



DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.



UNIVERSITAT ROVIRA i VIRGILI

**DISEÑO E IMPLEMENTACIÓN DE
UN CONVERTOR ANALÓGICO
DIGITAL ESCALABLE Y
PARAMETRIZABLE EN UNA FPGA**

Juan David Espitia Castillo

TESIS DOCTORAL
2022

UNIVERSITAT ROVIRA I VIRGILI

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo

UNIVERSITAT ROVIRA I VIRGILI

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo

UNIVERSITAT ROVIRA I VIRGILI

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Tesis doctoral

Juan David Espitia Castillo

Supervisado por:

Ph.D. Enrique Cantó Navarro

Ph.D. Enric Vidal-Idiarte

Departament D'Enginyeria Electrònica, Elèctrica i Automàtica Tarragona,
Espanya

Grupo de investigación:

Group of Automatic Control Industrial Electronics

Tarragona, España

UNIVERSITAT ROVIRA I VIRGILI

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo



FAIG CONSTAR que aquest treball, titulat "DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA.", que presenta Juan David Espitia Castillo per a l'obtenció del títol de Doctor, ha estat realitzat sota la meua direcció al Departament D'enginyeria electrònica Elèctrica i Automàtica d'aquesta universitat.

HAGO CONSTAR que el presente trabajo, titulado "DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA", que presenta Juan David Espitia Castillo para la obtención del título de Doctor, ha sido realizado bajo mi dirección en el Departamento D'enginyeria electrònica Elèctrica i Automàtica de esta universidad.

I STATE that the present study, entitled "DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA.", presented by Juan David Espitia Castillo for the award of the degree of Doctor, has been carried out under my supervision at the Department D'enginyeria electrònica Elèctrica i Automàtica of this university.

Tarragona, 25 de febrero de 2022

El/s director/s de la tesi doctoral
El/los director/es de la tesis doctoral
Doctoral Thesis Supervisor/s

Enrique Cantó Navarro
Firmado digitalmente por Enrique Cantó Navarro
Fecha: 2022.02.25 10:24:18
[firma] [firma] [firma]

Enrique Cantó Navarro

Enric Vidal Idiarte - DNI 40931578L (AUT)
Firmado digitalmente por Enric Vidal Idiarte - DNI 40931578L (AUT)
Fecha: 2022.02.25 11:03:11 +01'00'
[firma] [firma] [firma]

Enric Vidal-Idiarte

UNIVERSITAT ROVIRA I VIRGILI

DISEÑO E IMPLEMENTACIÓN DE UN CONVERTOR ANALÓGICO DIGITAL ESCALABLE Y PARAMETRIZABLE EN UNA FPGA

Juan David Espitia Castillo

AGRADECIMIENTOS

A lo largo del tiempo de desarrollo de este trabajo, he conocido y trabajado con diferentes personas que han aportado en diferentes maneras a la finalización de esta tesis. Es por esto que quisiera agradecer a todas las personas que permitieron que esto fuese una realidad. En primer lugar, agradecer a mis padres Diana Castillo y Humberto Espitia que junto a mi hermana Valentina Espitia y abuelos Foscarina Mejia, Jaime Castillo (QEPE) y Doralia Colonia (QEPE) son quienes me han apoyado en todas las maneras posibles.

A mis tutores de tesis Enrique Cantó Navarro PhD. y Enric Vidal-Idiarte PhD. por guiarme en todo el proceso, por su paciencia y continuas enseñanzas. Al grupo de investigación GAEI por su apoyo y colaboración en toda mi estancia en la universidad.

También agradecer los amigos que estuvieron en algún punto del proceso, a Eduardo, Yohana, Silvana, Catalina, Giancarlo, Lizeth, Patricia, Federica y demás personas que formaron parte de este viaje. Por último, a Patrycja Derebecka, porque ha sido el apoyo más importante en el desarrollo de esta tesis.

Resumen

La flexibilidad brindada por las FPGAs (Field Programmable Gate Arrays) permite la implementación de uno o varios conversores análogos digitales (ADC), cada uno configurado con una resolución y frecuencia de muestreo ($f_{Sampling}$) específicos, delimitado por la aplicación. Esta tesis doctoral presenta dos diseños para la implementación de un ADC de N-bit escalable y parametrizable en FPGA. EL primero es basado en el circuito one shot ADC y el segundo basado en SAR (Successive Approximation Register).

El primer diseño es un ADC de N-bit basado en el circuito one Shot, que permite la implementación del ADC a partir de un circuito RC y puertas lógicas. Se presenta una metodología sistemática para el diseño del ADC de N-bit a partir de la resolución, $f_{Sampling}$ deseada y rango de la tensión de entrada (V_{in}). La lógica del one shot es sintetizable y parametrizable, con pocos recursos de la FPGA utilizados y que puede ser extrapolable a otras familias de FPGA.

El segundo diseño es un ADC de N-bit basado en SAR a partir de diferentes módulos de implementación, como el modulador de ancho de pulso (PWM), filtro analógico paso bajo (LPF) y un comparador analógico. Se presenta una metodología sistemática que permite escoger los parámetros del LPF para un ADC con características específicas (resolución y $f_{Sampling}$).

Por último, se presenta diferentes optimizaciones en el módulo del PWM para mejorar $f_{Sampling}$ de los ADC de N-bit basado en SAR. El método para obtener los parámetros del LPF es adaptado a cada optimización. El PWM y la lógica SAR es sintetizable y parametrizable, utilizando pocos recursos, con el objetivo de ser portable a familias de FPGAs de bajo costo. La metodología y optimización ha sido comprobada en la FPGA Zynq-7000 de Xilinx, aunque es posible adaptarse a cualquier otra familia de FPGA.

Keywords – Conversor analógico digital (ADC), Successive approximation register (SAR), FPGA, Frecuencia de muestreo, one shot, Modulador de ancho de pulso (PWM), Conversor digital analógico (DAC), Filtro paso bajo.

Abstract

The flexibility provided by FPGAs permits the implementation of several ADCs, each one configured with the bit resolution and the sampling frequency required by the target application. The doctoral thesis presents two designs for the implementation of scalable and parametrizable N-bit Analog-to-Digital Converters (ADC) on FPGAs (Field Programmable Gate Arrays). The first design based on one shot circuit and the second design is based on a SAR (Successive Approximation Register).

The first design is N-bit ADC based on the one shot circuit. Combining a RC circuit and logic gates the ADC is implemented. A methodology for the implementation of a parametrizable one shot-based ADC is presented. Based on the $f_{Sampling}$, V_{in} range and resolution the parameters for the implementation are found. The one shot logic is synthesizable and parametrizable, using a low number of resources, in order to be portable to low-cost FPGA families.

The second design is an SAR-based ADC from a set of building modules, such as the Pulse-Width Modulator (PWM), external Low-Pass Filter (LPF) and the analog comparator. A systematic methodology for the implementation of a parametrizable SAR-based ADC from a set of building modules, such as the Pulse-Width Modulator (PWM), external Low-Pass Filter (LPF) and the analog comparator is presented. The presented method permits to choose the LPF parameters for the required performance (resolution bits and $f_{Sampling}$) of a SAR-based ADC.

Lastly, the thesis also presents several optimisations on the PWM module to enhance the sampling frequency of implemented ADCs, and the method to choose the LPF parameters is adapted. The PWM and SAR logic are synthesizable and parametrizable, using a low number of resources, in order to be portable to low-cost FPGA families. The methodology and PWM optimisations have been tested on a Zynq-7000 device from Xilinx despite they can be adapted to any other FPGA.

Keywords – analog-to-digital converter (ADC); One shot ADC; successive approximation register (SAR); FPGA; PWM; LPF

Índice

Resumen

Abstract

Abreviaciones 10

I. Introducción 12

I.I. Antecedentes	14
I.II. Objetivos	17
I.II.1. Objetivo general	17
I.II.2. Objetivos específicos	17
I.III. Contribución de esta tesis	18
I.IV. Organización del escrito	19

1. Capítulo 1 20

1.1. Resumen	20
1.2. One shot ADC	20
1.3. Simulación del ADC basado en el circuito one shot en una FPGA	22
1.3.1. Análisis de la linealidad de la respuesta del ADC basado en el circuito one shot	24
1.4. Método para obtener parámetros del ADC basado en el circuito one shot	31
1.5. Limitaciones de la implementación one shot ADC en una FPGA	32
1.6. Implementación del one shot ADC de 8-bit en una FPGA	32
1.7. Síntesis del diseño del ADC basado en one shot	35
1.7.1. ADC de 10-bit basado en el circuito one shot	35
1.8. Posibles métodos de mejora	37
1.9. Conclusiones	37

2. Capítulo 2 39

2.1. Resumen	39
2.2. ADC basado en SAR	39
2.2.1. DAC implementado con un LPF y un PWM	43
2.2.2. Diseño e implementación del PWM	45
2.2.3. Filtro pasa bajo	46
2.2.4. Implementación del filtro pasa bajo	52
2.2.5. Problemas prácticos de implementación de filtros	54
2.2.6. Análisis en frecuencia del PWM	56
2.2.7. Obtención de los parámetros del LPF para un ADC de N-bits basado en SAR	59
2.2.8. Rizado en estado estacionario	62
2.3. ADC de 8-bit basado en SAR con PWM y LPF	63
2.4. Síntesis del diseño del ADC basado en SAR	69
2.5. ADC de 10-bit basado en SAR con PWM y LPF	70
2.6. Conclusiones	73

3. Capítulo 3 74

3.1. Resumen	74
3.2. Diseño de un PWM por doble flanco de reloj	74
3.3. Obtención parámetros del LPF	76
3.4. ADC de 8-bit basado en SAR con PWM-DDR y LPF	77
3.4.1. Simulación de un ADC de 8-bit basado en SAR con PWM-DDR y LPF	79
3.4.2. Implementación ADC de 8-bit basado en SAR con PWM-DDR y LPF	81
3.5. ADC de 10-bit basado en SAR con PWM-DDR y LPF	84
3.6. Conclusiones	85
4. Capítulo 4	87
4.1. Resumen	87
4.2. División del PWM de N-bit en 2 PWMs de $\frac{N}{2}$ -bit	87
4.2.1. Obtención parámetros del LPF para un ADC de N-bit con 2 PWMs de $\frac{N}{2}$ -bit	89
4.2.2. ADC de 8-bit con 2 PWMs de 4-bit	91
4.3. División del PWM en K PWMs de $\frac{N}{K}$ -bit	95
4.3.1. Obtención de los parámetros del LPF para un ADC de N-bits con K PWMs de $\frac{N}{K}$ -bit	99
4.3.2. ADC de 8-bit con 4 PWM-DDR de 2-bit	101
4.3.3. ADC de 8-bit con 8 PWMs de 1-bit	103
4.3.3.1. Implementación ADC de 8-bit con 8 PWMs de 1-bit	105
4.4. Síntesis del diseño del ADC de N-bit con DAC con K PWMs de $\frac{N}{K}$ -bit	107
4.5. Diseño de un ADC de 10-bit	109
4.6. Conclusiones	112
5. Capítulo 5	114
5.1. Resumen	114
5.2. PWM con ciclo de trabajo distribuido	114
5.3. Obtención de los parámetros del LPF con PWM-DDC	118
5.4. ADC de 8-bit con PWM-DDC para diferentes optimizaciones	124
5.4.1. Simulación de un ADC de 8-bit con PWM-DDC-DDR de $\frac{N}{2}$ -bit	125
5.4.2. Implementación de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit	126
5.4.3. Simulación de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit	127
5.4.4. Implementación de un ADC de 8-bit con K=4 y PWM-DDC-DDR de 2-bit	129
5.5. Conclusiones	130
6. Capítulo 6	132
6.1. Resumen	132
6.2. Generalización sistemática de la arquitectura del ADC basado en SAR	132
6.3. Diseño de un ADC de 8-bit con $f_{Sampling}=800$ kSPS	140
6.4. Conclusiones	145
7. Conclusiones	146

