



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Diseño e implementación de un control automático de ganancia para un enlace de comunicaciones por luz visible

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e Imagen

AUTOR/A: Valls Sansaloni, Trinidad

Tutor/a: Pérez Pascual, M^a Asunción

CURSO ACADÉMICO: 2023/2024

Resumen

Las Comunicaciones por Luz Visible (VLC) han despertado un creciente interés en los últimos años como una tecnología de comunicación inalámbrica eficiente en términos energéticos para entornos interiores. Esto se debe al uso cada vez mayor de LEDs para iluminación y su capacidad para transmitir datos simultáneamente, mediante la modulación de intensidad. En este Trabajo Fin de Grado se ha implementado un Control Automático de Ganancia (AGC) para su uso en entornos VLC. El AGC se implementa de forma mixta con un amplificador controlado por voltaje y un dispositivo FPGA. Para diseñarlo se ha modelado usando Matlab/Simulink tanto la etapa analógica como la algorítmia digital. Para la implementación de la parte digital se ha usado System Verilog HDL y un dispositivo FPGA Cyclone IV de Intel-Altera. El sistema ha sido validado en un enlace VLC montado en el laboratorio.

Palabras clave: Control automático de ganancia; Amplificador controlado por voltaje; FPGA; VLC.

Resum

Les Comunicacions per Llum Visible (VLC) han despertat un creixent interès en els últims anys com una tecnologia de comunicació inalámbrica eficient en termes energètics per a entorns interiors. Això es deu a l'ús cada vegada major de LEDs per a il·luminació i la seua capacitat per a transmetre dades simultàniament, mitjançant la modulació d'intensitat. En aquest Treball de Fi de Grau s'ha implementat un Control Automàtic de Ganàncies (AGC) per al seu ús en entorns VLC. L'AGC es duu a terme de forma mixta amb un amplificador controlat per voltatge i un dispositiu FPGA. Per a dissenyar-lo s'ha modelat usant Matlab/Simulink tant l'etapa analògica com l'algorítmia digital. Per a la implementació de la part digital s'ha usat System Verilog HDL i un dispositiu FPGA Cyclone IV d'Intel-Altera. El sistema ha sigut validat en un enllaç VLC muntat al laboratori.

Paraules clau: Control automàtic de ganància; Amplificador controlat per tensió; FPGA; VLC.

Abstract

Visible Light Communications (VLC) have sparked growing interest in recent years as an energy-efficient wireless communication technology for indoor environments. This interest is driven by the increasing use of LEDs for lighting and their ability to simultaneously transmit data through intensity modulation. In this Final Degree Project, an Automatic Gain Control (AGC) has been implemented for use in VLC environments. The AGC is realized using a mixed approach with a voltage-controlled amplifier and an FPGA device. For the design, both the analog stage and the digital algorithm were modeled using Matlab/Simulink. The digital part was implemented using System Verilog HDL and an Intel-Altera Cyclone IV FPGA device. The system was validated in a VLC link set up in the laboratory.

Keywords: Automatic Gain Control; Voltage-controlled amplifier; FPGA; VLC.

A Asun, que ha sigut una excel·lent tutora.
A la meua estimada familia, que ha estat en aquesta aventura i en moltes més.
A Pau, amb qui he compartit les hores al lab i m'ha ajudat a seguir endavant.
I també als meus amics de la carrera, perquè sense ells, no hauria sigut el mateix.

Índice general

1. Introducción	1
1.1. Descripción del sistema y objetivos	2
1.2. Metodología empleada	3
1.3. Estructura de la memoria	4
2. Amplificador controlado por tensión	5
2.1. Características del AD8337	5
2.2. Caracterización del dispositivo	7
2.3. Modelo Simulink del ACV	9
3. DAC basado en PWM	11
3.1. Esquema general del DAC	11
3.2. Modelo del DAC	12
3.2.1. Modelo del Filtro Paso Bajo	12
3.2.2. Modelo del PWM	13
3.2.3. Configuración del DAC basado en PWM	15
4. Diseño del control automático de ganancia	19
4.1. Algoritmo de control automático de ganancia básico	19
4.2. Modelo Simulink del AGC con ACV y DAC basado en PWM	21
5. Implementación en FPGA del control automático de ganancia	25
5.1. Implementación del algoritmo AGC	25
5.1.1. Módulo saturador: b0_sat	26
5.1.2. Módulo valor absoluto: b1_abs	27
5.1.3. Filtro suavizador: b2_cic	27
5.1.4. Módulo logaritmo: b3_log	28
5.1.5. Módulo restador: b4_ref	28
5.1.6. Módulo multiplicador: b5_mult_kp	29
5.1.7. Módulo acumulador: b6_acc	29
5.1.8. Módulo de adaptación al PWM: b7_pre_PWM	29
5.1.9. Verificación del módulo AGC	30
5.2. Implementación del PWM	33
5.2.1. Verificación del módulo PWM	33
5.3. Implementación del bloque de comunicación con el PC	34
5.3.1. Módulo BCOM_CONF_REGS	35
5.3.2. Módulo BCOM_CONF_FSMS	36

5.3.3. Verificación del módulo de comunicación con el PC	38
5.4. Integración del sistema completo	38
6. Validación del sistema	41
6.1. Pruebas con la instrumentación del laboratorio	41
6.2. Pruebas con el banco de pruebas de VLC	43
7. Conclusiones y trabajos futuros	47
7.1. Conclusiones	47
7.2. Trabajos futuros	47
7.3. Relación con ODS	48
Bibliografía	49

Índice de figuras

1.1. Banco de pruebas del enlace de comunicaciones con VLC	2
1.2. Diagrama de bloques del sistema completo	3
2.1. Esquema del chip AD8337 de Analog Devices [3]	5
2.2. Ganancia del chip AD8337 con $V_{COM} = 0\text{ V}$ [3]	6
2.3. Respuesta en frecuencia del chip AD8337 [3]	6
2.4. Chip AGC	7
2.5. Circuito de la entrada de V_{com}	7
2.6. Ganancia real del chip	8
2.7. Respuesta en frecuencia real del chip	9
2.8. Modelo Simulink del ACV	9
2.9. Modelo real del AGC con $V_{COM} = 2,5\text{ V}$	10
3.1. Esquema general del convertor digital analógico	11
3.2. Circuito del filtro paso bajo del DAC	12
3.3. Respuesta en frecuencia del filtro paso bajo	13
3.4. Esquema general del PWM	13
3.5. Funcionamiento del PWM con PASO=3 e IN=30	14
3.6. Funcionamiento del PWM con PASO=3 y IN=190	15
3.7. Funcionamiento del PWM con PASO=80 e IN=128	16
3.8. Relación entre las frecuencias necesarias en el DAC	16
3.9. Funcionamiento del DAC basado en PWM	17
4.1. Modelo básico del AGC de Wireless Pi	19
4.2. Resultados del modelo AGC básico con una $kp = 2^{-8}$	20
4.3. Resultados del modelo AGC básico con una $kp = 2^{-3}$	21
4.4. Modelo completo del AGC	21
4.5. Resultados del modelo AGC completo con una $kp = 2^{-14}$	23
4.6. Resultados del modelo AGC completo con una $kp = 2^{-7}$	24
5.1. Diagrama de bloques de implementación del algoritmo AGC	26
5.2. Esquema del saturador	26
5.3. Diagrama del multiplexor para el valor absoluto	27
5.4. Circuito del filtro CIC	28
5.5. Esquema del filtro CIC	28
5.6. Esquema del multiplicador en coma flotante de la constante de proporcionalidad	29
5.7. Esquema del acumulador	29
5.8. Esquema del prePWM	30
5.9. Validación del bloque del AGC	30

5.10. Validación del módulo de saturación y valor absoluto del AGC	31
5.11. Validación del filtro suavizador del AGC con $M=4$	31
5.12. Validación del filtro suavizador del AGC con $M=14$	31
5.13. Validación del módulo logaritmo y señal de error del AGC	32
5.14. Validación del módulo del multiplicador y acumulador del AGC siendo $k_p=0,05$.	32
5.15. Validación del módulo del multiplicador y acumulador del AGC siendo $k_p=0,005$	32
5.16. Validación del módulo prePWM	33
5.17. Esquema general del PWM	33
5.18. Simulación de ModelSym del PWM con $IN = 190$	34
5.19. Simulación de ModelSym del PWM con $IN = 30$	34
5.20. Esquema de la comunicación entre FPGA y PC	35
5.21. Esquema de BCOM_CONF_REGS	35
5.22. Esquema de BCOM_CONF_FSMS	36
5.23. Diagrama de estados de la FSM MAIN	36
5.24. Datapath de las FSMs de lectura y escritura	37
5.25. Diagrama de estados de la FSM de escritura	37
5.26. Diagrama de estados de la FSM de lectura	38
5.27. Diagrama de bloques del sistema completo	39
6.1. Banco de pruebas con la instrumentación del laboratorio	41
6.2. Entrada (verde) y salida (amarilla) con el AGC parado	42
6.3. Entrada (verde) y salida (amarilla) con el AGC funcionando y $k_p = 2^{-9}$	43
6.4. Entrada (verde) y salida (amarilla) con el AGC funcionando y $k_p = 2^{-13}$	43
6.5. Banco de pruebas para la transmisión por luz visible	44
6.6. Diagrama de bloques del banco de pruebas	44
6.7. Prueba con transmisor y receptor cercanos y AGC desactivado	45
6.8. Prueba con transmisor y receptor lejanos y AGC desactivado	45
6.9. Prueba con transmisor y receptor cercanos y AGC activado	46
6.10. Prueba con transmisor y receptor lejanos y AGC activado	46

Índice de tablas

4.1. Tabla de la cuantificación del AGC	22
5.1. Tabla del funcionamiento del bit de control del saturador	26

Listado de siglas empleadas

ACV Amplificador Controlado por Voltaje.

ADC Analog-Digital Converter.

AGC Automatic Gain Control.

FPGA Field Programmable Gate Array.

HDL Hardware Description Language.

LED Light Emitting Diode.

PD Photo-Diode.

PWM Pulse-Width Modulation.

VLC Visible Light Communications.

Capítulo 1

Introducción

Este trabajo final de Grado (TFG) se enmarca dentro de las actividades del proyecto con título “Tecnologías ópticas sostenibles para comunicaciones de acceso con ondas milimétricas y luz visible en redes 6G (OPTIMISE)” y referencia PID2021-126514OB-I00, financiado por el Ministerio de Ciencia e Investigación. Se ha realizado en el Laboratorio de investigación del Grupo de Integración de Sistemas Electrónicos Digitales (GISED) del Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM). Una de las líneas de trabajo del proyecto OPTIMISE consiste en aprovechar para la transmisión de datos el despliegue de LEDs usados en la iluminación de los edificios. A esta línea de investigación se le denomina Comunicaciones por luz visible (VLC) [1, 2].

El control automático de ganancia (AGC) es una etapa requerida en los receptores de cualquier sistema de comunicación. En el caso de sistemas de comunicaciones digitales, el AGC se encarga de que la señal recibida mantenga la amplitud necesaria para que se digitalice aprovechando al máximo el rango de conversión de los conversores analógico-digital (ADC), lo que conlleva a una digitalización con la mayor resolución posible. En este TFG se va a diseñar e implementar un algoritmo de AGC que será utilizado en los receptores de señales transmitidas a través de los LEDs de iluminación. En la Figura 1.1 se muestra el banco de pruebas de un enlace de transmisión mediante LEDs de iluminación que está disponible en el laboratorio del grupo de investigación. La señal modulada se transmite a través del LED de iluminación (ubicado en el *truss*) y en la mesa 3 fotodiodos (PDs) captan la señal recibida que debe ser digitalizada para demodularla. El algoritmo de AGC se incorporará en este sistema entre cada PD y la etapa de digitalización, para evitar los cambios de amplitud de la señal, al cambiar la posición de los PDs.



Figura 1.1: Banco de pruebas del enlace de comunicaciones con VLC

1.1. Descripción del sistema y objetivos

En la Figura 1.2 se muestra el diagrama de bloques general del sistema AGC. Como puede verse, una parte del sistema se implementa en el dominio analógico y la otra en el digital.

En el dominio analógico se dispone de un chip amplificador controlado por voltaje (ACV), que permitirá modificar la amplitud de la señal de entrada, y un filtro paso bajo pasivo de 1^{er} orden, que posibilitará la realización de la conversión digital a analógica mediante una modulación de anchura de pulso (PWM). En la digitalización de la señal de salida del ACV se utilizará un conversor ADC, cuya salida se conectará a un dispositivo FPGA.

En el dispositivo FPGA se implementará el algoritmo de control del AGC, cuya salida modificará el voltaje a aplicar en el ACV. Para ello se implementará un conversor digital-analógico basado en PWM, implementando el PWM en el dispositivo FPGA. Tanto el algoritmo de AGC como el módulo PWM se controlará desde un PC conectado al dispositivo FPGA mediante el puerto serie RS232. Finalmente, la señal a la que se ha aplicado el control automático de ganancia se aplicará a una etapa posterior de procesado, cuya implementación no se abordará en este proyecto.

Para la realización del AGC se parte de los siguientes dispositivos:

- Tarjeta con circuitería analógica que contiene el ACV y el filtro paso bajo.
 - El ACV se ha implementado utilizando el circuito integrado AD8337 [3]. Este amplificador permite variar la amplitud de la señal de entrada en un rango de 24 dBs y el rango de amplitud a su salida es de unos 2,5 Vpp.
 - Un filtro paso bajo pasivo de 1^{er} orden implementado con una resistencia de 1 k Ω y un condensador de 10 nF, con frecuencia de corte $f_c = 15,9$ kHz.

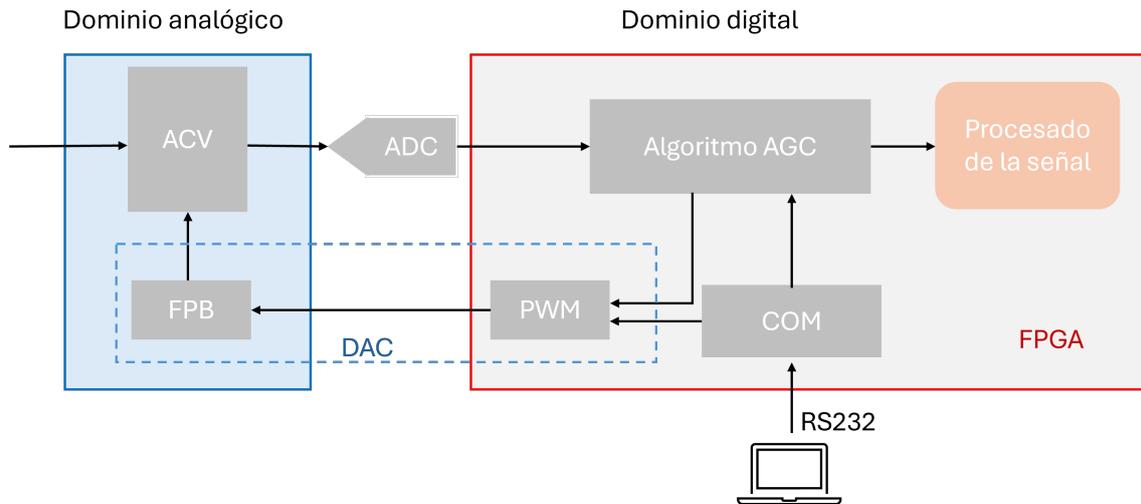


Figura 1.2: Diagrama de bloques del sistema completo

- Tarjeta DE2-115 con dispositivo FPGA Cyclone IV [4]. La frecuencia máxima de reloj que es capaz de soportar el dispositivo es $f_{clk} = 250$ MHz. La tarjeta dispone de un puerto serie para realizar la comunicación con el PC.
- Tarjeta THDB_ADA [5], que se conecta a la DE2-115, y contiene el chip AD9248, un ADC de 14 bits, rango de señal de entrada de $2 V_{pp}$ y tasa máxima de digitalización de 65 Msps.

El objetivo de este trabajo fin de grado es diseñar e implementar un control automático de ganancia (AGC) para controlar la amplitud de la señal recibida en un enlace de comunicaciones por luz visible, con un ancho de banda de 25 MHz. El control se realizará con un dispositivo FPGA, que modificará la ganancia de un amplificador controlado por voltaje. Todo el sistema deberá configurarse desde un PC, utilizando el bus de comunicaciones serie RS232.

1.2. Metodología empleada

A continuación se muestran fases en las que se ha dividido la realización de este trabajo:

1. Estudio del circuito integrado AD8337 y caracterización de la tarjeta ACV para conocer los rangos de voltaje de entrada, de amplificación y respuesta en frecuencia, y realización de un modelo funcional.
2. Estudio y modelado del algoritmo básico de AGC para entender su comportamiento.
3. Modelado del DAC basado en PWM y optimización de sus parámetros de diseño.
4. Modelado del algoritmo AGC usando el modelo de la tarjeta ACV y el modelo del PWM.
5. Modelado de precisión finita del algoritmo AGC para desarrollar un modelo de referencia para la verificación del hardware.

