



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**"IMPLEMENTACIÓN DE ACCIONES BÁSICAS DE CONTROL
EN ARREGLOS ANALÓGICOS PROGRAMABLES EN CAMPO (FPAA)"**

TESIS

PARA OBTENER EL TÍTULO DE:

INGENIERO EN ELECTRÓNICA

PRESENTA:

EMMANUEL AGUILAR AQUINO

DIRECTOR:

M. C. JORGE LUIS BARAHONA ÁVALOS

CO-DIRECTOR:

DR. ENRIQUE GUZMÁN RAMÍREZ

HUAJUAPAN DE LEÓN, OAXACA, JULIO DE 2015

Tesis presentada el 16 de Julio de 2015
ante los siguientes sinodales:

M.C. Arturo Pablo Sandoval García
M.C. Felipe Santiago Espinosa

Director:
M.C. Jorge Luis Barahona Ávalos
Co-director:
Dr. Enrique Guzmán Ramírez

*A mi madre:
María Luisa*

*A mis hermanos:
Alejandro y Martha,
Uriel, Maritza y Mariel*

Agradecimientos

Son muchos los sacrificios que se llevan a cabo cuando se quiere lograr una meta. Sin embargo, cuando ésta es alcanzada, la satisfacción que se genera es enorme y digna de pre-sunción. A través de los años, se aprenden diversas cuestiones, tales como la de supervivencia lejos de la familia y de la casa, el ser responsable ante un proyecto y el valorar los esfuerzos que día con día se presentan en el camino.

Por ello y más, doy gracias a Dios por permitirme llevar a cabo estos proyectos de vida al lado de diversas personas, las cuales me han mostrado el apoyo incondicional y a la vez he aprendido de sus consejos. Sin omitir el nombre de alguna de ellas, quiero dedicar este trabajo de tesis a las personas que estuvieron conmigo no sólo en el tiempo de la universidad, sino durante el recorrido a través de mi formación como estudiante.

Este trabajo quiero dedicarlo principalmente a mi madre, María Luisa, una persona ejemplar, capaz de dar el todo por el todo por los seres que ama, sus hijos, gracias a sus sabios consejos y regaños hoy día somos personas de bien, gente con carácter capaz de enfrentar las dificultades que se nos presenten en el camino. Mi madre, ejemplo de guerrera constante, digna de respeto y admiración. Madre, sabes que siempre agradezco lo mucho que has hecho por mi y por mis hermanos. Honestamente, me enorgullece el mostrar este trabajo a tu persona, gritándole al mundo que “todo es posible con esfuerzo y dedicación”.

A mis hermanos, Uriel y Alejandro, dos personas que desde chiquitos han sido y seguirán siendo mi mayor orgullo. Nunca olviden las veces que nos poníamos a discutir en nuestra infancia, el porqué es simple, siempre terminábamos riendo a carcajadas y por derecho con unos buenos chanclazos y regaños por parte de mamá, esos momentos son un grato recuerdo de nuestra niñez. Ustedes saben que los adoro y, quiero que este trabajo muestre para ustedes que el redirse no es una opción sino hay que luchar hasta lograr el objetivo. Aunado a ello, agradezco el apoyo brindado por sus esposas (Maritza y Martha, respectivamente). También a la pequeña Mariel, quien me ha brindado momentos de alegría con tan sólo verla sonreír, te quiero sobrina. Saben que ante cualquier circunstancia cuentan conmigo, los adoro hermanos.

A mis abuelos, Margarita y Rufino, quienes han estado con nosotros incondicionalmente. Gracias abuelitos por esos consejos que como bien dicen, “las canas no representan años sino sabiduría”. Gracias a Dios por permitirme compartir con ustedes este y muchos momentos de alegría.

A mis tíos y primos por sus palabras y el ánimo brindado cuando las cosas tornaban un tanto complicadas, gracias por todas esas pláticas que nos han dejado infinidad de enseñanzas. También agradezco por esos momentos de risa que a todos nos han alegrado.

A mi padre, Samuel, quién a pesar de la distancia, quiero agradecerle por el apoyo brindado a cumplir mi objetivo como estudiante. Quiero que sepas que estoy muy contento por ello.

A mis padrinos, Alba y Francisco, quienes han estado presentes conmigo desde chavito. Gracias por sus consejos y el apoyo incondicional que me han brindado. Sinceramente, me da gusto compartir con ustedes este logro.

No haciendo a un lado, agradezco los consejos y el apoyo brindado por un gran amigo, Gerard, quién al pasar de los años ha estado presente para darme mis buenos zapes. Sabes que te estimo y, honestamente, que afortunado de haber topado un amigo como tú, échale ganas, sabes que tienes un objetivo y quiero que lo cumplas, que éste trabajo de tesis sea motivación para tu formación como estudiante. Cuando sientas que las cosas van mal, recuerda que tienes mucho por lo cual levantarte. Y si un día tienes nada que hacer (raro en ti), invita a echar desmadre.

A mis ahijados, quienes me han enseñado que recordar es volver a vivir, gracias por esos momentos de risas interminables y por permitirme ser parte de su familia.

A todos aquellos amigos que en momentos difíciles estuvieron ahí para alentar y que en momentos de alegría estuvieron más que puestos a tirar la casa por la ventana: Nomin, Aldahir, Diego, Yessi, Flora, Isaac, Chuy, Pichon, Tere, Avispa, Chucho, Fabi, Sai, Pal. Que gusto me da compartir momentos como este con todos ustedes.

A Memin y Betin, por haber aprendido a trabajar en equipo, mostrando entre los 3 que no todo es diseñar PCBs y molestar a un tercero (recuerden que aún me la deben, par de castrosos). Recuerdo el día en que planteamos la idea de trabajar como equipo en nuestro primer semestre, todo un caos que afortunadamente resulto de lo mejor.

Al sierra, un amigo que hoy día no se encuentra con nosotros, pero en vida fue un excelente amigo. Donde quiera que estés, te agradezco el apoyo brindado en aquel periodo de estancias y por todas aquellas pláticas al entrar a la universidad.

A Esther y Ed, dos personas a las que estimo y agradezco por el apoyo brindado en diversas situaciones. Comparto con ustedes este trabajo y la alegría que ello implica.

A mi director de tesis, el M.C. Jorge Luis Barahona Ávalos, quien se mostró atento ante los avances de tesis presentados y ante mi insistencia por saber su criterio. Gracias profesor, por permitirme trabajar en este tema, honestamente, me quedo con la satisfacción de dar

un aporte práctico para la elaboración de circuitos analógicos ante diversas materias impartidas en la UTM. Aunado a ello, agradezco los consejos brindados para mi formación académica.

A mi co-director, el Dr. Enrique Guzmán Ramírez, con quien desde que platiqué por vez primera, se mostró interesado en proporcionar orientación al desarrollo de este trabajo. De igual manera, agradezco el tiempo dedicado en asesorías del área digital y por sus críticas para la redacción de este documento.

A mis sinodales, los profesores M.C. Felipe Santiago y M.C. Arturo Pablo Sandoval, por el tiempo brindado a la revisión y el dar sus puntos de vista para mejorar este documento de tesis.

A 2 de las profesoras del instituto de computación de la UTM, M.E.C. Celia Bertha Reyes Espinoza y Lic. María Pérez Córdoba Sánchez, por el tiempo dedicado a explicar conceptos de programación estructurada y estructura de datos que en su momento me fueron difíciles de interpretar. Gracias por aquellas enseñanzas y tips de programación que hoy día me son de gran utilidad.

Al profesor M.C. Salvador Montesinos Gonzáles, por el apoyo brindado a la colaboración en DIMAP y por los consejos brindados en el ámbito académico.

En general, a los profesores de la UTM, que por medio de clases o asesorías aportaron conocimientos para mi formación en el área de la ingeniería en electrónica.

Al apoyo brindado por la Secretaría de Educación Pública a través del programa de becas del SUBES denominado “Becas de Titulación 2014”.

Al equipo de Anadigm[®] *support*, que ante las dudas planteadas en mis mensajes, se mostraron atentos en resolver cuestiones en relación con el kit empleado para el desarrollo de esta tesis. Agradezco el permitirme interactuar con soporte técnico mediante el idioma inglés, cuestión que deja en mi persona una gran experiencia.

Finalmente, agradezco a todos por cada una de sus enseñanzas.

Índice

Dedicatoria	v
Agradecimientos	vii
Índice general	xi
Índice de figuras	xiv
Índice de tablas	xviii
1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Justificación	2
1.3. Hipótesis	3
1.4. Objetivos	3
1.4.1. Objetivo general	3
1.4.2. Objetivos específicos	4
1.5. Metas	4
1.6. Metodología de desarrollo	4
1.6.1. Identificación y delimitación del problema	4
1.6.2. Análisis del modelo del sistema	4
1.6.3. Diseño y modelado de controladores	5
1.6.4. Simulación	5
1.6.5. Implementación del prototipo	5
1.6.6. Experimentación	5
1.6.7. Documentación	5
1.7. Contenido del documento	5
2. Marco teórico	7
2.1. Sistemas de control retroalimentados	7
2.2. Sistemas de control en lazo cerrado	7
2.3. Controladores o compensadores	8
2.4. Clasificación de los controladores industriales	8

2.4.1.	Acción de control Proporcional (P)	9
2.4.2.	Acción de control Integral (I)	9
2.4.3.	Acción de control Derivativo (D)	10
2.4.4.	Acción de control Proporcional-Integral (PI)	10
2.4.5.	Acción de control Proporcional-Derivativo (PD)	11
2.4.6.	Acción de control Proporcional-Integral-Derivativo (PID)	11
2.5.	Diseño de reguladores del tipo P, PI, PID	12
2.5.1.	Reglas de Ziegler-Nichols	12
2.5.1.1.	Primer método para la sintonización de controladores	12
2.5.1.2.	Segundo método para la sintonización de controladores	13
2.6.	Arreglos Analógicos Programables en Campo (FPAA)	14
2.6.1.	Evolución de los FPAA	14
2.6.2.	Capacitores conmutados	15
2.6.2.1.	Sistemas muestreados	17
2.6.2.2.	Aplicaciones de la técnica SC	19
2.6.3.	Arquitectura de un FPAA	21
2.6.3.1.	El FPAA de Lattice	21
2.6.3.2.	El FPAA de Zetex	22
2.6.3.3.	El FPAA de Anadigm	22
2.6.4.	El FPAA AN231E04 de Anadigm	29
2.6.4.1.	Sección digital	30
2.6.5.	Programación de un FPAA	31
2.6.5.1.	Interfaz RS-232	33
2.6.5.2.	Interfaz USB	33
2.7.	Módulo de temperatura LTR701	34
2.8.	Estado del arte	36
3.	Modelado y simulación del sistema de control de temperatura	39
3.1.	Diseño de controladores PID	39
3.1.1.	Controlador Proporcional	40
3.1.2.	Controlador Proporcional-Integral	40
3.1.3.	Controlador Proporcional-Integral-Derivativo	41
3.1.4.	Controlador Proporcional-Derivativo	41
3.2.	Simulación de los controladores	42
3.2.1.	MATLAB Simulink [®]	42
3.2.1.1.	Controlador Proporcional	42
3.2.1.2.	Controlador Proporcional-Derivativo	43
3.2.1.3.	Controlador Proporcional-Integral	44
3.2.1.4.	Controlador Proporcional-Integral-Derivativo	45
3.2.2.	AnadigmDesigner2 [®]	46
3.2.2.1.	Modelado de la planta de temperatura	47

3.2.2.2.	Controlador Proporcional	51
3.2.2.3.	Controlador Proporcional-Derivativo	52
3.2.2.4.	Controlador Proporcional-Integral	53
3.2.2.5.	Controlador Proporcional-Integral-Derivativo	54
4.	Implementación del sistema de control de temperatura	57
4.1.	Sistema de control de temperatura	57
4.1.1.	Planta a controlar	58
4.1.2.	Definición del <i>setpoint</i>	58
4.1.3.	Acondicionamiento de señales	58
4.1.3.1.	Amplificador inversor	58
4.1.3.2.	Amplificador no inversor	59
4.1.4.	Control basado en FPAA	61
4.1.4.1.	Control Proporcional	62
4.1.4.2.	Control Proporcional-Derivativo	64
4.1.4.3.	Control Proporcional-Integral	65
4.1.4.4.	Control Proporcional-Integral-Derivativo	66
4.1.4.5.	Índice de desempeño	67
4.1.4.6.	Recursos empleados	69
5.	Conclusiones	71
5.1.	Trabajos futuros	72
	Apéndices	73
A.	Modelado e implementación de circuitos analógicos en AD2	75
A.1.	Interfaz AD2	75
A.2.	Modulador AM	81
A.2.1.	La envolvente de AM	81
A.2.2.	Diseño del circuito en AD2	81
A.3.	Linealización de funciones	84
A.3.1.	Parámetros de diseño	85
A.4.	PWM	89
A.4.1.	Diseño de circuitos PWM	89
A.5.	Filtros analógicos	93
A.5.1.	AnadigmFilter	94
A.5.1.1.	Gráfica del filtro	94
A.5.1.2.	Parámetros del filtro	95
A.5.1.3.	Características de la tarjeta	95
A.5.1.4.	Tipo de filtro	96
A.5.1.5.	Tipo familia	96
A.5.1.6.	Parámetros del CAM	96

A.5.1.7.	Análisis de datos	96
A.5.1.8.	Exportar e imprimir	96
A.5.2.	Ejemplos de diseño de filtros	97
A.5.2.1.	Filtro pasa altas	97
A.5.2.2.	Filtro pasa bandas	100
A.6.	AnadigmPID	102
A.6.1.	Pestañas	104
A.6.2.	Control PID de una planta	104
B.	Controladores PID digitales	107
B.1.	Acondicionamiento de las señales	108
B.2.	Módulo del ADC	108
B.2.1.	DIGILENT Pmod	108
B.2.1.1.	Pmod, interfaz tipo 2	108
B.2.2.	AD7476A	109
B.3.	Módulo del DAC	111
B.4.	Módulo de control	113
B.5.	Diseño final	115
B.6.	<i>Resources</i> : Una comparativa entre FPAA y FPGA	116
	Bibliografía	119

Índice de figuras

2.1. Diagrama de bloques de un sistema de control industrial [53]	8
2.2. Diagrama de bloques de un controlador PID	11
2.3. Curva de respuesta en forma de S, [53]	12
2.4. Respuesta al escalón unitario	13
2.5. Interpretación de la Ley de Ohm	16
2.6. Capacitor conmutado, [10]	16
2.7. OpAmp con SC, [10]	18
2.8. Fase de muestreo, [10]	18
2.9. Fase de transferencia de carga, [10]	19
2.10. Conmutación de capacitores, [10]	19
2.11. Polaridad en resistencias mediante SC, [10]	20
2.12. Muestreo de entrada y salida, [10]	20
2.13. Implementación de un filtro activo	21
2.14. Diagrama de bloques de la configuración de un FPAA	21
2.15. Arquitectura de un FPAA de primera generación, AN10E40, [4]	24
2.16. Arquitectura de un FPAA de segunda generación, AN221E04, [26]	25
2.17. Arquitectura de un CAB	26
2.18. Arquitectura del AN231E04, FPAA de 3 ^{ra} generación, [17]	29
2.19. Tarjeta de desarrollo con FPAA AN231E04	30
2.20. Sección digital del kit de desarrollo	31
2.21. Diagrama para la configuración de los datos en un FPAA	31
2.22. Diagrama de bloques del Núcleo de Arranque Analógico	32
2.23. Módulo térmico LTR701	34
2.24. Panel Frontal del LTR701	34
3.1. Diagrama de bloques del controlador P	43
3.2. Respuesta del controlador P	43
3.3. Diagrama de bloques del controlador PD	44
3.4. Respuesta del controlador PD	44
3.5. Diagrama de bloques del controlador PI	45
3.6. Respuesta del controlador PI	45
3.7. Diagrama de bloques del controlador PID	46

3.8.	Respuesta del controlador PID	46
3.9.	Sistema de primer orden. Filtro pasa-bajas	47
3.10.	Cuadro de diálogo para la configuración del filtro bilineal	49
3.11.	CAM para la simulación de la función de transferencia del LTR701	50
3.12.	Modelado del controlador P en la herramienta AD2	51
3.13.	Simulación del controlador P en la herramienta AD2	52
3.14.	Modelado del controlador PD en la herramienta AD2	52
3.15.	Simulación del controlador PD en la herramienta AD2	53
3.16.	Modelado del controlador PI en la herramienta AD2	53
3.17.	Simulación del controlador PI en la herramienta AD2	54
3.18.	Ajuste de del tiempo integral controlador PI	54
3.19.	Modelado del controlador PID en la herramienta AD2	55
3.20.	Simulación del controlador PID en la herramienta AD2	55
3.21.	Ajuste de del tiempo integral controlador PID	56
4.1.	Diagrama de bloques del sistema de control de temperatura	57
4.2.	CAS de entrada	59
4.3.	Acondicionamiento para señales de entrada	59
4.4.	Esquemático del amplificador no inversor	60
4.5.	Acondicionamiento para señales de salida	60
4.6.	Circuito impreso para el acondicionamiento de señales	61
4.7.	Implementación del control Proporcional	63
4.8.	Implementación del control Proporcional-Derivativo	65
4.9.	Implementación del control Proporcional-Integral	66
4.10.	Implementación del control Proporcional-Integral-Derivativo	68
4.11.	Error integral cuadrático, ISE	68
A.1.	AnadigmDesigner2 [®]	75
A.2.	Interfaz del programa AD2	76
A.3.	Insertando CAM	76
A.4.	Parámetros del CAM de ganancia invertida	77
A.5.	Parámetros del generador de funciones	78
A.6.	Parámetros de simulación	79
A.7.	Implementación de un circuito amplificador inversor	79
A.8.	Estadísticas de simulación	79
A.9.	Reconocimiento del kit de desarrollo dentro de AD2	80
A.10.	Configuración del puerto COM	80
A.11.	Respuesta del circuito amplificador inversor	81
A.12.	Modulación AM	82
A.13.	Modulador AM en AD2	83
A.14.	Respuesta de simulación del modulador AM en AD2	83
A.15.	Modulador AM modificado	84

A.16.Respuesta de simulación del modulador AM modificado	84
A.17.Comportamiento no lineal de una función	85
A.18.Circuito para la linealización de señales en AD2	86
A.19.Asignación de valores a la LUT en AD2	87
A.20.Respuesta de linealización	88
A.21.Generador de señal PWM	89
A.22.Diseño de PWM en AD2	90
A.23.Señal de salida PWM	91
A.24.Salida PWM con $M_a = 0,25$	92
A.25.Salida PWM con $M_a = 0,50$	92
A.26.Salida PWM con $M_a = 0,75$	92
A.27.Salida PWM con señales de diferente frecuencia	93
A.28.Respuesta de la señal PWM	93
A.29.Interfaz de la herramienta AnadigmFilter	94
A.30.Ejemplo del gráfico almacenado de un filtro	97
A.31.Diseño del filtro pasa altas	98
A.32.Circuito del filtro pasa bandas creado en AD2	98
A.33.Simulación del filtro pasa altas con $f = 250KHz$	99
A.34.Simulación del filtro pasa altas con $f = 100KHz$	99
A.35.Simulación del filtro pasa altas con $f = 750KHz$	99
A.36.Diseño del filtro pasa bandas	100
A.37.Circuito del filtro pasa bandas creado en AD2	101
A.38.Simulación del filtro pasabandas con $f = 15KHz$	101
A.39.Simulación del filtro pasabandas con $f = 24KHz$	102
A.40.Simulación del filtro pasabandas con $f = 50KHz$	102
A.41.Cuadro de diálogo para los opciones del circuito	103
A.42.Interfaz AnadigmPID	103
A.43.Control PID en AD2	105
A.44.Respuesta de simulación al controlador PID con un sistema en LUT	105
B.1. Tarjeta de desarrollo Nexys 3	107
B.2. PmodAD1 Digilent	108
B.3. Diagrama de tiempos para el convertidor AD7476A	109
B.4. FSM para el módulo del ADC	110
B.5. Simulación del módulo del ADC	111
B.6. PmodDA2 Digilent	111
B.7. Diagrama de tiempos para el convertidor DAC121S101	112
B.8. FSM para el módulo del DAC	112
B.9. Simulación del módulo del DAC	113
B.10.FSM para el módulo de control	114
B.11.FSM para el módulo del <i>Enable</i>	115

B.12.Simulación para la señal Enable	115
B.13.Diagrama de bloques para el diseño de un controlador PID digital	116
B.14.Distribución de los CAMs utilizados en el FPAA	117

Índice de Tablas

2.1. Parámetros K_p , T_i y T_d basados en el primer método de Ziegler-Nichols [53] . . .	13
2.2. Parámetros K_p , T_i y T_d basados en el segundo método de Ziegler-Nichols [53] . . .	14
2.3. Principales desarrolladores de dispositivos FPAA	15
2.4. FPAAs comerciales	15
2.5. Señales de Entrada	35
2.6. Señales de Salida	35
4.1. Recursos empleados en el FPAA	69
A.1. Parámetros para el modulador AM	82
A.2. Parámetros para el circuito de linealización	85
A.3. Valores para la señal de entrada	87
A.4. Valores para la señal de linealización	88
A.5. Configuración de los CAMs	91
A.6. Parámetros para filtro pasa altas	97
A.7. Parámetros para filtro pasa bandas	100
B.1. Configuración para los pines SPI	109
B.2. Reporte de síntesis en ISE 13.2 de Xilinx	117

Capítulo 1

Introducción

El diseño de los sistemas de control lineales en lazo cerrado tiene su origen en la década de los 40s del siglo pasado, cuando se llevaban a cabo pruebas de sistemas de control con métodos basados en la respuesta en frecuencia, en particular con los diagramas de Bode. A partir de ello, diversos sistemas de control industrial emplearon controladores de tipo PID (Proporcional-Integral-Derivativo) para el control de parámetros tales como la presión o la temperatura. Aunado a esto, Ziegler y Nichols establecieron reglas para sintonizar este tipo de controladores, las cuales son denominadas “reglas de sintonización de Ziegler-Nichols”, por su naturaleza, estas reglas son consideradas empíricas [53].

Por otro lado, un Arreglo Analógico Programable en Campo (*Field Programmable Analog Array*, **FPAA**, por sus siglas en inglés), es un circuito integrado que puede ser configurado para implementar diversas funciones analógicas, considerado el equivalente analógico de los circuitos digitales FPGA (*Field Programmable Gate Array*, por sus siglas en inglés). Los elementos básicos en un FPAA son los Bloques Analógicos Configurables (*Configurable Analogue Blocks*, **CAB**, por sus siglas en inglés) que manipulan las señales y las redes de las rutas de interconexión. Las funciones analógicas a ser implementadas son definidas por un conjunto de bits de configuración cargados desde los registros de desplazamiento internos. Los bloques analógicos tienen parámetros que se pueden programar para adaptarse a alguna aplicación particular. Cada CAB puede implementar una gran variedad de funciones de procesamiento de señales analógicas, tales como: amplificación, integración, diferenciación, suma, resta, multiplicación, comparación, logaritmos y exponenciales. La estructura de los componentes de un CAB depende de las tecnologías usadas. Los FPAAs son diseñados para operar en los dominios de tiempo continuo y discreto [50].

Un FPAA en **tiempo discreto** es diseñado con tecnología de capacitores conmutados o corriente conmutada. En este caso, la idea es obtener una resistencia variable a diferentes frecuencias para la conmutación del interruptor. Las ventajas de esta tecnología pueden apreciarse en términos de la capacidad de programación y la insensibilidad a la resistencia en los interruptores (switches) de programación. Por estas razones, se elige el enfoque de los capacitores conmutados sobre otras tecnologías como MOSFET y transconductancia [50].

Un FPAA en **tiempo continuo** usualmente se diseña usando tecnología de transconductores (elementos que convierten voltaje a corriente, por ejemplo, transistor bipolar o FET). Las celdas básicas consisten en un amplificador operacional (OpAmp) y capacitores programables vinculados a través de un arreglo basado en transconductores. Estos dispositivos presentan ventajas en términos de ancho de banda con reducido rango de programación para sus parámetros [50].

El trabajo de tesis presentado comprende las etapas de: caracterización de la función de transferencia correspondiente a la planta de temperatura, diseño de controladores con base en las reglas de Ziegler-Nichols que permitirá el cálculo de las ganancias necesarias para los mismos, simulación en el software AnadigmDesigner2[®] (AD2) e implementación en un FPAA de Anadigm (AN231E04) y su aplicación al módulo térmico LTR701.

1.1. Planteamiento del problema

Los circuitos electrónicos se pueden dividir en dos categorías: digitales y analógicos. La electrónica digital utiliza magnitudes con valores discretos, mientras que la electrónica analógica emplea magnitudes con valores continuos.

La implementación de circuitos sobre un protoboard o circuito impreso mediante componentes analógicos conlleva un tiempo amplio para su diseño e implementación, y debido a la cantidad de componentes empleados, presenta la posibilidad de la pérdida de alguna conexión o el mal funcionamiento de determinado componente, dificultando la verificación de su correcto funcionamiento o la solución de problemas cuando éstos aparecen. Además, estos diseños son susceptibles a ruido. Por otra parte, el diseño de circuitos digitales como en el caso de microcontroladores y FPGAs, presenta la desventaja de tener que discretizar las señales a procesar, aunado a esto, se requiere de conocimientos en programación, ya sea lenguaje C, ensamblador o HDL (*Hardware Description Language*). Además, el uso de un microcontrolador puede incrementar el tiempo de procesamiento del diseño debido a que este dispositivo usa un paradigma secuencial.

Considerando estos inconvenientes, el uso de circuitos analógicos configurables orientados a la implementación de sistemas mediante módulos predefinidos se presenta como una alternativa de las tecnologías mencionadas. En este sentido y considerando la revisión bibliográfica referente a la tecnología de los FPAAs, en este proyecto se pretende abordar el modelado de circuitos analógicos sobre un FPAA de Anadigm, desarrollando la sintonización para las acciones básicas de control (P, PI, PD, PID), y observar el comportamiento de las mismas al aplicarlo al control de un módulo de temperatura.

1.2. Justificación

Aunque la tecnología de los FPAAs no es nueva, el estudio bibliográfico realizado en este trabajo revela que su uso se ha hecho frecuente desde hace poco tiempo, tanto en el ámbito

académico como industrial. En este sentido, es necesario resaltar la nula aplicación de estos dispositivos en cursos pertenecientes a las ingenierías impartidas en la Universidad Tecnológica de la Mixteca. Debido a lo expuesto, y como una alternativa a la implementación “tradicional” de sistemas, la presente tesis presenta el modelado e implementación de controladores basados en FPAA. Mostrando además, que esta tecnología puede ser una herramienta de gran valía en el diseño, modelado e implementación de sistemas electrónicos.

La implementación de circuitos analógicos basados en FPAA presenta diversas ventajas cuando se aplican procesamiento de señales provenientes de sensores. En primera, elimina la necesidad de requerir un convertidor analógico a digital (ADC) para discretizar la señal a procesar y minimiza la pérdida de información (como ocurre durante los procesos de muestreo y cuantificación de estos dispositivos). Segundo, El FPAA puede absorber al circuito de acondicionamiento de la señal. Además, con los cálculos analógicos, el escalamiento entre resolución y el tiempo de cálculo es por lo general mucho mejor debido a que cada “bit” adicional no requiere sus propias operaciones.

A lo anterior hay que agregar que el uso de un FPAA, no requiere del diseño de módulos que lleven a cabo la interfaz con los elementos de discretización y la interpretación de la información digital, la sincronización de los datos a manipular y la definición del tipo de operaciones a emplear en el procesamiento de datos (algoritmos de punto fijo o punto flotante). Además, el paradigma de los FPAA no es secuencial, como en el caso de los procesadores. Finalmente, los elementos que conforman a un FPAA, principalmente amplificadores operacionales, y las herramientas ofrecidas por los fabricantes para modelar sistemas, hacen que el proceso de desarrollo de sistemas basados en FPAA sea relativamente simple y realizados en un tiempo corto.

1.3. Hipótesis

El desarrollo de controladores de forma analógica sobre tarjetas de adquisición basadas en FPAA, pueden mejorar su desempeño al minimizar las pérdidas generadas durante el proceso de muestreo en comparación con su implementación digital, donde el problema principal es la pérdida de datos al discretizar las funciones.

1.4. Objetivos

1.4.1. Objetivo general

Implementar y evaluar el desempeño de acciones básicas de control en dispositivos analógicos programables en campo (FPAA) sobre el módulo didáctico de temperatura (LTR701) que se tiene en el laboratorio de control de la Universidad Tecnológica de la Mixteca.

1.4.2. Objetivos específicos

1. Implementar y evaluar el desempeño de una acción de control proporcional (P) en un kit de desarrollo con FPAA de Anadigm Design.
2. Implementar y evaluar el desempeño de una acción de control proporcional-derivativa (PD) en un kit de desarrollo con FPAA de Anadigm Design.
3. Implementar y evaluar el desempeño de una acción de control proporcional-integral (PI) en un kit de desarrollo con FPAA de Anadigm Design.
4. Implementar y evaluar el desempeño de una acción de control proporcional-integral-derivativa (PID) en un kit de desarrollo con FPAA de Anadigm Design.

1.5. Metas

- * Diseñar los controladores PID mediante las reglas de sintonización de Ziegler-Nichols a partir de la función de transferencia de la planta de temperatura para su implementación en un FPAA.
- * Simular el control en lazo cerrado de los controladores diseñados y la función de transferencia de la planta en uso para observar el comportamiento dentro de su implementación analógica.
- * Diseñar e implementar la etapa de acondicionamiento entre la tarjeta FPAA y el módulo térmico LTR701.
- * Realizar pruebas de funcionamiento a la planta de temperatura a controlar.

1.6. Metodología de desarrollo

Con el propósito de cumplir con los objetivos planteados en este proyecto de tesis, la metodología utilizada en su desarrollo consta de 6 fases que son descritas a continuación.

1.6.1. Identificación y delimitación del problema

Se plantea el diseño de controladores analógicos sobre una tarjeta FPAA con la finalidad de observar su comportamiento al ser implementados sobre un sistema (módulo de temperatura LTR701) de control a lazo cerrado.

1.6.2. Análisis del modelo del sistema

En el desarrollo de esta etapa, se determinan los parámetros requeridos para el diseño de controladores PID mediante la función de transferencia que representa el comportamiento de la planta de temperatura.

1.6.3. Diseño y modelado de controladores

Con base en el método de Ziegler-Nichols, se calcularán los valores de las ganancias para el diseño de los controladores a desarrollar en la aplicación del control de lazo cerrado de la planta de temperatura.

1.6.4. Simulación

Mediante el uso de software especializado se lleva a cabo la simulación de los controladores antes diseñados con la finalidad de observar el comportamiento del sistema previo a su implementación, para ello, se hará uso de la herramienta AD2 que permite la integración de los bloques analógicos a programar dentro del FPAA. Si el diseño realizado cumple con los requerimientos para la estabilización del sistema, se procederá a la siguiente etapa, en caso contrario se hará un nuevo diseño de los controladores.

1.6.5. Implementación del prototipo

Una vez hechos los ajustes necesarios para la sintonización de los controladores y, con base en las simulaciones, observando que las ganancias son las adecuadas, se llevará a cabo la implementación práctica del controlador en el FPAA para el módulo térmico LTR701. Además, es necesario el diseño y construcción de las etapas de acondicionamiento para el adecuado funcionamiento del sistema.

1.6.6. Experimentación

Una vez adecuado el sistema y considerando los diseños propuestos, se llevarán a cabo las pruebas pertinentes y se verificará que los resultados obtenidos sean los adecuados. Al igual que en la etapa de “simulación”, si el controlador implementado resulta adecuado para la estabilización del sistema, se procederá a concluir la etapa de desarrollo del controlador en uso, en caso contrario, se retomará el diseño a partir de la etapa de “análisis del modelo del sistema”.

1.6.7. Documentación

Esta etapa de la metodología, requerirá de la escritura constante de los resultados obtenidos, aunado a ello, su revisión. Una vez concluida cada una de las etapas y realizado las correcciones pertinentes, se detallará el documento en conjunto para la presentación correspondiente al archivo de tesis.

1.7. Contenido del documento

Este documento está organizado de la siguiente manera. En el capítulo 2 se presentan los conceptos teóricos relacionados con el tema que se aborda, conteniendo la descripción para la sintonización de los controladores PID, la evolución, arquitectura y programación de un

FPAA, así como la descripción del LTR701. El capítulo 3 versa sobre el modelado de los controladores PID con base en las reglas de sintonización de Ziegler-Nichols y su simulación en los entornos de desarrollo MATLAB Simulink[®] y AD2. En el capítulo 4, se detalla la implementación del circuito de acondicionamiento para las señales de entrada y salida al FPAA, así como la programación del mismo. Aunado a ello, se muestran los resultados obtenidos para cada controlador implementado. En el capítulo 5, se presentan las conclusiones generadas por la presente investigación y se citan posibles aplicaciones a desarrollar como trabajos futuros. Finalmente, en los anexos se muestra la forma de utilizar el entorno AD2 y una propuesta para el modelado de las leyes de control mencionadas utilizando un lenguaje de descripción de hardware.

Capítulo 2

Marco teórico

Basándose en el modelo del sistema real a controlar, la teoría de control moderna permite la simplificación en el diseño, sin embargo, la estabilidad del sistema está estrechamente ligada con el error existente entre el sistema real y su modelo, lo cual conlleva a presentar la inestabilidad del sistema cuando se aplica un controlador diseñado con base en el modelo del sistema real. Para evitar este inconveniente, se deberá diseñar el sistema de control de acuerdo con el rango de los posibles errores para después sintonizar el controlador de manera que, si tomando como referencia al error del sistema cuando está en el rango establecido, el sistema de control diseñado permanecerá estable [53].

2.1. Sistemas de control retroalimentados

Se denomina sistema de control retroalimentado al sistema que mantiene determinada relación entre la salida y la entrada de referencia, donde la diferencia obtenida se utiliza como medio de control. El sistema de control de temperatura de un invernadero es un ejemplo de control retroalimentado, donde se efectúa la medición de la temperatura ambiente y se compara con una temperatura de referencia (temperatura deseada), el termostato activa o desactiva el equipo de calefacción o de enfriamiento para asegurar que la temperatura del entorno se mantenga en un nivel deseado, independientemente de las condiciones externas [53].

2.2. Sistemas de control en lazo cerrado

Los sistemas de control retroalimentados también se denominan “sistemas de control en lazo cerrado”. En la práctica, estos términos se emplean de manera indistinta. En un sistema de control en lazo cerrado, el controlador se alimenta con la señal del error de actuación, la cual se obtiene a partir de la diferencia entre la señal de entrada (o de referencia) y la señal de retroalimentación, la cual puede ser la señal de salida o una función de la señal de salida y sus derivadas y/o integrales. Esta acción permite reducir el error y establecer la salida del sistema a un valor deseado. El uso de la retroalimentación vuelve la respuesta del sistema relativamente insensible a las perturbaciones externas y a las variaciones internas en

los parámetros del sistema, permitiendo el uso de componentes relativamente poco precisos y baratos para obtener el control adecuado de una planta determinada [53].

2.3. Controladores o compensadores

Como ya se mencionó previamente, un controlador automático compara el valor real de la salida de una planta con la entrada de referencia, determina la desviación (error) y produce una señal de control que reduce este parámetro a cero o a un valor pequeño. La señal de control que produce el controlador automático se denomina “acción de control”. La Figura 2.1 muestra un diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de medición). El controlador detecta la señal de error, que por lo general está en un nivel de potencia muy bajo y la amplifica a un nivel lo suficientemente alto. La salida de un controlador automático se alimenta a un actuador, como un motor o una válvula neumática, un motor hidráulico o un motor eléctrico (el actuador es un dispositivo de potencia que produce la entrada para la planta de acuerdo con la señal de control, a fin de que la señal de salida se aproxime a la señal de entrada de referencia). El sensor es el dispositivo que convierte la variable de salida en otra variable manejable, como desplazamiento, presión, voltaje o temperatura, que se emplea para comparar la salida con la señal de entrada de referencia. Este elemento permite la retroalimentación del sistema en lazo cerrado. El valor de la entrada de referencia debe tener las mismas unidades que la señal de retroalimentación [53].

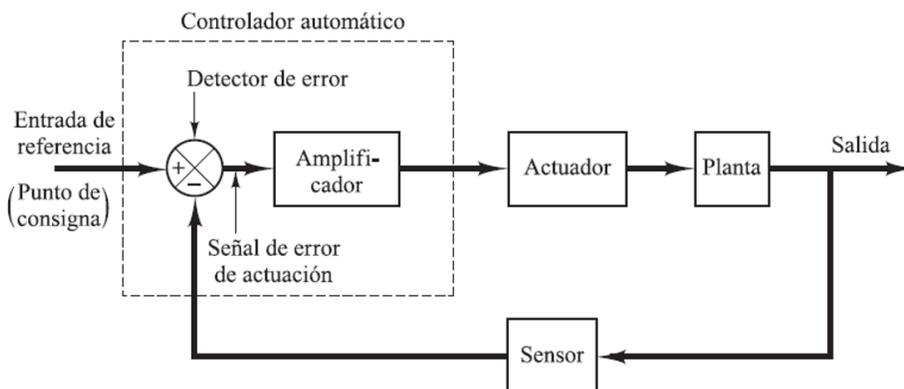


Figura 2.1: Diagrama de bloques de un sistema de control industrial [53]

2.4. Clasificación de los controladores industriales

Es importante señalar que más de la mitad de los controladores industriales empleados en la actualidad utilizan esquemas de control PID o PID modificado. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, los controladores PID resultan más útiles cuando el modelo matemático de la

planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo [53].

Los controladores industriales (también denominados como acciones básicas de control) se clasifican, de acuerdo con sus acciones de control, como:

1. Controladores ON-OFF o de dos posiciones
2. Controladores proporcionales (P)
3. Controladores integrales (I)
4. Controladores proporcionales-integrales (PI)
5. Controladores proporcionales-derivativos (PD)
6. Controladores proporcionales-integrales-derivativos (PID)

2.4.1. Acción de control Proporcional (P)

En el caso de la acción de control proporcional, la relación entre la salida del controlador $u(t)$ y la señal de error $e(t)$ se define de la siguiente manera:

$$u(t) = K_p e(t) \quad (2.1)$$

la cual se puede reescribir mediante la aplicación de la transformada de Laplace:

$$\frac{U(s)}{E(s)} = K_p \quad (2.2)$$

de la Ecuación (2.2), K_p se denomina ganancia proporcional.

En esencia, el controlador proporcional funciona como un amplificador con una ganancia ajustable [53]. Es decir, el valor de la salida de este controlador es proporcional al error y el aumento de la ganancia produce la disminución del mismo y mejora la velocidad de respuesta del sistema [38].

2.4.2. Acción de control Integral (I)

La función principal de la acción integral es establecer el valor de la salida del proceso de acuerdo con el valor del punto de referencia en estado estacionario. Con un control proporcional, normalmente existe un error de control en este estado. Con la acción integral, un error positivo pequeño dará lugar al incremento en la señal de control, mientras que un error negativo permite la disminución en el valor de la señal de control, no importa cuán pequeño sea el error. Un controlador de este tipo siempre dará cero al error en estado estacionario. Para

grandes valores de tiempos de integración, la respuesta decrece lentamente hacia el punto de ajuste (*setpoint*) [38].

En un controlador con acción integral [53], el valor de la salida ($u(t)$) cambia a razón proporcional a la señal de error $e(t)$. Es decir,

$$\frac{du(t)}{dt} = K_i e(t) \quad (2.3)$$

o bien,

$$u(t) = K_i \int_0^t e(t) dt \quad (2.4)$$

donde K_i es una constante ajustable.

La función de transferencia empleada por el controlador integral se define de la siguiente manera,

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.5)$$

2.4.3. Acción de control Derivativo (D)

El propósito de la acción derivativa es mejorar la estabilidad del lazo cerrado anticipando el efecto de la acción proporcional para estabilizar de forma más rápida la variable controlada ante cualquier perturbación. La aplicación de un control derivativo en un sistema nunca alcanzaría el estado estacionario, por lo que siempre debe utilizarse en combinación con otros controles por su influencia estabilizadora [38].

Un controlador derivativo genera una señal de control proporcional a la derivada de la señal de error.

$$u(t) = T_d \frac{de(t)}{dt} \quad (2.6)$$

donde T_d es el valor del tiempo derivativo.

2.4.4. Acción de control Proporcional-Integral (PI)

Un controlador Proporcional-Integral (PI) se define mediante

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2.7)$$

La función de transferencia del controlador obtenida al aplicar transformadas de Laplace está dada por la Ecuación (2.8).

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right) \quad (2.8)$$

donde T_i se denomina tiempo integral.

2.4.5. Acción de control Proporcional-Derivativo (PD)

La acción de control proporcional-derivativo (PD) se define mediante,

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2.9)$$

Cuya función de transferencia es

$$\frac{U(s)}{E(s)} = K_p (1 + T_d s) \quad (2.10)$$

2.4.6. Acción de control Proporcional-Integral-Derivativo (PID)

La combinación de las acciones de control básicas mencionadas previamente (P, I, D), se denomina acción de control Proporcional-Integral-Derivativa (PID), la cual presenta las ventajas de cada una de las tres acciones de control individuales. La Ecuación (2.11) modela el comportamiento de un controlador de este tipo.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.11)$$

La función de transferencia de esta acción de control se describe como,

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (2.12)$$

Considerando el concepto de controlador, en la Figura 2.2 se muestra la implementación del controlador PID mediante diagrama de bloques.

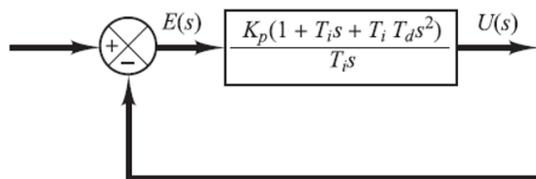


Figura 2.2: Diagrama de bloques de un controlador PID

2.5. Diseño de reguladores del tipo P, PI, PID

Los métodos de diseño para controladores PID son de gran importancia en la estabilización de sistemas reales que operan en la vecindad de puntos de equilibrio, prueba de ello es el uso frecuente y extenso de compensadores PID en aplicaciones industriales. Si se cuenta con el modelo matemático de la planta, es posible aplicar diversas técnicas para la sintonización y así obtener el valor de los parámetros del controlador que cumpla con las especificaciones para la estabilidad del sistema en lazo cerrado. Sin embargo, si la planta no presenta un modelo matemático fácil de obtener, es decir, no es posible un enfoque analítico para el diseño del controlador PID, es posible recurrir a procedimientos experimentales [53, 59].

2.5.1. Reglas de Ziegler-Nichols

Para sintonizar un controlador, se lleva a cabo el proceso de seleccionar los parámetros que cumplan con las especificaciones de comportamiento requeridas. Ziegler y Nichols sugirieron reglas para sintonizar los controladores PID basándose en las respuestas escalón experimentales o en el valor de K_p que produce estabilidad marginal cuando sólo se usa la acción de control proporcional. Estas técnicas son muy convenientes cuando el modelo matemático de la planta no se conoce, sin embargo, se pueden aplicar al diseño de sistemas con modelos matemáticos conocidos [53].

2.5.1.1. Primer método para la sintonización de controladores

Este método se puede aplicar si la respuesta de una planta muestra una curva con forma de S (Figura 2.3) dado que no contiene integradores ni polos dominantes complejos conjugados.