Índice de tablas

1.1. Características del circuito one shot para $T_{in} = 1 \mu s$ con comportamiento exponencial y aproximadamente lineal	27
1.2. B, V_{in}, T_{in} y número de cuentas posibles para un one shot con $T_{inMax}=1.28 \mu s$	34
1.3. Características del one shot para ADC de 10 bits con $f_{Sampling} = 8,94 kSPS$	36
1.4. B, V_{in}, T_{in} y número de cuentas posibles para un one shot con $f_{Sampling} = 8,94 kSPS$	37
2.1. $t_{Settling}$ normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .	51
2.2. Parámetros que afectan la dinámica del LPF	52
2.3. Valores normalizados a $f_c = 1 Hz$ del LPF Bessel	53
2.4. Comparativa de diferentes tipos de condensadores.	55
2.5. $O_{LPF}, f_c, t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 6-bit	60
2.6. Parámetros para la implementación del ADC de 8-bit	64
2.7. $O_{LPF}, f_c, t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 8-bit.	64
2.8. $O_{LPF}, f_c, t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 10-bit.	70
2.9. $O_{LPF}, f_c, t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 10-bit.	70
3.1. Comparación de las características de la implementación del LPF con PWM y PWM-DDR, para un ADC de 6-bit.	77
3.2. $O_{LPF}, f_c, t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$ para un ADC de 8-bit.	78
4.1. Características de la implementación del ADC de 8-bit con 2 PWM-DDR	94
4.2. Comparación de las características de implementación del LPF para un ADC de 6-bit con PWM-DDR y diferentes K.	101
4.3. Comparación de las características de la implementación del ADC de 8-bit con 2 PWM-DDR y 4 PWM-DDR	102
4.4. Comparación de las características de implementación del ADC de 8-bit con PWM-DDR y diferentes diseños	105
4.5. Comparación de las características de implementación del ADC de 10-bit con los diferentes diseños con $O_{LPF}=6$	112
5.1. Comparación de las características de implementación del LPF para un ADC de 6-bit con PWM y PWM-DDC de 6-bit	121
5.2. Comparación de las características de implementación del LPF para un ADC de 6-bit con PWMs y PWM-DDC de 3-bit.	122
5.3. $t_{Settling}$ normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .	122
5.4. Características del ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit.	125
5.5. Características del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.	128
6.1. $t_{Settling}$ normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .	136
6.2. Valores normalizados a $f_c = 1 Hz$ del LPF Bessel	138
6.3. Características del ADC de 8-bit con 2 PWM-DDC-DDR de 3-bit y uno de 2-bit.	142
6.4. Características del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.	143

Índice de figuras

1.1. Circuito one shot	20
1.2. Funcionamiento gráfico del one shot ADC	21
1.3. Comportamiento de V_C y V_{NOT} con $V_{in}=0$ V	23
1.4. Comparación del comportamiento de V_C y V_{NOT} con diferentes valores de V_{in}	24
1.5. T_{in} a partir de un barrido de V_{in}	25
1.6. B obtenida a partir de barrido de V_{in}	26
1.7. Comparación de la curva de carga de V_C de comportamiento exponencial y comportamiento aproximadamente lineal	27
1.8. T_{in} a partir de barrido de V_{in}	28
1.9. B obtenida a partir de barrido de V_{in}	28
1.10. V_C y V_{NOT} con error en descarga del condensador para $V_{in}=0$ V y $V_{Sumador}=3.5$ V	29
1.11. V_C y V_{NOT} considerando t_D del condensador	30
1.12. V_C y V_{NOT} con $f_{Sampling} = 1$ MSPS	31
1.13. Esquemático de la simulación del one shot ADC en Simulink-Matlab	33
1.14. B de 8-bit con $f_{Sampling}=42.518$ kSPS	33
1.15. Diferencia entre los pulsos adyacentes de un one shot ADC de 8-bit con $T_{inMax} = 1,28$ μ s	34
1.16. Resultado de la conversión del one shot de 10 bits con $f_{Sampling} = 8,94$ kSPS	36
2.1. Esquemático de un ADC basado en SAR	39
2.2. V_{DAC} y V_{in} en ADC basado en SAR de 7-bit	40
2.3. V_{DAC} y V_{in} para un ADC basado en SAR de 4-bit	41
2.4. V_{DAC} y V_{in} para un ADC de 4-bit basado en SAR con modificación en el estado inicial	42
2.5. V_{DAC} y V_{in} para un ADC basado en SAR de 8-bit	42
2.6. Esquemático del ADC basado en SAR con modificación del DAC	43
2.7. V_{LPF} para PWM de 3-bit y LPF con $f_C=4$ MHz y $O_{LPF}=3$, variando d entre 0 y $2^N - 1$	44
2.8. Maquina de estados del PWM	45
2.9. Comportamiento del contador ascendente con un PWM de 3 bits	46
2.10. Esquemático de la implementación de un contador ascendente para el PWM	46
2.11. Diagrama de Bode para las diferentes arquitecturas del LPF	48
2.12. Comparación de la respuesta al escalón de diferentes arquitecturas de LPF, con $f_C=1$ Hz y $O_{LPF}=3$	48
2.13. Ubicación de los polos del LPF de arquitectura Bessel de $O_{LPF}=5$	49
2.14. Comparación de la respuesta al escalón del LPF con $O_{LPF}=4$ y diferentes f_C	49
2.15. Comparación de la respuesta al escalón del LPF con $\omega_C = 2\pi \cdot 13 \cdot 10^6 \frac{rad}{s}$ para diferentes O_{LPF}	50
2.16. Respuesta al escalón normalizado del LPF Bessel para una $f_C=1$ Hz con diferentes O_{LPF}	50
2.17. $t_{Settling}$ Vs. f_C para diferentes O_{LPF} con f_C desde 1 Hz hasta 20 Hz	51
2.18. Arquitectura Sallen-Key para LPF de $O_{LPF}=2$	52
2.19. Efecto de la temperatura en un condensador de "thick film"	54
2.20. V_{PWM} y V_{LPF} utilizando un LPF Bessel con $f_C=200$ kHz, $O_{LPF}=3$, $f_{CLK}=100$ MHz, PWM de 8-bit y $D = \frac{128}{256}$	56
2.21. Esquemático de la magnitud de la respuesta en frecuencia del LPF	59

2.22. Magnitud de la respuesta en frecuencia del LPF con diferentes O_{LPF} . . .	60
2.23. Respuesta al escalón de los LPFs con f_C y O_{LPF} hallados	61
2.24. Fase y respuesta al escalón de los diferentes LPF con f_C y O_{LPF} hallados	62
2.25. V_{DAC} de $f_C=60$ kHz, $O_{LPF}=3$ con ciclos de trabajo de $\frac{128}{256}$ y $\frac{129}{256}$	62
2.26. V_{DAC} con LPF de $O_{LPF}=3$ y con $f_C=40$ kHz y $f_C=50$ kHz, el PWM con $D = \frac{128}{256}$	63
2.27. Parámetros del LPF con diferentes O_{LPF} implementando un ADC de 8-bit con $f_{Sampling} = 8$ kSPS y normalizado a $f_{clk} = 100$ MHz.	65
2.28. Implementación del LPF de $f_C=91.93$ kHz y $O_{LPF}=4$	66
2.29. Esquemático de la simulación del ADC basado en SAR en una FPGA . .	66
2.30. Comportamiento de V_{DAC} en el proceso de conversión del ADC	67
2.31. B obtenida a partir de V_{in}	68
2.32. Errores causados por mala selección de parámetros	68
2.33. f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 10-bit con $f_{Sampling}=1$ kSPS normalizado a $f_{CLK}=100$ MHz.	71
2.34. Comportamiento de V_{DAC} en el proceso de conversión del ADC	72
2.35. B obtenida a partir de V_{in}	72
3.1. Contador de 3-bit implementado para el PWM-DDR	75
3.2. Esquemático de la implementación de un contador de N-bit	75
3.3. Comparación de los V_{LPF} con $f_C=70$ kHz, $O_{LPF}=4$ y $D = \frac{1}{2}$ con un PWM con $f_{PWM}=100$ kHz y PWM-DDR con $f_{PWM}=200$ kHz	76
3.4. V_{LPF} de un PWM y PWM-DDR de 6-bit con los LPF de la Tabla 3.1 . .	77
3.5. f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=8$ kSPS con PWM-DDR normalizado a $f_{CLK}=100$ MHz. 78	
3.6. Esquemático de la implementación del en una FPGA con salida implementada con DDR	79
3.7. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit.	80
3.8. B obtenida a partir de V_{in} para un ADC de 8-bit	80
3.9. Placa de implementación del ADC de 8-bit	81
3.10. V_{DAC} de un ADC de 8-bit con V_{in} continuo	82
3.11. Esquemático de la implementación del ADC basado en SAR y DAC adicional para visualización	83
3.12. V_{in} y B obtenido por un ADC de 8-bit	83
3.13. Errores en B debido a escoger de manera errónea parámetros del LPF . .	84
3.14. f_C vs. $t_{Settling}$ para un ADC de 10-bit con $f_{Sampling}=8$ KSPS normalizado con $f_{CLK}=100$ MHz.	85
4.1. Esquemático ADC basado en SAR con DAC reemplazado con PWM y LPF	87
4.2. Esquemático de la división de d en ds de $\frac{N}{2}$ -bit	88
4.3. Esquemático del ADC basado en SAR con DAC reemplazado por 2 PWMs de $\frac{N}{2}$ -bit y 2 LPFs	89
4.4. f_C vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.	90
4.5. f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=8$ kSPS con 2 PWM-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.	91
4.6. Esquemático de la implementación del ADC con 2 PWM-DDR de $\frac{N}{2}$ -bit.	92
4.7. Circuito sumador-restador con AO para un ADC de 8-bit.	92