6. Codificación con System-Verilog HDL el algoritmo AGC y verificación mediante un banco de pruebas usando los vectores de test obtenidos del modelo de precisión finita.
7. Implementación del sistema en un dispositivo FPGA.
8. Realización de pruebas y validación del correcto funcionamiento.
9. Escritura de la memoria del proyecto.

Para la realización del proyecto se han usado las siguientes herramientas software:

- **Matlab/Simulink** para realizar los modelos enumerados anteriormente.
- **ModelSim** para depurar y verificar funcionalmente del código HDL.
- **Quartus II** para implementar el algoritmo en el dispositivo FPGA.
- **SignalTap** para la depuración del sistema implementado en el dispositivo FPGA.

Además, se han utilizado los equipos disponibles en el laboratorio del grupo de investigación (osciloscopio, generador de funciones, fuente de alimentación y el banco de comunicaciones de VLC) para las comprobaciones y demostración final del funcionamiento del sistema.

1.3. Estructura de la memoria

La memoria de este trabajo fin de grado se ha estructurado como se indica a continuación:

- **Capítulo 2:** Describe detalladamente el comportamiento del chip A8337 y sus circuitos, y posteriormente se compara con los resultados de su caracterización. Además, incluye el diseño del modelo de Simulink del ACV.
- **Capítulo 3:** Explica el convertor digital a analógico basado en la modulación de anchura de pulso y cómo seleccionar sus parámetros de diseño.
- **Capítulo 4:** Presenta el proceso del diseño del AGC en Simulink. Primero explicando el modelo del que hemos partido y luego explicando los nuevos bloques que se han añadido para conseguir mejorar el modelo y que su comportamiento se acerque al real.
- **Capítulo 5:** Presenta los bloques a implementar en la FPGA. Se han explicado detalladamente cada uno de ellos, cómo se han verificado y cómo se han integrado en el sistema final.
- **Capítulo 6:** Presenta las pruebas realizadas para validar el AGC implementado.
- **Capítulo 7:** Detalla las conclusiones finales del trabajo y se sugieren líneas de mejora.

Capítulo 2

Amplificador controlado por tensión

Para realizar el control automático de ganancia (AGC) se va utilizar el amplificador controlado por voltaje (ACV) de Analog Devices AD8337. En este capítulo se van a presentar las características principales de dicho chip y se va caracterizar su funcionamiento en la tarjeta analógica proporcionada para realizar el proyecto. Finalmente, se detallará cómo se ha realizado el modelo funcional de ACV con Matlab/Simulink, que será utilizado en el Capítulo 4 para completar el modelo del AGC.

2.1. Características del AD8337

El AD8337 es un amplificador de ganancia variable controlado por voltaje, de bajo ruido y diseñado para amplificar señales de hasta 100 MHz. En la Figura 2.1 se muestra un esquema del chip con su conexión típica alimentado con fuentes simétricas. Tal y como se muestra en la Figura, está formado por tres partes: dos operacionales no inversores y un atenuador. En primer lugar, aparece un preamplificador, es decir, un operacional no inversor que amplifica 6dB. En segundo lugar, se tiene un atenuador que atenúa entre 0 y 24 dB controlado por el pin7 (la señal de V_{GAIN}). Por último, un amplificador fijo, un segundo operacional, que amplifica 18 dB. De este modo, el chip puede amplificar señales entre 0 y 24 dB, controlando esta atenuación utilizando la señal de control V_{GAIN} .

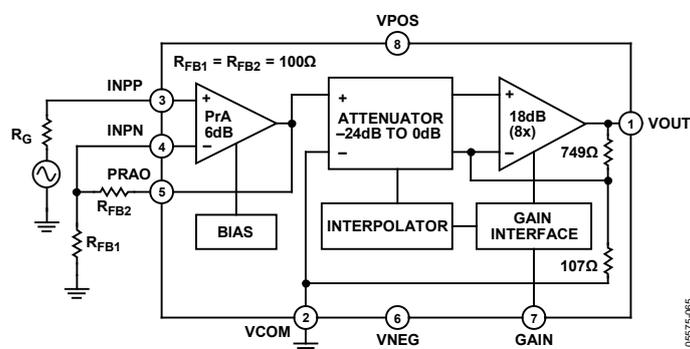


Figura 2.1: Esquema del chip AD8337 de Analog Devices [3]

A continuación, se muestran varias gráficas que describen el funcionamiento del circuito integrado AD8337. La Figura 2.2 muestra la ganancia (en dB) que introduce el dispositivo en función de la tensión de control (V_{GAIN}). Como se observa, presenta un comportamiento logarítmico para los valores de V_{GAIN} comprendidos entre -600 mV y 600 mV. Fuera de ese rango, la ganancia es constante. Hay que destacar que aunque en la gráfica citada el valor central de V_{GAIN} es 0V, este valor depende de la tensión que se fija en la entrada V_{COM} (pin 2 del chip).

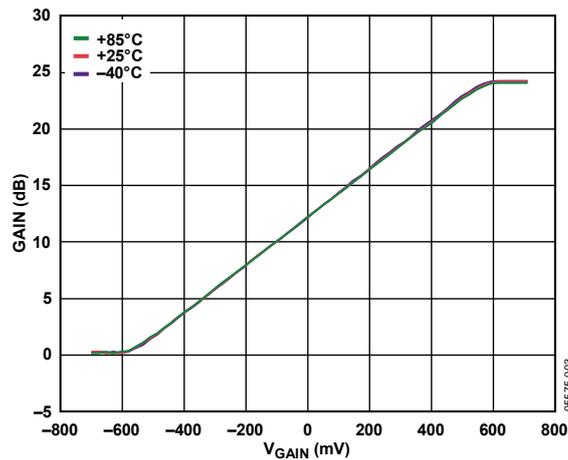


Figura 2.2: Ganancia del chip AD8337 con $V_{COM} = 0\text{ V}$ [3]

La Figura 2.3 muestra la respuesta en frecuencia (en dB) para distintos valores de la tensión de control V_{GAIN} (V). Como se observa, presenta una respuesta plana hasta unos 100 MHz. También se puede ver que para la tensión de V_{GAIN} más baja (-0,7 V) no amplifica ni atenúa, es decir, tiene ganancia de 0 dB, y en la tensión más alta (0,7 V), comprobamos que amplifica hasta 24 dB.

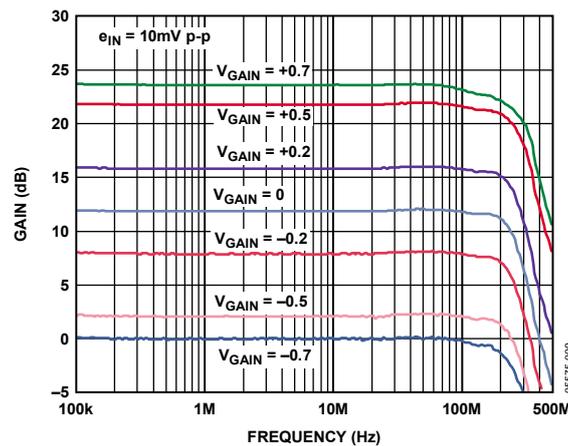


Figura 2.3: Respuesta en frecuencia del chip AD8337 [3]

2.2. Caracterización del dispositivo

En esta sección se muestran las pruebas realizadas para verificar las especificaciones del chip AD3883. A continuación, en la Figura 2.4 se muestra la tarjeta analógica que incluye el AD3883, y en la Figura 2.5 su esquema de implementación. En este caso el chip se alimenta asimétricamente entre 0 y 5 V y, por ello, se ha utilizado un divisor de tensión (Ec. 2.1), con resistencias del mismo valor ($R1 = R2 = 100\ \Omega$), para fijar el voltaje de referencia en su valor medio: $V_{COM} = 2,5\ \text{V}$.

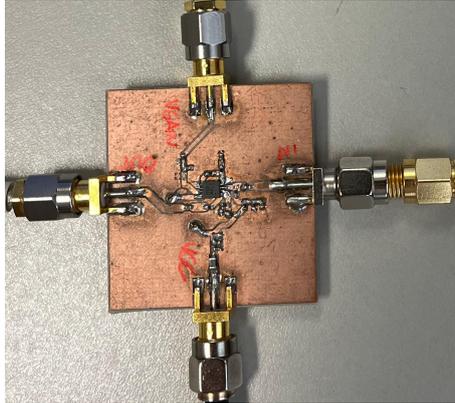


Figura 2.4: Chip AGC

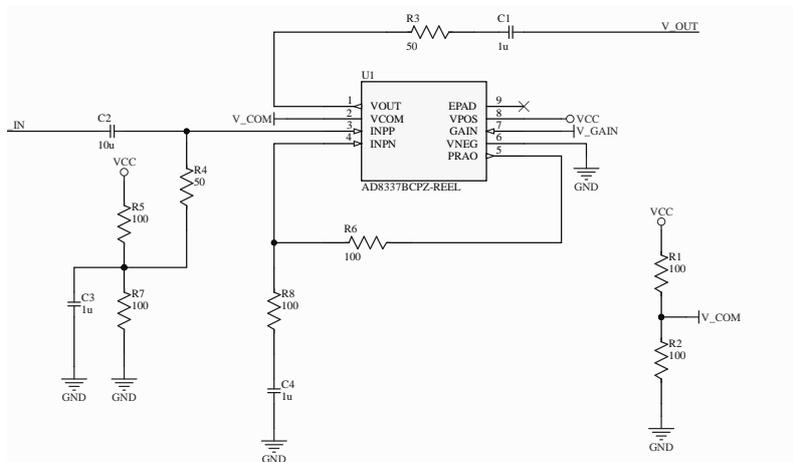


Figura 2.5: Circuito de la entrada de Vcom

$$V_{COM} = V_{IN} \cdot \frac{R2}{R1 + R2} \quad (2.1)$$

En primer lugar, para encontrar el rango real de ganancia que se puede obtener cuando $V_{COM} = 2,5\ \text{V}$ se ha medido la salida con el máximo y el mínimo V_{GAIN} posibles, manteniendo fija la amplitud de la señal de entrada. La señal de entrada se ha colocado en 160 mVpp a la frecuencia de 1 MHz y se ha medido con V_{GAIN} de 1,8 V, y 3,3 V.

Para el cálculo de la ganancia se ha utilizado la siguiente ecuación:

$$Gain = 20 \cdot \log_{10} \left(\frac{V_{OUT}}{V_{IN}} \right) \quad (2.2)$$

Con esta operación y haciendo las medidas necesarias obtenemos que la ganancia mínima es 0,37 dB y el máximo es 24,16 dB, por tanto, el rango de ganancias es de 23,78 dB, casi 24 dB, que era lo indicado en las especificaciones.

Posteriormente, se ha medido la ganancia para su rango de tensiones de control modificando V_{GAIN} entre 1,8 y 3,3 V. La gráfica se muestra en la Figura 2.6 y el resultado es coherente con la de las hojas de características del chip, mostrada en la Figura 2.2.

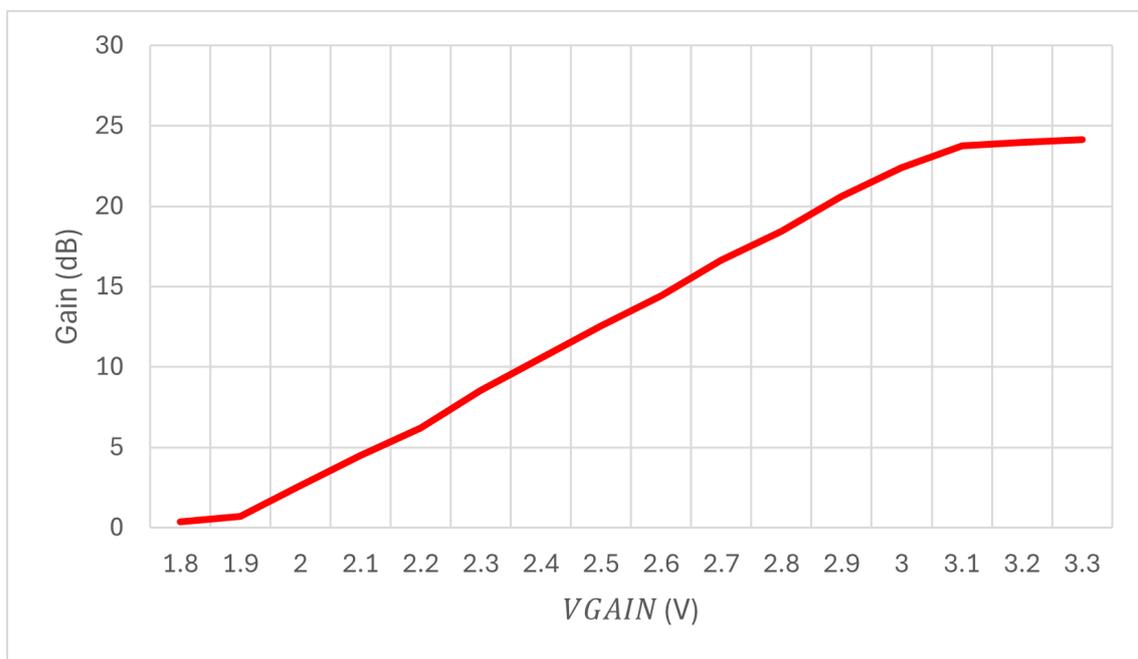


Figura 2.6: Ganancia real del chip

La siguiente prueba que se ha realizado ha sido medir la respuesta en frecuencia del chip. Se ha hecho un barrido de frecuencias entre 100 kHz y 200 MHz con diferentes tensiones de V_{GAIN} . Esto se ha representado en una gráfica en la Figura 2.7. Se puede observar que las frecuencias de corte obtenidas son menores a las especificadas en las hojas de características del chip (Figura 2.3), pero en cualquier caso suficiente para la aplicación de este trabajo, en el que se pretende controlar señales en un ancho de banda máximo de 25 MHz.

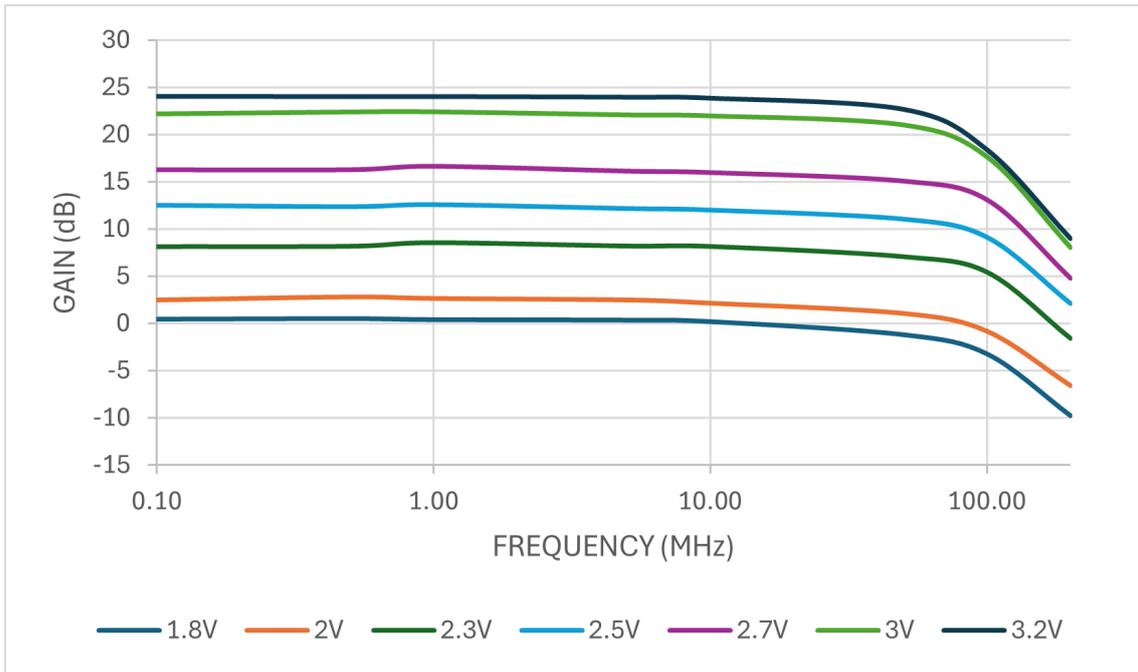


Figura 2.7: Respuesta en frecuencia real del chip

2.3. Modelo Simulink del ACV

El modelo del ACV se representa en la Figura 2.8. Lo forman un multiplicador para aplicar la ganancia, una tabla que tabula la relación exponencial de la entrada con el factor de amplificación y un saturador para limitar los valores de aplicación de la ganancia.