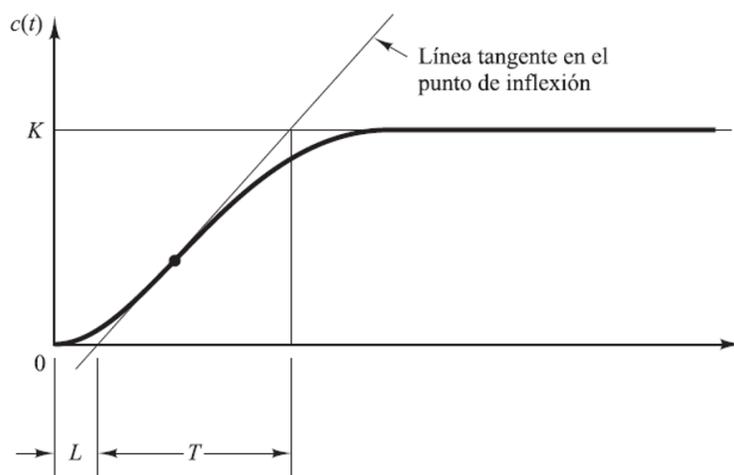


Figura 2.3: Curva de respuesta en forma de S, [53]

La curva en forma de S se puede generar mediante la experimentación o simulación dinámica de la respuesta al escalón unitario (Figura 2.4), caracterizada por dos parámetros:

el tiempo de retardo L y la constante de tiempo T , mismos que se determinan dibujando una recta tangente en el punto de inflexión de la curva y determinando las intersecciones de ésta con el eje del tiempo y con la línea $c(t) = K$, tal y como se muestra en la Figura 2.3.

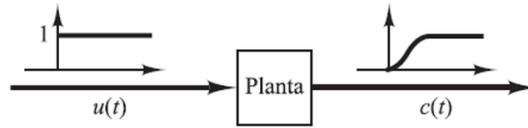


Figura 2.4: Respuesta al escalón unitario

En este caso, la función de transferencia $G(s)$ se aproxima mediante un sistema de primer orden con un retardo del modo siguiente:

$$G(s) = \frac{k}{Ts + 1} e^{-Ls} \quad (2.13)$$

De esta manera, Ziegler y Nichols sugirieron establecer los valores de K_p , T_i y T_d de acuerdo con las condiciones establecidas en la Tabla 2.1.

Tabla 2.1: Parámetros K_p , T_i y T_d basados en el primer método de Ziegler-Nichols [53]

Tipo de Controlador	Parámetro		
	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9\frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2\frac{T}{L}$	$2L$	$0,5L$

2.5.1.2. Segundo método para la sintonización de controladores

Para el uso de este método, primero se fija $T_i = \infty$ y $T_d = 0$. Usando la acción de control proporcional con la planta en un sistema en lazo cerrado, se incrementa el valor de K_p desde 0 hasta un valor crítico K_{cr} , en donde la salida presente oscilaciones sostenidas. Si la salida no presenta este tipo de oscilaciones para cualquier valor que pueda tomar K_p , entonces este método no se puede aplicar. Así, la ganancia crítica (K_{cr}) y el periodo (P_{cr}) correspondientes se determinan experimentalmente. La Tabla 2.2 muestra las condiciones requeridas para establecer los valores para la sintonización de los controladores PID.

Si el sistema tiene un modelo matemático conocido (como la función de transferencia), entonces se puede emplear el método del lugar de las raíces para encontrar la ganancia crítica y la frecuencia de las oscilaciones sostenidas (ω_{cr}), en este caso, se considera $2\pi/\omega_{cr} = P_{cr}$.

Tabla 2.2: Parámetros K_p , T_i y T_d basados en el segundo método de Ziegler-Nichols [53]

Tipo de Controlador	Parámetro		
	K_p	T_i	T_d
P	$0,5K_{cr}$	∞	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Estos valores se pueden determinar a partir de los puntos de cruce de las ramas del lugar de las raíces con el eje $j\omega$. De manera gráfica, si las ramas del lugar de las raíces no cortan al eje $j\omega$ este método no se puede aplicar.

2.6. Arreglos Analógicos Programables en Campo (FPAA)

El procesamiento de las señales analógicas en el dominio del tiempo es altamente factible y presenta ventajas importantes con respecto al procesamiento digital, es decir, no existe la necesidad de utilizar ADC y convertidores digital a analógico (DAC), generalmente consumen menos potencia y un área menor en el chip. Actualmente, se diseñan novedosos circuitos integrados analógicos y de señal mixta (analógica y digital) que usan nuevas técnicas de diseño para alcanzar altas velocidades en el procesamiento analógico. Estos son enfocados a obtener circuitos con características tales como ancho de banda amplio, alta linealidad, excelente relación señal/ruido y bajo consumo de potencia. Dentro de esta gama se encuentran los FPAA que brindan una solución eficaz a los problemas de prototipado rápido y simplifican la tarea de diseñar circuitos electrónicos analógicos [39].

2.6.1. Evolución de los FPAA

Con la aparición y evolución de los circuitos digitales programables (*Programmable Logic Devices*, PLDs, por sus siglas en inglés), surgió la necesidad de desarrollar circuitos integrados que permitieran programar e implementar circuitos de señal mixta en un solo chip, es por ello que desde la década de los 80, varias compañías y grupos de investigación (la mayoría de ellos pertenecientes a las universidades) han desarrollado y anunciado sus productos (Tabla 2.3) [39].

En la Tabla 2.4, se muestra un listado de las compañías fabricantes de semiconductores que en la actualidad producen circuitos FPAA [39].

Tabla 2.3: Principales desarrolladores de dispositivos FPAA

Año	Compañías y Grupos de Investigación
1988	Sivilotti (Caltech)
1990	Lee & Gulak (University of Toronto)
1990	Kawasaki Steel
1990	Pilkington Microelectronics
1991	Lee & Gulak (University of Toronto)
1994	IMP, Inc.
1994	Pierzchala & Perkowski (Portland State University)
1994	Chang et al (University of Nottingham)
1996	Embabi et al (Texas A&M University)
1996	Zetex Semiconductors Ltd.
1997	Guadet & Gulak (University of Toronto)
1997	Futura et al (Spain)
1997	Motorola
1998	Motorola e IMP, retiradas del mercado
1999	IspPAC de Lattice
1999	SIDSA FIPSOC anunció el nuevo chip
2000	Anadigm

Tabla 2.4: FPAAs comerciales

Fabricante	Modelo	Tecnología	Ancho de Banda
Lattice	IspPAC10	UltraMOS Tiempo Continuo	550 kHz (G=1)
	IspPAC20		330 kHz (G=100)
	IspPAC30		1.5 MHz
	IspPAC80		500 kHz
	IspPAC81		75 kHz
	IspPAC POWR1208		–
Zetex	TRAC20	Bipolar Tiempo Continuo	4 MHz
	TRAC20LH		12 MHz
Anadigm	AN10E40	Capacitor Conmutado	5 MHz
	AN120E04		2 MHz
	AN220E04		

2.6.2. Capacitores conmutados

La amplia gama de aplicaciones modernas de circuitos electrónicos se mantiene en la búsqueda de nuevas técnicas de fabricación y diseño que permitan optimizar su desempeño, minimizar los costos y el tamaño de los circuitos. Los capacitores conmutados (*Switched Capa-*

ditor, **SC**) surgen como respuesta a esta inquietud y abren paso a nuevos métodos de diseño de circuitos con mayor calidad que los circuitos clásicos. Su origen surge alrededor de los años 80, del siglo pasado, respondiendo a la necesidad de sustituir las grandes resistencias, las cuales eran imposibles de acomodar dentro de los integrados que cada día reducían en peso y tamaño. De esta manera, se fueron creando filtros con determinadas configuraciones que presentan la opción de ser programables y sintonizables [49].

Considérese un resistor R (Figura 2.5) que al aplicarle un voltaje V , se produce una corriente I , la cual es proporcional al voltaje aplicado según la Ley de Ohm (Ecuación 2.14). En dicha situación, se supone que la corriente I es producida por un movimiento de carga continuo y lineal en el tiempo [41].

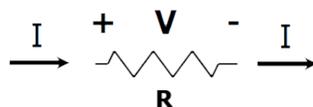


Figura 2.5: Interpretación de la Ley de Ohm

$$V = IR \quad (2.14)$$

Además del circuito mostrado en la Figura 2.6a que consta de un capacitor y dos interruptores (fases de funcionamiento), se puede obtener un resistor equivalente aplicando el análisis correspondiente.

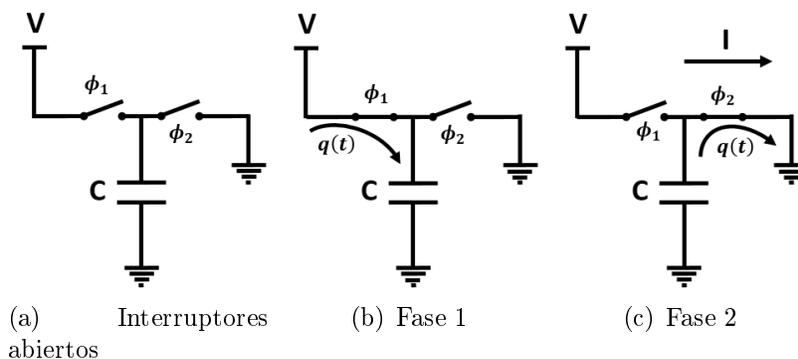


Figura 2.6: Capacitor conmutado, [10]

Cuando el interruptor ϕ_1 está cerrado y el interruptor ϕ_2 está abierto (Figura 2.6b), se produce un movimiento de la carga (q) hacia el capacitor C , provocando que este almacene voltaje (Ecuación 2.15) durante el periodo de tiempo que dura la fase 1.

$$q = CV \quad (2.15)$$

En cambio, cuando ocurre la conmutación de fase (Figura 2.6c), es decir, cuando el interruptor ϕ_1 está abierto y el interruptor ϕ_2 cerrado, se produce la descarga del capacitor.

Suponiendo que la duración del periodo (T) es igual para ambas fases y que el tiempo de conmutación entre los interruptores ϕ_1 y ϕ_2 es cero, se produce una corriente I de acuerdo a la Ecuación (2.16).

$$I_{avg} = \frac{\Delta Q}{\Delta t} = \frac{C * V}{T} \quad (2.16)$$

Para obtener el valor de la resistencia equivalente R , es necesario sustituir el valor de la corriente (Ecuación 2.16) en la Ecuación (2.14) correspondiente a la Ley de Ohm, de la siguiente manera,

$$\begin{aligned} R &= \frac{V}{I} \\ &= \frac{V}{\frac{C * V}{T}} \\ &= \frac{V * T}{C * V} \\ &= \frac{T}{C} \end{aligned}$$

Tomando en cuenta que el periodo es el valor inverso de la frecuencia, se obtiene que el valor de la resistencia equivalente (Ecuación 2.17) es inversamente proporcional al producto de la frecuencia f por la capacitancia C . Dando lugar a la variación del valor de la resistencia en función de la frecuencia de operación [10].

$$R = \frac{1}{fC} \quad (2.17)$$

A diferencia del comportamiento del resistor, el movimiento de carga en el capacitor no es continuo, sino que corresponde a un muestreo de valores, similar al comportamiento que desarrollan los sistemas electrónicos muestreados.

2.6.2.1. Sistemas muestreados

Los sistemas electrónicos basados en capacitores conmutados son sistemas muestreados que constan de dos fases: fase de muestreo y fase de transferencia de carga. Considerando el circuito de la Figura 2.7, con $C_{in} = C_{out}$ y 3 interruptores (S_1 , S_2 y S_3) ideales y perfectamente sincronizados [41], se describen a continuación las fases anteriormente citadas.

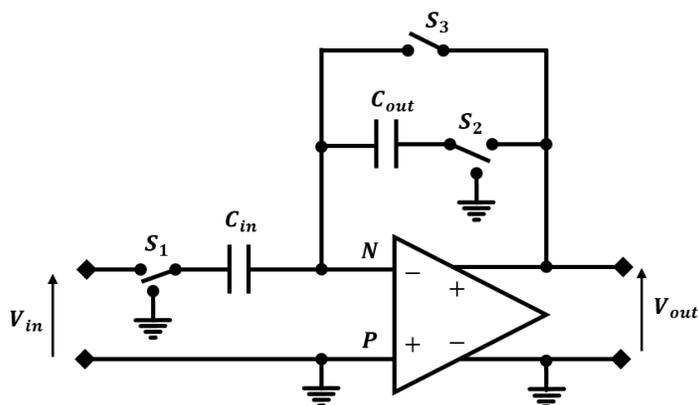


Figura 2.7: OpAmp con SC, [10]

Fase de muestreo. Durante esta fase, el circuito resultante es el mostrado en la Figura 2.8. El OpAmp se encuentra en configuración de seguidor de voltaje, por lo que durante esta fase el voltaje de salida es igual a cero ($V_{out} = 0V$), debido al cortocircuito virtual existente en la entrada negativa del OpAmp. Además, no existe corriente circulando a través del capacitor C_{out} que se encuentra cortocircuitado a tierra. Por otra parte, el capacitor C_{in} se carga por medio de la intensidad de corriente i producida por la fuente de voltaje de entrada (V_{in}).

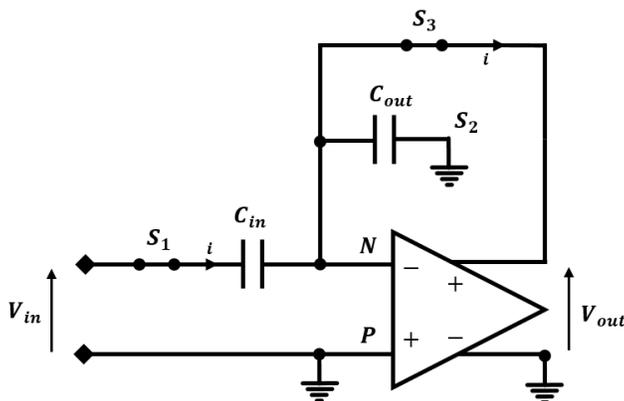


Figura 2.8: Fase de muestreo, [10]

Fase de transferencia de carga. Durante la fase de transferencia de carga (Figura 2.9), el capacitor C_{in} se encuentra cargado con el voltaje adquirido de la fase de muestreo, $V_{C_{in}} = V_{in}$. Cuando la conmutación ocurre, el OpAmp presenta retroalimentación negativa a través del capacitor C_{out} , por lo tanto, el capacitor C_{in} está cortocircuitado a tierra virtual debido al análisis ideal del OpAmp, provocando la circulación de una corriente (i) que permite almacenar carga en el capacitor C_{out} por medio del voltaje presente en C_{in} , resultando $V_{out} = V_{in}$.

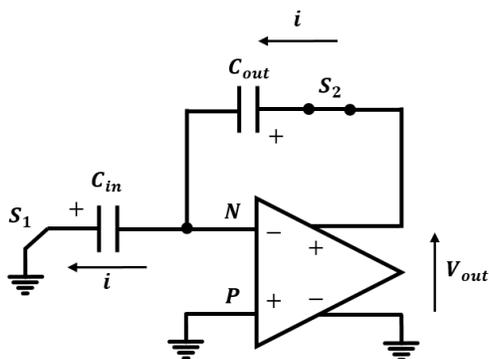


Figura 2.9: Fase de transferencia de carga, [10]

2.6.2.2. Aplicaciones de la técnica SC

Con base en la descripción del funcionamiento de un capacitor conmutado (Figura 2.6), se mostró el análisis matemático correspondiente para la configuración de un capacitor que permite obtener el valor de una resistencia equivalente, en función del valor de la capacitancia y de la frecuencia de conmutación del circuito. De esta manera y considerando el circuito de la Figura 2.10, se muestra la relación entre 2 resistencias derivadas de la técnica SC, en este caso, se observa la dependencia entre los valores de los capacitores (Ecuación 2.18). Cabe destacar que esta relación resulta independiente de la frecuencia de muestreo [10].

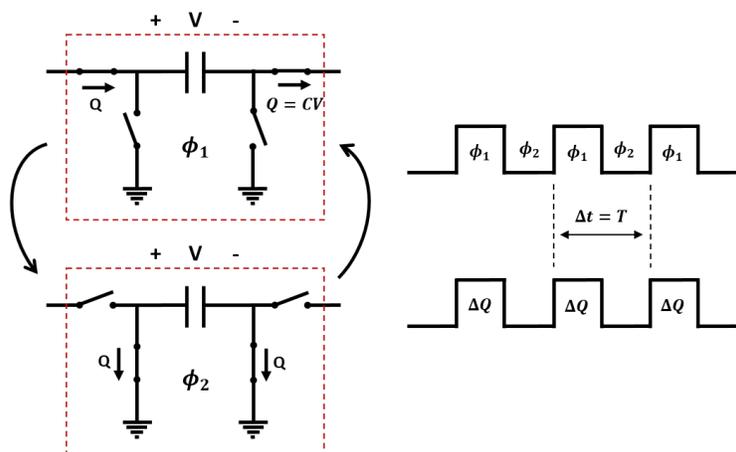


Figura 2.10: Conmutación de capacitores, [10]

$$\begin{aligned}
 \frac{R_1}{R_2} &= \frac{\frac{1}{f * C_1}}{\frac{1}{f * C_2}} \\
 &= \frac{f * C_2}{f * C_1} \\
 \therefore \frac{R_1}{R_2} &= \frac{C_2}{C_1} \tag{2.18}
 \end{aligned}$$

Además de la obtención de resistencias equivalentes a través de la técnica SC, se observa que la configuración de los interruptores produce una determinada polaridad (Figura 2.11).

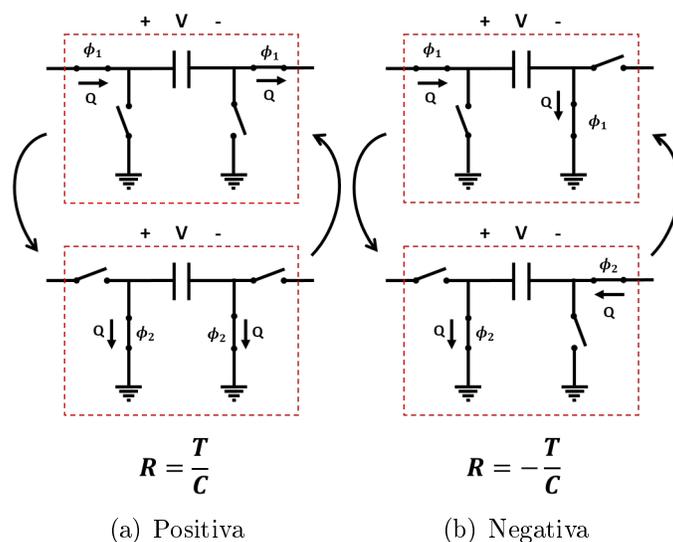


Figura 2.11: Polaridad en resistencias mediante SC, [10]

En el caso de la Figura 2.11b, la corriente producida en el instante ϕ_2 fluye en dirección opuesta a la polarización recibida por el capacitor en la fase ϕ_1 , por lo tanto, la resistencia equivalente asume un signo negativo. En la técnica SC de un sistema de muestreo analógico, las señales de entrada y salida se almacenan sólo durante la conmutación de ϕ_1 a ϕ_2 (Figura 2.12) [10].

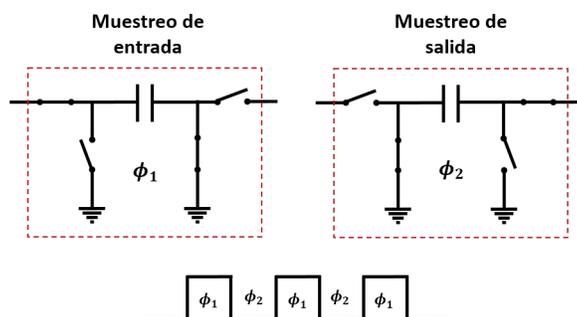


Figura 2.12: Muestreo de entrada y salida, [10]

Otra aplicación de esta técnica es el diseño de filtros activos y configuraciones con OpAmps (Figura 2.13).

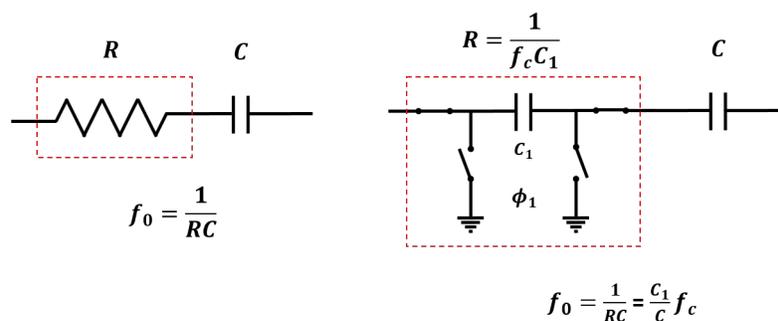


Figura 2.13: Implementación de un filtro activo

En el caso de los filtros, la frecuencia de corte se puede modificar al cambiar la frecuencia de reloj del sistema [10, 34].

2.6.3. Arquitectura de un FPAA

Un FPAA es un circuito que puede ser configurado para implementar una variedad de funciones analógicas, éste consta de un arreglo de CABs, una red de interconexión programable y un registro para almacenar los bits de configuración (Figura 2.14). De acuerdo a las características impuestas por el fabricante, la red de interconexión proporciona la ruta para efectuar las conexiones entre los CABs y los bloques de entrada y salida (I/O) [39].

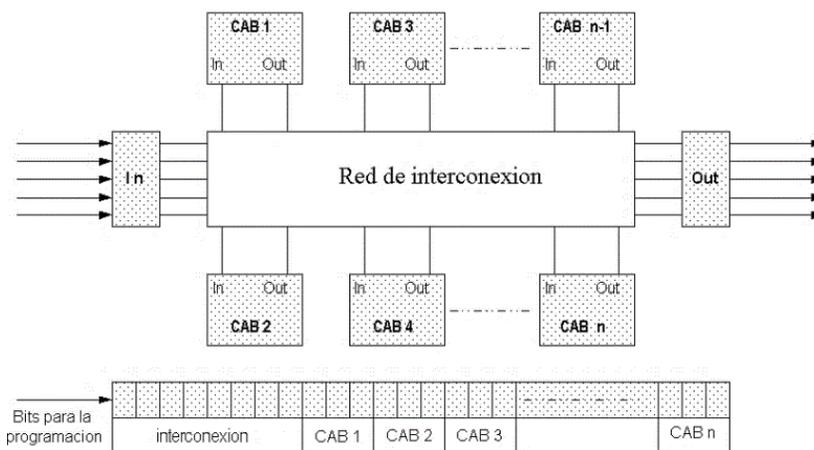


Figura 2.14: Diagrama de bloques de la configuración de un FPAA

2.6.3.1. El FPAA de Lattice

El elemento básico funcional de los circuitos FPAA de **Lattice** es el **PACell** (*Programmable Analog Cell*) que dependiendo de la arquitectura específica del circuito IspPAC, puede ser un amplificador de instrumentación, un amplificador-sumador u otra etapa activa elemental. En todos los circuitos IspPAC, las celdas programables PACells se combinan cuidadosamente

para formar macroceldas analógicas o **PACblocks**. En este caso, no se requiere ningún componente externo, lo cual hace flexible la implementación de funciones analógicas básicas tales como el filtrado con precisión, la suma o diferencia, la ganancia o atenuación y la conversión. Los circuitos IspPAC funcionan con una sola fuente de alimentación a 5V y ofrecen una arquitectura que es completamente diferencial desde la entrada hasta la salida. Lo cual duplica la eficiencia del rango dinámico versus I/O “*Single-Ended*” (voltaje de entrada). También, produce un funcionamiento mejorado con respecto a las especificaciones CMR (*Comun Mode Rejection*), PSR (*Power Supply Rejection*) y THD (*Total Harmonic Distortion*). La metodología de diseño y la programación en el sistema (*In-System Programmable*, ISP) de Lattice permite simplificar el proceso de diseño y acelerar la implementación del circuito analógico.

2.6.3.2. El FPAA de Zetex

Los FPAAs fabricados por **Zetex** se denominan **FPAD** (*Field Programmable Analog Device*, por sus siglas en inglés). Dentro de estos dispositivos se encuentran el **TRAC020** y **TRAC020LH** (versión del TRAC020 para baja potencia). El circuito TRAC se basa en una celda analógica configurable única, la cual es flexible a la programación de diferentes funciones tales como adición, negación, logaritmo, antilogaritmo, amplificación, diferenciación, integración, rectificación y seguidores de voltaje. Estas funciones son combinadas para implementar un sistema de procesamiento o acondicionamiento de señales. También, facilitan el uso de las técnicas estructuradas de diseño matemático. Las funciones básicas del TRAC020 pueden ser configuradas en cada una de las 20 celdas interconectadas entre sí para facilitar el diseño y la configuración de cualquier circuito analógico en el chip, es decir, la configuración es realizada de forma digital mediante un registro de desplazamiento, mientras que la señal permanece en el dominio analógico todo el tiempo, por lo que se evitan los errores de muestreo y retardos de procesamiento hallados en soluciones equivalentes con DSP (*Digital Signal Processing*).

2.6.3.3. El FPAA de Anadigm

La compañía Anadigm (*Analog Paradigm*) tiene dos familias de FPAAs, que están divididas en cuanto a su modelo de **reconfiguración**. La primera tiene reconfiguración **estática**, es decir, el dispositivo requiere un reinicio antes de cargar el flujo de bits (*bitstream*) de configuración. La segunda familia es **dinámicamente** reconfigurable, es decir, tiene la capacidad de reconfiguración dinámica en tiempo real, lo cual permite al dispositivo ser reconfigurado por el diseñador sin ser necesario reiniciar el sistema para cargar los cambios realizados. Esta última tiene dos tipos de memoria, la SRAM de respaldo (*Shadow*) y la SRAM de configuración. Los nuevos datos de configuración son almacenados en la *Shadow SRAM*, los cuales son transferidos a la SRAM de configuración en un flanco de reloj provisto para sincronizar la actualización de la función analógica del circuito [24, 39].

A la fecha, se han desarrollado tres generaciones de dispositivos, la primera de ellas ha sido retirada del mercado. La diferencia entre una y otra radica en el modelo de la arquitectura. Utilizando tecnología CMOS, se ha empleado la técnica SC para la aplicación de resistencias en el sistema de muestreo de los dispositivos de cada generación. Lo cual permite

que el circuito sea menos vulnerable a los cambios en los procesos y a las corrientes parásitas, aumentando el rendimiento de la programación del dispositivo. La desventaja de esta técnica se debe a las limitaciones en el rango de frecuencia de funcionamiento del componente, ya que cuando se utiliza el método de muestreo, la señal analógica de entrada debe cumplir con el teorema de muestreo de Nyquist, el cual indica que la frecuencia de muestreo debe ser al menos el doble del valor de la frecuencia de la señal a muestrear.

Para identificar las características primordiales de los chips FPAA de Anadigm, se debe referir a la nomenclatura que se les ha asignado, **ANxxxExx**, donde la primer “x” se refiere al tipo de reconfiguración, “1” para estática y “2” para dinámica; la segunda “x” indica la generación del producto, “2” para la segunda, “3” para la tercera; la tercer “x”, se refiere a la interfaz de E/S y al ADC, “0” para E/S fijas y ADC exclusivamente interno y “1” para E/S flexibles y ADC por CAM (*Configurable Analog Module*, por sus siglas en inglés); por último, las “x” restantes indican la cantidad de CABs disponibles en el componente. Cabe mencionar que esta nomenclatura no se aplica a los dispositivos de la primera generación [24, 49].

Primera generación. El FPAA AN10E40 es el único dispositivo de esta generación. Dispone de macros llamados **IPmodules** que permiten implementar funciones de amplificación, suma, integración, diferenciación, comparación, rectificación, fuentes de DC (voltajes de referencia), filtros, osciladores senoidales, circuitos S/H (*Sample and Hold*) y T/H (*Track and Hold*), es decir, puede ser utilizado en aplicaciones para la generación y el filtrado de señales, la implementación de circuitos de control, entre otros. El chip se divide en 20 CABs, cada uno con un OpAmp, cinco bancos de capacitores e interruptores. Cuatro de los bancos de capacitores se localizan entre el bloque de interconexión interno (*local routing connections*) y los interruptores, el banco restante está en el lazo de retroalimentación del OpAmp. En el lado izquierdo de los bancos de capacitores existen diversas etapas de entrada, a las cuales llegan las entradas denominadas *Local Inputs* y, sus salidas llegan a los bloques de interruptores. Del lado derecho del OpAmp están las salidas del CAB. La información para las interconexiones y el comportamiento de los CABs es almacenada en el bloque SRAM, la cual es cargada durante la configuración. Este proceso generalmente ocurre al energizar el circuito, pero puede ser reiniciado en cualquier instante. La habilidad para reconfigurar el bloque SRAM en cualquier momento le brinda al usuario una gran flexibilidad para diseñar un sistema. El AN10E40 está organizado en una matriz de CABs de 4x5, una red de interconexión para el reloj, interruptores y recursos para interconexión global y local (Figura 2.15). Cada CAB del AN10E40 es programable, lo cual permite una gran flexibilidad para diseñar diferentes circuitos para procesamiento analógico. La alimentación típica del circuito es de 5 V, con una estructura completamente diferencial.

La lógica para la configuración (*Configuration Logic*) y el registro de desplazamiento (*Shift Register*) trabajan en conjunto siempre que la configuración del chip esté en proceso. La matriz de CABs está rodeada por las celdas analógicas I/O programables, 13 en total, con dos OpAmp de soporte. El chip también tiene un generador de voltaje de referencia (V_{ref}) programable. Las celdas analógicas I/O son flexibles y permiten conectar directamente la circuitería del núcleo (*core*) del chip con los pines de entrada o salida.

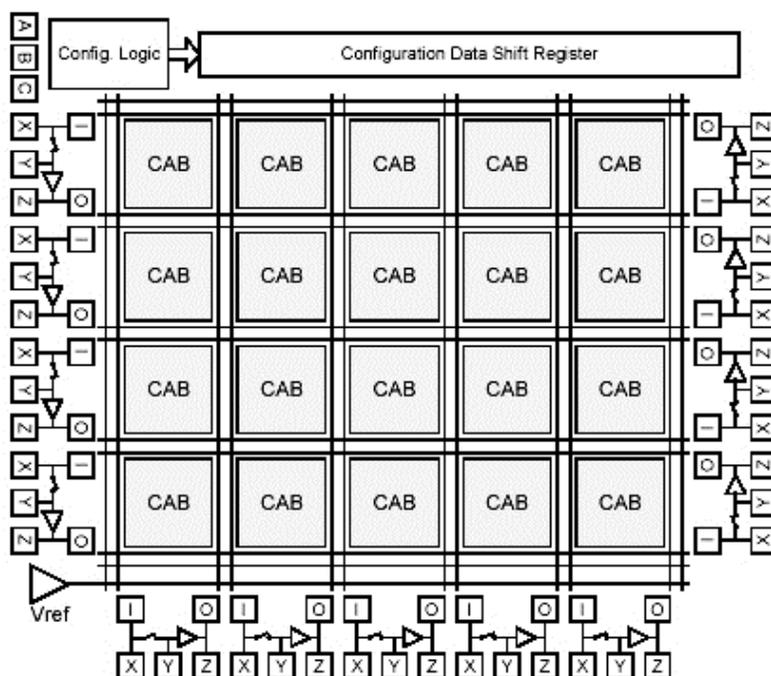


Figura 2.15: Arquitectura de un FPAA de primera generación, AN10E40, [4]

Debido a que el FPAA AN10E40 está basado en circuitos SC, sus señales de salida no están libres de la presencia de ruido, entonces, la flexibilidad de las celdas I/O es importante cuando se considera la implementación de filtros antialiasing. Debido a la naturaleza misma del sistema de datos muestreados (*Sampled Data System*), se debe tener cuidado en limitar el ancho de banda de la señal de entrada para evitar aliasing.

Segunda generación. También conocida como **AnadigmVortex**. Estos dispositivos emplean la arquitectura de Anadigm con alimentación de 5V. Considerando la nomenclatura ANx2xE04, se observa que se trata de un circuito con 4 CABs. En general, esta familia de dispositivos proporciona una mejora significativa en la relación señal/ruido, así como un mayor ancho de banda respecto de la anterior. También, cuentan con un CAM para ajustar las funciones no lineales, tales como la linealización de la respuesta de los sensores, la síntesis de formas de onda arbitraria o procesos de auto-calibración. Este ajuste es realizado por la LUT (*Look-Up Table*), que es compartida por todos los CABs. Otro cambio importante se debe a la configuración de la memoria [49].

Los FPAAs AN120E04 y AN220E04, son ejemplos de dispositivos de la familia de Anadigm-vortex, los cuales están basados en una arquitectura SC completamente diferencial. Constan de una matriz de CABs de 2x2, una red de interconexión programable, una LUT, 4 celdas analógicas de entrada (una de ellas con un multiplexor para 4 señales de entrada), 3 celdas de salida, reconfiguración estática y dinámica respectivamente y, E/S fijas con ADC interno. Mientras

que el FPAA AN221E04 cuenta con 7 Interfaces de E/S configurables (una con multiplexación 4:1), 2 Interfaces de salida dedicada, una LUT, un ADC tipo SAR (*Successive Approximation Register*) en cada CAB, un bloque generador de voltaje de referencia, un bloque de reloj del sistema, un bloque de interfaz de configuración, reconfiguración dinámica y un ADC CAM (Figura 2.16). Cada una de las celdas de entrada tiene un filtro antialiasing programable y un amplificador de alta ganancia con bajo offset de entrada. Los CABs pueden ser programados a través de la LUT, lo cual permite realizar la implementación de funciones arbitrarias [26, 39].

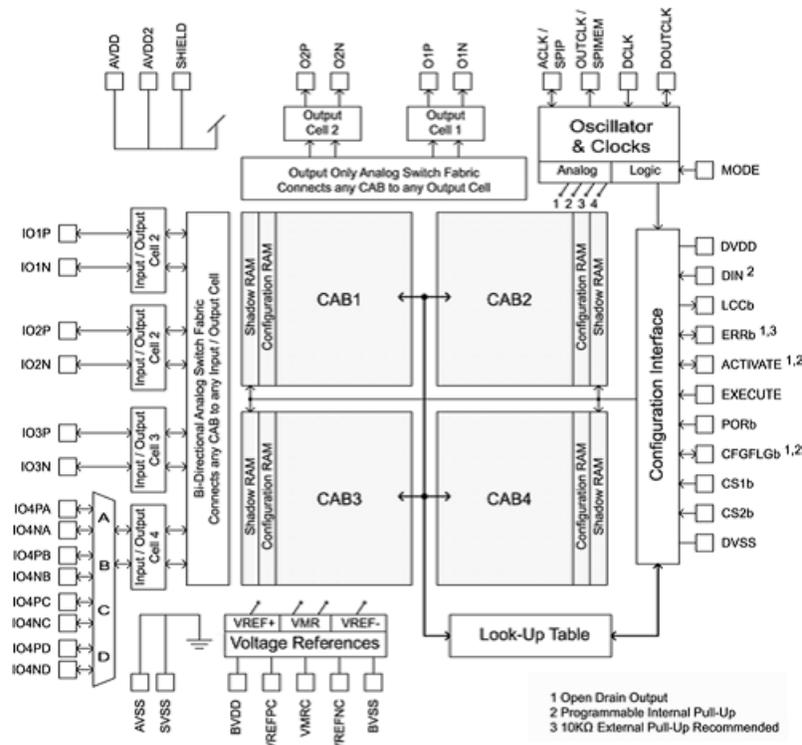


Figura 2.16: Arquitectura de un FPAA de segunda generación, AN221E04, [26]

CAB. El CAB mostrado en la Figura 2.17, correspondiente al AN221E04, está formado por dos matrices de conmutadores analógicos (*Switch Matrix*), un arreglo de 8 capacitores programables (situado entre las matrices) de pequeña dimensión e igual tamaño con un valor entre 0 y 255 unidades de la capacitancia. Seguida de 2 OpAmps diferenciales a 50MHz y un comparador, los cuales son retroalimentados a la matriz de conmutadores analógicos de la entrada.

Para que el circuito funcione de manera adecuada, se necesita del uso de señales de reloj no solapadas (*Non-Overlapping, NOL*), las cuales son proporcionadas por el bloque *NOL Clock Generator*, obtenidas a partir de las 4 señales de reloj del sistema.

Otro bloque dentro del CAB, está compuesto por un ADC con registro de aproximaciones sucesivas (SAR) de 8 bits, que junto a la LUT disponible en el FPAA se puede utilizar para

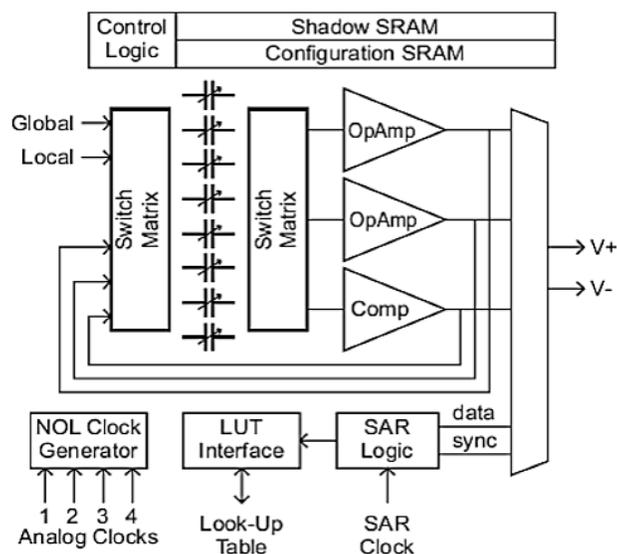


Figura 2.17: Arquitectura de un CAB

implementar funciones analógicas no lineales como la multiplicación de voltajes, el control automático de ganancias o la linealización. También se puede usar el ADC junto con alguna de las celdas de salida en modo digital para la obtención de un ADC de 8 bits.

Por último, el bloque de configuración y control lógico está compuesto por una memoria SRAM de configuración, una *shadow SRAM* y la lógica de control. Dicha estructura permite definir la topología necesaria dentro del CAB para implementar los diseños disponibles en la herramienta CAD que proporciona el fabricante.

“On the fly”. La memoria *shadow SRAM* se utiliza para precargar la configuración de los CABs, permitiendo reconfigurar el FPAA sin necesidad de reiniciarlo o interrumpir la conexión de las señales. Además, sólo se modifican los valores que se cambian en el diseño, no siendo necesaria la reescritura de toda la configuración e interrumpir la conexión de las señales, lo que permite cambios de configuración en periodos de tiempo relativamente cortos.

LUT. La única tabla de búsqueda de 256 bytes con 8 bits en su dirección de entrada disponible en el FPAA, cuenta con dos modos de funcionamiento. En el primero de ellos, la selección de la dirección de memoria de la LUT la utilizan los ADC SAR. En el segundo, la LUT emplea un contador incremental propio que se autoresetea al llegar al final de la cuenta. En este caso, la LUT sitúa datos contenidos en ella sobre dos zonas objetivo de escritura de la *shadow SRAM*, permitiendo activar la escritura de la SRAM de configuración y, por tanto, cambiar el diseño implementado en el FPAA mediante la activación de la señal externa EXECUTE, la definición de un punto de corte con un comparador, una detección de paso por cero o cuando el último byte de la nueva configuración está escrita en la *shadow SRAM*. Este modo se utiliza generalmente para la generación de señales arbitrarias o para modulaciones

temporales analógicas.

Operaciones con el ADC SAR. Cada CAB dispone de un SAR que permite la implementación de un ADC de 8 bits por aproximaciones sucesivas. Para ello, son necesarias dos señales de reloj con frecuencias en proporción 16 : 1. El reloj más lento (*Clock A*) determina la velocidad a la que se producen las conversiones sucesivas y su valor no debe exceder de 250KHz. El reloj más rápido (*Clock B*) se utiliza para hacer la conversión. Estas señales son generadas a partir de un circuito divisor de reloj. La conversión resultante del SAR representa la magnitud leída mediante un bit de signo y 7 bits de magnitud. Los valores en las entradas deben estar limitados en el rango *VMR* (*Voltage Main Reference*, cuyo valor nominal es de 2V) $\pm 1,5V$.

Hay dos posibilidades para dirigir el resultado del ADC SAR, hacia una dirección en memoria de la LUT o de regreso a su propio CAB. El uso más común es el de generar una dirección para la LUT. Después de cada conversión, el resultado de 8 bits es reconocido por la LUT como una nueva dirección. Un circuito de configuración carga el contenido de la dirección apuntada por la LUT en uno o dos registros de la *Shadow SRAM*. Un escenario de uso típico es cuando una señal de entrada requiere ser linealizada o calibrada. La señal externa es convertida por el ADC SAR y enviada a la LUT, lo que apunta a una dirección de la configuración de la imagen del dispositivo que contiene la tabla de linealización. A partir de ello, una nueva configuración sugerida por la LUT se carga en la memoria *Shadow SRAM*, por ejemplo, la ubicación que puede ajustar la ganancia de los OpAmps.

Por otra parte, al direccionar la salida ADC SAR de nuevo a su propio CAB, se activa la generación de funciones analógicas no lineales como la multiplicación de voltajes, la linealización y el control automático de ganancias. Otra alternativa es conectar las salidas del ADC directamente a las celdas de salida, en cuyo caso las interfaces se deben configurar en modo digital. Los datos se envían en serie en una secuencia de 8 bits iniciada por el bit más significativo (**MSB**, *Most Significant Bit*, por sus siglas en inglés).

Tercera generación. **AnadigmApex** representa la tercer generación de procesadores de señal analógica, dispositivos dinámicamente programables (*dynamically programmable Analog Signal Processor*, **dpASP**, por sus siglas en inglés) de Anadigm. Los FPAA AN131E04 y AN231E04 son dos miembros de esta familia, ambos dispositivos tienen 7 celdas I/O analógicas y 4 CABs. Estas estructuras se construyen a partir de la combinación de elementos de circuitos convencionales y SC que son programados desde fuera del chip mediante una memoria no volátil o por un procesador anfitrión. Los arreglos analógicos programables permiten la adaptabilidad y flexibilidad de circuitos analógicos que no eran posibles anteriormente.

La SRAM en los dispositivos AN23x es de reconfiguración dinámica. El comportamiento del dpASP puede ser modificado parcial o totalmente mientras se encuentra en operación. La reconfiguración dinámica permite al procesador central enviar la nueva configuración de los datos al dpASP mientras la configuración previa está ejecutándose. Una vez que la carga de datos se ha completado, la transferencia de la nueva configuración de la señal analógica de

procesamiento ocurre en un único ciclo de reloj.

La reconfiguración dinámica en el dispositivo AN23x permite al usuario desarrollar sistemas analógicos innovadores que se pueden actualizar (total o parcialmente) en la marcha (“*on the fly*”), tantas veces como sea necesario. Los dispositivos AN13x también se basan en la SRAM y pueden ser reprogramados las veces que se deseen, sin embargo, el dispositivo siempre debe ser reseteado antes de cargar una nueva configuración de datos.

La mayor parte del procesamiento de las señales analógicas se produce dentro de los CABs que son fabricados con circuitería SC completamente diferencial. Estos bloques comparten el acceso hacia la LUT que ofrece un método de ajuste a cualquier elemento programable dentro del dispositivo en respuesta a una señal o al tiempo de muestreo. La funcionalidad de la LUT facilita la creación de generadores de forma de onda arbitraria y funciones de transferencia no lineales (*companding*, linealización de sensores). Un generador de voltaje de referencia alimenta a cada CAB del dispositivo excluyendo la necesidad de cualquier circuito externo para el suministro del mismo [17].

El dispositivo AN231E04 (Figura 2.18) es un procesador de señales analógicas, ideal para el acondicionamiento de señales, filtrado, aplicación de ganancias, rectificadores, sumas, restas, multiplicación, entre otros. El dispositivo también permite la adaptación de funciones no lineales como la respuesta de linealización de un sensor y la síntesis de forma de onda arbitraria.

El FPAA consiste en una matriz de CABs de 2x2, rodeada por recursos de interconexión programables y celdas analógicas I/O con elementos activos. Su arquitectura es completamente diferencial con alimentación de 3,3V. El bloque del generador de reloj en el chip controla múltiples relojes en el dominio NOL generados de una fuente externa de reloj estable. La inclusión de una LUT de 8x256 bits permite la síntesis de formas de onda y diversas funciones no lineales. Además, tiene 7 celdas I/O configurables que pueden ser utilizadas como entradas o salidas, 4 de estas tienen integrado un amplificador diferencial. También, tiene un amplificador *Chopper* estabilizado que puede ser utilizado por 3 de las 7 celdas de salida.

