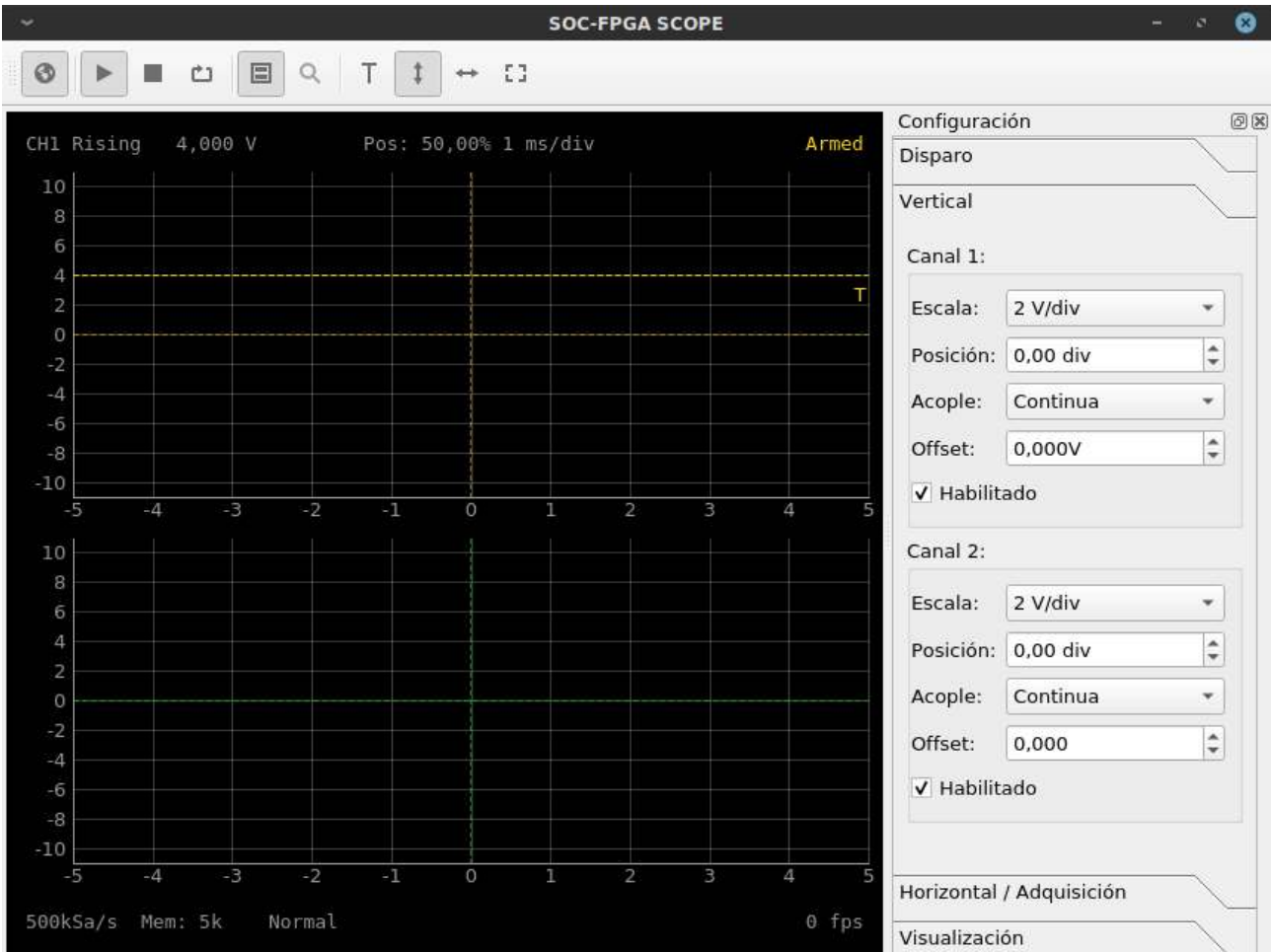


Figura 5.11: Modo de visualización multi-plot

Como se puede observar en la figura 5.11, el área dedicada a cada canal cuenta con los siguientes elementos independientes: grilla de visualización, marcadores de posición y escalas numéricas horizontales y verticales. Sin embargo, existe un único indicador de nivel de disparo, asociado al gráfico del canal que actúa como fuente de disparo. Por otro lado, en la figura 5.12 se observa la disposición de la interfaz del cliente para el modo single-plot.

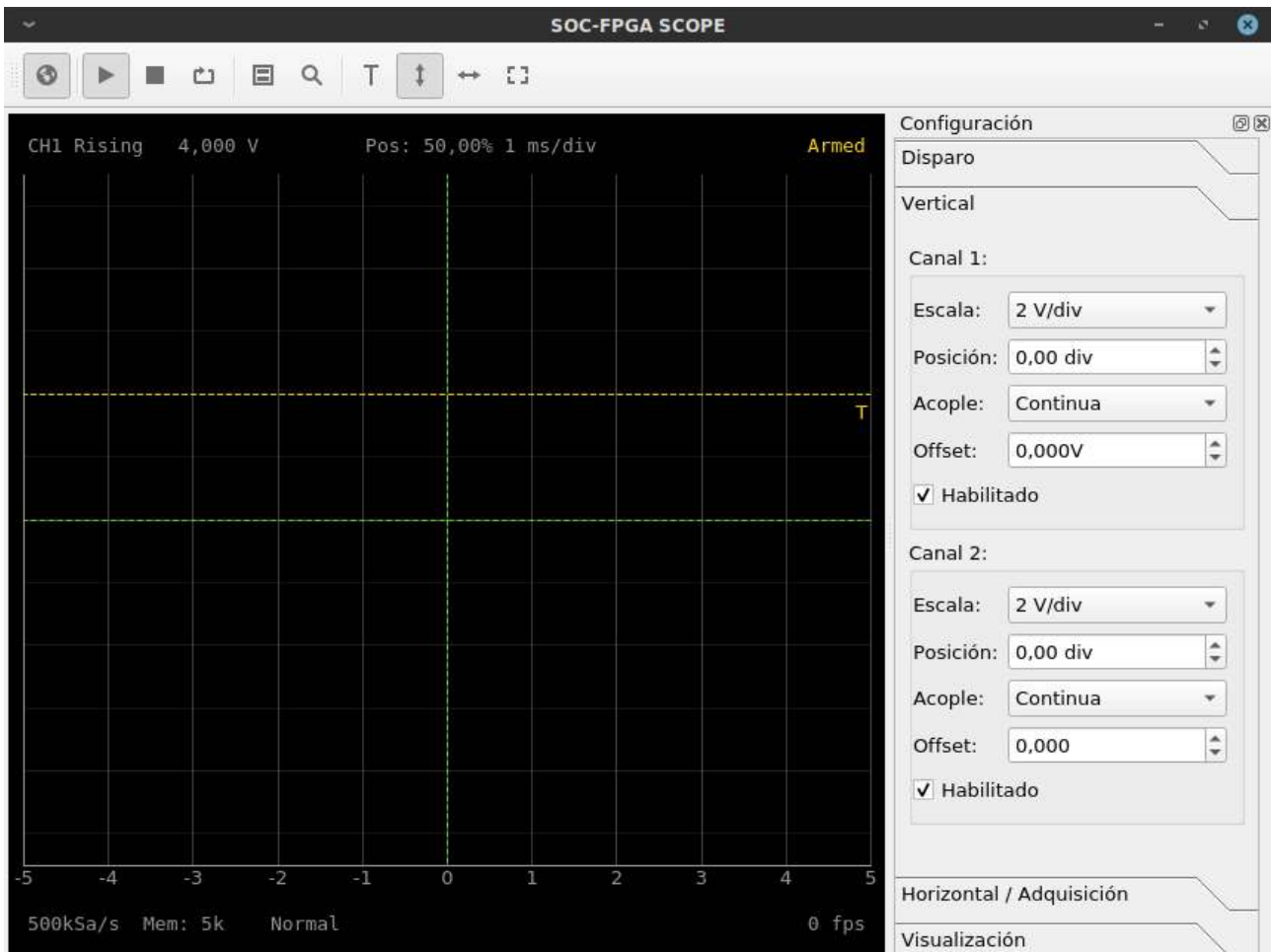


Figura 5.12: Modo de visualización single-plot

En este caso, ambos canales comparten la grilla de visualización, el indicador de posición horizontal y la escala numérica horizontal. Dado que cada canal puede utilizar un ajuste de posición y escala vertical diferente, resulta imposible contar con una escala numérica vertical común a ambos. Por este motivo, el elemento gráfico mencionado no forma parte del modo de visualización single-plot. Como se puede observar en la figura 5.12, el indicador de nivel de disparo se representa con el color asociado al canal que actúa como fuente de disparo. Esta diferenciación de color aplica también para los marcadores de posición vertical de cada canal.

Adicionalmente, se implementa una funcionalidad denominada modo zoom, que permite observar un detalle temporal de las formas de onda asociadas a ambos canales. Es decir, se agrega un gráfico secundario en el área de visualización, que permite analizar en detalle una fracción del periodo de tiempo asociado a una captura completa. La funcionalidad está disponible únicamente en el modo single-plot y en caso de ser activada, genera la siguiente disposición de elementos gráficos:

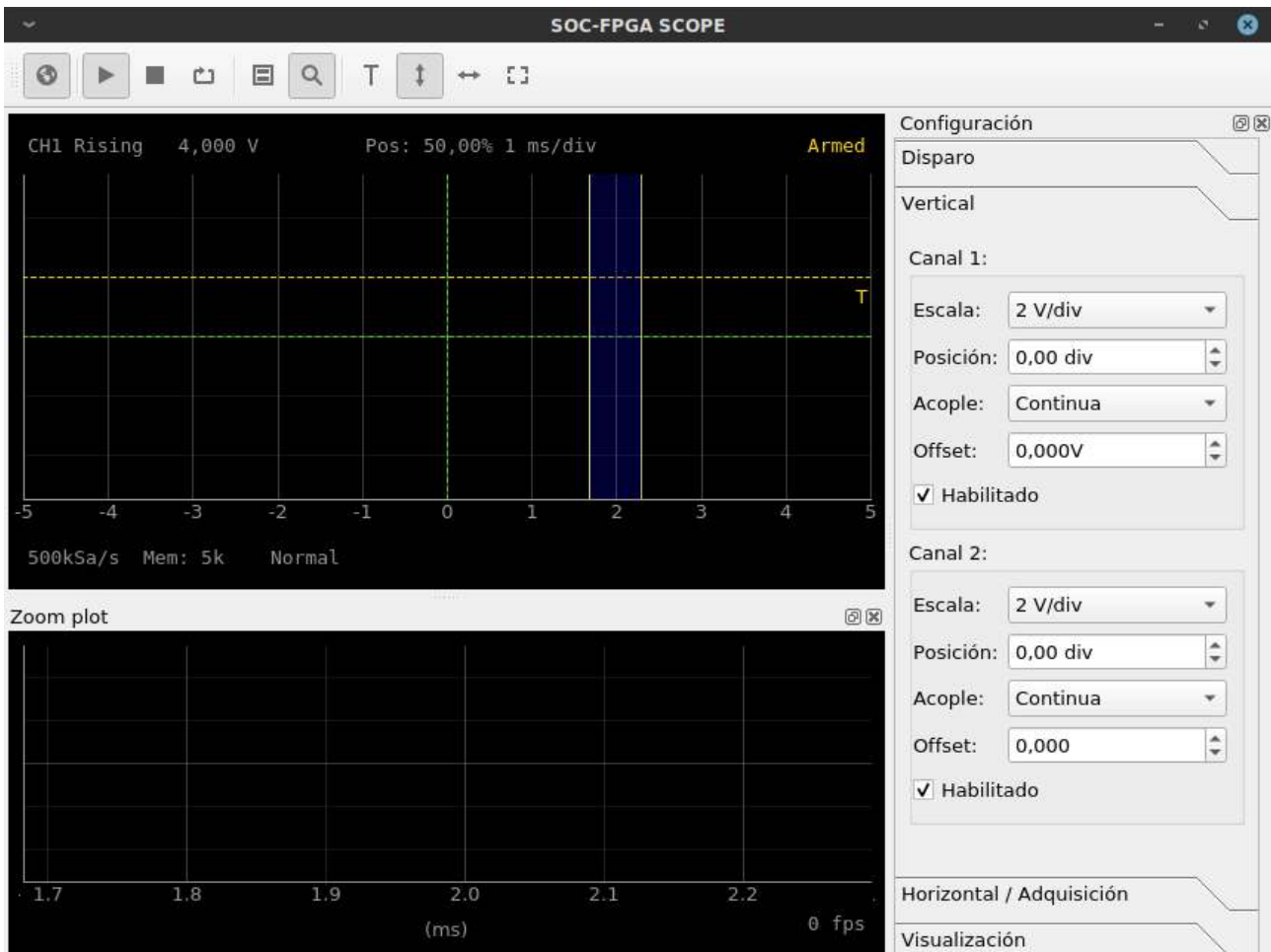


Figura 5.13: Funcionalidad de zoom (detalle temporal)

En el ejemplo de la figura 5.13, se utiliza la funcionalidad para observar un periodo de aproximadamente 1 ms, bajo una captura completa de 10 ms (1 ms/div). Los límites temporales asociados al gráfico secundario, son representados en el área de visualización principal mediante una región de color azul. Esta región puede ser redimensionada y desplazada, permitiendo analizar diferentes sectores de la captura completa.

5.5.2 Barra de herramientas

La barra de herramientas, ubicada en la parte superior de la interfaz, agrupa las acciones más relevantes asociadas al control y utilización del sistema de adquisición. Los diez botones de la barra se encuentran agrupados, mediante el uso de separadores, en cuatro categorías diferentes. Esta situación puede ser observada a continuación:



Figura 5.14: Íconos de la barra de herramientas

En base a la identificación de los botones y categorías realizada en la figura 5.14, se realiza una breve descripción de cada grupo de acciones:

1. Conexión: Consiste en un único ícono asociado a la conexión al servidor. La pulsación del botón despliega un diálogo que permite introducir la dirección IP asociada al servidor de adquisición. Una vez establecida la comunicación de manera exitosa, esta situación es reflejada visualmente mediante la actualización del estado de pulsación del ícono.
2. Control de flujo: Las tres acciones que componen esta categoría (2a, 2b y 2c), se asocian al control de flujo del proceso de adquisición. Los dos primeros botones, permiten iniciar (2a) o detener (2b) la captura continua de datos en tiempo real. La lógica del cliente actualiza el estado de pulsación de ambos íconos para reflejar la situación actual. Por otro lado, el tercer elemento del grupo (2c) permite realizar una única captura de datos, luego de lo cual el sistema pasa al estado “detenido”.
3. Visualización: El primer botón de este grupo (3a), permite conmutar entre el modo de visualización multi-plot y single-plot. Por otro lado, la segunda acción de esta categoría (3b) se asocia a la activación del modo zoom. Esta opción está únicamente disponible para el caso que se utilice el modo de visualización single-plot.
4. Selección de ajustes: La cuarta categoría de acciones está compuesta por cuatro íconos que permiten seleccionar el grupo de ajustes mostrados en el panel lateral de la interfaz. El estado de pulsación de los botones refleja, en un determinado momento, la pestaña que se encuentra activa en el panel. Las opciones posibles son disparo (4a), vertical (4b), horizontal (4c) y visualización (4d).

5.5.3 Panel de ajustes

El área de ajustes agrupa en cuatro categorías diferentes los diversos parámetros de configuración asociados a la adquisición y visualización de los datos. En el desarrollo de capítulos anteriores, se analizó la implementación técnica del sistema de adquisición y el efecto de cada uno de los ajustes por sobre la lógica programable y el front-end analógico. Teniendo en cuenta lo mencionado, en este apartado únicamente se analiza cada ajuste desde el punto de vista del cliente y su comportamiento.

5.5.3.1 Ajustes de disparo

La primera pestaña del panel agrupa los parámetros de configuración asociados al disparo. La disposición de los elementos gráficos y los valores por defecto de cada ajuste pueden ser observados a continuación:

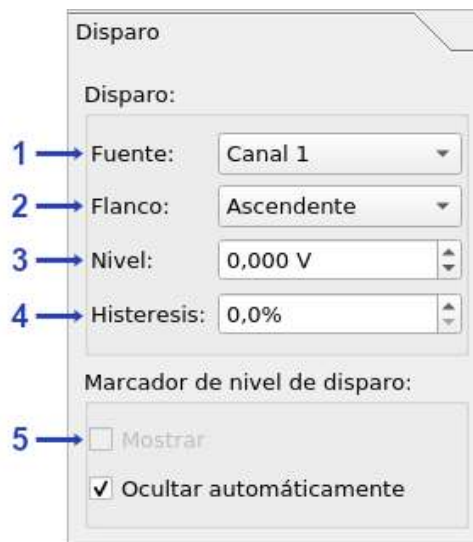


Figura 5.15: Panel de ajustes de disparo

En base a la identificación numérica de la figura 5.15, se realiza una breve descripción de cada ajuste de disparo:

1. Fuente: Menú desplegable que permite seleccionar cuál de los dos canales es analizado en búsqueda de un evento de disparo. Cabe destacar que, debido a la arquitectura del sistema, es posible utilizar como fuente de disparo un canal que no se encuentre habilitado.
2. Flanco: Menú desplegable que permite seleccionar el criterio utilizado para analizar los eventos de disparo. Los cuatro modos posibles son flanco ascendente, flanco descendente, ambos flancos o ninguno.
3. Nivel: Campo numérico que permite seleccionar el umbral de voltaje utilizado para la detección del flanco de disparo. El tipo de selector utilizado permite el desplazamiento del valor, en el rango ± 50 V, mediante las flechas del teclado o la rueda del mouse. El incremento de voltaje, en cada desplazamiento, es calculado por el software en base a la escala vertical del canal que actúa como fuente de disparo. De esta manera, se mantiene la sensibilidad del campo numérico en el rango de interés. Cabe destacar que también existe la posibilidad de establecer, mediante el teclado, un valor numérico arbitrario con una resolución de 1 mV.
4. Histéresis: Campo numérico que determina la separación entre los dos umbrales de voltaje utilizados en la detección del evento de disparo. El ajuste porcentual es realizado, a pasos de 0,2%, entre el valor mínimo de 0% y máximo de 50% (0% por defecto). El porcentaje es aplicado sobre el rango de visualización completo del canal que actúa como fuente de disparo. A modo de ejemplo, un valor del 20% representa una separación de dos divisiones verticales entre los dos niveles de tensión.
5. Visualización de nivel: Dos casillas que controlan la visibilidad del marcador de nivel de disparo en el gráfico principal. Las tres opciones posibles son, mostrado

permanentemente, ocultado permanentemente y automático. En el último caso, el indicador es mostrado en el área de visualización, por un periodo de 5 segundos, cuando se modifica algún ajuste vertical o de disparo.

5.5.3.2 Ajustes verticales

La segunda pestaña del panel agrupa los ajustes verticales para ambos canales del sistema de adquisición. La disposición de los elementos gráficos y los valores por defecto de cada campo se observan a continuación:

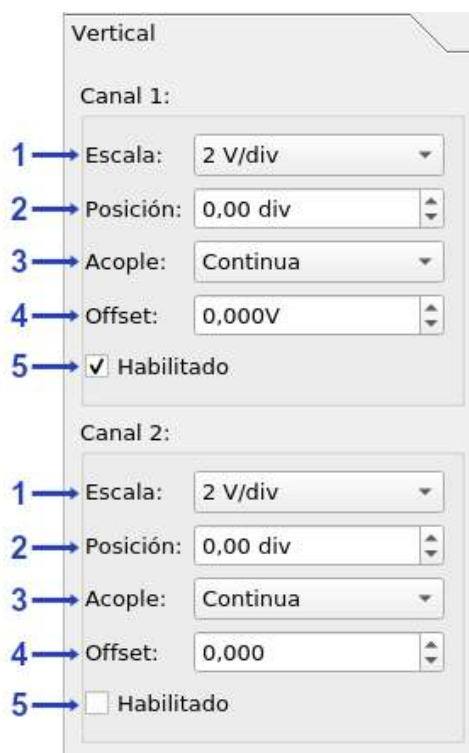


Figura 5.16: Panel de ajustes verticales

Dado que la implementación técnica de cada canal es idéntica, ambos canales comparten los mismos parámetros de configuración. Las funcionalidades asociadas a cada uno de los cinco ajustes verticales, en la figura 5.16, se analiza a continuación:

1. Escala: Menú desplegable que permite seleccionar entre las nueve escalas verticales implementadas. El efecto de un cambio en este parámetro es diferente dependiendo del modo de visualización. En el caso que se utilice un gráfico individual por cada canal (multi-plot), se realiza un re-escalado de los datos y se actualiza la escala numérica vertical para reflejar el cambio. En el modo single-plot, dado que no existe una escala numérica vertical, sólo se realiza el re-escalado de los datos. En ninguno de los dos casos se modifica o desplaza la grilla de visualización utilizada.
2. Posición: Campo numérico que permite ajustar la posición vertical de la señal en el rango ± 5 divisiones, en incrementos de 0,2 divisiones. Adicionalmente, se brinda la posibilidad de

introducir un valor arbitrario con una resolución de 0,01 divisiones. El ajuste posición vertical representa una funcionalidad implementada del lado del cliente, cuyo efecto depende del modo de visualización. En el modo multi-plot, el ajuste desplaza la señal, marcador de posición vertical, grilla y escala numérica del canal correspondiente. Por otro lado, en el modo single-plot, únicamente se desplaza la señal y marcador de posición vertical del canal asociado al cambio (en el gráfico común a ambos canales).

3. Acople: Menú desplegable que permite elegir entre las dos opciones de acoplamiento posibles, continua y alterna. Este ajuste no genera modificaciones en el estado del cliente de visualización, ya que se trata de una funcionalidad asociada al front-end analógico.
4. Offset: Campo numérico de ajuste del nivel de tensión sumado, de manera analógica, a la señal de entrada. El selector cuenta con una resolución de 1 mV y los pasos del ajuste son determinados de acuerdo a la escala vertical seleccionada, manteniendo la sensibilidad del campo numérico. El rango máximo del ajuste se limita por software a ± 15 V para las cinco escalas verticales mayores y $\pm 1,75$ V para las cuatro escalas menores. Los valores máximos y mínimos de offset se asocian al diseño e implementación técnica de la etapa analógica.
5. Habilitación: Casilla que controla el estado de habilitación de cada canal individual. Independientemente de este ajuste, la adquisición inicial de los datos en el buffer circular es realizada para ambos canales. Por otro lado, el procesamiento y envío de datos solo ocurre para los canales habilitados. En el modo multi-plot, un cambio en este ajuste se asocia al agregado, o eliminado, de una nueva área de visualización y todos sus elementos gráficos.

5.5.3.3 Ajustes horizontales

Los ajustes asociados a la base de tiempo y la configuración de adquisición, se encuentran agrupados en la tercera pestaña del panel. La disposición de los elementos gráficos y los valores por defecto se observan en la siguiente imagen:

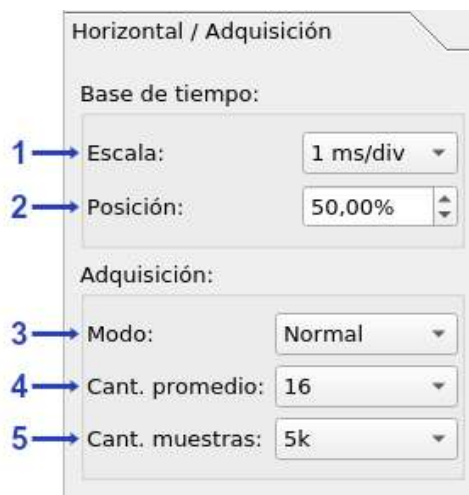


Figura 5.17: Panel de ajustes horizontales y de adquisición

En base a la numeración de los elementos gráficos del panel, en la figura 5.17, se realiza un breve análisis de cada ajuste y su efecto por sobre el comportamiento del cliente:

1. Escala: Menú desplegable que permite seleccionar entre las 16 escalas horizontales implementadas en la arquitectura de adquisición, de 100 $\mu\text{s}/\text{div}$ a 1 s/div . En el área de visualización, un cambio en este ajuste modifica la escala numérica horizontal, sin afectar la grilla y los marcadores de posición. Por otro lado, en base a la selección de base de tiempo, el software de cliente determina los valores de profundidades de memoria disponibles y sus correspondientes factores de decimación asociados.
2. Posición horizontal: Campo numérico que permite realizar el ajuste porcentual de posición horizontal, a pasos de 5%, en el rango de 0% a 100%. Adicionalmente, se provee la posibilidad de introducir manualmente un valor arbitrario con una resolución de 0,01%. Para ambos modos de visualización, un cambio en este parámetro se asocia a un desplazamiento horizontal de la grilla, marcador de posición horizontal y escala numérica horizontal.
3. Modo: Menú desplegable que permite seleccionar entre los cuatro modos posibles de adquisición, normal, alta resolución, detección de pico y promediado. Los primeros tres casos se corresponden con los algoritmos de decimación implementados por hardware, cuyo objetivo es reducir la cantidad de muestras a ser graficadas. Por otro lado, la opción de promediado se asocia al software de cliente. Esta funcionalidad actúa por sobre varios ciclos de adquisición completos, realizando un promediado coherente a nivel temporal de las últimas N capturas, siendo N una cantidad ajustable. Bajo esta modalidad, el ajuste de profundidad de memoria es brindado mediante el algoritmo de decimación normal.
4. Cant. promedio: Menú desplegable mediante el cual se selecciona la cantidad de ciclos de captura a ser promediados. Las opciones implementadas son 2, 4, 8, 16, 32, 64 y 128 formas de onda.
5. Cant. muestras: Menú desplegable asociado a la selección de la profundidad de memoria, es decir, la cantidad de muestras enviadas por el servidor y graficadas por el cliente. Los valores posibles para este ajuste se encuentran preestablecidos por diseño y dependen de la base de tiempo seleccionada. Los valores utilizados por defecto para cada base de tiempo tienen como objetivo evitar el consumo excesivo de recursos en el cliente y reducir el ancho de banda asociado al envío de datos.

5.5.3.4 Ajustes de visualización

El último sector del panel de configuración es utilizado para implementar una serie de ajustes asociados al área de visualización. Los mismos pueden ser observados en la figura 5.18, presentada a continuación:

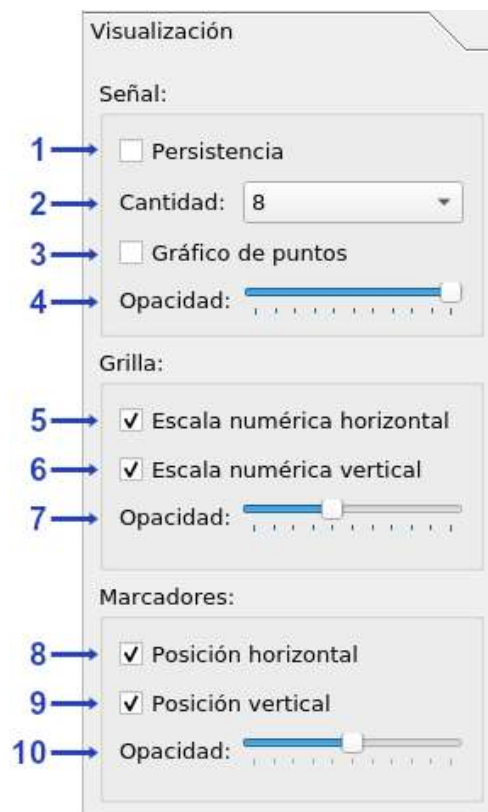


Figura 5.18: Panel de ajustes de visualización

Como se puede observar, se brinda un alto grado de flexibilidad en la manera en que se visualizan los datos de ambos canales. Los 10 ajustes que componen esta pestaña, de acuerdo a la figura 5.18, son analizados a continuación:

1. **Persistencia:** Casilla que permite activar o desactivar la persistencia en la visualización de las señales de ambos canales. Se trata de una funcionalidad implementada en el software cliente que permite mantener en pantalla el gráfico de las últimas N capturas realizadas, siendo N un valor ajustable. Bajo esta modalidad, la opacidad del trazo de la señal es controlada por el software cliente, de acuerdo a la cantidad de persistencia seleccionada.
2. **Cantidad de persistencia:** Menú desplegable asociado a la selección de cantidad de capturas a graficar bajo la modalidad de persistencia. Las opciones posibles son 2, 4, 8, 16, 32, 64 y 128 formas de onda.
3. **Gráfico de puntos:** Casilla que permite seleccionar la manera en que se representan las señales en el área de visualización. La primera opción utiliza una línea sólida, asociada a una interpolación lineal de los datos. La segunda opción, por otro lado, representa cada muestra de la señal mediante un punto en el gráfico.
4. **Opacidad de señal:** Botón deslizante que ajusta la opacidad utilizada en el trazo de las señales, en el rango de 0% a 100%. Este parámetro afecta todos los modos de visualización, excepto en el caso que se active la funcionalidad de persistencia.
5. **Escala numérica horizontal:** Casilla que controla la visibilidad de la escala numérica horizontal. Este ajuste afecta ambos modos de visualización, single-plot y multi-plot.