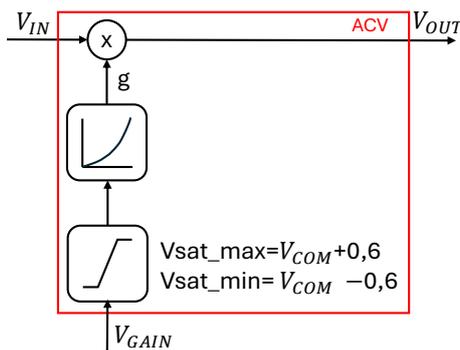


Figura 2.8: Modelo Simulink del ACV

La tabla permite calcular la ganancia a partir del valor de V_{GAIN} y V_{COM} . Los valores de la tabla se han obtenido aplicando la ecuación:

$$g = 10^{\left(\frac{(V_{GAIN} - V_{COM})}{\Delta V_{GAIN}} + \frac{1}{2}\right) \cdot \Delta G(dB)}, \quad (2.3)$$

donde $\Delta G(\text{dB}) = 24$ es el rango completo de amplificación dado en dBs y $\Delta V_{GAIN} = 1,2$ es la variación máxima de V_{GAIN} en voltios.

Tal y como se ha explicado anteriormente solo se amplifican valores alrededor de V_{COM} , exactamente entre $V_{GAIN} = V_{COM} \pm 0,6$ mV. Como tenemos la tensión centrada en 2,5 V, nuestro AGC amplificará desde 1,9 V hasta 3,1 V, y para el resto de valores de tensión satura. Para que ocurra lo mismo en nuestro modelo se ha añadido un saturador en el ACV que evita que se amplifiquen valores que están fuera de ese rango.

En la Figura 2.9 se muestra la representación de la ganancia en dB y en lineal obtenidas en la tabla del modelo al aplicar la ecuación 2.3. Se puede comprobar que corresponde al patrón de amplificación del chip ACV.

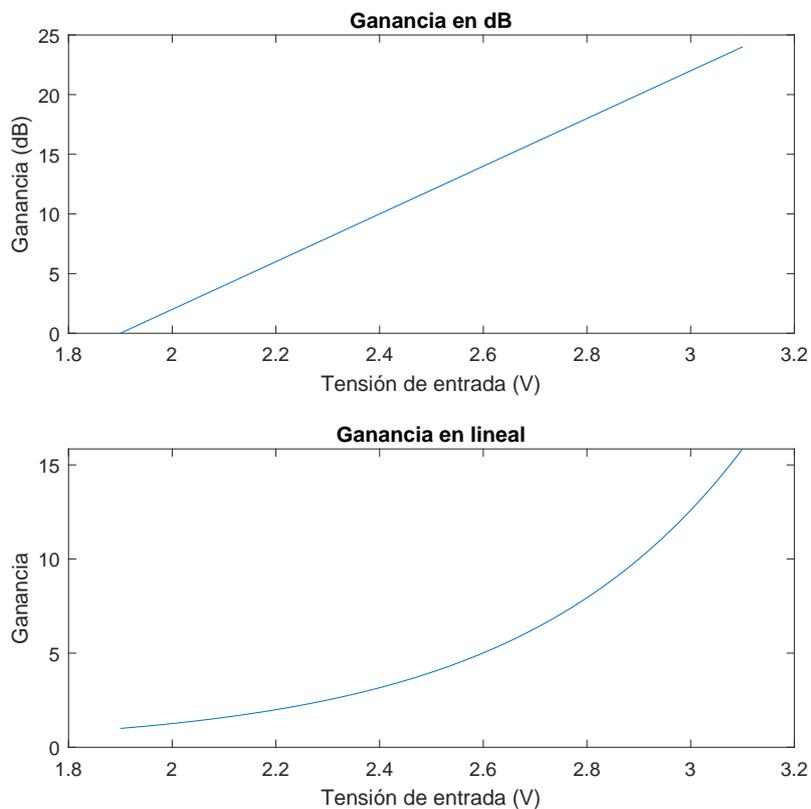


Figura 2.9: Modelo real del AGC con $V_{COM} = 2,5$ V

Finalmente, se ha comprobado que el modelo de Simulink funciona correctamente y que amplifica las señales de entrada al aplicarles los valores de voltaje V_{GAIN} dentro del rango de aplicación.

Capítulo 3

Convertidor digital analógico basado en la modulación por ancho de pulso

En este capítulo se explica el funcionamiento del convertor digital analógico basado en la modulación por ancho de pulso (PWM) y los dos bloques que lo forman. Además, se detalla cómo se ha modelado y se describe las decisiones que se han tomado para su diseño.

3.1. Esquema general del DAC

El DAC se utiliza para convertir al dominio analógico la salida digital de la FPGA y conectarla a la entrada del chip AD8337.

El convertor digital a analógico está constituido por dos bloques: un PWM y un filtro paso bajo. Véase en la Figura 3.1 cómo están conectados:

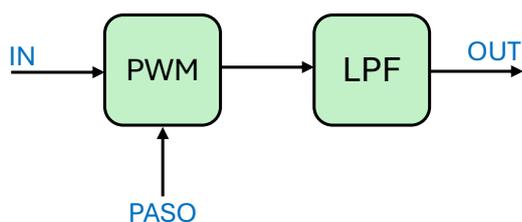


Figura 3.1: Esquema general del convertor digital analógico

El PWM está formado por un contador y un comparador y tiene dos entradas: la señal de entrada (IN), que es la señal que se quiere convertir al dominio analógico, y el PASO, que determina la frecuencia a la que funcionará el PWM. Esta frecuencia tiene que ser estrictamente mayor que la frecuencia de corte del filtro paso bajo (LPF) que le sigue (Figura 3.1), y la elección de ésta viene dada por unos criterios que se explicarán más a fondo a continuación.

El filtro paso bajo pretende eliminar los pulsos de la frecuencia de la señal cuadrada del PWM para dejar pasar simplemente la tensión continua, que corresponde al valor promedio de la señal.

3.2. Modelo del DAC

3.2.1. Modelo del Filtro Paso Bajo

El filtro paso bajo es parte del DAC y su función es dejar pasar sólo la parte continua de la señal cuadrada que se obtiene del PWM, a la entrada de V_{GAIN} del ACV (pin7 del chip AD8337). La señal cuadrada se genera para controlar el nivel de tensión en función de su ciclo de trabajo. Si la señal PWM se filtra con un filtro paso bajo con una frecuencia de corte lo suficientemente pequeña para eliminar la frecuencia del PWM, y lo suficientemente grande para dejar pasar los cambios de su valor medio producidos por el cambio de su ciclo de trabajo, se consigue obtener analógicamente la señal que está modulando el PWM.

El filtro implementado en la tarjeta analógica del ACV se muestra en la Figura 3.2. Es un filtro RC de 1^{er} orden.

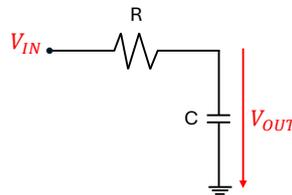


Figura 3.2: Circuito del filtro paso bajo del DAC

Al analizar el filtro se obtiene:

$$V_{OUT} = V_{IN} \cdot \frac{\frac{1}{C \cdot s}}{R + \frac{1}{C \cdot s}} = V_{IN} \cdot \frac{\frac{1}{R \cdot C}}{s + \frac{1}{R \cdot C}}. \quad (3.1)$$

De esta ecuación se obtiene su función de transferencia:

$$H(s) = \frac{V_{OUT}}{V_{IN}}(s) = \frac{w_c}{s + w_c}, \quad (3.2)$$

donde

$$w_c = \frac{1}{R \cdot C}. \quad (3.3)$$

Aunque el DAC se compone del PWM, implementado digitalmente, y el filtro, implementado analógicamente, su modelo se va a desarrollar solo en el dominio digital. Por tanto, la función de transferencia del filtro paso bajo digital de 1^{er} orden equivalente al analógico se muestra a continuación:

$$H(z) = \frac{V_{OUT}}{V_{IN}}(z) = \frac{1 - e^{-w_c T}}{1 - e^{-w_c T} \cdot z^{-1}} \quad (3.4)$$

En la tarjeta analógica del ACV los valores de los componentes eléctricos del filtro paso bajo son los siguientes: la resistencia tiene un valor de $1 \text{ k}\Omega$ y el condensador es de 10 nF . Por lo tanto, la frecuencia de corte del filtro es de $15,9 \text{ kHz}$. Esta frecuencia determina el ancho de banda

de digitalización del DAC basado en PWM. En la Figura 3.3 está representada la respuesta en frecuencia del filtro en el dominio analógico (curva azul), y su versión digital (curva marrón) muestreada con una frecuencia de muestreo de 50 MHz. Como se puede observar, la versión digital representa correctamente la analógica para frecuencias menores de 10 MHz.

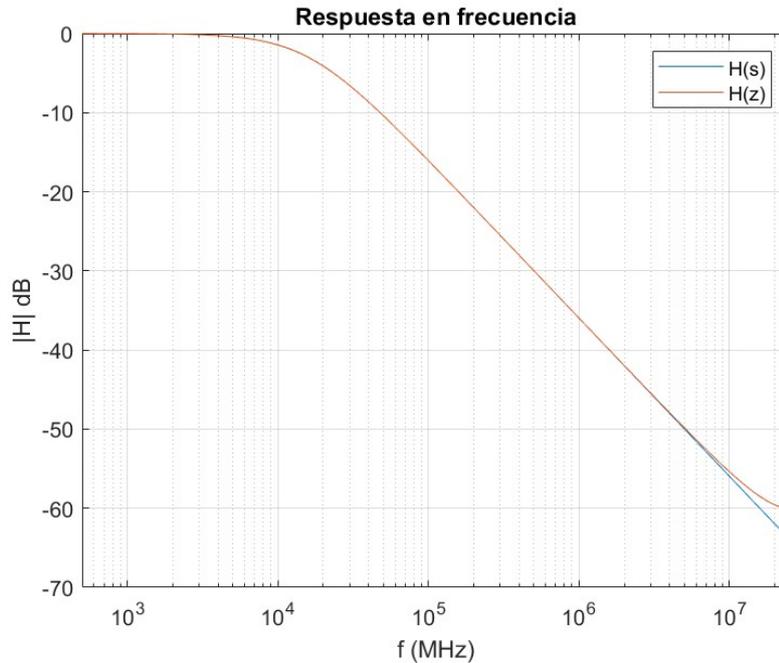


Figura 3.3: Respuesta en frecuencia del filtro paso bajo

3.2.2. Modelo del PWM

El PWM genera unos pulsos cuadrados en los que se puede modificar su ciclo de trabajo. Está formado por un contador de un número de bits determinado que incrementa su cuenta con el valor de su entrada, denominada PASO, y un comparador que compara la salida del contador con la señal que se desea modular (señal IN). Por otro lado, como la señal PWM se genera desde un pin del dispositivo FPGA, su voltaje variará entre 0 y 3,3 V. Por eso, en el modelo hay que multiplicar la salida del comparador por este valor. Su esquema se muestra en la Figura 3.4.

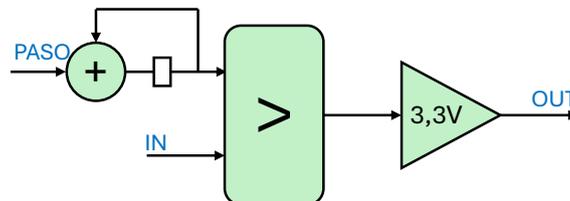


Figura 3.4: Esquema general del PWM

Se ha decidido que el PWM de nuestro DAC sea de 8 bits, por tanto, el contador sumará valores

dentro del rango de 0 a 255. Cuando se activa el PWM, el contador empieza a incrementar en cada flanco de reloj el valor introducido por la señal PASO. Este valor, junto con la frecuencia de reloj del contador, F_{clk} , determinan la frecuencia de la señal PWM. El valor de PASO para que el PWM tenga una frecuencia f_{PWM} se obtiene con la siguiente ecuación:

$$PASO = \frac{2^8 \cdot f_{PWM}}{F_{clk}} \quad (3.5)$$

La señal de entrada al PWM se compara con la señal a modular (señal IN) en el comparador. En caso de ser mayor, la salida es un 1; en caso de ser menor, es un 0. De este modo la señal de entrada define el ciclo de trabajo de la señal de salida, y, por tanto, su valor medio corresponderá al valor de la tensión continua de la señal PWM. Este valor se calcula como:

$$V_{medio} = (V_{CC}^+ - V_{CC}^-) \cdot DutyCycle, \quad (3.6)$$

donde V_{CC}^+ y V_{CC}^- son los valores de tensión máxima y mínima de la señal PWM y $DutyCycle$ es el ciclo de trabajo ($DutyCycle = \frac{IN}{2^8}$).

El funcionamiento del modelo de PWM se muestra en las Figuras 3.5 y 3.6, en las que la entrada IN vale 30 y 90, respectivamente, $PASO=3$, $V_{CC}^+ = 3,3 \text{ V}$ y $V_{CC}^- = 0 \text{ V}$.

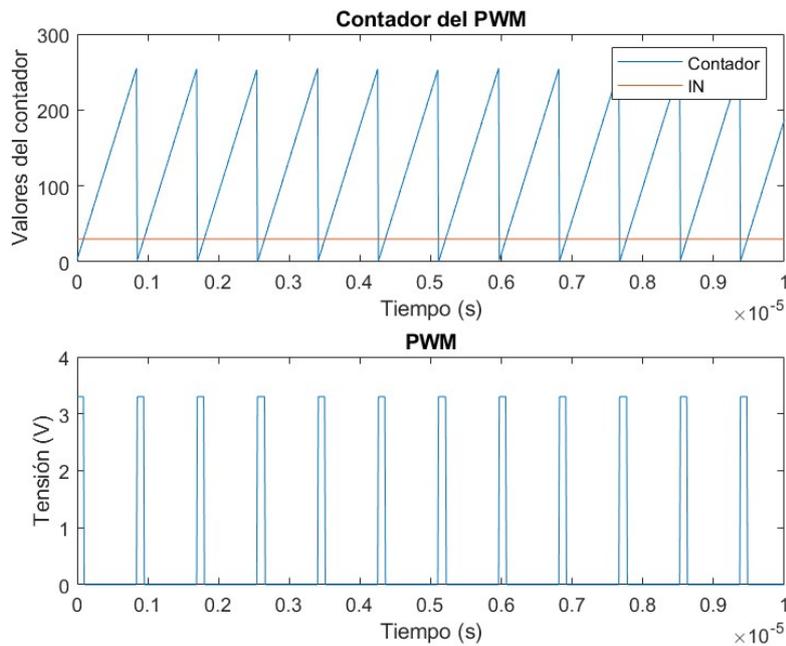


Figura 3.5: Funcionamiento del PWM con PASO=3 e IN=30

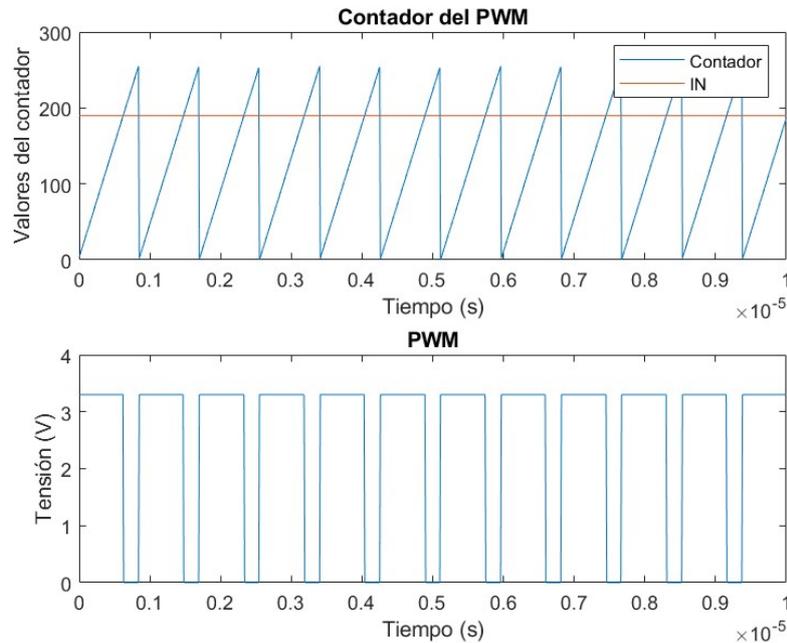


Figura 3.6: Funcionamiento del PWM con PASO=3 y IN=190

3.2.3. Configuración del DAC basado en PWM

En un DAC basado en PWM ideal el número de bits efectivos (b_{eff}) que se consigue en la conversión digital analógico depende únicamente del número de bits del contador. Un mayor número de bits conduce a una mejor representación de la señal. La relación entre el número de bits efectivos de un conversor y la relación señal-ruido (SNR) es [6]:

$$SNR = 6,02 \cdot b_{eff} + 1,76 \quad (3.7)$$

En nuestro caso el PWM es de 8 bits y, por tanto, idealmente el DAC debería conseguir una SNR=49,9 dB. Sin embargo, en el DAC basado en PWM existen otros factores que influyen en su funcionamiento:

- Si a la frecuencia del PWM el filtro paso bajo no atenúa lo suficiente, el residuo que no se llega a filtrar se suma a la señal PWM generando un error que reduce el SNR y, por tanto, el número de bits efectivos.
- Para evitar el efecto anterior se podría aumentar la frecuencia del PWM, pero esto también puede ser un problema. Al aumentar la frecuencia del PWM y acercarla a la de muestreo, el valor de PASO es muy grande, la señal PWM se representa con pocas muestras por periodo, y en ciclos consecutivos la comparación del contador con un valor constante da lugar a pulsos con distintos ciclos de trabajo. Este fenómeno se muestra en la Figura 3.7 para los valores de PASO=80 e IN=128.