4.8. f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=340$ kSPS con 2 PWM-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.	94
4.9. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con $f_{Sampling}=340$ kSPS, implementado con 2 PWM-DDR de 4-bit	94
4.10. B a partir de V_{in} con un ADC de 8-bit con $f_{Sampling}=340$ kSPS implementado con 2 PWM-DDR de 4-bit	95
4.11. Esquemático de la división de d en K ds de $\frac{N}{K}$ -bit	95
4.12. Esquemático del ADC basado en SAR con DAC reemplazado por PWMs de $\frac{N}{K}$ -bit	97
4.13. Esquemático de la división de d en K ds de $\frac{N}{K}$ -bit con $\frac{N}{K}$ inexacto	97
4.14. Circuito sumador analógico con AO para un ADC de N -bit con el PWM dividido en K PWMs de $\frac{N}{K}$ -bit.	99
4.15. f_C vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.	100
4.16. f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 4 PWM-DDR de 2-bit normalizado a $f_{CLK}=100$ MHz.	101
4.17. Circuito sumador con AO para un ADC de 8-bit con el PWM-DDR dividido en 4 PWMs de 2-bit.	102
4.18. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con 4 PWMs de 2-bit	103
4.19. Circuito de la red R-2R para implementar DAC.	103
4.20. Esquemático del ADC basado en SAR con DAC implementado con red R-2R	104
4.21. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R	104
4.22. f_C vs. $f_{Sampling}$ con los parámetros hallados para un ADC de 8-bit con diferentes diseños, $O_{LPF}=6$ y normalizado con $f_{CLK}=100$ MHz	105
4.23. Esquemático de la implementación del ADC de 8-bit con 8 PWMs de 1-bit	106
4.24. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R	106
4.25. Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R	107
4.26. Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 2 PWM-DDR de 5-bit	109
4.27. Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 2 PWM-DDR de 4-bit y un PWM-DDR de 2-bit	110
4.28. Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 5 PWM-DDR de 2-bit	110
4.29. Esquemático de la implementación del ADC de 10-bit con el DAC implementado red R-2R	111
4.30. f_C vs. $f_{Sampling}$ con los parámetros hallados para un ADC 10-bit con diferentes diseños, $O_{LPF}=6$ y normalizado con $f_{CLK}=100$ MHz	111
5.1. PWM y PWM-DDC de 3-bit con D y D_{DDC} igual a $\frac{1}{2}$	115
5.2. PWM-DDC de 3-bit variando D_{DDC}	115
5.3. V_{LPF} con un PWM y PWM-DDC de 3-bit con D y D_{DDC} de $\frac{3}{8}$, con LPF de $O_{LPF}=3$ y $f_C=4$ MHz	116
5.4. ΔV_{LPF} con un PWM-DDC de 3-bit con d pares e impares, con LPF de $O_{LPF} = 3$ y $f_C = 4$ MHz	117

5.5.	V_{LPF} con un PWM-DDC 3-bit y $D = \frac{3}{8}$, con LPF de $O_{LPF}=3$ y $f_C=4$ MHz	117
5.6.	f_C vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.	121
5.7.	f_C vs. $f_{Sampling}$ con los parámetros hallados para un ADC de 6-bit con diferentes diseños con $O_{LPF}=4$, dividiendo el PWM y PWM-DDC en $K=2$ y $K=3$	122
5.8.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 6-bit con PWM-DDC normalizado a $f_{CLK}=100$ MHz.	123
5.9.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 6-bit con 2 PWM-DDC de 3-bit normalizado a $f_{CLK}=100$ MHz.	123
5.10.	f_C Vs. $f_{Sampling}$ para un ADC 8-bit con diferentes implementaciones del PWM	124
5.11.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=230$ kSPS con 2 PWM-DDC-DDR de 4-bit normalizado con $f_{CLK}=100$ MHz.	125
5.12.	Esquemático de la implementación del ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit	126
5.13.	V_{DAC} de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit para V_{in} continuo	127
5.14.	V_{in} y B obtenido por un ADC de 8-bit con PWM-DDC-DDR de 4-bit . .	127
5.15.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=500$ kSPS con 4 PWM-DDC-DDR de 2-bit normalizado con $f_{CLK}=100$ MHz	128
5.16.	Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.	128
5.17.	Esquemático de la implementación del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit	129
5.18.	V_{DAC} de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit para V_{in} continuo	130
5.19.	V_{in} y B obtenido por un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit .	130
6.1.	Comportamiento del contador implementado con registro DDR	134
6.2.	V_{LPF} para LPFs de arquitectura Bessel con $f_C= 1$ Hz para diferentes O_{LPF} .	135
6.3.	Arquitectura Sallen-Key para LPF de $O_{LPF}=2$	138
6.4.	Circuito sumador analógico con AO para un ADC de N-bit con el PWM dividido en K PWMs de $\frac{N}{K}$ -bit.	139
6.5.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con PWM-DDC-DDR de 8-bit normalizado a $f_{CLK}=100$ MHz.	141
6.6.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 2 PWM-DDC-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.	141
6.7.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 3 PWM-DDC-DDR, 2 PWM-DDC-DDR de 3-bit y uno de 2-bit normalizado a $f_{CLK}=100$ MHz.	142
6.8.	f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 4 PWM-DDC-DDR de 2-bit normalizado a $f_{CLK}=100$ MHz.	143
6.9.	Circuito sumador-restador con AO para un ADC de 8-bit con $K=3$	144
6.10.	Esquemático de la implementación del ADC con 2 PWM-DDR de 3-bit y uno de 2-bit.	144
6.11.	Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con $f_{Sampling}=800$ kSPS.	145

Abreviaciones

Término	Descripción
FPGA	Matriz de puertas lógicas programables (Field programmable gate array)
LUT	Lookup table
f_{clk}	Frecuencia de reloj de la FPGA
V_{OH}	Tensión de salida para 'high' o '1' en la FPGA
V_{OL}	Tensión de salida para 'low' o '0' en la FPGA
ΔV_O	Diferencia de tensión entre V_{OH} y V_{OL}
ADC	Convertor analógico digital
$f_{Sampling}$	Frecuencia de muestreo del convertor analógico digital
$t_{Conversion}$	Tiempo de conversión del ADC
V_{fs}	Tensión de fondo de escala
V_{th}	Tensión de umbral de la entrada de la FPGA
V_{NOT}	Tensión de salida de la puerta lógica NOT
T_{in}	Anchura del pulso de V_{NOT}
T_{inMax}	Anchura del pulso de V_{NOT} para V_{in} mínimo
V_{NOR}	Tensión de salida de la puerta lógica NOR
V_{INICIO}	Señal de activación del proceso de conversión analógica digital
V_C	Tensión del condensador
t_D	Tiempo de descarga del condensador
$V_{Sumador}$	Tensión del sumador del circuito One shot.
$f_{counter}$	Frecuencia del contador
SAR	Registro de aproximaciones sucesivas (Successive approximation register)
DAC	Convertor digital analógico
V_{DAC}	Tensión de salida del convertor digital analógico
ΔV_{DAC}	Rizado de la tensión de salida del convertor digital analógico
MSB	Bit más significativo
LSB	Bit menos significativo
V_{in}	Tensión de entrada del ADC
$V_{Comparador}$	Tensión de salida del comparador analógico
PWM	Modulador por ancho de pulso
V_{PWM}	Tensión de salida del PWM
D	Ciclo de trabajo del modulador por ancho de pulso
f_{PWM}	Frecuencia de modulación de ancho de pulso
T_{PWM}	Periodo del modulador de ancho de pulso
LPF	Filtro paso bajo
Q	Factor de calidad del filtro paso bajo

Término	Descripción
f_C	Frecuencia de corte del filtro paso bajo en Hertz
O_{LPF}	Orden del filtro paso bajo
V_{LPF}	Tensión de salida del filtro paso bajo
$t_{Settling}$	Tiempo de establecimiento del LPF
ΔV_{LPF}	Rizado en la tensión de salida del filtro paso bajo
$\overline{V_{LPF}}$	Tensión media de la salida del filtro paso bajo
AO	Amplificador operacional
AD	Absorción dieléctrica del condensador
A_{LPF}	Atenuación en escala logarítmica del LPF
B	Resultado de la conversión analógica digital
DDR	Double data rate
ODDR	Registro dedicado para salidas Double Data Rate
PWM-DDR	Modulador de ancho pulso con salida DDR
PWM-DDC	Modulador de ancho pulso con ciclo de trabajo distribuido (Distributed PWM cycle)
V_{DDC}	Tensión de salida del DPC
f_{DDC}	Frecuencia de modulación del PWM-DDC

I. Introducción

En el campo de los sistemas digitales de control, las señales son obtenidas a partir de sensores para ejecutar las acciones necesarias para obtener la salida deseada, a partir de una referencia indicada por el usuario. El controlador digital es usualmente implementado en un dispositivo lógico programable como microcontrolador, CPLD, DSP, FPGA etc. [1, 2], que se basa en el ADC para adquirir y convertir la señal necesaria [3]. Hay aplicaciones en las que la unidad de control necesita múltiples señales para obtener la salida deseada [4, 5].

El ADC usualmente se utiliza en el lazo de retroalimentación del proceso de control y es una de las etapas más críticas del proceso. La conversión tradicionalmente se realiza a través de un chip externo dedicado para cada señal del proceso. Los parámetros que describen los ADC son su frecuencia de muestreo ($f_{sampling}$) y resolución (N), además del rango de su tensión de entrada (V_{in}).

Hay un gran interés en los sistemas mayormente digitales, donde los ADCs son implementados en el dominio digital, simplificando su integración en los controladores (microprocesadores, DSPs, FPGAs etc.) para construir un SoC (system on a chip). El flujo de diseño de la FPGA permite la integración de sistemas embebidos de circuitos digitales, tales como filtros FIR que podrían necesitar ADCs externos [6]. Utilizar ADCs discretos como LTC1406 (ADC de 8-bit con 200 MSPS (muestras por segundo)) en aplicaciones que requieran un alto número de ADCs, puede ser complicado debido a la limitada cantidad de pines de entrada/salida de dispositivos de bajo costo. En este caso, una posible solución es evitar la necesidad de ADCs mostrada en [7], donde una señal digital es conectada a una entrada de la FPGA y su anchura es medida por contadores internos. Las FPGA permiten la implementación de ADCs, los cuales requieren una entrada conectada a un filtro pasivo RC y una serie de puertas lógicas que actúan como comparadores cuando la tensión de umbral (V_{th}) es alcanzada. Sin embargo, la implementación de estos ADCs en FPGA presentan limitaciones tales como, circuitos analógicos externos (integradores o filtros), rendimientos más bajos comparados con ADCs discretos y mayor consumo.