6. Escala numérica vertical: Casilla que controla la visibilidad de la escala numérica vertical. Este ajuste sólo afecta el modo multi-plot, ya que en el caso de un único gráfico no se utiliza una escala numérica vertical.
7. Opacidad de grilla: Botón deslizante que ajusta el nivel de visibilidad de la grilla, tanto horizontal como vertical. El parámetro afecta en igual proporción a las grillas utilizadas en los modos single-plot y multi-plot. Adicionalmente, este ajuste aplica también a la grilla del gráfico asociado a la modalidad de zoom.
8. Marcador de posición horizontal: Casilla que permite mostrar u ocultar el marcador de posición horizontal. Este elemento gráfico es común a ambos canales y se encuentra presente en los gráficos multi-plot y single-plot, por lo que el ajuste aplica a ambos modos de visualización.
9. Marcador de posición vertical: Casilla que ajusta la visibilidad de los marcadores de posición vertical. El ajuste afecta por igual a las referencias de ambos canales, en los modos de visualización single-plot y multi-plot.
10. Opacidad de marcadores: Botón deslizante asociado a la opacidad del trazo segmentado utilizado en los indicadores de posición horizontal y vertical. Cabe destacar que este ajuste no afecta la visibilidad del marcador de nivel de disparo, el cual cuenta con ajustes independientes.

Capítulo 6

Validación

Este capítulo tiene como objetivo principal verificar y validar el diseño teórico realizado para el sistema de adquisición. El desarrollo de este capítulo se divide en tres secciones principales: armado de prototipo, mediciones y pruebas integrales. En el apartado 6.1 se describe el proceso de armado de las placas físicas que componen el front-end analógico. Se brindan detalles sobre la fabricación de los PCBs, el pedido de componentes electrónicos y el ensamblado. Finalmente, se analizan algunos errores no críticos detectados en el diseño físico y se presentan posibles soluciones a futuro.

La sección 6.2 se dedica a las mediciones que permiten validar el diseño del front-end. En este caso, se analiza la fuente de alimentación, las etapas analógicas y los elementos involucrados en la comunicación con la FPGA. Las mediciones presentadas tienen como objetivo evaluar diversos parámetros cualitativos y cuantitativos, entre los que destacan: linealidad, ancho de banda, estabilidad, niveles de ruido, precisión y exactitud. Adicionalmente, se analizan las desviaciones con respecto a los valores teóricos esperados, sus causas y posibles soluciones a futuro.

Finalmente, el apartado 6.3 consiste en una serie de pruebas integrales, que analizan el funcionamiento del sistema en su totalidad. Es decir, se utiliza el dispositivo de la manera en que fue diseñado, haciendo uso del software cliente para visualizar y caracterizar señales de entrada de diversa índole. Las pruebas son seleccionadas a fin de exigir el sistema y verificar su funcionamiento en los casos límites de uso.

Cabe destacar que el apartado de pruebas integrales cuenta con dos objetivos diferenciados, siendo el objetivo principal validar el diseño y verificar el correcto funcionamiento. Por otro lado, se busca complementar la información brindada en los capítulos anteriores sobre la funcionalidad del sistema y sus características. Es decir, se emplea un enfoque práctico para la descripción del sistema, presentando casos de uso y ejemplos de mediciones.

6.1 Armado de prototipo

6.1.1 Fabricación de PCBs

Una vez realizado y verificado el layout, se procede a la fabricación de las tres placas: front-end, fuente de alimentación y adaptador del conector. Luego de un análisis de las alternativas disponibles, se decide recurrir a la empresa JLCPCB con base en Shenzhen, China. Los motivos que determinan la elección son el bajo costo y tiempo de fabricación, en combinación con la alta calidad de los PCB de esa compañía. El fabricante cuenta con un servicio de cotización instantáneo en línea y brinda la posibilidad de verificar los archivos de fabricación antes del pago. Adicionalmente, se valora el testeado eléctrico completo de las placas que se lleva a cabo en la fábrica.

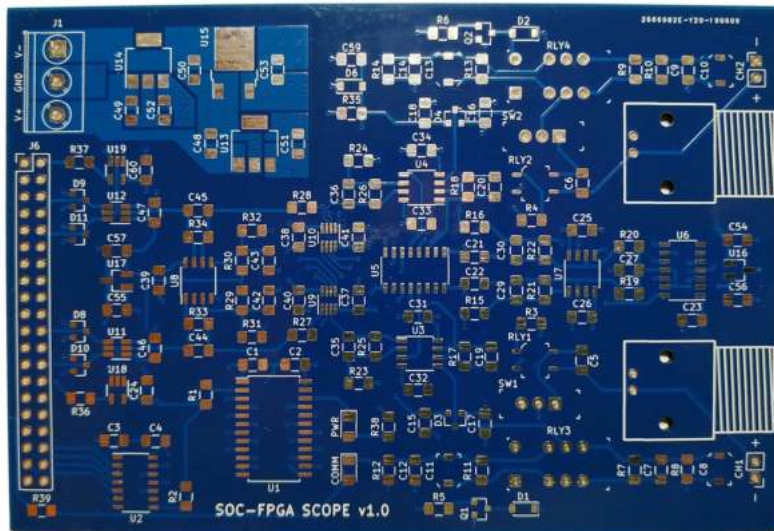


Figura 6.1: Placa principal



Figura 6.2: Placa adaptadora y fuente de alimentación

En las figuras 6.1 y 6.2 se pueden observar las tres placas fabricadas, antes del montaje de los componentes. Los PCB son fabricados en material FR-4 de 1,6 mm con un espesor de cobre de 35 μm en las capas externas y 17 μm en las internas. El costo de fabricación, a la fecha del proyecto, es de US\$ 36,9 para el PCB principal, US\$ 4 para la fuente y US\$ 5 para el adaptador del conector. Estos montos se corresponden con el pedido de cinco unidades de cada PCB, cantidad mínima del fabricante, y no consideran cargos de envío. Del análisis de los datos surge que el monto asociado a las tres placas que componen una unidad del sistema de adquisición es de US\$ 9,18. Por otro lado, el plazo de fabricación, a la fecha del proyecto, es de cinco días hábiles para las tres placas.

6.1.2 Pedido de componentes

Con respecto al pedido de los componentes electrónicos necesarios, el proveedor seleccionado es Mouser Electronics, con base en Texas, Estados Unidos. En este caso, los criterios que llevan a su elección son la amplia disponibilidad de componentes en el catálogo y en inventario, así como también los reducidos costos y tiempos de envío. Otro elemento que se tiene en consideración es la gran cantidad de información disponible en la página del proveedor y los motores de búsquedas paramétricas que brinda.

A continuación, se presenta una lista de los componentes requeridos de acuerdo al diseño del front-end y su fuente de alimentación. Se anexan las cantidades de cada elemento y los costos del proveedor en el momento del pedido. Cabe destacar que la lista es una versión simplificada, que agrupa elementos similares, como por ejemplo, las resistencias de diferentes valores.

Componente	Marca	Modelo	Cant.	Precio (US\$)
Conectores	Varios	Varios	6	4,83
Capacitores cerámicos 1206	Varios	Varios	54	12,17
Resistencias de película 1206	Varios	Varios	39	9,7
Capacitor variable 2-6pF	Knowles Voltronics	JZ060	2	4,48
Capacitor variable 5-30pF	Knowles Voltronics	JZ300	2	3,6
Diodo silicio (protección relé)	Diodes Incorporated	1N4148W-7-F	2	0,32
Diodo Schottky (protección ADC)	RHOM	BAT54SHMFH	4	0,64
Diodo silicio (protección entrada)	RHOM	BAV199HMFH	2	0,44
Diodo Zener 3,3 V	Diodes Incorporated	BZT52C3V3-7-F	1	0,2
Diodo LED SMD 1206 Verde	Lite-On	LTST-C150GKT	2	0,64
Regulador 5 V (0,5 A)	Texas Instruments	UA78M05C	1	0,62
Regulador -5 V (0,5 A)	On Semiconductor	MC79M05C	1	0,49
Regulador 3,3 V (0,5 A)	Texas Instruments	UA78M33C	1	0,58
Referencia 5 V (DAC)	Maxim Integrated	MAX6035BAUR	1	2,08
Referencia 3,3 V (ADC)	Texas Instruments	REF3033AIDBZR	1	1,39
Convertor niveles lógicos 4 bits	Texas Instruments	TXB0104DR	1	1,27
Convertor niveles lógicos 8 bits	Texas Instruments	SN74LVC8T245	1	1,02
Buffer Schmitt trigger	Texas Instruments	SN74LVC1G17	2	0,7
Amp. operacional dual 10 MHz	Texas Instruments	OPA2197	4	9,36
DAC dual 12 bits	Texas Instruments	MCP4922-ESL	1	2,69
Potenciómetro digital dual 8 bits	Maxim Integrated	DS1267BS	1	4,53
PGA 11 MHz	Analog Devices	LTC6910-1	2	5,52
ADC 1 MSa/s 12 Bits	Texas Instruments	ADCS7476	2	7,46
Switch SPDT deslizable	E-Switch	EG1218	2	1,34
Relé mecánico DPDT 5V	Kemet	EC2-5NU	2	4,62
Relé estado solido SPST	IXYS	CPC1017NTR	2	2,22
Mosfet canal N	On Semiconductor	NTR3C21NZT1G	2	0,7
Total			141	83,61

Tabla 6.1: Componentes, cantidad y precio para la placa principal

Componente	Marca	Modelo	Cant.	Precio (US\$)
Conectores	Varios	Varios	3	2,07
Capacitor electrolítico 1000 μ F	Nichicon	UVR1V102MHD	2	1,4
Capacitor electrolítico 2200 μ F	Nichicon	UVR1V222MHD	2	1,94
Capacitor cerámico 10 μ F	TDK	C3216X5R1E106	4	2,36
Resistencia 1206 1k Ω 1%	Panasonic	ERJ-8ENF1001V	2	0,2
Diodo rectificador 1 A (100 V)	Diodes Incorporated	1N4002-T	2	0,38
Diodo LED SMD 1206 Verde	Lite-On	LTST-C150GKT	2	0,64
Total			17	8,99

Tabla 6.2: Componentes, cantidad y precio para la fuente de alimentación

La tabla 6.1 se corresponde con el front-end analógico, el cual consta de 141 componentes con un costo total de US\$ 83,6. En la tabla 6.2, se presentan los elementos utilizados en la fuente de alimentación. En este caso, la cantidad total es de 17 componentes con un costo total de US\$ 9, es decir, US\$ 92,6 para ambas placas.

6.1.3 Armado de placas

En esta sección se describe el proceso de armado de la fuente de alimentación y el PCB principal. Los componentes de tipo SMD son montados con una estación de soldado de aire caliente con control de temperatura y flujo. Para el caso de los componentes through hole, se recurre un soldador de tipo lápiz. La verificación de las conexiones se realiza en base a mediciones de continuidad con multímetro y en base al uso de un microscopio electrónico.



Figura 6.3: Fuente de alimentación finalizada



Figura 6.4: PCB principal finalizado

En las figuras 6.3 y 6.4 se pueden observar ambas placas con los componentes montados. Cabe destacar que el orden en el que fueron soldados los componentes permitió la verificación cualitativa del funcionamiento de cada etapa involucrada. Durante esta fase se descubrieron algunos problemas no críticos que son analizados en la sección siguiente.

6.1.4 Errores de diseño físico de PCB

A continuación, se presentan dos errores detectados durante el montaje de la placa principal. Ninguno de los problemas reviste gravedad y su solución se presenta como mejoras a futuro para una posible segunda revisión del PCB.

El error más crítico se asocia al footprint utilizado para el potenciómetro digital en el diseño del PCB. La placa cuenta, para este integrado, con un footprint de tipo SOIC-16 Narrow, de acuerdo a la descripción en el catálogo del proveedor de componentes. Sin embargo, el IC usa un encapsulado SOIC-16 Wide, cuyo ancho es mayor. Teniendo en cuenta que el componente no existe en el tamaño presente en el diseño, la solución adoptada fue “doblar” los pines del integrado con el objetivo de poder soldarlo en la placa. Una posible solución a futuro sería rediseñar el PCB, utilizando el encapsulado adecuado para el componente.



Figura 6.5: Conector BNC, vista superior e inferior

El segundo error detectado se asocia al footprint utilizado en los conectores BNC. En la imagen 6.5 se presenta el formato físico del conector, el cual cuenta con dos pines de conexión y dos pines de montaje. El footprint utilizado en el diseño es incorrecto ya que, a pesar de contar con todos los pines, no expone una zona de cobre en la capa inferior que permita soldar los pines de montaje. En este caso, a fin de aumentar la rigidez estructural de los conectores, se utilizó un adhesivo durante la fase de montaje. Nuevamente, se trata de un problema de gravedad leve, que puede ser corregido en una nueva versión del PCB.

6.2 Mediciones

Una vez ensambladas ambas placas, se procede a iniciar la etapa de pruebas y mediciones, cuyo objetivo principal es validar el diseño teórico. Los resultados se presentan en diversas subsecciones, recurriendo al uso de tablas numéricas y capturas de pantalla de las formas de onda en un osciloscopio.

En la sección 6.2.2 se evalúa el comportamiento de la fuente de alimentación, presentando mediciones realizadas con y sin carga. En los apartados 6.2.3 y 6.2.4, se realiza el análisis de precisión del front-end, evaluando el error de offset y ganancia antes y después del proceso de calibración. Por otro lado, se dedica la sección 6.2.5 a la caracterización de la respuesta dinámica de la etapa de generación de offset. En este apartado se evalúa la respuesta del DAC y el amplificador ante un cambio en el ajuste de voltaje.

Luego, en las secciones 6.2.6 a 6.2.12 se analiza la respuesta temporal de las diferentes etapas involucradas en el acondicionamiento de la señal. En estos apartados se evalúa el acople de alterna, atenuadores de entrada, protección, potenciómetro digital y PGA, entre otras. Durante estas pruebas se emplea como señal de entrada un pulso digital de amplitud 3,3 V. Finalmente, en los apartados 6.2.13 y 6.2.14 se realizan mediciones de ancho de banda total y ruido pico a pico para cada escala en ambos canales.

6.2.1 Instrumental utilizado

Las mediciones se realizan con un osciloscopio digital de dos canales y 200 MHz de ancho de banda, modelo XDS3202E de Owon [95]. Adicionalmente, en los casos que resulta necesario, se recurre a un multímetro digital de 3 ½ dígitos UT151E de marca UNI-T.



Figura 6.6: Osciloscopio Owon XDS3202E utilizado en las mediciones

Para la alimentación del PCB principal se hace uso de una fuente lineal de laboratorio de tres canales UTP3305 de UNI-T. Esta última, a diferencia de la fuente diseñada, permite establecer un límite de corriente por canal. Esta funcionalidad minimiza las consecuencias de un posible cortocircuito causado durante la medición, reduciendo los potenciales daños ocasionados al circuito.



Figura 6.7: Fuente de laboratorio UNI-T UTP3305 utilizada para alimentar el front-end

Las señales de entrada, de tipo escalón, son generadas en base a una placa de desarrollo STM32F446 [96] de STMicroelectronics, basada en un microcontrolador ARM Cortex-M4. Este tipo de dispositivo es capaz de generar pulsos digitales cuyo tiempo de crecimiento, menor a 2 ns, excede el ancho de banda del front-end y osciloscopio utilizado [97], por ende, resulta apropiado para esta clase de mediciones. Adicionalmente, para las estimaciones de ancho de banda de las escalas, se utiliza un generador de funciones HP 33120A [98] de 15 MHz. En este caso, se emplea un instrumento especializado debido a la necesidad de generar señales senoidales de amplitud y frecuencia variable.

6.2.2 Medición de fuente lineal

En esta sección se presentan las mediciones realizadas por sobre el transformador y el PCB de rectificado y filtrado. En la figura 6.8 se puede observar la forma de onda a la salida del transformador bajo una situación de circuito abierto, es decir, sin carga.

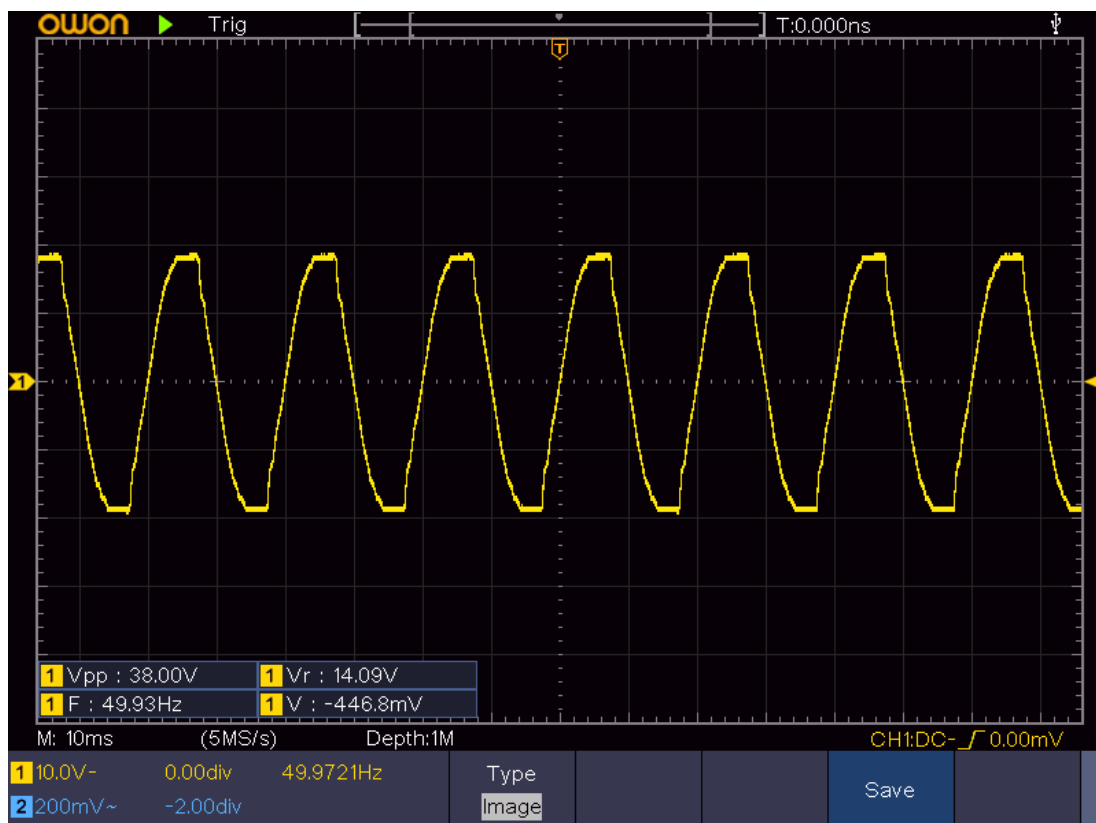
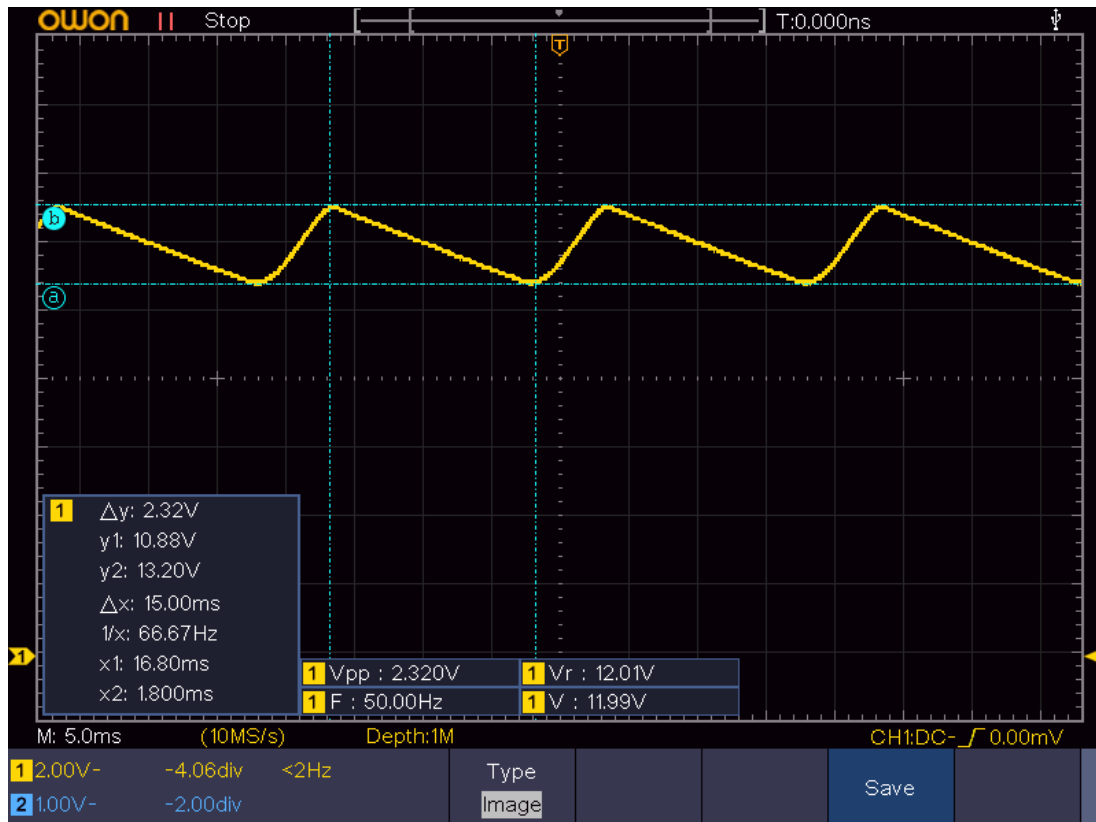


Figura 6.8: Forma de onda a la salida del transformador sin carga

La tensión de salida, de acuerdo al osciloscopio en la figura 6.8, es de 14,1 V_{RMS} . (19 V_P). Teniendo en cuenta la especificación del transformador de 12 V_{RMS} (17 V_P) a 1 A, el resultado es el esperado. Bajo esta situación, sin carga, se miden tensiones de salida, rectificadas y filtradas, de 19 V en ambas ramas, sin observarse variaciones apreciables.

Con el objetivo de verificar las especificaciones de ripple y tensión mínima, se evalúa la fuente bajo una corriente de carga I_L de 0,5 A, valor de diseño. Las mediciones son realizadas en la rama positiva de la fuente, siendo los valores válidos para la rama negativa, debido a ser idéntica. Para generar la corriente de carga, se emplea un regulador de 5 V UA7805 [99] de Texas Instruments, con una resistencia de 10 Ω conectada a su salida.



Bajo esta situación, se obtiene un ripple de 2,32 V_{PP} con una tensión mínima de 10,88 V y máxima de 13,2 V, siendo el valor medio de 12 V. Los valores teóricos, calculados en la sección 3.9, son de 3,5 V_{PP} para el ripple y 9 V para la tensión mínima a la salida. Los resultados obtenidos son levemente mejores a los calculados en el diseño teórico, en el cual se plantea el peor caso posible de acuerdo a las tolerancias de los componentes.

6.2.3 Calibración de offset

Antes de caracterizar todas las etapas involucradas en el acondicionamiento de la señal, se procede a calibrar el offset y la ganancia de las escalas de ambos canales. Con el objetivo de analizar la precisión inicial del sistema y de verificar el correcto funcionamiento de los algoritmos diseñados, se realizan varias mediciones antes y después de la calibración.

Cabe recordar que, de acuerdo al diseño teórico, el valor inicial programado en el DAC es de 2048 o 2,5 V a la salida del convertor. Bajo estas condiciones, para el caso de entrada nula, es esperable una tensión de 0 V en todas las etapas de acondicionamiento internas. En la última etapa, buffer del ADC, se desplaza la señal sumando un voltaje de 1,65 V, hecho que se corresponde con una lectura del ADC de 2048.

Como fue analizado en el apartado 5.3, el algoritmo de calibración de offset actúa por sobre la constante del DAC, cuyo valor inicial es 2048, buscando de manera iterativa el valor que genera el menor error para cada escala, es decir, la tensión más cercana a 1,65 V en la entrada del ADC.

Adicionalmente, de existir una diferencia remanente entre la lectura del ADC y el valor teórico de 2048, se corrige mediante operaciones aritméticas en la FPGA.

A continuación, se presentan las mediciones de offset, antes y después de la calibración. Los valores se miden a la salida del amplificador de ganancia programable, etapa previa al desplazamiento de 1,65 V. La razón de esta elección, radica en el hecho de que se tiene una mayor precisión analizando un voltaje cercano a 0 V, con respecto a medir una diferencia en 1,65 V. Pese a que la prueba no es la ideal, brinda un indicativo de los desajustes presentes en todas las etapas anteriores.

Escala (V/div)	G_{PGA} (V/V)	Offset (V_6) CH1 (mV)	Offset (V_6) CH2 (mV)	Offset (V_6) CH1 CAL (mV)	Offset (V_6) CH2 CAL (mV)
5	1	-8,4	-2,2	-5,1	-6
2	1	-8,5	-2,5	-4,9	-6,2
1	1	-8,9	-4,2	-5,3	-6,6
0,5	2	-13,9	-8,1	-4,3	-5,4
0,2	5	-28,8	-19,6	-4,4	-7,3
0,1	1	-9	-4,5	-5	-6
0,05	2	-14	-8,7	-6	-6
0,02	5	-29,1	-21,5	-2,1	-8,1
0,01	10	-54,3	-42,5	0	-3

Tabla 6.3: Mediciones de offset antes y después de la calibración para cada escala

En la tabla 6.3 se puede observar que el offset antes de la calibración varía significativamente con la selección de escala y depende mayoritariamente de la ganancia del PGA. Esto se debe a que dicho elemento amplifica los desajustes presentes en las etapas anteriores. Para comprender los resultados obtenidos luego de la calibración, se debe tener en cuenta que el voltaje a la salida del PGA que genera 1,65 V a la entrada del ADC es de -4,95 mV. Esta tensión surge de realizar los cálculos con los valores de componentes reales de la última etapa, sin realizar aproximaciones. Los valores medidos oscilan alrededor de la tensión necesaria. En las últimas dos escalas existe un error remanente mayor, asociado al aumento de la ganancia del front-end, lo que reduce la resolución del DAC y por ende su capacidad de calibrar el offset.

Escala (V/div)	N_{CAL} DAC CH1	N_{CAL} ADC CH1	N_{CAL} DAC CH2	N_{CAL} ADC CH2
5	7	0	-8	0
2	6	0	-6	0
1	3	0	-2	1
0,5	4	-1	1	-1
0,2	4	-1	2	1
0,1	3	0	-1	0
0,05	3	1	1	0
0,02	4	-3	2	2
0,01	4	-6	3	-4

Tabla 6.4: Parámetros de calibración del DAC y ADC para cada escala

A fin de completar el análisis, se listan en la tabla 6.4 los parámetros de calibración de offset que surgen de la ejecución del algoritmo. N_{CAL} DAC representa la constante de calibración del DAC, es

decir, el valor que debe ser sumado a 2048 en la programación del convertor. El segundo parámetro, $N_{CAL ADC}$, se refiere al error remanente luego de ajustar el voltaje del DAC, es decir, la diferencia entre el valor numérico medido en el ADC con respecto al esperado de 2048. Nuevamente, se puede observar que el error remanente se incrementa al aumentar la ganancia del front-end, siendo el peor caso de -6 LSB para el primer canal en la escala de 10 mV.

6.2.4 Calibración de ganancia

Luego de ajustar el offset de las escalas, se procede a ejecutar la calibración de ganancias. Cabe recordar que este algoritmo opera en cada escala, generando en base al DAC una excursión de voltaje en la entrada del ADC. La excursión obtenida, para cada escala, en la entrada del ADC es comparada con los valores teóricos esperados, de acuerdo a la ganancia de diseño requerida. En base a ambos resultados, se calcula un valor de corrección que es aplicado en las operaciones aritméticas de la FPGA.