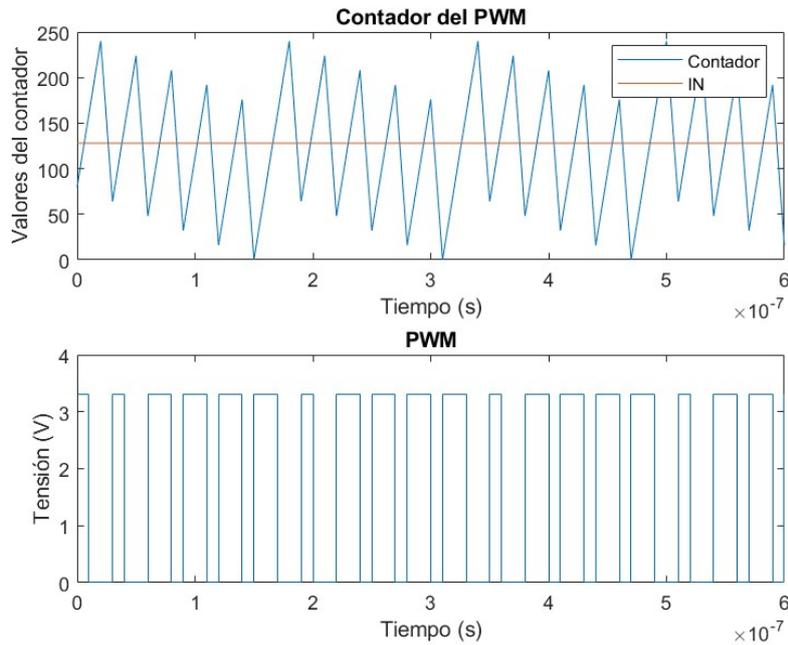


Figura 3.7: Funcionamiento del PWM con PASO=80 e IN=128

En la Figura 3.8 se ve la relación entre las diferentes frecuencias que afectan al diseño del DAC basado en PWM: la frecuencia de corte del filtro paso bajo, la frecuencia del PWM y la de muestreo. Como criterio de diseño se ha tomado elegir la frecuencia del PWM a la que el filtro paso bajo atenúe al menos 50 dB y fijar la frecuencia de muestreo a la frecuencia máxima alcanzable por el dispositivo FPGA, que en este caso es 250 MHz. El bloque PWM es un circuito sencillo que fácilmente podrá funcionar a dicha frecuencia.

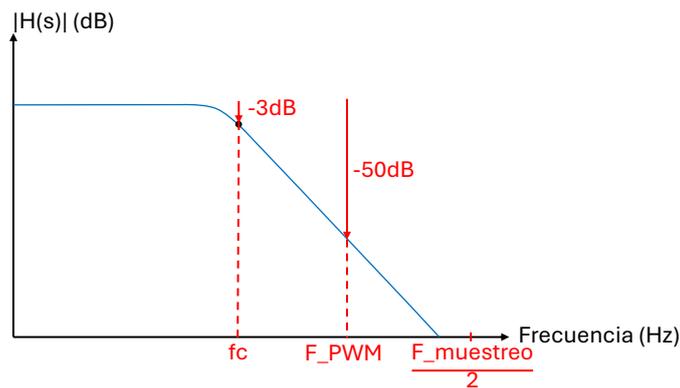


Figura 3.8: Relación entre las frecuencias necesarias en el DAC

Aplicando ese criterio de diseño y partiendo de la respuesta en frecuencia del filtro paso bajo (Figura 3.3), se obtiene que $F_{PWM} \geq 5,02 \text{ MHz}$. Para comprobarlo, se ha calculado la SNR con

el modelo de Simulink y se ha obtenido que para $F_{PWM} = 5,02 \text{ MHz}$ y $F_s = 250 \text{ MHz}$ la $SNR = 51,3 \text{ dB}$, por lo tanto, aplicando la ecuación 3.7, el número de bits efectivos del convertor es $b_{eff} = 8 \text{ bits}$, siendo este el número de bits esperado. En la Figura 3.9 se muestra la conversión al dominio analógico de una señal sinusoidal con el modelo de DAC basado en PWM en las condiciones anteriormente indicadas.

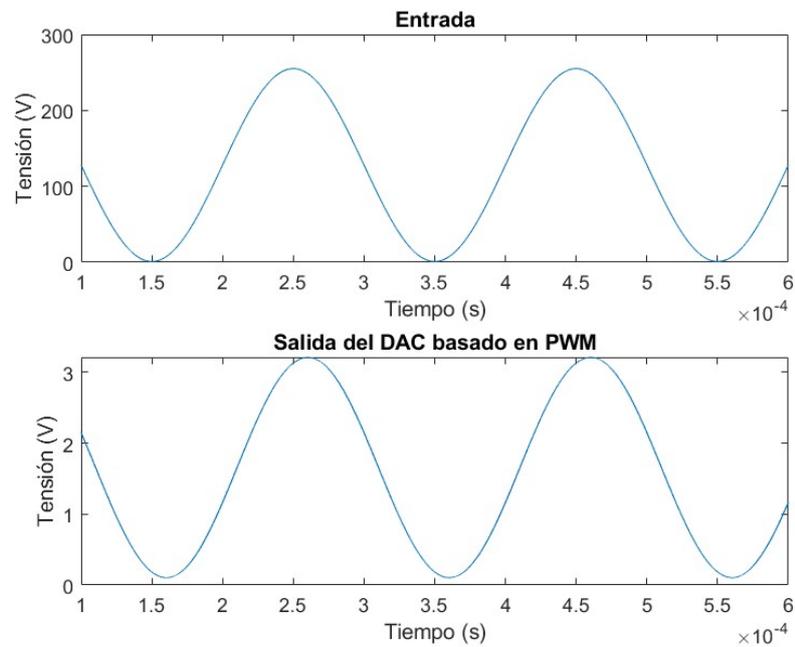


Figura 3.9: Funcionamiento del DAC basado en PWM

Capítulo 4

Diseño del control automático de ganancia

En este capítulo se va a explicar el diseño del control automático de ganancia y se presentarán los modelos del mismo realizados con Matlab/Simulink. Se ha partido del modelo de AGC de [7] y se ha ido ampliando el modelo para incluir el comportamiento del amplificador controlado por voltaje, el convertor digital-analógico basado en PWM. Posteriormente, se ha diseñado el modelo de precisión finita que tiene un comportamiento similar al del AGC real.

4.1. Algoritmo de control automático de ganancia básico

Para elaborar nuestro modelo del AGC, hemos tomado como referencia un modelo básico extraído de [7]. En la Figura 4.1 puede verse este modelo. En este modelo, $x[n]$ es la señal de entrada, $z[n]$ la salida a la que se ha aplicado el control de ganancia, R es el valor de referencia de ganancia al que debe converger el algoritmo y k_p la constante de proporcionalidad del lazo de control.

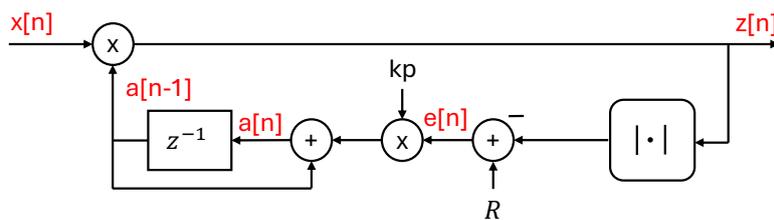


Figura 4.1: Modelo básico del AGC de Wireless Pi

La señal $z[n]$ es el resultado de la multiplicación entre la señal de entrada, $x[n]$, y un factor de amplificación, $a[n - 1]$, que se obtiene en el algoritmo:

$$z[n] = x[n] \cdot a[n - 1]. \quad (4.1)$$

Este factor de amplificación depende de la misma $z[n]$ y los valores R y k_p . Como se desea mantener constante la amplitud de la señal, se genera una señal de error, $e[n]$, restando a la referencia

de amplitud, R , el módulo de $z[n]$:

$$e[n] = R - |z[n]|, \quad (4.2)$$

es decir, lo que falta para llegar al valor de tensión que se busca. La señal de error se multiplica por la constante de proporcionalidad, kp , y se acumula para eliminar el error:

$$a[n] = kp \cdot e[n] + a[n - 1]. \quad (4.3)$$

Por tanto, los parámetros importantes para controlar el AGC son R y kp :

- R es el nivel de referencia de amplitud. Es el valor al que se pretende llegar en el AGC aunque se cambie la amplitud de la señal de entrada.
- kp : es la constante de proporcionalidad. Al cambiar kp variará el tiempo de respuesta, esto es, el tiempo que tarda la señal de salida en alcanzar el valor de amplitud de referencia. A menor kp , menor peso tiene la señal de error al acumularse y más tiempo se tardará en estabilizar la señal. En cambio, si kp es mayor, el error tendrá mayor peso y la respuesta del sistema será más rápida, pero se producirá más rizado antes de estabilizar la señal.

El algoritmo se ha modelado con Simulink. Para hacer las pruebas se ha usado una entrada sinusoidal con una frecuencia de 1 MHz con variaciones de amplitud entre 0,01 y 0,03. Se han hecho dos pruebas cambiando la constante de proporcionalidad: en primer lugar, se ha simulado con una constante de proporcionalidad de $kp = 2^{-8}$, un valor relativamente pequeño en comparación con el segundo, $kp = 2^{-3}$. En ambos casos el nivel de referencia es $R = 0,5$. Las señales resultantes se muestran en las figuras 4.2 y 4.3. Se aprecia que cuando kp es más pequeña, la señal tarda más tiempo en llegar al valor de referencia que con un valor de kp mayor. Eso se debe al cambio de la constante de proporcionalidad, por tanto, se deduce que el algoritmo funciona correctamente.

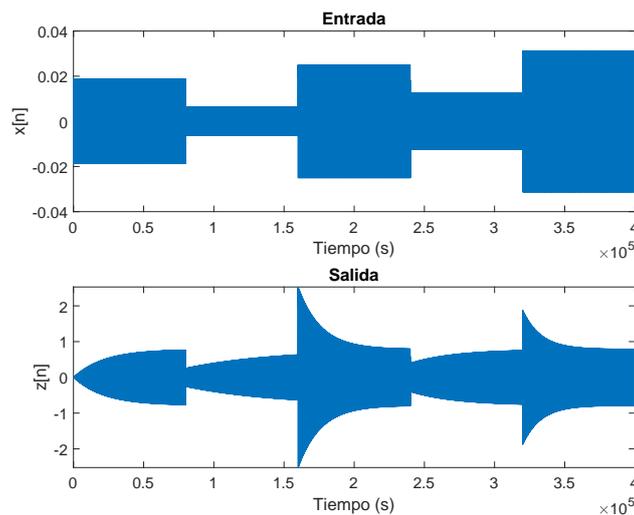


Figura 4.2: Resultados del modelo AGC básico con una $kp = 2^{-8}$

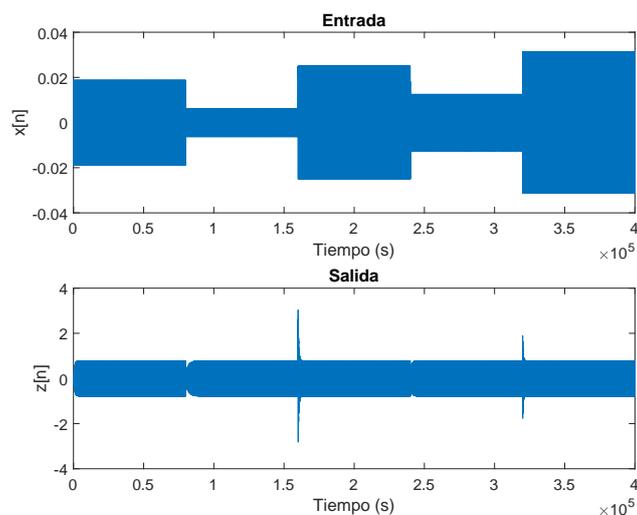


Figura 4.3: Resultados del modelo AGC básico con una $kp = 2^{-3}$

4.2. Modelo Simulink del AGC con ACV y DAC basado en PWM

En la sección anterior se ha explicado el modelo básico de un AGC. Sin embargo, el AGC en este proyecto utiliza dispositivos, como el ACV o el DAC basado en PWM, cuyo comportamiento no se ha modelado con el algoritmo AGC básico. Por ello, en esta sección se va a explicar cómo se ha modificado el algoritmo básico para diseñar un modelo más completo, que represente mejor el comportamiento real de nuestro AGC.

El modelo completo se muestra en la Figura 4.4.

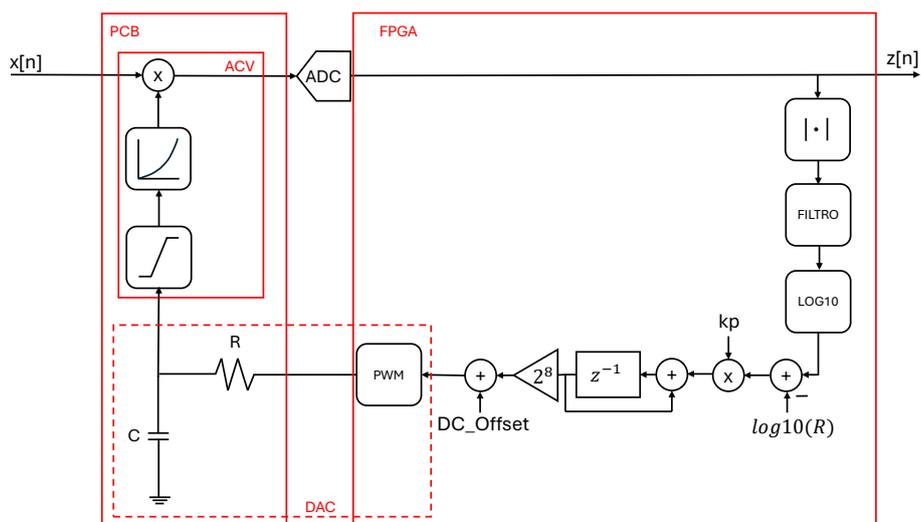


Figura 4.4: Modelo completo del AGC

A continuación, se presentan las modificaciones realizadas sobre el modelo básico.

1. En el modelo básico, la variación de ganancia se modela simplemente con un multiplicador de la señal de entrada y el factor de amplificación tal y como sale del acumulador, pero este no es el comportamiento real del ACV. El ACV se basa en el chip AD8337 y la ganancia se aplica exponencialmente controlada por un rango limitado de tensiones (V_{GAIN} entre 1,9 y 3,1 V). Por tanto, el multiplicador se ha sustituido por el modelo del ACV desarrollado en el Capítulo 2.
2. Se ha añadido un filtro promediador después del cálculo del valor absoluto para suavizar las transiciones y evitar que el AGC intente corregir el nivel con demasiada frecuencia. El filtro simplemente calculará la media entre los últimas M muestras:

$$\bar{z}[n] = \frac{1}{M} \sum_{i=0}^{M-1} |z[n-i]|. \quad (4.4)$$

3. El ACV aplica exponencialmente el factor de amplificación que genera el AGC. Por tanto, para que la señal de error del lazo no se vea alterada por el comportamiento exponencial, hay que incluir tras el filtro suavizador un operador que calcule el logaritmo de su salida. y por ende, también se hace del valor de referencia porque se suma a esa señal.

$$e[n] = \log_{10}(R) - \log_{10}(\bar{z}[n]). \quad (4.5)$$

4. Se ha añadido un modelo simple de ADC. Se compone únicamente de un saturador que limita el rango de la señal a ± 1 V y un cuantificador que limita la precisión a 14 bits.
5. Se ha incluido el modelo de conversor digital a analógico del Capítulo 3, compuesto por el módulo PWM y el filtro paso bajo. Además, como el PWM trabaja con valores enteros de 8 bits, la salida del acumulador se tiene que escalar por el factor 2^8 . Por otro lado, como el rango de tensiones que admite el ACV se debe introducir con un offset de 2,5 V, se tiene que añadir un sumador que sume dicho valor de offset.
6. Por último, una vez que el modelo de Simulink ha funcionado correctamente trabajando con formato *double*, se ha realizado un nuevo modelo incluyendo la precisión finita en todos los operadores. Los formatos numéricos aplicados en el modelo se muestran en la Tabla 4.1, donde el formato numérico se expresa como [N,F], siendo N el número de bits totales y F el número de bits fraccionales. Por otro lado, la letra que precede al formato indica si es codificación sin signo (u) o con signo en complemento a dos (s).