Tradicionalmente, los controladores digitales son implementados en microcontroladores o DSPs debido a su facilidad de desarrollo proporcionado por los lenguajes de programación

y compiladores. [8, 9]. Sin embargo, su capacidad de cómputo es limitada por una ejecución secuencial de las instrucciones. La matriz de puertas lógicas programables (FPGA por su acrónimo en inglés de 'field-programmable gate array') son dispositivos lógicos programables, que permiten la implementación de circuitos digitales utilizando sus recursos de hardware. El circuito es descrito en un lenguaje de descripción de circuitos digitales (VHDL o Verilog) [10], y es sintetizado e implementado en el componente a través de herramientas de diseño. Las FPGAs están compuestas de un gran número de elementos lógicos tales como, lookup tables (LUT), flip-flops, bloques de entrada y salidas entre otros [11], que son conectados a través de enrutamientos configurables. Los recursos de hardware permiten la implementación de circuitos paralelos para múltiples controladores que aceleran el cómputo en las FPGAs comparado con microcontroladores o DSPs.

Aplicaciones como SDR (Radio definida por software) necesita ADCs con $f_{Sampling}$ altas en la etapa de IF (frecuencia intermedia) para habilitar el procesamiento digital. Los SDRs más avanzados son capaces de remover la etapa de IF, reduciendo los amplificadores operacionales, filtros y otros elementos analógicos, utilizando ADCs con altas prestaciones (GSPS). Los ADCs de altas prestaciones usualmente son implementados en circuitos integrados a medida y utilizan técnicas de muestreo intercalado que requieren varios ADCs en paralelo para convertir la misma señal [12–15].

Los ADCs usualmente no son integrados en FPGAs de bajo costo. Algunas FPGAs de altas prestaciones pueden integrar ADCs, que pueden ser insuficientes para múltiples controladores convirtiendo simultáneamente varias señales a diferentes $f_{Sampling}$, resolución y rangos de tensión. Por ejemplo, en los On-Board chargers (OBC) para coches eléctricos, se necesita leer una gran cantidad de señales para acciones de control o para protección y seguridad del OBC [16–18]. En [19, 20], se necesita obtener múltiples señales para controlar simultáneamente diferentes convertidores en una sola DSP, la cual tiene un procesador implementado por una FPGA. En aplicaciones biomédicas, se requiere adquirir múltiples señales simultáneamente [21], mientras que en sistemas de gestión de baterías, diferentes señales son utilizadas al mismo tiempo para monitorizar y para acciones de protección y seguridad [22]. En dichos casos, se utilizan ADCs externos que incrementan el coste de implementación.

La tesis propone una metodología para implementar un ADC de N-bit escalable y

parametrizable en una FPGA. La escalabilidad permite la implementación de varios ADCs en una sola FPGA, cada uno configurado individualmente de acuerdo a los parámetros del ADC (N , $f_{Sampling}$). La primera arquitectura del ADC esta basada en un conversor time-to-digital a partir del circuito one shot, el cual es construido a partir de un circuito RC y puertas lógicas. El contador ascendente y la lógica del one shot puede ser implementada usando LUTs y flip-flops disponibles en la mayoría de las familias de FPGAs. La segunda arquitectura esta basada en que la lógica SAR controla un DAC (conversor digital analógico), el cual es implementado utilizando un PWM y un LPF externo. La lógica SAR y el PWM son implementados usando LUTs y flip-flops de la FPGA. El circuito no depende de hardware especial para componentes en particular o entradas externas como el space vector PWM [23], sinusoidal PWM [24] o el PWM de cambio de fase [25] usado en otras aplicaciones, que limitarían la portabilidad a diferentes FPGAs. Se presentan diferentes optimizaciones para el PWM con el objetivo de incrementar $f_{Sampling}$.

I.I. Antecedentes

Los ADCs de tipo Flash están basados en un conjunto de comparadores analógicos y un codificador con prioridad. Cada comparador genera una respuesta digital de comparar V_{in} con una tensión de referencia escalada. A pesar de que genera altas $f_{Sampling}$, tienen como desventaja el consumo y la linealidad. Los ADCs estocásticos mejoran la linealización de la conversión, incrementando el número de comparadores y añadiendo un circuito de interpolación en cada canal digital. Son circuitos con señales mixtas que usualmente son implementados en circuitos integrados customizados CMOS [26] (8190 comparadores, 6.2 ENOB, 100 MSPS), [27] (2040 comparadores, 5.2 ENOB, 320 MSPS), [28] (2047 comparadores, 5.7 ENOB, 210 MSPS). Sin embargo, los ADCs tipo flash no son convenientes para las FPGAs debido a su gran número de comparadores externos y entradas utilizadas. Los ADCs de tipo VCO (oscilador controlado por tensión) pueden alcanzar altas $f_{Sampling}$. Están compuestos de un oscilador, un cuantificador y un módulo diferencial. En [29, 30] se presenta un ADC basado en VCO, el cual es sintetizable y descrito utilizando HDL, implementado en celdas CMOS de 65 nm. Una arquitectura para eliminar la no linealidad de los núcleos y el cuantificador reemplazándolos con un PWM y filtros es presentada en [31]. Los ADCs tipo VCO descritos son implementados

en tecnologías CMOS y requieren un gran número de entradas y métodos para mejora la linealidad. El ADC tipo sigma-delta presentado en [32] esta implementado con un circuito analógico simple en una FPGA virtex-4, utilizando las entradas diferenciales y requiere circuitos digitales complejos. Un ADC tipo sigma-delta con alta precisión y baja $f_{Sampling}$ es implementado en [33] con una FPGA A3PE3000, es usado para balancear un giroscopio. Usando la entrada diferencial de baja tensión de la FPGA y un integrador, [34] implementa un ADC tipo sigma-delta para un sensado de corriente.

Muchos de los ADCs con altas $f_{Sampling}$ están basados en arquitecturas de conversores time-to-digital (TDC) [35–37]. Se basan en la medición de la carga de un circuito RC hasta llegar a V_{in} . Un comparador genera un pulso digital proporcional a V_{in} y el circuito TDC mide la anchura del pulso. Aunque el circuito TDC pueda ser implementado fácilmente con un contador digital, se obtiene altas $f_{Sampling}$ cuando se utiliza redes de retardo. Las redes de retardo están compuestas por un gran número de secciones de la FPGA, dedicadas a transmitir la entrada a la salida con un retardo conocido, ya sea por estimación o medido. La precisión del ADC basado en TDC es afectado por las tolerancias en R y C, además de las fluctuaciones de temperatura. Es por esto, que requieren calibración periódica.

El ADC de 7-bit con 600 MSPS presentado en [38] es un ADC implementado en una FPGA UltraScale+ de Xilinx, con un circuito de calibración periódica. El ADC implementa una red de retardos, un circuito de detección de flancos, un codificador y un circuito de calibración. En este caso en particular, el retardo es medido a través de la red compuesta por 426 multiplexores XOR de la FPGA. El ADC no requiere componentes externos ya que el buffer de salida (OBUF) y el buffer de entrada diferencial (DIFFINBUF) de la UltraScale+ reemplaza el capacitor, la resistencia y el comparador analógico. Esta FPGA permite la configuración del slew rate y la impedancia de salida deseada en el OBUF, que es conectado externamente al DIFFINBUF, generando una capacitancia parásita. El DIFFINBUF es usado como un comparador analógico, conectando una entrada al OBUF y la otra a V_{in} , generando una salida digital cuando la entrada diferencial exceda la tensión de umbral. Sin embargo, la tolerancia en el proceso de fabricación de los CMOS puede afectar significativamente los valores eléctricos, como la capacitancia parásita, la tensión de umbral del DIFFINBUF y el transistor de impedancia del OBUF.

El ADC de 6-bit de 200 MSPS presentado en [35] es más simple. Es implementado con

una red de retardo, pero basado solo en multiplexores de la FPGA Spartan-6. Esta implementación requiere una resistencia externa para cargar la capacitancia parásita del DIFFINBUF. Otro ejemplo es el ADC de 6-bit presentado en [36], el cual debe ser mantenido en temperaturas criogénicas con helio líquido a 4^o K. También utiliza DIFFINBUF conectado a una resistencia externa, aunque utiliza ISERDES (entrada serial a paralelo) de la FPGA Artix-7 para implementar la red de retardos.

Utilizar hardware específico de cada FPGA para implementar el ADC hace que la portabilidad a otras familias de FPGA sea difícil. El hardware que implementa la red de retardos no puede ser enrutado por herramientas automáticas, por lo que deben ser manualmente configuradas en la FPGA para obtener los retardos deseados. Sin embargo, la diferencia en los parámetros eléctricos entre las familias de FPGA, requieren rediseñar el codificador y la red de retardos. Finalmente, estos ADCs están diseñados y probados para ADCs con parámetros específicos, por lo que no puede ser parametrizados para otras resoluciones o $f_{Sampling}$.

Los ADCs basados en SAR no pueden alcanzar altas $f_{Sampling}$ como los basados en TDC, aunque tienen algunas ventajas, como la portabilidad hacia otras familias de FPGA y su adaptabilidad a diferentes requisitos del ADC. El ADC basado en SAR esta compuesto por la lógica SAR, un DAC y un comparador analógico. El PWM con filtro es la forma más económica de implementar un DAC como es descrito en [39]. El ADC basado en SAR presentado en [40] es implementado con un DAC [41, 42] compuesto de un PWM Dyadic y un filtro RC pasivo, el cual es sintetizable en standard cells de tecnología CMOS de 40 nm. La R del filtro es implementada en una capa resistiva de poli-silicona y C es implementada utilizando metal-aislamiento-metal. El rendimiento del ADC es muy limitado, generando $f_{Sampling} = 2.8$ kSPS y 7-bit de resolución. Comparado con otros ADCs que son parcial o completamente sintetizables en standard cells, como la arquitectura flash, sigma-delta o VCO, su mayor ventaja es la reducción del área de implementación y su bajo consumo.

Un reciente ADC basado en SAR es presentado en [43], mejorando los anteriores ADCs, alcanzando $f_{Sampling} = 10$ MSPS y 7.5 ENOB. El ADC es sintetizado en standard cells de 65 nm usando tecnología CMOS. Esta basado en un DAC resistivo, LUT de compensación y un comparador OAI (lógica de inversión AND-OR). El comparador OAI reemplaza la lógica simple usada como comparador analógico, permitiendo a la puerta lógica OR que

restablezca el nodo del transistor de la puerta lógica AND, con el objetivo de eliminar las cargas residuales que afectan la linealidad del ADC. El DAC resistivo permite eliminar el LPF, aumentando $f_{Sampling}$, pero incrementa el número de entradas necesarias en la FPGA, además requiere un circuito de compensación debido a la resistencia no lineal de los transistores del inversor. Implementar un DAC interno en la FPGA no es posible, debido a que no permiten el uso de resistencias internas, excepto aquellas salidas con pull-up/pull-down. Los valores de las resistencias varían mucho entre familias de FPGA por lo que su portabilidad sería complicada.