El diseño de los circuitos se lleva a cabo mediante el software AD2, una herramienta basada en diagramas de bloques de alto nivel. Las funciones de los circuitos se representan en CAMs, bloques configurables con mapeo sobre porciones de CABs. El software y la tarjeta de desarrollo facilitan el prototipado rápido de cualquier circuito diseñado en esta herramienta.

Dentro de las aplicaciones a las cuales se ha orientado el FPAA AN231E04 se encuentran el procesamiento de señales analógicas, RFID (*Radio Frequency IDentification*), software de control de periféricos de sistemas analógicos en tiempo real, sensores inteligentes, filtrado adaptativo, DSP *front-end* adaptativo, control industrial adaptativo y automatización, sistemas de auto-calibración, compensadores para envejecimiento de componentes de sistema, recalibración dinámica de sistemas remotos y acondicionamiento de señales de frecuencia ultra baja [14, 33].

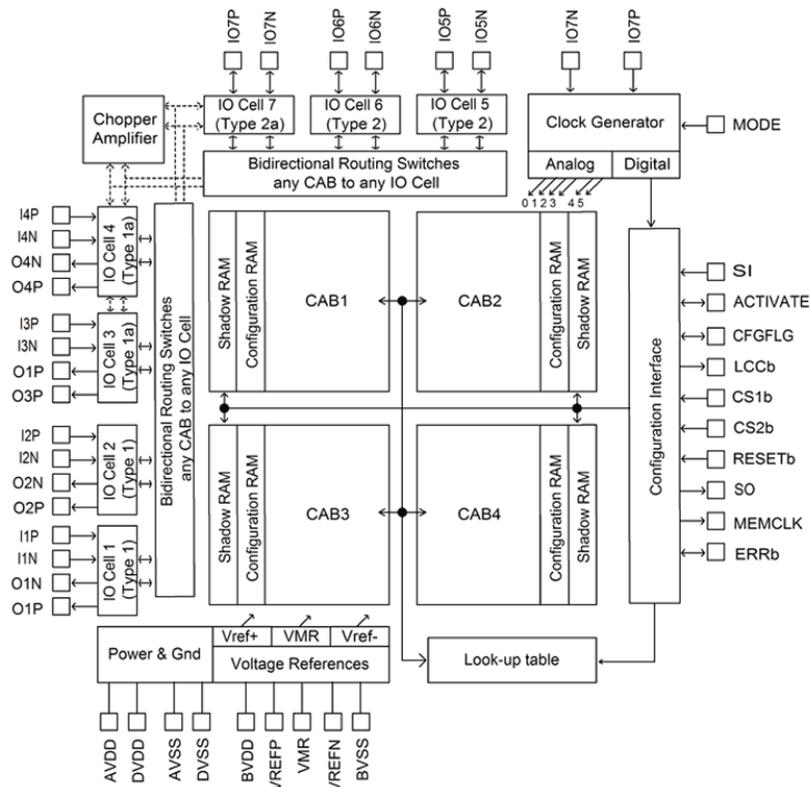


Figura 2.18: Arquitectura del AN231E04, FPAA de 3^{ra} generación, [17]

Teóricamente, los filtros de orden superior se pueden implementar en un dpASP de Anadigm mediante el uso de múltiples dispositivos con 8 polos en cada uno de ellos. Todos los efectos previos son acumulativos, provocando la limitación de los filtros a construir. Es por ello que una “regla de oro” es considerar que un filtro con un orden alrededor de 10 a 12 se puede realizar sin las consideraciones especiales de los efectos secundarios previamente mencionados [10].

2.6.4. El FPAA AN231E04 de Anadigm

El kit de desarrollo AN231K04-DVLP4 (Figura 2.19) es una plataforma de fácil acceso diseñada para ayudar al usuario en la implementación y prueba de circuitos analógicos. Dentro de las características principales se encuentran un dispositivo FPAA AN231E04, interfaz PC serial para la descarga de los circuitos diseñados en AD2, un oscilador externo de 16MHz , 2 circuitos impresos para la configuración de filtros Rauch y pines para la conexión de múltiples tarjetas con la finalidad de evaluar sistemas multi-chip [19, 27].

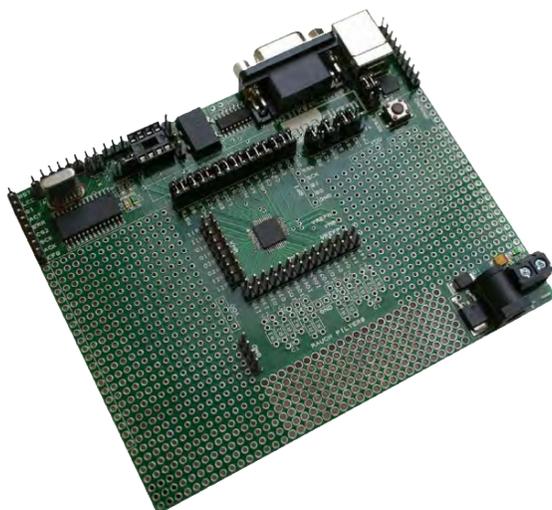


Figura 2.19: Tarjeta de desarrollo con FPAA AN231E04

La interfaz del kit de desarrollo se compone de 2 secciones:

✓ **Analógica.**

En este bloque se implementarán los diseños hechos en AD2, el software de desarrollo proporcionado por Anadigm. Sus componentes se mencionaron en la sección anterior.

✓ **Digital.**

Es la encargada de recibir e interpretar los datos generados en AD2, los cuales permitirán realizar la configuración del bloque analógico para implementar el diseño realizado.

2.6.4.1. Sección digital

La sección digital en el kit de desarrollo provee una interfaz serial de la tarjeta a la PC, la cual permite la configuración directa (prototipado instantáneo) del dpASP hacia AD2, el uso normal del dpASP no requiere de esta interfaz digital, ya que se puede programar directamente desde una interfaz SPI (*Serial Peripheral Interface*, por sus siglas en inglés). Cuando el diseño final de un circuito se encuentra listo para ser implementado, los datos de configuración se almacenan en un microcontrolador [3] o en una EEPROM.

Esta sección se encuentra a lo largo de la parte superior de la tarjeta de desarrollo (Figura 2.20) y está conectada al resto del kit por medio de la configuración del jumper *J5*. Es posible inhibir la sección digital del kit y dejar solo la sección analógica para proveer una interfaz digital externa con los pines situados en el borde de la tarjeta. Además, la sección tiene un tranceptor USB, un tranceptor RS-232, un microcontrolador PIC que permite la conversión serial ASCII a bit y un par de LEDs que indican el estado de la programación.

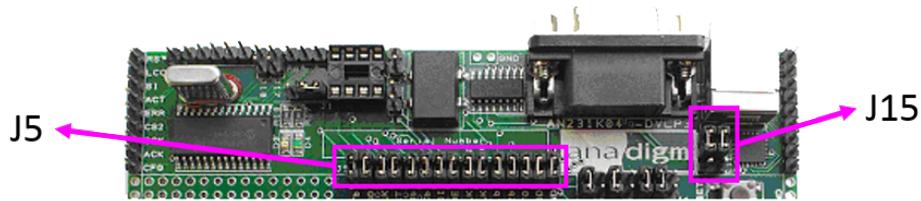


Figura 2.20: Sección digital del kit de desarrollo

Si la sección digital del kit de desarrollo se quita o simplemente es ignorada (quitando los jumper en $J5$), el dpASP se puede configurar directamente utilizando cualquier procesador con una interfaz SPI o con algún puerto configurado con las señales apropiadas. Esto se lleva a cabo mediante la conexión directa de señales al dpASP (área cerca del jumper $J5$) o por la conexión hacia el conjunto de pines marcados como SPI. El control dinámico del dpASP se puede realizar por medio del software a través de esta conexión.

A manera de comentario, Anadigm no recomienda ningún procesador o controlador específico debido a que sus productos funcionan con la mayoría de estos. En base a ello, el desarrollador puede implementar sus diseños a través del procesador de su preferencia y conectarlo a través del jumper $J5$ para tener el control dinámico del dpASP [19, 27].

2.6.5. Programación de un FPAA

Un FPAA se puede configurar en dos modos, mediante el uso del puerto USB y por medio de la interfaz SPI EEPROM. La segunda opción se selecciona cuando el diseño está finalizado (Figura 2.21).

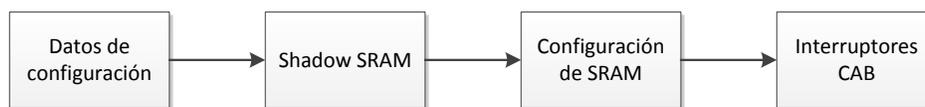


Figura 2.21: Diagrama para la configuración de los datos en un FPAA

El esquema de aplicación es más potente cuando el dpASP está configurado como esclavo de datos en serie. En el modo de operación dinámica con el puerto USB, un procesador principal (PC) envía los datos de configuración al dpASP utilizando el bus SPI compatible, con lo cual, el usuario tiene la facilidad de configurar al FPAA con su diseño para la verificación práctica del mismo durante la fase de desarrollo. Los datos para la configuración se almacenan en una memoria de configuración SRAM que permite cargar una configuración diferente del circuito como una tarea en segundo plano sin interrumpir la funcionalidad del circuito actual. Un microcontrolador PIC embebido en la tarjeta es el interprete de los bits de configuración recibidos de la PC que serán enviados al chip FPAA. El programa cargado en el

PIC, se denomina “núcleo de arranque analógico” (*Analogic Boot Kernel*, **ABK**, por sus siglas en inglés). Este núcleo es un conjunto de requisitos de software y protocolos de comunicación diseñado para permitir a una PC con AD2 programar y controlar el FPAA. El archivo de configuración de datos del dpASP se puede utilizar para programar una SPI PROM para la operación estática independiente o su compilación en un microcontrolador para el modo de operación dinámico a través de código fuente [3, 14].

El diagrama de bloques de la Figura 2.22 muestra el funcionamiento del ABK.

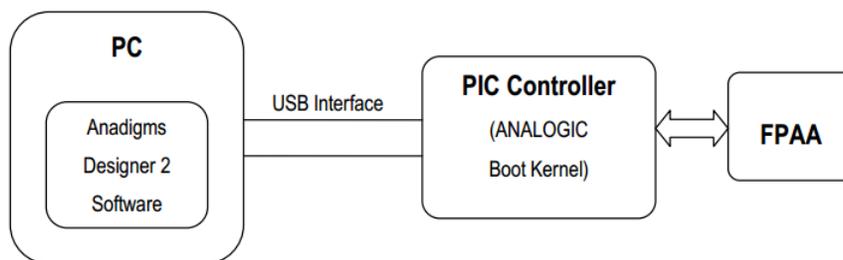


Figura 2.22: Diagrama de bloques del Núcleo de Arranque Analógico

Los FPAA ofrecen a los usuarios la capacidad de reconfiguración dinámica del dispositivo en un sistema. Esta capacidad se deriva de la arquitectura del FPAA AnadigmApex que incluye la memoria de configuración shadow RAM, permitiendo de esta manera que los datos de configuración sean cargados en el FPAA mientras éste se encuentra en funcionamiento, es decir, sin necesidad de apagar el dispositivo. Con esta funcionalidad, el FPAA se puede modificar sobre la marcha (“*on the fly*”) mediante la carga de nuevas configuraciones del dispositivo en tiempo real [52].

El diseño de circuitos para el dpASP se logra usando AD2. Este software presenta un entorno para el diseño de circuitos de manera gráfica en la que los bloques de construcción básicos para el procesamiento de señales analógicas se lleva a cabo al situar los CAM deseados y la conexión entre ellos. Dentro de las operaciones que permiten realizar los CAMs se encuentran las de ganancia, filtros, sumadores y rectificadores. Los parámetros específicos para cada uno de los bloques (valor de la ganancia, frecuencia de corte, polaridad de las entradas, entre otros) son establecidos por el usuario. AD2 genera un archivo de datos de configuración que el dpASP utiliza para programar una SPI PROM para la operación estática independiente. El software de desarrollo también genera código fuente en C para el microcontrolador que permite la generación de nuevos datos de configuración al dpASP sobre la marcha y la posterior reconfiguración dinámica. El comportamiento del procesamiento de las señales se puede ajustar mientras el sistema se mantiene continuamente en operación. El comportamiento de la circuitería analógica es controlado por el contenido en la memoria de configuración (SRAM) del dpASP, la cual debe ser programada después del encendido. La interfaz de configuración se presenta como un puerto esclavo de datos seriales a un microprocesador compatible con señales SPI [17].

La programación del kit de desarrollo se lleva a cabo mediante el uso de la interfaz del puerto serial RS-232 o por el puerto USB.

2.6.5.1. Interfaz RS-232

Una vez que el kit de desarrollo se ha encendido, basta con conectarlo al puerto serie de una PC mediante un cable RS-232 estándar. Se debe verificar que la posición de los jumper en *J15* estén situados en la parte superior de los pines (ver Figura 2.20). Una vez que el diseño de determinado circuito ha sido hecho en AD2, se procede a verificar que el puerto serial haya sido detectado por AD2 para poder descargar el diseño hacia el dpASP. Concluida esta acción, si el LED ubicado junto al PIC en la sección digital verde se ilumina, indica que la programación se ha efectuado de manera correcta, en caso contrario, si el LED rojo se ilumina implica que la configuración falló. Si esto sucede, se debe comprobar que la alimentación del kit en los pines marcados con VDD sea de +3,3V. Si la configuración se ha realizado correctamente, entonces el circuito creado en AD2 se programará en el dpASP. Las entradas y salidas analógicas se puede acceder a través de los pines que rodean al FPAA. Se debe tener en cuenta que los pines externos están conectados a las E/S analógicas del dpASP, mientras que los pines internos están todos conectados a VMR, una conexión a la tierra de la señal analógica (+1,5V).

2.6.5.2. Interfaz USB

El kit de desarrollo AN231K04-DVLP3 utiliza el controlador “USB to UART Bridge” de la compañía *Silicon Labs*. La tarjeta se conecta mediante un cable USB estándar, que se instala como un puerto USB tradicional e internamente es reconocido por la PC y el software como un puerto serial. La posición de los jumper en *J15* debe ser en la parte inferior de los pines (ver Figura 2.20).

Debido a que la interfaz serial tiene un interprete para las instrucciones en modo serial, se debe instalar el driver correspondiente antes de poner en marcha el kit de desarrollo. Para ello, se considera que AD2 ha sido instalado en la PC. Posteriormente, se instala el driver “CP210x” de Silicon Labs que se encuentra disponible en el CD incluido en el kit de desarrollo o el centro de descargas de Anadigm (http://www.anadigm.com/sup_downloadcenter.asp?tab=des&offset=15).

Una vez energizado el kit de desarrollo (+3,3V), se debe verificar dentro de la interfaz de AD2 que el puerto haya sido reconocido. En la lista desplegable de la pestaña “Port” del menú “Settings/Preferences...” se debe seleccionar el puerto COM asignado (“COM* - CP2101 USB to UART Bridge Controller”, donde * es el número de puerto asignado por la PC a esta interfaz). Realizada esta configuración, se puede utilizar la interfaz USB para la programación del FPAA sin problema alguno [27].

2.7. Módulo de temperatura LTR701



Figura 2.23: Módulo térmico LTR701

Como se muestra en la Figura 2.24, el sistema de control de flujo de aire y temperatura LTR701 cuenta con 5 variables de salida analógica (posición del acelerador (4), temperatura (7, 8), presión (9) y velocidad del flujo de aire (10)) y 2 variables de entrada analógica (velocidad del ventilador (5) y salida del calefactor (6)), con las cuales se permite examinar las configuraciones de los controladores empleados para el control del sistema [48].

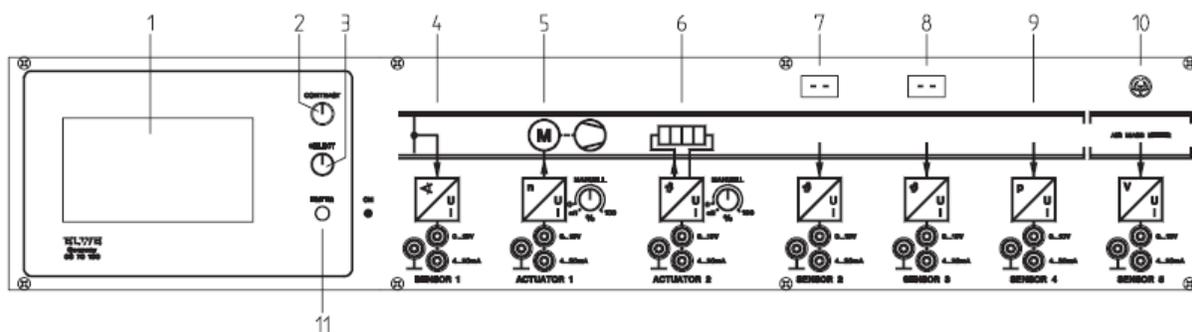


Figura 2.24: Panel Frontal del LTR701

El módulo cuenta con un termopar tipo K que permite la medición de la temperatura. Para poder activar el calefactor mediante incrementos de voltaje, es necesario que el ventilador este operando a un 10% de su potencia, en caso contrario, el calefactor estará apagado. Hay que tener en cuenta que este dispositivo está protegido contra sobrecalentamiento por medio de un interruptor térmico ($\gamma = 140^{\circ}C$). Además, la corriente es medida y limitada por el sistema.

Las Tablas 2.5 y 2.6 muestran las señales que se pueden medir y aplicar a la planta de temperatura LTR701.

Tabla 2.5: Señales de Entrada

Actuador	Señal de actuación	Voltaje	Corriente
1	Velocidad del Ventilador	0 - 10 V	4 - 20 mA
2	Salida del Calefactor		

Tabla 2.6: Señales de Salida

Sensor	Parámetro de Medición	Voltaje	Corriente
1	Ángulo del Regulador	0 - 10 V	4 - 20 mA
2	Temperatura		
3			
4	Presión		
5	Flujo de Aire		

Si se desea incrementar la potencia del calefactor o del ventilador, se debe suministrar incrementos de voltaje a $1V$ para elevar el 10% de la potencia de los mismos, en caso de suministrar corriente, habrá que aplicar el 6,25% de la potencia para incrementar $1mA$. Cuando se está efectuando la medición de temperatura, ya sea en el sensor 2 o 3, se presenta un cambio en su medición en el momento en que hay un incremento de $10^{\circ}C/V$ o de $6,25^{\circ}C/mA$.

Cuando al sistema se conectan controladores externos (controladores industriales o una PC con la interfaz apropiada), las siguientes opciones de control pueden ser configuradas con el LTR701.

- ✓ Lazos de control simple
 - Control de flujo de aire con la temperatura del aire no controlada.
 - Control de la temperatura del aire afectando la salida del calefactor con flujo de aire no controlado.
 - Control de la temperatura del aire afectando el flujo de aire con la salida al calefactor no controlada.
- ✓ Control en cascada
 - Control de la temperatura del aire, el flujo de aire es la variable secundaria controlada mediante el ajuste de la salida del calefactor.
- ✓ Sistemas de control multi-variable

- Dos lazos de control independientes: temperatura y flujo de aire, puede ser desacoplados.

2.8. Estado del arte

Una vez concluida la introducción al presente trabajo y a los elementos que serán utilizados en su implementación, es obligatorio realizar un recuento de aquellos trabajos, existentes en la literatura, que guardan una estrecha relación con el presentado en esta investigación.

El trabajo propuesto por Lita *et al.* [51] presenta una aproximación para la implementación de un controlador PID usando un FPAA aplicado al control de temperatura en el dominio nuclear. Esta investigación muestra que el uso de un circuito FPAA para la implementación del controlador PID, permite un ajuste más fácil que el diseño analógico clásico.

En [61] Visan *et al.* desarrollan un algoritmo de control para un PID reconfigurable en un FPAA aplicado a un equipo electrónico de temperatura. Los parámetros del controlador PID se ajustan de manera constante durante la operación del sistema. La lectura de la temperatura se realiza con el sensor LM35Z conectado al controlador a través de un circuito de acondicionamiento implementado también en el FPAA. Para llevar a cabo estas acciones, se utilizó el dispositivo FPAA AN221E04, mismo que permite la reconfiguración dinámica del sistema a través de la interfaz serial con una PC.

En el proyecto presentado en [42], Delgadillo y Piraján plantean como objetivo incursionar en el área de la instrumentación industrial moderna mediante el uso de FPAA para el acondicionamiento de sensores en cooperación con procesadores de señal mixta PSoC (*Programmable System on Chip*). Como principal argumento de los beneficios otorgados por esta unión de tecnologías, los autores mencionan que mediante el acoplamiento de estas plataformas se tiene la posibilidad de medición y control centralizado y descentralizado, incursionando a lo que hoy día se conoce como CADSP (*Cooperative Analog-Digital Signal Processing*).

En el trabajo presentado en [54] se desarrolla una interfaz llamada MIDT (*Motorola Interactive Design Tool: Herramienta de Diseño Interactivo Motorola*) que permite el cableado virtual (*drag & drop*) y elimina los requerimientos para la vinculación de los componentes de hardware, con el objetivo de proveer a los estudiantes una manera fácil y rápida de construcción de hardware complejo, y permitirles observar la relación entre el procesamiento de señales analógicas y digitales.

Pierzchala *et al.* muestran como un FPAA de propósito general se puede usar para la implementación de diversos circuitos lógicos. En su investigación, comprueban que un FPAA provee la oportunidad de experimentar con hardware realizando varios circuitos lógicos sin la necesidad de diseñarlos y fabricarlos. El FPAA utilizado en este trabajo se basa en la implementación de la tecnología de un arreglo de transistores bipolares [55].

La comparación presentada en [57] de las tecnologías FPAA y FPGA para circuitos reconfigurables muestra un caso de estudio de control industrial, el cual consiste en la configuración clásica compuesta por un proceso de segundo orden y un controlador PID. La diferencia de la implementación en un FPGA con respecto al FPAA es el uso de la transformada Z, lo cual requiere de una conversión en el primero tanto para la entrada como la salida. Los resultados muestran que la aproximación al modelo matemático de la planta es más precisa con el uso de un FPAA que de un FPGA.

En el desarrollo de [60] se presenta la implementación de un sistema multi-canal para el procesamiento de señales sísmicas mediante un FPAA en el área de la tomografía acústica de suelos. El circuito diseñado consta de 4 bloques básicos: un amplificador de radiofrecuencia con ajuste programable para la ganancia de 0 - 60 dB con un ancho de banda de $3KHz$ a $11KHz$, un filtro pasabanda, un rectificador de onda completa y un ADC. El desarrollo del sistema electrónico propuesto permite la emulación en tiempo real, permitiendo ajustes de parámetros y modificaciones en la estructura que adecuen el sistema a las necesidades que mejor se adapten a la aplicación.

Capítulo 3

Modelado y simulación del sistema de control de temperatura

En esta sección se describe el desarrollo realizado para el diseño de los controladores PID. Se presenta el procedimiento llevado a cabo para obtener el valor de las ganancias de los controladores con base en los parámetros presentes en la función de transferencia que modela el comportamiento de la planta de temperatura (LTR701) obtenido del manual de la misma. Una vez obtenidos los valores correspondientes, se procede a la simulación de los controladores para observar su comportamiento en lazo cerrado con la planta de temperatura, dicha simulación se efectuó en el software propio del FPAA (AD2) y MATLAB Simulink[®].

3.1. Diseño de controladores PID

Para el cálculo de los valores de las ganancias requeridas para los controladores PID a implementar, se hizo uso de las reglas de sintonización de Ziegler-Nichols, para lo cual se tomó de referencia la Tabla 2.1.

Con base en el manual del módulo de temperatura a controlar (LTR701), se tiene que la Ecuación (3.1) modela el comportamiento del sensor que mide la temperatura mediante un termopar de tipo K proveniente de la planta (SENSOR 3). Dicha función de transferencia es un sistema de primer orden con retardo.

$$G(s) = \frac{0,485}{12,6s + 1} e^{-0,8s} \quad (3.1)$$

Para la sintonización de las ganancias con el método de Ziegler-Nichols, es necesario comparar las Ecuaciones 3.1 y 2.13, con la finalidad de obtener el valor de los parámetros requeridos. Al efectuar dicho proceso, se obtienen los valores de la ganancia estática del sistema ($k = 0,485$), la constante de tiempo ($T = 12,6$) y el tiempo de retardo ($L = 0,8$), con los cuales se procede a efectuar la sintonización de los controladores.

3.1.1. Controlador Proporcional

Para el diseño del controlador P se requiere del cálculo de una ganancia proporcional, misma que se obtendrá a partir de la Ecuación (3.2).

$$K_p = \frac{T}{L} \quad (3.2)$$

Tomando en cuenta los valores de los parámetros T y L , se realiza el cálculo de la ganancia proporcional.

$$\begin{aligned} \Rightarrow K_p &= \frac{12,65}{0,8} \\ &= 15,75 \end{aligned}$$

Una vez que se obtiene el valor de este parámetro, se sustituye en la Ecuación (3.3) que describe la función de transferencia correspondiente al controlador P.

$$G_P(s) = K_p \quad (3.3)$$

$$\therefore G_P(s) = 15,75$$

3.1.2. Controlador Proporcional-Integral

Para el cálculo de las ganancias del controlador PI se requiere de 2 parámetros, la ganancia proporcional (K_p) y el tiempo integral (T_i). Tomando en cuenta los valores T y L de la Ecuación (3.1) se efectúan los siguientes cálculos.

$$\begin{aligned} K_p &= 0,9 \frac{T}{L} & T_i &= \frac{L}{0,3} \\ &= 0,9 * \frac{12,65}{0,8} & &= \frac{0,8}{0,3} \\ &= 14,175 & &= 2,667 \end{aligned}$$

Al sustituir los valores de los parámetros obtenidos en la Ecuación (3.4), se obtiene la función de transferencia correspondiente al controlador PI.

$$\begin{aligned}
 G_{PI}(s) &= K_p \left(1 + \frac{1}{T_i s} \right) & (3.4) \\
 \Rightarrow G_{PI}(s) &= 14,175 \left(1 + \frac{1}{2,667s} \right) \\
 &= 14,175 + \frac{14,175}{2,667s} \\
 \therefore G_{PI}(s) &= 14,175 + \frac{5,315}{s}
 \end{aligned}$$

3.1.3. Controlador Proporcional-Integral-Derivativo

Para la sintonización de las ganancias del controlador PID se requiere del cálculo de 3 parámetros, la ganancia proporcional (K_p), el tiempo integral (T_i) y el tiempo derivativo (T_d). Sustituyendo los valores en las ecuaciones correspondientes se realiza el cálculo de los parámetros requeridos.

$$\begin{aligned}
 K_p &= 1,2 \frac{T}{L} & T_i &= 2L & T_d &= 0,5L \\
 &= 1,2 * \frac{12,65}{0,8} & &= 2 * 0,8 & &= 0,5 * 0,8 \\
 &= 18,9 & &= 1,6 & &= 0,4
 \end{aligned}$$

Al sustituirlas en la Ecuación (3.5), se obtiene la función de transferencia del controlador PID.

$$\begin{aligned}
 G_{PID}(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) & (3.5) \\
 \Rightarrow G_{PID}(s) &= 18,9 \left(1 + \frac{1}{1,6s} + 0,4s \right) \\
 &= 18,9 + \frac{18,9}{1,6s} + 18,9 * 0,4s \\
 \therefore G_{PID}(s) &= 18,9 + \frac{11,813}{s} + 7,56s
 \end{aligned}$$

3.1.4. Controlador Proporcional-Derivativo

Para la sintonización del controlador PD, se considera el tiempo integral del diseño del controlador PID como infinito ($T_i = \infty$), y se sustituyen los valores de la ganancia proporcional (K_p) y el tiempo derivativo (T_d), como se presenta a continuación.

$$\begin{aligned}
 G_{PD}(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\
 &= K_p \left(1 + \frac{1}{\infty} + T_d s \right) \\
 &= K_p (1 + 0 + T_d s) \\
 \Rightarrow G_{PD}(s) &= K_p (1 + T_d s) \\
 &= 18,9 (1 + 0,4s) \\
 &= 18,9 + 18,9 * 0,4s \\
 \therefore G_{PD}(s) &= 18,9 + 7,56s
 \end{aligned}$$

3.2. Simulación de los controladores

Una vez realizada la sintonización para los controladores PID y con la finalidad de observar el comportamiento del sistema ante el control propuesto, se procede a efectuar la simulación de estos en lazo cerrado con el modelo de la planta de temperatura en uso. Para ello, se hizo uso de la herramienta MATLAB Simulink[®] así como el software que proporciona el fabricante de la tarjeta FPAA, AD2.

3.2.1. MATLAB Simulink[®]

En primer instancia, se hace uso del entorno de MATLAB Simulink[®]. El cual permite la simulación de sistemas de control mediante diagramas de bloques, dicho proceso se basa en la definición del comportamiento de estos sistemas mediante funciones de transferencia, operaciones matemáticas y señales de diferente tipo.

Con la finalidad de analizar el comportamiento de los controladores PID diseñados, se considera un *setpoint* de $3,5V$ equivalente a $35^{\circ}C$ requeridos para estabilizar el sistema de temperatura. Para la simulación de dicho sistema dentro de Simulink, se requiere de bloques de ganancias, puntos de suma, función de transferencia, retardo y respuesta al escalón.

3.2.1.1. Controlador Proporcional

La simulación del controlador P se efectúa en base el diagrama mostrado en la Figura 3.1, para el cual, la respuesta al escalón de $3,5V$ se muestra en la Figura 3.2, donde se observa que el sistema no alcanza la estabilidad al *setpoint* requerido y se tiene que el error en estado estacionario es de aproximadamente $11,6\%$ del valor deseado. Lo anterior es una situación normal derivado del hecho de que el sistema de temperatura está siendo modelado con base en un modelo de primer orden con retardo y por tanto un sistema tipo cero.

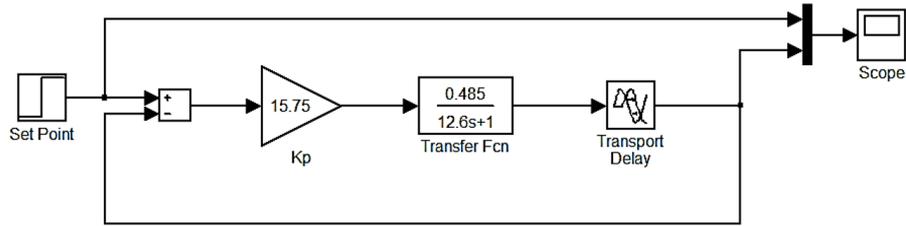


Figura 3.1: Diagrama de bloques del controlador P

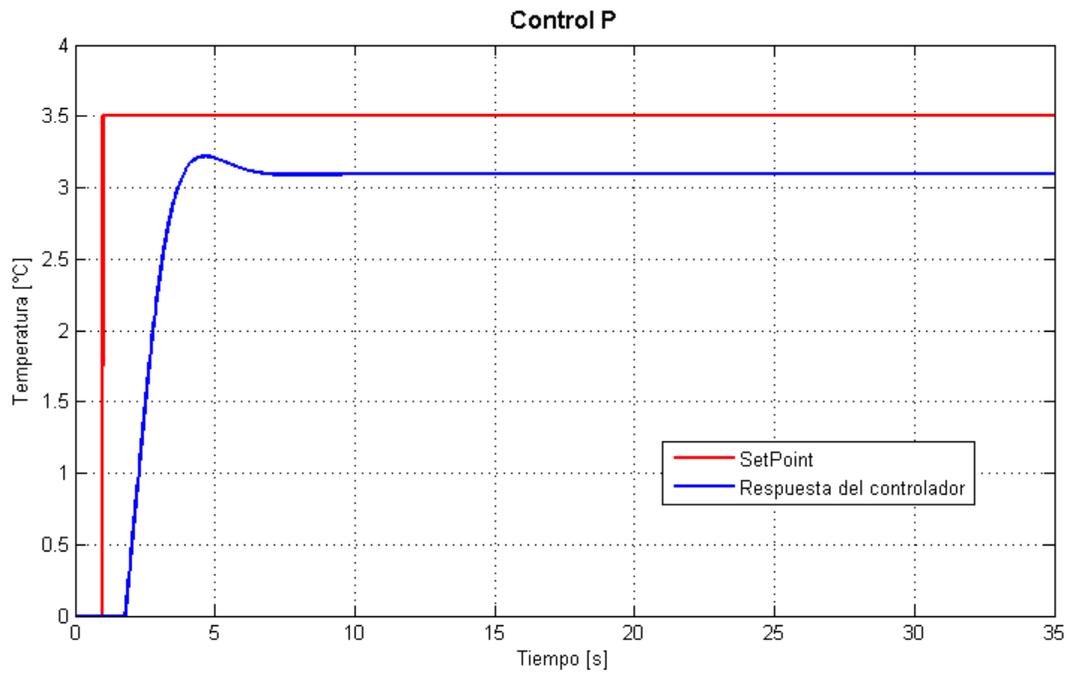


Figura 3.2: Respuesta del controlador P

3.2.1.2. Controlador Proporcional-Derivativo

Los resultados de simulación (Figura 3.4) del sistema mostrado en Figura 3.3, correspondiente al controlador PD, muestra que el error en estado estacionario obtenido es reducido a 9,84% del valor deseado, con la adición de la acción de control derivativa al controlador proporcional, se observa además que desaparece el pequeño sobretiro mostrado en la Figura 3.2, y que la estabilidad del sistema se alcanza en un tiempo menor al controlador P.

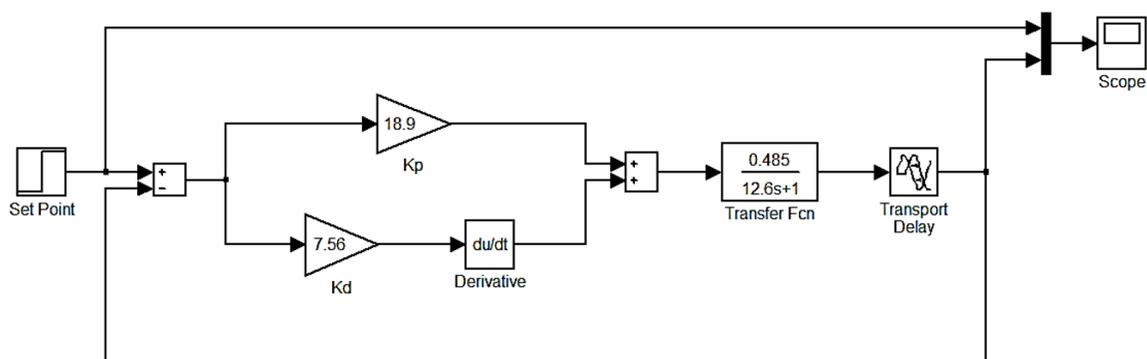


Figura 3.3: Diagrama de bloques del controlador PD

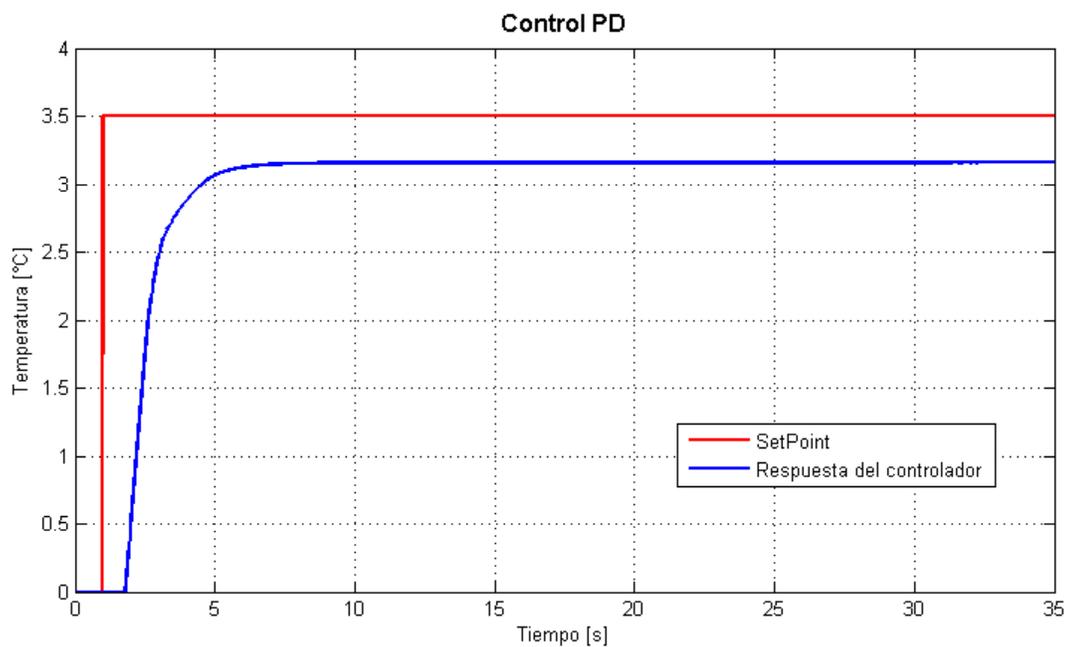


Figura 3.4: Respuesta del controlador PD

3.2.1.3. Controlador Proporcional-Integral

En el caso de la simulación del controlador PI (Figura 3.5), se observa que el comportamiento del sistema presenta una sobreimpulso de 38,9% y se estabiliza al valor deseado presentando una inestabilidad mínima (ver Figura 3.6).

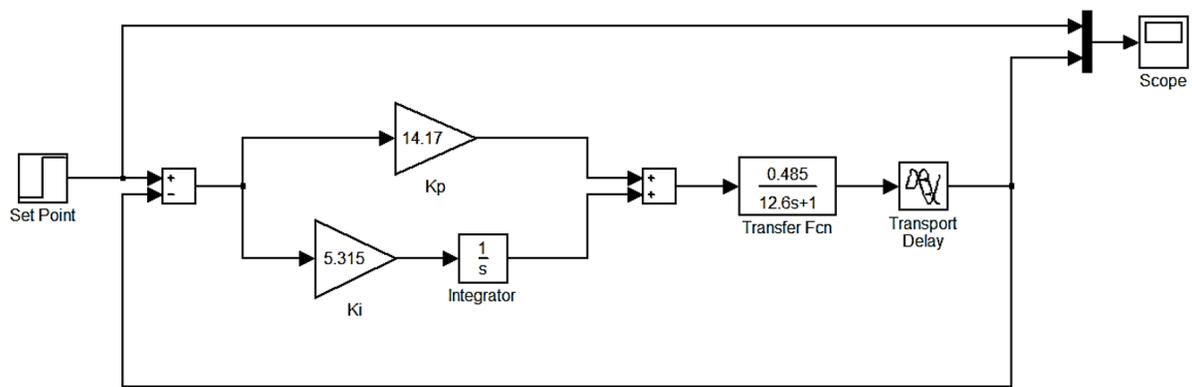


Figura 3.5: Diagrama de bloques del controlador PI

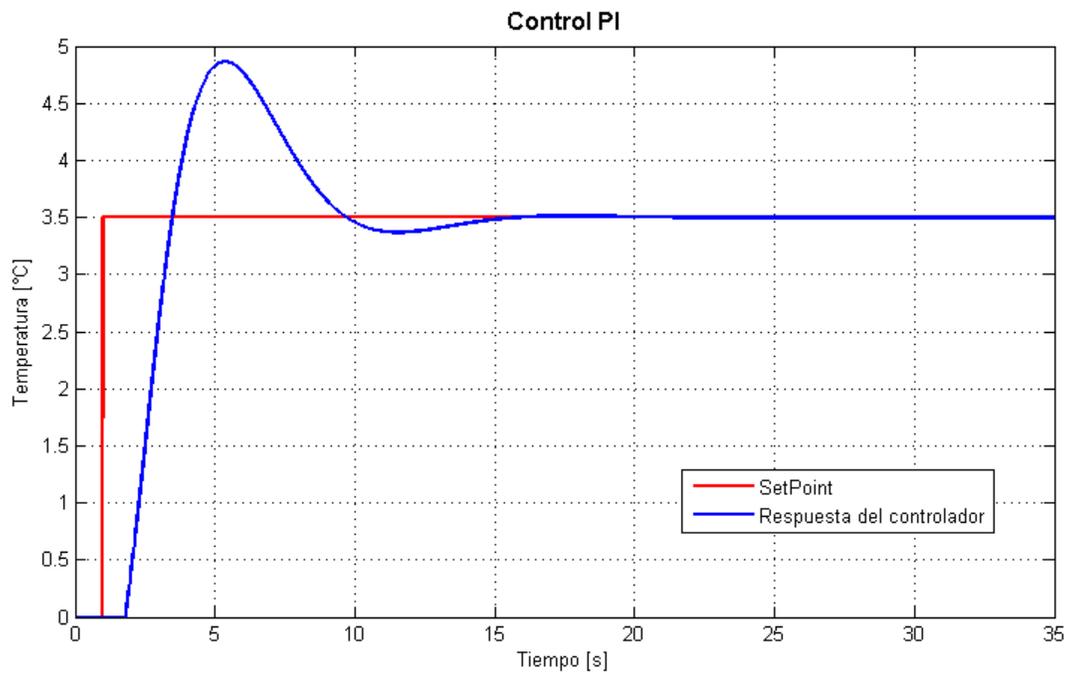


Figura 3.6: Respuesta del controlador PI

3.2.1.4. Controlador Proporcional-Integral-Derivativo

Al simular el controlador PID (Figura 3.7), se observa que la respuesta del sistema (Figura 3.8) presenta un sobretiro de aproximadamente 54%.

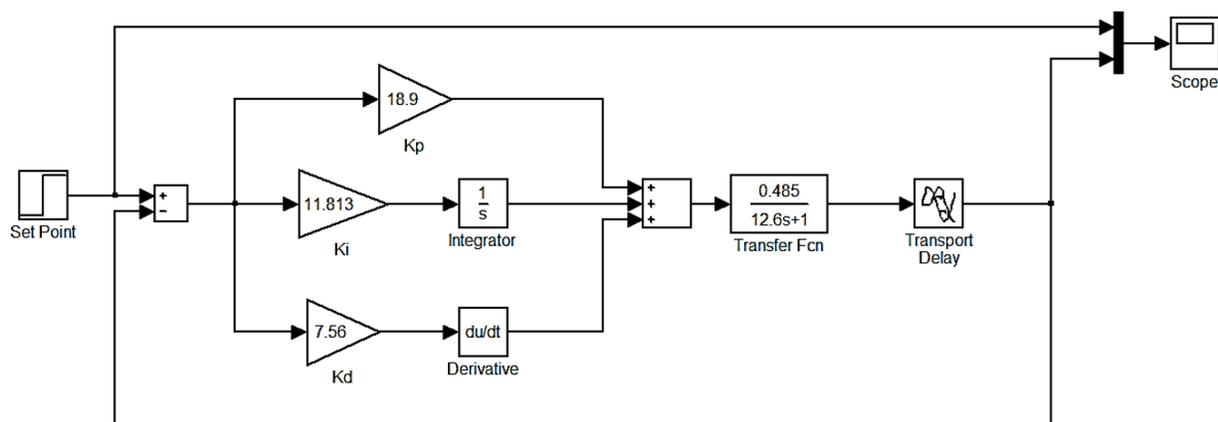


Figura 3.7: Diagrama de bloques del controlador PID

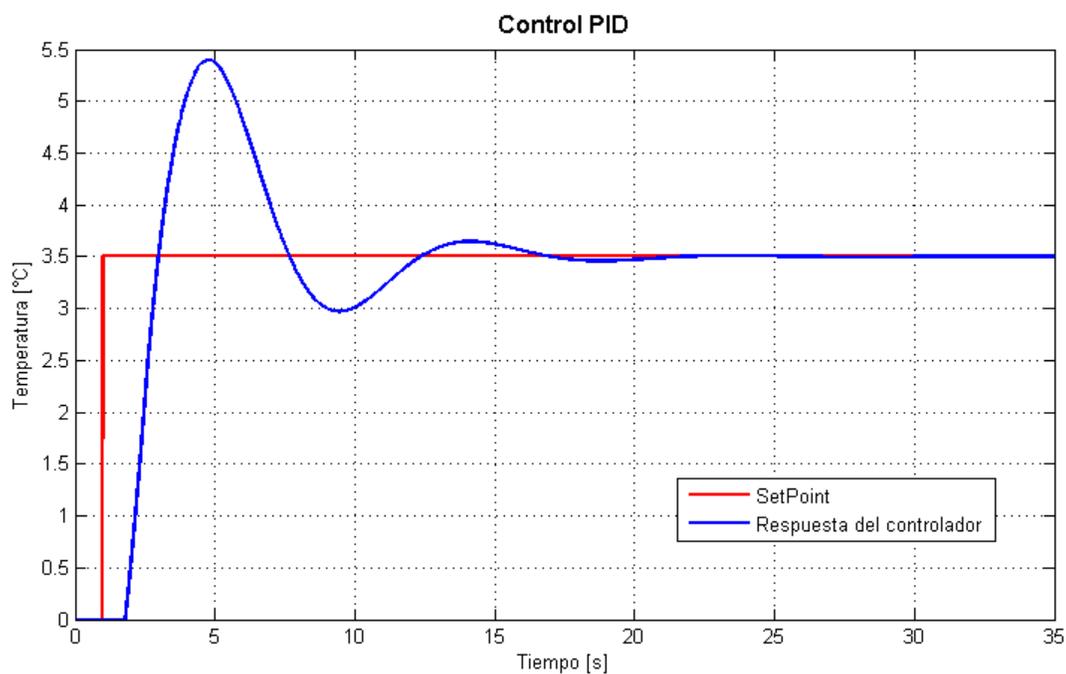


Figura 3.8: Respuesta del controlador PID

El error en estado estacionario de los controladores PI y PID se vuelve cero.