Escala (V/div)	$N_{PP DAC}$	$N_{PP ADC}$	$N_{PP ADC}$ (med) CH1	$N_{PP ADC}$ (med) CH2	Error % CH1	Error % CH2
5	3110	1861,59	1873,77	1873,75	0,65 %	0,65 %
2	3110	2326,99	2339,10	2341,66	0,52 %	0,63 %
1	2488	3723,18	3743,34	3750,29	0,54 %	0,73 %
0,5	1244	3723,18	3743,80	3751,57	0,55 %	0,76 %
0,2	498	3726,17	3734,38	3731,43	0,22 %	0,14 %
0,1	2206	3721,73	3690,36	3697,90	-0,84 %	-0,64 %
0,05	1104	3725,11	3695,06	3699,77	-0,81 %	-0,68 %
0,02	440	3711,61	3668,91	3658,21	-1,15 %	-1,44 %
0,01	220	3711,61	3663,40	3639,15	-1,30 %	-1,95 %

Tabla 6.5: Errores de ganancia obtenidos en la calibración para cada escala

En la tabla 6.5 se presentan los resultados obtenidos, en la primera columna se presentan los rangos de excursión alrededor del valor central para el DAC, mientras que en la segunda columna se listan las variaciones ideales esperadas en el ADC. Los rangos reales obtenidos al realizar la prueba en ambos canales se pueden observar en las siguientes dos columnas. Cabe recordar que el factor de corrección es calculado de acuerdo a la ecuación 6.1.

$$G_{CAL} = \frac{N_{ADC}(teórico)}{N_{ADC}(medido)} \tag{6.1}$$

El producto entre este factor y la señal de entrada corrige el error de ganancia. Dado que las constantes de calibración son muy cercanas al valor unitario, en la tabla se presentan los errores relativos porcentuales de ganancia. Estos valores se obtienen en base a la siguiente ecuación:

$$E_{\%} = (100) \cdot \frac{N_{ADC}(medido) - N_{ADC}(teórico)}{N_{ADC}(teórico)} \tag{6.2}$$

Se puede concluir que los errores de ganancia son mínimos, siendo el peor caso -2 % para la escala de 10 mV/div del segundo canal. El grado de exactitud obtenido es atribuido, principalmente, al uso de resistencias con tolerancias del 1%.

6.2.5 Etapa de generación de offset

Con el objetivo de verificar el comportamiento dinámico del DAC y la posterior etapa de amplificación, se evalúa la respuesta ante un cambio de voltaje de ambos componentes. Teniendo en cuenta el rango del DAC, de 0 V a 5 V, se programa el convertor para generar un escalón de 0,25 V a 4,75 V (90 % del rango total). A continuación, en la figura 6.10, se presenta la forma de onda a la salida del convertor, para el caso del primer canal:

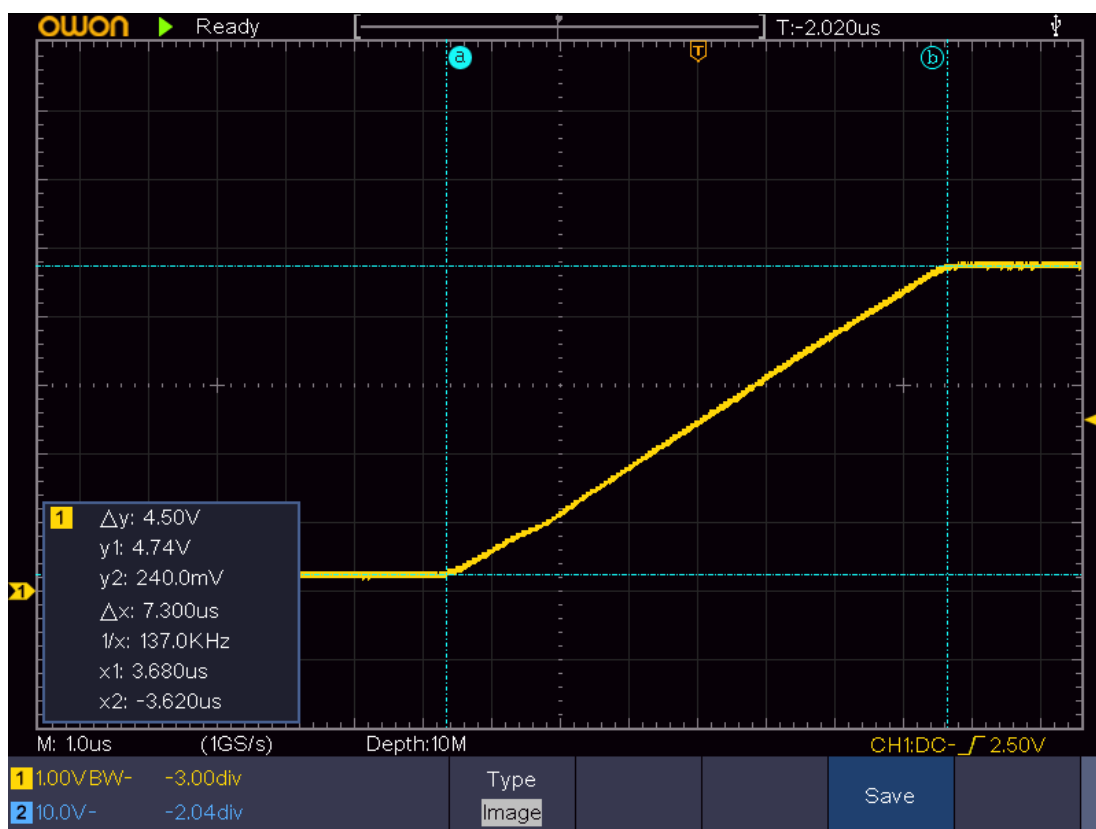


Figura 6.10: Salida del DAC ante un escalón de 4,5 V_{PP} centrado en 2,5 V

El factor limitante en la respuesta es el slew rate del buffer interno de salida del DAC. Teniendo en cuenta que se llega al nuevo voltaje en 7,3 μs, se calcula a continuación el slew rate:

$$SR(DAC) = \frac{\Delta V}{\Delta t} = \frac{4,5 V}{7,3 \mu s} = 0,6 (V/\mu s) \quad (6.3)$$

El valor obtenido es levemente mayor al especificado de la hoja de datos [37] (0,55 V/μs típico) y resulta aceptable para la generación de un offset. La misma prueba es repetida para evaluar la

respuesta del amplificador presente a continuación del DAC. En este caso, se programa el convertor para generar un escalón de $2,25 V_{PP}$ alrededor de $2,5 V$, es decir, $4,5 V_{PP}$ a la salida del amplificador. A continuación, en la figura 6.11, se observa el resultado obtenido, para el primer canal de adquisición:

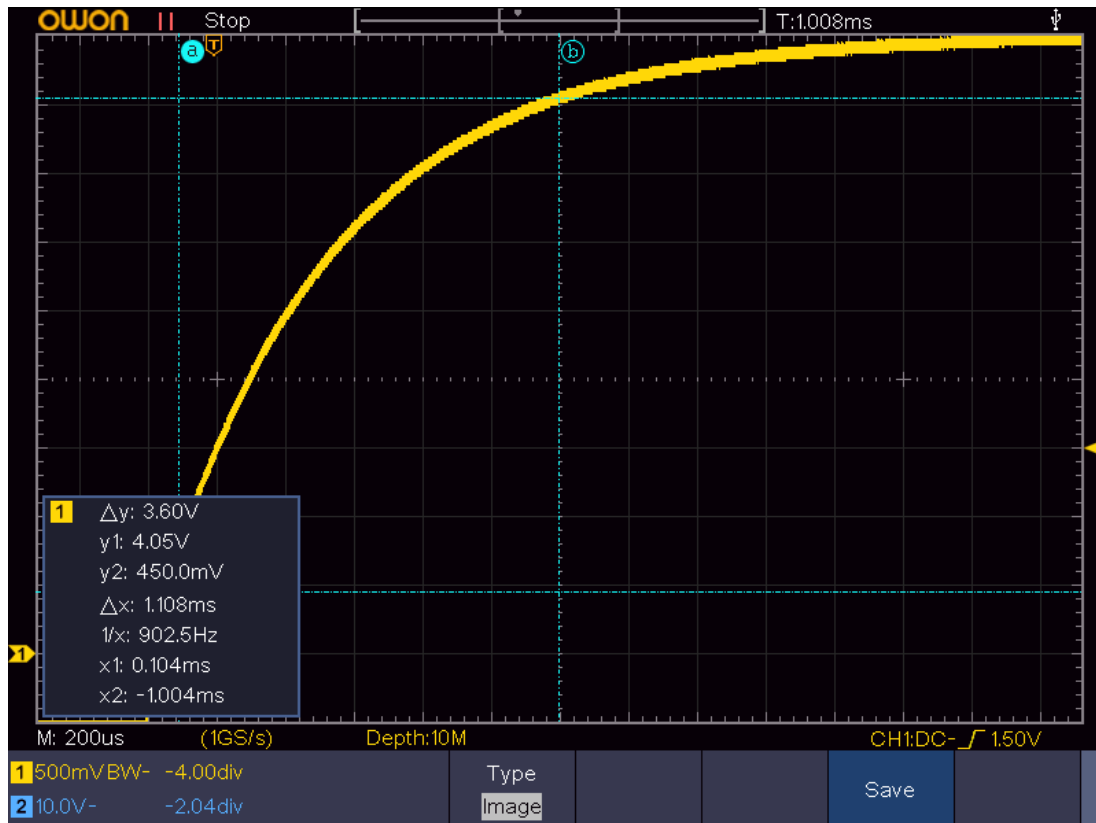


Figura 6.11: Salida del amplificador de offset ante un escalón de $2,25 V_{PP}$ centrado en $2,5 V$

Es pertinente aclarar que, debido al reducido ancho de banda del amplificador, su tiempo de respuesta es órdenes de magnitud mayor al del DAC. Esto significa que la salida del DAC, limitada por el slew rate, puede ser considerada un escalón ideal a efectos de analizar el comportamiento del amplificador. En base a los cursores de la figura 6.11, el tiempo de crecimiento, entre el 10% y el 90%, es de $1,1 ms$. Este parámetro es utilizado a continuación para calcular la frecuencia de corte:

$$B(3 dB) = \frac{0,35}{t_r} = \frac{0,35}{1,1 ms} = 318,2 Hz \quad (6.4)$$

La fórmula 6.4 relaciona el ancho de banda y el tiempo de crecimiento en un filtro pasa bajos de primer orden. Su aplicación en sistemas con otro tipo de respuesta en frecuencia representa una aproximación. El valor obtenido de $318 Hz$ es idéntico al estimado en la etapa de diseño, en la sección 3.4.10. En el caso del segundo canal, el ancho de banda asociado al amplificador del DAC es de $323 Hz$, siendo las formas de onda similares.

6.2.6 Entrada

En las siguientes subsecciones se analiza la respuesta al escalón de las diferentes etapas involucradas en el acondicionamiento de la señal a ser adquirida. Se utiliza como entrada una señal cuadrada de 1 kHz y una amplitud pico a pico de 3,3 V, centrada en 1,65 V. La conexión entre la placa de desarrollo encargada de generar la señal y el front-end se realiza con una punta de prueba de osciloscopio en el ajuste X1. Es decir, se ingresa la forma de onda a través de los conectores BNC presentes en el PCB de acondicionamiento.

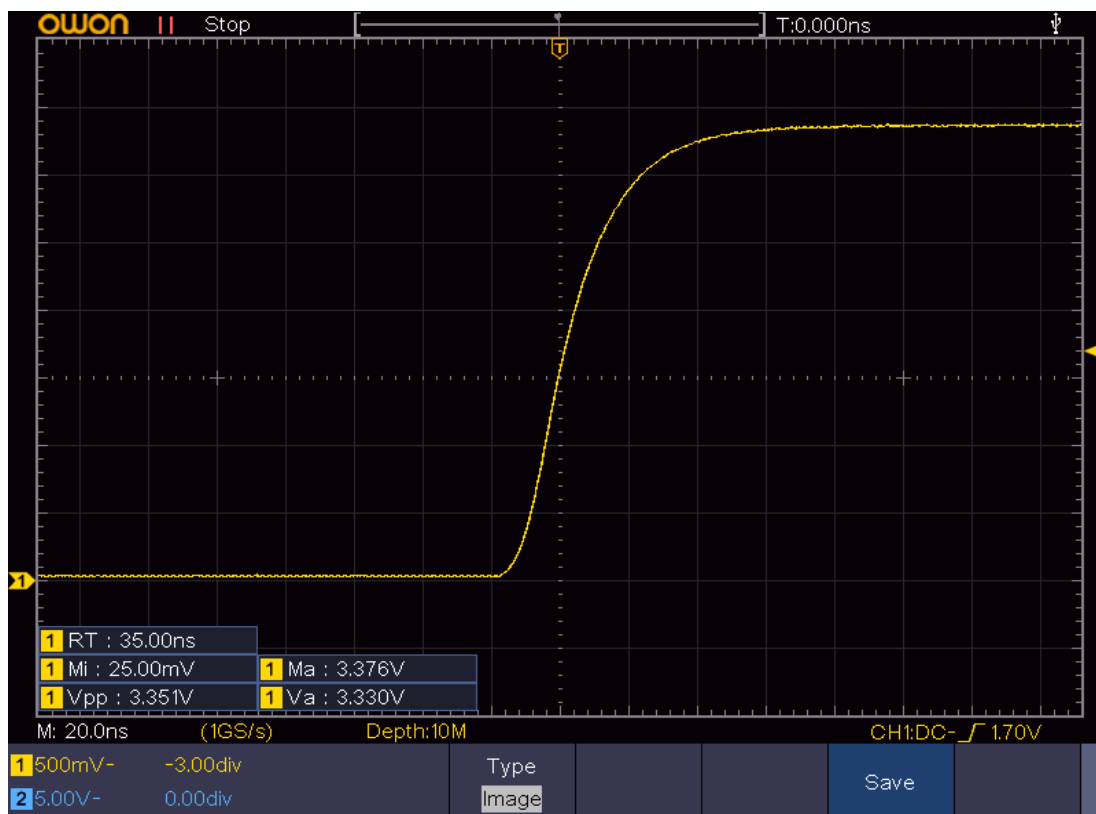


Figura 6.12: Escalón de 3,3 V_{PP} de amplitud utilizado como señal de prueba

En la figura 6.12 se observa un detalle del flanco ascendente de la señal utilizada, medida en el capacitor de acople de alterna del front-end, luego del conector BNC del primer canal. El uso de una punta de prueba X1 reduce el tiempo de crecimiento inicial del pulso digital origina a un valor aproximado de 35 ns.

La configuración del front-end utilizada en las futuras mediciones se corresponde con la escala de 0,2 V/div, cuya ganancia total es de 0,75 V/V. Esta escala se elige debido a que presenta un rango de entrada en V_{IN} de 4 V_{PP} (correspondientes a 3 V_{PP} en el ADC) por lo que representa el ajuste más apropiado para una entrada de 3,3 V_{PP} de amplitud. Esta configuración se corresponde con el atenuador de entrada de 0,1 veces, una constante del potenciómetro K_P de 0,67 y una ganancia del PGA de 5 V/V.

6.2.7 Acoplamiento de alterna

La primera etapa en el acondicionamiento de la señal es el acople de alterna. Para verificar la frecuencia de corte de la red y su respuesta temporal, se observa la forma de onda luego del capacitor de acople, para el caso de la señal analizada en la sección anterior. Debido a que el atenuador de entrada forma parte de la red de acople de alterna, la medición se repite para ambos atenuadores. A modo de ejemplo, en la figura 6.13 se presenta el resultado obtenido para el atenuador de 0,1 veces del primer canal:

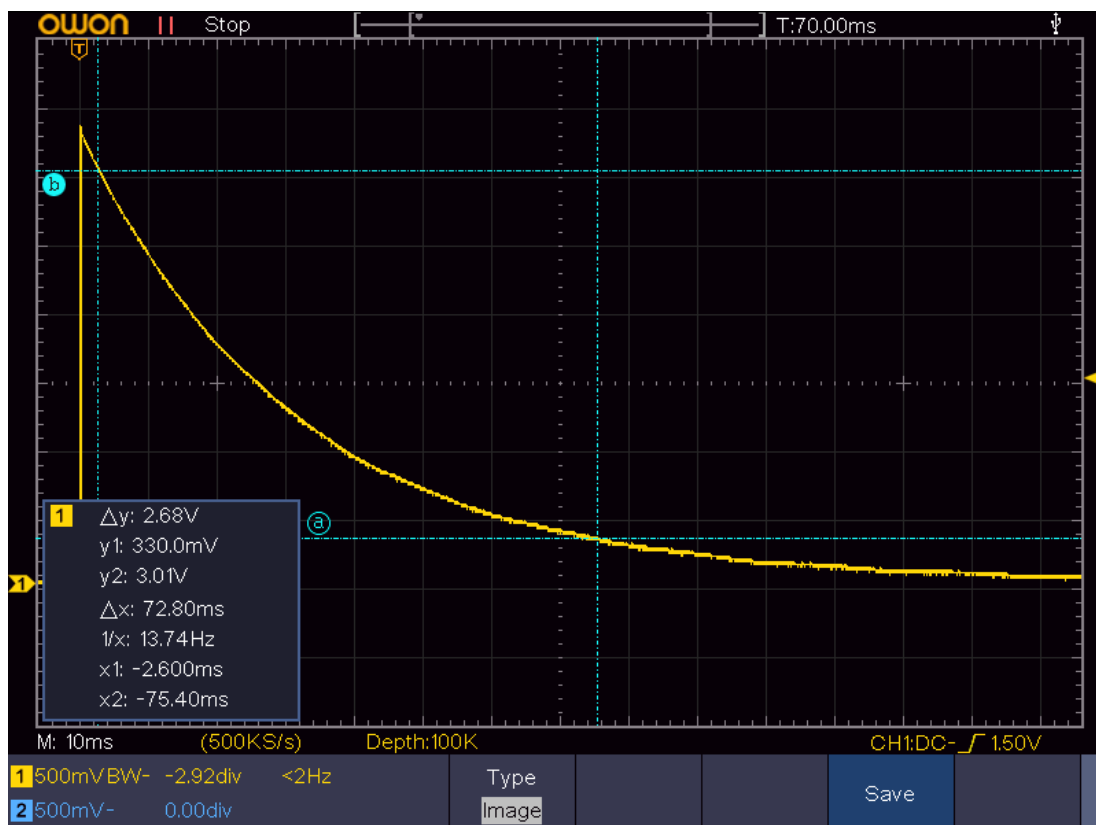


Figura 6.13: Respuesta del acople de alterna ante un escalón positivo de 3,3 V_{PP}

La forma de onda obtenida es la esperada para una red de tipo pasa altos, siendo el tiempo de decrecimiento medido, entre el 10% y el 90%, de 72,8 ms. En base a este valor se puede obtener una estimación de la frecuencia de corte de 3 dB, recurriendo a la siguiente expresión:

$$B(3\text{ dB}) = \frac{0,35}{t_r} = \frac{0,35}{72,8\text{ ms}} = 4,8\text{ Hz} \quad (6.5)$$

El valor de 4,8 Hz, obtenido en la ecuación 6.5, es idéntico al estipulado en los cálculos teóricos de la sección 3.4.2. Cabe destacar que no se observan modificaciones en los tiempos de decrecimiento para la selección del atenuador de 0,9 veces. Para el caso del segundo canal, el ancho de banda asociado al acople de alterna de ambos atenuadores es de 4,5 Hz.

La medición para el resto de combinaciones de atenuadores y canales brinda valores y formas de onda similares, por lo que se omiten los resultados.

6.2.8 Atenuador compensado

La siguiente etapa a ser caracterizada es el atenuador seleccionable de entrada. En primer lugar, se realiza el ajuste de compensación de los dos atenuadores de cada canal en base a las formas de onda visualizadas en el software de adquisición. Bajo estas condiciones, se procede a medir la salida del atenuador de 0,1 veces del primer canal, obteniendo la siguiente forma de onda:

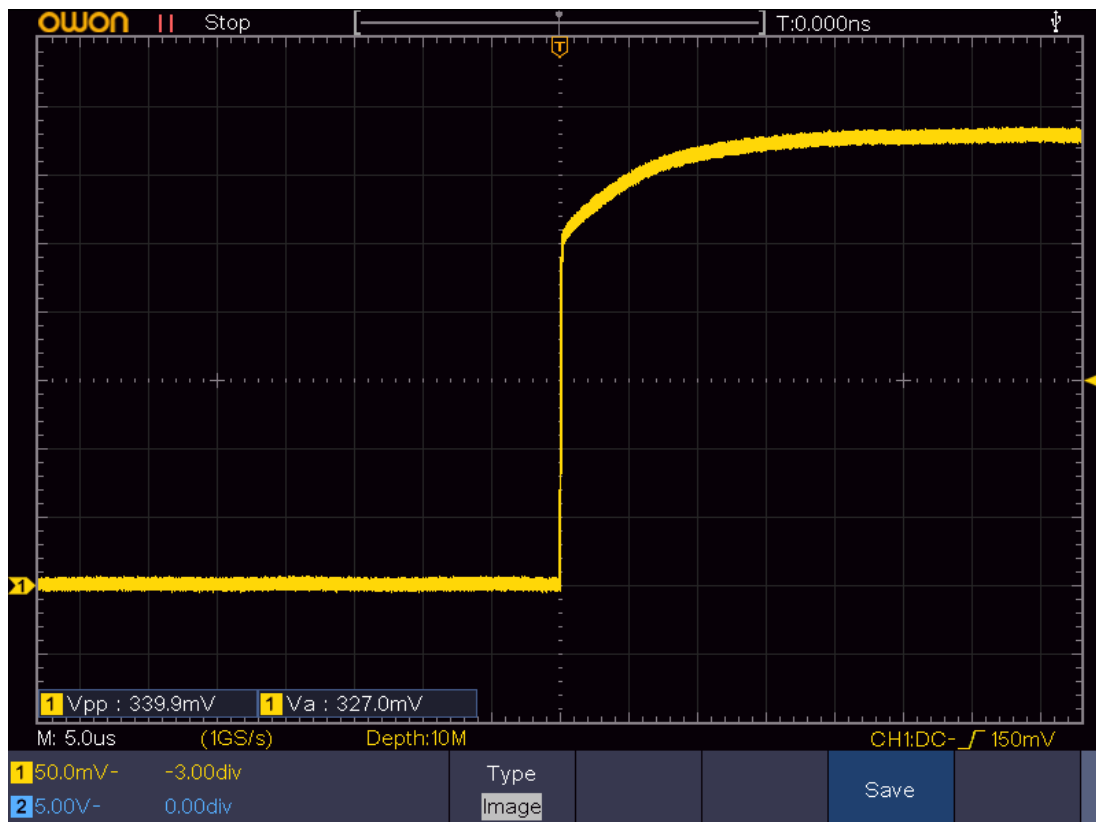


Figura 6.14: Salida del atenuador de 0,1 ante pulso digital de 3,3 V_{PP}

La respuesta al escalón, en la figura 6.14, es típica para una red con un polo y cero. El valor en el instante inicial está determinado por el divisor capacitivo, pudiendo ser mayor (subcompensación) o menor (sobrecompensación) al valor final, determinado por el divisor resistivo. En este caso, la amplitud medida de 327 mV_{PP} es la esperada para una señal de entrada de 3,3 V_{PP} y una atenuación de 10 veces.

El caso presentado en la figura 6.14 se corresponde con un ajuste correcto del atenuador realizado en base a la forma de onda visualizada en el software del cliente. Sin embargo, el efecto de carga de la punta de prueba del osciloscopio empleado en la medición modifica el ajuste, generando un caso de subcompensación. Cabe destacar que, con el objetivo de reducir la carga capacitiva, la medición se realiza con una punta de prueba en el ajuste X10 [100].

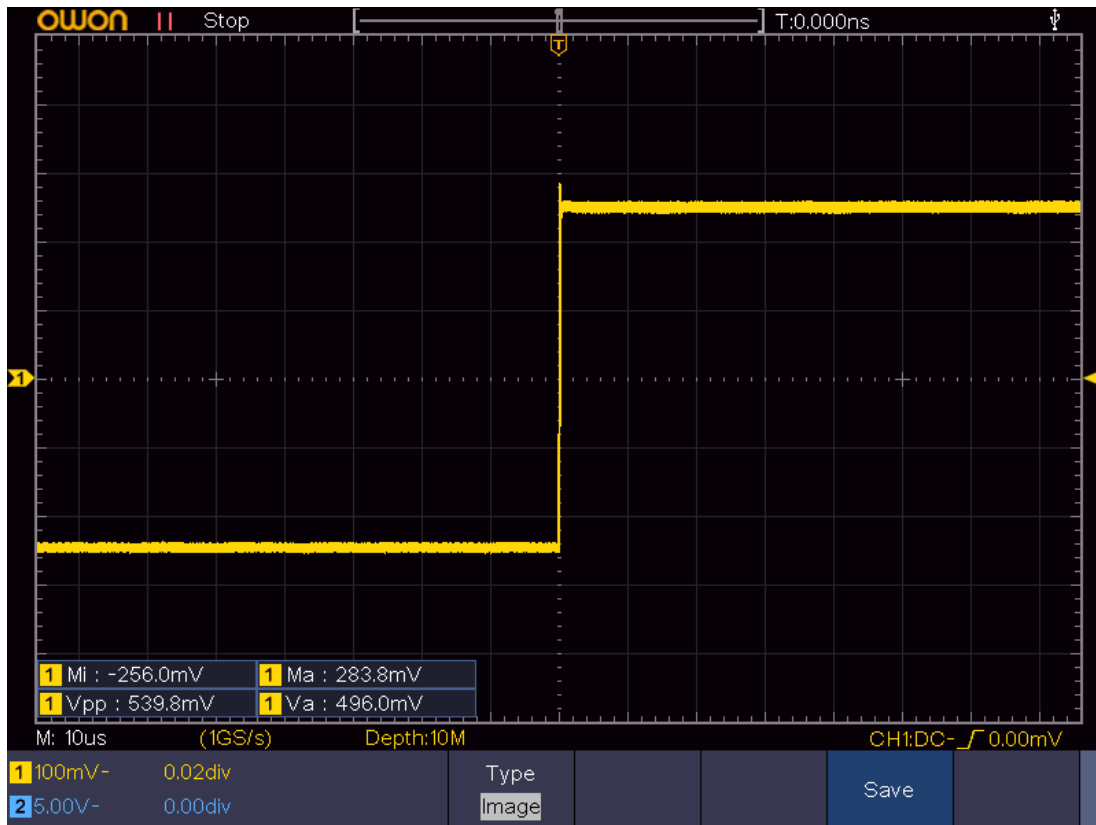


Figura 6.15: Salida del primer amplificador operacional ante pulso digital de 3,3 V_{PP}

Teniendo en cuenta lo mencionado, se decide caracterizar el atenuador en base a mediciones realizadas a la salida del primer amplificador operacional, lugar de baja impedancia. En la figura 6.15 se puede ver que la forma de onda es la esperada para un ajuste correcto de compensación. Teniendo en cuenta la amplitud a la salida del atenuador, de 330 mV_{PP}, el valor medido de 496 mV_{PP} en esta etapa es el esperado para una ganancia de 1,5 veces.

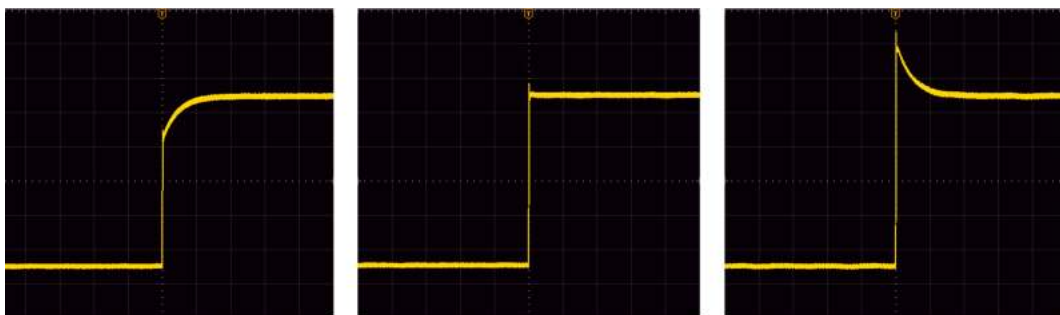


Figura 6.16: Salida del primer amplificador para los ajustes del atenuador de 0,1 veces

A modo demostrativo, se presenta en la figura 6.16 los tres casos posibles de ajuste para el atenuador de entrada. Las formas de onda se corresponden, nuevamente, al atenuador de 0,1

veces del primer canal medido a la salida del primer amplificador operacional. De izquierda a derecha se pueden observar los casos de sobrecompensación, ajuste correcto y subcompensación, respectivamente.