I.II. Objetivos

I.II.1. Objetivo general

El objetivo principal de esta tesis doctoral es generar una metodología sistemática para el diseño e implementación de un convertor analógico digital de N-bit escalable y parametrizable en una FPGA a partir de los requerimientos del diseño y las posibilidades de la FPGA.

I.II.2. Objetivos específicos

- Debido a las diferentes problemáticas ambientales se ha hecho necesario la implementación de sistemas eléctricos capaces de reducir de manera sustancial la emisión de gases contaminantes, es por esto se requiere optimizar el consumo energético y su gestión a través de la implementación de herramientas optimizadas, permitiendo la disminución de coste y consumo.
- La adquisición de datos es uno de los procesos más importantes en el control. Aprovechando las ventajas de la FPGA, se busca diseñar e implementar un convertor análogo digital que requiera el uso de la menor cantidad de entradas/salidas digitales y de elementos electrónicos, pero con prestaciones suficientes para la aplicación.
- Cada día son implementados controles más sofisticados, que incluyen diferentes variables al proceso. A medida que el número de variables aumenta, es necesario

la inclusión de más ADC, incrementando el coste de implementación. Por lo tanto, es necesario implementar un convertor analógico digital de bajo costo escalable y parametrizable que se ajuste a los parámetros de la aplicación y que permita su extrapolación a diferentes familias de FPGA.

I.III. Contribución de esta tesis

Esta tesis fue realizada con el objetivo de diseñar e implementar un ADC de N-bit escalable y parametrizable para una FPGA. Los principales aportes de esta tesis doctoral son:

- El estudio y análisis de un ADC basado en el circuito one shot, donde se muestran las ventajas y desventajas del mismo. Además se presenta una metodología para diseñar e implementar el ADC a partir de la resolución, rango de V_{in} , $f_{Sampling}$ deseada y los parámetros de la FPGA.
- El estudio y análisis del ADC basado en SAR, donde se reemplaza el convertor digital analógico (DAC) por un modulador de ancho de pulso (PWM) y un filtro paso bajo (LPF). Se presenta una metodología para la obtención de los parámetros del LPF a partir de la resolución, $f_{Sampling}$ deseada y los parámetros de la FPGA.
- Se presenta una optimización en la implementación del PWM, donde se utilizan las salidas de double data rate (DDR) para aumentar la frecuencia del PWM (f_{PWM}), de manera que se mejoren las prestaciones del ADC. Se modifica la metodología para obtener los parámetros del LPF a partir de la resolución, $f_{Sampling}$ deseada y los parámetros de la FPGA.
- Se presenta una optimización en la implementación del PWM, donde se busca paralelizar el PWM con el objetivo de aumentar la frecuencia del PWM (f_{PWM}). Para esto se divide el PWM en K PWMs de $\frac{N}{K}$ -bit, cada PWM resultante con su respectivo LPF. Se modifica la metodología de obtención de los parámetros del LPF a partir de la resolución, $f_{Sampling}$ deseada y los parámetros de la FPGA.
- Se presenta una optimización en la implementación del PWM, donde se implementa el PWM de ciclo de trabajo distribuido (PWM-DDC) con el objetivo de disminuir el rizado de los LPFs y de esta manera mejorar las prestaciones del ADC. Se modifica

la metodología de obtención de los parámetros del LPF a partir de la resolución, $f_{Sampling}$ deseada y los parámetros de la FPGA.

- Se presenta un método generalizado para el diseño e implementación del ADC de N-bit basado en SAR, reemplazando el DAC con el conjunto de PWM y LPFs a partir de la resolución, $f_{Sampling}$ deseada y los parámetros de la FPGA. Se prioriza el coste de implementación y la mínima cantidad de salidas de la FPGA.

I.IV. Organización del escrito

El escrito está compuesto de 6 capítulos, los cuales están organizados de la siguiente manera:

En el capítulo 1, el estudio y análisis de un ADC a partir del circuito one shot en una FPGA.

En el capítulo 2, se presenta el análisis de un ADC basado en la arquitectura SAR, donde se reemplaza el DAC con un PWM y un LPF.

En el capítulo 3 se presenta la implementación del PWM con el double data rate (DDR) con el fin de aumentar f_{PWM} y mejorar las prestaciones del ADC.

En el capítulo 4 se presenta una optimización en la implementación del PWM, donde se divide el PWM de N-bit en K PWMs de $\frac{N}{K}$ -bit, con el objetivo de aumentar f_{PWM} y mejorar las prestaciones del ADC.

En el capítulo 5 se implementa el PWM de ciclo de trabajo distribuido (PWM-DDC) con el objetivo de aumentar f_{PWM} y mejorar las prestaciones del ADC.

En el capítulo 6 se presenta una metodología sistemática para encontrar los parámetros del LPF, para la implementación de un ADC de N-bit basado en SAR, priorizando los costes de implementación y la menor cantidad de salidas de la FPGA.

En el capítulo 7 se presentan las conclusiones.

1. Capítulo 1

1.1. Resumen

En este capítulo se estudia la implementación de un convertor analógico digital (ADC) a partir de una conversión de tiempo a salida digital (TDC). Esta estrategia implementa el ADC a partir de puertas lógicas, resistencias y condensadores, ofreciendo versatilidad en el diseño en diferentes parámetros como el rango de tensión de entrada, frecuencia de muestreo ($f_{Sampling}$), resolución etc. Será implementado en una FPGA y se analizarán las ventajas y desventajas del diseño.

1.2. One shot ADC

Una de las estrategias de implementación de un convertor analógico digital a través de un circuito TDC es el one shot ADC, presentada en [44]. El objetivo de esta técnica es aprovechar el proceso de carga y descarga de un condensador a través de un circuito RC y su combinación con puertas lógicas. El esquemático se puede apreciar en la Fig. 1.1.

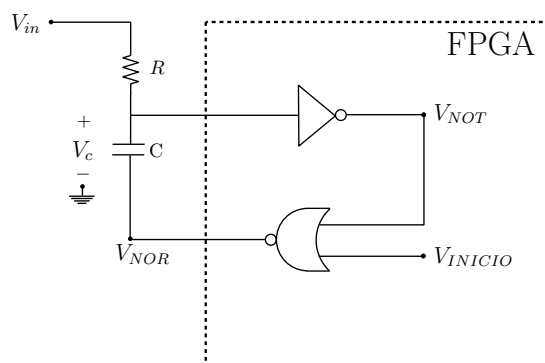


Figura 1.1: Circuito one shot

Al activar la señal de inicio, se obtiene un pulso digital en la puerta NOT cuya anchura (T_{in}) depende de la tensión analógica de entrada (V_{in}) y de los valores R y C . A partir de la medida de T_{in} se obtiene el resultado de la conversión (B). Este circuito será implementado en una FPGA, aprovechando su versatilidad y su alta velocidad de procesamiento [45, 46].

Un requisito para que el sistema funcione es que V_{in} sea mayor a la tensión de umbral (V_{th})

de la puerta lógica NOT [47]. Cuando esto se cumple, T_{in} se genera de la siguiente manera. En estado estacionario la señal de activación (V_{INICIO}) está en '0' (V_{OL}) por lo que la salida de la puerta lógica NOR (V_{NOR}) es '1' (V_{OH}). En este momento por el condensador no pasa corriente debido a que está completamente cargado. En estado estacionario V_{in} genera a la salida de la puerta lógica NOT (V_{NOT}) un '0'.

Cuando V_{INICIO} es '1', V_{NOR} es '0'. En este justo momento la tensión del condensador (V_C) desciende hasta $V_{in} - \Delta V_O$, donde $\Delta V_O = V_{OH} - V_{OL}$. Esto hace que la salida de la puerta NOT (V_{NOT}) cambie a '1', iniciando el T_{in} .

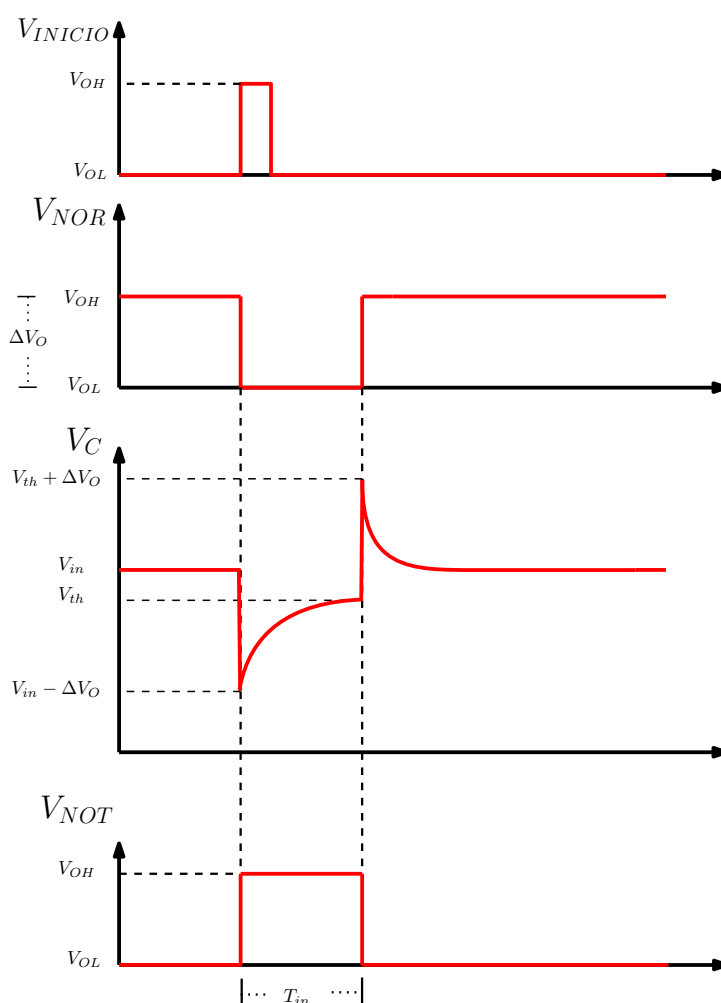


Figura 1.2: Funcionamiento gráfico del one shot ADC

Por último, el condensador empieza su proceso de carga, hasta llegar a V_{th} . Es aquí donde V_{NOT} vuelve a '0' finalizando T_{in} , provocando que V_{NOR} cambie a '1'. V_C cambia abruptamente a $V_{th} + \Delta V_O$, comenzando su proceso de descarga hasta llegar a V_{in} . Se

muestran las señales V_{NOT} , V_{NOR} , V_C y V_{INICIO} en el proceso de conversión en la Fig. 1.2. En la Eq. 1.1 se muestra como se rige T_{in} a partir de ΔV_O , V_{th} , V_{in} , R y C .

$$T_{in} = RC \cdot \ln\left(\frac{\Delta V_O}{V_{in} - V_{th}}\right) \quad (1.1)$$

La Eq. 1.1 muestra que T_{in} es inversamente proporcional a $V_{in} - V_{th}$. Cuando V_{in} es aproximadamente V_{th} , V_C desciende hasta $V_{th} - \Delta V_O$ por lo que toma más tiempo al condensador cargarse hasta V_{th} [48], conllevando a un valor mayor de T_{in} . Ahora bien, en el caso opuesto si V_{in} es similar a ΔV_O , la tensión a la que V_C desciende es más cercana a V_{th} por lo que es fácilmente recuperable, haciendo que T_{in} sea menor.