3.2.2. AnadigmDesigner2[®]

La compañía Anadigm provee un entorno, denominado AD2 [14], que permite al usuario modelar, simular y configurar sus diseños analógicos en un FPAA específico, con dicho entorno, el usuario puede re-configurar al FPAA en tiempo de operación, es decir, se pueden

efectuar cambios en los parámetros del circuito y volver a configurarlo sin necesidad de detener el funcionamiento del sistema. Bastará con arrastrar y cablear los CAMs correspondientes para la creación de un circuito.

El FPAA usado en el sistema de control de temperatura es el **AN231E04**, el cual permite la manipulación de señales de $3,3V$ en sus pines de entrada y salida. Con base en las condiciones antes citadas referentes a la planta de temperatura en uso (LTR701), se adecuan las señales en una relación de $10 : 3$ para las señales de entrada y de $3 : 10$ para las de salida, es decir, un voltaje de $10V$ equivaldrá a $3V$ según el acondicionamiento propuesto para la tarjeta.

3.2.2.1. Modelado de la planta de temperatura

Cabe destacar que la función de transferencia del LTR701 corresponde a un sistema de primer orden (ver Ecuación (3.1)). Para el caso de la simulación de dicho sistema, se observa que dentro del entorno de AD2 no hay un bloque que permita simular la Ecuación (3.1) como tal. Para ello y tomando en cuenta que la programación de los CAMs dentro del FPAA se basa en la técnica de capacitores conmutados con OpAmps, se describe el funcionamiento de un sistema de primer orden implementado con dicha técnica.

Considerando la configuración de un filtro pasa-bajas de primer orden (Figura 3.9) y la función de transferencia correspondiente (Ecuación (3.6)), se obtendrá la expresión necesaria para el modelado de la planta de temperatura.

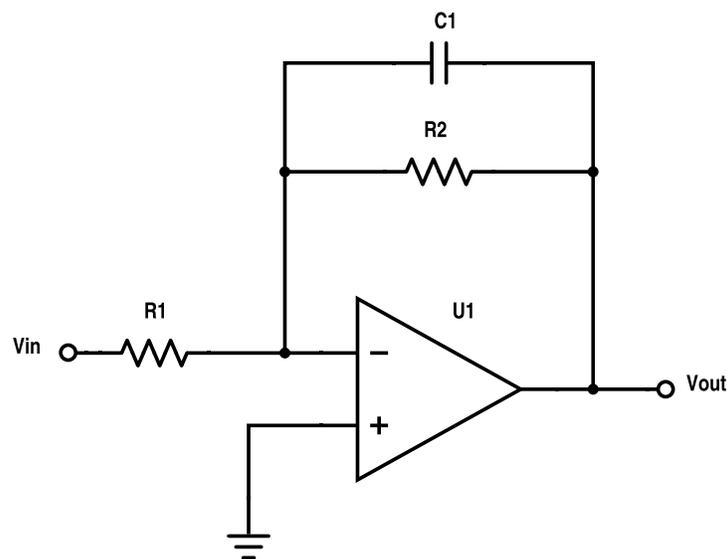


Figura 3.9: Sistema de primer orden. Filtro pasa-bajas

$$G(s) = \frac{V_0(s)}{V_{in}(s)}$$

$$G(s) = -\frac{1}{R_1 \left(\frac{1}{R_2} + C_1 s \right)} \quad (3.6)$$

$$= -\frac{1}{R_1 \left(\frac{1+R_2 C_1 s}{R_2} \right)}$$

$$= -\frac{1}{\frac{R_1}{R_2} (1 + R_2 C_1 s)}$$

$$= -\frac{\frac{R_2}{R_1}}{1 + R_2 C_1 s}$$

$$G(s) = -\frac{\frac{R_2}{R_1}}{R_2 C_1 s + 1} \quad (3.7)$$

A partir del inconveniente antes citado, el re-ordenamiento de la Ecuación (3.6) a Ecuación (3.7) y tomando en cuenta que el FPAA incluye 2 CAMs para filtrado, el filtro bilineal (*FilterBilinear*) y el filtro bicuadrático (*FilterBiquad*), mismos que permiten la implementación de filtros de 1er y 2do orden respectivamente, se presenta la alternativa para el modelado.

Con la ayuda de los CAMs antes mencionados, se implementa la función de transferencia que modela al LTR701 (Ecuación (3.1)). Para este caso, se requiere el uso del filtro bilineal, ya que su implementación con OpAmps presenta un comportamiento similar a la planta en uso. Al revisar la documentación correspondiente al mismo, se observa que la ecuación característica que modela su comportamiento está dada por la Ecuación (3.8).

$$\frac{V_{out}(s)}{V_{in}(s)} = \pm \frac{2\pi f_0 G}{s + 2\pi f_0} \quad (3.8)$$

Donde G es la ganancia pasa-banda y f_0 es la frecuencia de corte a la cual la ganancia es igual a $-3 + 20\text{Log}(G)$.

$$\frac{V_{out}(s)}{V_{in}(s)} = \pm \frac{2\pi f_0 G \frac{1}{2\pi f_0}}{\frac{1}{2\pi f_0} (s + 2\pi f_0)}$$

$$= \pm \frac{G}{\frac{1}{2\pi f_0} s + 1} \quad (3.9)$$

Reescribiendo la Ecuación (3.8) se obtiene la Ecuación (3.9), esto con la finalidad de compararla con los parámetros de las Ecuaciones (3.1) y (3.7).

Para el cálculo de la frecuencia de corte (f_0) se considera la siguiente igualdad:

$$\frac{1}{2\pi f_0} = R_2 C_1 = 12,6$$

$$\Rightarrow f_0 = \frac{1}{2\pi * 12,6} = 12,63mHz$$

Mientras que para el valor de la ganancia (G) se tiene:

$$G = \frac{R_2}{R_1} = 0,485$$

$$\Rightarrow G = 0,485$$

El objetivo de la comparación previa permite definir el valor de los parámetros requeridos para la implementación del filtro bilineal dentro del entorno AD2, dichos valores se aprecian en la Figura 3.10, en la que además se observa el diagrama correspondiente al filtro del CAM antes citado, y las opciones para la modificación de la frecuencia a la cual conmutaran los capacitores del CAM al ser programado sobre el FPAA, éstas son el tipo de filtro (pasa-bajas, pasa-altas, pasa-todo y, polos y ceros) usado, la fase de entrada para el muestreo (la cual permite la conmutación de acuerdo a la señal proveniente del CAM previo dentro de un sistema), la polaridad, los recursos usados para su programación y la habilitación de un amplificador *Chopping*.

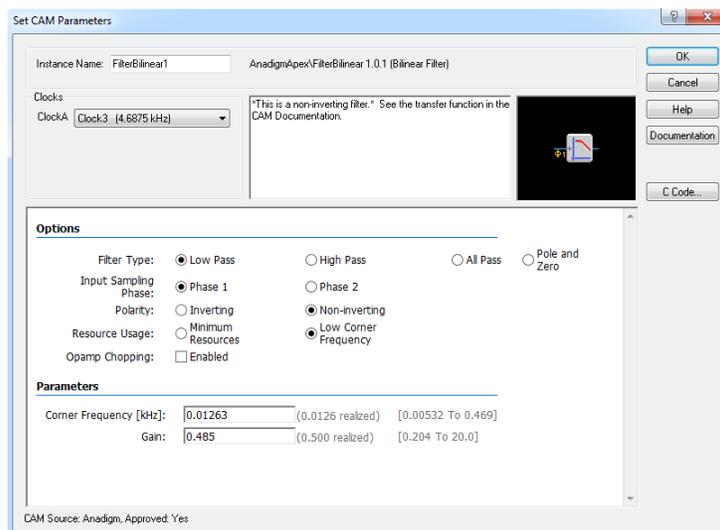


Figura 3.10: Cuadro de diálogo para la configuración del filtro bilineal

Para poder definir el valor de los parámetros del CAM usado en el filtro, se ajustó la frecuencia de trabajo de los FPAA a $300KHz$, lo que se logra dando doble clic sobre el chip y modificando el valor del parámetro $ACLK (f_c)$. Posteriormente, se añade el CAM configurado

para simular la planta de temperatura dentro del chip correspondiente (Figura 3.11). En este caso se habilitan 2 celdas en modo bypass, una configurada como entrada y otra como salida.

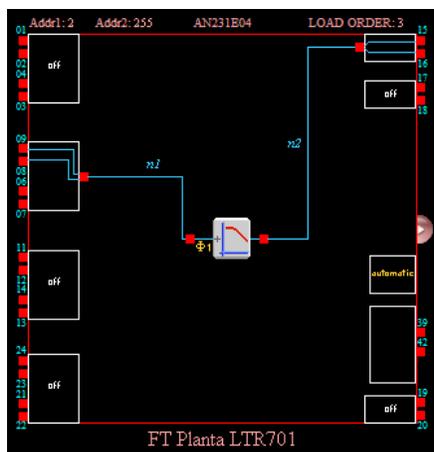


Figura 3.11: CAM para la simulación de la función de transferencia del LTR701

Dentro del entorno de simulación, se considera el uso de 2 chips FPAA, el primero de ellos para el diseño del controlador y el segundo para la planta de temperatura, esto con la finalidad de poder programar en el FPAA solo el controlador cuando se desee efectuar algún ajuste de ganancias y adaptarlo al sistema físico durante su implementación.

A continuación se analizan los resultados de simulación obtenidos para el sistema de control de temperatura modelado con AD2. Se considera al igual que en el caso de la simulación en el entorno de MATLAB Simulink®, el valor del *setpoint* de $3,5V$ ($35^{\circ}C$), para lo cual se realiza un ajuste en base al acondicionamiento propuesto para las señales manipuladas en el FPAA, es decir, habrá que obtener el equivalente de $3,5V$ en la escala de $10 : 3$, por lo que se tiene:

$$\begin{aligned}
 SP_{FPAA} &= \frac{3}{10} * SP \\
 &= \frac{3}{10} * 3,5V \\
 \Rightarrow SP_{FPAA} &= 1,05V
 \end{aligned}$$

El valor obtenido será la referencia de entrada al FPAA, mismo que se simula con un generador de funciones configurado de manera que proporcione una señal cuadrada con una frecuencia de $1Hz$ y una amplitud de $1,05V$.

Los CAMs utilizados para la simulación del sistema son: sumadores (*SumDiff*), ganancias (*Gain*), muestreo y retención (*Hold*), integradores (*Integrator*), derivadores (*Differentiator*) y

filtro bilineal (*FilterBilinear*).

Para el chip utilizado en la implementación de los controladores PID se tienen activas 3 celdas, 2 de las cuales son definidas como entradas, mismas que representan el punto de suma del sistema de control, en este caso se tiene el valor deseado (*setpoint*) y el valor de retroalimentación y, la celda restante se declara como salida, la cual permite observar el comportamiento del controlador implementado. La frecuencia de trabajo de este chip es la misma que se define para la implementación del filtro bilineal.

El tiempo de simulación considerado es de $500ms$, el equivalente a la mitad del periodo de la señal designada como *setpoint*. Las gráficas obtenidas (Figuras 3.13, 3.15, 3.18 y 3.21) muestran la respuesta del controlador implementado (*Channel 2*), el error obtenido (*Channel 4*) y el valor deseado para el sistema (*Channel 1*).

3.2.2.2. Controlador Proporcional

El modelado del controlador P implementado en AD2 (ver Figura 3.12), brinda la respuesta de simulación al escalón de $1,05V$ mostrada en la Figura 3.13, donde se observa que el sistema no se estabiliza al valor de temperatura especificado en el *setpoint* y se tiene que el error en estado estacionario es de aproximadamente 4% del valor deseado. Se observa que en aproximadamente $16ms$ el sistema comienza a estabilizarse a un valor aproximado de $0,93V$.

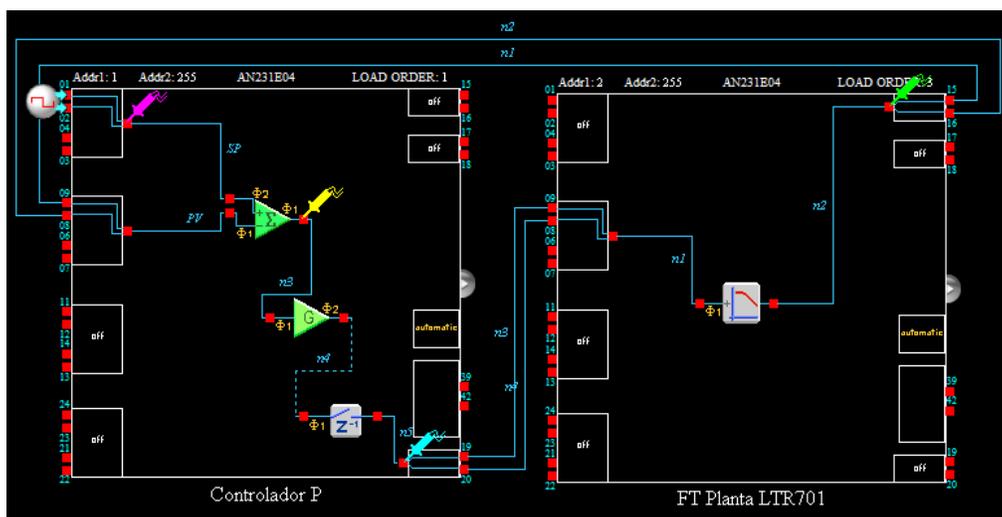


Figura 3.12: Modelado del controlador P en la herramienta AD2

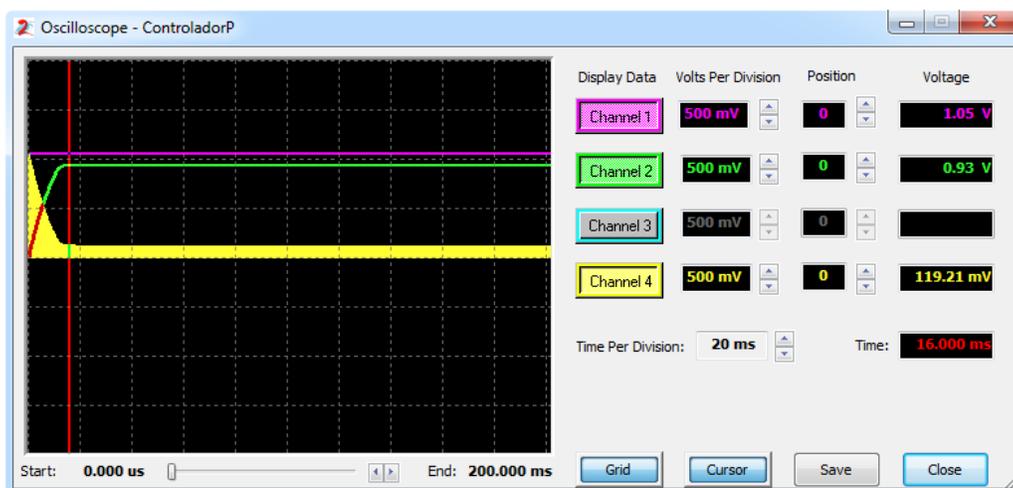


Figura 3.13: Simulación del controlador P en la herramienta AD2

3.2.2.3. Controlador Proporcional-Derivativo

La Figura 3.14 muestra el diseño del controlador PD implementado dentro del FPAAs. El resultado de la simulación del sistema (Figura 3.15), muestra que el comienzo de la estabilidad de la temperatura se da en un tiempo aproximado de $20,5\text{ms}$, y que el error en estado estacionario obtenido es de $10,7\%$.

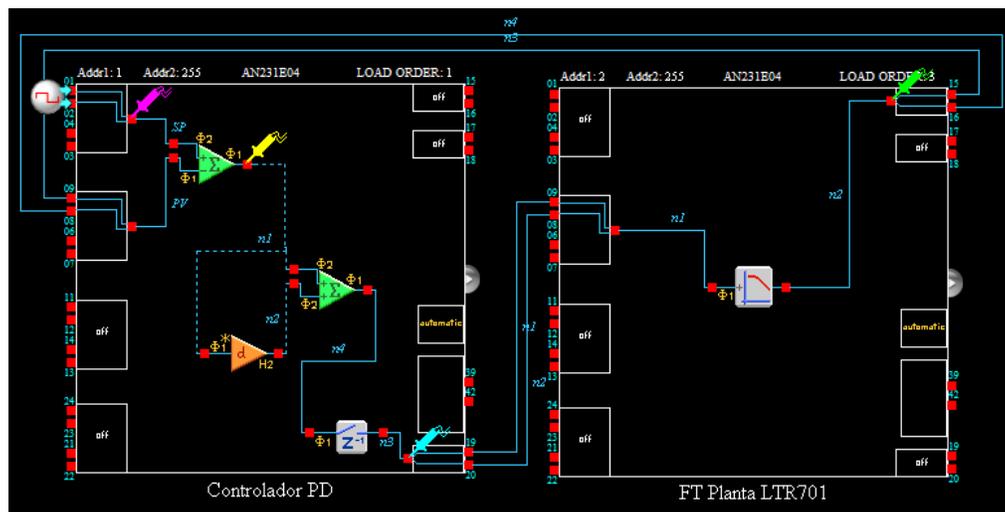


Figura 3.14: Modelado del controlador PD en la herramienta AD2

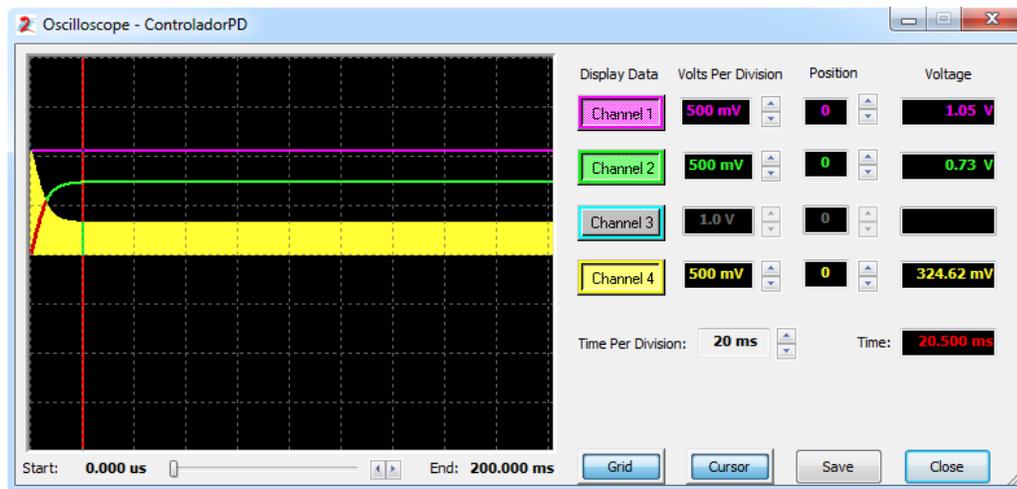


Figura 3.15: Simulación del controlador PD en la herramienta AD2

3.2.2.4. Controlador Proporcional-Integral

En el caso de la simulación del controlador PI (Figura 3.16), se observa que el sistema alcanza el valor que se establece en el *setpoint*, el inconveniente en este caso es la oscilación presente en la respuesta del sistema para los parámetros calculados (Figura 3.17).

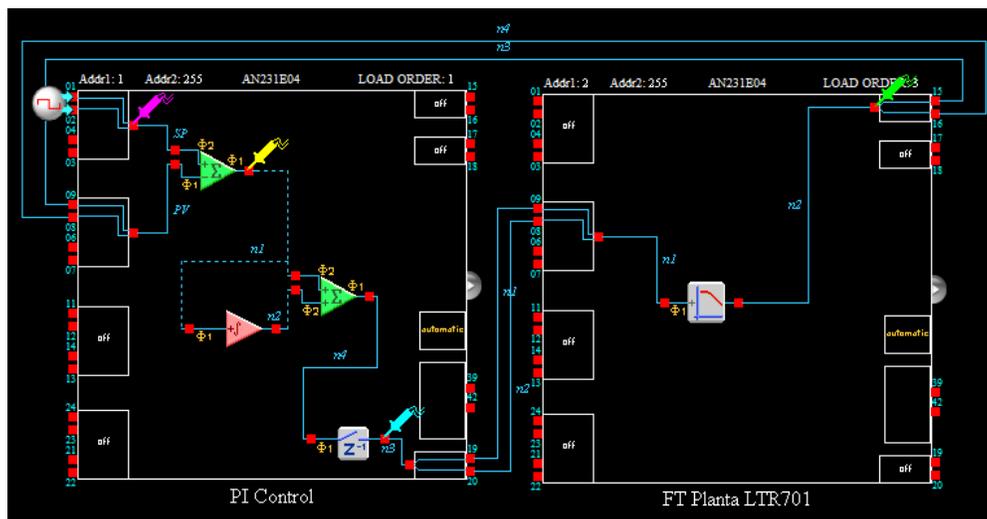


Figura 3.16: Modelado del controlador PI en la herramienta AD2

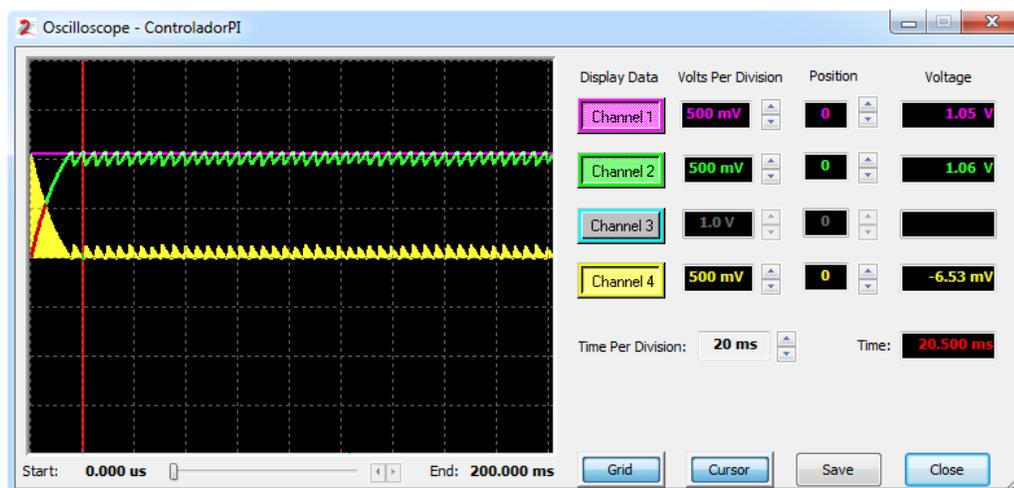


Figura 3.17: Simulación del controlador PI en la herramienta AD2

Con la finalidad de observar que el controlador PI permite la estabilidad del sistema, se reduce el valor del tiempo integral, para ello se considera la milésima parte del valor calculado. Obteniendo de esta manera una respuesta estable del sistema (Figura 3.18), donde el error en estado estacionario igual a cero se alcanza en 222,5ms aproximadamente.

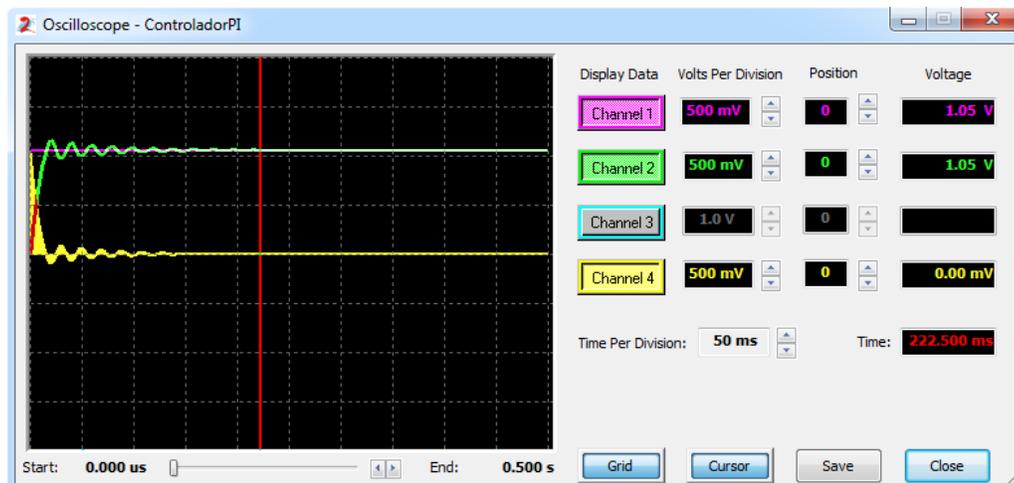


Figura 3.18: Ajuste de del tiempo integral controlador PI

3.2.2.5. Controlador Proporcional-Integral-Derivativo

Al simular el controlador PID (Figura 3.19), se observa que la respuesta del sistema (Figura 3.20) alcanza el valor deseado de temperatura en un tiempo menor al controlador PI y al igual que en esta última acción de control, el sistema muestra inestabilidad en la respuesta del sistema de control.

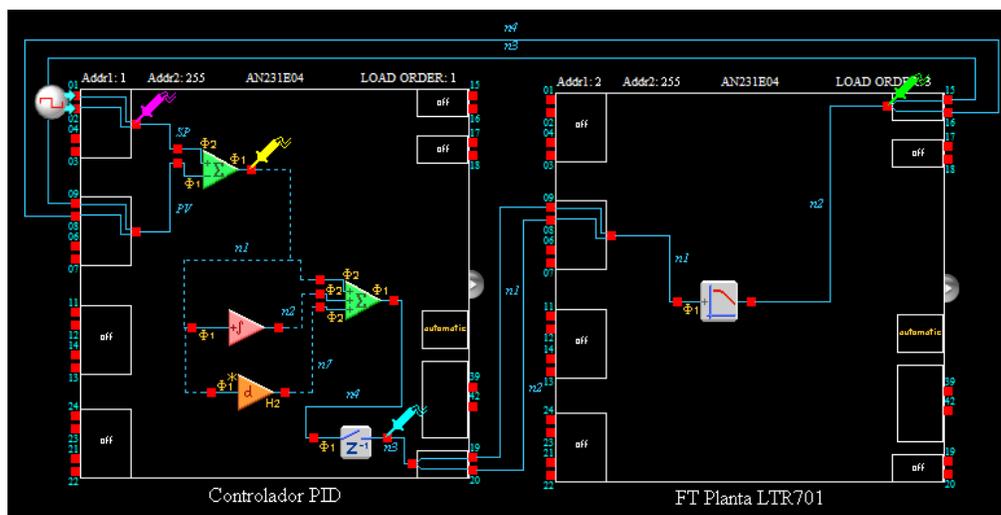


Figura 3.19: Modelado del controlador PID en la herramienta AD2

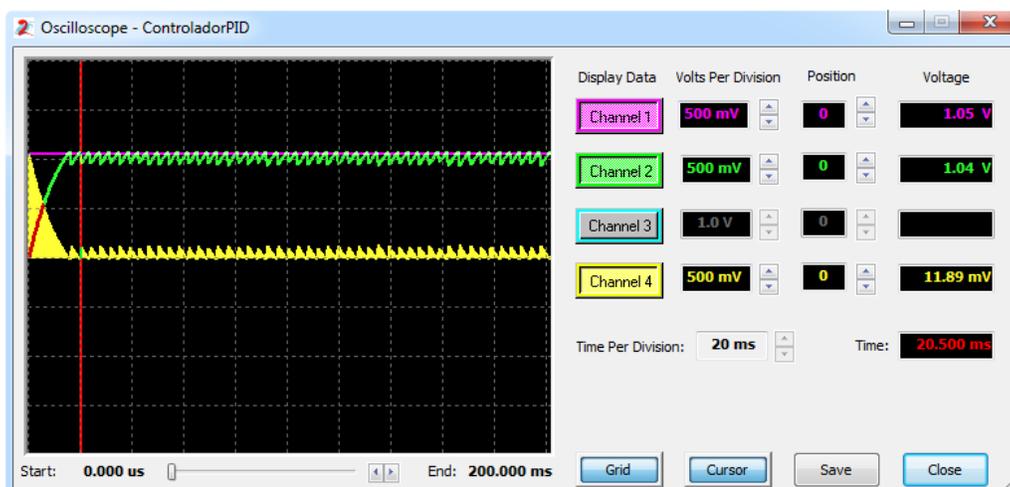


Figura 3.20: Simulación del controlador PID en la herramienta AD2

De manera similar que en el caso del controlador PI, se reduce el valor del tiempo integral para observar la estabilidad del sistema (Figura 3.21). Obteniendo un error en estado estacionario igual a cero. A diferencia del controlador anterior, presenta menos oscilaciones en la respuesta del sistema y se establece en un tiempo menor.

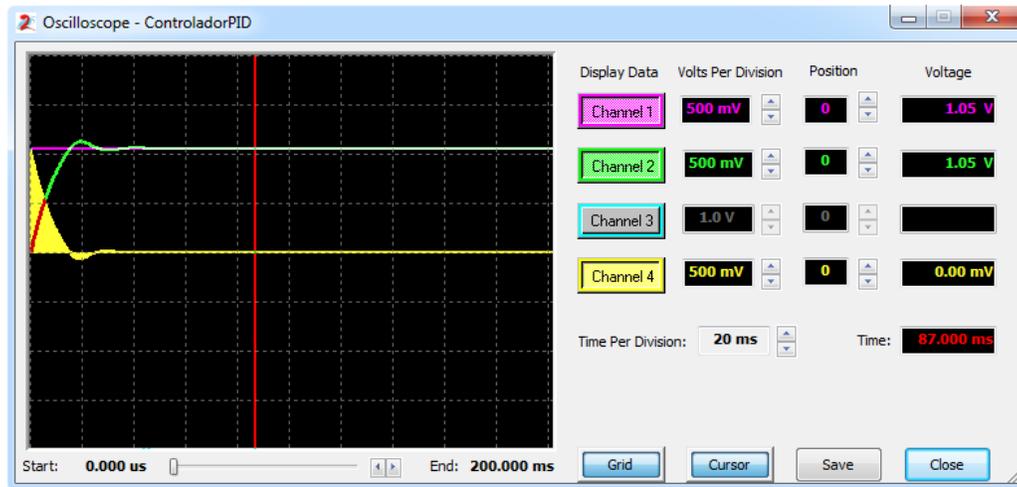


Figura 3.21: Ajuste de del tiempo integral controlador PID

Como se observa en los diagramas de simulación (Figuras 3.12, 3.14, 3.16 y 3.19), se hizo uso de un CAM para el muestreo de los datos (*Hold*), con la finalidad de adquirir el valor del voltaje procesado en la fase de muestreo y mantenerlo en la salida durante la fase de transferencia de carga.

Capítulo 4

Implementación del sistema de control de temperatura

En esta sección se describe la implementación sobre el FPAA de los controladores PID sintonizados por las reglas de Ziegler-Nichols y su aplicación a la planta LTR701.

4.1. Sistema de control de temperatura

El diagrama de bloques mostrado en la Figura 4.1, muestra el sistema utilizado en este trabajo de tesis y que será utilizado para implementar las leyes de control sintonizadas en el capítulo anterior.

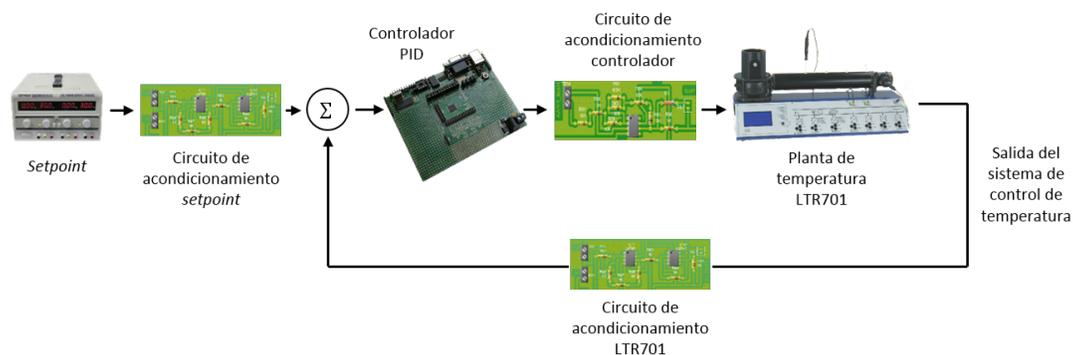


Figura 4.1: Diagrama de bloques del sistema de control de temperatura

Los elementos que integran al mencionado sistema son los siguientes:

- ✓ La planta a controlar
- ✓ Definición del *setpoint*
- ✓ Los módulos de acondicionamiento de las señales
- ✓ El control basado en el FPAA

4.1.1. Planta a controlar

La planta a controlar consiste en el módulo térmico descrito en la Sección 2.7.

4.1.2. Definición del *setpoint*

Este parámetro está controlado por una fuente de voltaje, misma que asume el papel del *setpoint* dentro del sistema de control. Los valores requeridos para este caso varían en el rango de $0 - 10V$ que con base en la documentación del LTR701, permiten la manipulación de temperaturas en el rango de $0 - 100^{\circ}C$.

Para llevar a cabo el control del módulo de temperatura, se considera establecer el valor del ventilador (*ACTUATOR* 1) a un 40% de su funcionamiento.

4.1.3. Acondicionamiento de señales

Tomando en cuenta la descripción dada en el capítulo anterior acerca de los valores de voltajes permitidos en los pines del FPAA usado (AN231E04), se realiza el diseño de las etapas de acondicionamiento con una relación $10 : 3$ para las entradas y $3 : 10$ para las salidas. Por lo que se consideró la implementación de un **amplificador inversor** y de un **no inversor**.

Para tal fin, se utilizaron OpAmps UA741, los cuales requieren una fuente simétrica de $15V$.

4.1.3.1. Amplificador inversor

Considérese el circuito mostrado en la Figura 4.2. El diseño de este circuito de acondicionamiento de señal (CAS) tiene por función reducir el valor de los voltajes provenientes de la fuente de DC definida como *setpoint* y del sensor 3 del LTR701 (salida de la planta), los cuales serán leídos por 2 de los pines de entrada del FPAA (*IOCell1* e *IOCell2*, pines 1, 2, 8 y 9, respectivamente).

Considerando que la ganancia de un amplificador inversor está dada por la Ecuación (4.1),

$$G_{inv} = -\frac{V_0}{V_i} = -\frac{R_f}{R_1} \quad (4.1)$$

y tomando en cuenta la condición para el acondicionamiento del voltaje de entrada al FPAA se obtiene:

$$G_{inv} = -\frac{R_f}{R_1} = \frac{3}{10}$$

Haciendo $R_1 = 10K\Omega$ y considerando la última expresión para la ganancia no es difícil apreciar que $R_f = 3K\Omega$. Cabe destacar que esta ganancia es negativa, por lo que será necesario

añadir un amplificador inversor de ganancia unitaria en serie, para obtener a la salida una señal con voltajes únicamente positivos, para este circuito se utilizan resistencias de $10K\Omega$. El circuito que cumple estas funciones se puede apreciar en la Figura 4.2. Al simular este circuito, se observa que la salida del CAS (señal senoidal azul) muestra la reducción de la señal de entrada (gráfico rojo) en la proporción requerida (Figura 4.3), cumpliendo de esta manera con el diseño propuesto.

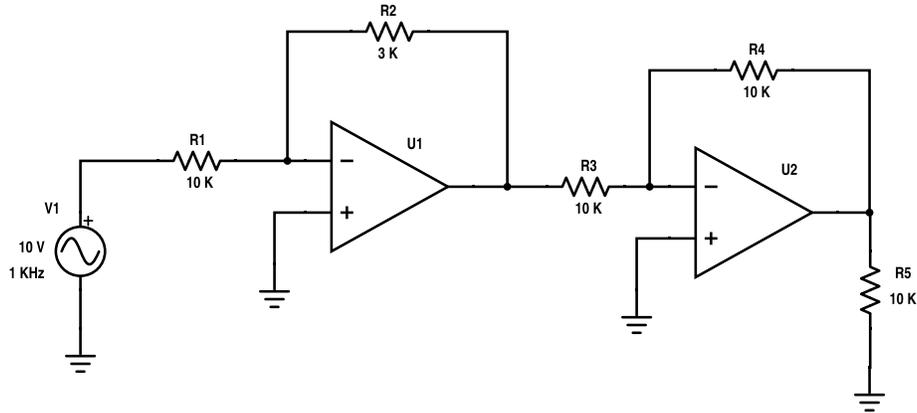


Figura 4.2: CAS de entrada

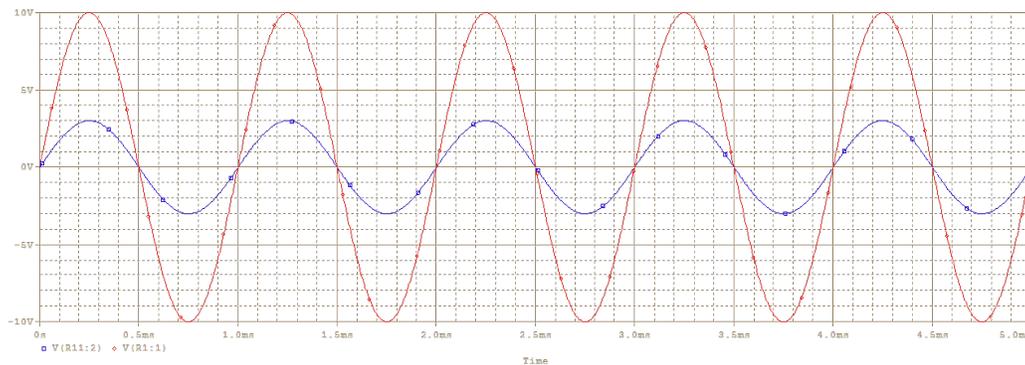


Figura 4.3: Acondicionamiento para señales de entrada

4.1.3.2. Amplificador no inversor

Considérese el circuito mostrado en la Figura 4.4. El objetivo de este circuito es permitir la amplificación de la señal proveniente de los pines 19 y 20 del FPAA (*IOCell7*), la cual será conectada al calefactor (*ACTUATOR 2*) del LTR701. Al comparar la ganancia del amplificador no inversor (Ecuación (4.2)) con el valor propuesto para la amplificación de la señal, se tiene,

$$G_{NoInv} = \frac{V_0}{V_i} = \frac{R_f}{R_1} + 1 \quad (4.2)$$

$$\begin{aligned} G_{NoInv} &= \frac{R_f}{R_1} + 1 = \frac{10}{3} \\ \Rightarrow \frac{R_f}{R_1} &= \frac{10}{3} - 1 \\ &= \frac{7}{3} \end{aligned}$$

De donde se obtienen los valores de las resistencias de retroalimentación ($R_f = 70K\Omega$) y de entrada ($R_1 = 30K\Omega$). Los resultados de la simulación para este circuito son los mostrados en la Figura 4.4, de la cual se observa que la señal de entrada (senoidal roja) es amplificada en la proporción requerida.

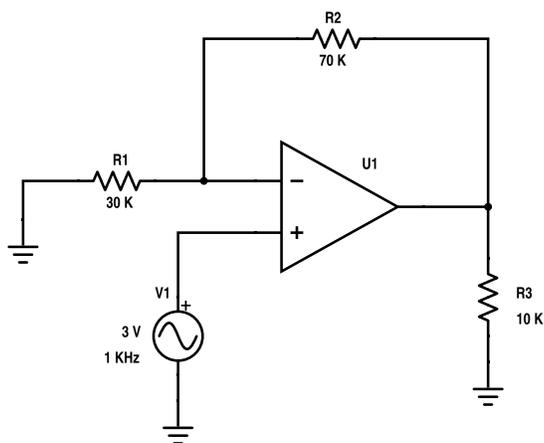


Figura 4.4: Esquemático del amplificador no inversor

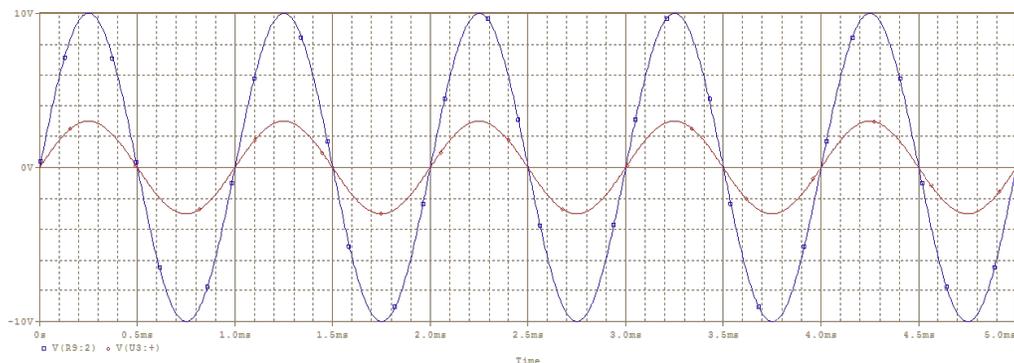


Figura 4.5: Acondicionamiento para señales de salida

El multiplicador asignado a las resistencias utilizadas se eligió de los componentes que se tenían a la mano, de ser posible se pueden ocupar múltiplos de estas, siempre y cuando

cumpla con el valor obtenido para las ganancias de los amplificadores. En la Figura 4.6 se muestra el diseño del circuito impreso realizado para el escalamiento de las señales de entrada y salida al FPAA.

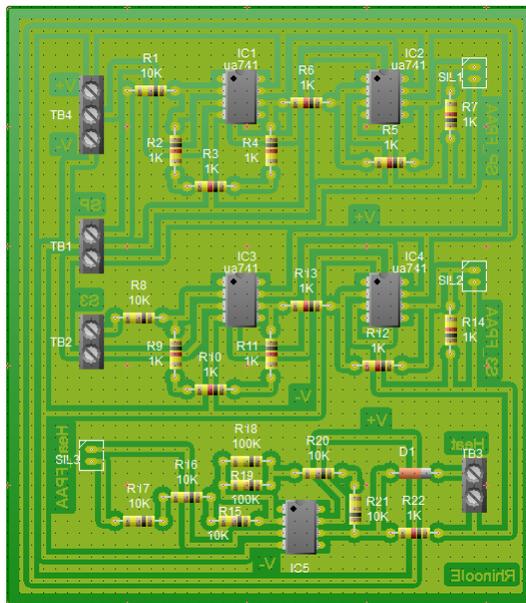


Figura 4.6: Circuito impreso para el acondicionamiento de señales

4.1.4. Control basado en FPAA

En esta sección se describen los resultados obtenidos al implementar los controladores PID en la tarjeta FPAA (AN231E04) en lazo cerrado con la planta de temperatura (LTR701). Las gráficas mostradas detallan el comportamiento del sistema mediante la medición de su respuesta, se efectúa además la comparación con la simulación del sistema hecha en MATLAB Simulink[®].