Durante esta fase se detectó un error menor de diseño en el atenuador de 0,9 veces. Para este caso, el rango de ajuste por diseño permite compensar capacidades parásitas de 0 pF a 19,4 pF. En base a las mediciones se estimó un valor aproximado de 15 pF para la capacidad de los diodos y la entrada del amplificador. Como consecuencia, el rango de ajuste no se encuentra centrado, correspondiéndose un 75% a sobrecompensación y el 25% restante a subcompensación. El rango de ajuste puede ser modificado fácilmente reemplazando uno de los capacitores del atenuador. Sin embargo, el error no fue corregido, ya que se puede realizar la compensación de todas formas.

6.2.9 Amplificador sumador de offset

En este apartado se analiza la respuesta al escalón del primer amplificador operacional, que actúa como buffer y sumador de offset. En la figura 6.15, de la sección anterior, se puede observar la presencia de un sobrepico en la respuesta temporal del amplificador. Esta situación se evidencia en mayor detalle a continuación:

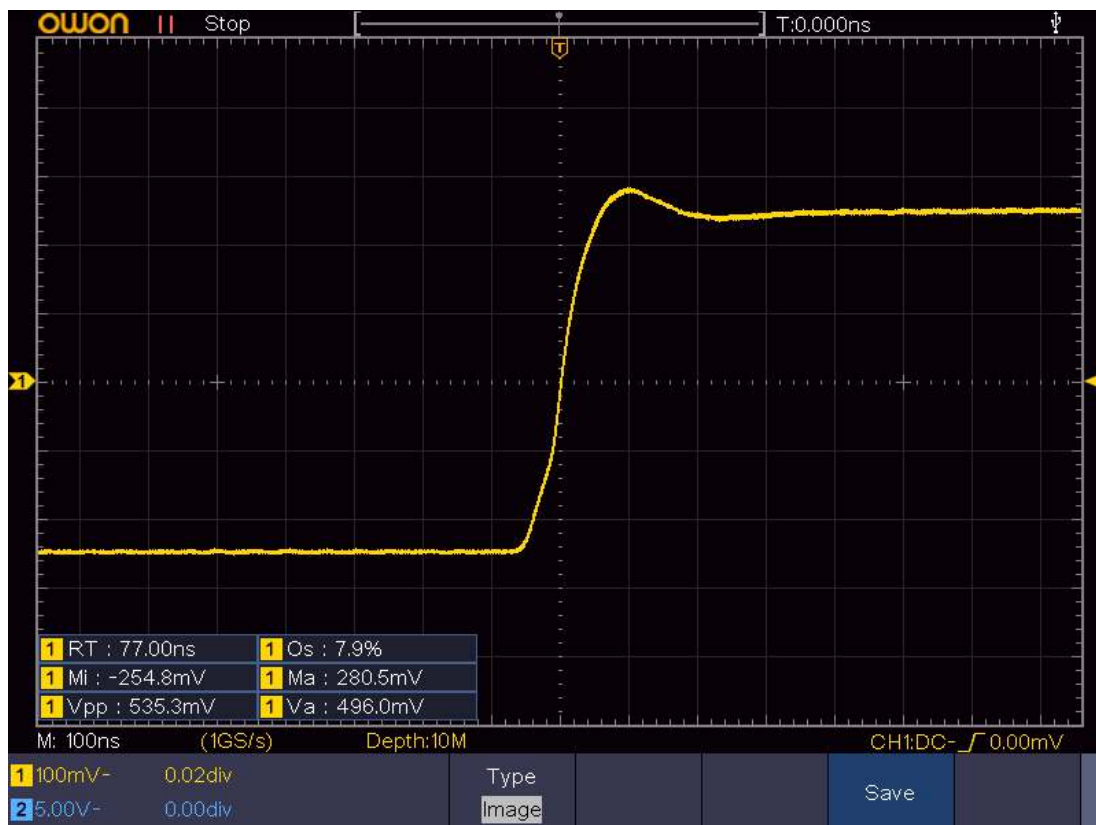


Figura 6.17: Detalle temporal de la salida del primer amplificador ante un pulso de 3,3 V_{PP}

La forma de onda visualizada, de acuerdo a la figura 6.17, tiene una amplitud de 496 mV_{PP} y un sobrepico aproximado de 39 mV (7,9%). El tipo de respuesta observado, denota un margen de fase y estabilidad menor al calculado teóricamente en la sección 3.4.5. En base a mediciones posteriores, se determinó que la diferencia se debe a la capacidad parásita a la entrada del potenciómetro digital de la siguiente etapa. Este elemento interactúa con la impedancia de salida del amplificador, de 375Ω [24], produciendo un polo que agrega fase en el cruce de 0 dB de la ganancia de lazo abierto.

El sobrepico observado se corresponde con un valor de capacidad parásita mayor al obtenido de la hoja de datos del fabricante del potenciómetro, de 5 pF [32]. En secciones posteriores, se demuestra que el sobrepico de esta etapa no afecta la respuesta final. El ancho de banda de esta etapa se puede estimar en base al tiempo de crecimiento.

$$t_r = \sqrt{(t_{r(out)})^2 - (t_{r(in)})^2} \quad (6.6)$$

La ecuación 6.6 permite calcular el tiempo de crecimiento asociado a una etapa en base a los tiempos de crecimiento medidos a la entrada y a la salida. Cabe destacar que se trata de una aproximación, aplicable a sistemas en los que la respuesta no tiene sobrepicos significativos. Teniendo en cuenta lo mencionado, se realiza el cálculo a continuación:

$$t_r = \sqrt{(77 \text{ ns})^2 - (35 \text{ ns})^2} = 68,6 \text{ ns} \quad (6.7)$$

$$B(3 \text{ dB}) = \frac{0,35}{68,6 \text{ ns}} = 5,1 \text{ MHz} \quad (6.8)$$

El valor obtenido, de 5,1 MHz, es levemente menor al obtenido teóricamente por simulación en la sección 3.4.5, de 8,6 MHz.

6.2.10 Potenciómetro digital y buffer de PGA

La siguiente etapa a ser caracterizada es el potenciómetro digital utilizado como atenuador variable de 256 pasos. Se procede a medir la salida del potenciómetro para el caso de una forma de onda cuadrada de 496 mV_{PP} a su entrada, es decir, la señal de la etapa anterior en la figura 6.17. Cabe recordar que la escala seleccionada, para la caracterización de todas las etapas, es de 0,2 V/div, ajuste que se corresponde con una constante del potenciómetro K_P de 0,67.

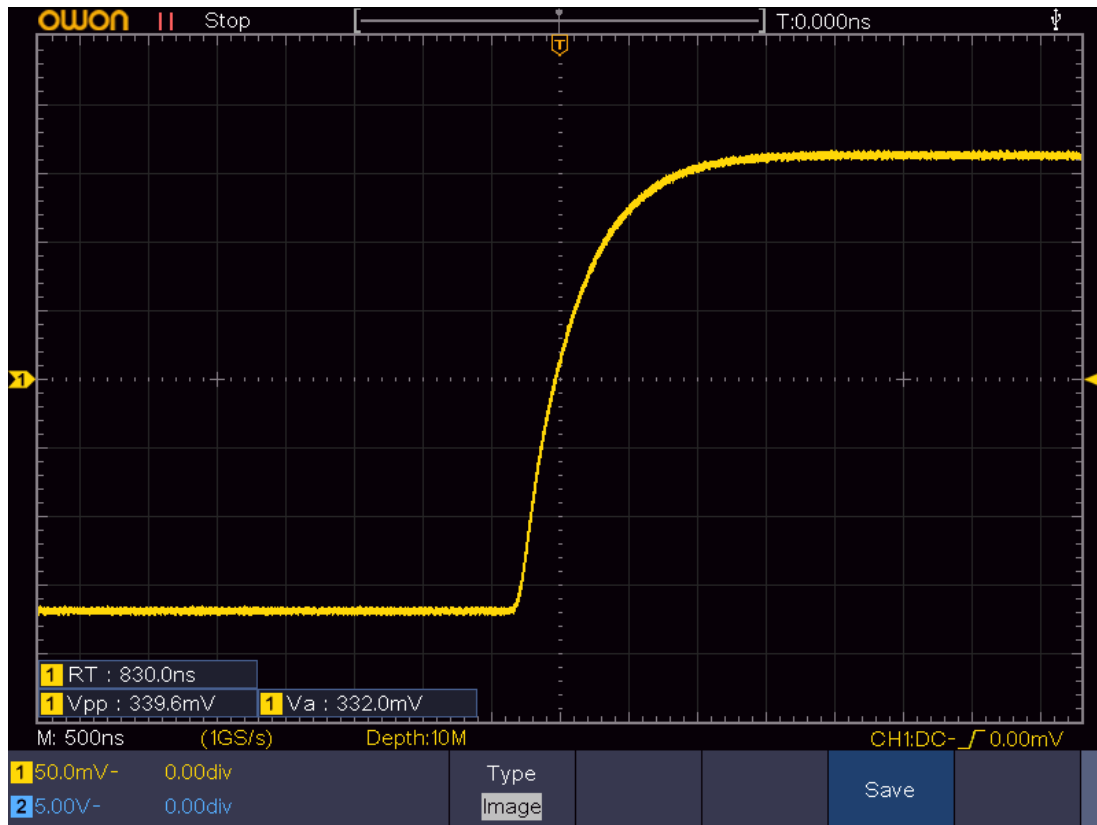


Figura 6.18: Salida del potenciómetro digital ante un pulso de $3,3 V_{PP}$ ($K_P 0,67$)

En la figura 6.18 se puede observar que la amplitud de la señal obtenida, de 332 mV_{PP} , es la esperada para el ajuste seleccionado. Sin embargo, se observa una respuesta tipo pasa bajos con un límite de banda considerablemente mayor al calculado teóricamente en la sección 3.4.6, de $4,9 \text{ MHz}$. Como consecuencia del efecto de filtrado, el sobrepico presente a la entrada es eliminado y el tiempo de crecimiento aumenta considerablemente. Dado que la capacidad de carga de la punta de prueba, aun en el ajuste X10, afecta la respuesta [100], se mide la forma de onda a la salida del amplificador posterior.

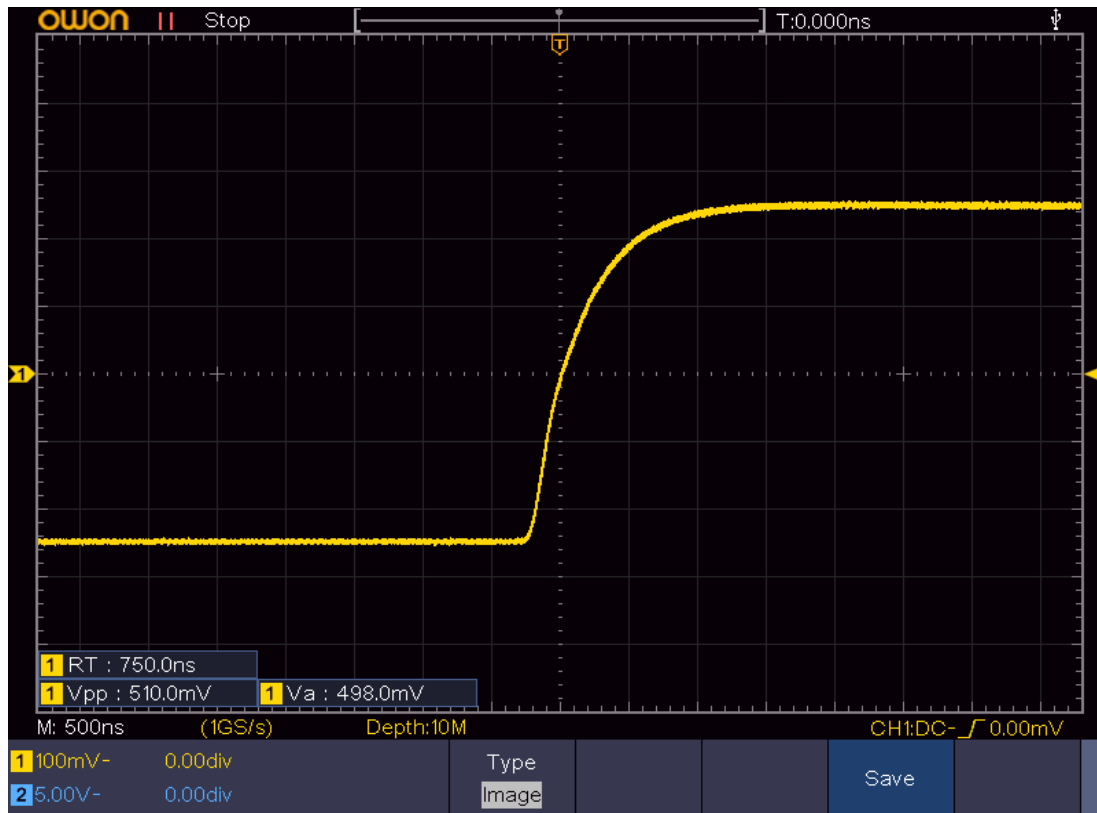


Figura 6.19: Salida del buffer del PGA ante un pulso de $3,3 V_{PP}$

En este caso, en la figura 6.19, se obtiene una amplitud de 498 mV_{PP} , asociada a la ganancia de 1,5 veces y un tiempo de crecimiento de 750 ns , similar al de la salida del potenciómetro. Al igual que en secciones anteriores, se hace uso del tiempo de crecimiento para estimar el ancho de banda de esta etapa. Los cálculos realizados se presentan a continuación:

$$t_r = \sqrt{(750 \text{ ns})^2 - (77 \text{ ns})^2} = 746 \text{ ns} \quad (6.9)$$

$$B(3 \text{ dB}) = \frac{0,35}{t_r} = \frac{0,35}{746 \text{ ns}} = 470 \text{ kHz} \quad (6.10)$$

La considerable diferencia entre la frecuencia de corte teórica esperada y la obtenida, es atribuible a una mayor capacidad de carga a la salida del potenciómetro. Para analizar la situación y estimar el valor de capacidad real, se recurre a los siguientes diagramas esquemáticos [33]:

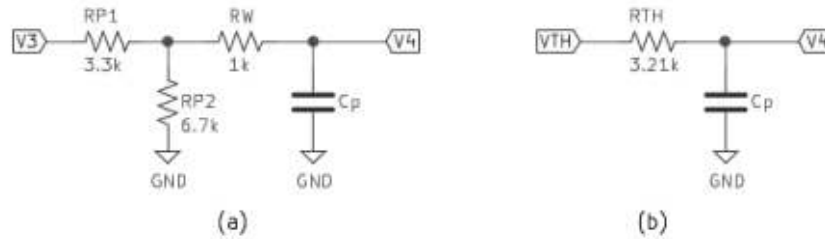


Figura 6.20: (a) Circuito equivalente de potenciómetro digital (b) Equivalente de Thévenin

En la figura 6.20(a) se puede observar un circuito equivalente del potenciómetro digital, teniendo en cuenta sus características reales. Las resistencias R_{P1} de 3,3 k Ω y R_{P2} de 6,7 k Ω conforman el atenuador para el caso de la constante K_P elegida, de 0,67. Por otro lado, R_W representa la resistencia no nula del contacto central del potenciómetro, o resistencia de wiper, su valor según la hoja de datos es aproximadamente 1 k Ω [32]. Finalmente, se agrega la capacidad C_P , asociada a la salida del potenciómetro digital y la entrada del amplificador operacional. Cabe recordar que, en el diseño teórico de la sección 3.4.6, se utiliza un valor de 7 pF para el potenciómetro y 6 pF para el amplificador operacional, de acuerdo al análisis de las hojas de datos de ambos componentes [32] [24].

En la figura 6.20(b), se presenta un equivalente de Thévenin del circuito de la figura 6.20(a). Teniendo en cuenta que se trata de una red pasa bajos de primer orden, se puede obtener la capacidad C_P a partir del ancho de banda. Los cálculos se presentan a continuación:

$$R_{TH} = (R_{P1} // R_{P2}) + R_W = 3,21 \text{ k}\Omega \quad (6.11)$$

$$B(3 \text{ dB}) = \frac{1}{2 \cdot \pi \cdot R_{TH} \cdot C_P} = 470 \text{ kHz} \rightarrow C_P = 105 \text{ pF} \quad (6.12)$$

Debido a la considerable diferencia entre los valores obtenidos y los teóricos esperados, se realizaron diferentes mediciones para validar el ancho de banda de la etapa y la capacidad calculada, obteniendo resultados similares. El aumento en la capacidad de carga es atribuido, principalmente, al potenciómetro digital utilizado.

Un análisis de las hojas de datos de potenciómetros similares del mismo fabricante revela valores de capacidad de salida en el orden de los 50 pF [101]–[103]. Teniendo en cuenta lo mencionado, se concluye que el valor de capacidad utilizado en el diseño, de 7 pF, es incorrecto. Esto se atribuye a un error en la hoja de datos [32] o, en un defecto, a una interpretación incorrecta de la misma.

Esta discrepancia afecta el cálculo de ancho de banda total para la etapa de acondicionamiento. Bajo estas condiciones, el elemento que fija la frecuencia de corte es el potenciómetro digital, en lugar del PGA. Cabe destacar que, en la sección 6.2.13 se analizan los anchos de banda totales asociados al front-end para todas las escalas verticales.

6.2.11 PGA y buffer de ADC

El siguiente elemento a analizar es el amplificador de ganancia programable, cuya ganancia para el ajuste de escala utilizado es de 5 V/V. A su salida se obtiene una amplitud de 2,48 V y un tiempo de crecimiento de 750 ns, este último idéntico al de la etapa anterior. Este resultado es el esperado, teniendo en cuenta que el ancho de banda del PGA, de 5 MHz, es considerablemente superior al de la etapa anterior. Por otro lado, en la figura 6.21, se presenta la forma de onda final, a la salida del buffer al ADC.

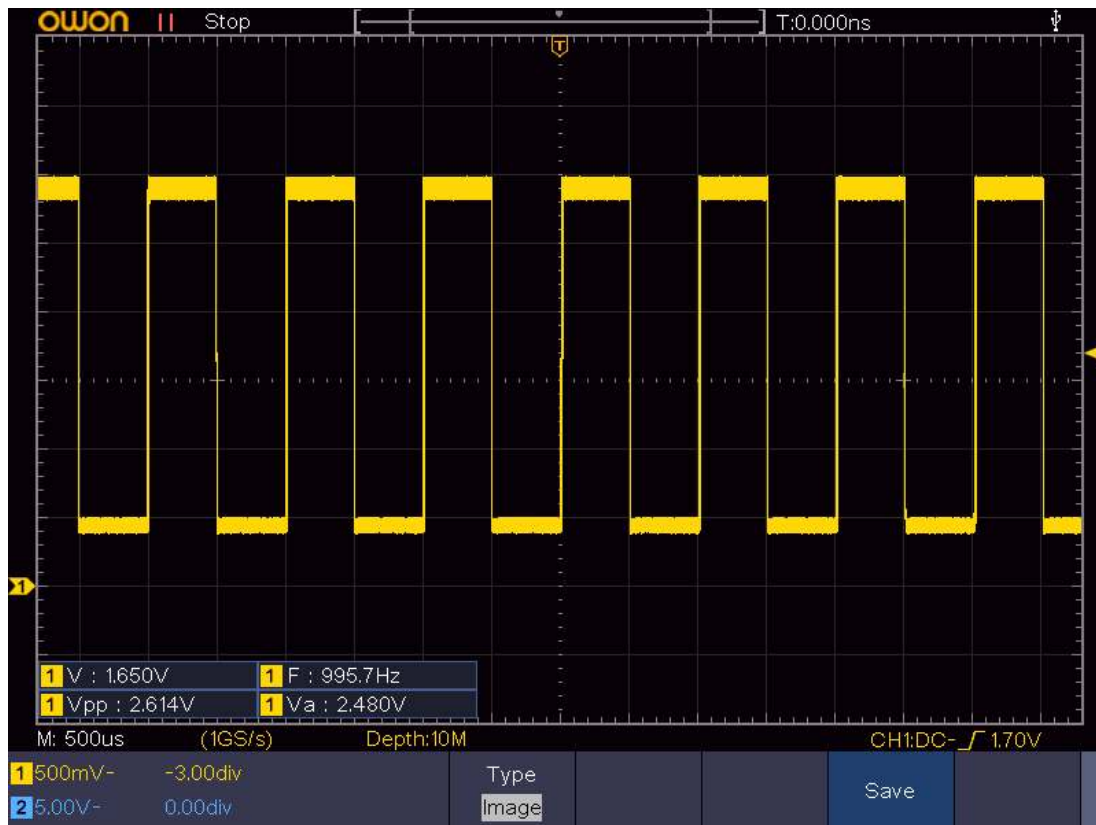


Figura 6.21: Forma de onda final ante un pulso de 3,3 V_{PP} en la escala 0,2 V/div

La medición de la figura 6.21 es realizada bajo acople de continua, lo que permite visualizar el desplazamiento de la señal a un valor medio de 1,65 V. En este caso, se obtiene una amplitud y tiempo de crecimiento idénticos a la etapa anterior, es decir, 2,48 V y 750 ns. Estos valores son los esperados en base a los cálculos realizados en el diseño teórico.

Es pertinente aclarar que el trazo grueso en la señal cuadrada no se debe a la presencia de ruido significativo en las etapas anteriores. El efecto se asocia al proceso de muestreo y retención realizado por los ADC de arquitectura SAR o aproximaciones sucesivas [104]. A fin de visualizar mejor el fenómeno, se presenta la siguiente captura de pantalla:

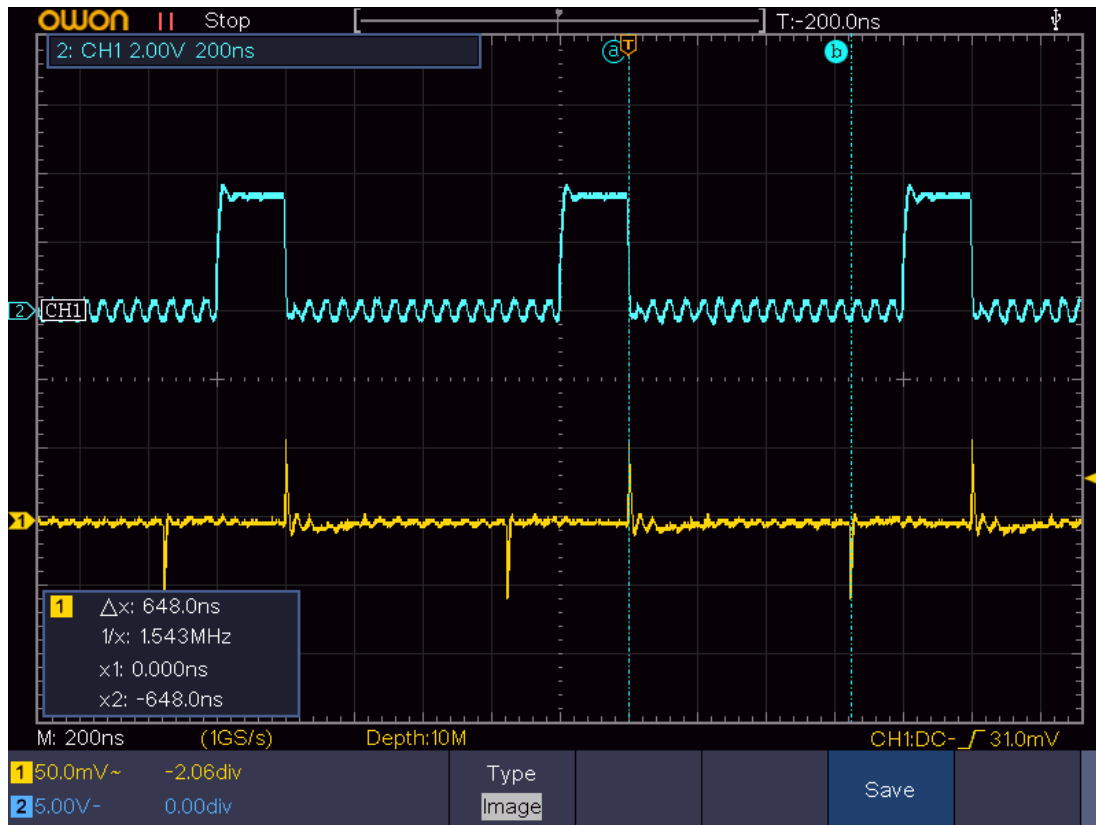


Figura 6.22: Efecto de muestreo y retención de ADC en arquitectura SAR

En la figura 6.22 se realiza una medición, bajo acople de alterna, de la salida del buffer al ADC para el caso de entrada nula, es decir, el nivel de 1,65 V. Adicionalmente, se presenta la señal de chip select, proveniente de la FPGA, para la interfaz SPI del conversor. De acuerdo a la hoja de datos del ADC [36], el flanco descendente de CS finaliza el proceso de muestreo, desconectando el capacitor de muestreo de la entrada del ADC. Esto se asocia con el flanco ascendente observado en la señal analógica. Durante los siguientes ciclos de reloj se realiza el proceso de conversión, en la etapa de retención. Finalmente, luego de 13 ciclos o 650 ns el capacitor se vuelve a conectar a la salida del buffer, generando un flanco descendente en la señal analógica, asociado al inicio de la nueva etapa de retención.

En este tipo de conversores resulta pertinente verificar que el ancho de banda de la etapa previa al conversor permita una respuesta lo suficientemente rápida ante el cambio de impedancia del ADC [105]. En la figura 6.22 se puede observar que esta situación se cumple, es decir, la señal regresa a su estado estacionario entre dos pulsos consecutivos, siendo la amplitud de los pulsos de 100 mV_{PP}.

6.2.12 Protección de entrada

En esta sección se evalúa el comportamiento de los circuitos involucrados en la protección ante sobretensiones en la entrada. Cabe recordar que el mecanismo actúa cuando se supera el rango aproximado de ± 4 V en la entrada no inversora del primer amplificador operacional. Dicha

situación se puede dar por dos razones: una entrada fuera del rango máximo del front-end o una selección incorrecta de escala, y por consiguiente, de atenuador de entrada.

Con el objetivo de caracterizar la red de protección, se ingresa al front-end con un pulso de 0 V a 25 V, con el atenuador de 0,9 seleccionado. Esta situación se corresponde con una tensión en la entrada no inversora de 22,5 V por lo que la protección se activa limitando la tensión. Cabe destacar que el tiempo de crecimiento del pulso utilizado es menor a 10 ns.

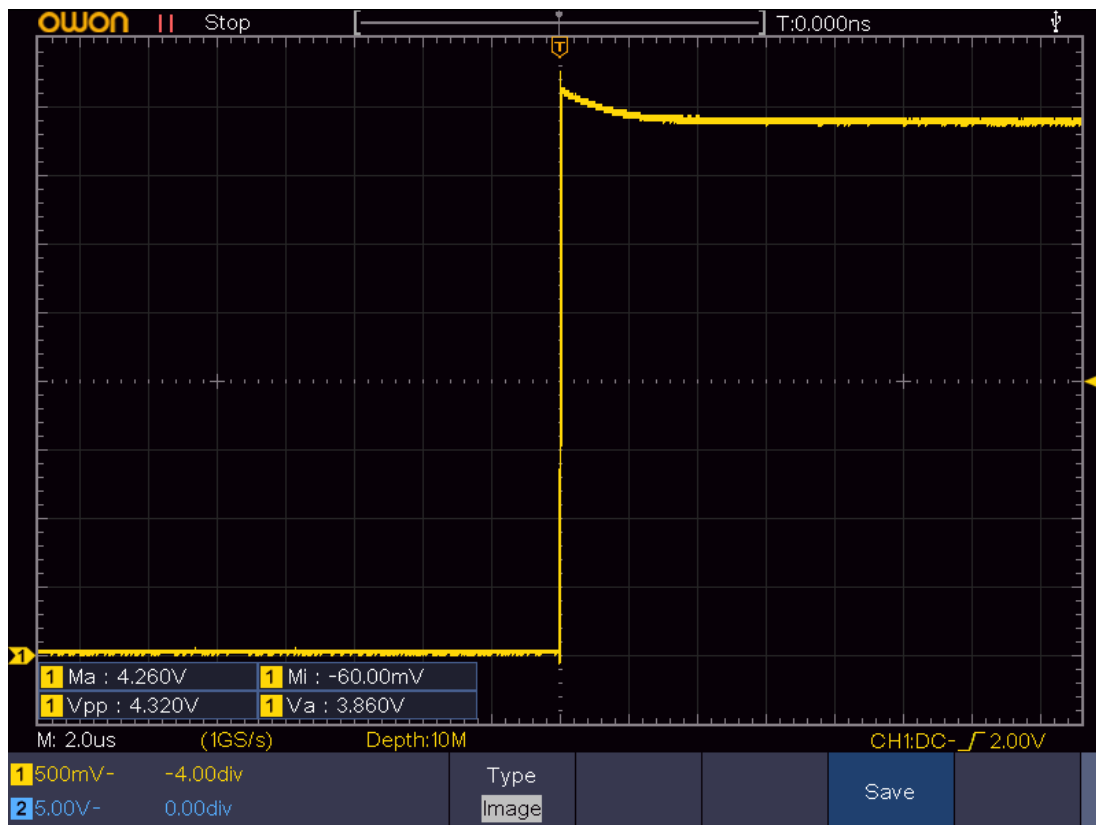


Figura 6.23: Protección ante un pulso de 25 V y una selección del atenuador de 0,9

En la figura 6.23 se observa la forma de onda obtenida en los diodos de protección. Bajo estas condiciones de prueba, la tensión final es de 3,86 V siendo el máximo valor 4,26 V. La convergencia al valor final es de tipo exponencial, con un tiempo de respuesta de la protección aproximado de 2 μs. La misma medición es repetida ante el caso de un pulso negativo (0 V a -25 V), obteniéndose una tensión final de -3,82 V y un mínimo de -4,18 V, siendo el tiempo de respuesta idéntico. Finalmente, se concluye que los resultados obtenidos son los esperados a nivel teórico, por lo que el circuito implementado funciona correctamente.