Salida del bloque	Cuantificación
ADC	s[14,13]
CIC	u[9,9]
LOG	u[16,15]
REF	s[17,15]
MULT_KP	s[39,37]
ACC	s[36,37]
PRE_PWM	u[8,8]

Tabla 4.1: Tabla de la cuantificación del AGC

El modelo completo del AGC se ha realizado con Simulink y se ha comprobado su correcto funcionamiento mediante simulación. Como demostración al modelo del AGC se le ha aplicado una señal de entrada sinusoidal de 1 MHz con valores de amplitud que varían entre 0,008 y 0,4 V. La frecuencia del PWM es de 51 MHz, el sistema tiene una frecuencia de muestreo de 250 MHz y el valor de referencia se ha fijado a 0,5. Los resultados se muestran en las Figuras 4.5 y 4.6, cuyas constantes de proporcionalidad son $kp = 2^{-14}$ y $kp = 2^{-7}$, respectivamente. Como en el segundo caso, el valor de kp es mayor que en el primero, este converge más rápido al valor de referencia, aunque aparece más rizado. En las dos Figuras se muestran dos versiones de aplicación del algoritmo AGC, la primera, titulada “Salida del modelo completo ideal”, se ha simulado con el modelo completo sin precisión finita y sin ADC. La segunda, titulada “Salida del modelo completo cuantificado”, incluye la precisión finita y el ADC, por eso en ésta se ve la señal saturada a ± 1 en algunos puntos.

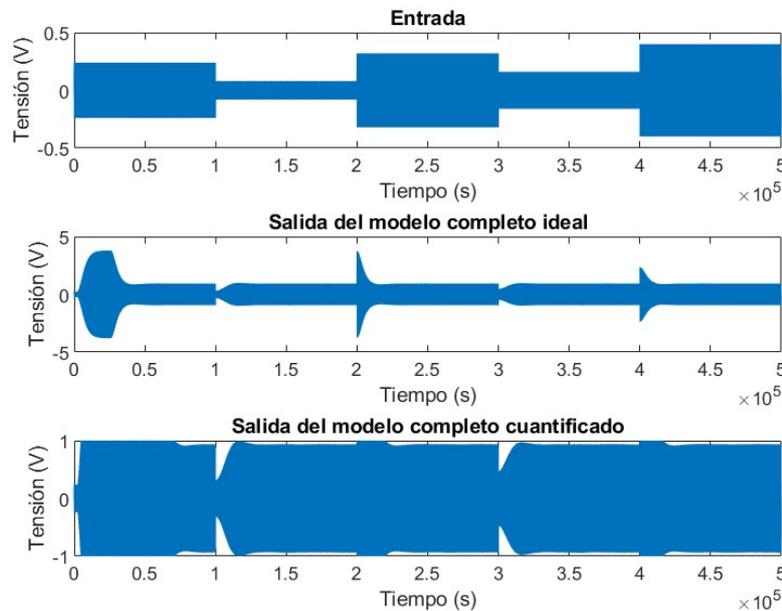


Figura 4.5: Resultados del modelo AGC completo con una $kp = 2^{-14}$

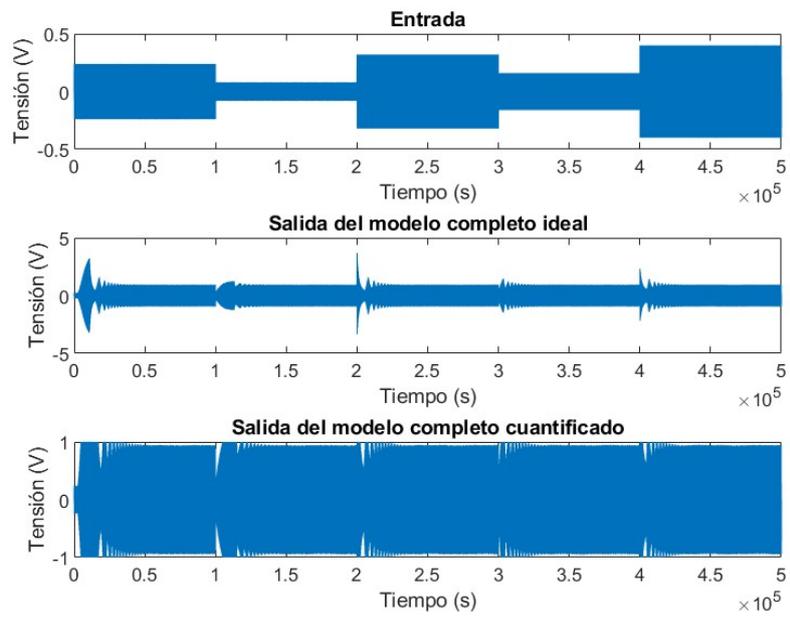


Figura 4.6: Resultados del modelo AGC completo con una $k_p = 2^{-7}$

Capítulo 5

Implementación en FPGA del control automático de ganancia

En este capítulo se detalla la implementación del algoritmo de control automático de ganancia en la FPGA a partir del modelo presentado en el Capítulo 4. Para facilitar su implementación se ha dividido en diferentes bloques, que serán verificados independientemente. Los bloques necesarios para el funcionamiento del AGC son los siguientes:

- Algoritmo del AGC
- PWM (Pulse Width Modulation)
- Bloque de comunicación

A parte de estos bloques, también hay otros bloques importantes para el correcto funcionamiento del sistema: el PLL, para generar relojes, y los sincronizadores, para intercambiar señales entre dominios de reloj. Estos bloques se comentarán al final del capítulo.

Por último, todos los bloques del algoritmo del AGC se han codificado en lenguaje System Verilog y se han verificado comparando la salida del modelo HDL con la del modelo de precisión finita de Simulink.

5.1. Implementación del algoritmo AGC

El lazo del AGC, tal y como se ha visto en capítulos anteriores, tiene muchas operaciones internas, y por eso, se ha dividido en diferentes bloques para hacer más sencilla su codificación y verificación. En la Figura 5.1 se muestran los distintos bloques que forman el algoritmo y los formatos numéricos utilizados en las conexiones entre los mismos. A continuación, en diferentes subsecciones, se explican los bloques.

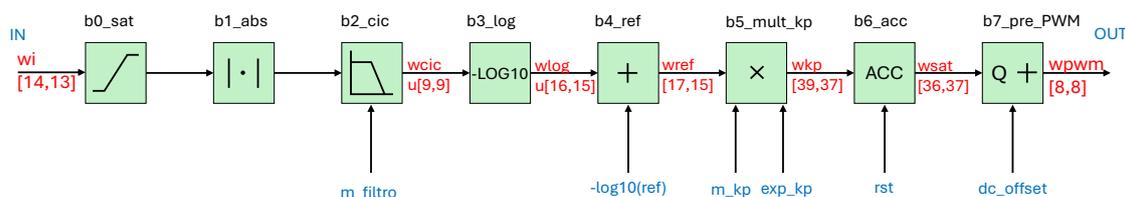


Figura 5.1: Diagrama de bloques de implementación del algoritmo AGC

5.1.1. Módulo saturador: b0_sat

En primer lugar, tras entrar la señal a la FPGA a través de un conversor analógico a digital ha sido cuantificada con un total del 14 bits siendo 13 de ellos fraccionales. La función de este bloque es saturar la señal entre $\pm(1 - 2^{-13})$ para que posteriormente, al calcular el valor absoluto, no haya valores fuera del rango posible de la cuantificación y se produzca un desbordamiento de la señal.

Además, como el saturador también se usa en otros bloques, se ha parametrizado la codificación para usarse en cualquier otra situación. Esta saturación se controla desde los bits más significativos (MSB), según la Tabla 5.1. Cuando los MSBs son 00 o 11 el valor de la señal no está entre los valores que se van a saturar, y por tanto, se queda igual. En cambio, si son 01 el valor es más alto que el nivel de saturación y por eso se queda en el valor máximo. Lo mismo ocurre cuando son 10, que el valor es menor al nivel de saturación mínimo y se queda en el valor mínimo.

2 MSBs	Bit de control
00	0
01	1
10	2
11	0

Tabla 5.1: Tabla del funcionamiento del bit de control del saturador

La Figura 5.2 muestra el esquema del circuito saturador.

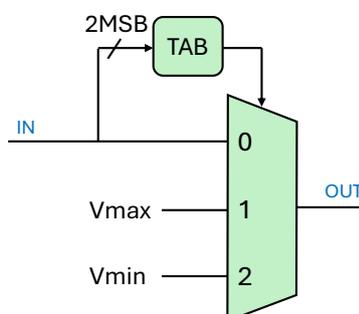


Figura 5.2: Esquema del saturador

5.1.2. Módulo valor absoluto: b1_abs

Este segundo bloque tiene la función de calcular el valor absoluto de la señal. Solo cambia el signo de la señal en caso de tener muestras negativas. No tiene verificación propia y se ha codificado directamente en el bloque general del lazo. Su esquema de implementación se muestra en la Figura 5.3.

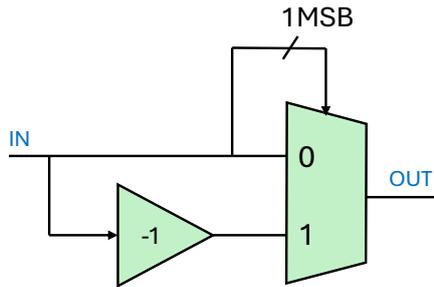


Figura 5.3: Diagrama del multiplexor para el valor absoluto

5.1.3. Filtro suavizador: b2_cic

Este bloque es un filtro suavizador, que realiza el promedio de M muestras. Se ha implementado con un filtro CIC (Cascade Integrator-Comb), que es equivalente a un filtro FIR discreto con M coeficientes de valor uno. Este filtro suma los valores de las últimas M muestras de entrada. Su salida hay que dividirla por la cantidad de valores que hay (M). Por lo tanto, para cada muestra de entrada calcula el valor medio de las últimas M muestras que van entrando.

Para la implementación del filtro partimos de la siguiente función de transferencia:

$$H(z) = 1 + z^{-1} + \dots + z^{-(M-1)}. \quad (5.1)$$

Esta función de transferencia se puede simplificar si se utiliza la fórmula de la suma de la progresión geométrica. Con ella llegamos a esta función de transferencia:

$$H(z) = \frac{1 - z^{-M}}{1 - z^{-1}} = \frac{Y(z)}{X(z)} \quad (5.2)$$

Lo desglosamos en denominador y numerador y obtenemos sus ecuaciones en diferencias finitas:

$$H_c(z) = 1 - z^{-M} = \frac{V(z)}{X(z)} \rightarrow v[n] = x[n] - x[n - M] \quad (5.3)$$

$$H_i(z) = \frac{1}{1 - z^{-1}} = \frac{Y(z)}{V(z)} \rightarrow y[n] = y[n - 1] + v[n] \quad (5.4)$$

Con estas fórmulas se obtiene el esquema de implementación de este bloque (Figura 5.4).

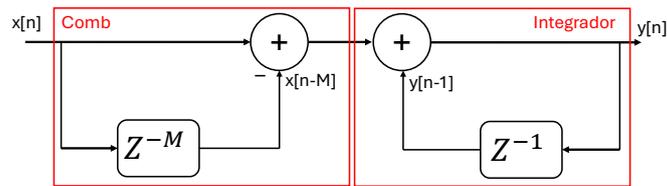


Figura 5.4: Circuito del filtro CIC

En la Figura 5.5 se muestra el esquema completo del filtro suavizador a implementar en la FPGA. Teniendo en cuenta que en el primer sumador se necesita sumar la muestra actual con la que llegó hace M instantes de muestreo, se ha creado un array de 16 registros para poder elegir el valor de M entre 2 y 17 (a parte de los registros del array, hay uno después del multiplexor). Además, en vez de implementar una división, que es un operador complejo de diseñar, se ha incluido un multiplicador a la salida del filtro para multiplicar por el valor $1/M$, cuyos valores han sido almacenados en una ROM. Este valor se saca de la ROM para todos los posibles valores de M . El valor de M se configurará a través de una entrada de este bloque.

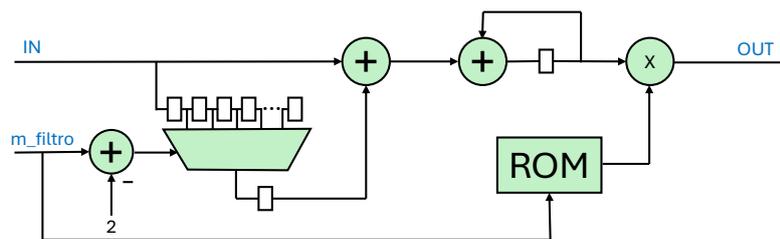


Figura 5.5: Esquema del filtro CIC

5.1.4. Módulo logaritmo: b3_log

Este bloque calcula el logaritmo en base 10 de la señal. Esta operación se ha implementado con una ROM. Se ha creado en Matlab un fichero con los resultados de esta operación para todos los posibles valores de entrada teniendo en cuenta la cuantificación que corresponde. Como 0 es un valor posible en la cuantificación de entrada, pero el resultado de este es ∞ , el primer valor del fichero (que debería ser ∞) es el mismo que el segundo valor, por tanto, la señal sale saturada en el límite inferior en $\log_{10}(1 - 2^{-9})$.

5.1.5. Módulo restador: b4_ref

Este bloque solo es un restador que resta un valor de referencia a la señal.

5.1.6. Módulo multiplicador: b5_mult_kp

En este bloque se calcula la multiplicación entre la señal y la constante de proporcionalidad. Los multiplicadores de la FPGA permiten la multiplicación de datos de hasta 18 bits. Estos multiplicadores no son suficientemente grandes para esta aplicación porque necesitamos multiplicar datos muy pequeños, por tanto, para su implementación, enviamos el dato en coma flotante: por una parte la mantisa, y por otra el exponente siguiendo la fórmula:

$$kp = m \cdot 4^{-exp} \quad (5.5)$$

En este caso se multiplica la señal por la mantisa, que tendrá 18 bits, y posteriormente, se hace un desplazamiento hacia la derecha de un total de bits igual al doble del valor del exponente. El esquema de implementación se muestra en la Figura 5.6.

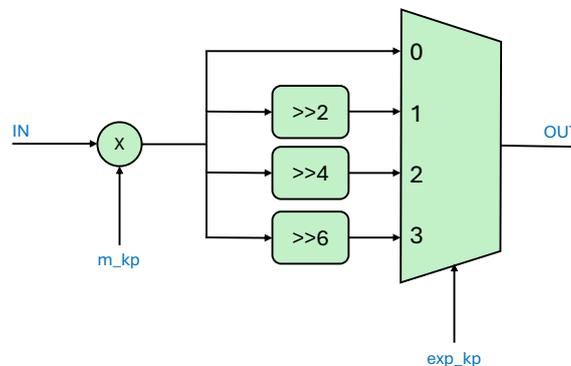


Figura 5.6: Esquema del multiplicador en coma flotante de la constante de proporcionalidad

5.1.7. Módulo acumulador: b6_acc

Este bloque es un acumulador que irá acumulando la señal de error tras multiplicarse por kp . Además incluye dos saturadores que limitan los valores a acumular a $\pm 0,25$, con el fin de limitar el rango de valores que se aplican a la entrada del DAC (el ACV solo admite entradas entre $\pm 0,6$ mV). Su esquema de implementación se muestra en la Figura 5.7.

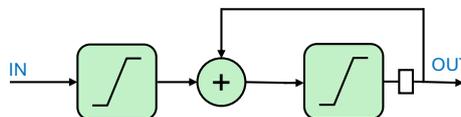


Figura 5.7: Esquema del acumulador

5.1.8. Módulo de adaptación al PWM: b7_pre_PWM

En este bloque se prepara la señal para posteriormente aplicarla al PWM. Para ello se cuantifica la señal a los 8 bits de entrada del PWM. Además, se suma un valor (DC Offset) para que no esté

centrado en 0, sino que el resultado final este entre 0 y 3,3 V. El esquema de implementación se muestra en la Figura 5.8.