La simulación del sistema en MATLAB Simulink[®] requiere la realización de una aproximación de Padé (Ecuación (4.3)) para el retardo de la planta, lo cual permite que ésta se convierta en un sistema de fase no mínima, al tener un cero en el semiplano positivo.

$$\begin{aligned}
 e^{-sT_d} &= \frac{1 - \frac{T_d s}{2}}{1 + \frac{T_d s}{2}} & (4.3) \\
 &= \frac{2 - T_d s}{2 + T_d s} \\
 &= \frac{-(T_d s - 2)}{T_d s + 2} \\
 &= -\frac{s - \frac{2}{T_d}}{s + \frac{2}{T_d}}
 \end{aligned}$$

Las pruebas realizadas al implementar las leyes de control sobre el FPAA mostraron los resultados esperados, es decir, permitían la estabilidad del sistema al valor deseado, este parámetro se varió para observar el comportamiento del sistema. En los casos del controlador P y PD, el control no alcanzó en su totalidad el valor deseado, lo cual indica la existencia del error en estado estacionario, lo cual se debe al comportamiento característico de dichos controladores, a diferencia del PI y PID, donde pasado un tiempo el valor de la temperatura se establece al valor deseado (*setpoint*).

Para el análisis de los resultados obtenidos, se efectuó la medición de los voltajes leídos del SENSOR 3 del LTR701 mediante la DAQ USB1208FS de *MEASUREMENT COMPUTING* [40], se utilizó el “*Strip Chart*” del software TracerDAQ para facilitar la medición de los valores leídos. La tarjeta se configura en modo de operación “*singled ended*” permitiendo utilizar el canal 0 (pin 1) como entrada referenciado a tierra (pin 3). La medición de los valores se efectuó por un tiempo de 15 minutos. Una vez finalizada la medición y almacenados los datos en pantalla, se guardan estos en un archivo con extensión “.txt”, mismo que sirve para la comparación de la simulación hecha en MATLAB Simulink®. El *setpoint* utilizado en esta sección es de 5V equivalente a 50°C según las especificaciones del manual de la planta de temperatura LTR701.

Con este fin, se escribe el código dentro de un “*script*” de MATLAB®, en el cual se almacenan en un arreglo los datos medidos con la DAQ y se muestra la gráfica correspondiente, se genera y se muestra la señal del *setpoint*, se obtiene la aproximación para el retardo mediante el comando “*pade*” y se efectúa el producto con la definición de la función de transferencia de la planta de temperatura, se define el controlador y se obtiene la función de transferencia del control en lazo cerrado, para finalmente aplicar la respuesta al escalón (*setpoint*) con el comando “*lsim*” y mostrar el comportamiento de la salida del sistema.

Debido al tiempo empleado para la medición de los valores del comportamiento de las acciones básicas de control diseñadas y con base en el tiempo definido para el intervalo de muestreo ($h = 0,05s$), se define un vector tiempo (t) representativo con valores inicial y final de 0 y 18000 muestras, respectivamente. Este vector indica el tiempo total requerido para la medición de valores en términos del número de muestras generadas.

4.1.4.1. Control Proporcional

Debido a que un sistema de temperatura presenta una respuesta lenta, el controlador P no alcanza el valor del *setpoint* establecido. Al ejecutar el Código 4.1, se observa que la simulación del controlador proporcional mediante la aproximación de Padé presenta un error en estado estacionario de 11,6%, mientras que para los valores medidos de la implementación del controlador en el FPAA se tiene un error del 30%. La diferencia entre los errores de simulación y de implementación se debe a parámetros no considerados para el modelado del módulo de temperatura en el entorno donde se encuentra. La respuesta del sistema se muestra en la Figura 4.7.

Código 4.1: Script realizado para el controlador P

```

1  % Controlador P
2  clc;
3
4  t = 0:5:18000;
5
6  % Valores medidos del FPAA
7  temp = load('D:\MedicionesControlLTR701\MeasControlP_1.txt');
8  plot(temp(:,1), temp(:,2), 'm')
9
10 SP = 5; % Valor Set Point
11 hold on
12 u = [zeros(1,5), SP*ones(1,length(t)-5)]; % Señal para el SP
13 plot(t, u, 'r')
14
15 [NUM, DEN] = pade(0.8, 1); % Retardo del sistema
16 num = conv(0.485, NUM);
17 den = conv(DEN, [12.6 1]);
18 G = tf(num, den); % Función de transferencia de la Planta de Temperatura
19
20 Kp = 15.75; % Ganancia proporcional
21
22 hold on
23 [n, d]=tfdata(feedback(Kp*G, 1));
24 y=lsim(n, d, u, t); % Respuesta al escalón con valor SP
25 plot(t, y, 'b')
26 hold off

```

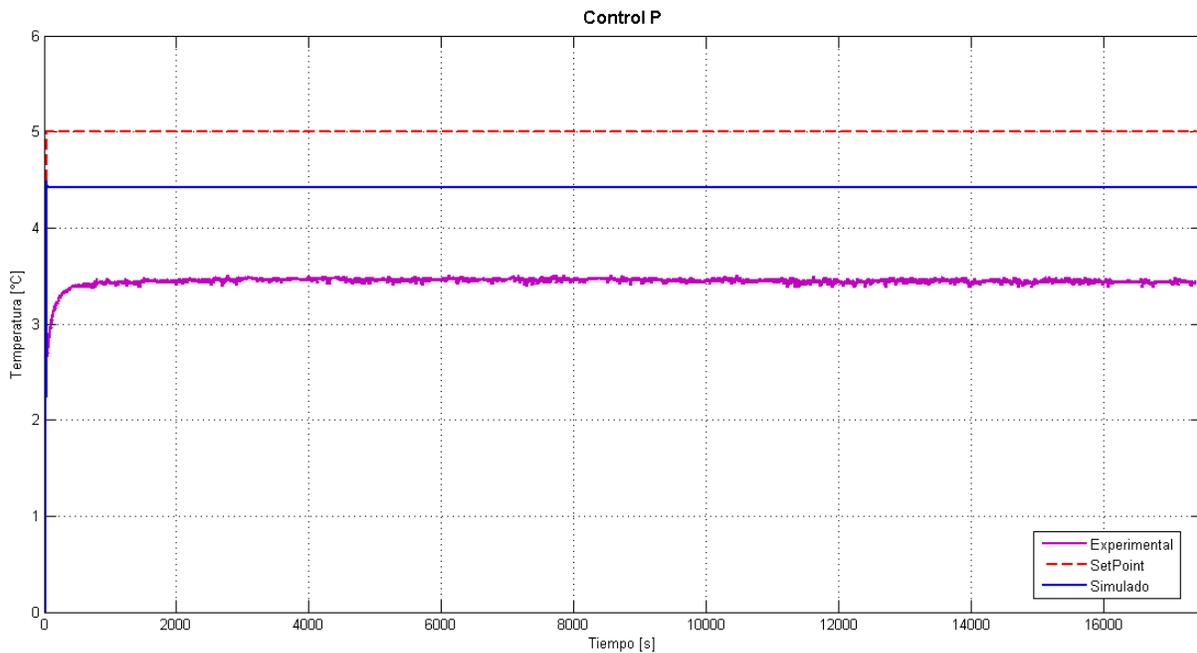


Figura 4.7: Implementación del control Proporcional

4.1.4.2. Control Proporcional-Derivativo

A la acción de control proporcional se le añade la constante de tiempo derivativa para formar el control PD, la adición de este parámetro tiene la finalidad de mantener el error en un valor mínimo, corrigiéndolo de manera proporcional a la velocidad a la cual se produce, y evitando su incremento. Al ejecutar el Código 4.2 y efectuar la comparación de la señal proveniente de la implementación del control PD en el FPAA y la simulación hecha mediante la aproximación de Padé (Figura 4.8) se observa que el error en estado estacionario decrementa a un 10% en ambos casos, como era de esperarse, la estabilidad del sistema mejora su comportamiento en comparación con el controlador P.

Código 4.2: Script realizado para el controlador PD

```
1  % Controlador PD
2  clc;
3
4  t = 0:5:18000;
5
6  % Valores medidos del FPAA
7  temp = load('D:\MedicionesControlLTR701\MeasControlPD_1.txt');
8  plot(temp(:,1), temp(:,2), 'm')
9
10 SP = 5; % Valor Set Point
11 hold on
12 u = [zeros(1,5), SP*ones(1,length(t)-5)];
13 plot(t, u, 'r')
14
15 [NUM, DEN] = pade(0.8, 1); % Retardo del sistema
16 num = conv(0.485, NUM);
17 den = conv(DEN, [12.6 1]);
18 G = tf(num, den); % Función de transferencia de la Planta de Temperatura
19
20 s = tf('s');
21 Kp = 18.9;
22 Td = 0.4;
23 Gpd = Kp*(1 + Td*s); % FT del control PD
24
25 hold on
26 [n, d]=tfdata(feedback(Gpd*G, 1));
27 y=lsim(n, d, u, t); % Respuesta al escalón con valor SP
28 plot(t, y, 'b')
29 hold off
```

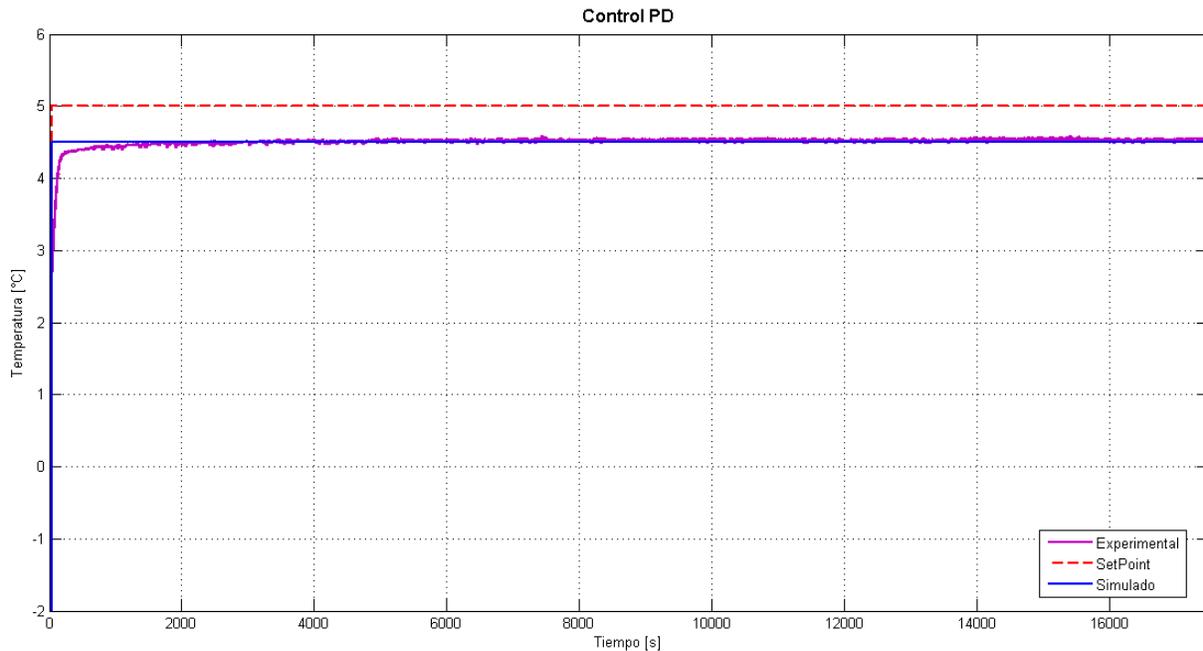


Figura 4.8: Implementación del control Proporcional-Derivativo

4.1.4.3. Control Proporcional-Integral

El Código 4.3 representa la comparación de la señal medida y simulada del controlador PI, el cual muestra que el error en estado estacionario para ambos casos es aproximadamente cero, lo cual permite la estabilidad del sistema al valor que se desea establecer la temperatura del sistema, esto es una de las características primordiales de este tipo de controlador (Figura 4.9).

Código 4.3: Script realizado para el controlador PI

```

1  % Controlador PI
2  clc;
3
4  t = 0:5:18000;
5
6  % Valores medidos del FPAA
7  temp = load('D:\MedicionesControlLTR701\MeasControlPI_1.txt');
8  plot(temp(:,1), temp(:,2), 'm')
9
10 SP = 5; % Valor Set Point
11 hold on
12 u = [zeros(1,5), SP*ones(1,length(t)-5)];
13 plot(t, u, 'r')
14
15 [NUM, DEN] = pade(0.8, 1); % Retardo del sistema

```

```

16 num = conv(0.485, NUM);
17 den = conv(DEN, [12.6 1]);
18 G = tf(num, den); % Función de transferencia de la Planta de Temperatura
19
20 s = tf('s');
21 Kp = 14.175;
22 Ti = 2.667;
23 Gpi = Kp*(1+1/(Ti*s)); % FT del control PI
24
25 hold on
26 [n, d]=tfdata(feedback(Gpi*G, 1));
27 y=lsim(n, d, u, t); % Respuesta al escalón con valor SP
28 plot(t, y, 'b')
29 hold off

```

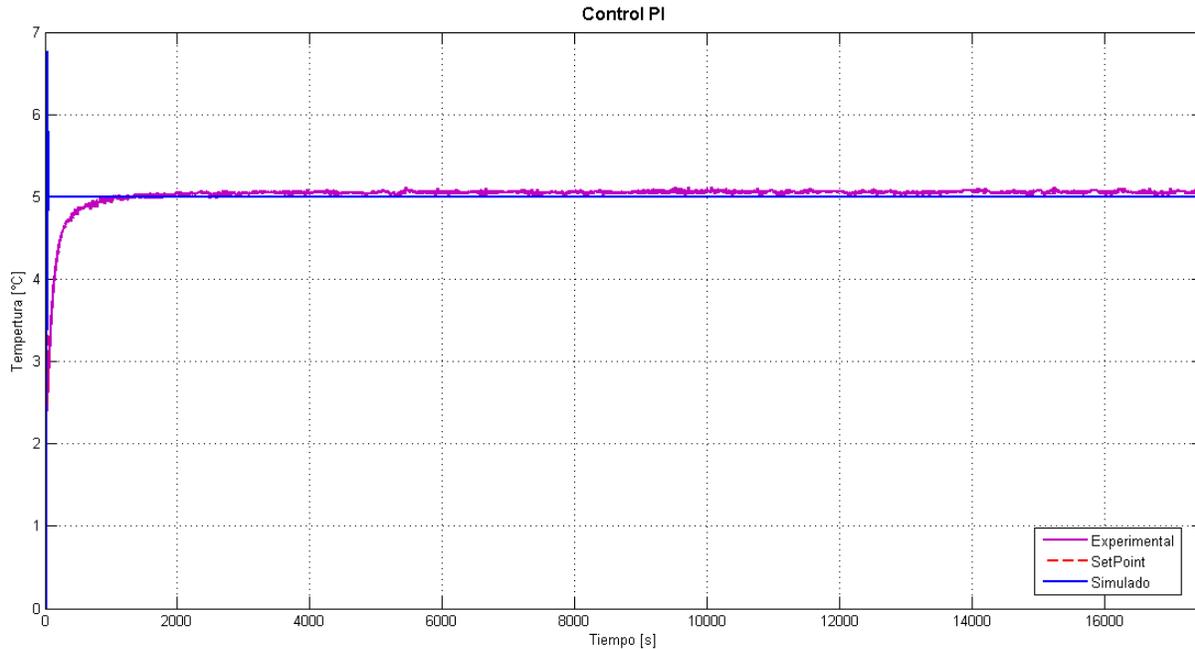


Figura 4.9: Implementación del control Proporcional-Integral

4.1.4.4. Control Proporcional-Integral-Derivativo

Una vez observado el comportamiento de los controladores P, PD y PI, se efectúa el análisis de las 3 acciones de control en conjunto, es decir, del controlador PID. Con el cual se tiene que el error en estado estacionario de la simulación es cero, mientras que el de las mediciones hechas de la implementación del controlador PID en el FPAA, muestra un error 1,2% sobre el valor de referencia. Lo cual es permitido por el controlador, ya que mantiene la temperatura al valor al cual se establece en el *setpoint*.

Código 4.4: Script realizado para el controlador PID

```
1 % Controlador PID
2 clc;
3
4 t = 0:5:18000;
5
6 % Valores medidos del FPAA
7 temp = load('D:\MedicionesControlLTR701\MeasControlPID_1.txt');
8 plot(temp(:,1), temp(:,2), 'm')
9
10 SP = 5; % Valor Set Point
11 hold on
12 u = [zeros(1,5), SP*ones(1,length(t)-5)];
13 plot(t, u, 'r')
14
15 [NUM, DEN] = pade(0.8, 1); % Retardo del sistema
16 num = conv(0.485, NUM);
17 den = conv(DEN, [12.6 1]);
18 G = tf(num, den); % Función de transferencia de la Planta de Temperatura
19
20 s = tf('s');
21 Kp = 18.9;
22 Ti = 1.6;
23 Td = 0.4;
24 Gpid = Kp*(1 + 1/(Ti*s) + Td*s); % FT del control PID
25
26 hold on
27 [n, d]=tfdata(feedback(Gpid*G, 1));
28 y=lsim(n, d, u, t); % Respuesta al escalón con valor SP
29 plot(t, y, 'b')
30 hold off
```

Además, se observa que mediante el modelado con la aproximación de Padé, los controladores PD y PID presentan un subvalor en la respuesta transitoria del sistema, igual al 60% del valor del *setpoint* establecido, lo cual se debe al comportamiento del sistema de fase no mínima. Mientras que los controladores PI y PID presentan un subvalor de 35,3% y 26% respectivamente.

4.1.4.5. Índice de desempeño

En la Figura 4.11 se muestra la evolución en el tiempo del índice de desempeño conocido como ISE (*Integral Square Error*) para cada uno de los controladores evaluados. Como puede apreciarse, a menor índice ISE, se tiene mejor desempeño, lo cual se muestra en el comportamiento de los controladores PI y PID.

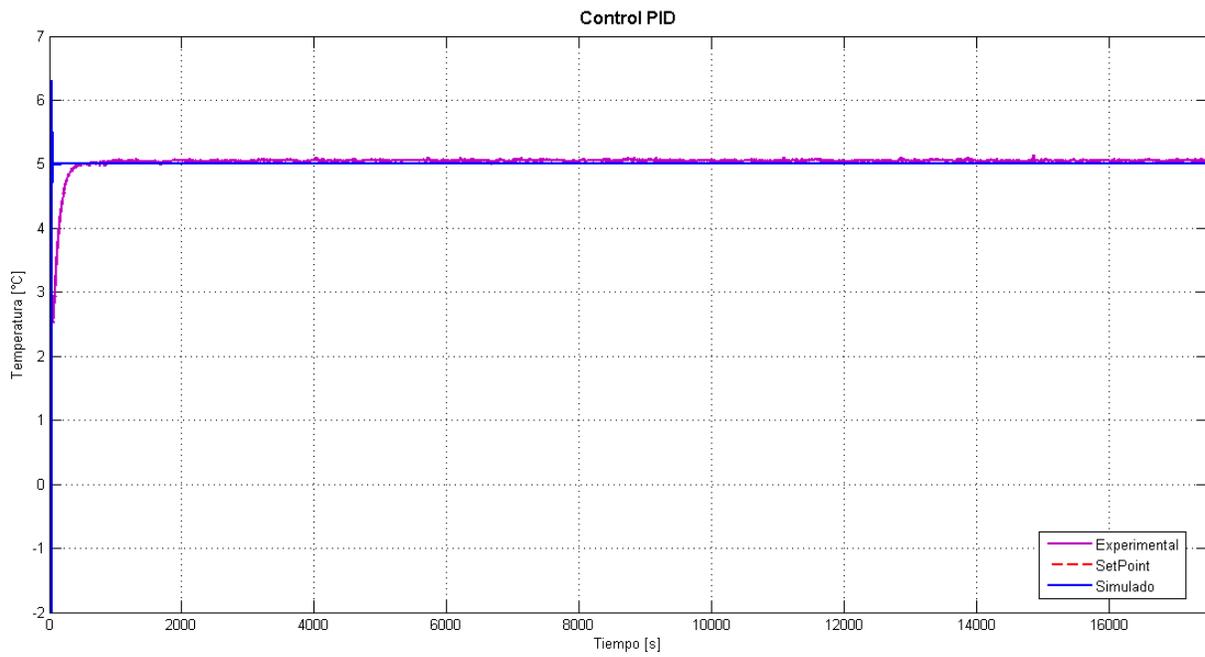


Figura 4.10: Implementación del control Proporcional-Integral-Derivativo

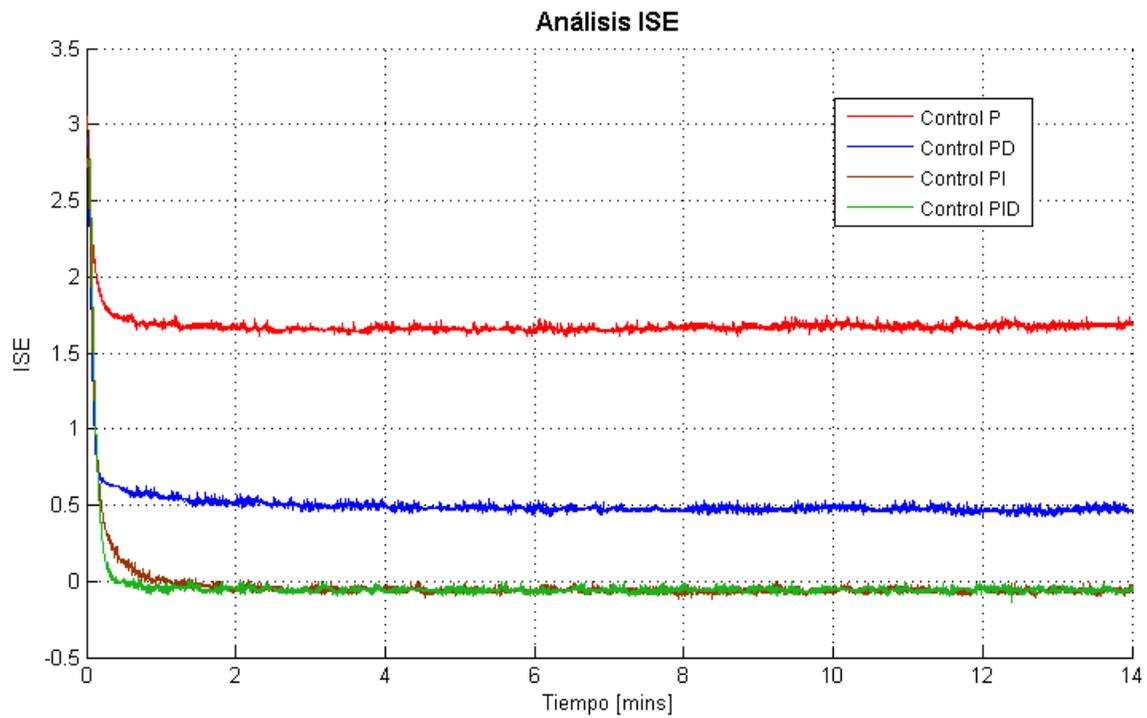


Figura 4.11: Error integral cuadrático, ISE

4.1.4.6. Recursos empleados

A partir de la Tabla 4.1 se detalla la distribución de los recursos empleados para cada una de las acciones básicas de control basadas en FPAA.

✓ Control P

- *SumDiff* CAM. Situado en el CAB 2, utilizando 3 capacitores y 1 OpAmp.
- *Gain* CAM. Configurado en el CAB 1 con 2 capacitores y 1 OpAmp.
- *Hold* CAM. Al igual que el CAM previo, utiliza 2 capacitores y 1 OpAmp.

✓ Control PD

- *SumDiff* 1 y 2 CAM. Situados en el CAB 1, utilizando 3 capacitores y 1 OpAmp cada uno.
- *Differentiator* CAM. Configurado en el CAB 2 con 3 capacitores y 1 OpAmp.
- *Hold* CAM. Situado en el CAB 2, utilizando 2 capacitores y 1 OpAmp.

✓ Control PI

- *SumDiff* 1 y 2 CAM. Situados en el CAB 1, utilizando 3 capacitores y 1 OpAmp cada uno.
- *Integrator* CAM. Configurado en el CAB 2 con 2 capacitores y 1 OpAmp.
- *Hold* CAM. Utiliza 2 capacitores y 1 OpAmp situados en el CAB 2.

✓ Control PID

- *SumDiff* 1 y 2 CAM. Situados en el CAB 1, utilizando 3 y 4 capacitores respectivamente y 1 OpAmp cada uno.
- *Integrator* CAM. Configurado en el CAB 2 con 2 capacitores y 1 OpAmp.
- *Differentiator* CAM. Configurado en el CAB 3 con 3 capacitores y 1 OpAmp.
- *Hold* CAM. Establecido en el CAB 2, utiliza 2 capacitores y 1 OpAmp.

Tabla 4.1: Recursos empleados en el FPAA

Controlador	Número de capacitores en uso	Número de OpAmps en uso	Porcentaje de capacitores empleados	Porcentaje de OpAmps empleados
P	7	3	21.875 %	37.5 %
PI	10	4	31.25 %	50 %
PD	11	4	34.375 %	50 %
PID	14	5	43.75 %	62.5 %

Capítulo 5

Conclusiones

En el presente trabajo de tesis, se presentó la implementación de controladores PID basados en FPAA, utilizando el kit de desarrollo *AN231K04-DVLP4* de Anadigm, y su aplicación al control del módulo térmico *LTR701*. A este respecto, se establecen las siguientes conclusiones:

- ✓ El uso de un FPAA en la implementación de circuitos analógicos proporciona un rápido prototipado, permitiendo observar el comportamiento del circuito al instante. Además, reduce el número de componentes para su implementación, es decir, para el diseño de un circuito analógico y su implementación en un FPAA, solo es necesario modelar las características del circuito deseado y configurar el dispositivo; dichas tareas son realizadas fácilmente a través de la herramienta AD2, la cual además permite realizar otras tareas como una simulación funcional. Por su parte, una implementación tradicional es más laboriosa al tener que realizar el circuito impreso del diseño, o armarlo en un protoboard, además de ser más susceptible a fallas y también más difícil de verificar.
- ✓ La implementación de controladores PID basados en FPAA muestra estabilidad cuando operan en el sistema, lo cual es acorde a la respuesta obtenida mediante simulación; esta característica demuestra que un FPAA es un dispositivo viable para implementar sistemas de control en lazo cerrado.
- ✓ Debido a la presencia de capacitores en la arquitectura de un FPAA, existe ruido en las señales procesadas, por lo que se debe tener especial cuidado al momento de asignar las frecuencias de muestreo para los CAMs y el funcionamiento del FPAA.
- ✓ Un FPAA al presentar características de un sistema muestreado, tiene limitaciones en cuanto a la frecuencia de muestreo, es decir, la relación entre esta frecuencia y el valor de los parámetros de configuración de los CAMs influye directamente en el comportamiento del circuito modelado en el FPAA. Debido a esto, el diseñador debe establecer la relación adecuada entre uno y otro. De esta manera, se observa que los valores para los parámetros de configuración de los CAMs tienden a ser finitos, por lo que no siempre será posible obtener la configuración deseada.

- ✓ En comparación con un diseño digital, la implementación de circuitos usando la tecnología de los FPAA simplifica el diseño de estos al no requerir el uso de ADCs y DACs, lo que reduce el tiempo de ejecución y minimiza la pérdida de datos.

5.1. Trabajos futuros

Con base en los resultados obtenidos y a la revisión bibliográfica de los FPAA, se proponen los siguientes trabajos futuros:

- ✓ Aplicación de los controladores PID a plantas de segundo orden.
- ✓ Evaluación de la interfaz digital para la implementación de un sistema de ajuste automático.
- ✓ Aplicación de módulos PWM para diferentes tipos de convertidores de voltaje.
- ✓ Modelado del retardo de un sistema.

Apéndice

Apéndice A

Modelado e implementación de circuitos analógicos en AD2

Para efectuar el diseño de circuitos analógicos dentro de un FPAA, es necesario utilizar la herramienta CAD que provee Anadigm[®]. AD2 es una herramienta que permite la implementación de estos circuitos mediante *Módulos Analógicos Configurables* (*Configurable Analog Modules*, CAMs), los cuales permiten un rápido prototipado de circuitos con tan solo situar bloques, conectarlos y configurarlos en base a los requerimientos previstos por el diseño [11, 14].

A.1. Interfaz AD2

Como se mencionó previamente, el software de desarrollo para la implementación de circuitos analógicos es AD2 (Figura A.1), el cual se puede descargar desde la página del proveedor del FPAA (<http://www.anadigm.com/>). Se debe crear un usuario para poder acceder al archivo de descarga y a la licencia del producto. La versión disponible a la fecha es la 2.8.0.1.



Figura A.1: AnadigmDesigner2[®]

Una vez descargado e instalado el programa, se accede al mismo (Figura A.2), lo que prosigue a esto es el efectuar el test de evaluación para observar la funcionalidad del programa.

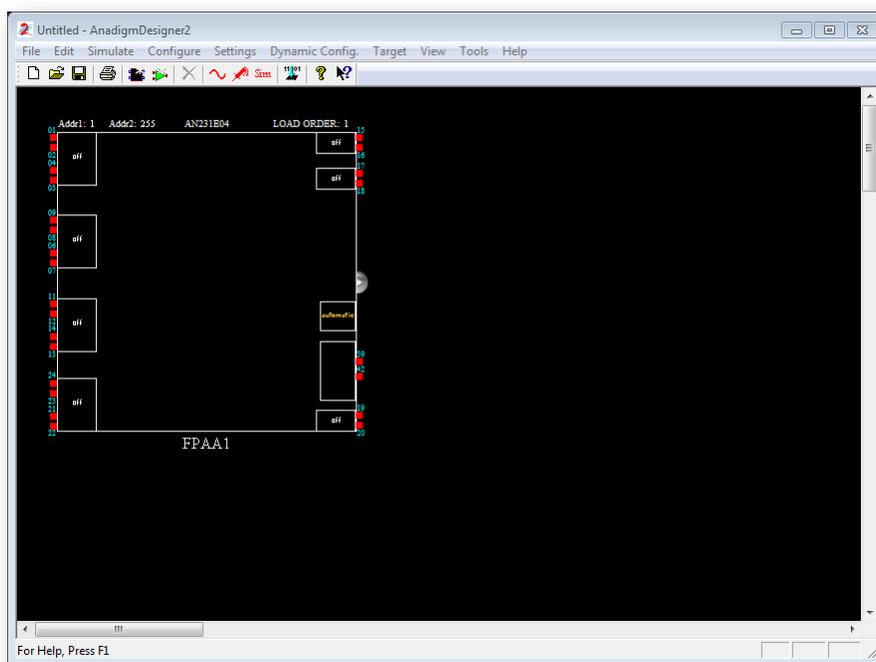


Figura A.2: Interfaz del programa AD2

Cerciorarse de que el chip insertado en el entorno de trabajo de AD2 sea el correspondiente al **AN231E04**, debido a que es el incorporado en el kit de desarrollo en uso. Una vez hecha esta aclaración, se inserta el CAM deseado, ya sea mediante el menú “*Edit/Insert New CAM...*” o presionando la tecla “M” (Figura A.3). Observe que la biblioteca de funciones de donde se obtendrán los CAMs es la correspondiente a AnadigmApex, la cual es la orientada a dispositivos FPAA de tercera generación.

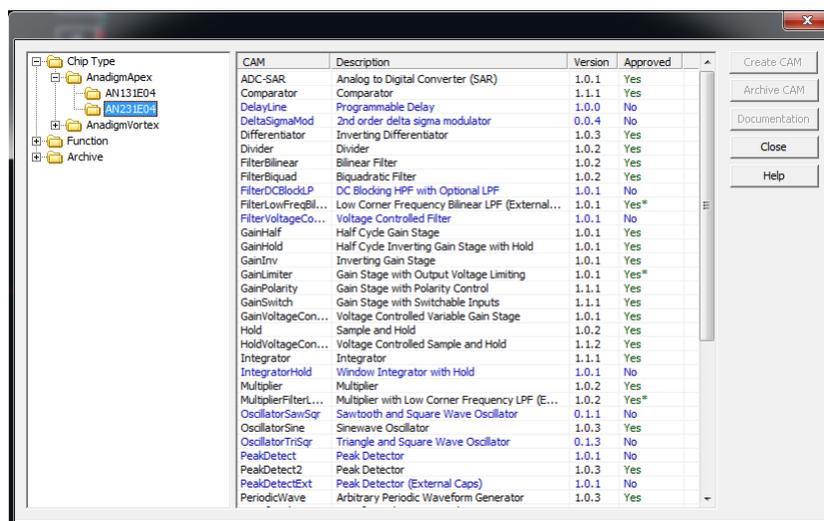


Figura A.3: Insertando CAM

Al seleccionar un CAM, se activan las opciones para su creación y almacenamiento, así como su documentación. Dentro de esta última opción, se encuentra información correspondiente a la compatibilidad de hardware (dispositivos FPAA's en los cuales se puede utilizar), su descripción, los CAMs relacionados, el diagrama del circuito equivalente SC y la función de transferencia correspondiente, además de la fase a la cual conmuta cada interruptor del circuito, las características de salida del CAM, el código en C requerido para su configuración automática y algunas notas de diseño.

Para insertar un CAM, es necesario seleccionar la opción “*Create CAM*” y posicionarlo en el área correspondiente al chip, seguido a esto, aparece un cuadro que permite la configuración de los parámetros del módulo a insertar, estos son diferentes para cada CAM utilizado.

Después de haber insertado y configurado los parámetros del CAM **GainInv** (Figura A.4) a una ganancia invertida de 2, se conecta hacia las celdas de entrada y salida correspondientes. Para activar una celda I/O en el FPAA, se debe dar doble clic sobre la celda deseada y establecerla como entrada o salida en modo “*Bypass*”.

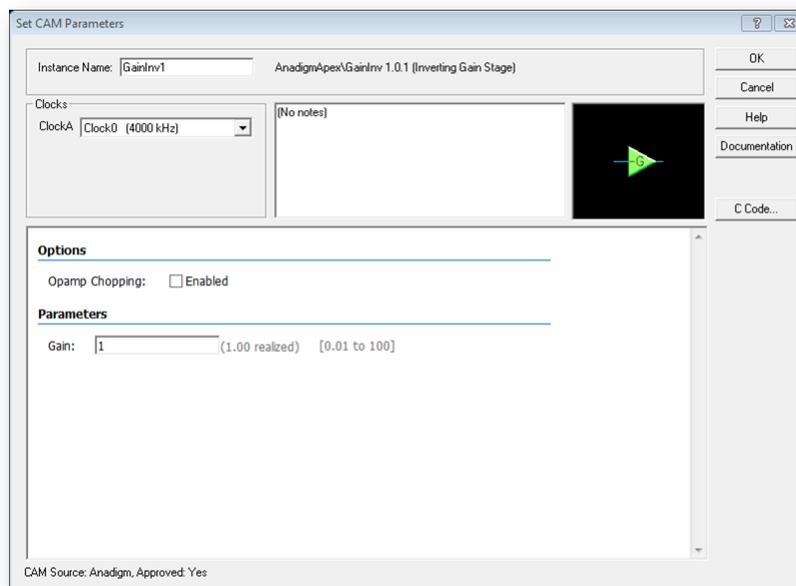


Figura A.4: Parámetros del CAM de ganancia invertida

Una vez establecidas las conexiones en el circuito de prueba, se procede a su simulación. Para ello, se debe insertar un “generador de señales” y “puntas de medición”.

Existen 4 formas de insertar un generador de funciones. La primera de ellas es a través el menú “*Simulate/Create Signal Generator*”, la segunda es por medio del “clic derecho” sobre el área de trabajo y escoger la opción “*Create Signal Generator*”, la tercera mediante el ícono “*Create Signal Generator*” (representado por una onda senoidal en color rojo) ubicado en la

barra de herramientas y, la cuarta a través de la tecla “M”. Al dar doble clic sobre el bloque generado, se muestra un cuadro que permite establecer los parámetros para la señal (Figura A.5).

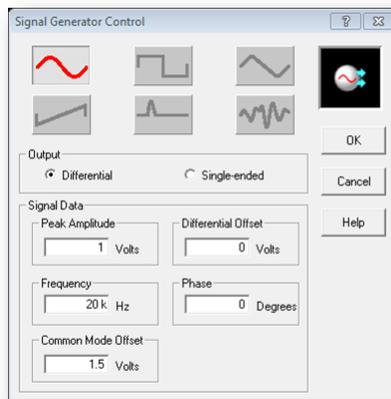


Figura A.5: Parámetros del generador de funciones

De manera similar se tienen estas opciones para establecer las puntas de medición, la diferencia radica en el nombre de la acción, ya que para este caso, se elige la opción “*Create Oscilloscope Probe*”, cambiando la tecla “M” por “P” y el ícono por el de una punta de prueba del mismo color que el anterior. La limitante es que se dispone sólo de 4 puntas.

Para llevar a cabo la simulación en AD2, se cuenta con 3 formas de hacerlo. Mediante el menú “*Simulate/Begin Simulation*”, con el ícono “*Sim*” de la barra de herramientas o a través de la tecla “F5”. Previo a esta acción, se puede modificar el tiempo de simulación del circuito por medio del menú “*Simulate/Setup Simulation...*” (Figura A.6). Con las opciones disponibles se permite establecer el tiempo de inicio y fin de simulación, la visualización dinámica del comportamiento de las señales I/O y el código correspondiente a la programación de las ecuaciones del circuito y, las estadísticas de la ejecución de la simulación.

El circuito amplificador inversor creado y su respuesta de simulación se muestra en la Figura A.7, la señal de entrada es una sinusoidal de $1V_{pp}@20KHz$. La salida del amplificador proporciona una señal a la misma frecuencia pero invertida y amplificada al doble de su valor original.

Las estadísticas obtenidas después de ejecutar la simulación del circuito de prueba se muestran en la Figura A.8.

Una vez realizada la simulación y analizado la respuesta obtenida, se procede a la programación del FPAA. Como se describió en la Sección 2.6.5, un FPAA puede ser programado a través de la interfaz serial (RS-232) o por medio del puerto USB, en este caso y debido a las características de la PC en uso, se emplea el cable USB para la programación del kit de desarrollo. Para operar en este modo y antes de conectar el kit por primera vez en una PC,

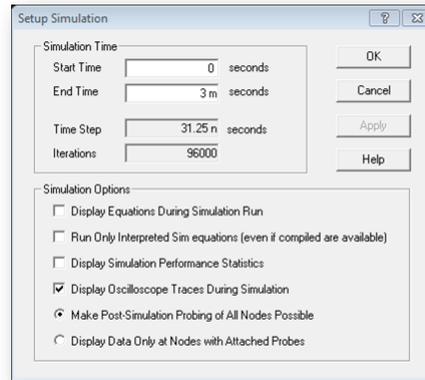


Figura A.6: Parámetros de simulación

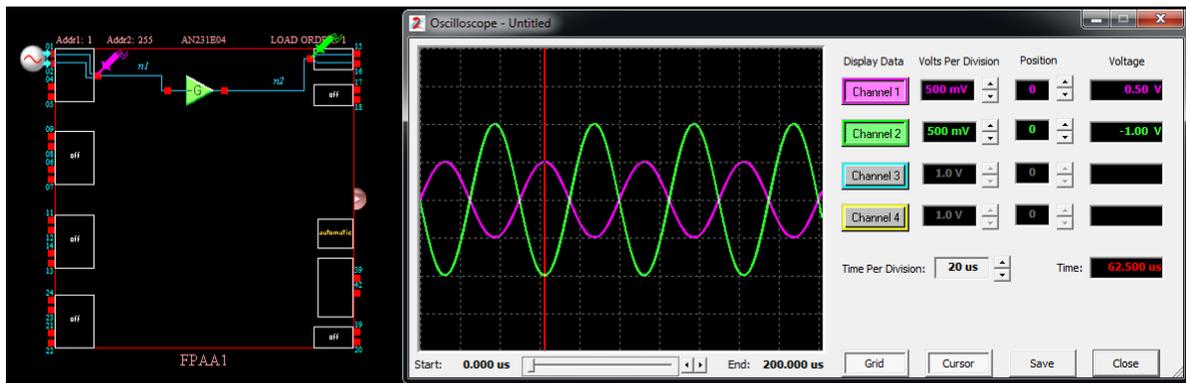


Figura A.7: Implementación de un circuito amplificador inversor

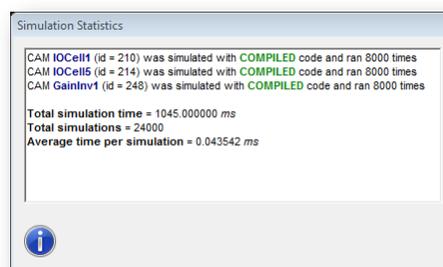


Figura A.8: Estadísticas de simulación

es necesario tener instalado el driver proporcionado por *Silicon Labs* [27].

Después de conectar la tarjeta a la PC y alimentarla con un voltaje de 3,3V, se debe verificar que AD2 la haya reconocido en el entorno, para ello, se debe revisar en el menú “*Target/Display Board Information*”, si la tarjeta ha sido reconocida con éxito, se mostrará un mensaje como el de la Figura A.9a), en caso contrario (Figura A.9b)), se debe cerciorar de que

el kit se este alimentando con el voltaje pertinente o en su defecto, que AD2 tenga configurado el puerto COM. Para lo cual se debe revisar en el menú “*Settings/Preferences.../Port*” y verificar que el la lista desplegable de la opción “*Select Port*” contenga el puerto activado (Figura A.10).

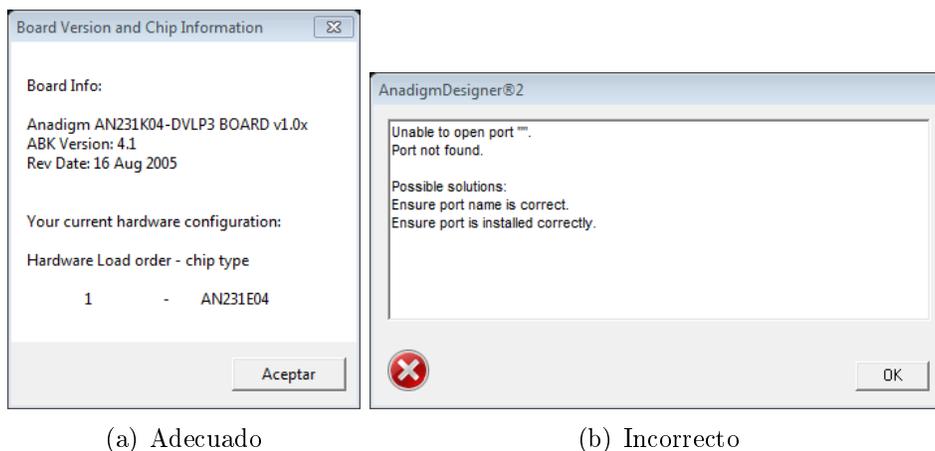


Figura A.9: Reconocimiento del kit de desarrollo dentro de AD2

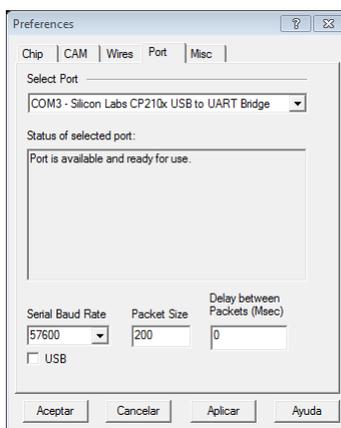


Figura A.10: Configuración del puerto COM

Posterior a esta revisión, el kit de desarrollo está listo para ser programado con el diseño creado en AD2, para esto, se procede a descargar el archivo hacia el FPAA mediante una de las 4 formas posibles. A través del menú “*Configure/Write Configuration to Data Port*”, seleccionando el chip del diseño y elegir la opción “*Write Configuration to Data Port (selected chips only)*” de la lista desplegable al dar clic sobre este, por el ícono presente en la barra de tareas (flecha azul con dígitos binarios) o presionando la combinación de teclas “*Ctrl + w*”.