6.2.13 Ancho de banda total

En esta sección se presenta una tabla con el ancho de banda de 3 dB obtenido para cada escala. Con el objetivo de realizar esta medición, se utiliza una entrada de tipo senoidal, cuya amplitud es la apropiada para cada escala y se ajusta su frecuencia hasta obtener una caída de 3 dB en la

entrada del ADC. Los resultados obtenidos se enumeran en la tabla 6.6, presentada a continuación:

Escala (V/div)	G_{TOT} (V/V)	V_{IN} (V_{PP})	V_7 (3 dB) (V_{PP})	B (3 dB) CH1 (kHz)	B (3 dB) CH2 (kHz)
5	0,06	20	0,85	450	500
2	0,075	20	1,06	420	480
1	0,15	10	1,06	420	490
0,5	0,3	5	1,06	420	490
0,2	0,75	2	1,06	420	480
0,1	1,5	1	1,06	500	580
0,05	3	0,5	1,06	500	580
0,02	7,5	0,2	1,06	480	550
0,01	15	0,1	1,06	450	500

Tabla 6.6: Anchos de banda obtenidos para cada escala

En la primera columna de la tabla 6.6 se listan las ganancias totales de la etapa de acondicionamiento, es decir, desde V_{IN} a V_7 . En la siguiente columna se muestran los valores de amplitud de entrada utilizados. Las tensiones son elegidas de manera que representen un 50% del rango total, correspondiente a una excursión de $1,5 V_{PP}$ a la salida, es decir, $1,06 V_{PP}$ a la frecuencia de corte. La única excepción es el caso de la escala de 5 V/div, ya que el valor requerido es superior al rango máximo del generador de funciones utilizado. En las últimas dos columnas se presentan los anchos de banda obtenidos, cuyos valores oscilan en el rango 420 kHz a 580 kHz.

A pesar de que los anchos de banda reales son menores a los esperados teóricamente, resultan suficientes de acuerdo a la velocidad de muestreo del sistema de adquisición. Idealmente, la máxima frecuencia posible a la entrada es de 500 kHz, de acuerdo al teorema de Nyquist. Sin embargo, en la práctica los algoritmos de interpolación son efectivos con una frecuencia de muestreo, al menos, 2,5 veces superior a la máxima componente frecuencial. Del análisis realizado, se concluye que el ancho de banda obtenido, cuyo mínimo valor es 420 kHz, incluso permite una incorporación a futuro de algoritmos de interpolación de los datos.

6.2.14 Análisis de ruido

Finalmente, se lleva a cabo un análisis del ruido total asociado a la etapa de acondicionamiento. Luego de evaluar varias alternativas, se considera que la mejor caracterización del ruido está dada por el valor pico a pico observado en los datos adquiridos por el ADC, bajo la situación de entrada nula. Para realizar esta prueba, se analiza una muestra de los valores numéricos de 12 bits enviados por el ADC para el caso de entrada nula.

El valor pico a pico se determina como la diferencia entre el máximo y mínimo valor presente en los datos. Se emplea un tamaño de muestra de 100.000 valores, lo cual se corresponde con 0,1 s de adquisición a 1 MSa/s. Este tamaño de muestra permite que existan cinco ciclos de línea, ya que el mismo representa la mayor contribución de ruido en la excursión pico a pico observada.

Escala (V/div)	G_{TOT} (V/V)	G_{PGA} (V/V)	Ruido (ADC) CH1 (LSBs)	Ruido (ADC) CH2 (LSBs)	Ruido (V_{IN}) CH1 (mV _{pp})	Ruido (V_{IN}) CH2 (mV _{pp})
5	0,06	1	2	2	26,86	26,86
2	0,075	1	2	2	21,48	21,48
1	0,15	1	2	2	10,74	10,74
0,5	0,3	2	3	4	8,06	10,74
0,2	0,75	5	7	9	7,52	9,67
0,1	1,5	1	2	2	1,07	1,07
0,05	3	2	3	4	0,81	1,07
0,02	7,5	5	7	9	0,75	0,97
0,01	15	10	13	17	0,70	0,91

Tabla 6.7: Ruido en el ADC y referido a la entrada para cada escala

En la tabla 6.7 se pueden observar los resultados obtenidos en esta prueba. Los niveles de ruido se encuentran en el rango 1 LSB a 17 LSB, siendo el peor caso para la escala de 10 mV/div del segundo canal. Se listan también las ganancias del PGA para cada ajuste de escala, dado que existe una fuerte correlación entre su valor y el nivel de ruido observado.

Adicionalmente, en las últimas dos columnas se listan los valores de ruido pico a pico equivalente en la entrada del front-end. Para hacer la estimación, se recurre a la siguiente relación:

$$R(V_{IN}) = \frac{V_{REF}}{2^N} \cdot N_{ADC} \cdot (G_{TOT})^{-1} \quad (6.13)$$

El primer término en la fórmula 6.13 se corresponde con la resolución del ADC utilizado, en este caso aproximadamente 805,6 μ V. El segundo término, representa los valores medidos de cantidad de LSB de ruido. A fin de reflejar el nivel de ruido a un rango de entrada, se divide en el tercer término por la ganancia total del front-end para cada escala. Finalmente, del análisis realizado, se deduce que los niveles son aceptables, obteniéndose una buena resolución en la entrada para todas las escalas.

6.3 Pruebas integrales

En la etapa final de verificación, se realizan mediciones que involucran a todos los elementos del sistema de adquisición: front-end analógico, servidor y cliente de visualización. Las pruebas llevadas a cabo permiten cumplir con los siguientes objetivos:

- Verificar el correcto funcionamiento del sistema en su totalidad y la comunicación entre sus componentes.
- Demostrar la utilidad de los diferentes controles y funcionalidades a la hora de visualizar y analizar señales de diversa índole.
- Evaluar el comportamiento de los algoritmos de procesamiento y la arquitectura de adquisición.
- Evaluar la performance de la etapa de acondicionamiento analógica en sus límites de funcionamiento.

Para obtener las señales de entrada requeridas, se emplean dos generadores de funciones HP 33120A [98], es decir, uno por canal. Este instrumento cuenta con un ancho de banda de 15 MHz y es capaz de sintetizar formas de onda senoidales, cuadradas y triangulares, entre otras. Adicionalmente, el generador cuenta con la capacidad de generar ruido blanco Gaussiano y formas de onda arbitrarias. La conexión entre el instrumento y el sistema de adquisición se realiza mediante un cable coaxial de 50 Ω , utilizando los conectores tipo BNC del front-end.

6.3.1 Ajustes verticales y resolución

Las primeras pruebas tienen como objetivo evaluar la capacidad del front-end para visualizar señales de pequeña amplitud. A fin de aumentar la complejidad de la medición, se agrega un offset de voltaje a la entrada, modificando su valor medio. La configuración utilizada para la medición inicial es la siguiente:

- Señal de entrada (canal 1): cuadrada, frecuencia 1 kHz (periodo 1 ms), amplitud 0,1 V_{PP}, offset de 0,2 V.
- Ajuste de disparo: fuente canal 1, flanco ascendente, nivel 0,2 V, histéresis 1%.
- Ajuste vertical (canal 1): escala 5 V/div, posición vertical 0 divisiones, acople de continua, offset 0 V.
- Ajuste horizontal: escala 0,5 ms/div (5 ms total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 5.000 (1 MSa/s).
- Ajuste de visualización: multi-plot, zoom desactivado, persistencia desactivada.

Cabe destacar que la amplitud de la señal es la mínima posible para el generador utilizado [98]. De manera similar, el valor de offset es el máximo posible para el nivel de amplitud seleccionado. La forma de onda elegida, de tipo cuadrada, permite analizar y visualizar cualquier tipo de transitorio o alinealidad presente en la etapa analógica.

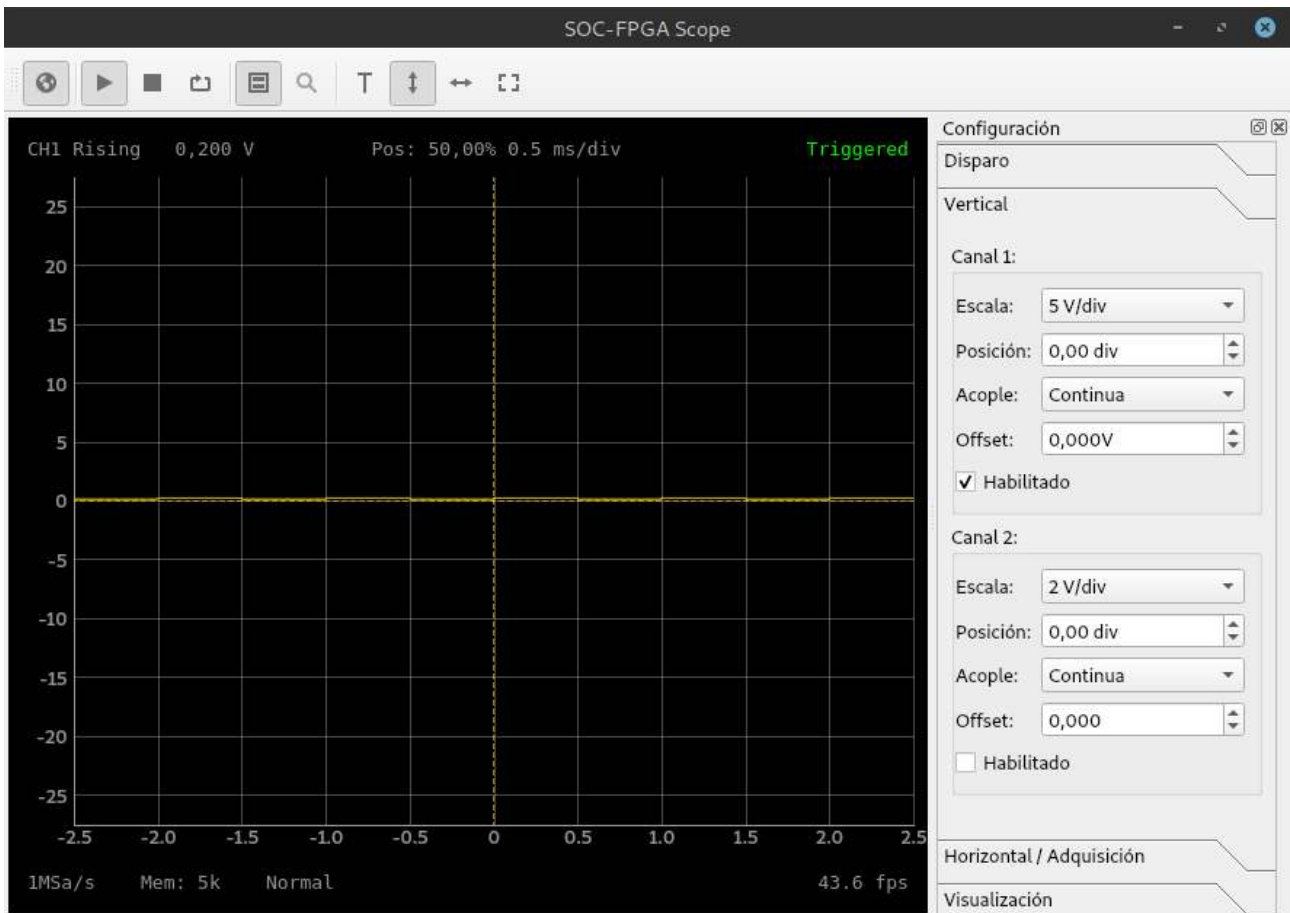


Figura 6.24: Señal cuadrada de 1 kHz observada en escala de 5 V/div

De acuerdo a la figura 6.24, bajo esta configuración la señal es apenas discernible. Con el objetivo de mejorar la medición, se amplía la escala vertical a 50 mV/div, máximo valor posible para la amplitud de entrada. A continuación, se presenta la forma de onda obtenida:

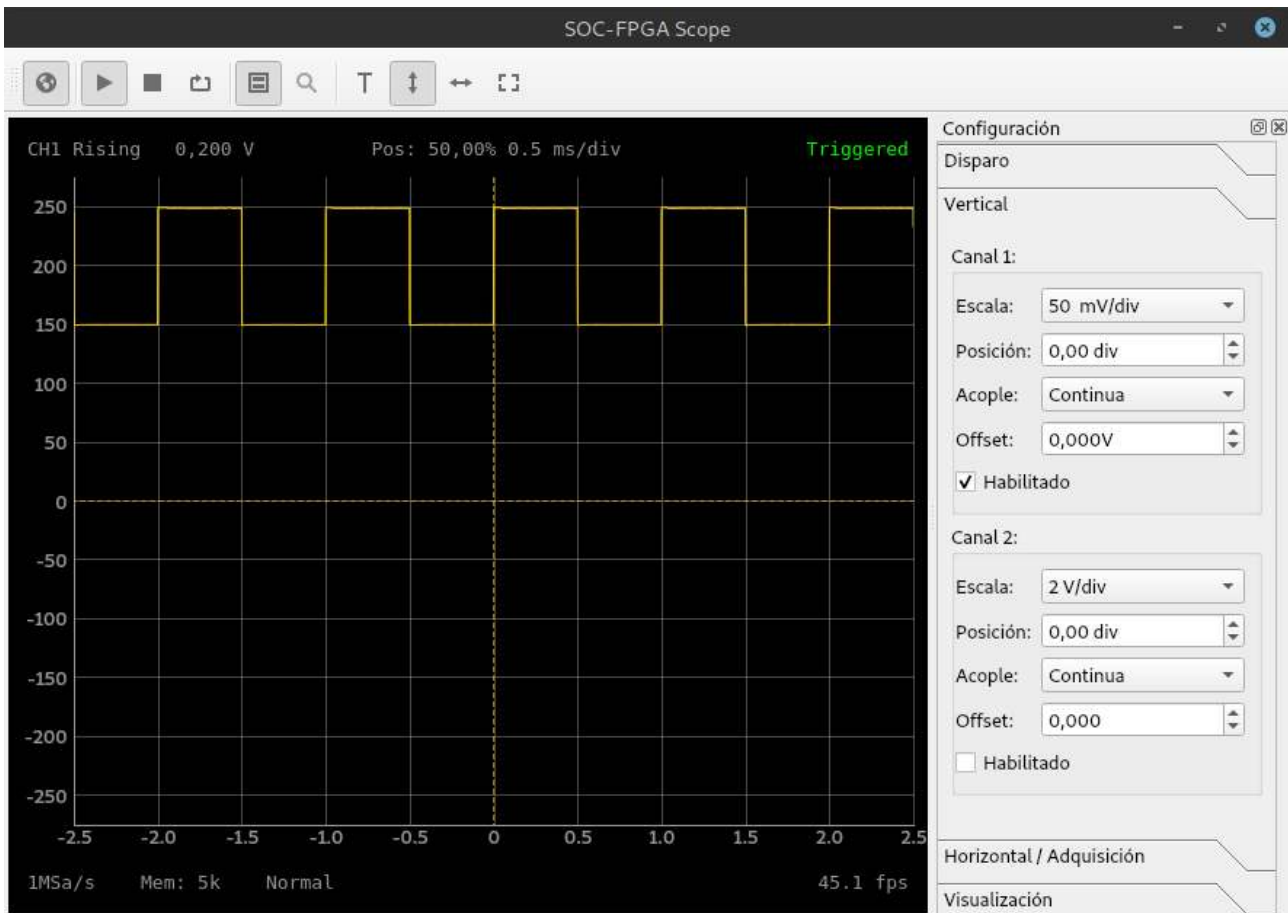


Figura 6.25: Señal cuadrada de 1 kHz observada en escala de 50 mV/div

Se puede ver, en la figura 6.25, que el sistema presenta una buena respuesta ante un pulso cuadrado a la entrada. De la forma de onda obtenida se concluye que, para la frecuencia de muestreo del sistema, el ancho de banda del front-end resulta suficiente. Finalmente, para obtener la mejor visualización posible, se emplea la funcionalidad de offset de voltaje por hardware, restando el valor de continua de 0,2 V.

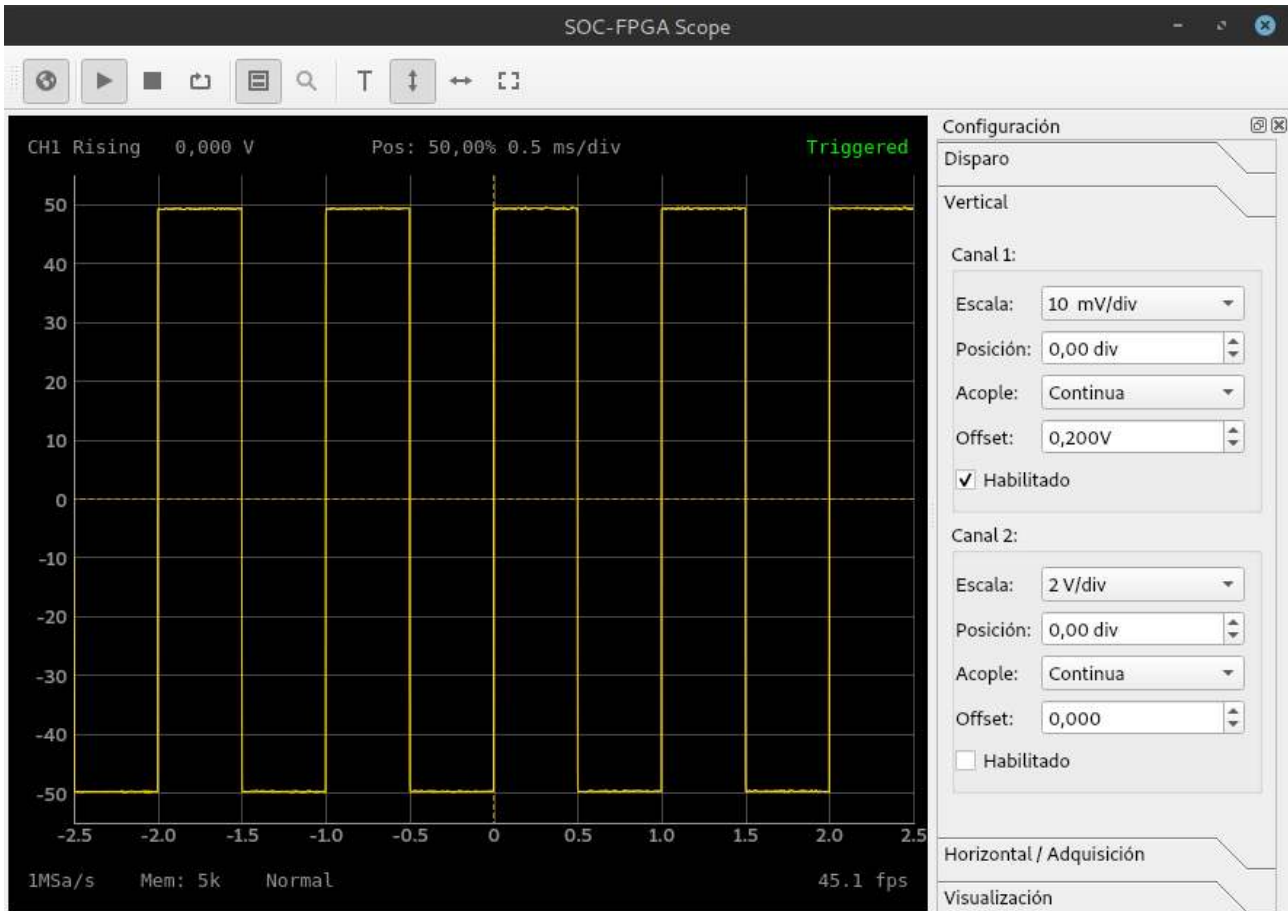


Figura 6.26: Señal cuadrada de 1 kHz observada en escala de 10 mV/div (offset restado)

En este caso, la funcionalidad de offset permite utilizar el mínimo valor de escala vertical disponible, de 10 mV/div. En esta medición, es destacable el bajo nivel de ruido presente y la precisión y exactitud del sistema. Un elemento que influye significativamente en el resultado obtenido es el hecho de emplear algoritmos de calibración de ganancia y offset.

6.3.2 Acople de alterna y offset de voltaje

La siguiente serie de pruebas realizadas tiene como objetivo demostrar uno de los casos posibles en los que la funcionalidad de offset de voltaje permite una mejor medición que el acople de alterna. Los ajustes utilizados para la primera medición, realizada bajo acople de alterna, son los siguientes:

- Señal de entrada (canal 1): cuadrada, frecuencia 10 Hz (periodo 0,1 s), amplitud 5 V_{pp}, offset de 5 V.
- Ajuste de disparo: fuente canal 1, flanco ascendente, nivel 0 V, histéresis 1%.
- Ajuste vertical (canal 1): escala 1 V/div, posición vertical 0 divisiones, acople de alterna, offset 0 V.

- Ajuste horizontal: escala 50 ms/div (0,5 s total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 5.000 (10 kSa/s).
- Ajuste de visualización: multi-plot, zoom desactivado, persistencia desactivada.

En este caso, se elige el mayor valor de amplitud y offset posible en el generador de funciones [98], situación que permite evaluar el comportamiento del front-end ante rangos amplios de señales.

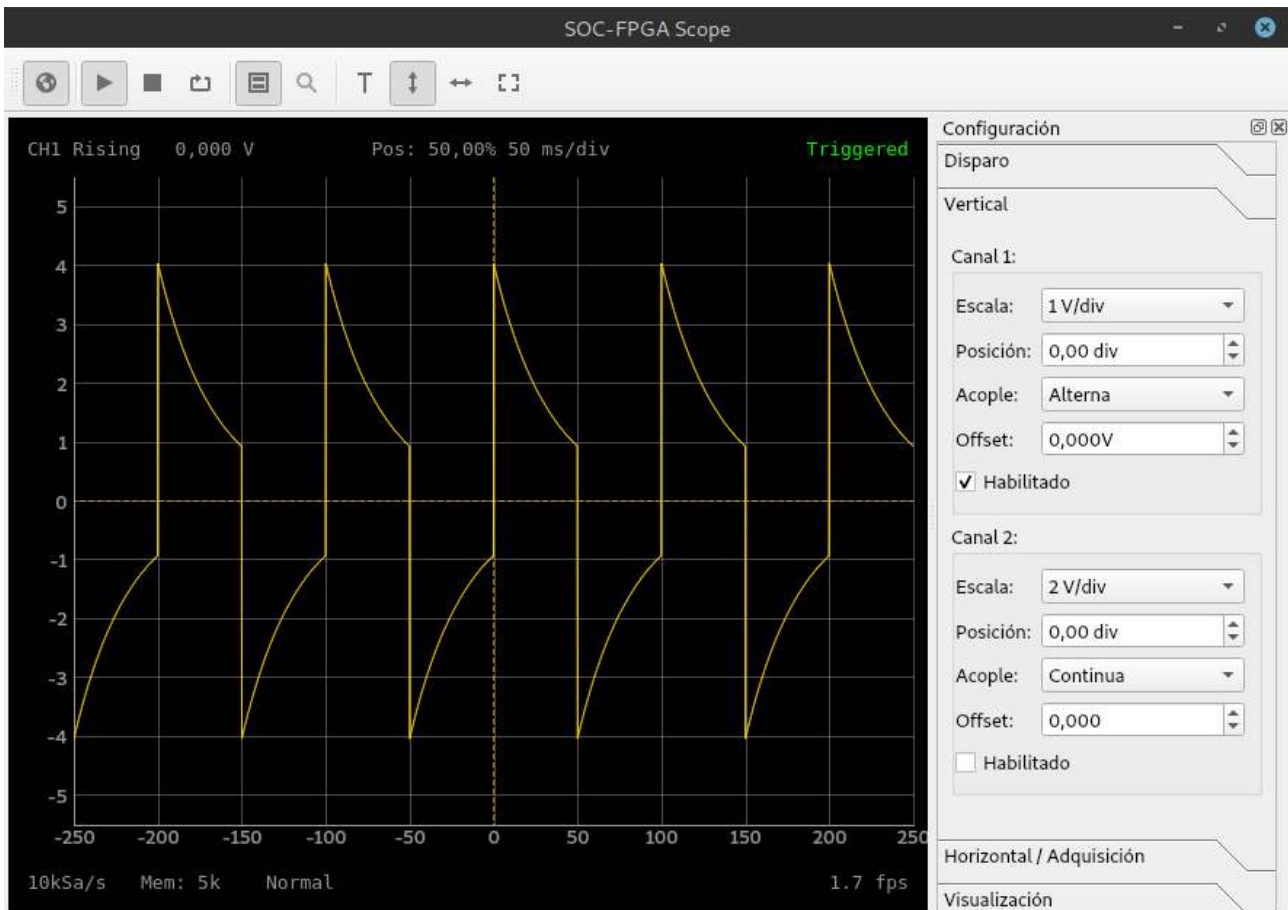


Figura 6.27: Señal cuadrada de 10 Hz y 5 V_{PP} observada bajo acople de alterna

Debido a la baja frecuencia de la señal de entrada, la respuesta del acople de alterna modifica considerablemente la forma de onda original. En este caso, el ancho de cada pulso, de 50 ms, se corresponde con 1,5 constantes de tiempo del filtro de acople, siendo la constante 33 ms. Bajo esta situación, resulta evidente que el efecto del filtro no es despreciable.

Con el objetivo de mejorar la medición, se desactiva el acople de alterna y se resta el valor medio de 5 V mediante el ajuste de offset. Bajo estas condiciones, es posible seleccionar la escala vertical de 0,5 V/div, en lugar de 1 V/div.

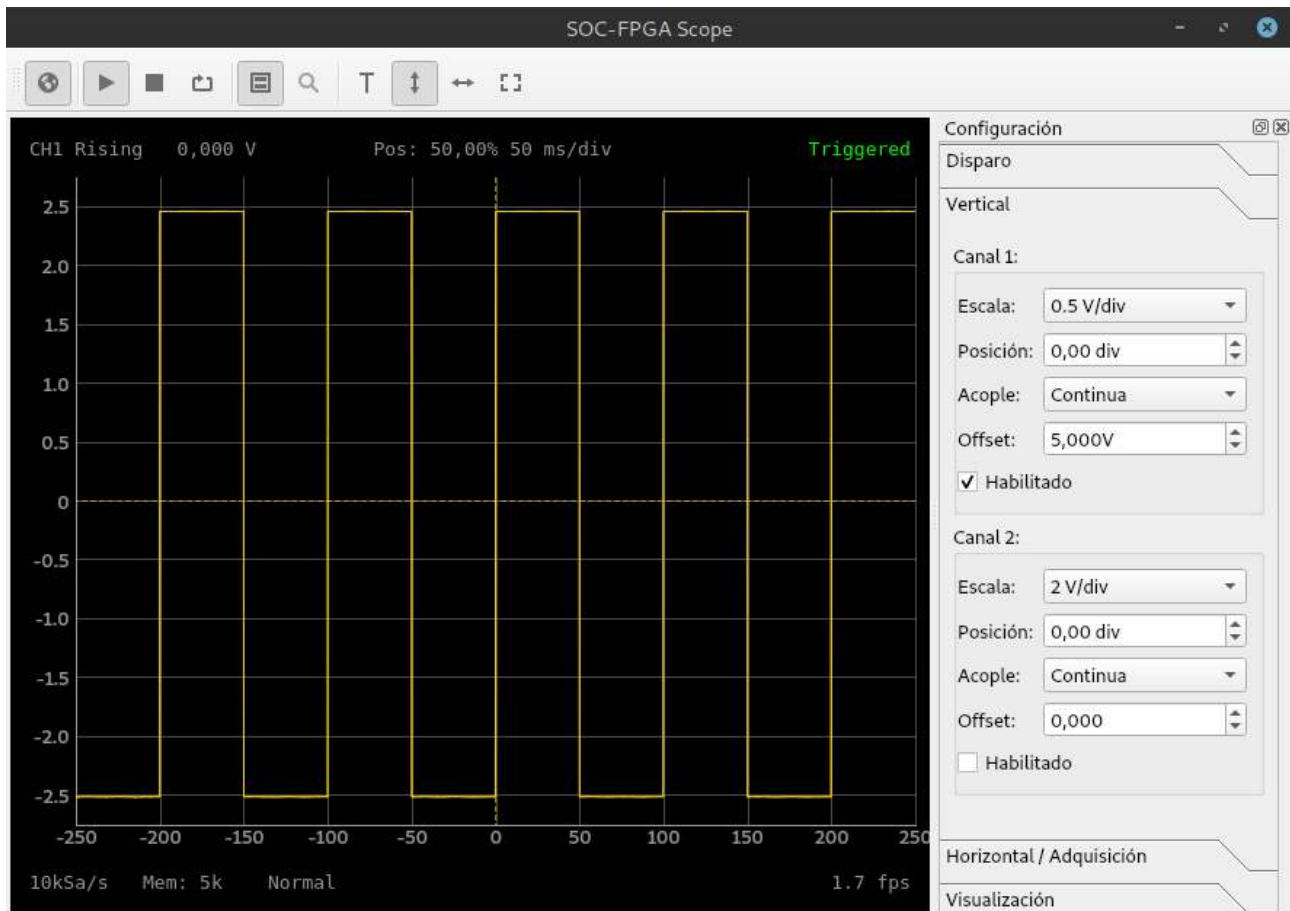


Figura 6.28: Señal cuadrada de 10 Hz y 5 V_{PP} (offset restado)

En la figura 6.28 se puede ver que la señal no se ve afectada por el filtro, por lo que se obtiene un resultado más representativo de la entrada. Este caso es un claro ejemplo de la utilidad de la etapa de offset analógico. Otro posible caso de uso es en señales cuyo valor medio se modifica en el tiempo, como por ejemplo, modulaciones del tipo PWM.