Para evitar los valores negativos o iguales a 0 en el denominador de la Eq. 1.1 se añade a V_{in} una tensión sumadora ($V_{Sumador}$), mediante un circuito sumador de tensión implementado con amplificador operacional, obteniendo así la Eq. 1.2. Para hallar $V_{Sumador}$ se debe cumplir para todos los valores de V_{in} la condición mostrada en la Eq. 1.3a, debido a que es necesario que V_C sea menor a V_{th} para generar T_{in} . Si el rango de V_{in} va desde 0 V hasta el fondo de escala (V_{fs}), V_{fs} se ve limitado por $V_{Sumador}$ como se muestra en la condición de la Eq. 1.3b.

$$T_{in} = R \cdot C \cdot \ln\left(\frac{\Delta V_O}{V_{in} + V_{Sumador} - V_{th}}\right) \quad (1.2)$$

$$V_{in} + V_{Sumador} - \Delta V_O < V_{th} \quad (1.3a)$$

$$V_{fs} < V_{th} + \Delta V_O - V_{Sumador} \quad (1.3b)$$

1.3. Simulación del ADC basado en el circuito one shot en una FPGA

A continuación, se realiza la simulación del ADC en la FPGA Zynq 7000 de Xilinx. Esta FPGA tiene incorporado un ADC de 12 bits de 2 canales y frecuencia de muestreo ($f_{Sampling}$) de 1 MSPS (10^6 muestras por segundo) con un rango entre 0 V y 1 V. Teniendo

en cuenta lo anterior se propone diseñar un ADC basado en el circuito one shot que alcance la $f_{Sampling}$ del ADC incluido en la FPGA.

Considerando dicha información, se obtiene el R, C y T_{in} máximo (T_{inMax}) a partir de la Eq. 1.2. El T_{inMax} es generado cuando V_{in} es igual a la tensión mínima de entrada del ADC. Se decidió implementar el circuito digital en la propia FPGA aprovechando su versatilidad y su capacidad de procesamiento, además de utilizar la menor cantidad de elementos externos.

Los valores de la FPGA son $V_{th}=1.3$ V, $\Delta V_O=3.3$ V, mientras que $V_{Sumador}=1.3$ V y $f_{Sampling}=1$ MSPS por lo que $T_{inMax}=1$ μ s ($T_{inMax} = \frac{1}{f_{Sampling}}$). Con estos valores se obtiene $R=344$ Ω y $C=500$ pF.

Para comprobar el funcionamiento del sistema, se muestra el resultado de la simulación del circuito con la herramienta Simscape-Simulink de Matlab, donde se ofrece una librería que permite incluir tanto los componentes del circuito como los elementos configurados en la FPGA. Las puertas lógicas NOR y NOT se implementan mediante LUT (Lookup table). Las LUT permiten implementar funciones combinatoriales, donde se determina la respuesta en función del valor de las entradas. Esta es la manera en que se sintetizan las lógicas combinatoriales en la FPGA [49, 50].

Una vez el condensador vuelve a su punto inicial V_{in} , el sistema está a la espera de ser iniciado nuevamente y el proceso puede ser repetido periódicamente. Este proceso se puede apreciar en la Fig. 1.3, donde se realizó la simulación para un $V_{in}=0$ V y los valores anteriormente mencionados.

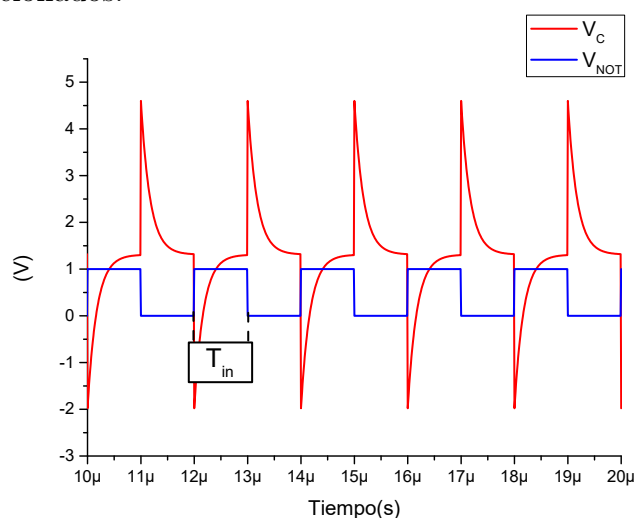


Figura 1.3: Comportamiento de V_C y V_{NOT} con $V_{in}=0$ V

Para observar el comportamiento del ADC basado en el circuito one shot, se muestra el resultado de la simulación de dos valores diferentes de V_{in} , de manera que se aprecie como se comporta V_C en cada caso. En la Fig. 1.4 se muestra el resultado de la simulación cuando $V_{in}=0.5\text{ V}$ y $V_{in}=0\text{ V}$. Es posible ver que la caída de tensión en V_C es mayor al disminuir V_{in} . Dicha diferencia en V_C se ve reflejada en T_{in} , ya que a mayor V_{in} menor T_{in} , visualizando una relación inversamente proporcional.

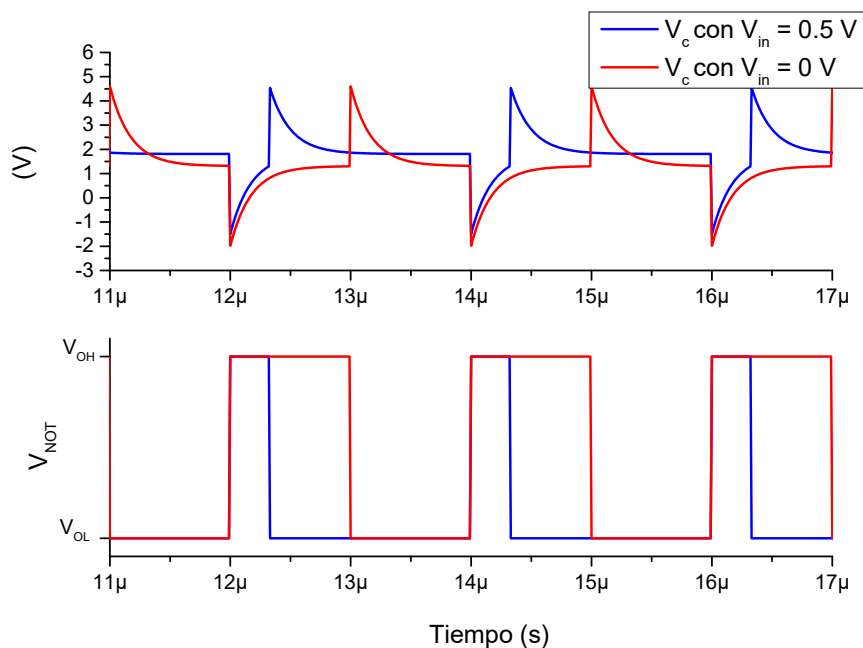


Figura 1.4: Comparación del comportamiento de V_C y V_{NOT} con diferentes valores de V_{in}

1.3.1. Análisis de la linealidad de la respuesta del ADC basado en el circuito one shot

Una vez mostrada la simulación del comportamiento del ADC basado en el circuito one shot con dos valores diferentes de V_{in} , se realiza una simulación con todos los posibles valores de V_{in} a través de un barrido de tensión.

Para comparar el one shot ADC con el ADC incorporado en la FPGA Zynq-7000, se utiliza el mismo rango de V_{in} , en este caso está entre 0 V y 1 V . Una vez delimitado el rango del ADC, se genera un barrido de V_{in} y se obtiene T_{in} a partir de la Eq. 1.2, generando así la Fig. 1.5. Se observa una clara no linealidad en los resultados obtenidos.

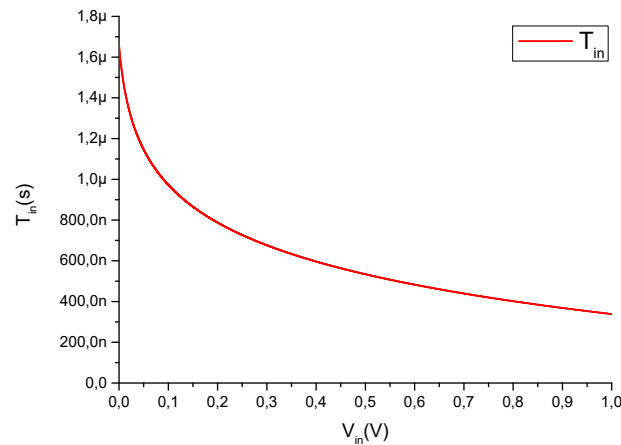


Figura 1.5: T_{in} a partir de un barrido de V_{in}

El proceso de cómputo de T_{in} se desarrolla a continuación. Una vez generado V_{NOT} , se obtiene el tiempo de duración del pulso. Para esto se implementó un contador ascendente que indique el número de cuentas que V_{NOT} estuvo activo. Con este número de cuentas y la frecuencia de reloj (f_{clk}) de la FPGA es posible obtener un valor aproximado de T_{in} .

Una vez obtenido el número de cuentas que V_{NOT} estuvo activo, la FPGA debe generar B a partir de la Eq. 1.2. Si se quiere obtener B directamente desde la Eq. 1.2, es necesario que la FPGA implemente un circuito aritmético que contiene división, cálculo exponencial, suma y resta, por lo que es complejo y tomaría varios ciclos de reloj realizar dicha operación.