Una vez programado el FPAA, éste está listo para efectuar con las diversas configuraciones programadas. Permitiendo al usuario reconfigurar el diseño (cambiar los valores en los parámetros de los CAMS) las veces que desee sin necesidad de desconectar el circuito del

sistema en el cual está operando.

Tras programar el archivo creado sobre el kit de desarrollo, se observa la respuesta mostrada en la Figura A.11.

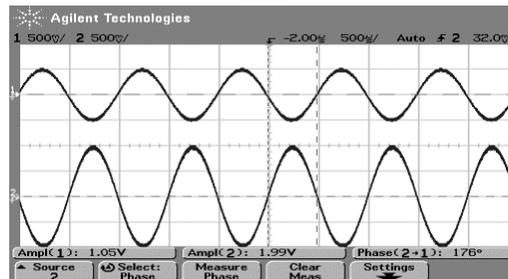


Figura A.11: Respuesta del circuito amplificador inversor

A.2. Modulador AM

La Amplitud Modulada (AM) es el proceso de cambiar la amplitud de una señal portadora con una frecuencia alta de acuerdo con la amplitud de la señal modulante o información.

Un modulador de AM es un dispositivo no lineal con dos señales de entrada: una señal portadora de amplitud constante y una señal de información. La información modula la señal portadora, proveniente de una onda de frecuencia única o compleja, compuesta de diferentes frecuencias (una o más fuentes). Debido a que la información actúa sobre la portadora, se le denomina señal modulante o moduladora.

A.2.1. La envolvente de AM

Existen diferentes formas de modulación en amplitud. La más sencilla de ellas es la portadora de AM de doble banda lateral (*Double Side Band Frequency Carrier*, **DSBFC**). La Figura A.12 muestra un modulador AM de este tipo en el dominio del tiempo. La onda modulada contiene todas las frecuencias que componen la señal AM y se utilizan para llevar la información a través del sistema. A la forma de la onda modulada se le denomina envolvente. Cuando se aplica una señal moduladora, la amplitud de la onda de salida varía de acuerdo a esta. De tal manera que la forma de la envolvente de AM es idéntica a la forma de la señal moduladora.

A.2.2. Diseño del circuito en AD2

Para llevar a cabo el diseño del modulador AM, se crea un proyecto nuevo de la forma en como se describió en la Sección A.1 y se insertan los CAMs necesarios. Los módulos a utilizar y los valores asignados a cada parámetro son los detallados en la Tabla A.1.

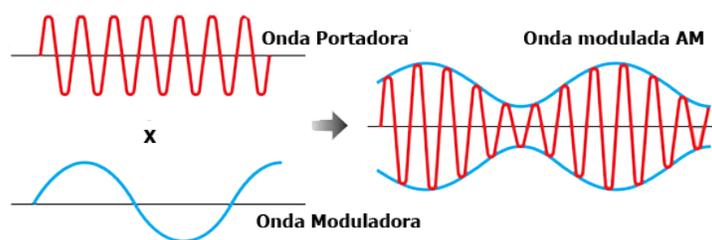


Figura A.12: Modulación AM

Tabla A.1: Parámetros para el modulador AM

CAM	Parámetro
Multiplicador (Multiplier)	Clock A: Clock 3 @ 250 KHz Clock B: Clock 0 @ 4 MHz Sample and Hold: Off Multiplier Factor: 1
Sumador (SumDiff)	Clock A: Clock 3 @ 250 KHz Output Phase: Phase 1 Input 1: Inverting Input 2: Non-inverting Input 3, 4: Off Gain 1, 2: 1
Voltaje de Referencia (Voltage)	Polarity: Positive (+2V)
Oscilador Senoidal (OscillatorSine)	Clock A: Clock 3 @ 250 KHz Osc. Frequency: 12 KHz Peak Amplitude: 0.5 V
Filtro Bicuadrático 1 (FilterBiquad)	Clock A: Clock 3 @ 250 KHz Filter Type: High PassFilter Topology: Type II Corner Frequency: 0.5 KHz Gain: 1 Quality Factor: Default
Filtro Bicuadrático 2 (FilterBiquad)	Clock A: Clock 3 @ 250 KHz Filter Type: Low PassFilter Topology: Automatic Corner Frequency: 8 KHz Gain: 1 Quality Factor: Default

La Figura A.13 muestra el circuito diseñado para el modulador AM. La señal portadora se genera a partir del CAM del oscilador senoidal, mientras que la señal moduladora se obtiene de

una fuente externa, en el caso de la simulación se aplica una señal senoidal de $1V$ de amplitud a una frecuencia de $1,5KHz$ con *offset* diferencial de $250mV$.

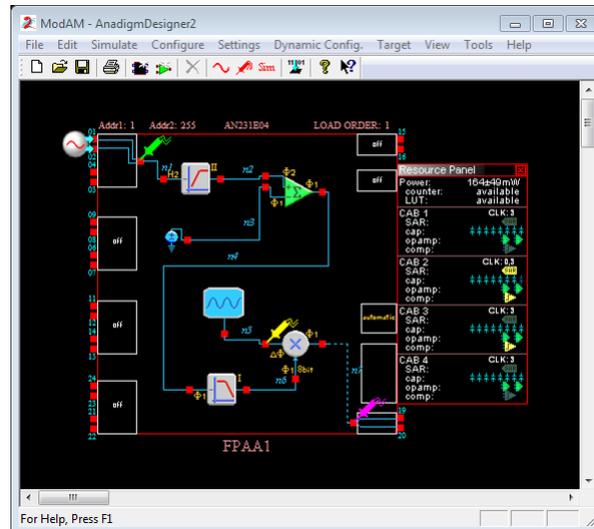


Figura A.13: Modulador AM en AD2

La respuesta de la simulación a un tiempo de $4ms$ ilustrada en la Figura A.14 muestra la envolvente generada para este tipo de modulación (*Channel 3*). Esta última se observa con mayor detalle al añadir un nuevo chip al diseño de simulación, al cual se inserta el CAM “Hold” estableciendo el “Clock A @ 250 KHz”. Esta acción se lleva a cabo con fines de análisis y debido a que el “FPAA1” utilizó todos los recursos disponibles para la tarjeta impidiendo agregar otro CAM al diseño.

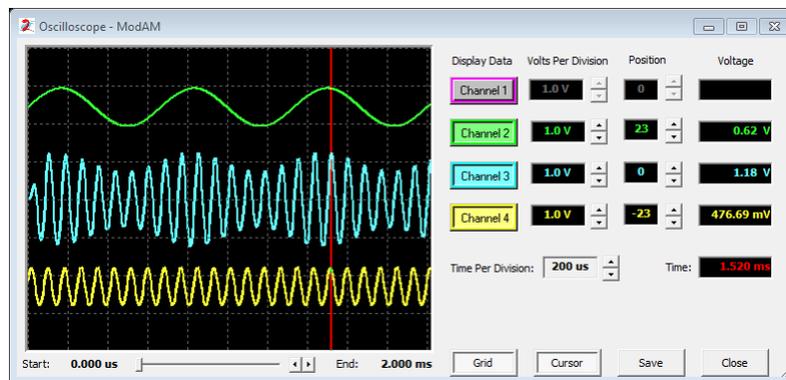


Figura A.14: Respuesta de simulación del modulador AM en AD2

Una variante del diseño mostrado en la Figura A.13, se obtiene al cambiar la fuente interna que genera la señal portadora (*CAM OscillatorSine*) por una externa, es decir, la señal portadora ahora es leída por el FPAA a través de un *generador de funciones*. Se agrega también el *CAM Hold*. Esta modificación se ilustra en la Figura A.15. Además, se hace hincapié en que la cantidad de recursos utilizados (“Resource Panel”) por los diseños es diferente, siendo el segundo quien emplea una cantidad menor de la arquitectura del FPAA.

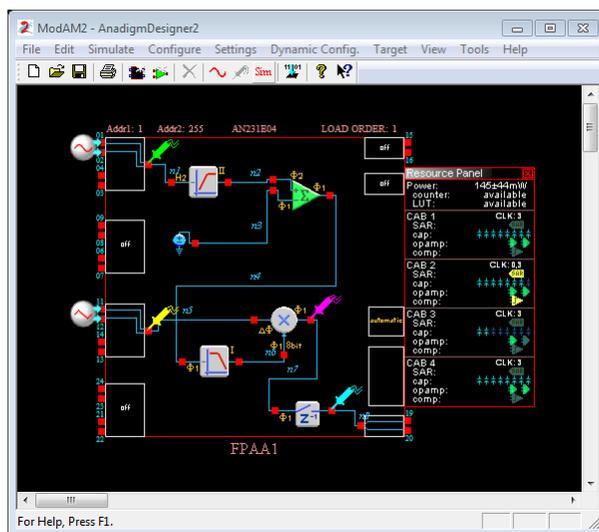


Figura A.15: Modulador AM modificado

Al efectuar la simulación, se observa el mismo comportamiento del caso anterior. La señal portadora en este caso se cambia por una señal triangular.

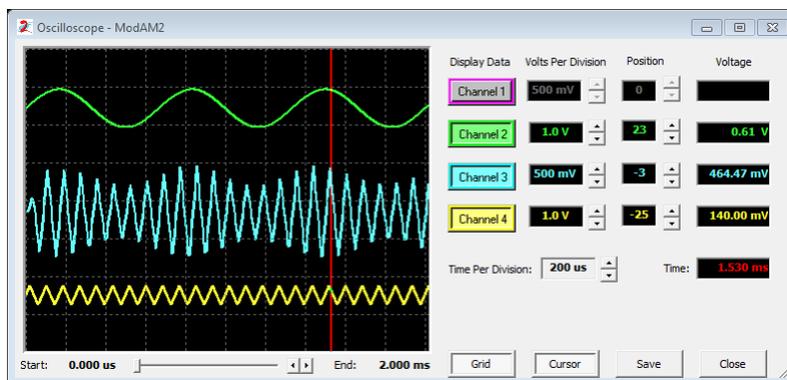


Figura A.16: Respuesta de simulación del modulador AM modificado

Para tener más en claro la forma en como funciona el CAM del multiplicador, se recomienda revisar la documentación que provee Anadigm [12].

A.3. Linealización de funciones

La linealización es el proceso de transformar una señal no lineal, en una lineal. Para ello es necesario efectuar determinados procedimientos matemáticos o físicos, con el objetivo de trasladar el valor medido en un valor equivalente en una línea recta.

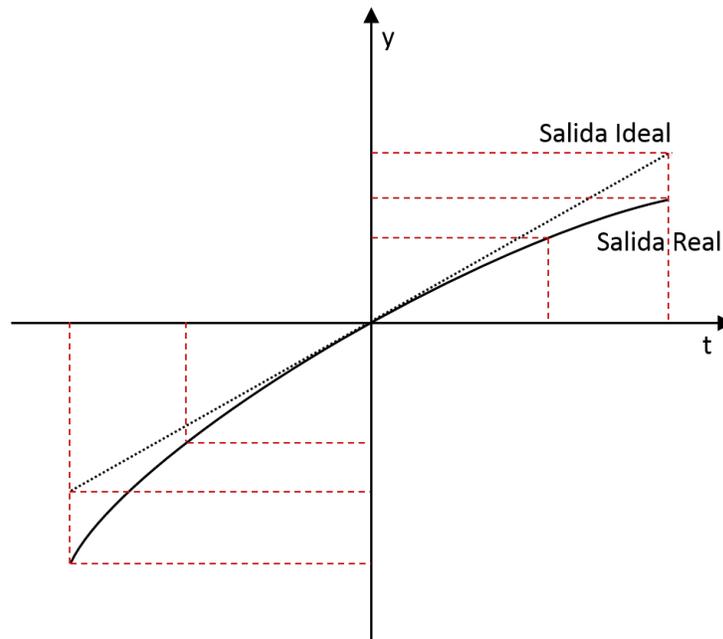


Figura A.17: Comportamiento no lineal de una función

A.3.1. Parámetros de diseño

Para llevar a cabo la simulación de este diseño, se requiere del uso de 2 chip FPAA, uno para el ajuste de la señal de entrada (acondicionamiento) y otro para el ajuste lineal de las mediciones hechas. El CAM primordial para este caso es el “*TransferFunction*” [22].

Tras crear un proyecto nuevo, insertar los CAMs necesarios y configurar sus parámetros (Tabla A.2), se obtiene el circuito mostrado en la Figura A.18.

Tabla A.2: Parámetros para el circuito de linealización

CAM	Parámetro
Función de Transferencia FPAA1 y FPAA2 (TransferFunction)	ClockA: Clock 3 @ 250 KHz ClockB: Clock 0 @ 4 MHz Output Hold: On
Sumador (SumDiff)	Clock A: Clock 3 @ 250 KHz Output Phase: Phase 1 Input 1: Non-inverting Input 2: Inverting Input 3, 4: Off Gain 1, 2: 1
Sample & Hold	ClockA: Clock 3 @ 250 kHz Input Sampling Phase: Phase 1

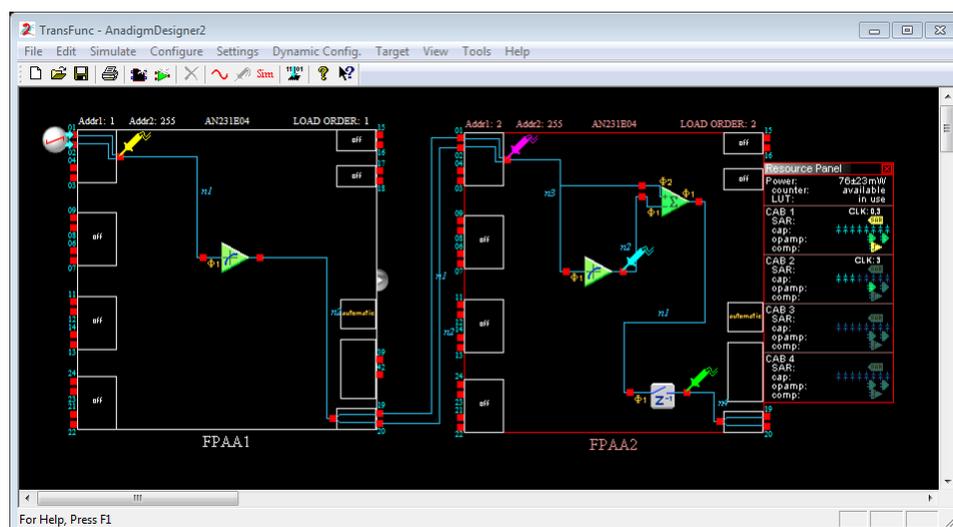
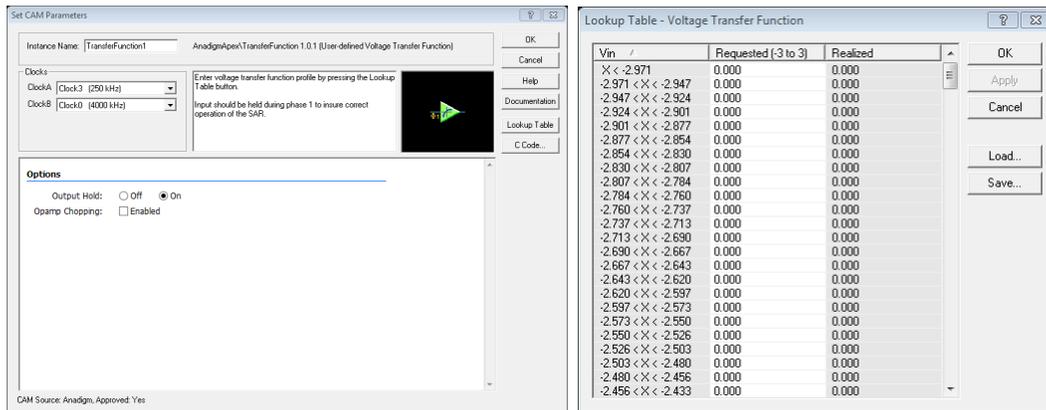


Figura A.18: Circuito para la linealización de señales en AD2

Debido a que el CAM *TransferFunction* utiliza la LUT para almacenar los valores correspondientes a una función arbitraria, se accede a sus registros al dar doble clic sobre este módulo. De la ventana que aparece (Figura A.19a)), acceder a la opción “*Lookup Table*”, la cual muestra la matriz correspondiente a la LUT de 256 posiciones. En esta nueva ventana se pueden observar 3 columnas *Vin*, *Requested* y *Realized*. *Vin* muestra los rangos de valores en la entrada del módulo que van de -3 a $3V$, a cada rango de valores se le puede asignar un valor de salida en la columna *Requested* y, en la columna *Realized* se puede observar el valor que el módulo asignará a la salida del mismo. Estos valores no se pueden modificar directamente en esta ventana (Figura A.19b)), para ello es necesario cargar un archivo con extensión “.*csv*” que se puede crear en *Excel* o desde el *Bloc de notas*. Además de contener la información de la función a generar, debe cumplir con registrar 256 valores en una columna. Teniendo disponible este archivo, se cargan los valores hacia la LUT mediante botón “*Load...*”, esta acción abrirá otra ventana, la cual es similar al cuadro de diálogo que muestran diversos programas para abrir un archivo, una vez abierta esta opción, se busca el archivo correspondiente y se da clic en el botón “*Abrir*”. Hecho esto, se observa que la columna *Requested* ha sido llenada con los valores leídos del archivo, para asignar estos valores a los registros de la LUT, se debe dar clic en el botón “*Apply*”, lo que mostrará en la columna *Realized* los valores asignados. Para culminar la asignación de valores a los registros de la LUT, se cierran las ventanas restantes por medio del botón “*OK*”. De esta manera se ha configurado el CAM *TransferFunction* para la implementación de una señal no lineal.

Los archivos “.*csv*” creados para las señales de entrada y de linealización se basan en los valores mostrados en las Tablas A.3 y A.4 respectivamente.

La simulación se efectúa con un tiempo de $1ms$. El generador de funciones se configura de manera que proporcione una señal rampa con salida diferencial de amplitud $3V_p$ a $2KHz$ de frecuencia. Considerando ángulo de desfase de 0° y *offset* diferencial de $3V$. La respuesta de simulación obtenida se muestra en la Figura A.20, donde se observa que la señal de salida



(a) Parámetros de configuración

(b) Registros de la LUT

Figura A.19: Asignación de valores a la LUT en AD2

Tabla A.3: Valores para la señal de entrada

0.000	0.000	0.000	0.000	0.000	0.000	0.012	0.203	0.793	1.561	2.312	2.851
0.000	0.000	0.000	0.000	0.000	0.000	0.015	0.223	0.826	1.597	2.343	2.867
0.000	0.000	0.000	0.000	0.000	0.000	0.019	0.244	0.859	1.633	2.373	2.883
0.000	0.000	0.000	0.000	0.000	0.000	0.024	0.266	0.892	1.669	2.402	2.897
0.000	0.000	0.000	0.000	0.000	0.000	0.028	0.289	0.926	1.705	2.431	2.911
0.000	0.000	0.000	0.000	0.000	0.000	0.033	0.312	0.959	1.740	2.460	2.924
0.000	0.000	0.000	0.000	0.000	0.000	0.039	0.335	0.993	1.776	2.488	2.936
0.000	0.000	0.000	0.000	0.000	0.000	0.044	0.360	1.028	1.811	2.516	2.947
0.000	0.000	0.000	0.000	0.000	0.000	0.050	0.385	1.062	1.846	2.543	2.957
0.000	0.000	0.000	0.000	0.000	0.000	0.056	0.411	1.097	1.881	2.589	2.966
0.000	0.000	0.000	0.000	0.000	0.000	0.063	0.437	1.132	1.916	2.595	2.974
0.000	0.000	0.000	0.000	0.000	0.000	0.070	0.464	1.167	1.951	2.620	2.981
0.000	0.000	0.000	0.000	0.000	0.000	0.077	0.491	1.203	1.985	2.645	2.986
0.000	0.000	0.000	0.000	0.000	0.000	0.085	0.520	1.238	2.019	2.669	2.991
0.000	0.000	0.000	0.000	0.000	0.000	0.093	0.548	1.274	2.053	2.692	
0.000	0.000	0.000	0.000	0.000	0.000	0.101	0.577	1.310	2.087	2.714	
0.000	0.000	0.000	0.000	0.000	0.000	0.110	0.607	1.345	2.120	2.736	
0.000	0.000	0.000	0.000	0.000	0.000	0.119	0.637	1.381	2.153	2.757	
0.000	0.000	0.000	0.000	0.000	0.001	0.130	0.667	1.417	2.186	2.778	
0.000	0.000	0.000	0.000	0.000	0.003	0.147	0.698	1.453	2.218	2.797	
0.000	0.000	0.000	0.000	0.000	0.006	0.165	0.730	1.489	2.250	2.816	
0.000	0.000	0.000	0.000	0.000	0.009	0.184	0.761	1.525	2.281	2.834	

(Channel 2) tiene un comportamiento lineal en comparación con la señal de entrada (Channel 1).

Tabla A.4: Valores para la señal de linealización

0.000	0.000	0.000	0.000	0.000	0.000	-0.155	-0.370	-0.220	-0.049	0.131	0.224
0.000	0.000	0.000	0.000	0.000	0.000	-0.235	-0.364	-0.209	-0.037	0.131	0.224
0.000	0.000	0.000	0.000	0.000	0.000	-0.330	-0.358	-0.209	-0.024	0.141	0.223
0.000	0.000	0.000	0.000	0.000	0.000	-0.358	-0.351	-0.197	-0.024	0.149	0.221
0.000	0.000	0.000	0.000	0.000	0.000	-0.371	-0.351	-0.185	-0.012	0.158	0.218
0.000	0.000	0.000	0.000	0.000	0.000	-0.379	-0.345	-0.185	0.000	0.158	0.216
0.000	0.000	0.000	0.000	0.000	0.000	-0.386	-0.337	-0.173	0.000	0.165	0.211
0.000	0.000	0.000	0.000	0.000	0.000	-0.389	-0.329	-0.161	0.013	0.173	0.207
0.000	0.000	0.000	0.000	0.000	0.000	-0.392	-0.321	-0.161	0.024	0.180	0.198
0.000	0.000	0.000	0.000	0.000	0.000	-0.393	-0.313	-0.148	0.024	0.180	0.187
0.000	0.000	0.000	0.000	0.000	0.000	-0.394	-0.303	-0.137	0.036	0.187	0.175
0.000	0.000	0.000	0.000	0.000	0.000	-0.394	-0.303	-0.137	0.047	0.192	0.155
0.000	0.000	0.000	0.000	0.000	0.000	-0.395	-0.294	-0.124	0.047	0.198	0.132
0.000	0.000	0.000	0.000	0.000	0.000	-0.393	-0.284	-0.124	0.059	0.203	0.105
0.000	0.000	0.000	0.000	0.000	0.000	-0.392	-0.284	-0.112	0.070	0.203	
0.000	0.000	0.000	0.000	0.000	0.000	-0.393	-0.274	-0.099	0.070	0.208	
0.000	0.000	0.000	0.000	0.000	0.000	-0.389	-0.264	-0.099	0.081	0.212	
0.000	0.000	0.000	0.000	0.000	0.000	-0.387	-0.254	-0.087	0.091	0.216	
0.000	0.000	0.000	0.000	0.000	0.001	-0.383	-0.254	-0.074	0.091	0.218	
0.000	0.000	0.000	0.000	0.000	0.003	-0.380	-0.242	-0.074	0.102	0.221	
0.000	0.000	0.000	0.000	0.000	-0.042	-0.374	-0.232	-0.062	0.112	0.222	
0.000	0.000	0.000	0.000	0.000	-0.042	-0.374	-0.232	-0.049	0.122	0.224	

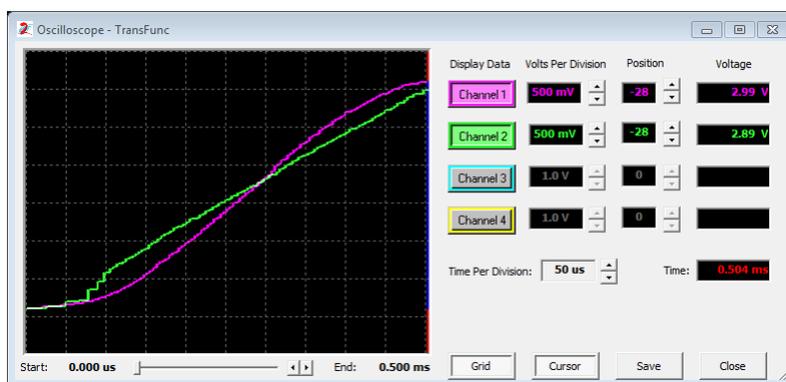


Figura A.20: Respuesta de linealización

Para implementar el circuito de manera física, el FPAA1 se deberá sustituir por el circuito de acondicionamiento correspondiente, ya sea el caso de la lectura de valores de algún

potenciómetro o de un sensor. El circuito presente en el chip FPAA2 es el que deberá ser programado sobre el kit de desarrollo. De ser necesario, cambiar los valores en la LUT por los que modelen el comportamiento que se desee linealizar.

A.4. PWM

La modulación por ancho de pulsos (*Pulse Width Modulation*, **PWM**) es la técnica en la cual se modifica el ciclo de trabajo de una señal periódica (por ejemplo, una onda senoidal), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida (Figura A.21). Una de las entradas se conecta a un oscilador de onda dientes de sierra, mientras que la otra queda disponible para la señal moduladora. En la salida, la frecuencia es generalmente igual a la de la señal dientes de sierra y el ciclo de trabajo está en función de la señal portadora.

La Ecuación A.1 define el comportamiento para el ciclo de trabajo de un PWM.

$$M_a = \frac{A_r}{A_c} \quad (\text{A.1})$$

Donde M_a se conoce como índice de modulación, A_r denota la amplitud de la señal de referencia y A_c la amplitud de la señal portadora. El rango de valores permitidos para el índice de modulación está comprendido entre 0 y 1.

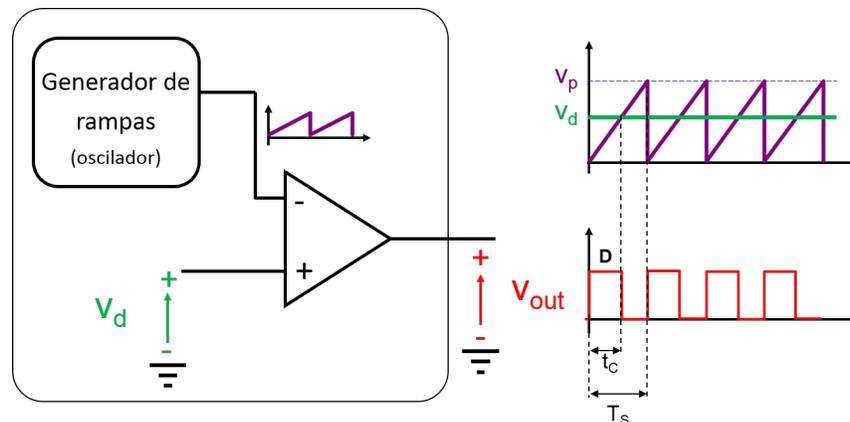


Figura A.21: Generador de señal PWM

A.4.1. Diseño de circuitos PWM

Se diseña un PWM para que su comportamiento muestre el 25 %, 50 % y 75 % de su ciclo de trabajo. Para ello, se considera una señal portadora diente de sierra de $2V@15KHz$ con

offset de $2V$. En base a la Ecuación A.1 y sustituyendo los valores correspondientes, se obtiene el valor de la amplitud para la señal de referencia,

$$A_r = M_a * A_c \quad (\text{A.2})$$

1. $M_a = 0,25$.

$$A_r = (0,25) * (2V) = 0,5V$$

2. $M_a = 0,50$.

$$A_r = (0,50) * (2V) = 1,0V$$

3. $M_a = 0,75$.

$$A_r = (0,75) * (2V) = 1,5V$$

La Figura A.22 muestra el circuito PWM diseñado para una señal diente de sierra y un voltaje de referencia (A_r). La configuración de los parámetros de los CAMs utilizados se muestra en la Tabla A.5.

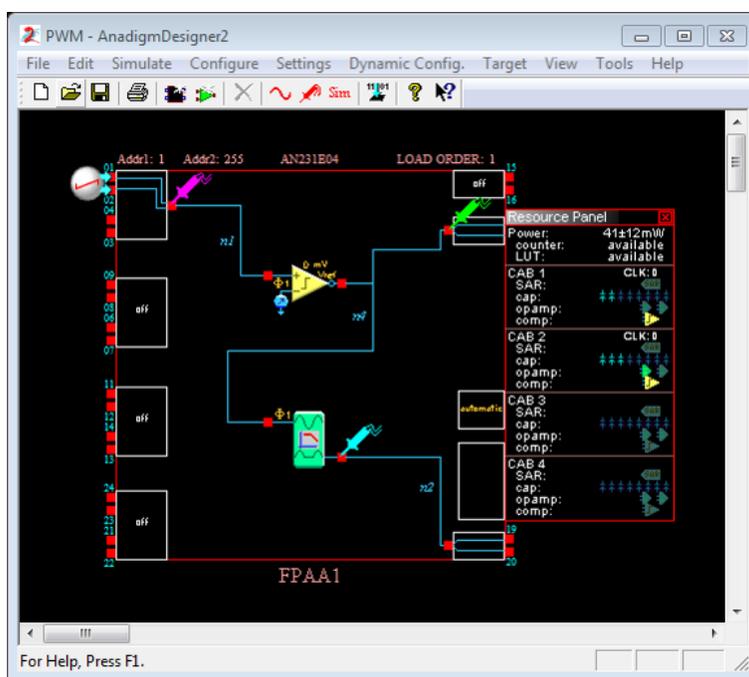
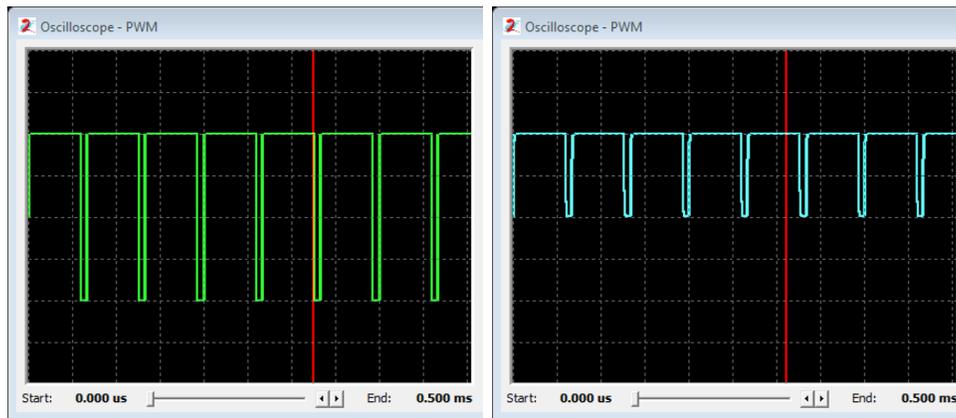


Figura A.22: Diseño de PWM en AD2

Para llevar a cabo la simulación, se activan 3 de las celdas I/O del FPAA, una como entrada y dos como salida. El valor para la amplitud de la señal de referencia se modifica en el campo *reference voltaje* del CAM *comparator*. Del circuito diseñado, se observan 2 salidas, la primera de ellas (*IOCell6*) muestra la señal PWM saturada a $\pm 2V$ para sus niveles lógicos alto y bajo respectivamente (Figura A.23a)). La segunda salida (*IOCell7*) presenta la señal PWM rectificada por medio del CAM *RectifierFilter*, limitando a esta en el rango $0 - 2V$ (Figura A.23b)). La elección de la salida PWM dependerá de la aplicación destino.

Tabla A.5: Configuración de los CAMs

CAM	Parámetro
Comparator	Clock A: Clock 0 @ 4MHz Compare To: Variable Reference Input Sampling: Phase 1 Output Polarity: Inverted Output Synch: None Reference Voltaje: 1
RectifierFilter	Clock A: Clock 0 @ 4MHz Rectification: Positive Half Wave Polarity: Non-inverting Input Sampling: Phase 1 Corner Frequency: 350 KHz Gain: 1



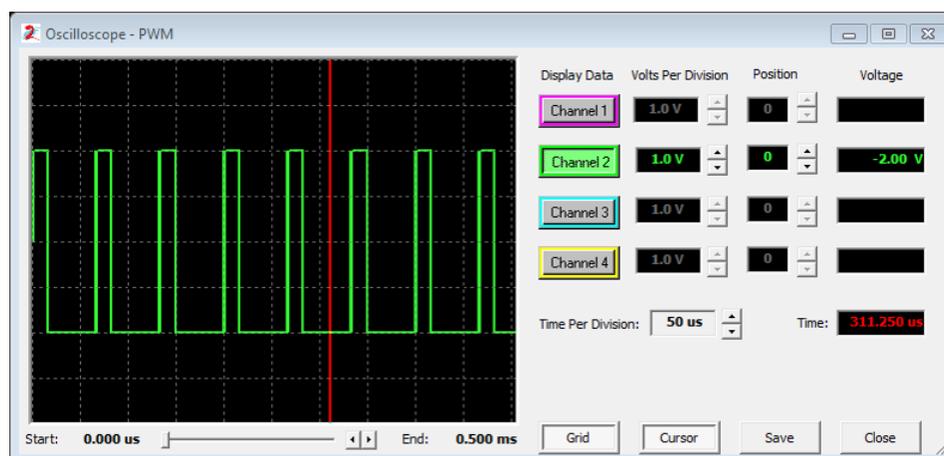
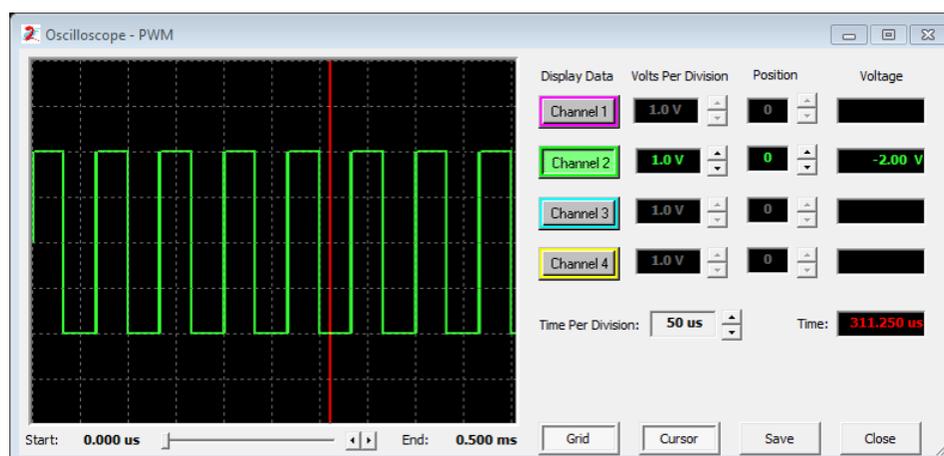
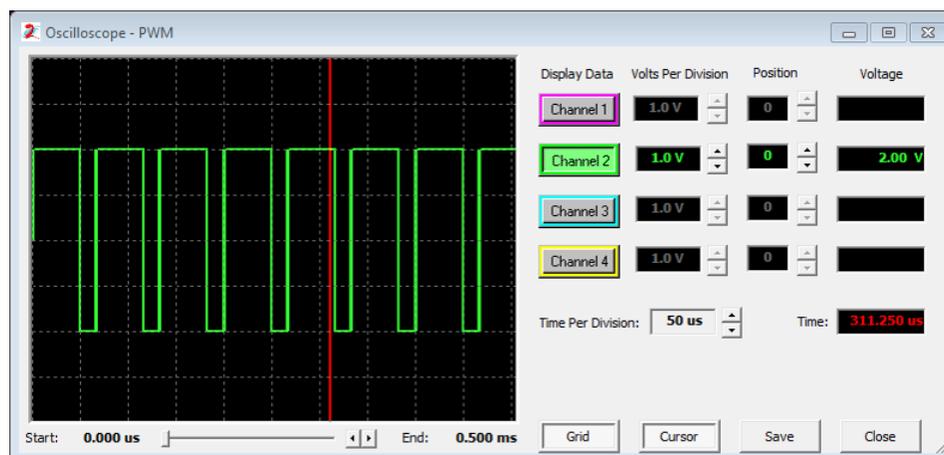
(a) Completa

(b) Rectificada

Figura A.23: Señal de salida PWM

Las Figuras A.24-A.26 muestran la respuesta de simulación ($1ms$) para diferentes índices de modulación.

Las Figuras A.27 y A.28 muestran el diseño de un PWM con 2 señales a diferente frecuencia. La señal portadora es una onda senoidal de $2,5V_p$ de amplitud a $12KHz$ de frecuencia. Por otra parte, la señal de referencia es una onda triangular de $2,5V_p$ de amplitud a $100KHz$ de frecuencia. El tiempo de simulación empleado es de $2ms$.

Figura A.24: Salida PWM con $M_a = 0,25$ Figura A.25: Salida PWM con $M_a = 0,50$ Figura A.26: Salida PWM con $M_a = 0,75$

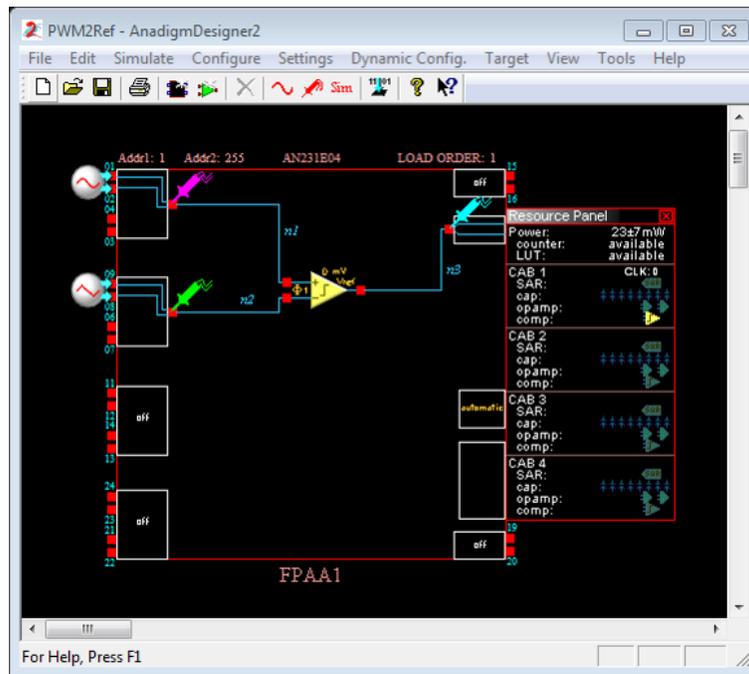


Figura A.27: Salida PWM con señales de diferente frecuencia

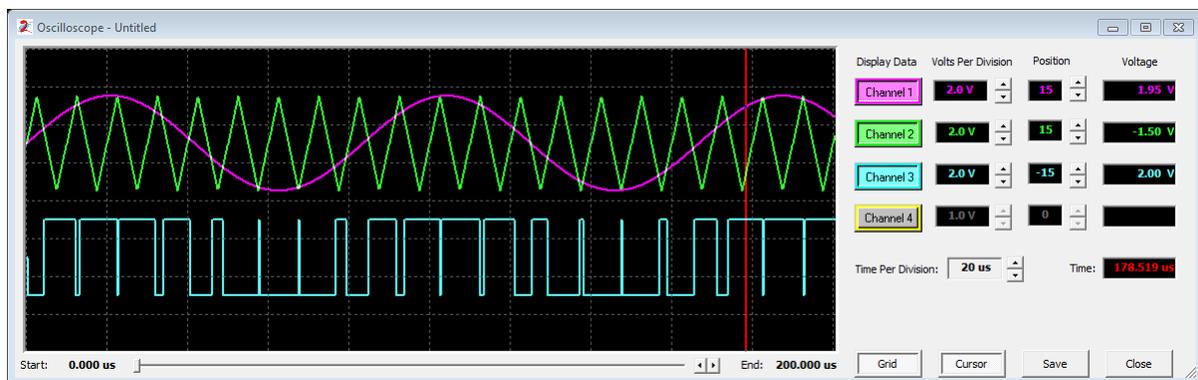


Figura A.28: Respuesta de la señal PWM

A.5. Filtros analógicos

Del conjunto de CAMs que ofrece AD2 se encuentran los filtros “pasa bajas”, “pasa altas” y “pasa banda” de 1^{er} y 2^{do} orden, los cuales sólo requieren de establecer el valor de los parámetros para la frecuencia de corte, la ganancia y el factor de calidad (Q). Este tipo de filtros se pueden conectar en cascada para obtener filtros de orden superior, sin embargo, no es una manera eficaz para su diseño, ya que por lo general requiere de efectuar cálculos manuales y del uso de capacitores adicionales para el filtrado. En respuesta a esta problemática, Anadigm incorpora dentro del entorno AD2 una herramienta que permite crear de manera automática el diseño

para este tipo de filtros. *AnadigmFilter* automatiza completamente el diseño y aplicación de filtros de orden superior en cuestión de segundos.

A.5.1. AnadigmFilter

Es la herramienta en la cual se diseñará el filtro de orden superior para después exportarlo al entorno de AD2. Con la finalidad de realizar el diseño del filtro de orden superior, AnadigmFilter establece los FPAAs que sean necesarios para su implementación y ubicará en ellos los CAMs de filtro de orden inferior junto con los parámetros indicados y sus conexiones.

A través de la opción “AnadigmFilter” del menú “Tools” en el entorno de AD2, se accede a la ventana principal de esta herramienta (Figura A.29). Las secciones presentes en la interfaz permiten visualizar la gráfica del filtro, establecer los valores para los parámetros del filtro, el chip en el que se insertará el filtro y la frecuencia del reloj de funcionamiento, elegir el tipo y la aproximación del tipo de filtro, así como la visualización de los parámetros de los CAMs utilizados.

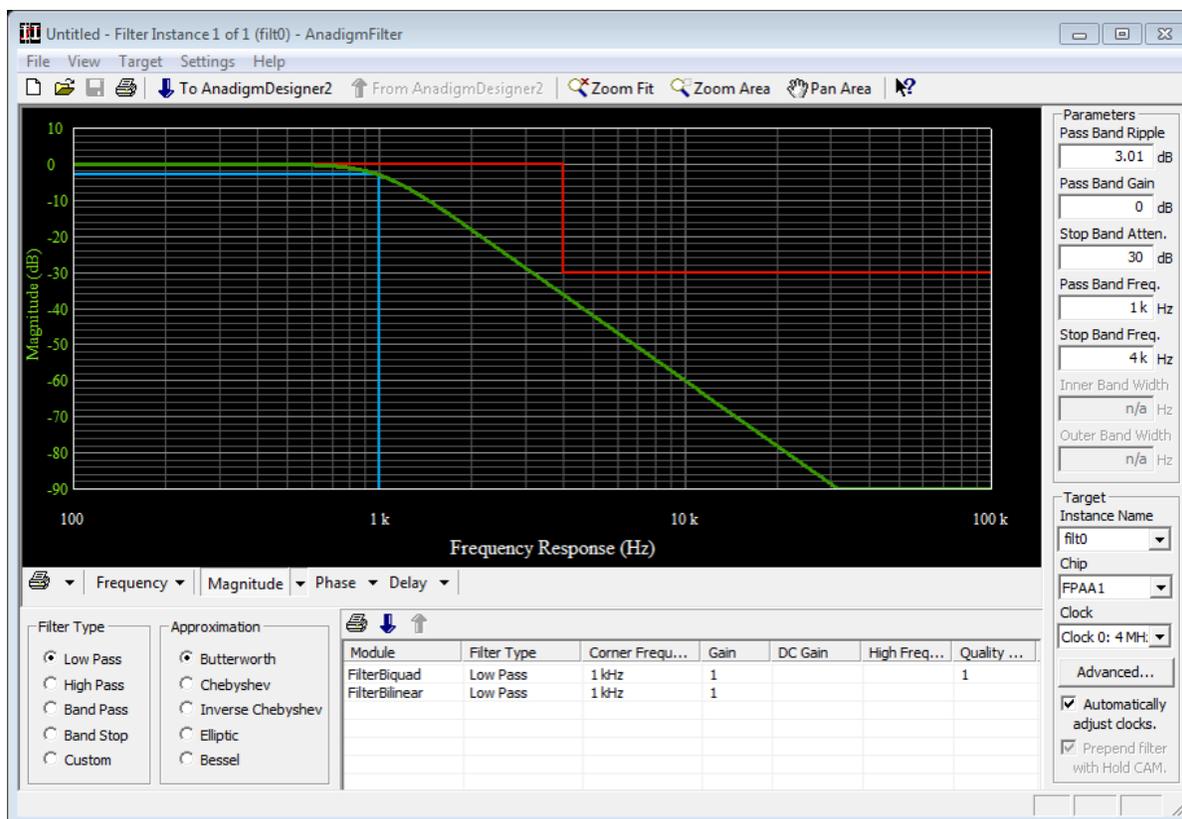


Figura A.29: Interfaz de la herramienta AnadigmFilter

A.5.1.1. Gráfica del filtro

En esta sección de la interfaz se observan 3 gráficas:

1. **Gráfica azul.** La información mostrada en la parte vertical y horizontal indica la frecuencia y el valor del rizo en el filtro pasa banda respectivamente.
2. **Gráfica roja.** La parte vertical indica la frecuencia de rechazo de banda. La parte horizontal superior representa la ganancia pasa banda del filtro y la horizontal inferior la atenuación en la banda de rechazo.
3. **Gráfica verde.** Muestra la respuesta del filtro, su trazo depende del tipo de filtro y de la aproximación seleccionada. A partir del menú “*Magnitude*”, esta gráfica se puede visualizar de dos formas, “magnitud vs frecuencia” y “voltaje vs frecuencia”.