6.3.3 Algoritmos de procesamiento y persistencia

Con el objetivo de evaluar la efectividad de los algoritmos de decimación incorporados en la arquitectura de adquisición, se lleva a cabo una serie de pruebas bajo una entrada de tipo ruido blanco Gaussiano. Por otro lado, la naturaleza no periódica de la señal utilizada, permite evaluar la utilidad y efecto en la visualización de la funcionalidad de persistencia [106]. En este caso se emplea una señal de baja amplitud, cuyo valor medio no es nulo. La configuración para la medición inicial es la siguiente:

- Señal de entrada (canal 1): ruido blanco Gaussiano, amplitud 0,5 V_{RMS}, offset de 1 V.
- Ajuste de disparo: ninguno (disparo continuo).
- Ajuste vertical (canal 1): escala 20 mV/div, posición vertical 0 divisiones, acople de continua, offset 1 V.

- Ajuste horizontal: escala 10 ms/div (100 ms total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 100.000 (1 MSa/s).
- Ajuste de visualización: multi-plot, zoom desactivado, persistencia activada (16 formas de onda).

Para esta prueba, se presenta directamente, en la figura 6.29, la forma de onda para el caso del offset restado por hardware. La escala elegida, de 20 mV/div, y la cantidad de persistencia, de 16 formas de onda, aseguran una buena visualización para este tipo de entrada.

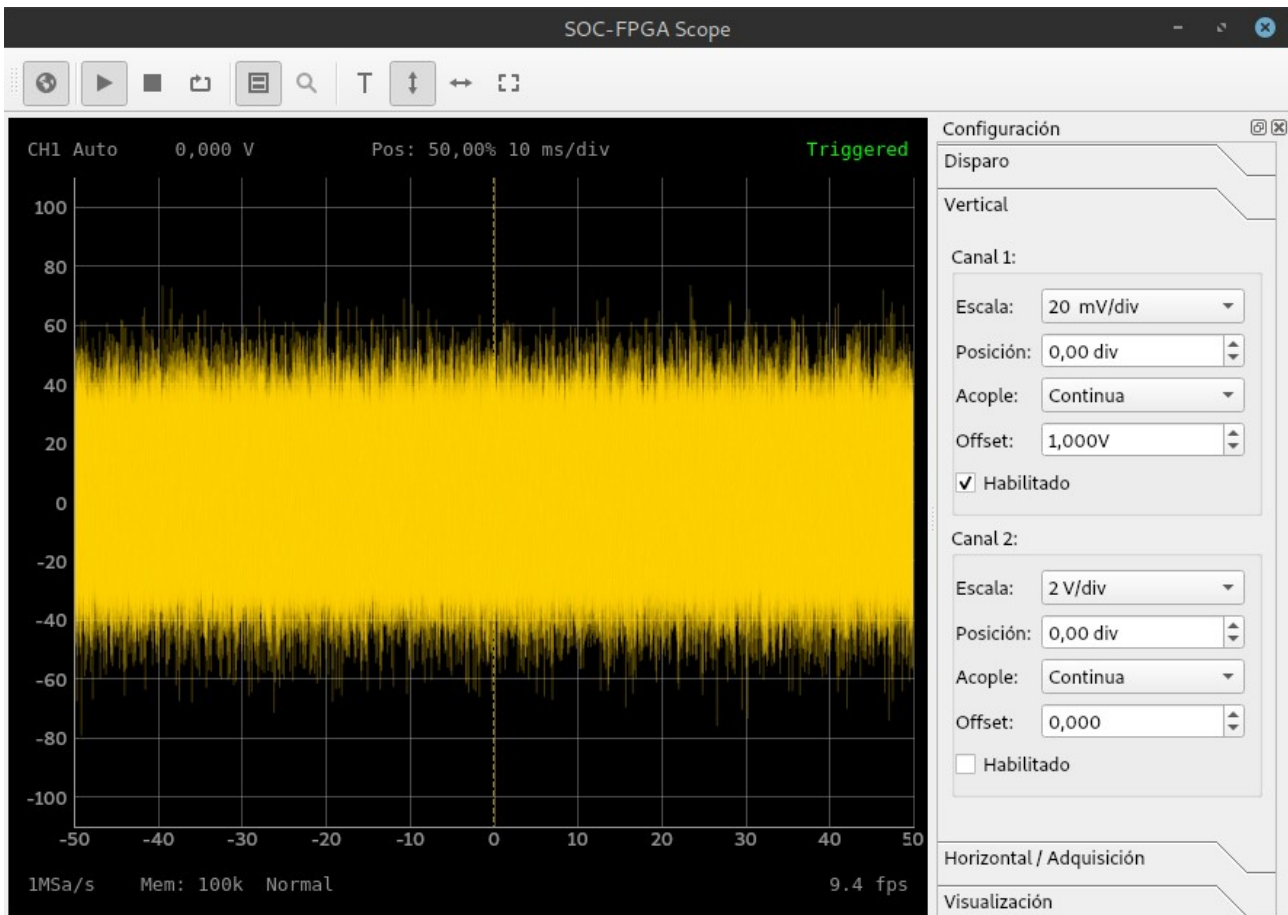


Figura 6.29: Ruido blanco Gaussiano de $0,5 V_{RMS}$ sin decimación

Como se puede observar en la figura 6.29, se utiliza la máxima frecuencia de muestreo posible, de 1 MSa/s, por lo que no ocurre ningún proceso de decimación. Con el objetivo de evaluar los diferentes algoritmos de adquisición, se reduce el tamaño de memoria a 1.000 muestras. Esta situación se corresponde con un factor de decimación de 100 veces y una frecuencia de muestreo equivalente de 10 kSa/s. Bajo estas condiciones se obtiene la siguiente medición:

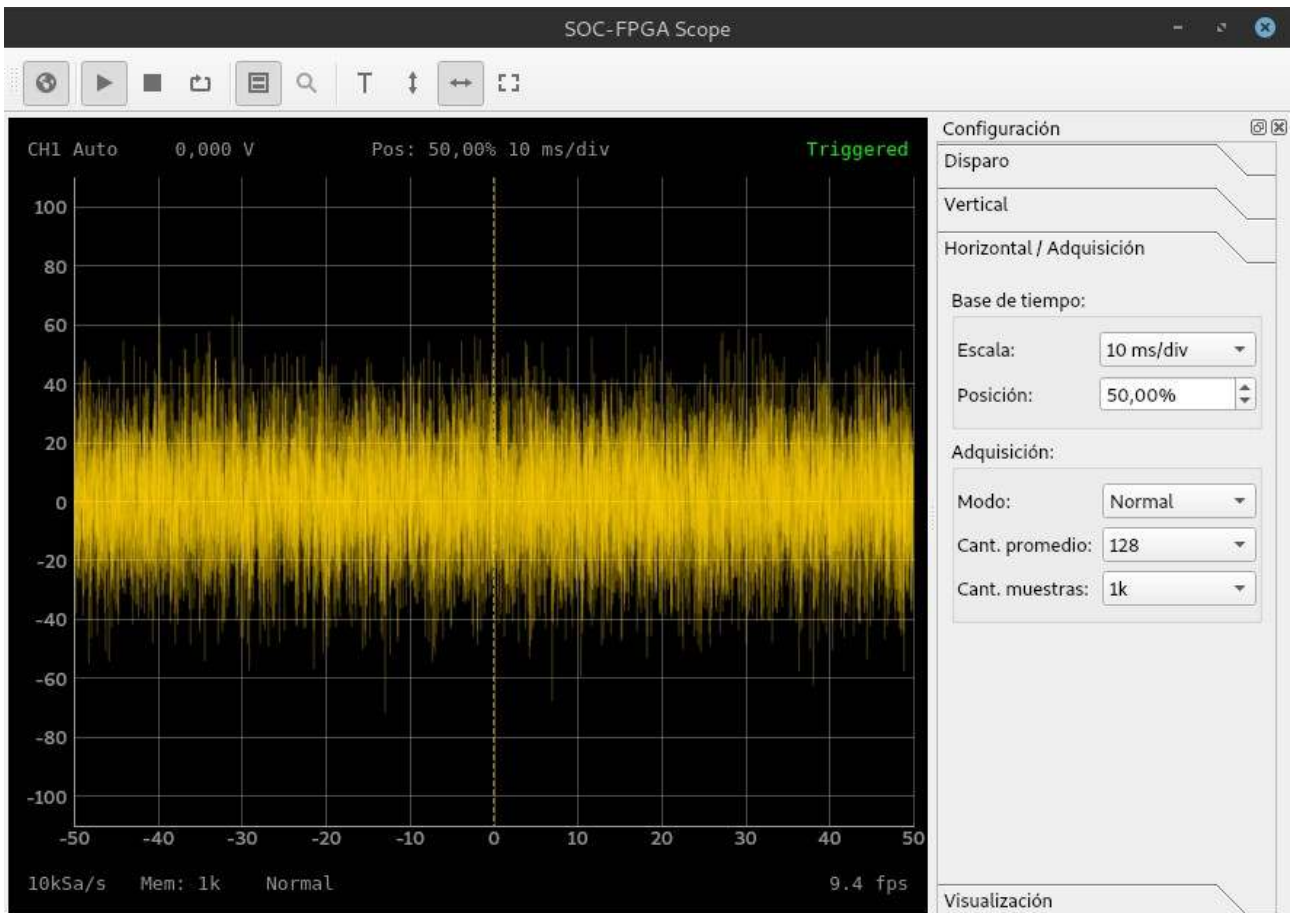


Figura 6.30: Ruido blanco Gaussiano de $0,5 V_{RMS}$ decimación normal ($D=100$)

En la figura 6.30 se denota una reducción considerable en la densidad de los datos graficados, comparado al caso de la figura 6.29. Con respecto a la amplitud de ruido, los valores pico a pico no se modifican considerablemente. A continuación, en la figura 6.31, se presenta la misma medición, realizada bajo el algoritmo de detección de pico:

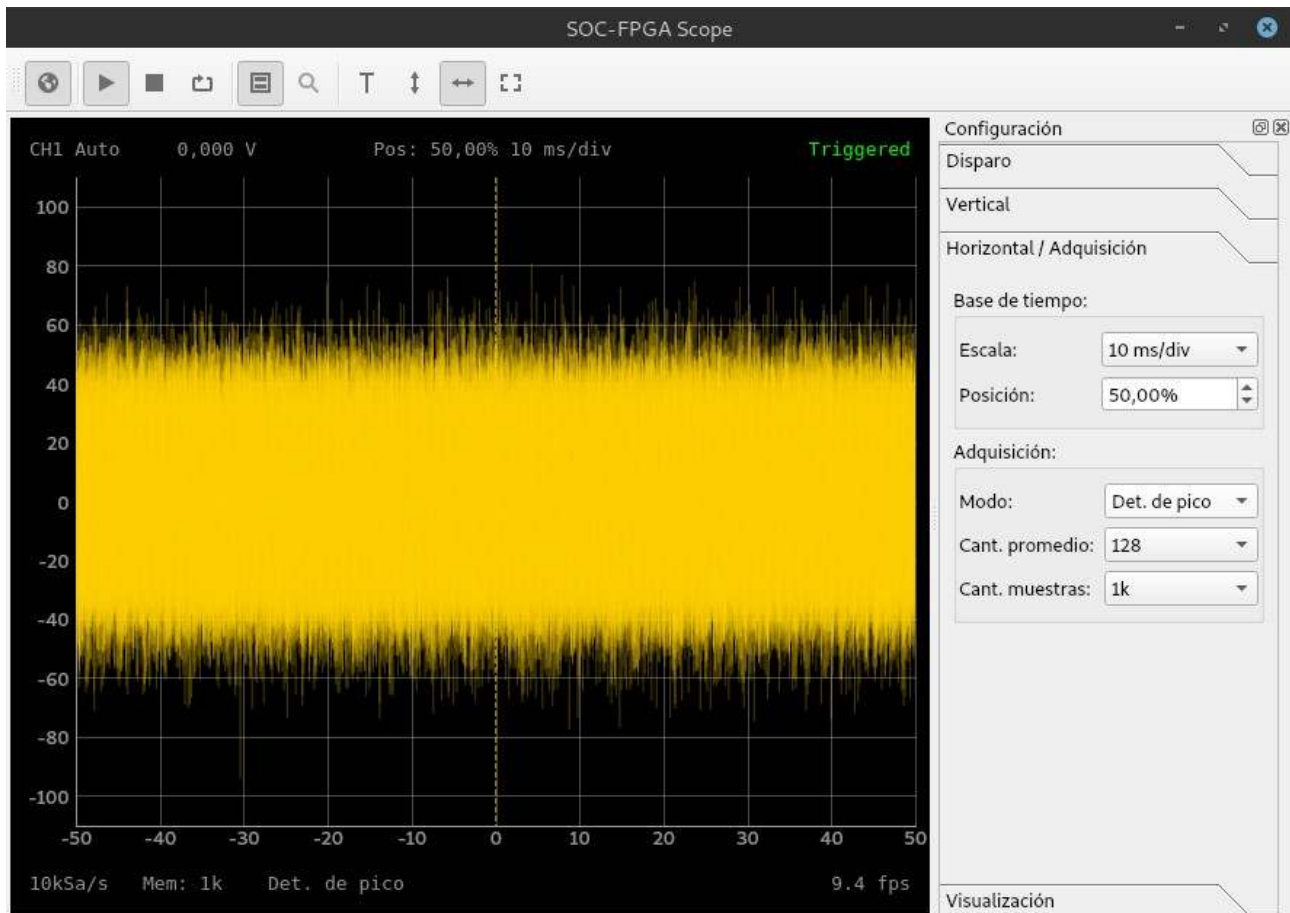


Figura 6.31: Ruido blanco Gaussiano de $0,5 V_{RMS}$ algoritmo de detección de pico ($D=100$)

Como es de esperar, este modo de adquisición resalta los valores máximos y mínimos de cada intervalo de decimación. En este tipo de entrada, de naturaleza aleatoria, el efecto neto es un aumento en la amplitud observada, de aproximadamente $120 mV_{PP}$. Este algoritmo puede ser utilizado para detectar transitorios de pequeña duración a lo largo de un elevado periodo de tiempo.

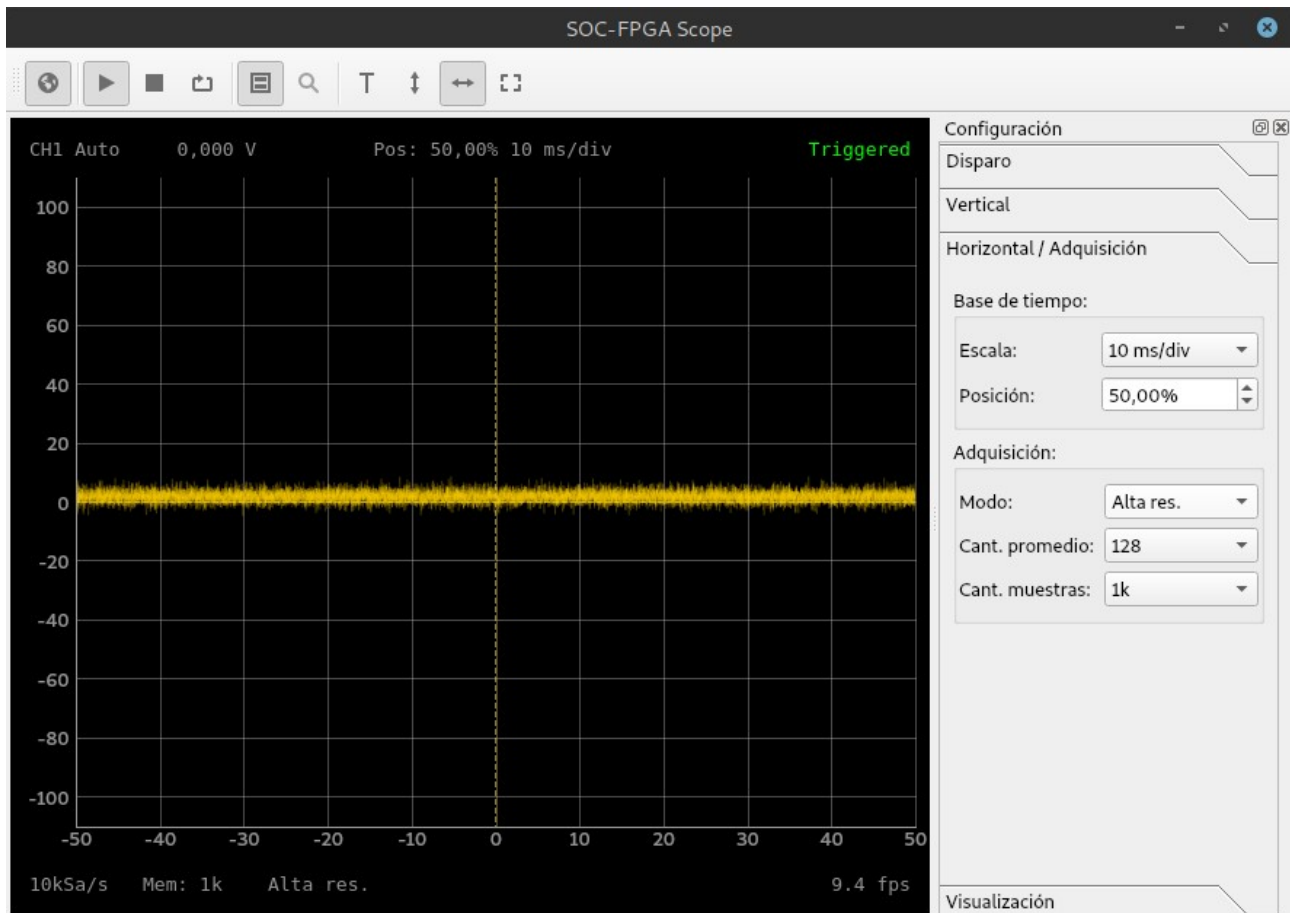


Figura 6.32: Ruido blanco Gaussiano de $0,5 V_{RMS}$ algoritmo de alta resolución ($D=100$)

En la figura 6.32 se efectúa la misma medición, recurriendo en este caso al modo de alta resolución. El efecto del promediado bajo el intervalo de decimación de 100 muestras reduce considerablemente el ruido observado. Este modo de adquisición puede ser utilizado para mediciones de señales inmersas en ruido. Adicionalmente, la aplicación de este algoritmo permite reducir el ruido introducido por el front-end, mejorando la resolución del sistema [76].

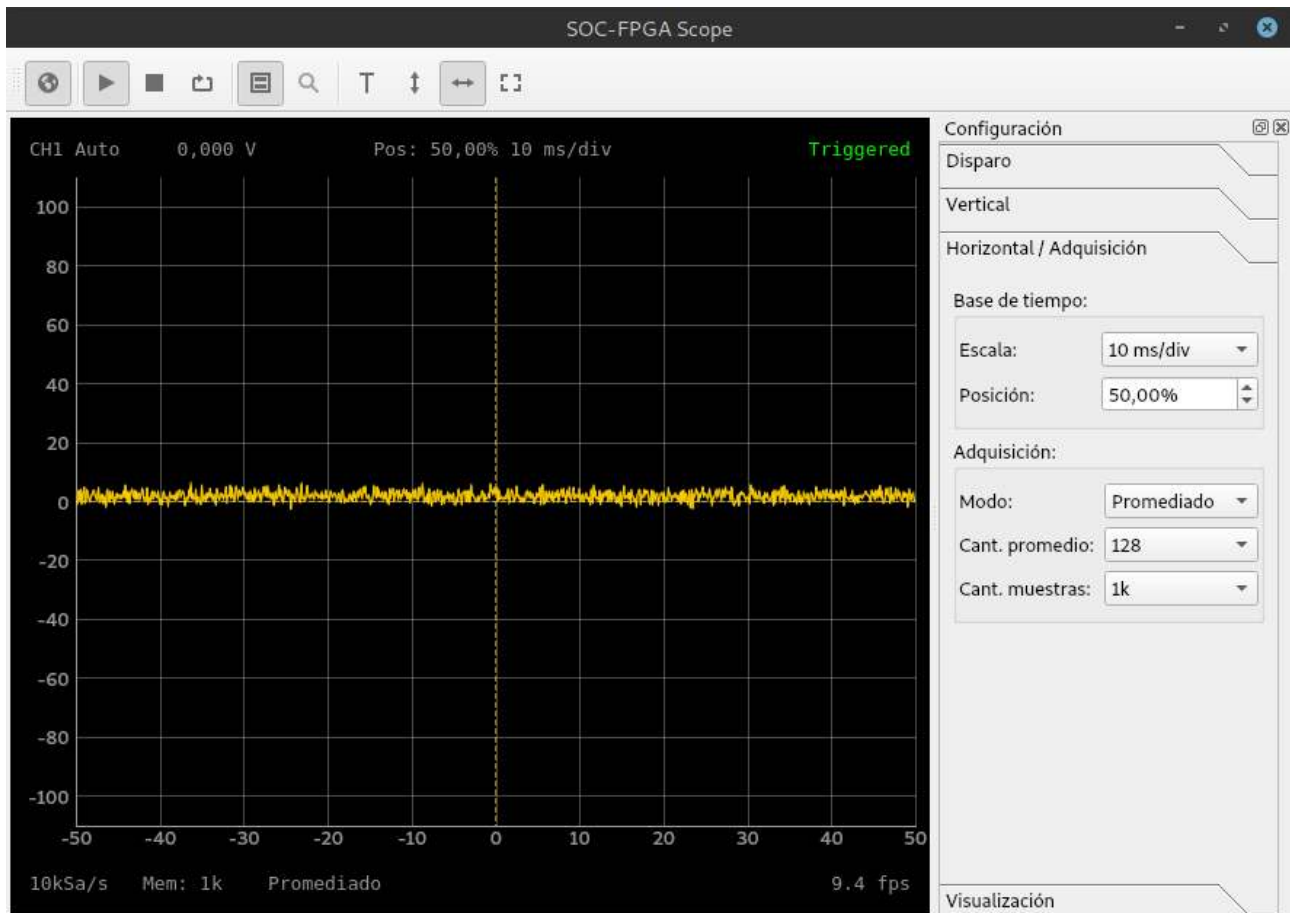


Figura 6.33: Ruido blanco Gaussiano de $0,5 V_{RMS}$ promediado de 128 capturas ($D=100$)

Finalmente, se configura el software para realizar decimación normal y promediar a lo largo de 128 capturas completas. El resultado obtenido en la figura 6.33 es similar al caso de alta resolución, es decir, una disminución considerable de la amplitud del ruido. Sin embargo, el algoritmo de alta resolución tiene la ventaja de poder ser aplicado a señales no periódicas y disparos simples, debido a que opera en una única captura.

Por otro lado, el modo de promediado permite aumentar la frecuencia de muestreo, ya que no requiere de un factor de decimación para operar sobre los datos. Es decir, se puede obtener una reducción del ruido sin necesidad de reducir la frecuencia de adquisición. La incorporación de los dos modos de adquisición mencionados otorga flexibilidad al sistema y permite mejorar el resultado bajo un caso específico de medición.

6.3.4 Profundidad de memoria y modo zoom

El objetivo de esta serie de pruebas es evaluar el comportamiento del sistema ante capturas de gran cantidad de muestras. Cabe recordar que la arquitectura de adquisición utiliza la memoria RAM principal del sistema, lo que permite implementar un buffer de 10 millones de muestras por canal. La elevada profundidad de memoria brinda la posibilidad de hacer capturas durante periodos prolongados de tiempo a la máxima frecuencia de muestreo posible [107].

Para poder analizar las señales en este tipo de capturas, el sistema cuenta con un modo de zoom en la base de tiempo horizontal, el cual permite “recorrer” la forma de onda con un mayor detalle. Para evaluar esta funcionalidad, se recurre a la siguiente configuración de medición:

- Señal de entrada (canal 1): senoidal, frecuencia 5 kHz (periodo 0,2 s), amplitud 1 V_{PP}, offset de 0 V.
- Ajuste de disparo: fuente canal 1, flanco ascendente, nivel 0 V, histéresis 1%.
- Ajuste vertical (canal 1): escala 0,1 V/div, posición vertical 0 divisiones, acople de continua, offset 0 V.
- Ajuste horizontal: escala 1 ms/div (10 ms total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 10.000 (1 MSa/s).
- Ajuste de visualización: single-plot, zoom activado, persistencia desactivada.

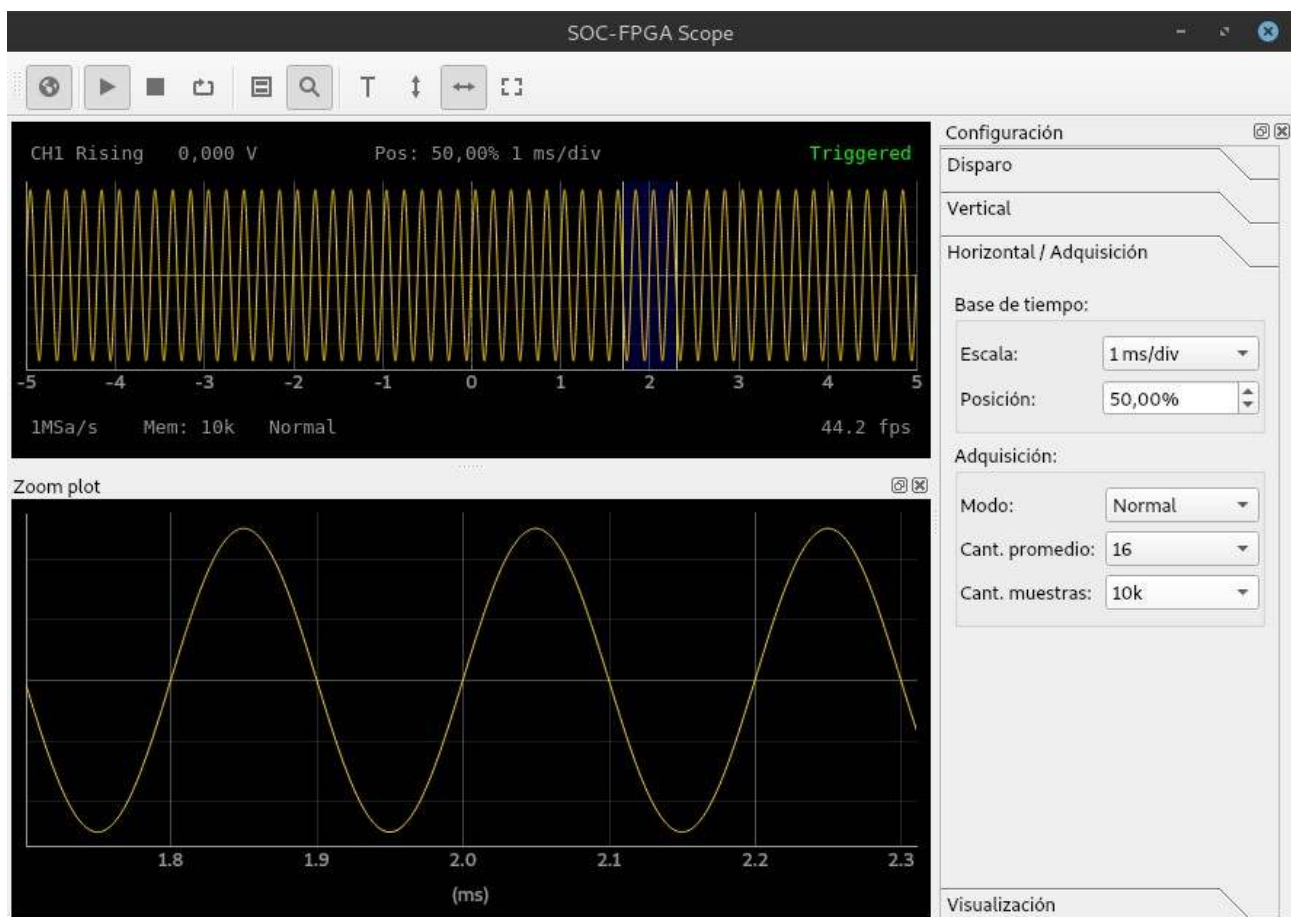


Figura 6.34: Modo zoom aplicado a senoidal de 5 kHz en base de tiempo de 1 ms/div

En la figura 6.34, se hace uso del modo de zoom para observar la señal en el intervalo aproximado de 1,7 ms a 2,3 ms. La implementación realizada de esta funcionalidad, permite seleccionar la región a visualizar en detalle, delimitada en color azul. Cabe destacar que ambos gráficos se encuentran sincronizados y se actualizan en tiempo real.