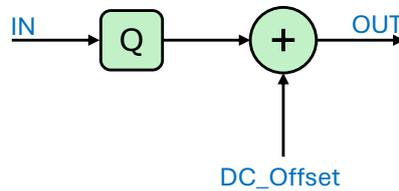


Figura 5.8: Esquema del prePWM

5.1.9. Verificación del módulo AGC

Todos los bloques se han codificado con System Verilog y sus verificaciones han seguido la misma metodología. Se ha creado un banco de pruebas en System Verilog que compara la señal de la simulación del modelo HDL con la señal del modelo de Simulink con precisión finita, la cual se encuentra almacenada en unos ficheros de texto. Compara todas las muestras y, en caso de ser alguna diferente, suma 1 en un contador de errores. Esto se reflejará en las capturas de las simulaciones de la siguiente manera:

- Aparecerán dos señales de cada bloque pintadas de blanco, uno nombrado como *[nombre del bloque]_data_M*, que son los resultados del módulo, y otro nombrado como *[nombre del bloque]_data_F*, que representa los resultados del modelo de Simulink.
- Posteriormente, está la señal contador del bloque (*sample_cnt*) pintada en naranja. Esta cuenta todas las muestras que hay en la señal del bloque para que, en caso de haber algún error, poder escribir un comentario diciendo en qué muestra no son iguales ambas señales.
- Por último, está el contador de errores (*error_cnt*) en color magenta. Este cuenta el número de muestras que no coinciden al comparar la señal del Simulink y del ModelSim.

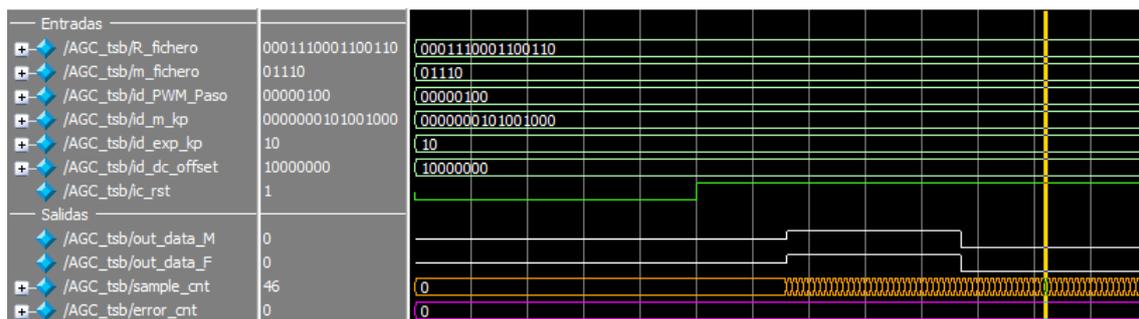


Figura 5.9: Validación del bloque del AGC

En primer lugar, se muestra en la Figura 5.9 las entradas y la salida del lazo AGC. La señal del contador de errores no muestra ningún error, pero, igualmente se va a mostrar la verificación de todos los bloques, uno por uno.

Los primeros bloques en ser verificados son el saturador (b0) y el valor absoluto (b1), se ve la simulación en la Figura 5.10. Tal y como vemos, las formas de onda de ambas señales son coherentes con las operaciones correspondientes. Como la señal de entrada no tiene una amplitud grande no se percibe la saturación en $b0_data_M$. En cambio, el valor absoluto se ve a simple vista que se ha calculado correctamente ($b1_data_M$).

Además, los resultados también son coherentes con la simulación de Simulink, ya que los contadores de errores no marcan ningún error.

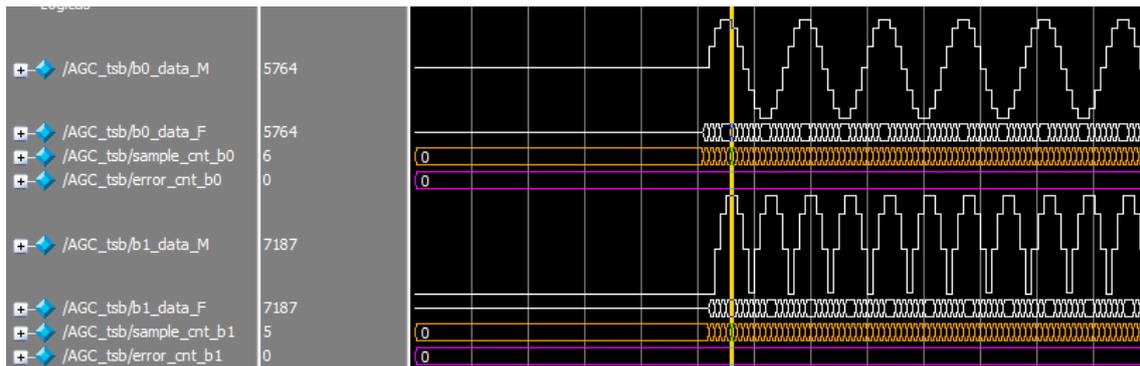


Figura 5.10: Validación del módulo de saturación y valor absoluto del AGC

El siguiente bloque a simular es el filtro suavizador CIC. Se han incluido dos simulaciones. En la Figura 5.11 el filtro se ha configurado con $M = 4$ y en la Figura 5.12 con $M = 14$. Se aprecia que con $M = 14$ el suavizado es mucho mejor.



Figura 5.11: Validación del filtro suavizador del AGC con $M=4$

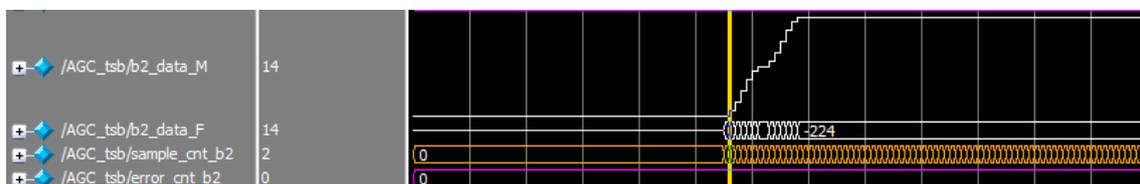


Figura 5.12: Validación del filtro suavizador del AGC con $M=14$

La siguiente verificación va dirigida a los bloques 3 y 4, es decir, el logaritmo y el de la resta del valor de referencia. La simulación está representada en la Figura 5.13.

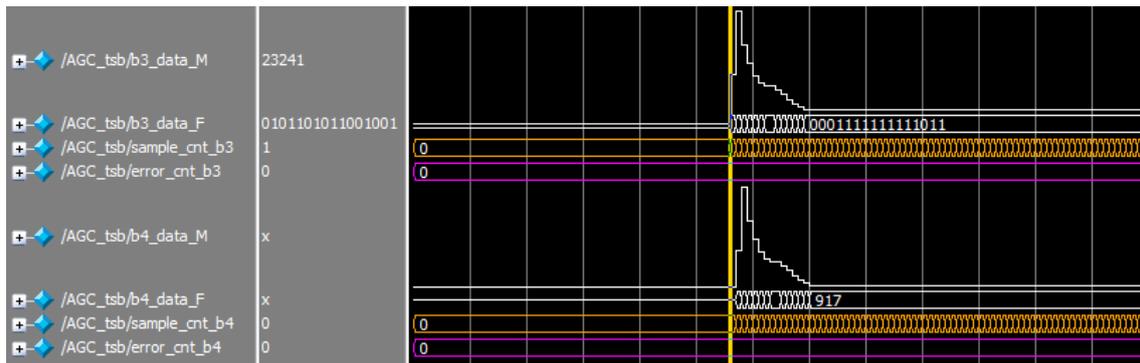


Figura 5.13: Validación del módulo logaritmo y señal de error del AGC

La explicación de la simulación de los dos siguientes bloques se realiza a la vez para mostrar el efecto de distintos valores de kp en el incremento del acumulador. Este efecto puede verse en las Figuras 5.14 y 5.15. En la primera simulación la kp tiene un valor 10 veces mayor que en la segunda simulación y, por tanto, la velocidad de la primera es mayor que la de la segunda.



Figura 5.14: Validación del módulo del multiplicador y acumulador del AGC siendo $kp=0,05$



Figura 5.15: Validación del módulo del multiplicador y acumulador del AGC siendo $kp=0,005$

Por último, en el AGC tenemos el bloque del prePWM que tiene un cuantificador y una suma. Véase simplemente que no tiene ningún error comparado con el modelo de Simulink en la Figura 5.16.

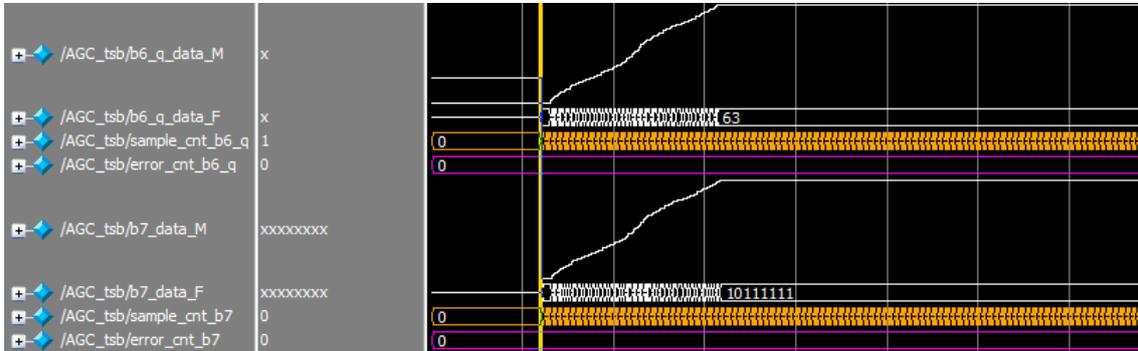


Figura 5.16: Validación del módulo prePWM

5.2. Implementación del PWM

El esquema del PWM puede verse en la Figura 5.17. Tal y como se ve en la anterior figura, hay un acumulador que suma la señal PASO, señal que determina la frecuencia del PWM. Posteriormente, se compara el contador con la señal de entrada IN.

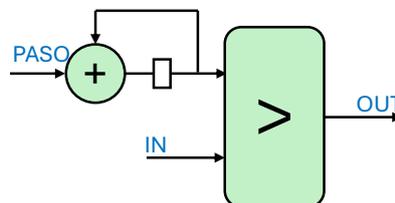


Figura 5.17: Esquema general del PWM

5.2.1. Verificación del módulo PWM

En las Figuras 5.18 y 5.19 se muestran el valor del contador del PWM y su salida, tras la comparación, cuando está configurado con $PASO = 1$ y entradas $IN = 190$ e $IN = 30$. Se observa la reducción del ciclo de trabajo de la señal PWM al disminuir el valor de la entrada IN.

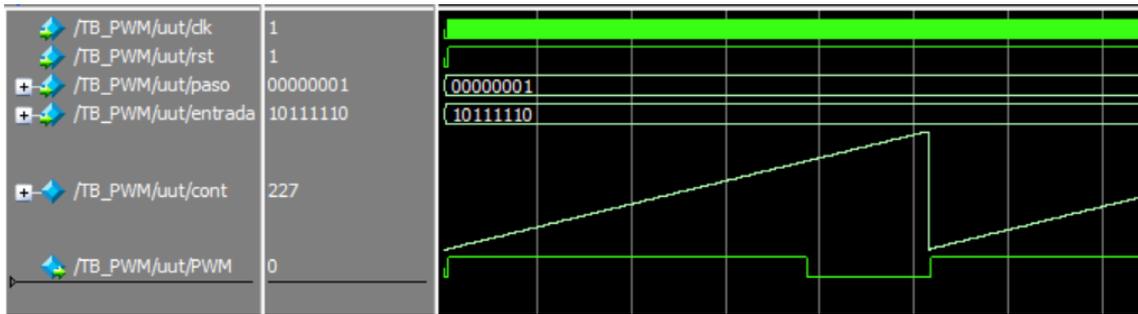


Figura 5.18: Simulación de ModelSym del PWM con $IN = 190$

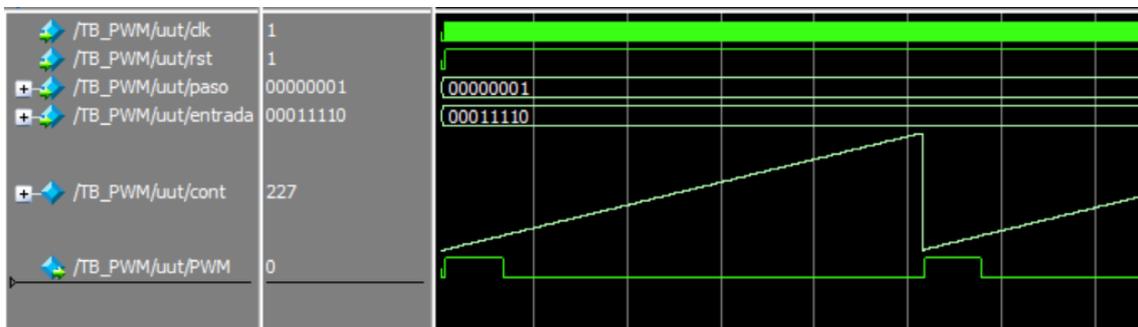


Figura 5.19: Simulación de ModelSym del PWM con $IN = 30$

5.3. Implementación del bloque de comunicación con el PC

En esta sección se describe la implementación de la comunicación entre el dispositivo FPGA y el PC, a través de un puerto serie RS232, para poder escribir y leer en el PC los valores para configurar el AGC. Se basa en una práctica realizada en la asignatura “Implementación de Sistemas de Comunicaciones” de la Especialidad de Sistemas de Telecomunicación, y se ha adaptado para transmitir los bytes necesarios para configurar el sistema AGC. En este caso se ha preparado para:

- enviar al PWM las señales PASO e IN;
- enviar al AGC la entrada M del filtro suavizador, el valor de referencia, la mantisa y el exponente de la constante de proporcionalidad, el offset del PWM y un byte de control que contiene un reset, un bit de control de un multiplexor, y otro bit que controla la congelación del acumulador del bloque 6.

Los valores de estos parámetros se agrupan en bytes y se transmiten byte a byte (siempre en el mismo orden) en serie desde el PC usando Matlab. Para transmitirlos y escribirlos en el dispositivo FPGA, o leerlos, se incluye al principio un byte de instrucción que indica si se va a realizar una operación de escritura o lectura de dichos parámetros.

El sistema se compone de 3 bloques (BCOM_RS232, BCOM_CONF_REGS y BCOM_CONF_FSMS) y su esquema de conexión se muestra en la Figura 5.20. El módulo BCOM_RS232 se

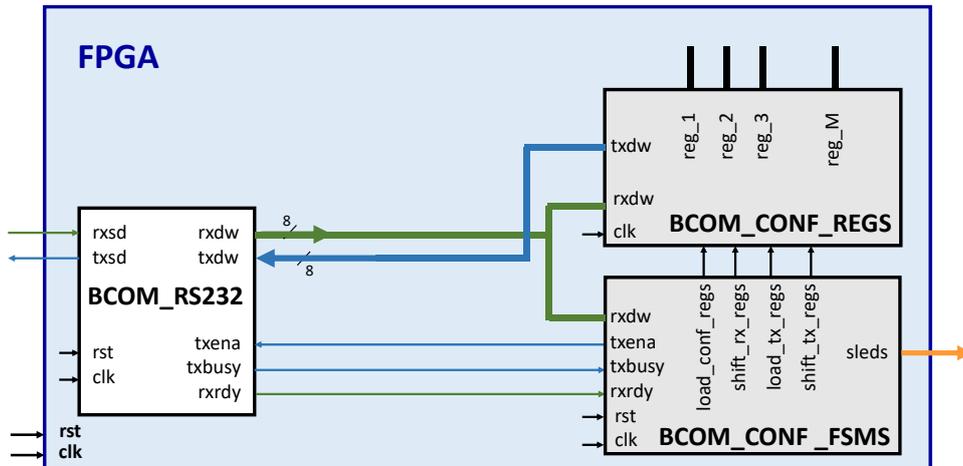


Figura 5.20: Esquema de la comunicación entre FPGA y PC

encarga de la implementación de la capa física de la comunicación serie. Este módulo no se ha realizado en este proyecto y se ha utilizado uno ya implementado. Los otros dos se han diseñado e implementado en este proyecto y se comentan a continuación.

5.3.1. Módulo BCOM_CONF_REGS

BCOM_CONF_REGS es el bloque que contiene los registros para la recepción y transmisión de los datos. Concretamente contiene 3 registros de M bytes conectados como se indica en la Figura 5.21. A este bloque le llegan M bytes en serie, desde rxdw, y se cargan en el registro de desplazamiento RX cuando la señal de entrada shift_rx_regs está activa. Al registro de configuración se le pasan los datos en paralelo cuando la señal de entrada load_conf_regs está activa. El contenido de este registro es el que se utilizará para configurar el AGC, por tanto, sus salidas se conectarán a los bloques del AGC y PWM. Todos estos bytes se enviarán al registro de desplazamiento TX cuando se active load_tx_regs y se sacarán en serie cuando se active shift_tx_regs por la salida txdw. Al enviarlos en serie podremos leerlos en el PC para comprobar que se han enviado correctamente al dispositivo FPGA.