El gráfico mostrado en pantalla permite la manipulación del valor en sus parámetros al mover las líneas correspondientes. Estos cambios se verán reflejados en la sección “*Parameters*” de la interfaz.

A.5.1.2. Parámetros del filtro

Esta sección de la interfaz se denomina “*Parameters*”, en la cual se permite configurar los diferentes parámetros del filtro (el rizo de pasa banda, la ganancia de pasa banda, la atenuación de rechazo de banda, la frecuencia de pasa banda, la frecuencia de rechazo de banda, el ancho de banda de pasa banda y el ancho de banda de rechaza banda). Los cuales dependen del tipo del filtro. El valor de cada parámetro se puede establecer hasta con 3 cifras significativas después del punto decimal. Al modificar alguno de estos parámetros, el gráfico del filtro se ve afectado ajustandose al valor asignado.

A.5.1.3. Características de la tarjeta

En esta sección (“*Target*”) se encuentran los parámetros de configuración utilizados para el funcionamiento adecuado del filtro en el FPAA.

1. ***Instance Name*.** Debido a que es posible implementar 2 o más filtros en un solo chip, es necesario separarlos por nombres para su identificación y configuración. La opción “*New Instance...*” permite el diseño de un nuevo filtro.
2. ***Chip*.** Por medio de la lista desplegable que presenta esta opción, se puede escoger el chip en el cual se desea colocar un filtro, razón por la que es posible utilizar más de un chip para crear un filtro orden superior o alguna otra aplicación.
3. ***Clock*.** Para el adecuado funcionamiento de un filtro y debido a que un FPAA cuenta con varias señales de reloj, es necesario seleccionar la fuente de reloj que el filtro utilizará en su implementación.
4. ***Advanced*.** El botón disponible para esta opción muestra una ventana que permite administrar la configuración del reloj y las instancias creadas para los filtros.

Si se desea que las señales de reloj se ajusten de forma automática y con la finalidad de optimizar los recursos del sistema, se deberá activar la casilla con la opción “*Automatically adjust clocks*”.

A.5.1.4. Tipo de filtro

De las opciones disponibles en el bloque “*Filter Type*”, se permite elegir el tipo de filtro a diseñar: pasa bajas, pasa altas, pasa banda y rechaza banda. La opción “*Custom*” permite establecer como predefinidas las características del último filtro configurado.

A.5.1.5. Tipo familia

En el bloque “*Approximation*” se elige el tipo de familia deseada para el filtro, de las opciones disponibles se tienen las aproximaciones por Butterworth, Chebyshev, Inverse Chebyshev, Elíptico y Bessel.

A.5.1.6. Parámetros del CAM

En esta sección de la interfaz de AnadigmFilter se observan los CAMs y el valor de sus parámetros que se requieren para la implementación del filtro diseñado. Además de esto, se observan 3 íconos con funciones específicas.

1. ***Print List***. Representada por el ícono de una impresora. Permite generar una lista que se puede imprimir con el nombre de los CAMs a implementar y sus parámetros internos.
2. ***Send to AnadigmDesigner2***. La función de esta opción es enviar el filtro diseñado dentro de la interfaz de AnadigmFilter hacia AD2 con lo cual se generan los CAMs y las conexiones internas necesarias para el funcionamiento del filtro. Su ícono es una flecha color azul apuntando hacia abajo de la ventana.
3. ***Update from AnadigmDesigner2***. Los cambios realizados en algún CAM del filtro dentro de la interfaz de AD2, se pueden ver reflejados en AnadigmFilter al pulsar el ícono con una flecha de color azul apuntando hacia arriba.

A.5.1.7. Análisis de datos

Por medio del menú “*File/Export/Frequency Response Data (.csv)*”, se pueden guardar los datos obtenidos de la respuesta del filtro. El archivo “.csv” creado contiene 2 columnas, una con el valor de la frecuencia (Hz) y la otra con la magnitud obtenida (dB).

A.5.1.8. Exportar e imprimir

Con la finalidad de almacenar la información del filtro de manera gráfica, se accede al menú “*File*” de 2 maneras: “*File/Export/Filter Design Report (.html)*” o “*File/Print/Print Filter Design Report*”, lo cual permite generar un archivo para impresión o su almacenamiento en PDF.

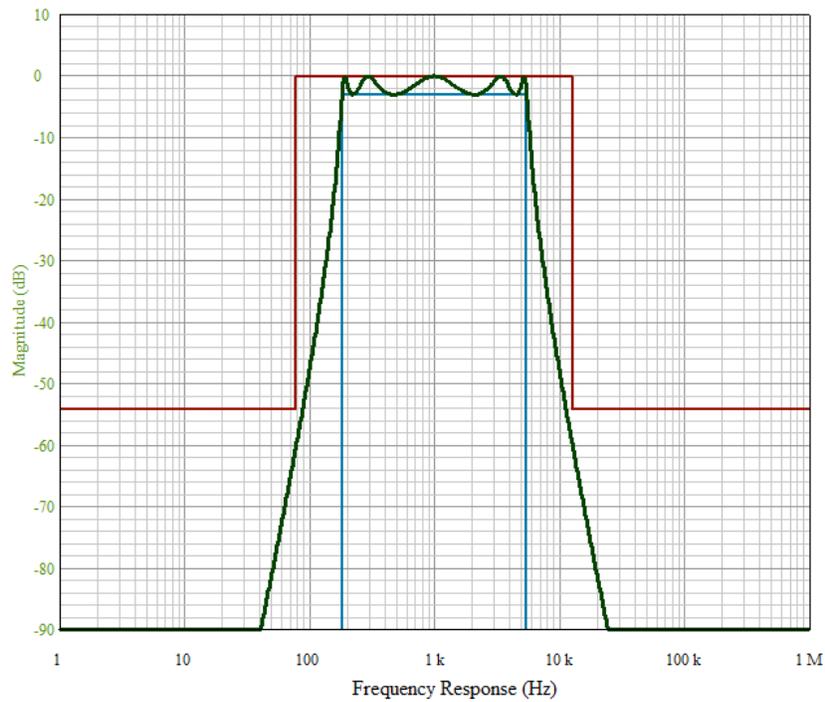


Figura A.30: Ejemplo del gráfico almacenado de un filtro

A.5.2. Ejemplos de diseño de filtros

Los siguientes ejemplos de filtros son diseñados con la ayuda de AnadigmFilter.

A.5.2.1. Filtro pasa altas

Este tipo de filtro atenúa las señales cuya frecuencia es menor que la frecuencia central f_c . Para observar este comportamiento, se diseña el filtro en base a las características mostradas en la Tabla A.6.

Tabla A.6: Parámetros para filtro pasa altas

Parámetro	Valor
Filter Type	High Pass
Approximation	Chebyshev
Pass Band Ripple	3 dB
Pass Band Gain	1 dB
Stop Band Attenuation	30 dB
Pass Band Frequency	250 KHz
Stop Band Frequency	50 KHz
Clock	Clock 0

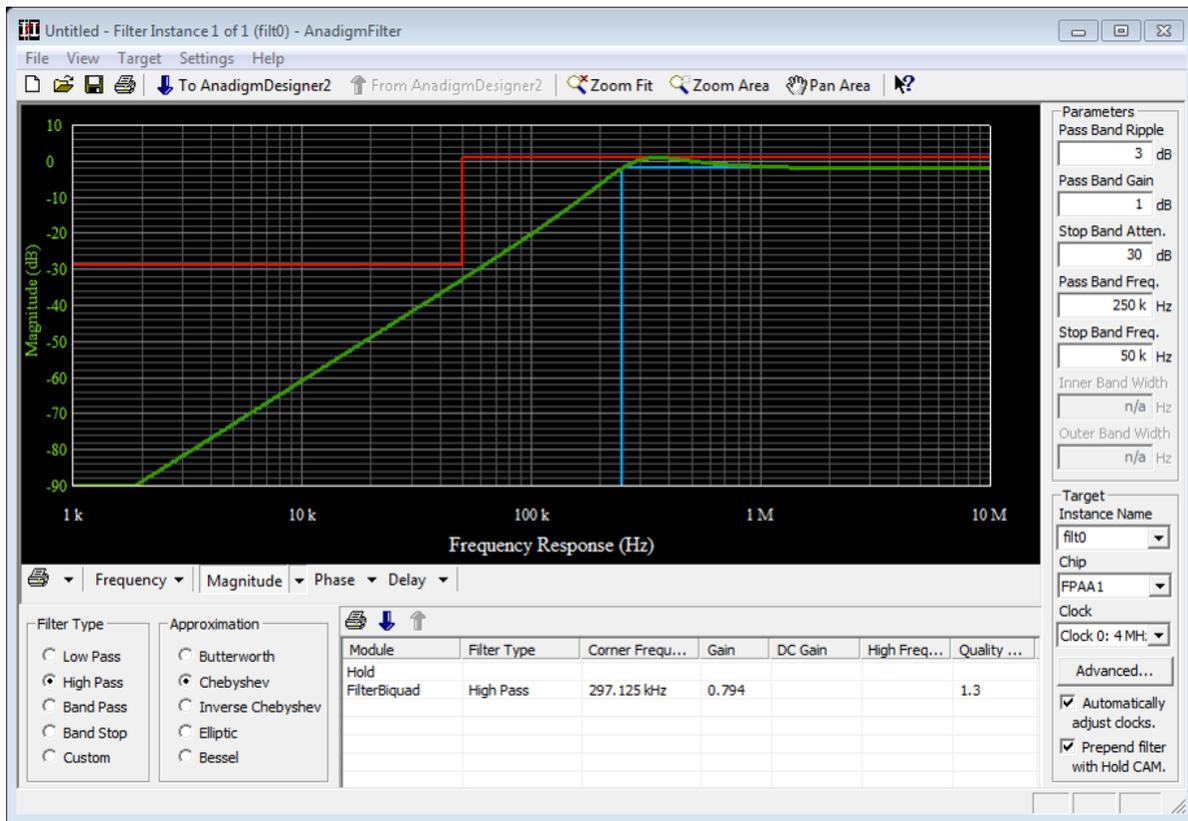


Figura A.31: Diseño del filtro pasa altas

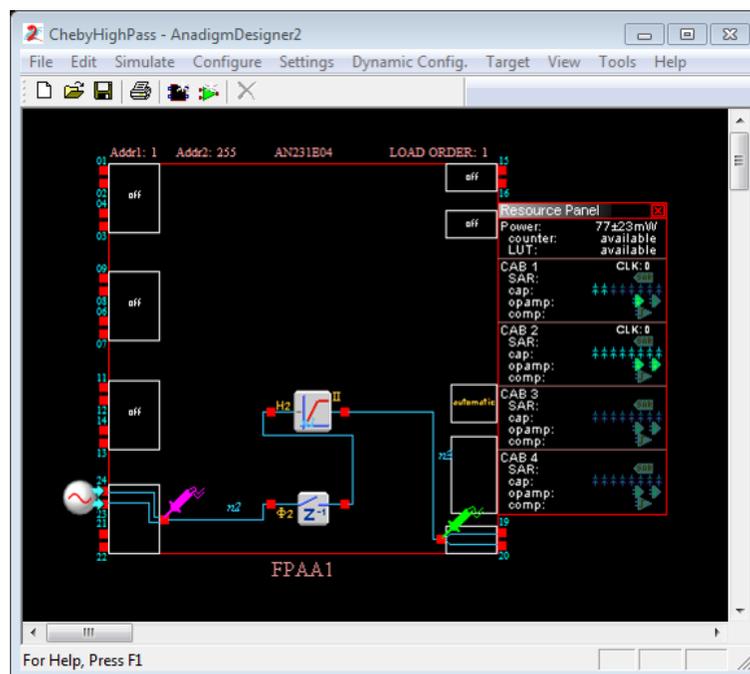


Figura A.32: Circuito del filtro pasa bandas creado en AD2

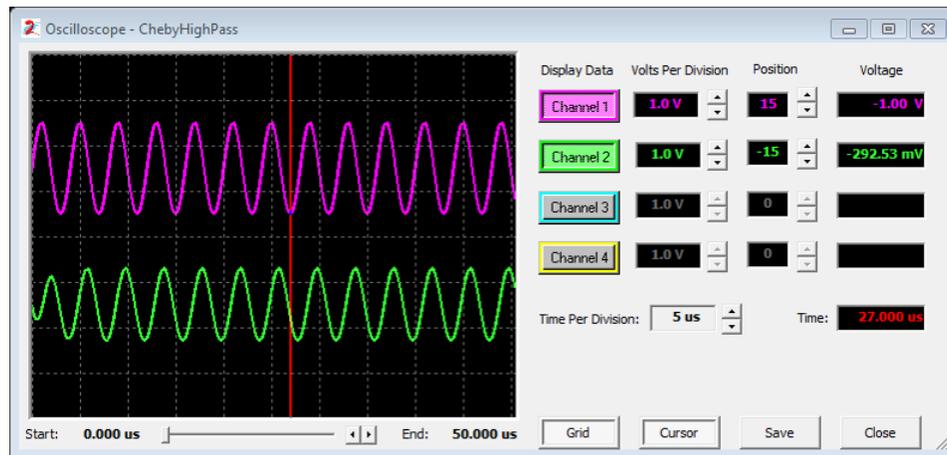


Figura A.33: Simulación del filtro pasa altas con $f = 250\text{KHz}$

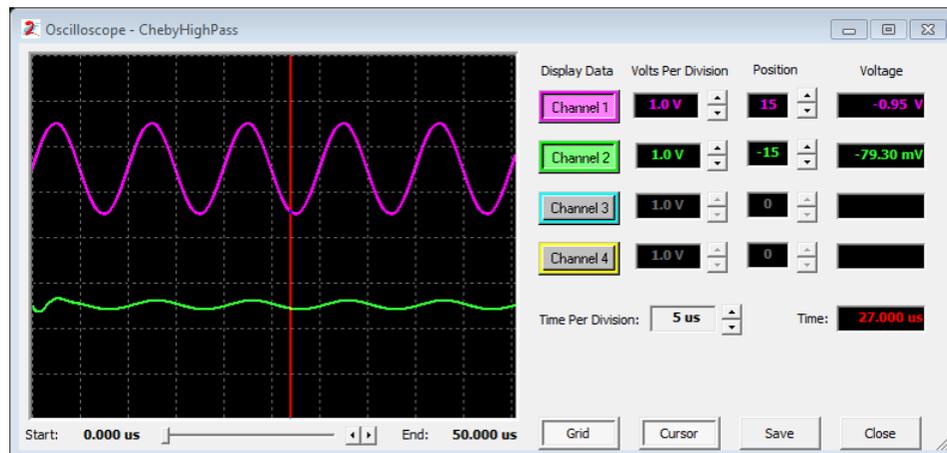


Figura A.34: Simulación del filtro pasa altas con $f = 100\text{KHz}$

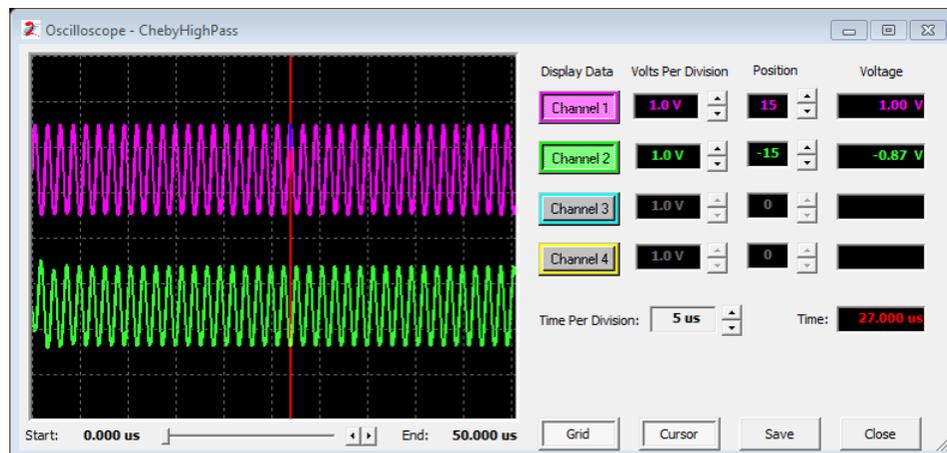


Figura A.35: Simulación del filtro pasa altas con $f = 750\text{KHz}$

A.5.2.2. Filtro pasa bandas

Al diseñar el filtro pasa bandas con las características mostradas en la Tabla A.7, se obtiene la respuesta mostrada en la Figura A.36.

Tabla A.7: Parámetros para filtro pasa bandas

Parámetro	Valor
Filter Type	Band Pass
Approximation	Butterworth
Pass Band Ripple	3 dB
Pass Band Gain	1 dB
Stop Band Attenuation	30 dB
Center Frequency	24 KHz
Stop Band Width	6 KHz
Pass Band Width	3 KHz
Clock	Clock 0

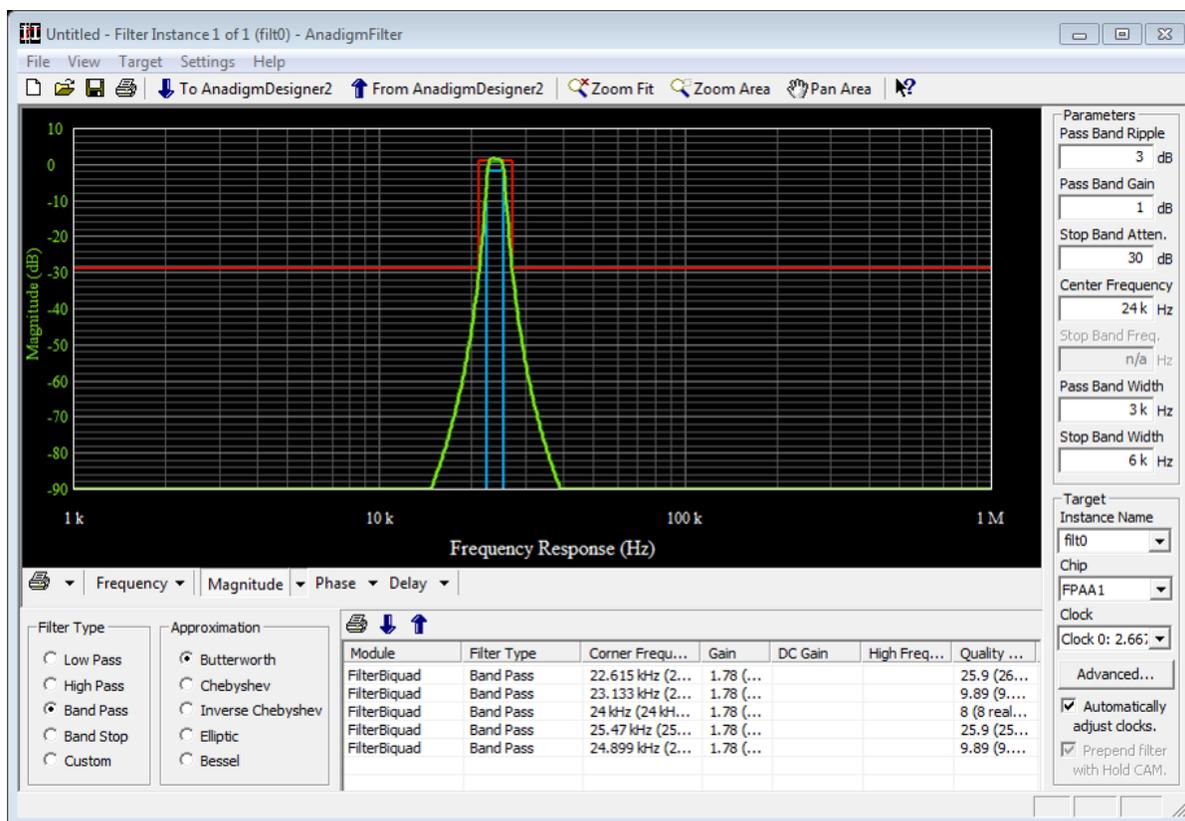


Figura A.36: Diseño del filtro pasa bandas

Para la simulación del circuito creado (Figura A.37), se activan las celdas 4 y 7 como entrada y salida respectivamente. Se añaden 2 puntas de medición en las celdas I/O habilitadas y un generador de funciones configurado con una señal senoidal de $2V_{PP}@24KHz$ (*offset* de modo común a $0V$). Como es de esperarse, un filtro de este tipo sólo permite el paso de las frecuencias contenidas en la banda formada por los valores “*Pass Band Width*” y “*Stop Band Width*”, teniendo como valor principal la “frecuencia central”, las frecuencias restantes son atenuadas.

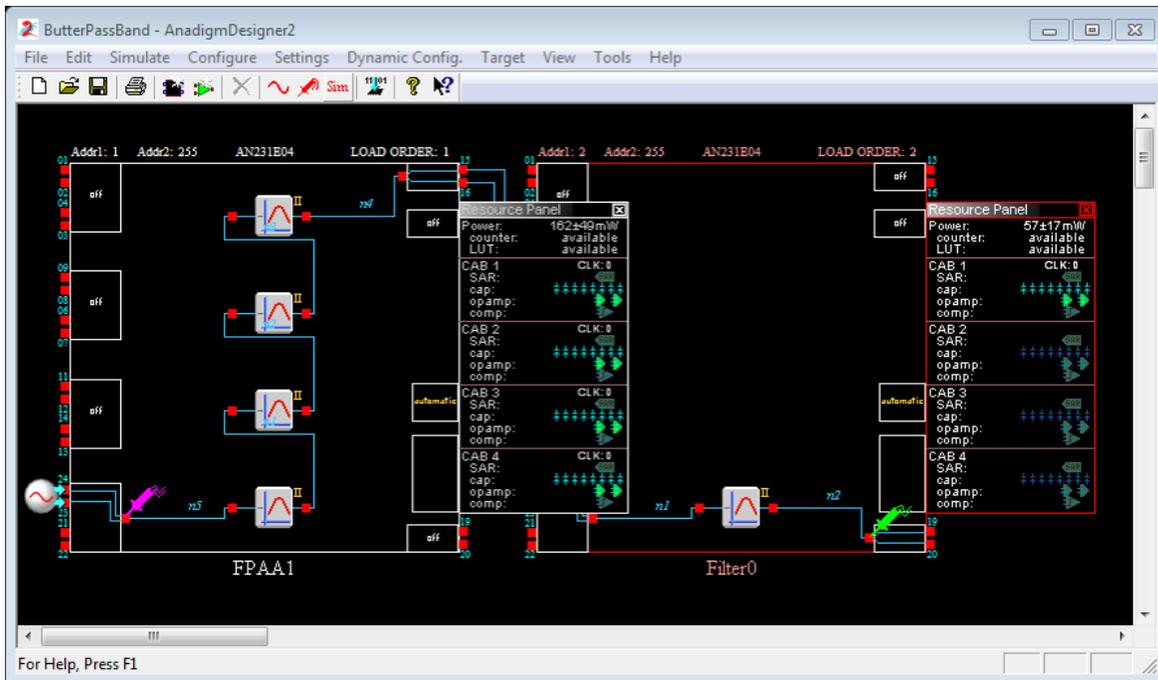


Figura A.37: Circuito del filtro pasa bandas creado en AD2

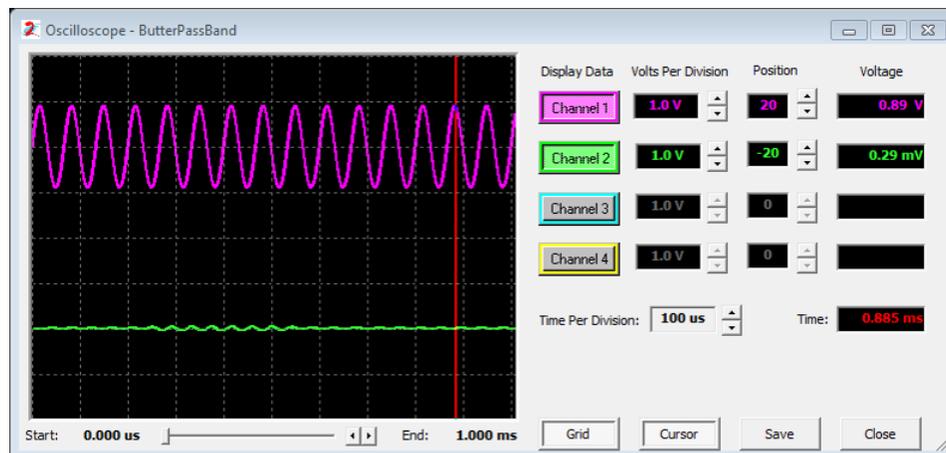


Figura A.38: Simulación del filtro pasabandas con $f = 15KHz$

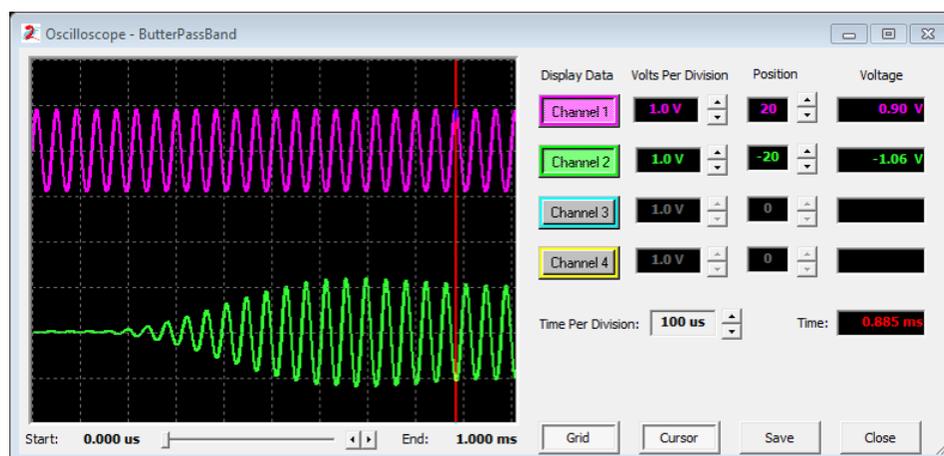


Figura A.39: Simulación del filtro pasabandas con $f = 24KHz$

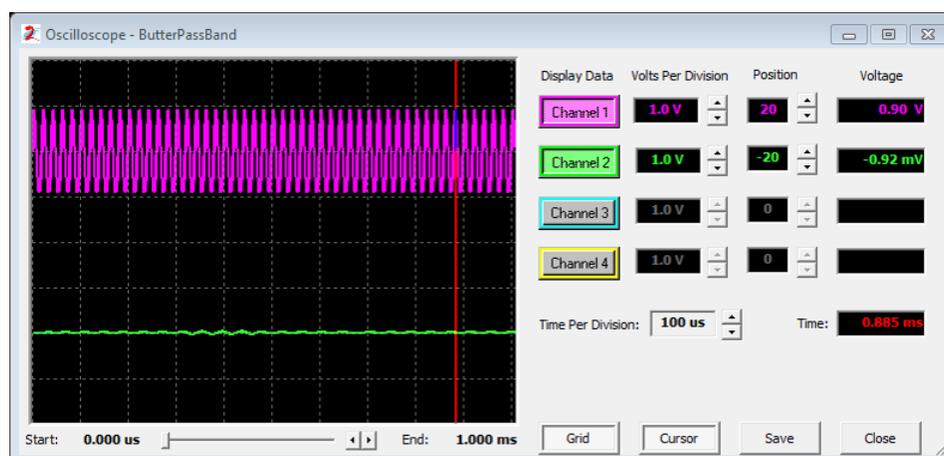


Figura A.40: Simulación del filtro pasabandas con $f = 50KHz$

A.6. AnadigmPID

Para llevar a cabo una implementación rápida y automática de un controlador PID dentro del entorno de AD2 se cuenta con la herramienta AnadigmPID, a la cual se accede a través del menú “Tools”.

AnadigmPID no tiene soporte para la implementación automática de controladores PID en FPAAs de tercera generación, por lo que de ser necesario, se implementa sobre un FPAA de segunda generación y se traslada al chip requerido [5].

Al acceder a la interfaz de AnadigmPID, se observa el cuadro de diálogo mostrado en la Figura A.41, en la cual se permite seleccionar o añadir el chip FPAA en el cual se insertará el controlador PID. Después de seleccionar el chip a utilizar, AnadigmPID inserta los CAMs

necesarios en la interfaz de AD2 para su análisis, simulación e implementación.

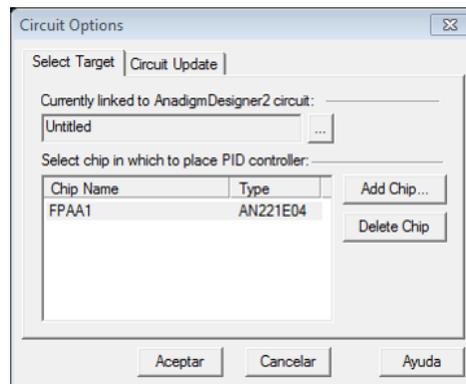


Figura A.41: Cuadro de diálogo para los opciones del circuito

La interfaz correspondiente a esta herramienta se muestra en la Figura A.42, la cual cuenta con 7 pestañas elementales para el diseño del controlador.

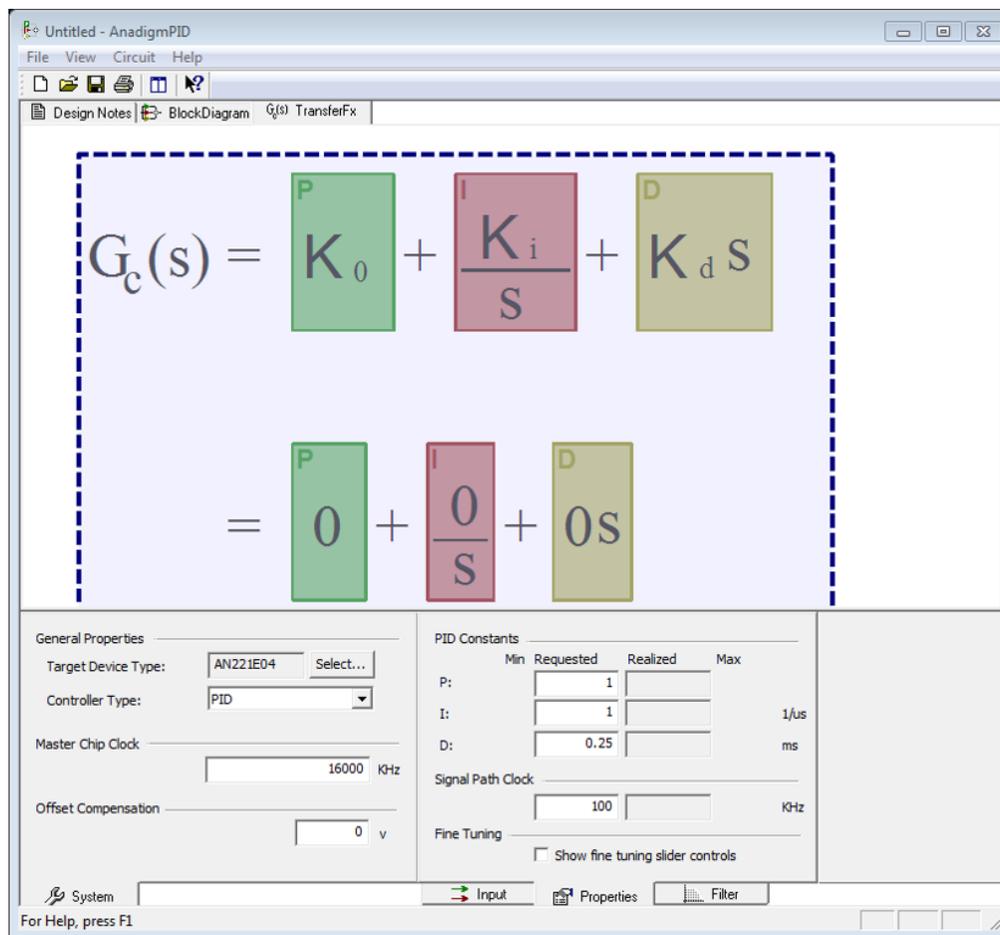


Figura A.42: Interfaz AnadigmPID

A.6.1. Pestañas

1. **Design Notes.** Brinda información respecto a las ecuaciones de las constantes del controlador.
2. **Block Diagram.** Muestra el diagrama a bloque del sistema en lazo cerrado a implementar.
3. **$G_c(s)$ TransferFx.** Se observa la ecuación correspondiente a la función del control PID diseñado.
4. **System.** Esta pestaña permite configurar las características del sistema, tales como la familia del FPAA utilizado, el tipo de controlador PID y la frecuencia del reloj principal.
5. **Input.** Permite el control sobre el valor de las entradas al circuito del controlador, amplificando o invirtiendo las señales de retroalimentación a la planta (PV) y *setpoint* (SP). También existe la opción para implementar un SP interno, con lo que AnadigmPID coloca un voltaje de referencia en el interior del FPAA.
6. **Properties.** Mediante esta pestaña se establecen los valores de las constantes asociadas al circuito de control, K_P , K_I y K_D . Los valores deseados se introducen en la columna “Requested”, en base a ello, AnadigmPID establece los valores de ajuste en la columna “Realized”. Los rangos permitidos para cada una de las ganancias no son del todo independientes el uno del otro. En particular, la frecuencia deseada para la señal de reloj (*Signal Path Clock*), establece la frecuencia a la cual trabajarán los CAMs del controlador. Al cambiar este valor, se aprecia un cambio en los CAMs y en su rango de respuesta. Al hacer clic en cualquiera de los hipervínculos (texto subrayado en color azul), se abrirá la ventana de ayuda. La opción para el ajuste fino (*Fine Tuning*) muestra una ventana con 4 barras que permiten el ajuste de las ganancias y la señal de reloj.
7. **Filter.** Permite incluir un filtro pasa bajas bilineal o bicuadrático en la salida del controlador. Por lo general, la constante del filtro se especifica mediante la frecuencia de corte F_o .

A.6.2. Control PID de una planta

En este ejemplo de aplicación, se diseña un controlador PID (Figura A.43) en un FPAA AN221E04 mediante la herramienta AnadigmPID, posteriormente, se trasladan los CAMs hacia un chip AN231E04. El valor de las ganancias para la parte proporcional, integral y derivativa son respectivamente $K_p = 30$, $K_i = 10$ y $K_d = 25$. La planta a controlar es una función arbitraria, que se modela mediante el CAM *TransferFunction* que permite implementar una función no lineal mediante una LUT, los datos que modelan el comportamiento de dicha planta son los mostrados en la Tabla A.3.

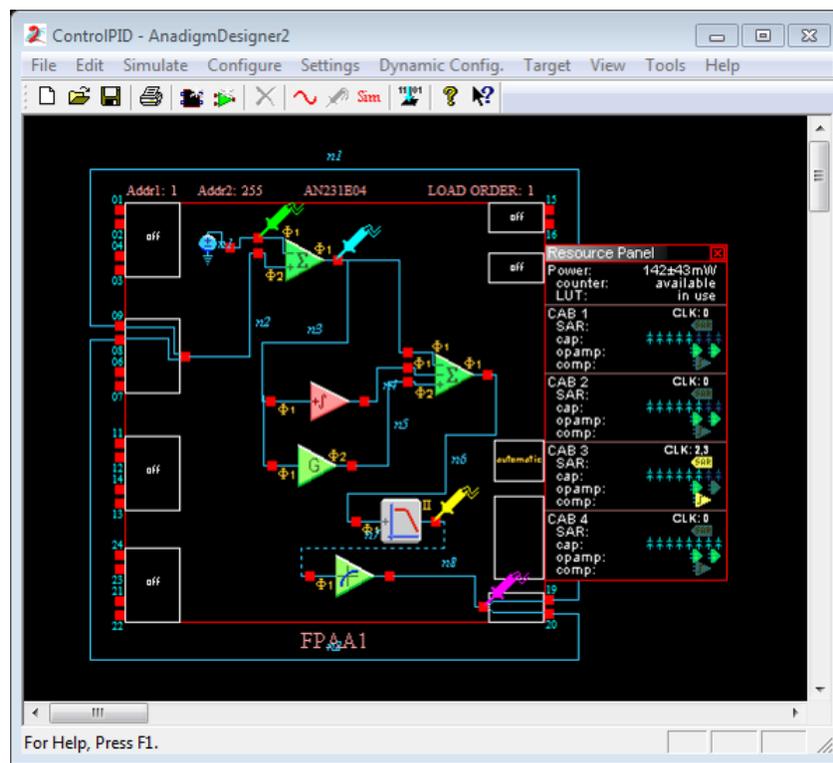


Figura A.43: Control PID en AD2

La Figura A.44 muestra la respuesta del controlador PID aplicado a la señal arbitraria en un tiempo de simulación de $5ms$. Con lo cual se observa que el controlador PID actúa de manera adecuada sobre el sistema, permitiendo la estabilidad en estado estacionario a un valor SP de $2V$.

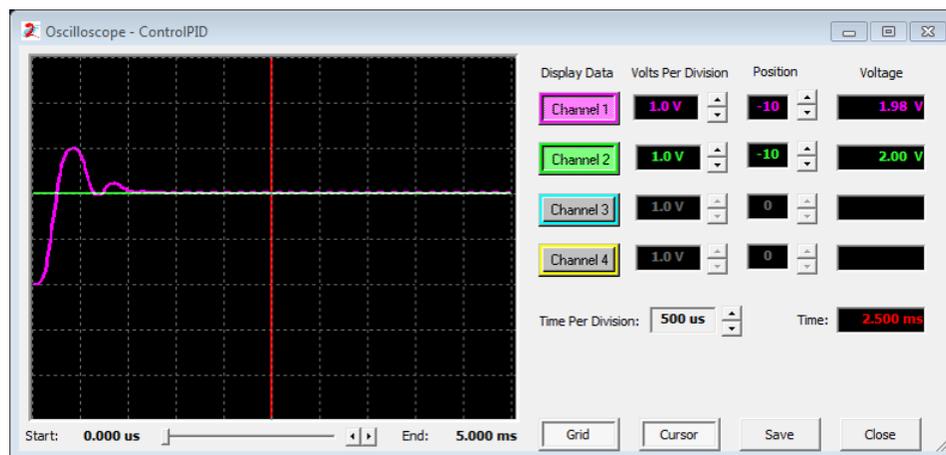


Figura A.44: Respuesta de simulación al controlador PID con un sistema en LUT

Apéndice B

Controladores PID digitales

Para el diseño de este tipo de controladores, se utilizó la tarjeta de desarrollo Nexys 3 (Figura B.1) de Xilinx [47], la cual dentro de los recursos disponibles cuenta con un FPGA Spartan 6 *XC6SLX16*, un reloj de *100 MHz* y 4 sockets para módulos de expansión (Pmod *Connectors*) útiles para interfaz con periféricos de entrada/salida externos.

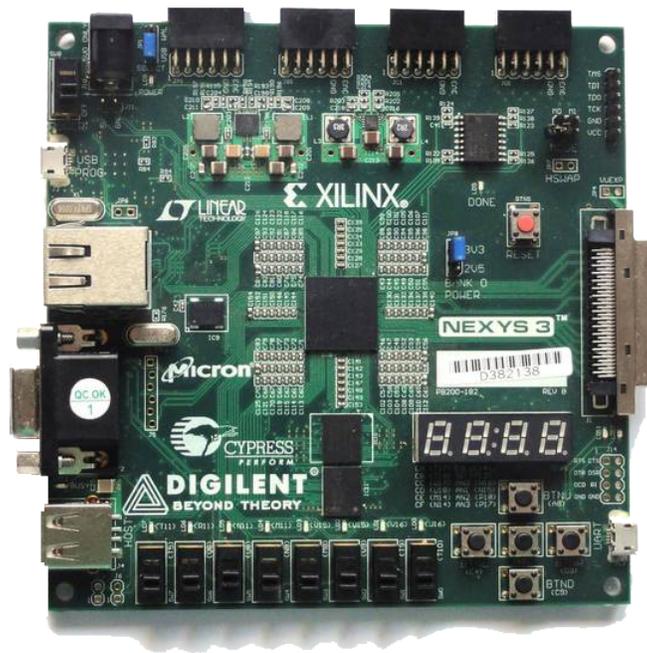


Figura B.1: Tarjeta de desarrollo Nexys 3

Las siguientes etapas se consideran necesarias para la implementación de los controladores PID en el FPGA.

1. Acondicionamiento de las señales de entrada y salida
2. Discretización de los parámetros a procesar

3. Cálculo del error y del valor de la señal de control
4. Conversión de valores digitales a analógicos

B.1. Acondicionamiento de las señales

Al igual que en el caso del FPAA, las señales provenientes y emitidas hacia la planta de temperatura necesitan ser adecuadas a los valores permitidos por el FPGA, para ello se tiene que los sockets para los Pmod se alimentan con un voltaje de 3.3 V proveniente de la tarjeta. Con la finalidad de proteger la tarjeta Nexys 3 y los módulos a utilizar, se opta por el acondicionamiento de los voltajes de entrada y salida en un rango de 0 a 3 V , por lo cual el circuito de acondicionamiento utilizado en la implementación con el FPAA es de utilidad para este caso.

B.2. Módulo del ADC

B.2.1. DIGILENT Pmod

El módulo utilizado para la conversión de valores analógicos a digitales es el PmodAD1 de Digilent, el cual contiene 2 ADC de 12 bits, **AD7476A**, que manipulan voltajes analógicos en el rango $0 - 3,3\text{V}$. Para el envío de datos del módulo hacia el dispositivo maestro, cuenta con el bus SPI/MICROWIRE que trabaja con frecuencias de hasta 20MHz [46].

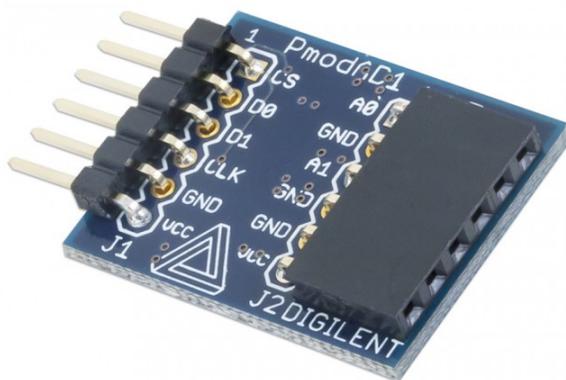


Figura B.2: PmodAD1 Digilent

B.2.1.1. Pmod, interfaz tipo 2

Esta interfaz proporciona un puerto SPI. El dispositivo al cual se conecta generalmente actúa como un dispositivo SPI maestro mientras que el Pmod funge como dispositivo SPI esclavo. Cuando este módulo se coloca en un conector de 12 pines, se debe conectar en los pines 1-6, es decir, la fila superior del conector [45].