En la segunda prueba realizada, se eleva el tiempo de captura a un segundo, conservando la frecuencia de muestreo de 1 MSa/s. Esta configuración se corresponde con una captura total de un millón de muestras. El resultado obtenido se observa a continuación:

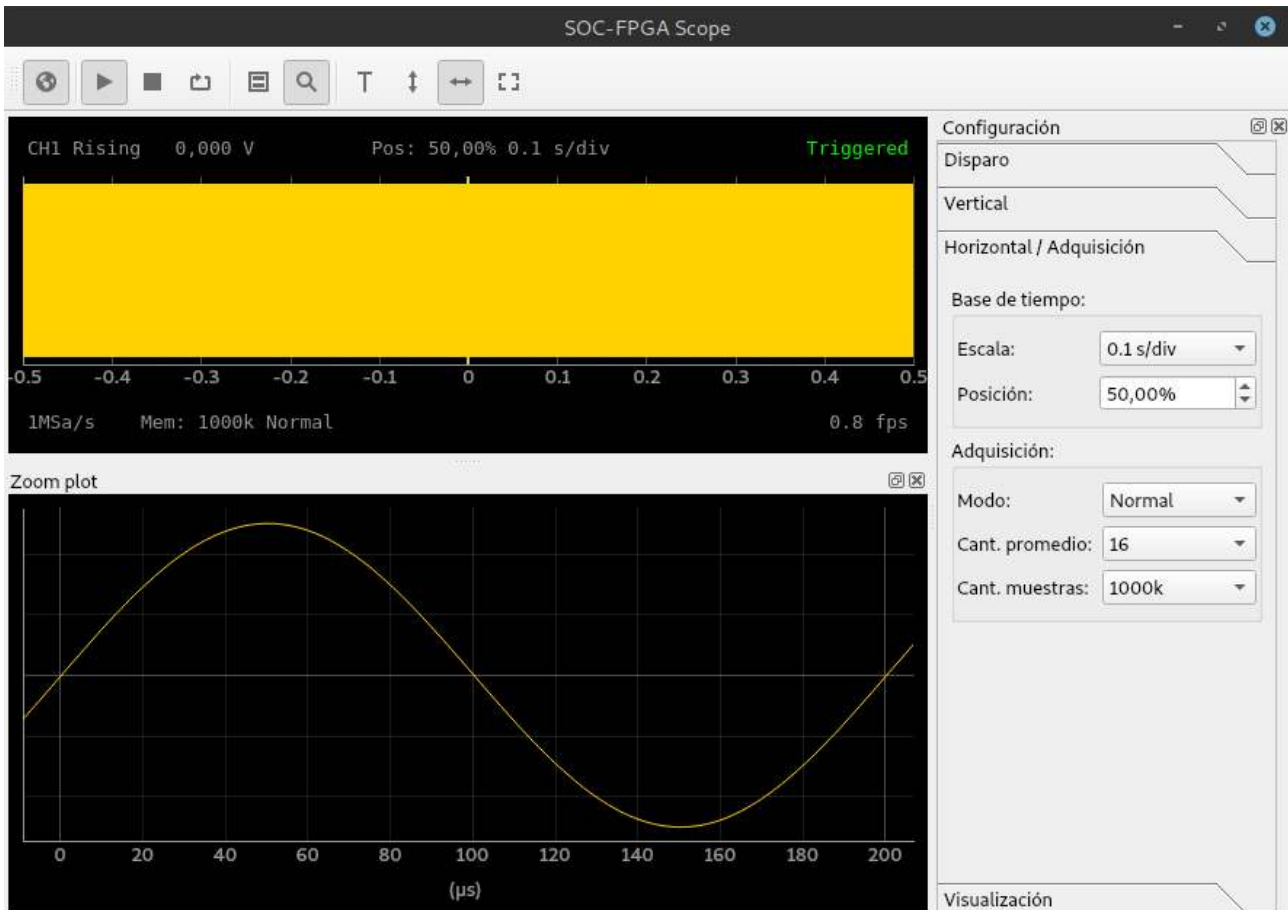


Figura 6.35: Modo zoom aplicado a senoidal de 5 kHz en base de tiempo de 0,1 s/div

En este caso, se emplea el modo zoom para analizar el primer periodo de la señal luego del evento de disparo. Bajo estas condiciones, la captura consiste en 5.000 ciclos de la señal, por lo que no puede ser observada en la base de tiempo completa. Este tipo de medición solo es posible en un sistema de adquisición con una elevada profundidad de memoria.

6.3.5 Modos de visualización de dos canales

Este apartado tiene como objetivo analizar los dos modos de visualización multicanal implementados en el software de cliente. En el modo de funcionamiento single-plot, ambos canales se visualizan en una misma grilla de 10 x 10 divisiones. El ajuste de posición vertical permite desplazar las señales a fin de superponerlas o separarlas. Por otro lado, en el modo multi-plot, ambos canales son visualizados en gráficos independientes, cada uno con su grilla y escala numérica. A fin de demostrar estos aspectos, se utiliza la siguiente configuración:

- Señal de entrada (canal 1): cuadrada, frecuencia 1 kHz (periodo 1 ms), amplitud 1 V_{PP}, offset de 0 V.
- Señal de entrada (canal 2): senoidal, frecuencia 1 kHz (periodo 1 ms), amplitud 3 V_{PP}, offset de 1,65 V.
- Ajuste de disparo: fuente canal 2, flanco descendente, nivel 1,5 V, histéresis 1%.
- Ajuste vertical (canal 1): escala 0,1 V/div, posición vertical 3 divisiones, acople de continua, offset 0 V.
- Ajuste vertical (canal 2): escala 0,5 V/div, posición vertical -4 divisiones, acople de continua, offset 0 V.
- Ajuste horizontal: escala 0,2 ms/div (2 ms total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 2.000 (1 MSa/s).
- Ajuste de visualización: single-plot, zoom desactivado, persistencia desactivada.

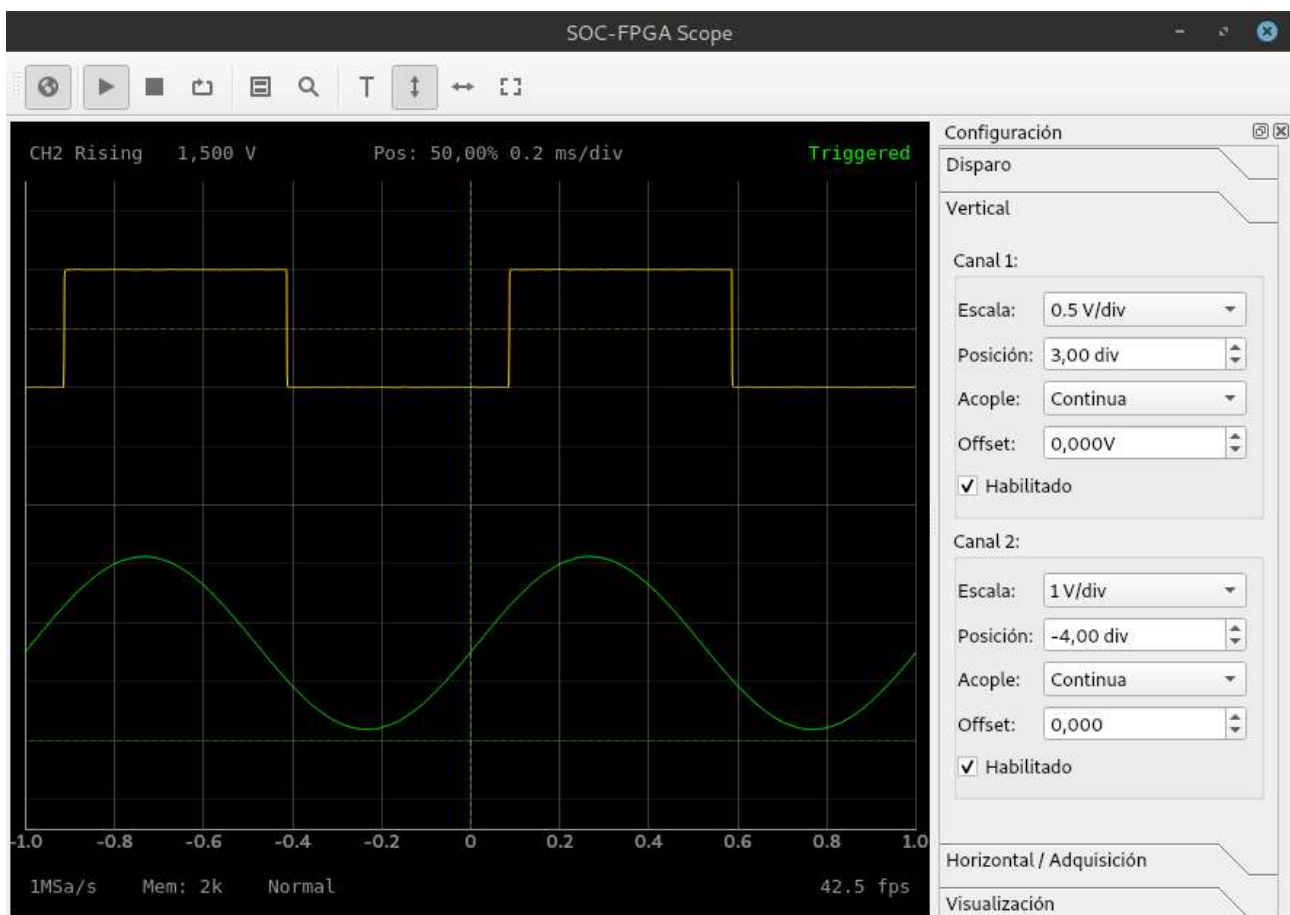


Figura 6.36: Visualización de ambos canales en un mismo gráfico (single-plot)

En la figura 6.36 se observa el modo de funcionamiento single-plot. En este caso, ambos canales son separados verticalmente haciendo uso del ajuste de posición. Es decir, se introduce un desplazamiento de +3 divisiones y -4 divisiones para las formas de onda, respectivamente. Esta

situación puede ser observada en la ubicación de los marcadores de posición de cada canal, así como también en la configuración del panel de ajustes. Resulta pertinente aclarar que este modo de visualización no permite contar con una escala vertical común para ambas formas de onda. Esto se debe al hecho de que el voltaje absoluto depende de la posición y escala seleccionada. Esta situación es diferente en el modo multi-plot, presentado a continuación:

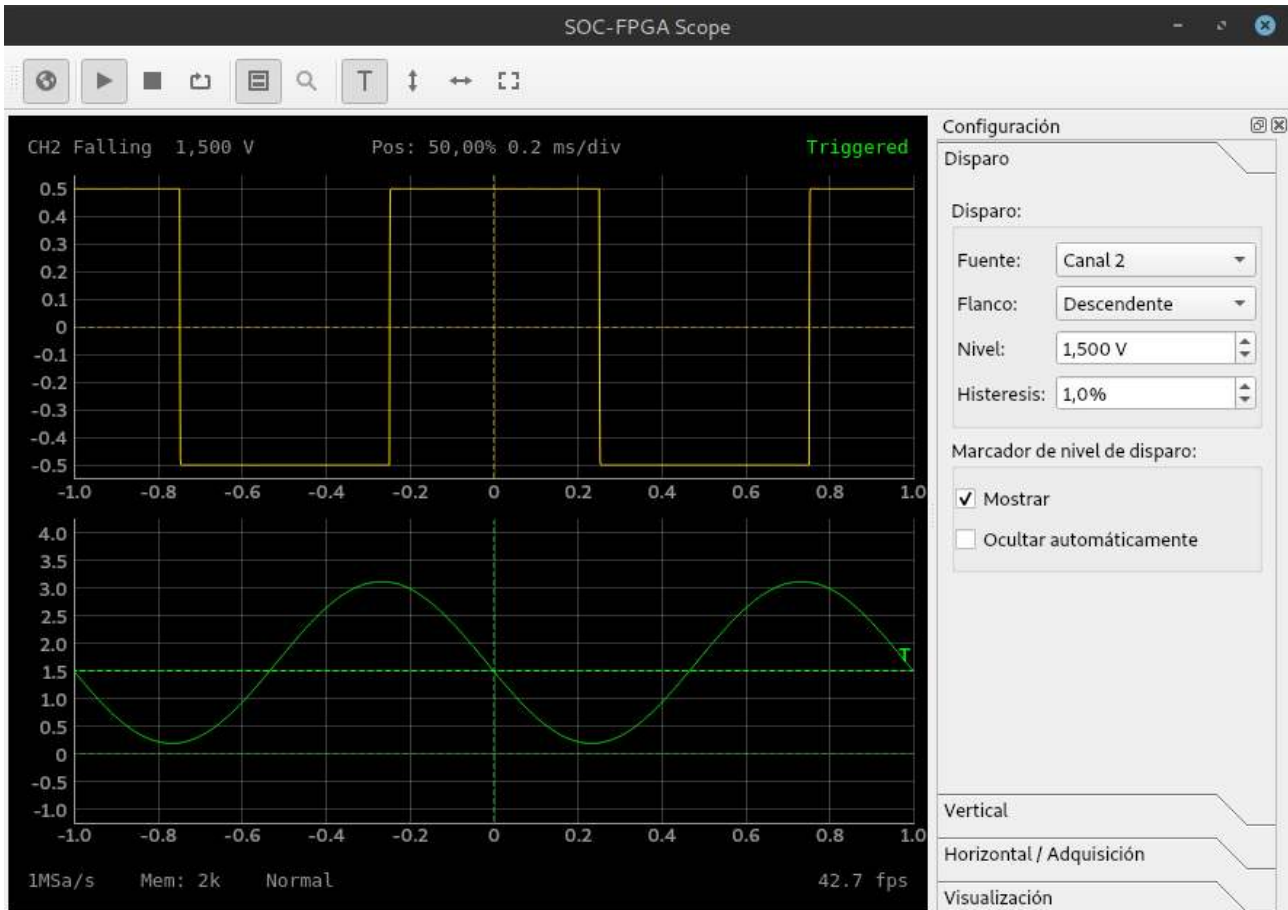


Figura 6.37: Visualización de ambos canales en gráficos separados (multi-plot)

En este caso, cada canal se grafica por separado, lo cual permite contar con una escala numérica vertical independiente. El ajuste de posición desplaza la señal, el marcador de posición, la grilla y la escala numérica. A modo de ejemplo, en la figura 6.37, el desplazamiento para los canales es de 0 divisiones y -3 divisiones, respectivamente. Del análisis realizado, se deduce que el primer caso resulta de utilidad para comparar dos formas de onda adquiridas en simultáneo. Por otro lado, el segundo modo de funcionamiento brinda mayor comodidad e información a la hora de visualizar las señales por separado.

6.3.6 Modo zoom

Por último, en este apartado se analiza la situación en la que se observa un detalle horizontal, mediante el modo de zoom, para el caso en que ambos canales se encuentran activados. Para demostrar esta funcionalidad se utiliza la siguiente configuración

- Señal de entrada (canal 1): triangular, frecuencia 1 kHz (periodo 1 ms), amplitud 3 V_{PP}, offset de 0 V.
- Señal de entrada (canal 2): senoidal, frecuencia 1 kHz (periodo 1 ms), amplitud 3 V_{PP}, offset de 0 V.
- Ajuste de disparo: fuente canal 2, flanco ascendente, nivel 0 V, histéresis 1%.
- Ajuste vertical (canal 1): escala 0,5 V/div, posición vertical 0 divisiones, acople de continua, offset 0 V.
- Ajuste vertical (canal 2): escala 0,5 V/div, posición vertical 0 divisiones, acople de continua, offset 0 V.
- Ajuste horizontal: escala 10 ms/div (100 ms total), posición horizontal 50%, modo de adquisición normal, cantidad de muestras 100.000 (1 MSa/s).
- Ajuste de visualización: single-plot, zoom activado, persistencia desactivada.

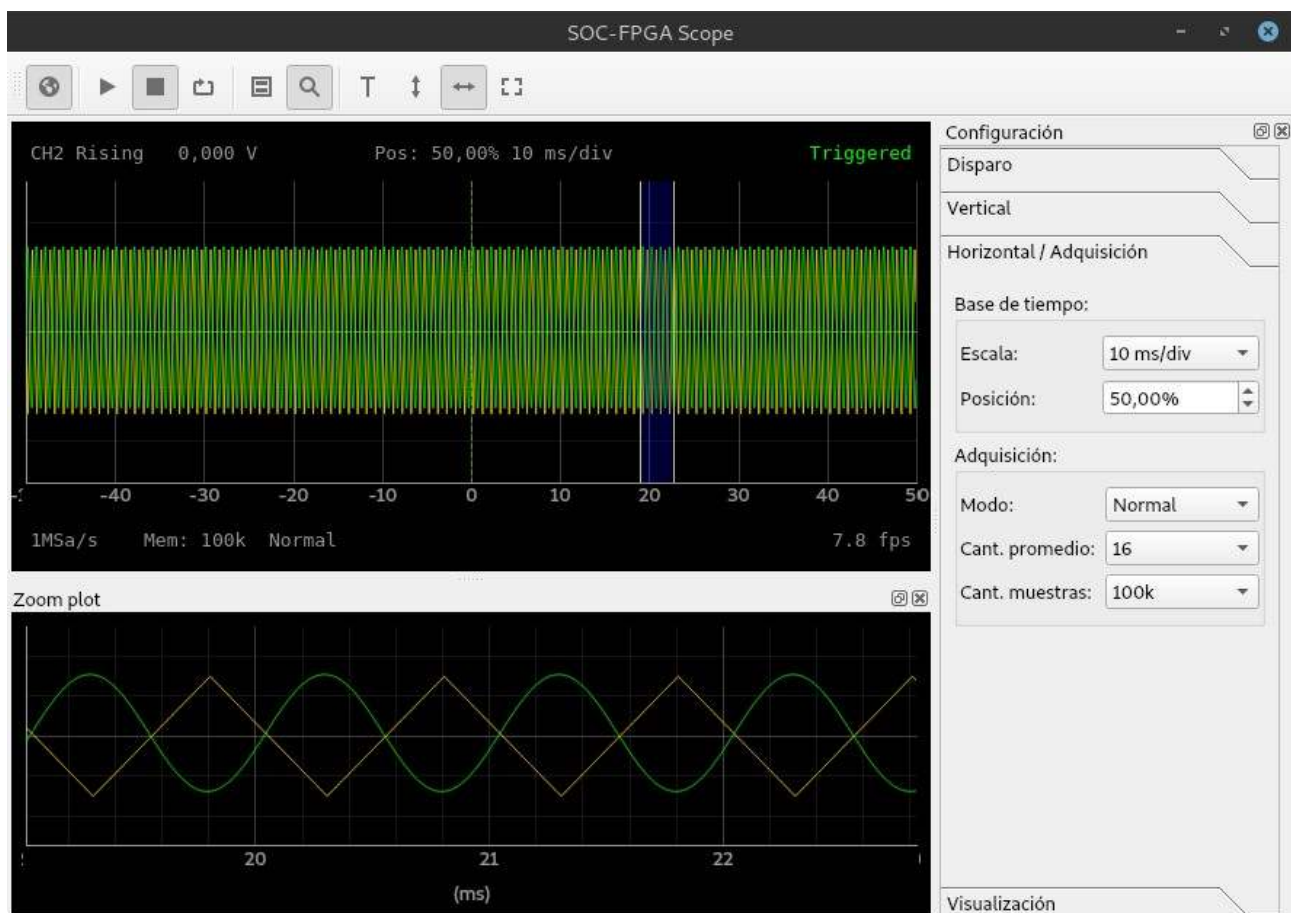


Figura 6.38: Visualización de ambos canales en modo single-plot con zoom activado

Como se puede ver en la figura 6.38, se hace uso de la utilidad de zoom para observar en detalle aproximadamente cuatro ciclos (4 ms) de ambas señales de entrada. Cabe destacar que el modo zoom solo puede ser activado en el caso de que se visualicen ambos canales en un único gráfico, es decir, el modo single-plot.

Capítulo 7

Conclusiones

7.1 Objetivos alcanzados

El desarrollo de este proyecto demostró la factibilidad del diseño de un sistema de adquisición basado en arquitectura SoC FPGA. Se considera que los objetivos y lineamientos generales, enumerados en la sección 1.1, fueron alcanzados de manera exitosa. Es decir, se logró implementar un sistema abierto modular y flexible, capaz de actuar como base para desarrollos futuros que requieran adquisición y procesamiento digital de señales en tiempo real.

En la introducción se llevó a cabo un análisis de mercado y se estableció un precio objetivo de US\$ 250 para el sistema de adquisición, de acuerdo a las alternativas existentes. De acuerdo al apartado 6.1, los costos asociados a la fabricación de un prototipo del dispositivo fueron de US\$ 90 para la placa de desarrollo, US\$ 92,6 para los componentes electrónicos que conforman el front-end y US\$ 9,2 para los PCBs del front-end y la fuente de alimentación. Estos valores dan un total de US\$ 191,8, por lo que a la fecha del proyecto existe un margen de US\$ 58,2 respecto al posible precio de venta. Resulta pertinente aclarar que este análisis consiste en una aproximación, ya que no se tienen en cuenta las tasas de impuestos y cargos de envío, y se emplean precios unitarios para los componentes.

El diseño de hardware y software llevado a cabo para el dispositivo permitió cumplir con los diversos requerimientos funcionales y no funcionales, establecidos durante la definición del producto, en el apartado 1.2. Por otro lado, la implementación de un prototipo funcional del sistema, permitió verificar el diseño teórico y evaluar la performance en condiciones reales. Durante esta fase de pruebas, se detectó que la única especificación que no fue cumplida de forma satisfactoria es el ancho de banda de la etapa de acondicionamiento analógica, ya que se obtuvo un valor mínimo de 420 kHz, siendo el valor especificado de 500 kHz. En el siguiente apartado, dedicado a las mejoras a futuro del sistema, se proponen las soluciones que permiten cumplir de manera exitosa con este requerimiento.

7.2 Líneas de desarrollo futuras

En esta sección se analizan las posibles mejoras que pueden ser implementadas en los diferentes componentes del sistema de adquisición. Los cambios propuestos se agrupan en tres etapas de desarrollo, de acuerdo al grado de complejidad y la profundidad de las mismas. En una primera instancia, se proponen algunas modificaciones sencillas que se asocian al front-end y su fuente de alimentación. Las mismas tienen como objetivo corregir errores de diseño y agregar algunos elementos que fueron omitidos en la primera revisión, es decir, no alteran sustancialmente el diseño y especificaciones de ninguna de las dos placas. Las mismas se enumeran a continuación:

1. Corregir el footprint usado en los conectores BNC de entrada de ambos canales, a fin de poder soldar los pines de montaje. Este error de diseño físico fue analizado en el apartado 6.1.4.
2. Modificar el diseño del atenuador de entrada de 0,9 veces, para ambos canales, a fin de centrar el rango de ajuste de compensación. Este error de diseño, evaluado en la sección 6.2.8, puede ser corregido modificando el valor de uno de los capacitores del atenuador, por lo que no requiere un rediseño del PCB.
3. Seleccionar otro potenciómetro digital con menores valores de capacidad parásita de entrada y salida. De esta forma, se mejora la estabilidad de la etapa anterior y se aumenta el ancho de banda total asociado a cada escala, de acuerdo a lo evaluado en el apartado 6.2.10. Este cambio permite cumplir con el ancho de banda mínimo de 500 kHz, establecido en las especificaciones, en la sección 1.2.3.
4. El ítem anterior implica, de manera indirecta, la corrección del error de diseño físico asociado al uso de un footprint incorrecto para el potenciómetro digital. Esta situación fue analizada en la sección 6.1.4.
5. Incluir agujeros de montaje en las placas del front-end y la fuente de alimentación. De esta forma, se pueden montar los componentes del sistema de adquisición en una carcasa o gabinete, mejorando la robustez del dispositivo.
6. Incorporar en las entradas de alimentación del front-end protección contra sobretensiones y polaridad inversa. La protección puede ser implementada de manera sencilla en base a fusibles de montaje superficial y diodos, aunque existen alternativas de protección más complejas y robustas.

Por otro lado, se propone duplicar la cantidad de canales de entrada del dispositivo, en base al uso de dos PCBs de acondicionamiento. Esta posibilidad se tuvo en cuenta durante la fase de diseño de los diferentes componentes del sistema, a fin de facilitar la modificación en un futuro. A modo de ejemplo, se diseñó una fuente de alimentación independiente, que puede ser compartida por ambas etapas de acondicionamiento. Por otro lado, se utilizó una plataforma de desarrollo con dos conectores de 40 pines, cuyo diagrama de conexiónado es idéntico. Finalmente, se llevó a cabo un diseño de arquitectura de adquisición y software que permite aumentar la cantidad de canales con la mínima cantidad de cambios manuales posibles.

En una segunda etapa, se sugiere aumentar la funcionalidad del sistema en base a una serie de modificaciones en la lógica programable y el software de visualización. Las mismas se enumeran y describen brevemente a continuación:

1. Incorporar en el bloque de interfaz de los ADC, en la lógica programable, algoritmos de procesamiento digital de señales. Cabe recordar que este bloque opera de forma continua en el flujo de datos en tiempo real proveniente de los conversores. De lo mencionado se deduce que resulta de utilidad, por ejemplo, agregar un filtro digital programable. Cabe destacar que las operaciones por sobre las señales de entrada afectan las etapas posteriores, incluso a la lógica de disparo.
2. Modificar el bloque de disparo, en la lógica programable, para incorporar otras modalidades más avanzadas. A modo de ejemplo, en algunos casos resulta de utilidad disparar ante un pulso positivo o negativo cuya duración sea mayor, menor o igual a un

tiempo preestablecido. Por otro lado, se puede incorporar un modo que permita disparar únicamente ante un pulso con determinado tiempo de crecimiento o decrecimiento.