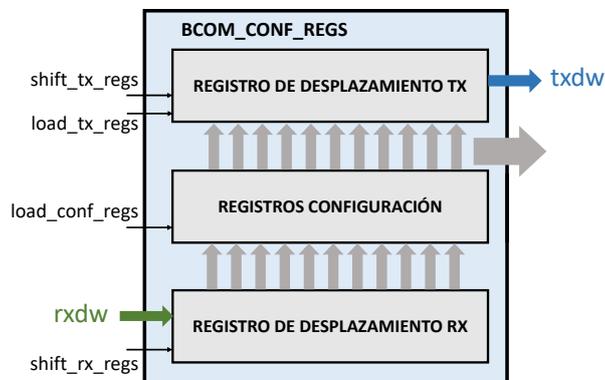


Figura 5.21: Esquema de BCOM_CONF_REGS

5.3.2. Módulo BCOM_CONF_FSMS

BCOM_CONF_FSMS es el bloque encargado de gestionar la comunicación. Instancia las tres máquinas de estados que necesita en funcionamiento: FSM_MAIN, FSM_WR y FSM_RD. Su esquema de conexión se muestra en la Figura 5.22. La primera de ellas se ocupa de leer los códigos que entran y detectar los códigos de las instrucciones lectura y escritura y en este caso activa las señales para que empiecen a trabajar las otras dos máquinas de estados. La segunda máquina de estados se ocupa de los procesos necesarios para la escritura del código, y la última, de la lectura.

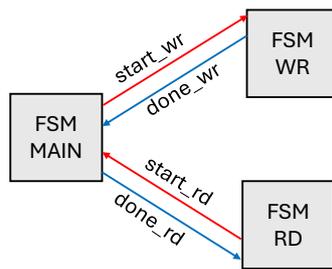


Figura 5.22: Esquema de BCOM_CONF_FSMS

El diagrama de estados de FSM MAIN se presenta en la Figura 5.23. En primer lugar, hay un estado que está a la espera de la entrada de algún dato para activar rxdy y pasar a otro estado donde se comprueba si el dato recibido es una de las instrucciones que activan una de las dos funciones (leer o escribir), en este caso es 1111000 para leer, y 00001111 para escribir. En caso de serlo, se activa start_rd o start_wr para pasar al estado de leer o de escribir, hasta que se reciba con done_wr o done_rd la indicación de que se ha finalizado la tarea correspondiente y vuelva al estado inicial de espera. Además, en caso de recibir una señal que no activa ninguna de las dos tareas, permanece en un estado de error hasta que recibe un código correcto.

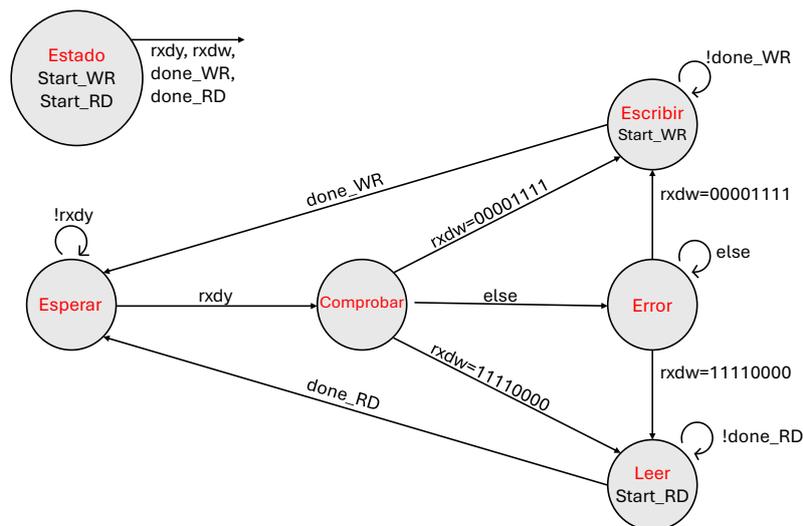


Figura 5.23: Diagrama de estados de la FSM MAIN

Las siguientes dos máquinas de estados son de alto nivel, por tanto, para facilitar su implementación, se ha diseñado un datapath para hacer otras operaciones. En este caso se ha implementado en el datapath un contador, un registro y un comparador. En la Figura 5.24 se detalla el esquema del datapath que utiliza tanto la máquina de estados de lectura como el de escritura.

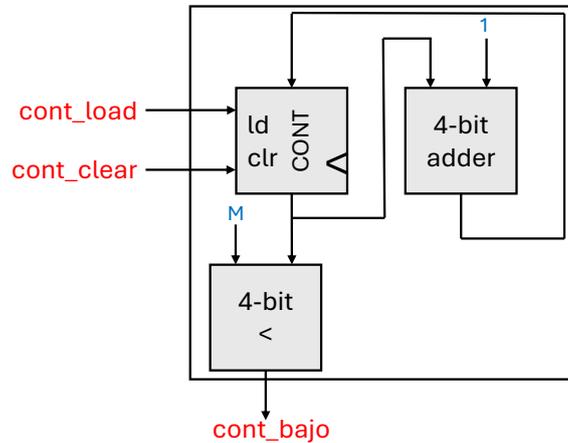


Figura 5.24: Datapath de las FSMs de lectura y escritura

En la figura 5.25 se muestra el diagrama de estados de la máquina de estados de escritura. Vemos que no se activa hasta obtener la señal de start_wr que se activa en fsm_main. Al activarse, espera a que se reciba otro dato, es decir, que se active rxdy, mientras que cont_bajo siga siendo 1, incrementa el sumatorio y hace el desplazamiento. En el momento en que cont_bajo ya no sea 1, se cargan todos los bytes del primer registro en el segundo y se finaliza la tarea activando la señal done_wr, haciendo un reset y volviendo a empezar.

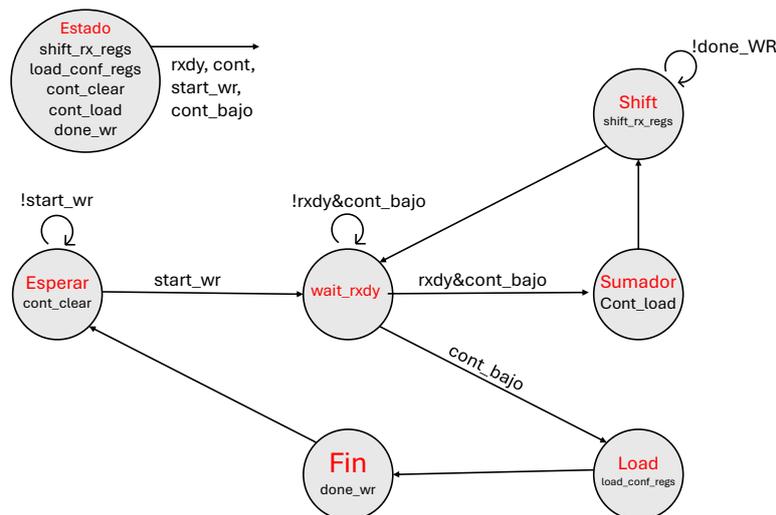


Figura 5.25: Diagrama de estados de la FSM de escritura

En la Figura 5.26 se muestra el diagrama de estados de la máquina de estados de lectura. Vemos que

no se activa hasta obtener la señal de start_rd que se activa en fsm_main. Al activarse, se cargan todos los datos en paralelo, posteriormente se transmite un dato, cuando termina la tarea suma un valor al sumador y desplaza el dato. En caso de haber terminado de transmitir y desplazar todos los datos pasa al estado fin, y si no, vuelve al bucle.

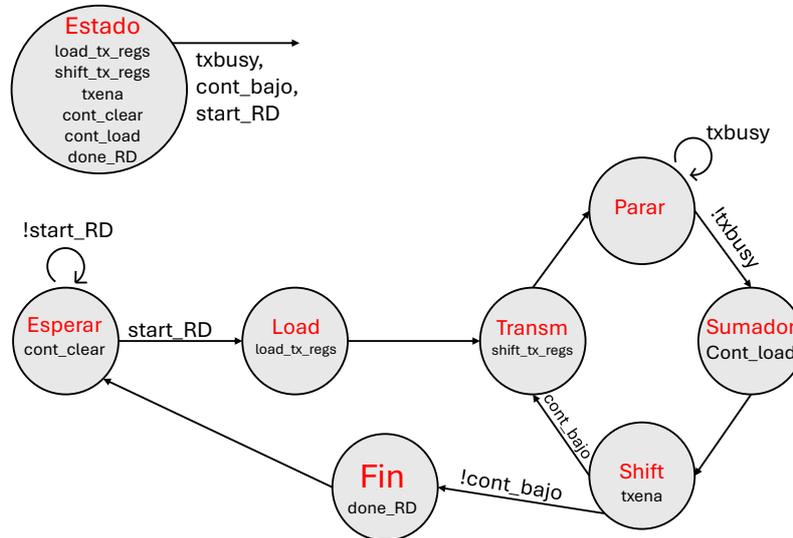


Figura 5.26: Diagrama de estados de la FSM de lectura

5.3.3. Verificación del módulo de comunicación con el PC

Para verificar el módulo de comunicación se ha realizado un script de Matlab que configura el puerto serie y envía y recibe 8 bytes. Se ha implementado el módulo en el dispositivo FPGA y se ha conectado al PC por el puerto USB-serie. En el dispositivo FPGA se ha utilizado la herramienta SignalTap para comprobar que los registros internos se escriben correctamente con los valores enviados desde Matlab.

5.4. Integración del sistema completo

Para finalizar la implementación del AGC en la FPGA falta unir estos bloques ya explicados anteriormente. El diagrama de bloques del sistema completo está en la Figura 5.27.

Los bloques PWM, algoritmo AGC y el módulo de comunicaciones funcionan con relojes diferentes. El PWM a 250 MHz, el AGC a 62,5 MHz y el módulo de comunicaciones a 230,400 kHz. Por ello, se han utilizado sincronizadores (registros de desplazamiento de 3 registros) para el intercambio de datos entre los diferentes dominios de reloj. Los 3 relojes se han generado mediante un PLL a partir de la frecuencia de 50 MHz que se genera en la tarjeta DE2-115.

Desde el bloque de comunicación se envía al AGC y al PWM los valores para configurarlos. En concreto se envían los siguientes datos al PWM:

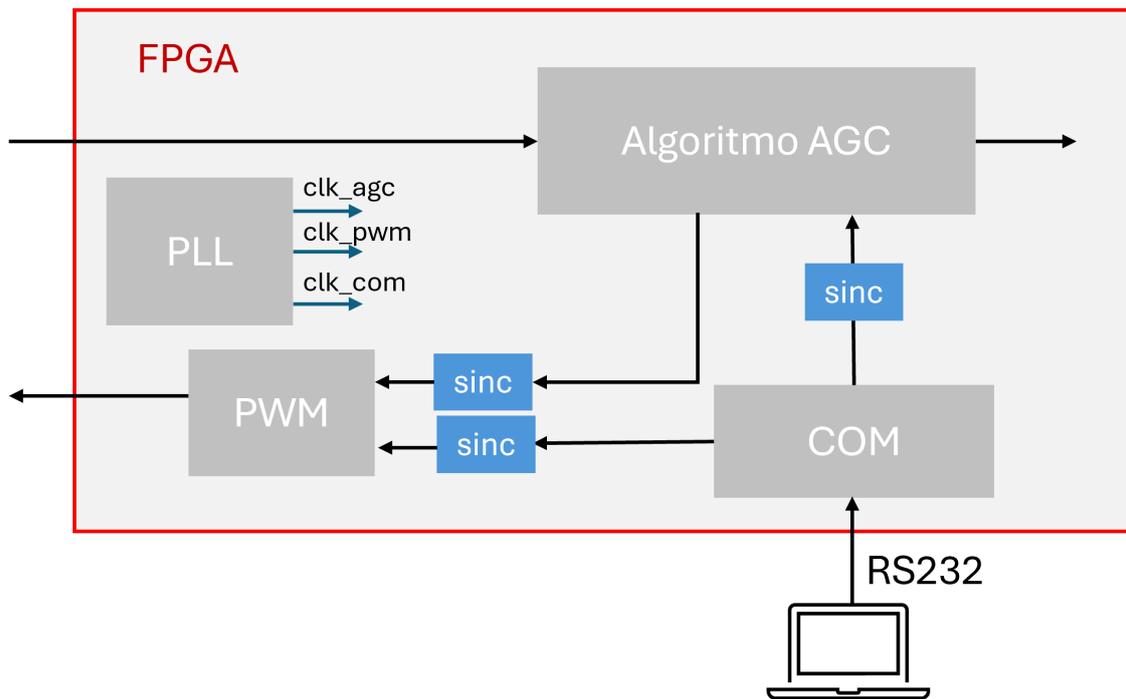


Figura 5.27: Diagrama de bloques del sistema completo

- PASO (1 byte): una señal que define cuanto suma el acumulador cada vez. Esto marca la frecuencia del PWM.
- IN (1 byte): una señal preparada para entrar al PWM.
- mux_IN (1 bit): una señal de control que elige si queremos que en el PWM entre una señal creada desde Matlab, o la señal del lazo.

Y los datos de configuración del lazo del AGC son los siguientes:

- M (5 bits): el número de muestras que se promedian en el filtro.
- R (2 bytes): el valor de referencia del AGC.
- m_kp (2 bytes): la mantisa de la constante de proporcionalidad en coma flotante.
- exp_kp (2 bit): el exponente de la constante de proporcionalidad en coma flotante.
- dc_offset (1 byte): valor para centrar la señal entre 0 y 255, los posibles valores de entrada del PWM.
- rst (1 bit): el reset del acumulador.
- congelar_acc (1 bit): una señal de control que detiene el acumulador o lo hace funcionar.

Capítulo 6

Validación del sistema

En este capítulo se van a presentar las pruebas realizadas para comprobar el correcto funcionamiento del AGC implementado en el dispositivo FPGA conectado a la tarjeta analógica con el ACV. Se presentan, en primer lugar, las pruebas realizadas con la instrumentación del laboratorio y, posteriormente, la comprobación realizada con el banco de pruebas de comunicaciones por luz visible.

6.1. Pruebas con la instrumentación del laboratorio

En esta sección se muestran las pruebas realizadas con el chip ACV y con el dispositivo FPGA, usando un generador de funciones y monitorizando la salida con un osciloscopio. Para ello, la señal digitalizada en el dispositivo FPGA se conecta a un convertor DAC, de forma se pueda visualizar el efecto de la aplicación del AGC.

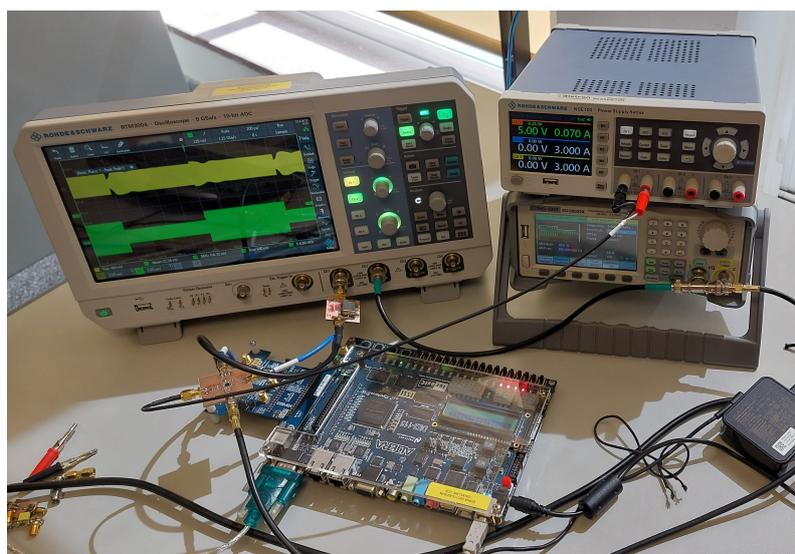


Figura 6.1: Banco de pruebas con la instrumentación del laboratorio

El banco de pruebas del experimento se puede ver en la Figura 6.1. Tal y como se puede ver se utiliza un osciloscopio con ambas señales: la de entrada (en verde), que se introduce con el generador de funciones, y la de salida (en amarillo), que se saca del dispositivo FPGA a través de un DAC. El generador de funciones se ha configurado para generar una señal modulada en AM de frecuencia de 6,25 MHz con una moduladora cuadrada de 1 kHz, cuya amplitud varía entre 250 mVpp y 100 mVpp. De esa forma se puede comprobar la forma de onda y tiempo de respuesta del AGC ante cambios de entrada abruptos.