Tabla B.1: Configuración para los pines SPI

Pin	Dirección de la señal
1	SS Out
2	MOSI Out
3	MISO In
4	SCK Out
5	GND
6	VCC

SS *Slave Select*. Activo en bajo para activar al dispositivo esclavo
MOSI *Master Out Slave In*. Datos del dispositivo maestro al esclavo
MISO *Master In Slave Out*. Datos del dispositivo esclavo al maestro
SCK *Serial clock*. Señal de reloj del dispositivo maestro al esclavo

B.2.2. AD7476A

Como se mencionó anteriormente, el Pmod requerido para el ADC, contiene 2 integrados AD7476A, los cuales requieren de 16 ciclos de reloj para llevar a cabo la conversión de un dato analógico a digital (Figura B.3). El valor correspondiente para los tiempos de conversión fue obtenido de la tabla *Timing Specifications* que provee el fabricante de los convertidores. La trama de bits leída consta de 4 ceros y 12 bits de dato válidos [43].

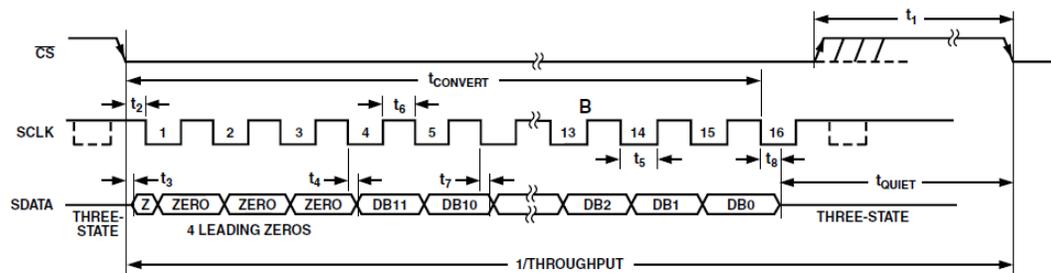


Figura B.3: Diagrama de tiempos para el convertidor AD7476A

De acuerdo al diagrama mostrado en la Figura B.3, se observa que la trama de conversión de datos inicia cuando la señal CS (*Chip Select*) se establece en bajo. A partir de ello, se genera la señal de reloj correspondiente y en cada uno de sus flancos de subida se obtiene un bit del dato. Pasado los 16 ciclos de reloj requeridos, la señal CS es desactivada. Estas señales son generadas en el FPGA, el cual funge como dispositivo maestro.

La señal de reloj utilizada para los módulos a diseñar es la proveniente del reloj del FPGA que trabaja a $100MHz$, previo a ello, esta señal se adecua a una frecuencia de $50MHz$, la cual se modifica por medio del divisor que incluye el bloque DCM de Xilinx, con la finalidad de evitar desfases en la misma y lograr que llegue de manera sincrónica a todos los bloques.

El módulo utilizado para adquirir la información del ADC fue diseñado como una máquina de estados finitos (FSM, *Finite State Machine*) que trabaja a $50MHz$, la cual es representada en la Figura B.4.

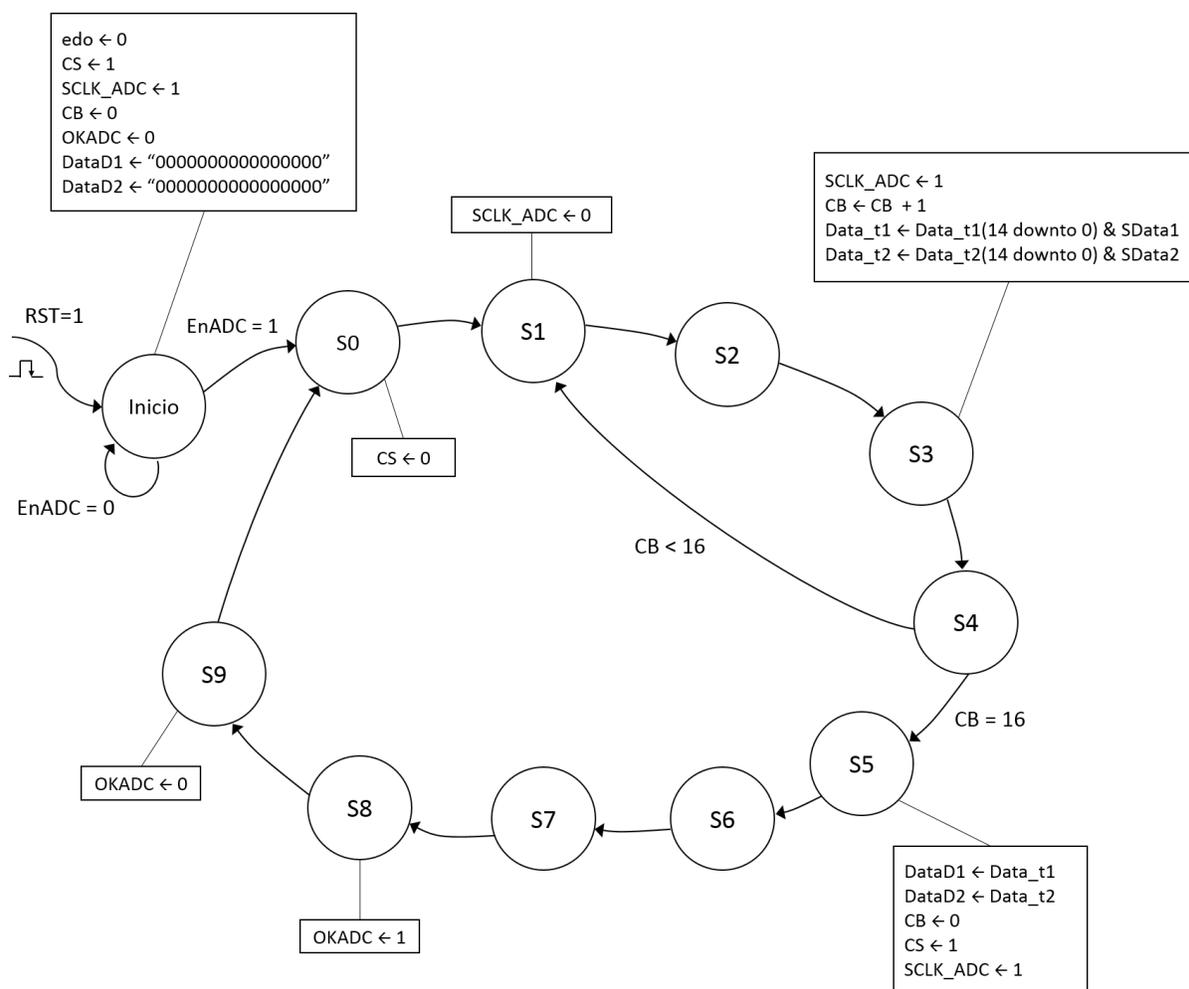


Figura B.4: FSM para el módulo del ADC

El objetivo de este módulo es indicar al ADC que inicie una conversión y obtener los resultados que han sido generados, los cuales están presentes en las señales de entrada SData1 y SData2, respectivamente.

La señal de reloj tiene un periodo de $80ns$, y es generada a partir de la señal proveniente del DCM. Esto origina una frecuencia de $12,5MHz$ para el SCLK, la cual está en el rango permitido para operar con el ADC.

Finalmente, la variable OKADC indica cuando la conversión ha sido completada.

La Figura B.5 muestra el resultado de la simulación hecha con la herramienta *ISim 13.2* de *Xilinx*, del módulo del ADC, donde se aprecia el comportamiento de las señales generadas.

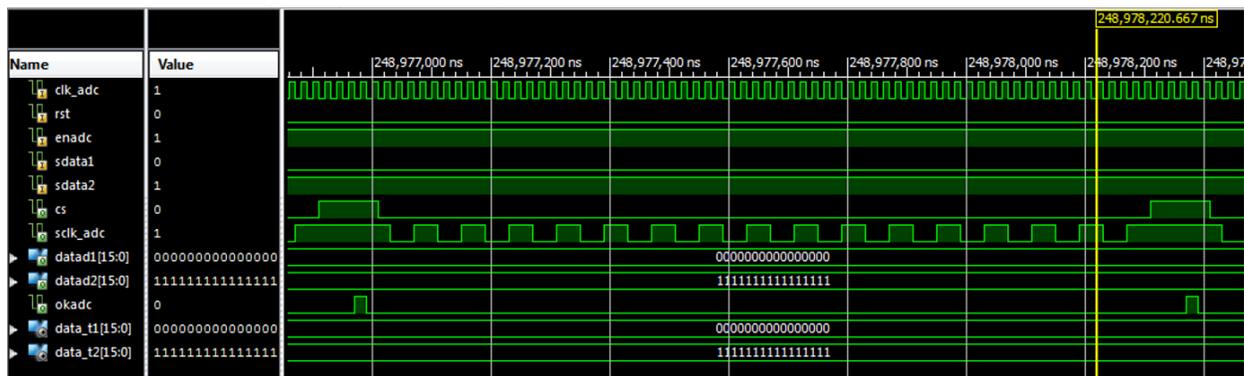


Figura B.5: Simulación del módulo del ADC

B.3. Módulo del DAC

Para convertir los resultados del controlador a su valor analógico correspondiente, se utiliza el PmodDA2 de digilent, el cual consta de 2 DAC de 12 bits, **DAC121S101**. Los voltajes analógicos de salida oscilan en el rango 0 – 3,3V. Al igual que el ADC, la interfaz de comunicación con el dispositivo maestro es por medio del bus SPI/MICROWIRE [44].

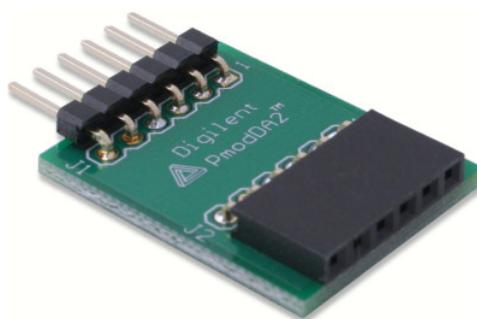


Figura B.6: PmodDA2 Digilent

Las especificaciones para el diseño del módulo del DAC son las mismas que se consideraron en el diseño del módulo ADC, por ejemplo, la señal de reloj aplicada al módulo para la conversión digital es el proveniente del DCM. La diferencia entre ambos diseños radica en el diagrama de tiempos empleado. Para el diseño del presente módulo se considera el diagrama de tiempos mostrado en la Figura B.7 [58].

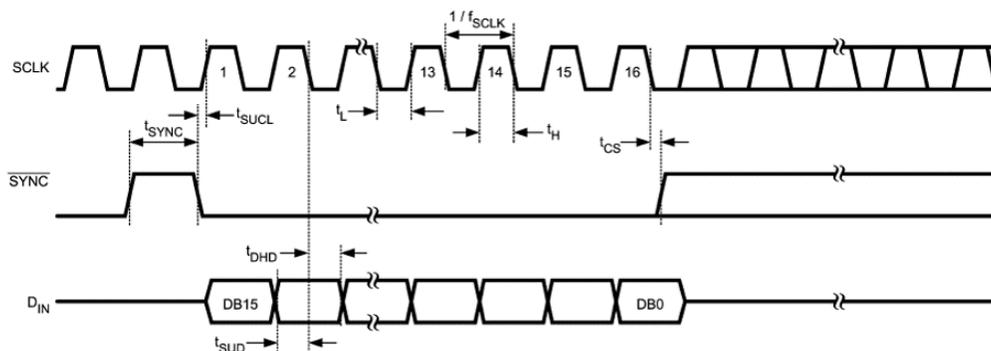


Figura B.7: Diagrama de tiempos para el convertidor DAC121S101

Con base en el diagrama de tiempos de la Figura B.7, se modela la FSM correspondiente al módulo del DAC (ver Figura B.8).

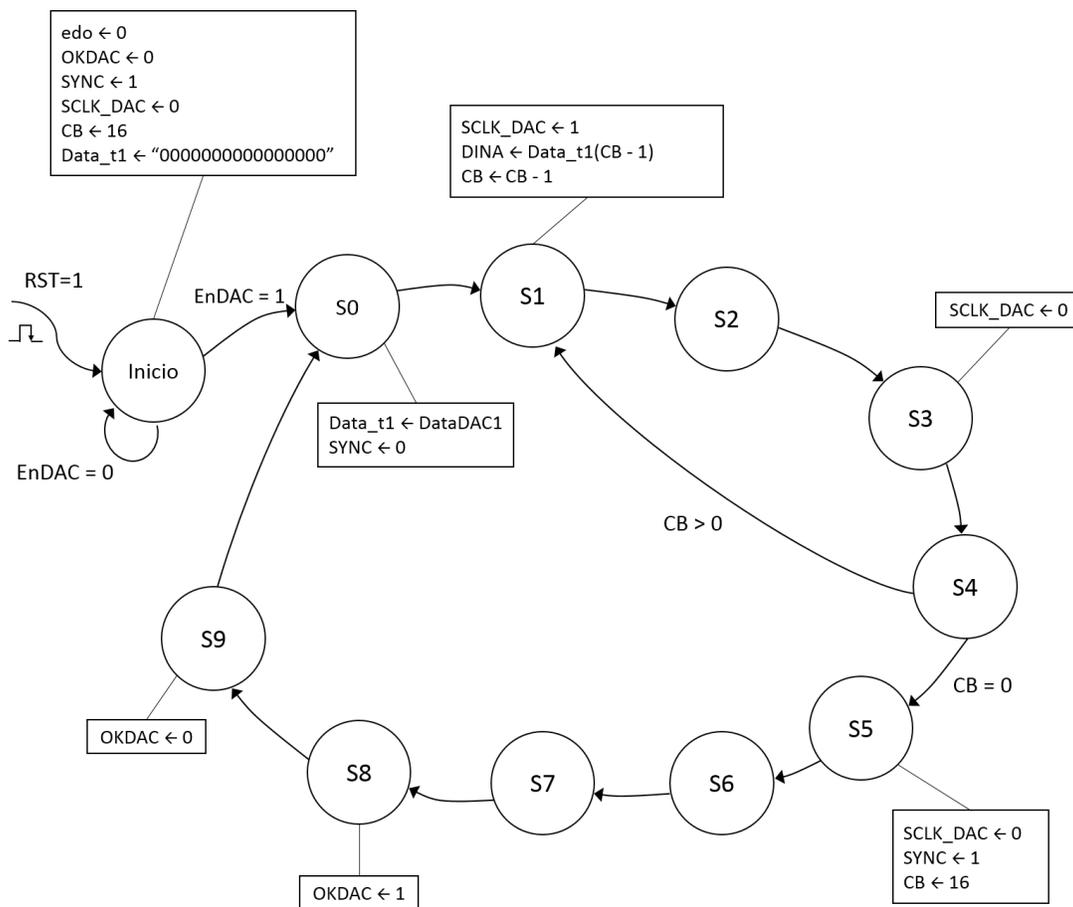


Figura B.8: FSM para el módulo del DAC

El objetivo de este módulo es enviar una trama de 16 bits al dispositivo esclavo (Pmod-DA2) para su conversión a un valor analógico. La variable Data_t1 almacena de manera

temporal el valor digital presente en la entrada del PmodDA2. Una vez ejecutados los 16 ciclos de reloj correspondientes para la conversión, la salida del PmodDA2 tendrá listo el valor analógico correspondiente a la señal de control aplicada, además la variable OKDAC, que indica al módulo de control la ejecución exitosa de dicha acción.

La Figura B.9 muestra el comportamiento en simulación de las señales generadas por el módulo del DAC. El proceso de conversión da inicio cuando la señal de sincronización (*SYNC*) cambia a nivel bajo después de t_{SYNC} ns de estabilidad en alto. La frecuencia de trabajo de la señal *SCLK_DAC* es de $12,5MHz$.

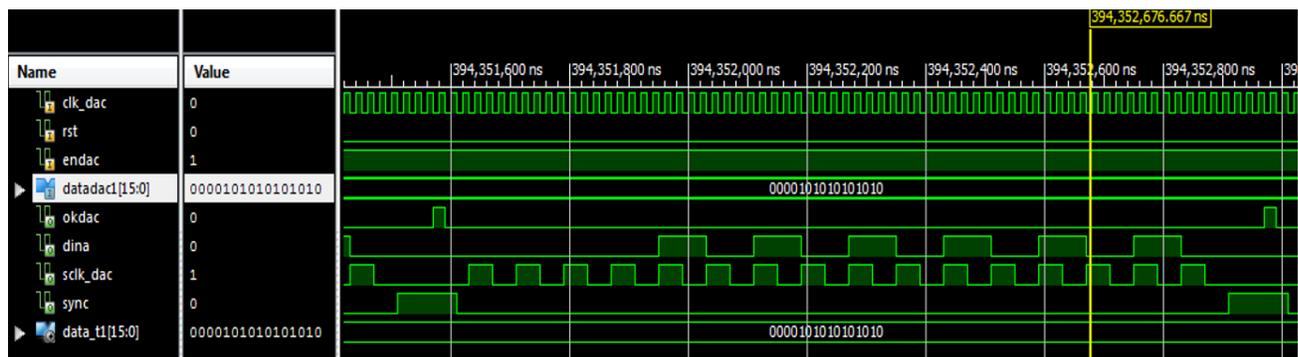


Figura B.9: Simulación del módulo del DAC

B.4. Módulo de control

Compuesto por los submódulos ProporcionalC y EnModule. El primero de ellos, el cual se nombrará “control maestro”, controla a los módulos antes descritos, es decir, indica al ADC (*ADCModule*) cuando iniciar una conversión de datos, una vez concluida la conversión (cuando la señal OKADC es activa), el control maestro efectúa la resta de los valores provenientes del SP y Feedback y, aplicar la ley de control correspondiente al error obtenido (resta). Posterior a estos cálculos, el módulo DAC (*DACModule*) será habilitado para efectuar la conversión del valor obtenido, el control maestro da inicio a la conversión de un nuevo dato.

La FSM mostrada en la Figura B.10, describe el comportamiento del módulo antes citado.

De la FSM mostrada, se observa la definición de la constante proporcional, K_P , como un vector binario de valor $(10000)_2$ equivalente a 16 decimal, el cual permite implementar la ley de control proporcional sobre el sistema. Notese también que en la descripción del módulo, se activan las salidas pertinentes para habilitar los módulos ya sea para el ADC, Control o DAC, permitiendo con ello llevar el orden de las operaciones dentro del sistema. En los estados 5 a 9, el control maestro efectúa las operaciones correspondientes a la ley de control aplicada, teniendo el tiempo suficiente para efectuarlas sin problema ($20ns$ por estado). En

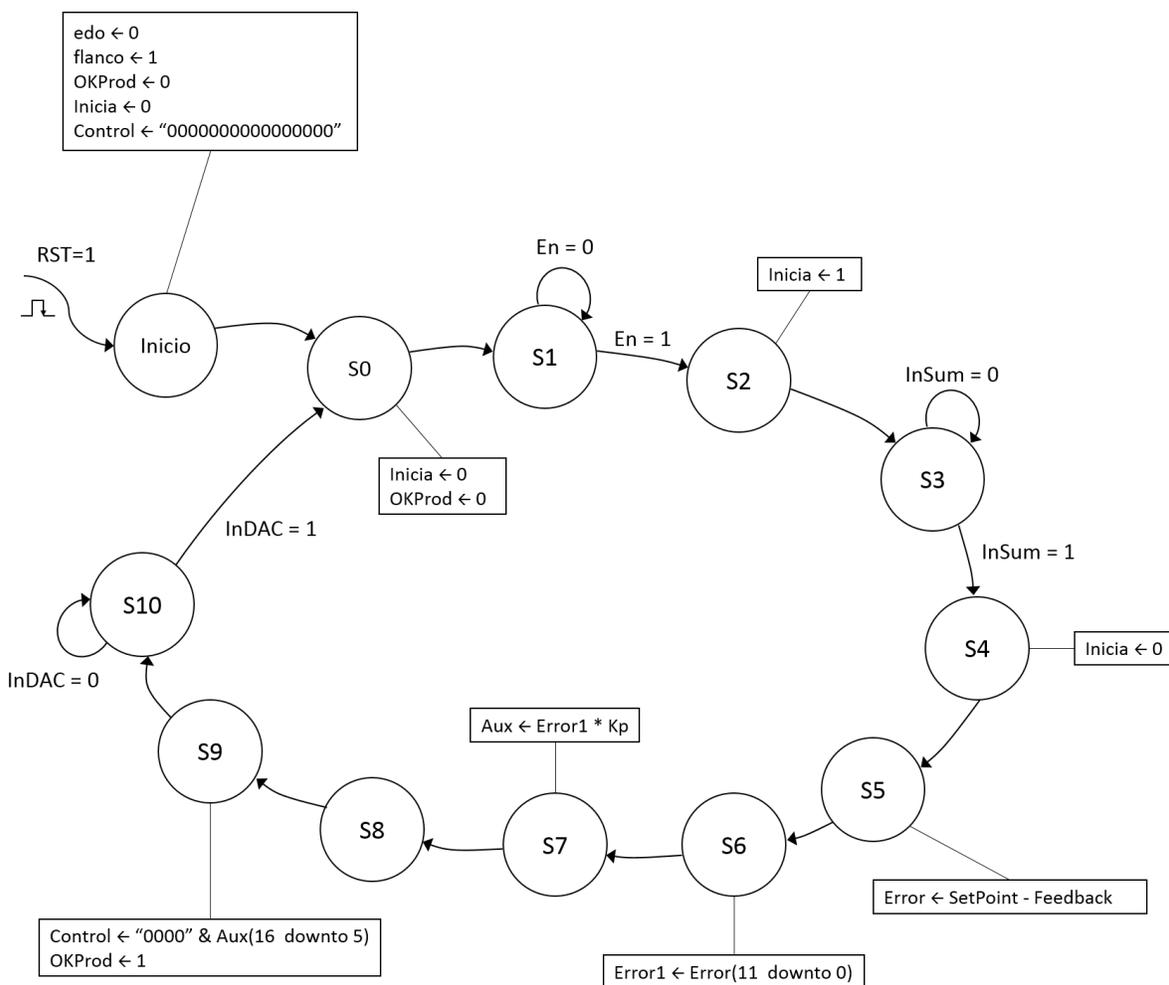


Figura B.10: FSM para el módulo de control

el estado 6, se toman los 12 bits correspondientes a la trama de dato válido obtenidos del ADC.

A pesar del orden establecido para la manipulación de datos analógicos y digitales, hay que añadir un módulo que permita habilitar al control maestro, es decir, un módulo que permita el control para la adquisición y procesamiento de nuevos valores. Para ello, el submódulo EnModule basado en una máquina de estados, contabiliza el número de pulsos ejecutados antes de permitir la siguiente conversión, es decir, se define el intervalo en el cual se adquieren y convierten datos.

Tomando en cuenta los módulos implementados y el análisis de los ciclos de reloj empleados para su ejecución, se tiene lo siguiente:

- ✓ **ADCModule** y **DACModule**. Presentan **10 estados** cada uno, de los cuales del 1 al 4 se repiten **16 veces** debido a la señal de reloj generada, esto origina un total de **70 ciclos de reloj** para la ejecución de cada uno.

- ✓ **ProporcionalC.** Este módulo contiene 11 estados, de los cuales, el 3 y 10 esperan los ciclos de reloj correspondiente a la ejecución de los módulos ADC y DAC. Por lo que se tiene un total de **149 ciclos de reloj** en su ejecución.

Con base en esta consideración, se implementa la señal *Enable* con 161 ciclos de reloj, el primero de ellos en alto y los demás en bajo. Esto con la finalidad de controlar el ancho de banda del sistema y proporcionar determinado control al momento de efectuar la adquisición y procesamiento de valores. La FSM diseñada para esta acción se muestra en la Figura B.11.

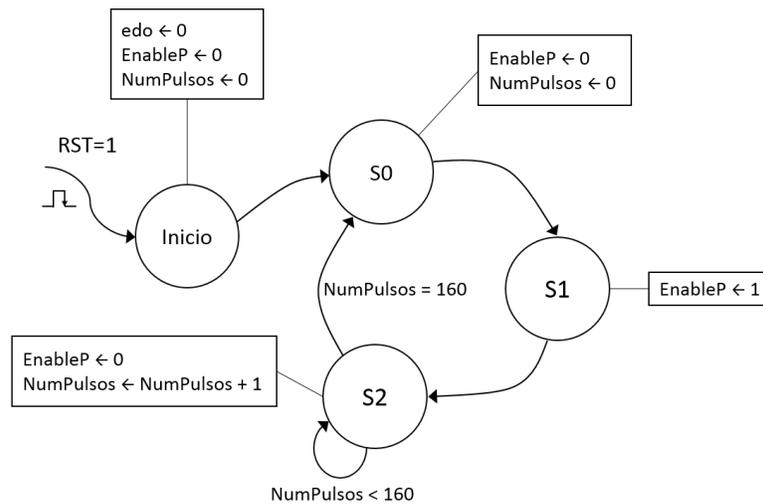


Figura B.11: FSM para el módulo del *Enable*

La simulación de la señal *Enable* (Figura B.12) muestra el ancho de banda calculado en base al número de ciclos de reloj por la duración del estado, $(161) * (20ns) = 3,22\mu s$.

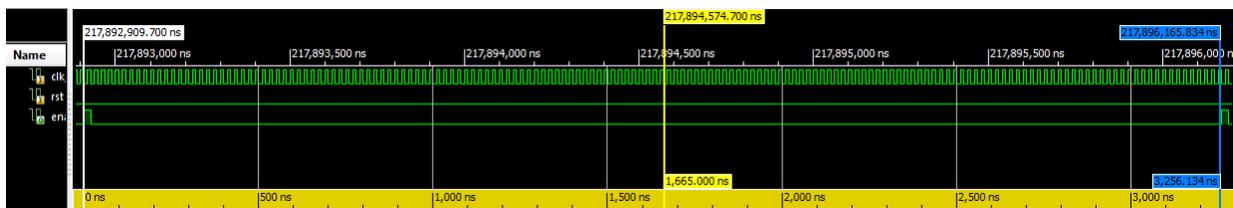


Figura B.12: Simulación para la señal *Enable*

B.5. Diseño final

En la Figura B.13, se muestra la conexión propuesta para los bloques creados a partir de los módulos antes descritos y la configuración aplicada al bloque DCM.

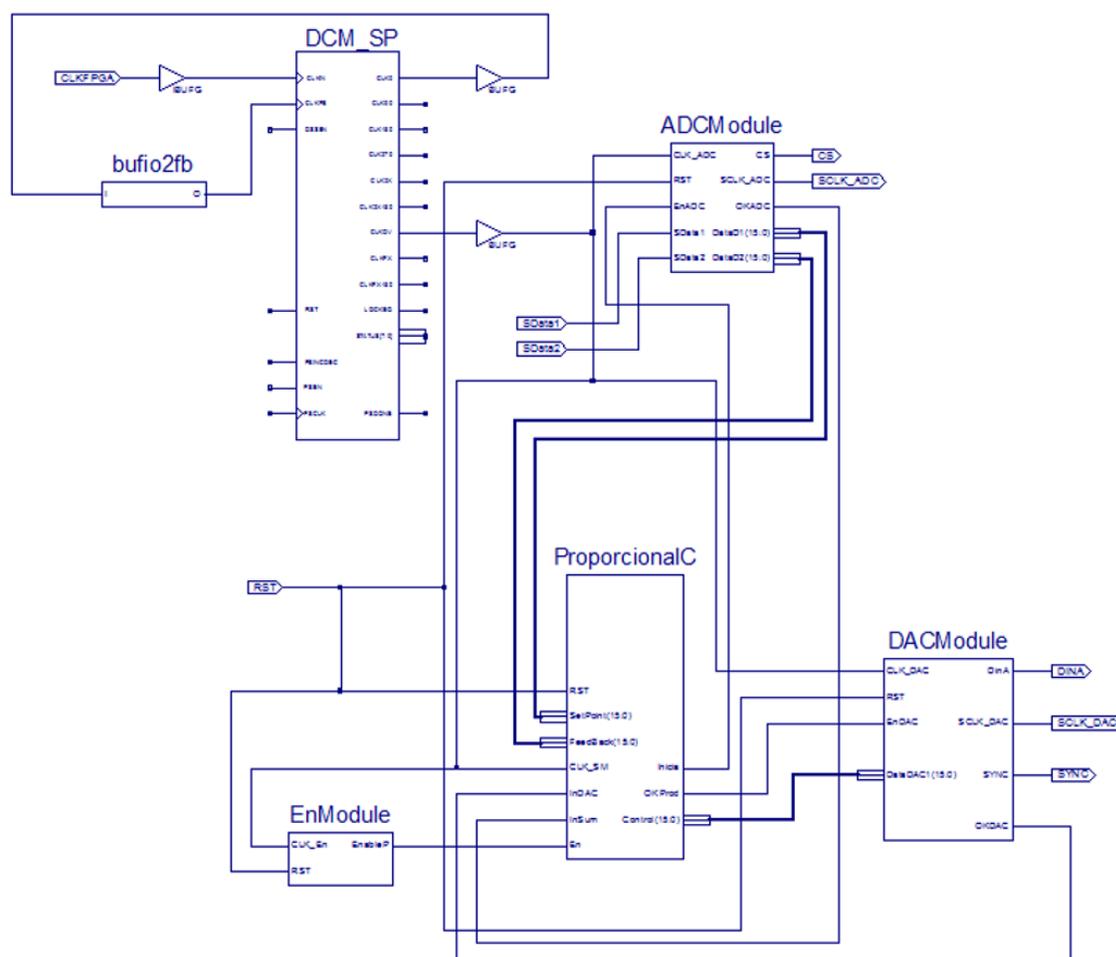


Figura B.13: Diagrama de bloques para el diseño de un controlador PID digital

B.6. *Resources*: Una comparativa entre FPAA y FPGA

Considerando el caso del controlador proporcional, en esta sección se muestra la comparación entre la cantidad de recursos utilizados a manera de síntesis de un FPGA y la simulación con un FPAA.

Tras realizar la síntesis e implementación del diseño en el ISE de Xilinx, en la Tabla B.2 se muestran los recursos principales y el porcentaje de su aplicación para la implementación del diseño propuesto en la Figura B.13. Para este caso, la cantidad de recursos empleados es mínima, permitiendo que el FPGA tenga disponible al menos el 98 % de estos para otras aplicaciones.

Con base en la información descrita en la Sección 4.1.4.6 en relación a los recursos empleados por el controlador P (Figura B.14), se tiene que su implementación en el FPAA, muestra el uso parcial de 2 CABs, en el primero de ellos se utilizan 4/8 capacitores y los 2 OpAmp,

Tabla B.2: Reporte de síntesis en ISE 13.2 de Xilinx

Recurso empleado	Utilizados	Disponibles	% utilizado
Registros Slice	18	18224	1 %
Usados como Flip-Flop	18		
Usados como Latches	0		
Slice LUTs	16	9112	1 %
Slices ocupados	7	2278	1 %
IOBs	9	232	3 %

mientras que en el segundo, se emplean 3/8 capacitores y 1 OpAmp. Lo que conlleva a mencionar que de los recursos disponibles en el FPAA, se utiliza el 21,875 % de los capacitores y el 37,5 % de los OpAmps, dejando libres los recursos del SAR y comparadores.

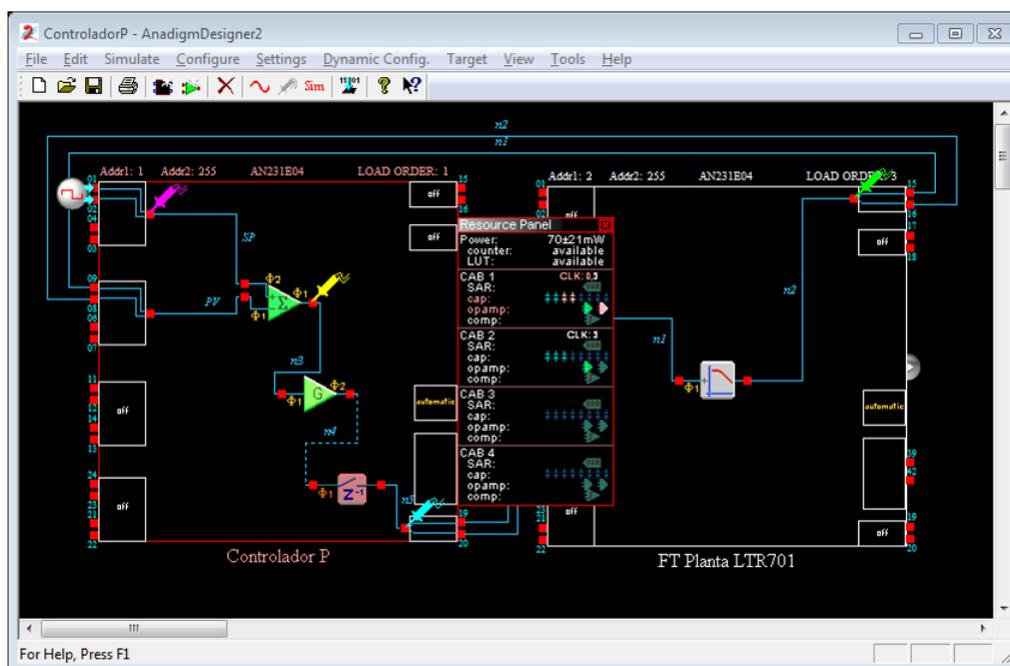


Figura B.14: Distribución de los CAMs utilizados en el FPAA

A simple vista, la implementación de un circuito digital requiere de un tiempo mayor para llevar a cabo el diseño que su contraparte analógica. En este caso, las herramientas disponibles se manipulan desde diferentes entornos, para el FPGA se emplea la programación concurrente, lo cual implica definir las instrucciones por medio de lenguaje HDL, mientras que para el FPAA se utiliza la programación por medio de objetos, los cuales basta con arrastrar, cablear y configurar sus parámetros para implementar determinado circuito. Esto proporciona un ambiente fácil, rápido y accesible para los usuarios, queda a criterio del programador la elección de uno de estos o la aplicación en base a otra herramienta.

Efectuar una comparativa entre la cantidad de recursos utilizados entre un FPAA y un FPGA resulta inadecuado, debido a que cada uno cumple con funciones distintas, es decir, cuentan con arquitecturas diferentes, se podría citar que el FPGA tiene una arquitectura más compleja que el FPAA, sin embargo, si se considera una reducción en cuanto a su implementación y a la arquitectura de estos chips, se tiene que con un FPAA se aprovecha gran parte de los recursos disponibles en sus diseños, mientras que un FPGA emplea una parte mínima de los que provee.

Bibliografía

- [1] ANADIGM. AN231001-U301 Multiples FPAA. Anadigmvortex technical training, Anadigm, Inc.
- [2] ANADIGM. Procedure for Programming the AnadigmVortex Development Board PIC. Design brief, Anadigm, Inc.
- [3] ANADIGM. Understanding the Anadigm[®] Boot Kernel (ABK). Design brief 205, Anadigm, Inc.
- [4] ANADIGM. AN10E40. Field Programmable Analog Array. Datasheet, Anadigm, Inc., 2001.
- [5] ANADIGM. Anadigm FPAA Solutions for PID Control. Customer presentation, Anadigm, Inc., August 2002.
- [6] ANADIGM. Automatic Gain Control. Application note 206, Anadigm, Inc., 2002.
- [7] ANADIGM. Conditioning PBX Signals with FPAA. Application note 203, Anadigm, Inc., 2002.
- [8] ANADIGM. Design Considerations in Mixed Signal System Boards. Application note 202, Anadigm, Inc., 2002.
- [9] ANADIGM. Interfacing Analog Signals to the Anadigmvortex FPAA Devices. Application note 205, Anadigm, Inc., 2002.
- [10] ANADIGM. *Switched Capacitor: Sampled Data Systems*. Anadigm, Inc., 2002.
- [11] ANADIGM. Understanding Anadigm[®] Configurable Analog Modules (CAMs). Anadigmvortex technical training, Anadigm, Inc., November 2002.
- [12] ANADIGM. Understanding the Anadigm[®] Multiplier CAM. Design brief 208, Anadigm, Inc., 2002.
- [13] ANADIGM. *Using the Anadigm FPAA to Interface With Sensors*, 2002.
- [14] ANADIGM. *AnadigmDesigner 2 User Manual*. Anadigm, Inc., 2004.
- [15] ANADIGM. Audio Effects - Phase Shifter. Application note, Anadigm, Inc., April 2004.

-
- [16] ANADIGM. Embedded Subwoofer Conditioner. Design brief, Anadigm, Inc., January 2004.
 - [17] ANADIGM. *AN13x series, AN23x series, AnadigmApex dpASP Family User Manual*, 2006.
 - [18] ANADIGM. Auto-nulling of OpAmps within the AN231E04 dpASP. Application note 302, Anadigm, Inc., 2007.
 - [19] ANADIGM. *AnadigmApex Development Board (version 3) - Quick Start Guide*, 2008.
 - [20] ANADIGM. Creating a DLL from AnadigmDesigner2 C-code. Application note 208, Anadigm, Inc., July 2009.
 - [21] ANADIGM. State-Driven Control of a dpASP using a Microchip PIC. Application note 207, Anadigm, Inc., July 2009.
 - [22] ANADIGM. Understanding the Anadigm[®] TransferFunction CAM. Design brief 210, Anadigm, Inc., 2009.
 - [23] ANADIGM. Using the On-Chip Temperature Sensor on the AN231E04. Application note, Anadigm, Inc., August 2009.
 - [24] ANADIGM. *ANADIGM. The dpASP Company - delivering dynamically programmable Analog Signal Processing*. Anadigm, Inc., Diciembre 2011.
 - [25] ANADIGM. Anadigmvortex Thermocouple Solution. Application note, Anadigm, Inc., October 2011.
 - [26] ANADIGM. AN221E04 Datasheet - Dynamically Reconfigurable FPAA with Enhanced I/O. Datasheet, Anadigm, Inc., 2012.
 - [27] ANADIGM. *AN231K04-DVLP3 - AnadigmApex Development Board*, 2013.
 - [28] ANADIGM. Apex Filter Demo. Application note, Anadigm, Inc., September 2013.
 - [29] ANADIGM. Boxcar Integration, an example using AN231E04 in a Pulse Induction Metal Detector. Application note 308, Anadigm, Inc., October 2013.
 - [30] ANADIGM. Capacitance Measurement Using AN231E04 dpASP. Application note 306, Anadigm, Inc., January 2013.
 - [31] ANADIGM. Interfacing Analog Signals to the AnadigmApex (3.3v) Family of dpASP's. Application note 303, Anadigm, Inc., 2013.
 - [32] ANADIGM. Low Power & Standby Modes In Apex. Application note, Anadigm, Inc., September 2013.
 - [33] ANADIGM. AN231E04 Datasheet Rev 1.2 - 3rd Generation Dynamically Reconfigurable dpASP. Datasheet, Anadigm, Inc., 2014.

- [34] ANADIGM. *CAM Documentation*, Octubre 2014.
- [35] ANADIGM. Addressing Multiple FPAA's Using a SPI Interface. Application note 310, Anadigm, Inc., January 2015.
- [36] ANADIGM. How to generate a swept frequency signal generator for AnadigmDesigner2. Application note 316, Anadigm, Inc., February 2015.
- [37] ANADIGM. PIC32 Embedded Audio Sinewave Generator. Application note 315, Anadigm, Inc., January 2015.
- [38] ASTROM, K., AND TORE, H. *PID controllers: theory, design, and tuning*. Instrument Society of America, Research Triangle Park, NC, 1995.
- [39] CAICEDO, R., AND VELASCO, J. Diseño de Circuitos Análogos Usando FPAA's. In *IX Workshop Iberchip* (2003).
- [40] COMPUTING, M. *USB-1208FS. USB-based Analog and Digital I/O Module User's Guide*. Measurement Computing Corporation, 2007.
- [41] DE LEÓN, V. Sistema de Adquisición de Datos Basado en FPAA. Tesis de licenciatura, Universidad Politécnica de Madrid, España, Septiembre 2012.
- [42] DELGADILLO, A., AND PIRAJÁN, R. Uso de arreglos analógicos programables (FPAA) en instrumentación industrial. *Tecnura* 12, 24 (2009), 43–54.
- [43] DEVICES, A. AD7476A/AD7477A/AD7478A. Datasheet, Analog Devices, Inc., 2002.
- [44] DIGILENT. *Digilent PmodDA2TM Digital To Analog Module Converter Board*, September 2006.
- [45] DIGILENT. *Digilent Pmod Interface Specification*. Digilent, Inc., 2011.
- [46] DIGILENT. *Digilent PmodAD1TM Analog To Digital Module Converter Board*, December 2011.
- [47] DIGILENT. *Nexys3 Board Reference Manual*. Digilent, Inc., 2013.
- [48] ELWE. *Operating Manual. Controlled Air Mass and Temperature System with Actuators and Sensors LTR701*. AMIRA, 2002.
- [49] HERNÁNDEZ, J., AND LUNA, N. Laboratorio de Electrónica Analógica usando Arreglos Analógicos Programables por Campo (FPAA). Tesis de licenciatura, Universidad Pontificia Bolivariana, Bucaramanga, Enero 2011.
- [50] ILA, V., BATLLE, J., AND CUFI, X. Recent trends in FPAA devices. In *International Workshop on Systems, Signals and Image Processing* (2002), pp. 180–190.

- [51] LITA, I., VISAN, D., AND CIOC, B. FPAA based PID controller with Applications in Nuclear Domain. In *Electronics Technology (ISSE). 32nd International Spring Seminar, IEEE* (2009), pp. 1–4.
- [52] NILOGIC PVT. LTD. *ni2 designs. Analogic The Programmable Analog Solution. User manual Version 2.0.*
- [53] OGATA, K. *Ingeniería de Control Moderna.* Prentice Hall. PEARSON, 2010.
- [54] PALUSINSKI, O., ANDERSON, D., GETTMAN, D., MARCJAN, C., AND ANDERSON, H. Motorola Field Programmable Analog Arrays in Simulation, Control, and Circuit Design Laboratories. *SIMULATION SERIES 29* (1997), 41–44.
- [55] PIERZCHALA, E., PERKOWSKI, M., AND GRYGIEL, S. A Field Programmable Analog Array for continuous, fuzzy and valued logic applications. In *Multiple-Valued Logic. IEEE Proceedings, Twenty-Fourth International Symposium* (1994), pp. 148–155.
- [56] SCHITTER, G., AND PHAN, N. Field Programmable Analog Array (FPAA) based Control of an Atomic Force Microscope. In *American Control Conference. Seattle, Washington, IEEE* (2008), pp. 2690–2695.
- [57] SELOW, R., LOPES, H., AND LIMA, C. A comparison of FPGA and FPAA technologies for a signal processing application. In *Field Programmable Logic and Applications, FPL 2009 International Conference, IEEE* (2009), pp. 230–235.
- [58] SEMICONDUCTOR, N. DAC121S101 12-Bit Micro Power Digital-to-Analog Converter with Rail-to-Rail Output. Datasheet, National Semiconductor Corporation, June 2005.
- [59] TAVAKOLI, S., AND TAVAKOLI, M. Optimal tuning of PID controllers for first order plus time delay models using dimensional analysis. In *The Fourth International Conference on Control and Automation (ICCA 03)* (2003), pp. 942–946.
- [60] VECA, A., AND IMHOF, A. Sensores Programables para Tomografía Sísmica de Suelos Basados en FPAA. In *5to congreso iberoamericano de sensores, IBERSENSOR* (2006).
- [61] VISAN, D., LITA, I., AND CIOC, B. Temperature control system based on adaptive PID algorithm implemented in FPAA. In *Electronics Technology (ISSE). 34nd International Spring Seminar, IEEE* (2011), pp. 501–504.