3. Modificar el bloque de disparo, en la lógica programable, para incluir filtros digitales que permitan eliminar o atenuar ciertos rangos de frecuencia. Por ejemplo, se puede incorporar un filtro pasa-bajos, destinado a rechazar las componentes de alta frecuencia y ruido. Por otro lado, un filtro pasa-altos permite eliminar el valor medio y variaciones lentas en la señal de disparo.
4. Agregar algoritmos que permitan medir de forma automática diversos parámetros por sobre las señales de entrada. A modo de ejemplo, resulta de utilidad conocer el valor máximo, mínimo y RMS de la señal, así como también su frecuencia, periodo y ciclo de trabajo, de corresponderse. Esta funcionalidad puede ser implementada fácilmente en el cliente de visualización, en base a la librería de cálculo científico NumPy. Sin embargo, se propone implementar los algoritmos en el bloque de procesamiento en la FPGA, disminuyendo los requerimientos por sobre la PC cliente y aumentando la fluidez del sistema.
5. Permitir al usuario visualizar la forma de onda que surge como resultado de realizar operaciones matemáticas entre los canales. A modo de ejemplo, se puede incorporar un nuevo gráfico que represente la suma, resta, multiplicación o división de dos canales. Por otro lado, resultaría interesante brindarle al usuario la posibilidad de introducir una ecuación matemática arbitraria. Esta funcionalidad se beneficiaría de una aceleración de los algoritmos mediante hardware, aprovechando la disponibilidad de lógica programable. Sin embargo, en este último caso sería difícil incorporar la posibilidad de graficar una ecuación arbitraria, por lo que no se descarta una implementación en el software cliente.
6. Incorporar funcionalidades de análisis frecuencial de la señal de entrada de cada canal. Típicamente, en sistemas de procesamiento digital, se suele aplicar el algoritmo de transformada rápida de Fourier o FFT (por sus siglas en inglés). Al igual que los casos anteriores, una implementación en la lógica programable brindaría grandes beneficios de rendimiento en el sistema. Existen en el mercado bloques de diseño lógico para FPGAs destinados a ejecutar la FFT, incluso provistas por el fabricante del dispositivo utilizado. Sin embargo, para su incorporación, se deben realizar modificaciones en la arquitectura de adquisición, ya que los bloques FFT suelen operar sobre tamaños fijos de buffer, típicamente en potencias de dos. Adicionalmente, se debe modificar el diseño de software para realizar un gráfico frecuencial y permitir ajustar los diversos parámetros asociados a la transformada.
7. Agregar al cliente un menú o acción que permita iniciar el proceso de calibración automática. De este modo, se elimina la necesidad de tener que ejecutar manualmente el algoritmo de calibración en la placa de desarrollo. De forma adicional, el software puede informar al usuario los parámetros de calibración y errores de offset y ganancia obtenidos.
8. Brindar en el cliente la posibilidad de almacenar de forma permanente los datos correspondientes a un ciclo de captura completo. Es decir, se propone exportar los datos, a fin de poder analizarlos y procesarlos en un software externo, como por ejemplo MATLAB. Adicionalmente, resulta de utilidad poder realizar una captura de pantalla del área de visualización y guardarla en algún formato de imagen apropiado.

9. Incorporar en el software cliente marcadores y cursores que permitan realizar mediciones por sobre las formas de onda asociadas a ambos canales. Se propone incorporar dos cursores verticales, a fin de poder medir diferencias de voltaje, y de forma análoga, dos cursores horizontales que permitan hacer mediciones de tiempo y frecuencia.
10. Implementar en el software cliente una funcionalidad de configuración automática (autoset). La misma consiste de un algoritmo que obtiene los parámetros de visualización óptimos de acuerdo a las señales de entrada. A modo de ejemplo, se pueden ajustar las escalas verticales y horizontales, así como también el nivel y flanco de disparo. Por otro lado, resulta de utilidad incorporar un botón o menú que permita restablecer todos los ajustes del sistema a los valores por defecto (preset).
11. Agregar un modo de visualización XY, es decir, graficar las variaciones de un canal en el eje horizontal y las variaciones del otro canal en el eje vertical. De esta forma, se obtiene una curva paramétrica que permite analizar la relación entre ambas señales.

Por último, en una tercera etapa, se propone aumentar la velocidad de muestreo del dispositivo, permitiendo su uso para la medición y análisis de señales de mayor frecuencia. Se considera que este parámetro es un factor altamente limitante en el diseño actual del sistema de adquisición. Sin embargo, un aumento del mismo requiere profundas modificaciones, sobre todo en el diseño del front-end.

Para lograr un aumento de la frecuencia de operación, se deben emplear conversores analógico digital con una interfaz de datos de tipo paralelo, o en su defecto, serie diferencial. Por otro lado, con el objetivo de conservar la integridad de las señales en la comunicación entre el front-end y la placa de adquisición, resulta necesario eliminar el cable cinta de 40 pines. Una alternativa aceptable sería diseñar un PCB que se monte directamente en las cabeceras de 40 pines de la placa de desarrollo.

Adicionalmente, se debe rediseñar completamente la etapa analógica, empleando componentes con un mayor ancho de banda, tales como amplificadores de ganancia variable controlados por tensión (VGAs). De esta forma, se pueden eliminar los componentes que actúan como limitantes en el ancho de banda, como por ejemplo, el potenciómetro digital y las etapas de amplificación discretas.

Bibliografía

- [1] Institute of Electrical and Electronics Engineers, *IEEE recommended practice for software requirements specifications*. New York: IEEE, 1998.
- [2] National Instruments, “NI Product Guide,” *Document 350034W*, Jul. 2013. http://www.ni.com/pdf/productguide/350034W_product_guide.pdf (accessed Sep. 03, 2019).
- [3] National Instruments, “NI cRIO-9063 Specifications,” *Document 376330A-02*, Oct. 2015. http://www.ni.com/pdf/productguide/350034W_product_guide.pdf (accessed Sep. 03, 2019).
- [4] National Instruments, “NI 9215 Datasheet,” *Document 373779A-02*, Mar. 2016. https://www.ni.com/pdf/manuals/373779a_02.pdf (accessed Sep. 03, 2019).
- [5] Digilent, “OpenScope MZ Reference Manual,” *Document 500-324*, May 2017. <https://reference.digilentinc.com/reference/instrumentation/openscope-mz/reference-manual> (accessed Sep. 03, 2019).
- [6] Red Pitaya, “STEMlab Documentation,” *Document Release 0.97*, Sep. 2019. <https://buildmedia.readthedocs.org/media/pdf/redpitaya/latest/redpitaya.pdf> (accessed Sep. 03, 2019).
- [7] Intel, “Intel FPGA Product Catalog,” *Document SG-PRDCT-19.3*, Nov. 2019. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/sg/product-catalog.pdf> (accessed Dec. 10, 2019).
- [8] Intel, “Intel User-Customizable SoC FPGAs,” *Document Broch 1018-1.0*, Nov. 2019. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/br/br-soc-fpga.pdf> (accessed Dec. 10, 2019).
- [9] Intel, “Cyclone V Device Datasheet,” *Document CV-51002*, Nov. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51002.pdf (accessed Dec. 10, 2019).
- [10] Terasic, “DE0-Nano-SoC User Manual,” *Document v1.5*, Dec. 2015. https://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual.pdf (accessed Dec. 10, 2019).
- [11] Intel, “Cyclone V Device Handbook - Volume 1: Device Interfaces and Integration,” *Document CV-5V2*, Oct. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_5v2.pdf (accessed Dec. 10, 2019).
- [12] Intel, “Cyclone V Device Overview,” *Document CV-51001*, May 2018. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_51001.pdf (accessed Dec. 10, 2019).
- [13] Intel, “Cyclone V Hard Processor System Technical Reference Manual,” *Document CV-51001*, Jun. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-v/cv_5v4.pdf (accessed Dec. 10, 2019).
- [14] S. Furber, *ARM system-on-chip architecture*, 2. ed. Harlow: Addison-Wesley, 2000.

- [15] Intel, "Cyclone V and Arria V SoC Device Design Guidelines," *Document AN-796*, Jul. 2019. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an-cv-av-soc-ddg.pdf> (accessed Dec. 10, 2019).
- [16] Texas Instruments, "CDCE937 Flexible Low Power LVCMOS Clock Generator With SSC Support For EMI Reduction," *Datasheet SLAS564G*, Oct. 2016. <http://www.ti.com/lit/ds/symlink/cdce937.pdf> (accessed Dec. 10, 2019).
- [17] Cypress, "CY7C68013A/CY7C68014A EZ-USB FX2LP USB Microcontroller High-Speed USB Peripheral Controller," *Datasheet 38-08032 Rev. AC*, Jul. 2019. <https://www.cypress.com/file/138911/download> (accessed Dec. 10, 2019).
- [18] Analog Devices, "LTC2308 Low Noise, 500ksps, 8-Channel, 12-Bit ADC," *Datasheet 2308fc*, Oct. 2016. <https://www.analog.com/media/en/technical-documentation/data-sheets/2308fc.pdf> (accessed Dec. 10, 2019).
- [19] ISSI, "IS43TR16256A 512Mx8, 256Mx16 4Gb DDR3 SDRAM," *Datasheet IS43TR16256A Rev. I1*, Feb. 2018. <http://www.issi.com/WW/pdf/43-46TR16256A-85120AL.pdf> (accessed Dec. 10, 2019).
- [20] Microchip, "KSZ9031RNX Gigabit Ethernet Transceiverwith RGMII Support," *Datasheet DS00002117F*, Jul. 2016. <http://ww1.microchip.com/downloads/en/devicedoc/00002117f.pdf> (accessed Dec. 10, 2019).
- [21] Microchip, "USB3300 Hi-Speed USB Host, Device or OTG PHY with ULPI Low Pin Interface," *Datasheet DS00001783C*, Jun. 2016. <http://ww1.microchip.com/downloads/en/DeviceDoc/00001783C.pdf> (accessed Dec. 10, 2019).
- [22] FTDI, "FT232R USB UART IC," *Datasheet FT_000053 v2.5*, Apr. 2019. https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf (accessed Dec. 10, 2019).
- [23] Analog Devices, "ADXL345 3-Axis, Digital Accelerometer," *Datasheet D07925-0-6/15(E)*, Jun. 2015. https://www.analog.com/media/en/technical-documentation/data-sheets/AD7476A_7477A_7478A.pdf (accessed Dec. 10, 2019).
- [24] Texas Instruments, "OPAx197 36-V, Precision, Rail-to-Rail Input/Output, Low Offset Voltage, Operational Amplifiers," *Datasheet SBOS737C*, Mar. 2018. <http://www.ti.com/lit/ds/symlink/opa2197.pdf> (accessed Dec. 10, 2019).
- [25] IXYS, "CPC1017N Single-Pole, Normally Open 4-Lead SOP OptoMOS Relay," *Datasheet DS-CPC1017N-R08*, Jun. 2013. <https://ar.mouser.com/datasheet/2/240/CPC1017N-1546456.pdf> (accessed Dec. 10, 2019).
- [26] ROHM Semiconductor, "BAV199HMFH Switching Diode (Low leakage)," *Datasheet BAV199HMFH Rev. 005*, Nov. 2018. <https://ar.mouser.com/datasheet/2/348/bav199hmfht116-e-1507681.pdf> (accessed Dec. 10, 2019).
- [27] KEMET, "EC2/EE2 Series Miniature Signal Relays," *Datasheet R7002_EC2_EE2*, Jul. 2017. https://ar.mouser.com/datasheet/2/212/KEM_R7002_EC2_EE2-1104574.pdf (accessed Dec. 10, 2019).
- [28] ON Semiconductor, "NTR3C21NZ Power MOSFET," *Datasheet NTR3C21NZ/D Rev. 2*, Sep. 2018. <https://ar.mouser.com/datasheet/2/308/NTR3C21NZ-D-1022933.pdf> (accessed Dec. 10, 2019).

-
- [29] Diodes Incorporated, "BAV16W/1N4148W SURFACE MOUNT FAST SWITCHING DIODE," *Datasheet DS30086 Rev. 28-2*, Jan. 2017. <https://ar.mouser.com/datasheet/2/115/ds30086-60872.pdf> (accessed Dec. 10, 2019).
- [30] Analog Devices, "Op Amp Input Overvoltage Protection," *Analog Dialogue 50-05*, May 2016. <https://www.analog.com/media/en/analog-dialogue/volume-50/number-2/articles/op-amp-input-overvoltage-protection.pdf> (accessed Dec. 10, 2019).
- [31] Texas Instruments, "OPA197 Simulation Model," *OPA197 PSpice Model Rev. B*, Feb. 2019. <http://www.ti.com/lit/zip/sboma34> (accessed Dec. 10, 2019).
- [32] Maxim Integrated, "DS1267B Dual Digital Potentiometer," *Datasheet 19-6589 Rev. 1*, Jan. 2014. <https://ar.mouser.com/datasheet/2/256/DS1267B-239545.pdf> (accessed Dec. 10, 2019).
- [33] Analog Devices, "Digital Potentiometers," *Document MT-091*, Jan. 2009. <https://www.analog.com/media/en/training-seminars/tutorials/MT-091.pdf> (accessed Dec. 10, 2019).
- [34] Analog Devices, "LTC6910-1/LTC6910-2/LTC6910-3 Digitally Controlled Programmable Gain Amplifiers in SOT-23," *Datasheet 6910123fb*, Jun. 2016. <https://www.analog.com/media/en/technical-documentation/data-sheets/6910fb.pdf> (accessed Dec. 10, 2019).
- [35] Analog Devices, "Protecting ADC Inputs," *Document TA15638-0-4/17(A)*, Apr. 2017. <https://www.analog.com/media/en/technical-documentation/tech-articles/Final-Protecting-ADC-Inputs.pdf> (accessed Dec. 10, 2019).
- [36] Texas Instruments, "ADCS747x 1-MSPS, 12-Bit, 10-Bit, and 8-Bit A/D Converters," *Datasheet SNAS192G*, May 2016. <http://www.ti.com/lit/ds/symlink/adcs7476.pdf> (accessed Dec. 10, 2019).
- [37] Microchip, "MCP4921/4922 12-Bit DAC with SPI Interface," *Datasheet DS21897A*, May 2004. <https://ar.mouser.com/datasheet/2/268/21897a-70809.pdf> (accessed Dec. 10, 2019).
- [38] Lite-On, "LTST-C150GKT SMD LED," *Datasheet DS-22-98-0004 Rev. A*, Feb. 2005. <https://ar.mouser.com/datasheet/2/239/Lite-On-LTST-C150GKT-1175255.pdf> (accessed Dec. 10, 2019).
- [39] H. W. Johnson and M. Graham, "Ribbon Cables," in *High-speed digital design: a handbook of black magic*, Englewood Cliffs, N.J.: Prentice Hall, 1993.
- [40] Texas Instruments, "SN74LVC8T2458-Bit Dual-Supply Bus Transceiver With Configurable Voltage Translation and 3-State Outputs," *Datasheet SCES584B*, Nov. 2014. <http://www.ti.com/lit/ds/symlink/sn74lvc8t245.pdf> (accessed Dec. 10, 2019).
- [41] Texas Instruments, "XB0104 4-Bit Bidirectional Voltage-level Translator With Automatic Direction Sensing and ± 15 -kV ESD Protection," *Datasheet SCES650I*, Mar. 2018. <http://www.ti.com/lit/ds/symlink/txb0104.pdf> (accessed Dec. 10, 2019).
- [42] Analog Devices, "Voltage Reference Design for Precision Successive-Approximation ADCs," *Analog Dialogue 47-06*, Nov. 2019. <https://www.analog.com/media/en/analog-dialogue/volume-47/number-2/articles/precision-successive-approximation-adcs.pdf> (accessed Dec. 10, 2019).
- [43] Texas Instruments, "SN74LVC1G17-Q1 Single Schmitt-Trigger Buffer," *Datasheet SCES663B*, Oct. 2018. <http://www.ti.com/lit/ds/symlink/sn74lvc1g17-q1.pdf> (accessed Dec. 10, 2019).

- [44] H. W. Johnson and M. Graham, "Terminations," in *High-speed digital design: a handbook of black magic*, Englewood Cliffs, N.J: Prentice Hall, 1993.
- [45] ROHM Semiconductor, "BAT54SHMFH Schottky Barrier Diode," *Datasheet BAT54SHMFH*, Aug. 2016. <https://ar.mouser.com/datasheet/2/348/bat54shmfh-e-1267047.pdf> (accessed Dec. 10, 2019).
- [46] Texas Instruments, " μ A78Mxx Positive-Voltage Regulators," *Datasheet SLVS059T*, Jan. 2015. <http://www.ti.com/lit/ds/symlink/ua78m.pdf> (accessed Dec. 10, 2019).
- [47] ON Semiconductor, "MC79M00 Series 500 mA Negative Voltage Regulators," *Datasheet MC79M00/D Rev. 15*, Jul. 2019. <https://ar.mouser.com/datasheet/2/256/MAX6035-1515756.pdf> (accessed Dec. 10, 2019).
- [48] Diodes Incorporated, "BZT52C2V0-BZT52C51 SURFACE MOUNT ZENER DIODE," *Datasheet DS18004 Rev. 38-2*, Oct. 2017. <https://ar.mouser.com/datasheet/2/115/ds18004-1142534.pdf> (accessed Dec. 10, 2019).
- [49] W. Kester, "Data Converter Support Circuits," in *Data conversion handbook*, Amsterdam; Boston: Elsevier; Newnes, 2005.
- [50] Maxim Integrated, "MAX6035 High-Supply-Voltage, Precision Voltage Reference in SOT23," *Datasheet 19-2606; Rev 3*, Nov. 2016. <https://ar.mouser.com/datasheet/2/256/MAX6035-1515756.pdf> (accessed Dec. 10, 2019).
- [51] Texas Instruments, "REF30xx 50-ppm/ $^{\circ}$ C Max, 50- μ A, CMOS Voltage Reference in SOT-23-3," *Datasheet SBVS032H*, Feb. 2018. <http://www.ti.com/lit/ds/symlink/ref3033.pdf> (accessed Dec. 10, 2019).
- [52] Texas Instruments, "Introduction to Power Supplies," *Document SNVA006B*, May 2004. <http://www.ti.com/lit/an/snva006b/snva006b.pdf> (accessed Dec. 10, 2019).
- [53] Digilent, "PmodAD1," *Schematic 500-064*, Jan. 2011. https://reference.digilentinc.com/_media/reference/pmod/pmodad1/pmodad1_sch.pdf (accessed Dec. 10, 2019).
- [54] H. W. Johnson and M. Graham, "Ground Planes and Layer Stacking," in *High-speed digital design: a handbook of black magic*, Englewood Cliffs, N.J: Prentice Hall, 1993.
- [55] J. G. Proakis and D. G. Manolakis, "Multirate Digital Signal Processing," in *Digital Signal Processing*, New Delhi: Prentice-Hall of India, 2007.
- [56] Analog Devices, "AD7476A/AD7477A/AD7478A 2.35 V to 5.25 V, 1 MSPS, 12-/10-/8-Bit ADCs in 6-Lead SC70," *Datasheet D02930-0-1/16(G)*, Jan. 2016. https://www.analog.com/media/en/technical-documentation/data-sheets/AD7476A_7477A_7478A.pdf (accessed Dec. 10, 2019).
- [57] Intel, "Intel Quartus Prime Standard Edition User Guide - Platform Designer," *Document UG-20174*, May 2019. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-platform-designer.pdf> (accessed Dec. 10, 2019).
- [58] Intel, "Intel FPGA IP," *Document Broch-1010-5.1*, Nov. 2019. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/br/br-intellectual-property-brochure.pdf> (accessed Dec. 10, 2019).
- [59] Intel, "Avalon Interface Specifications," *Document MNL-AVABUSREF*, Oct. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/mnl_avalon_spec.pdf (accessed Dec. 10, 2019).

- [60] S. Kilts, "Clock Domains," in *Advanced FPGA Design*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.
- [61] MathWorks, "Simulink User Guide," *Document Version 10.0 Release 2019b*, Sep. 2019. https://www.mathworks.com/help/pdf_doc/simulink/sl_using.pdf (accessed Dec. 10, 2019).
- [62] MathWorks, "HDL Coder User Guide," *Document Version 3.15 Release 2019b*, Sep. 2019. https://www.mathworks.com/help/pdf_doc/hdlcoder/hdlcoder_ug.pdf (accessed Dec. 10, 2019).
- [63] MathWorks, "Stateflow User Guide," *Document Version 10.1 Release 2019b*, Sep. 2019. https://www.mathworks.com/help/pdf_doc/stateflow/sf_ug.pdf (accessed Dec. 10, 2019).
- [64] R. Nane *et al.*, "A Survey and Evaluation of FPGA High-Level Synthesis Tools," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 10, pp. 1591–1604, Oct. 2016, doi: 10.1109/TCAD.2015.2513673.
- [65] S. Kilts, "Advanced Simulation," in *Advanced FPGA Design*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.
- [66] Intel, "Intel Quartus Prime Standard Edition User Guide - Debug Tools," *Document UG-20182*, Sep. 2018. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-debug.pdf> (accessed Dec. 10, 2019).
- [67] Intel, "Intel Quartus Prime Standard Edition User Guide - Timing Analyzer," *Document UG-20183*, Nov. 2018. <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug-qps-timing-analyzer.pdf> (accessed Dec. 10, 2019).
- [68] S. Kilts, "Static Timing Analysis," in *Advanced FPGA Design*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.
- [69] Rohde & Schwarz, "The Impact of Digital Oscilloscope Blind Time on Your Measurements," *Document 1ER02_0e*, Aug. 2010. https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1er02/1ER02_1e.pdf (accessed Dec. 10, 2019).
- [70] Intel, "Embedded Peripherals IP User Guide - SPI Core," *Document UG-01085*, Aug. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_embedded_ip.pdf (accessed Dec. 10, 2019).
- [71] Intel, "Embedded Peripherals IP User Guide - PIO Core," *Document UG-01085*, Aug. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_embedded_ip.pdf (accessed Dec. 10, 2019).
- [72] Rohde & Schwarz, "Benefits of the R&S RTO Oscilloscope's Digital Trigger," *Document 1ER04_0e*, Apr. 2012. https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/00aps_undefined/Benefits_of_RTO_digital_trigger_system_2.pdf (accessed Dec. 10, 2019).
- [73] Intel, "Embedded Peripherals IP User Guide - Modular Scatter-Gather DMA Core," *Document UG-01085*, Aug. 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_embedded_ip.pdf (accessed Dec. 10, 2019).
- [74] J. Silva, V. Sklyarov, and I. Skliarova, "Comparison of On-chip Communications in Zynq-7000 All Programmable Systems-on-Chip," *IEEE Embed. Syst. Lett.*, vol. 7, no. 1, pp. 31–34, Mar. 2015, doi: 10.1109/LES.2015.2399656.

-
- [75] Rohde & Schwarz, "High-Resolution Measurements with R&S Oscilloscopes," *Document 1TD06_0e*, Apr. 2015. https://cdn.rohde-schwarz.com.cn/pws/dl_downloads/dl_application/application_notes/1td06/1TD06_0e_RTO_HiDef.pdf (accessed Dec. 10, 2019).
- [76] Keysight Technologies, "Evaluating High-Resolution Oscilloscopes," *Document 5991-1617EN*, Dec. 2017. <https://www.keysight.com/zz/en/assets/7018-03772/application-notes/5991-1617.pdf> (accessed Dec. 10, 2019).
- [77] W. Kester, "Fundamentals of Sampled Data Systems," in *Data conversion handbook*, Amsterdam; Boston: Elsevier; Newnes, 2005.
- [78] Intel, "HPS SoC Boot Guide - Cyclone V SoC," *Document AN-709*, Jan. 2016. <https://www.intel.la/content/dam/www/programmable/us/en/pdfs/literature/an/an709.pdf> (accessed Dec. 10, 2019).
- [79] C. Simmonds, "All About Bootloaders," in *Mastering Embedded Linux Programming - Second Edition.*, Birmingham: Packt Publishing, 2017.
- [80] B. Ward, "How the Linux Kernel Boots," in *How Linux works*, 2nd edition., San Francisco: No Starch Press, 2015.
- [81] B. Ward, "How User Space Starts," in *How Linux works*, 2nd edition., San Francisco: No Starch Press, 2015.
- [82] Intel, "Intel SoC FPGA Embedded Development Suite User Guide," *Document UG-1137*, May 2019. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_soc_edu.pdf (accessed Dec. 10, 2019).
- [83] C. Simmonds, "Creating a Storage Strategy," in *Mastering Embedded Linux Programming - Second Edition.*, Birmingham: Packt Publishing, 2017.
- [84] C. Simmonds, "Configuring and Building the Kernel," in *Mastering Embedded Linux Programming - Second Edition.*, Birmingham: Packt Publishing, 2017.
- [85] C. Simmonds, "Learning About Toolchains," in *Mastering Embedded Linux Programming - Second Edition.*, Birmingham: Packt Publishing, 2017.
- [86] C. Simmonds, "Debugging with GDB," in *Mastering Embedded Linux Programming - Second Edition.*, Birmingham: Packt Publishing, 2017.
- [87] S. Chacon, *Pro Git*, Second edition. New York, NY: Apress, 2014.
- [88] M. Kerrisk, "Fundamentals Of Shared Libraries," in *The Linux programming interface: a Linux and UNIX system programming handbook*, San Francisco: No Starch Press, 2010.
- [89] M. Kerrisk, "Memory Mappings," in *The Linux programming interface: a Linux and UNIX system programming handbook*, San Francisco: No Starch Press, 2010.
- [90] G. van Rossum, *The Python Language Reference*, Release 3.7.6. Python Software Foundation, 2019.
- [91] G. van Rossum, "Networking And Interprocess Communication," in *The Python Library Reference*, Release 3.7.6., Python Software Foundation, 2019.
- [92] G. van Rossum, "Concurrent Execution," in *The Python Library Reference*, Release 3.7.6., Python Software Foundation, 2019.

- [93] G. van Rossum, "Generic Operating System Services," in *The Python Library Reference*, Release 3.7.6., Python Software Foundation, 2019.
- [94] G. van Rossum, "Data Model," in *The Python Language Reference*, Release 3.7.6., Python Software Foundation, 2019.
- [95] Owon, "XDS Series Digital Oscilloscope," *Datasheet XDS3202E/A*, Mar. 2017. https://static.eleshop.nl/mage/media/downloads/OWON_XDS_Series_n-in-1_Digital_Oscilloscope_technical_specs.pdf (accessed Dec. 10, 2019).
- [96] STMicroelectronics, "STM32 Nucleo-64 boards," *Document UM1724 Rev 13*, Apr. 2019. https://www.st.com/resource/en/user_manual/dm00105823.pdf (accessed Dec. 10, 2019).
- [97] Keysight Technologies, "Understanding Oscilloscope Frequency Response and Its Effect on Rise-Time Accuracy," *Document 5988-8008EN*, Dec. 2017. <https://www.keysight.com/zz/en/assets/7018-01129/application-notes/5988-8008.pdf> (accessed Dec. 10, 2019).
- [98] Keysight Technologies, "33120A Function/Arbitrary Waveform Generator," *Datasheet 5968-0125EN*, Nov. 2019. <https://www.keysight.com/us/en/assets/7018-06809/data-sheets/5968-0125.pdf> (accessed Dec. 10, 2019).
- [99] Texas Instruments, " μ A78xx Fixed Positive Voltage Regulators," *Datasheet SLVS056P*, Jan. 2015. <http://www.ti.com/lit/ds/slvs056p/slvs056p.pdf> (accessed Dec. 10, 2019).
- [100] Tektronix, "How Oscilloscope Probes Affect Your Measurement," *Document 51W-30013-0*, Jan. 2014. https://download.tek.com/document/51W_30013_0_MR_Letter.pdf (accessed Dec. 10, 2019).
- [101] Maxim Integrated, "MAX5389 Dual, 256-Tap, Volatile, Low-Voltage Linear Taper Digital Potentiometer," *Datasheet 19-5141; Rev 3*, Sep. 2014. <https://datasheets.maximintegrated.com/en/ds/MAX5389.pdf> (accessed Dec. 10, 2019).
- [102] Maxim Integrated, "MAX5391/MAX5393 Dual 256-Tap, Volatile, Low-Voltage Linear Taper Digital Potentiometers," *Datasheet 19-5035; Rev 3*, Oct. 2014. <https://datasheets.maximintegrated.com/en/ds/MAX5391-MAX5393.pdf> (accessed Dec. 10, 2019).
- [103] Maxim Integrated, "MAX5402 256-Tap, μ PoT, Low-Drift, Digital Potentiometer," *Datasheet 9-1896; Rev 0*, Jan. 2001. <https://datasheets.maximintegrated.com/en/ds/MAX5402.pdf> (accessed Dec. 10, 2019).
- [104] W. Kester, "Data Converter Architectures," in *Data conversion handbook*, Amsterdam; Boston: Elsevier; Newnes, 2005.
- [105] Analog Devices, "Front-End Amplifier and RC Filter Design for a Precision SAR Analog-to-Digital Converter," *Analog Dialogue 46-12*, Dec. 2012. <https://www.analog.com/media/en/analog-dialogue/volume-46/number-4/articles/front-end-amp-and-rc-filter-design.pdf> (accessed Dec. 10, 2019).
- [106] Keysight Technologies, "Oscilloscope Display Quality Impacts Ability to View Subtle Signal Details," *Document 5989-2003EN*, Dec. 2017. <https://www.keysight.com/zz/en/assets/7018-01274/application-notes/5989-2003.pdf> (accessed Dec. 10, 2019).
- [107] Keysight Technologies, "Three Compelling Reasons for Deep Acquisition Memory," *Document 5991-1822EN*, Aug. 2014. <https://www.keysight.com/us/en/assets/7018-03812/application-notes/5991-1822.pdf> (accessed Dec. 10, 2019).