La primera prueba se ha realizado parando el acumulador. Al configurarlo así, el lazo del control del AGC no está funcionando y, por tanto, la salida es idéntica a la entrada. En la Figura 6.1 se muestra la captura de la pantalla del osciloscopio.



Figura 6.2: Entrada (verde) y salida (amarilla) con el AGC parado

Si funcionase el acumulador, podría verse la señal de salida aumentar o disminuir, dependiendo del valor de amplitud, convergiendo al valor de la referencia.

El siguiente experimento se ha realizado reseteando el acumulador y, posteriormente, activándolo. Además, se ha configurado el AGC con una constante de proporcionalidad de $kp = 2^{-9}$, que es un valor grande para esta constante. El resultado se muestra en la Figura 6.3. El sistema tiene una respuesta rápida y una sobre-oscilación alta, pero converge al valor de referencia. Se aprecia que aunque cambie la amplitud de la señal, el AGC consigue que la señal siempre alcance la misma amplitud.

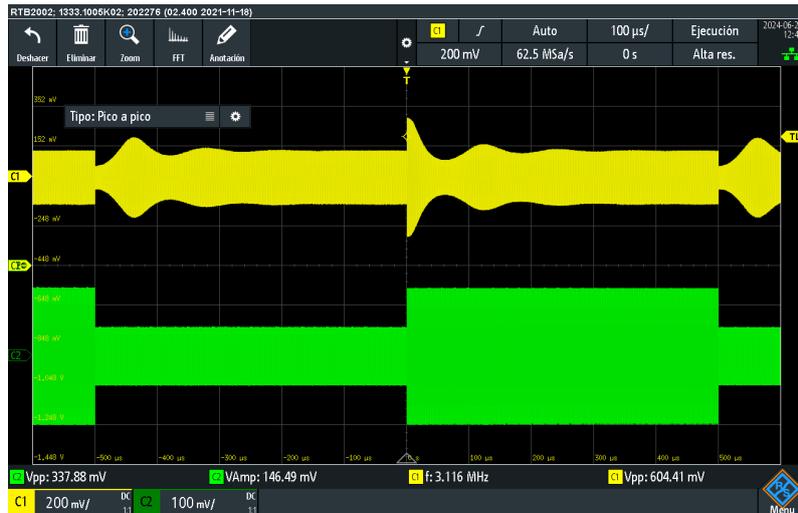


Figura 6.3: Entrada (verde) y salida (amarilla) con el AGC funcionando y $kp = 2^{-9}$

Por último, si se disminuye la constante de proporcionalidad se debe ver que la señal tarda más en llegar al valor de referencia, que llega más suave y sin tanto rizado. Para comprobar este efecto se ha cambiado la constante de proporcionalidad al valor $kp = 2^{-13}$. El resultado se muestra en la Figura 6.4.



Figura 6.4: Entrada (verde) y salida (amarilla) con el AGC funcionando y $kp = 2^{-13}$

6.2. Pruebas con el banco de pruebas de VLC

Un vez hemos comprobado que el AGC funciona correctamente con la instrumentación, se ha montado el sistema AGC en un banco de pruebas para la transmisión por luz visible. En la Figura 6.5 puede verse el banco de pruebas. Dispone de un LED de iluminación (a la izquierda en la foto)

que se utilizará como transmisor y un fotodiodo (a la derecha) que se utiliza como receptor. El LED está conectado al generador de funciones mediante un driver con un bias-T. El bias-T lo forman una condensador y una bobina y permite inyectar la señal al LED (a través del condensador), manteniéndolo alimentado (a través de la bobina). El fotodiodo está conectado al sistema AGC y su salida al osciloscopio. El diagrama de bloques del banco de pruebas se muestra en la Figura 6.6.

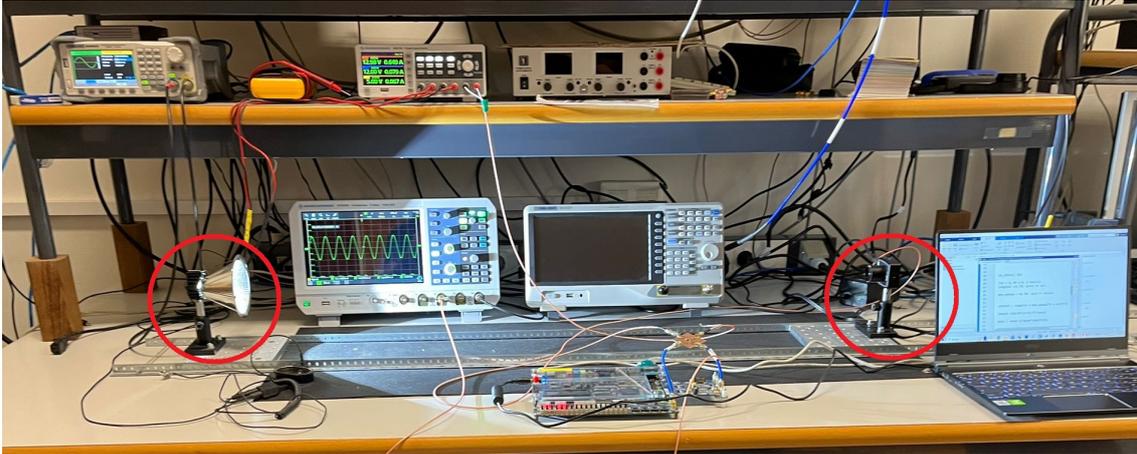


Figura 6.5: Banco de pruebas para la transmisión por luz visible

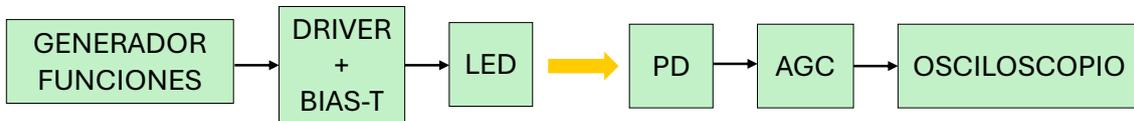


Figura 6.6: Diagrama de bloques del banco de pruebas

En este experimento se han hecho varias pruebas. En primer lugar se ha probado el sistema sin el acumulador del AGC. En este caso no debería funcionar nada y se vería en el osciloscopio que si el receptor está más cerca del transmisor la señal tiene mayor valor de tensión de pico a pico, y si está más lejos, una tensión más pequeña. Las Figuras 6.7 y 6.8 muestran esta situación. En la primera de ellas el receptor y el transmisor está más cerca que en la segunda, por eso, se puede ver que la señal del osciloscopio es mayor en la primera que en la segunda.

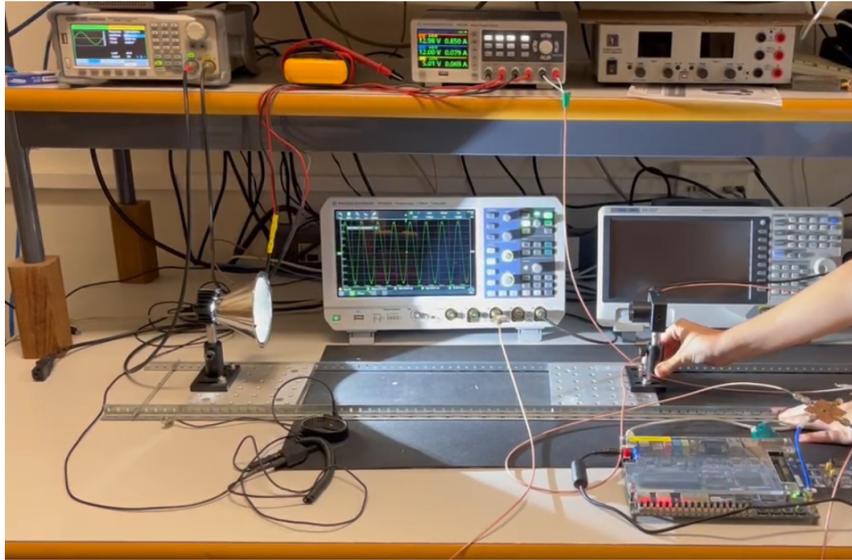


Figura 6.7: Prueba con transmisor y receptor cercanos y AGC desactivado

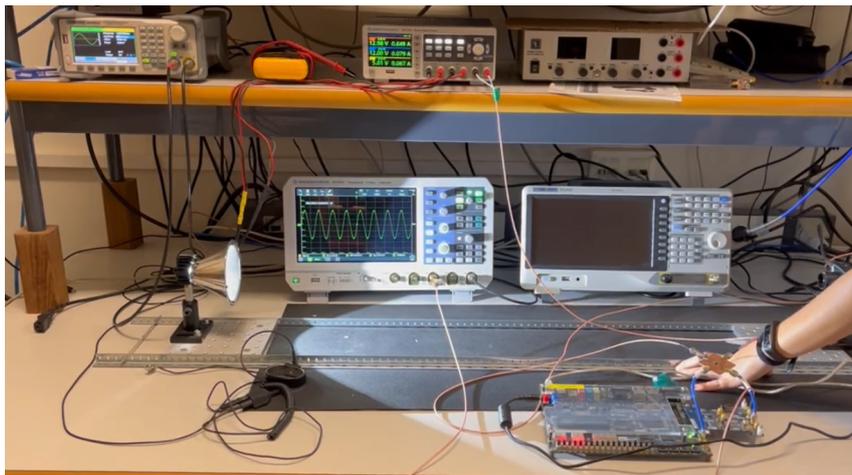


Figura 6.8: Prueba con transmisor y receptor lejanos y AGC desactivado

En la siguiente prueba se ha activado el AGC. Ahora, por mucho que se acerquen el transmisor y el receptor la señal no tiene que sufrir ninguna alteración. Las Figuras 6.9 y 6.10 muestran los resultados de esta segunda prueba. La primera de ellas es con el fotodiodo y el LED más juntos que la segunda, pero en esta, en cambio, se ve la señal del osciloscopio igual, porque esta funcionando correctamente el AGC.

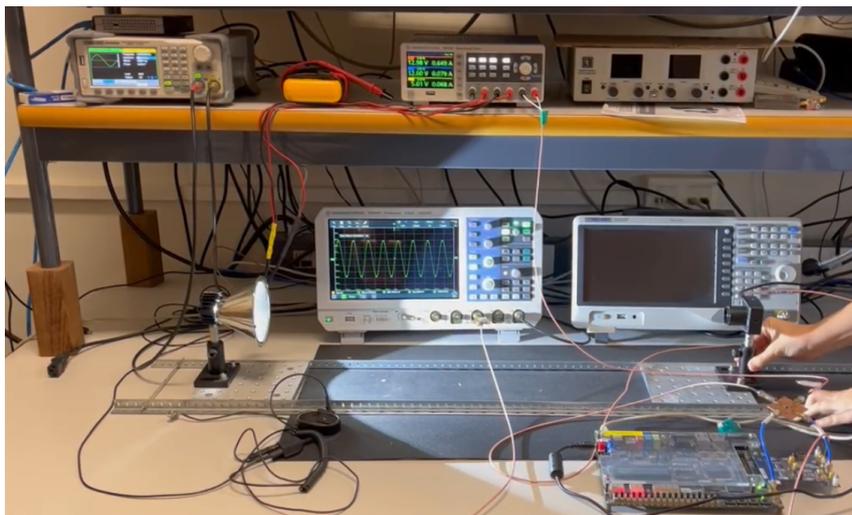


Figura 6.9: Prueba con transmisor y receptor cercanos y AGC activado

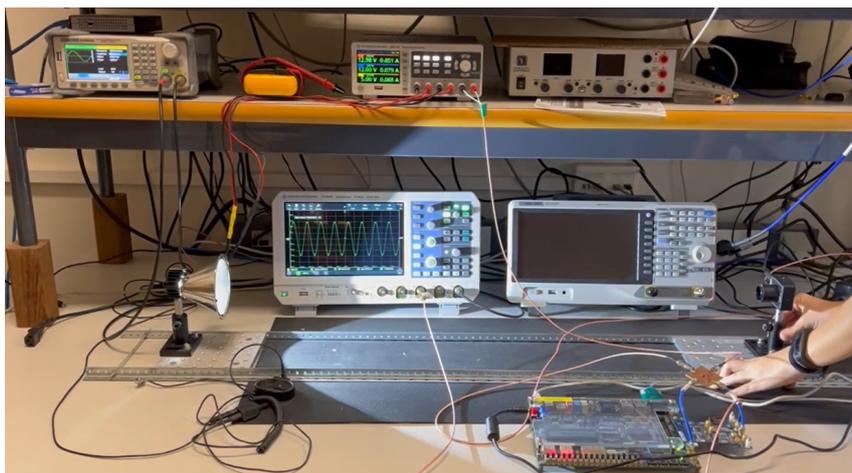


Figura 6.10: Prueba con transmisor y receptor lejanos y AGC activado

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

En este TFG se ha diseñado e implementado en hardware un algoritmo para realizar el control automático de ganancia (AGC) en un receptor. Este bloque se integrará en un sistema de transmisión de datos mediante luz visible y permitirá ajustar la señal recibida al rango de digitalización del conversor analógico-digital.

El AGC usa un amplificador controlado por voltaje y un lazo de control digital implementado en un dispositivo FPGA. La conversión digital-analógica necesaria para modificar la ganancia del amplificador se ha realizado mediante un modulador de anchura de pulsos implementado en el dispositivo FPGA y un filtro paso bajo pasivo.

Para el diseño del AGC se ha realizado un modelo con Matlab/Simulink tanto de la parte analógica como de la digital. El lazo de control se ha codificado con System Verilog HDL y se ha verificado usando el modelo de referencia realizado con Matlab. Además, se ha implementado un bloque de comunicación serie entre el dispositivo FPGA y un PC, con el que se puede configurar el lazo de control.

El sistema se ha validado transmitiendo una señal senoidal en el banco de pruebas para transmisión mediante luz visible disponible en el laboratorio, consiguiendo alcanzar los objetivos del proyecto.

7.2. Trabajos futuros

Por último, en cuanto a posibles líneas de mejora, es importante señalar que durante la fase de validación del AGC solo se ha comprobado cualitativamente su correcto funcionamiento. Sin embargo, no se ha estimado cuantitativamente el ruido que introduce el lazo de realimentación en el sistema ni su impacto en la transmisión de datos. Por lo tanto, este trabajo podría mejorarse midiendo cómo la activación del AGC afecta a las prestaciones de una transmisión real de datos y tratando de optimizar los parámetros de diseño (frecuencia, bits del PWM y coeficientes del filtro) para minimizar dicho ruido.

Otra posible mejora es añadir la circuitería necesaria para conmutar entre dos juegos de coeficientes de proporcionalidad del AGC: uno que acelera el tiempo de respuesta y otro que lo ralentiza.

Actualmente, el coeficiente se configura desde el PC, pero en un sistema autónomo debe estar fijado en el firmware del receptor. En este caso, sería útil disponer de un mecanismo de conmutación de coeficientes de forma que, si el AGC entra en su zona de saturación, conmute a un coeficiente rápido para facilitar la salida, y cuando está enganchado, use un coeficiente lento que minimice el ruido del lazo.

7.3. Relación con ODS

El desarrollo de sistemas VLC no solo representa una innovación tecnológica en el campo de las comunicaciones, sino que también tienen un impacto positivo en la sostenibilidad, la eficiencia energética y la infraestructura necesaria, alineándose con varios de los ODS de la ONU. Su contribución más significativa está en el ODS 7: Energía asequible y no contaminante.

Los sistemas VLC pueden utilizar la infraestructura de iluminación LED existente para transmitir datos, lo que permite una doble funcionalidad (iluminación y comunicación) sin necesidad de energía adicional significativa. Esto promueve el uso eficiente de la energía y apoya el desarrollo de tecnologías limpias y sostenibles.

Bibliografía

- [1] Parth H. Pathak et al. “Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges”. En: *IEEE Communications Surveys and Tutorials* 17.4 (2015), págs. 2047-2077. DOI: 10.1109/COMST.2015.2476474.
- [2] Pau Salvador et al. “Model and Methodology to Characterize Phosphor-Based White LED Visible Light Communication Links”. En: *Sensors* 23.10 (2023). ISSN: 1424-8220. DOI: 10.3390/s23104637. URL: <https://www.mdpi.com/1424-8220/23/10/4637>.
- [3] *AD8337: General-Purpose, Low Cost, DC-Coupled VGA*. Analog Devices. URL: <https://www.analog.com/en/products/ad8337.html>.
- [4] *Altera DE2-115 Development and Education Board*. TerasIC. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=502>.
- [5] *A High Speed AD/DA Card*. TerasIC. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=73&No=278&PartNo=1#contents>.
- [6] Richard G. Lyons. *Understanding Digital Signal Processing*. ISBN: 0-13-702741-9. Pearson Education, Inc., 2004.
- [7] Qasim Chaudhari. *Wireless Communications From the Ground Up*. ISBN: 1729732232. CreateSpace Independent Publishing Platform, 2018.