Para implementarlo de forma sencilla se genera una tabla con todos los posibles valores de B a partir de las cuentas posibles ($0 : 2^N - 1$). El cálculo de esta tabla la realiza el sintetizador, calculando B a partir de la Eq. 1.4. Una vez sintetizada dicha tabla, se accede a ella según con el valor de T_{in} y se obtiene el B correspondiente.

$$V_{in} = \frac{\Delta V_O}{e^{\left(\frac{T_{in}}{RC}\right)}} + V_{th} - V_{Sumador} \quad (1.4)$$

$$T_{in} = \frac{Cuentas}{f_{clk}}$$

Para comprobar el proceso de obtención teórico de B correspondiente a V_{in} a partir de T_{in} , se realiza una simulación en Matlab del procedimiento explicado anteriormente. En

primer lugar, se generan todos los posibles valores de V_{in} . A partir de ellos se obtiene el T_{in} correspondiente con la Eq. 1.2. Este proceso se realizará para un ADC de 8-bit como primera aproximación, para ser comparado con el ADC de la FPGA Zynq 7000. En primer lugar, se obtiene LSB a partir de la Eq. 1.5. LSB es la tensión necesaria en V_{in} para generar un cambio en B .

$$LSB = \frac{\Delta V_O}{2^N} \quad (1.5)$$

La Fig. 1.6 muestra la relación de B con V_{in} , donde se puede apreciar que B se ve afectada por la no linealidad existente entre V_{in} y T_{in} . Para tensiones de V_{in} cercanas a $0 V$, B incrementa en 1 bit por cada LSB . Para tensiones cercanas a V_{fs} los cambios en B son mayores a 1 para cada LSB . Cuando hay B que no son representados, se considera que el ADC presenta *missing codes*.

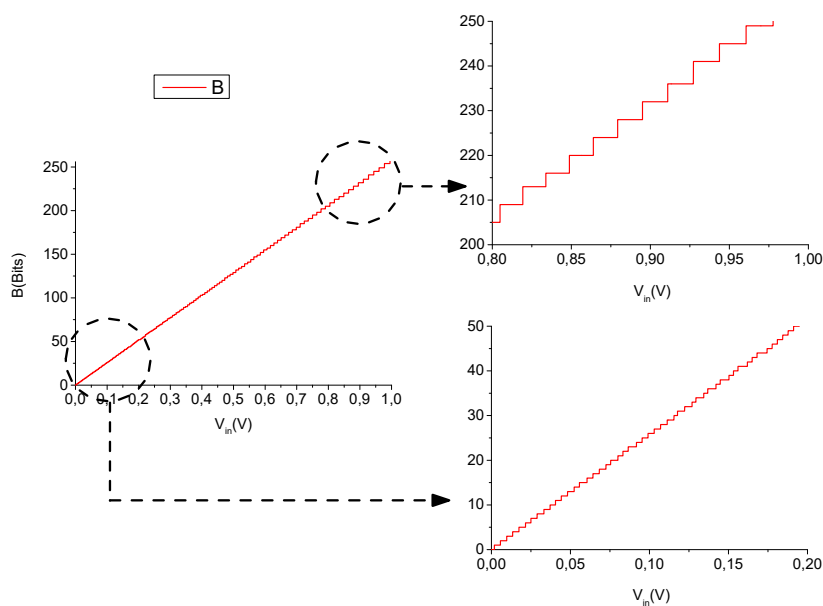


Figura 1.6: B obtenida a partir de barrido de V_{in}

Para evitar la no linealidad del one shot ADC se decide trabajar en la parte más lineal de la curva de carga de V_C . Para esto se incrementa la constante de tiempo del condensador, permitiendo que durante T_{in} , V_C tenga un comportamiento aproximadamente lineal. Esta modificación permitiría que el error por la no linealidad sea menor a 1 LSB . La diferencia se puede ver en la Fig. 1.7, donde el T_{in} es el mismo en los dos casos, pero se trabaja en una zona diferente de la curva de carga de V_C . Los valores utilizados para obtener la Fig. 1.7 son mostrados en la Tabla 1.1.

	Valores originales	Valores modificados
T_{inMax}	1 μs	1 μs
$V_{Sumador}$	1,35 V	3,5 V
ΔV_O	3,3 V	3,3 V
V_{th}	1,3 V	1,3 V
R	344 Ω	4,93 k Ω
C	500 pF	500 pF

Tabla 1.1: Características del circuito one shot para $T_{in} = 1 \mu s$ con comportamiento exponencial y aproximadamente lineal

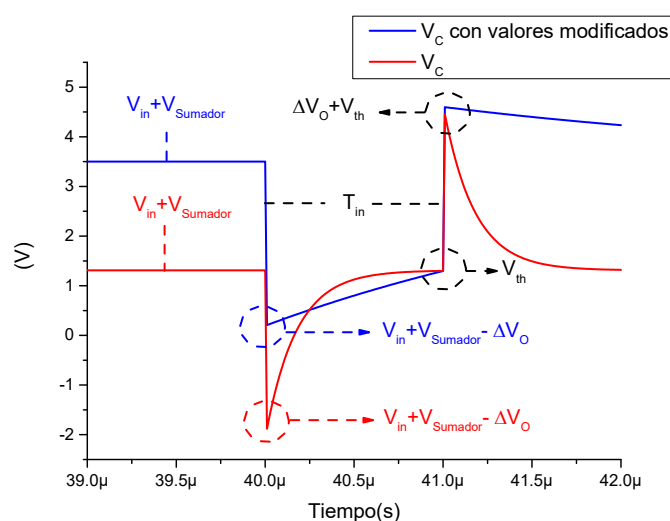


Figura 1.7: Comparación de la curva de carga de V_C de comportamiento exponencial y comportamiento aproximadamente lineal

Para lograr que la curva de carga de V_C tenga un comportamiento aproximadamente lineal, se debe modificar la constante de tiempo del condensador y/o el $V_{Sumador}$. Se empieza modificando $V_{Sumador}$ y a partir de este se modifica la constante de tiempo del condensador, siempre en base al T_{in} requerido.

También se debe tener en cuenta varios aspectos para obtener los valores. En primer lugar, para modificar el $V_{Sumador}$ se debe considerar el rango de V_{in} . En el caso propuesto se trabaja con un rango entre 0 V y 1 V. Ahora bien, una vez iniciado el proceso de conversión, V_C desciende hasta $V_{in} + V_{Sumador} - \Delta V_O$. Para todo el rango de V_{in} , $V_{in} + V_{Sumador} - \Delta V_O$ debe ser mayor a V_{th} , siendo el caso crítico cuando $V_{in} = V_{fs}$.

Para escoger el $V_{Sumador}$ se deben seguir ciertos criterios. $V_{Sumador}$ debe ser lo más elevado

posible para que así, V_C tenga un comportamiento aproximadamente lineal. Otro criterio explicado anteriormente, es el mostrado en la Eq. 1.3a, que indica que $V_{Sumador}$ debe ser mayor a V_{th} . Por lo tanto, el rango de V_{in} del ADC es otro factor a tener en cuenta en la selección del $V_{Sumador}$.

Se hallan los valores $V_{Sumador} = 3,5 V$, $R = 4,93 k\Omega$ y $C = 500 pF$ para un $T_{inMax} = 1 \mu s$. A partir de un barrido de V_{in} se obtiene la Fig. 1.8, donde se puede apreciar un comportamiento más lineal. Además, se observa en la Fig. 1.9 que la respuesta es equidistante en casi la totalidad los puntos de la gráfica, en comparación con la Fig. 1.6.

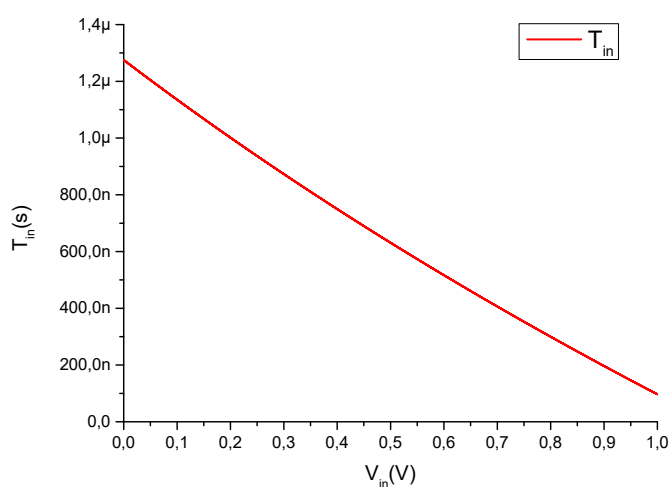


Figura 1.8: T_{in} a partir de barrido de V_{in}

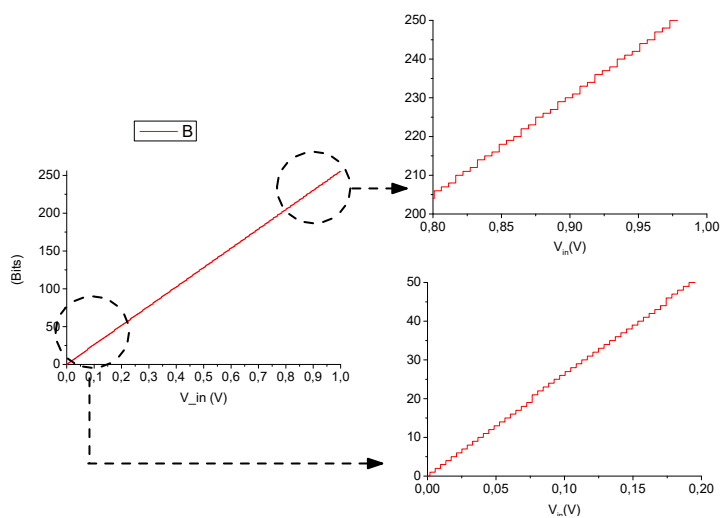


Figura 1.9: B obtenida a partir de barrido de V_{in}

El anterior análisis teórico fue realizado a partir de valores matemáticos ideales de

los diferentes componentes. Ahora se realiza con la herramienta Simulink de Matlab, la simulación de la implementación en una FPGA del ADC con los valores obtenidos anteriormente. Los resultados de la simulación se presentan en la Fig. 1.10, donde se observa un problema con la estrategia lineal. Una vez terminado el pulso de anchura T_{in} , V_C aumenta hasta $\Delta V_O + V_{th}$ y es aquí cuando empieza su proceso de descarga hasta llegar a V_{in} , su estado estacionario. El resultado esperado para $V_{in} = 0$ V era $T_{in} = 1$ μ s.

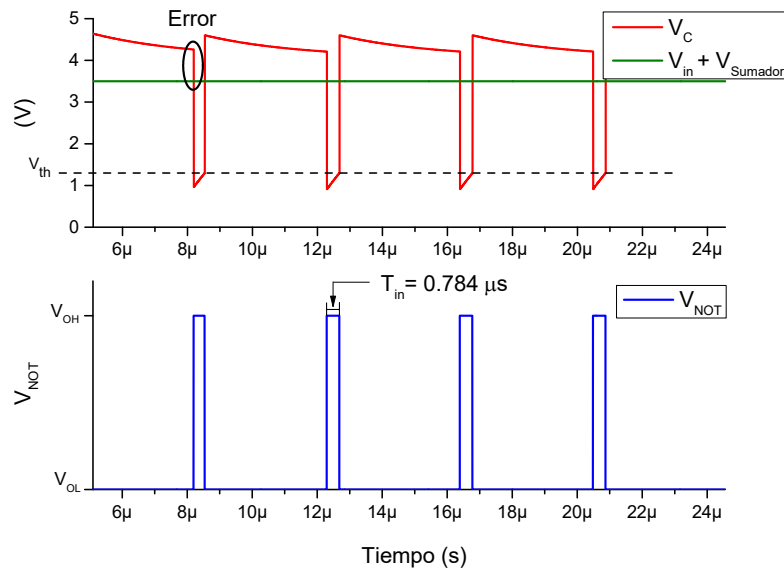


Figura 1.10: V_C y V_{NOT} con error en descarga del condensador para $V_{in} = 0$ V y $V_{Sumador} = 3.5$ V

En la Fig. 1.10 V_C no alcanzó a descargarse por completo antes de iniciar la siguiente conversión. Esto es debido a que se aumentó la constante de tiempo RC con el objetivo de tener un comportamiento más lineal entre T_{in} y V_{in} . La siguiente conversión no sería de V_{in} sino de $V_{in} + Error$, lo que conllevaría a T_{in} erróneos.

Para obtener una conversión sin el error presentado anteriormente, en el caso de un ADC de 8-bit, V_C debe encontrarse al 99.6% ($100 - \frac{100}{2^N}$) de descarga, de manera V_C sea menor a 1 LSB . Partiendo de la ecuación de descarga del condensador mostrada en la Eq. 1.6, se obtiene el tiempo de descarga (t_D) necesario para que la medida sea correcta. Se presenta la Eq. 1.7 en función de N , V_{th} y ΔV_O . Para el caso de un ADC de 8-bit, el t_D debe ser mínimo $7,0712 \cdot R \cdot C$ para que V_C sea menor a 1 LSB .

$$V_C(t) = (V_{th} + \Delta V_O)e^{-\frac{t}{RC}} \quad (1.6)$$

$$t_D = -\ln\left(\frac{\frac{1}{2^N}}{V_{th} + \Delta V_O}\right) \cdot RC \quad (1.7)$$

Para obtener la $f_{Sampling}$ del ADC a partir de T_{inMax} y t_D se utiliza la Eq. 1.8a. También es posible obtener $f_{Sampling}$ a partir de los parámetros del ADC, $V_{Sumador}$, R , C y V_{in} , además de los parámetros de la FPGA, V_{th} y ΔV_O siguiendo la Eq. 1.8b. En el caso de un ADC de 8-bit con un $T_{inMax}=1 \mu s$, con $R=4.93 \text{ k}\Omega$, $C=500 \text{ pF}$ y $V_{Sumador}=3.5 \text{ V}$, el tiempo de descarga t_D de $17.43 \mu s$, se obtendría una $f_{Sampling}=54.259 \text{ kSPS}$. En la Fig. 1.11 se tiene en cuenta el t_D hallado.

$$f_{Sampling} = \frac{1}{T_{inMax} + t_D} \quad (1.8a)$$

$$f_{Sampling} = \frac{1}{RC} \frac{1}{\ln\left(\frac{\Delta V_O (V_{th} + \Delta V_O)}{\frac{1}{2^N} (V_{in} + V_{Sumador} - V_{th})}\right)} \quad (1.8b)$$

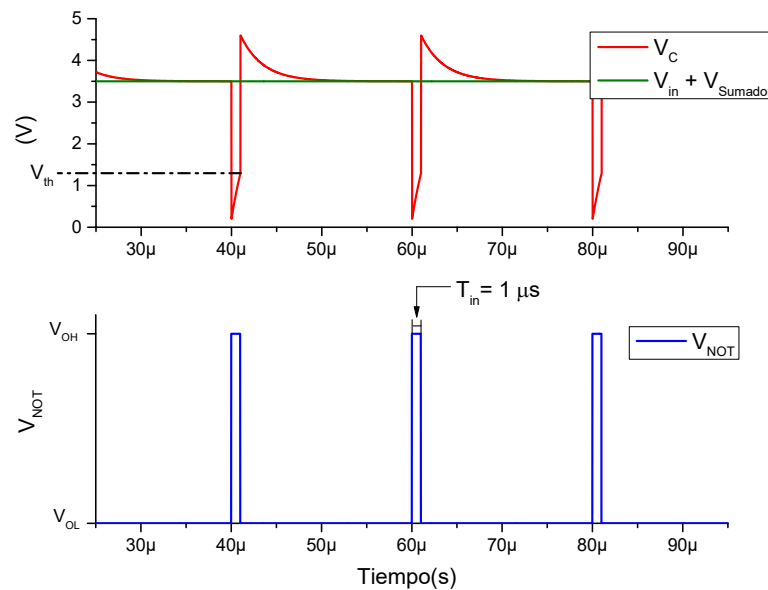


Figura 1.11: V_C y V_{NOT} considerando t_D del condensador

1.4. Método para obtener parámetros del ADC basado en el circuito one shot

Para obtener cada uno de los elementos del one shot ADC se deben tener en cuenta las características anteriormente mencionadas, como una mayor linealidad y t_D del condensador. Se debe tener claro la $f_{Sampling}$ a la que se quiere implementar el ADC, además del rango de V_{in} .

En primer lugar, a partir de la Eq. 1.3a, se obtiene $V_{Sumador}$. Una vez obtenido el $V_{Sumador}$ y con la Eq. 1.8b se obtienen R y C, siempre fijando el valor de uno de ellos.

Para ejemplificar el método, se implementa un ADC de 8-bit con un rango de V_{in} entre 0 V y 1 V con un $f_{Sampling}=1$ MSPS. Los parámetros de la FPGA son $V_{th}=1.3$ V, $\Delta V_O=3.3$ V. Se obtiene $V_{Sumador}=3.5$ V, además de $R=267 \Omega$ con la Eq. 1.9 suponiendo $C=500$ pF. Con esta información se obtiene $T_{inMax}=54.23$ ns a partir de la Eq. 1.2 y $t_D=945.77$ ns a partir de la Eq. 1.7. En la Fig. 1.12 se muestra el comportamiento de V_C y V_{NOT} con los valores hallados.

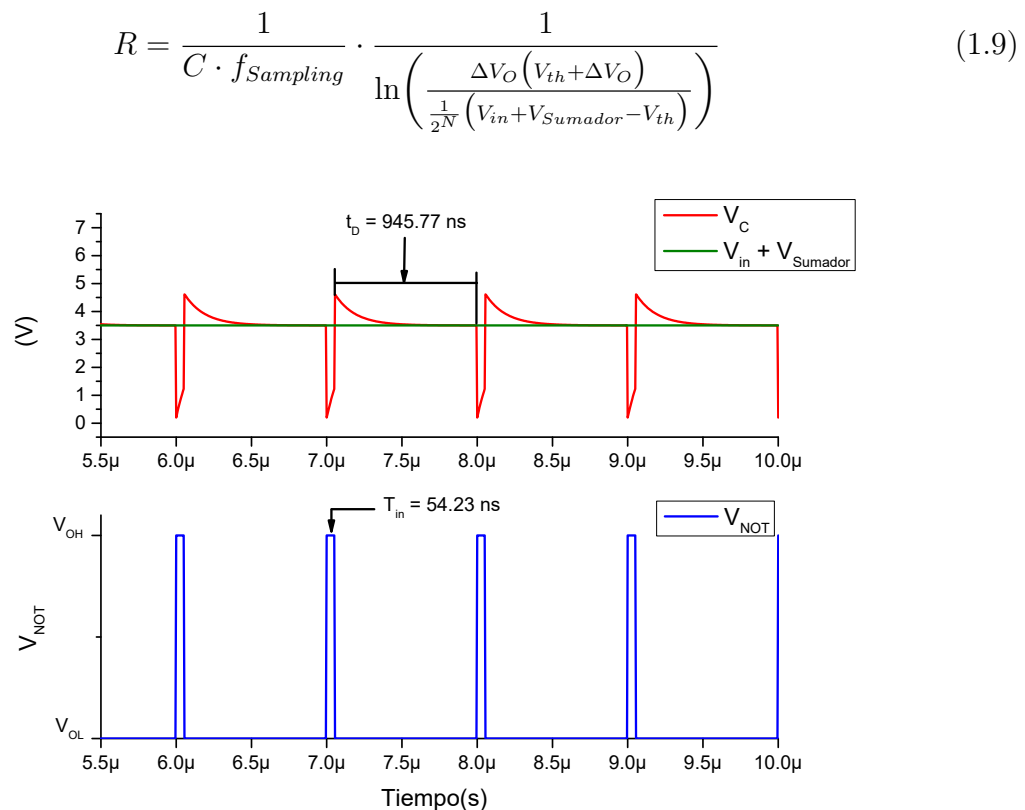


Figura 1.12: V_C y V_{NOT} con $f_{Sampling} = 1$ MSPS

1.5. Limitaciones de la implementación one shot ADC en una FPGA

Teniendo en cuenta que la implementación será hecha en una FPGA, se debe considerar el factor de la f_{clk} de la propia FPGA. En el caso de la FPGA usada como referencia se utiliza una $f_{clk}=100$ MHz, $V_{th}=1.3$ V y $\Delta V_O=3.3$ V, pero es posible realizar la implementación sobre diferentes familias de FPGA, ajustando los valores de V_{th} , ΔV_O y f_{clk} .

Para un one shot ADC de 8-bit, con $f_{Sampling}=1$ MSPS es necesario un $T_{inMax}=54.23$ ns y $t_D=945.77$ ns, estos valores fueron hallados en la sección 1.4. El contador implementado debe ser capaz de realizar 2^N cuentas en T_{inMax} . Es decir, para el caso de 8-bit realizar 256 cuentas en 54.23 ns, para así generar correctamente la conversión digital. Para poder realizar las 256 cuentas en dicho tiempo es necesaria una frecuencia del contador ($f_{counter}$) de 4.72 GHz, que se obtiene a partir de la Eq. 1.10.

$$f_{counter} = \frac{256}{54,23 \text{ ns}} \quad (1.10)$$

Además de ser una $f_{counter}$ muy alta esta frecuencia varía a medida que lo hace N, puesto que, a mayor N, t_D aumenta y el t_{inMax} disminuye, por lo que se necesitara mayor $f_{counter}$ para realizar las 2^N cuentas.

En el caso de tener una $f_{clk}=100$ MHz, para implementar el ADC de 8-bit es necesario un $T_{inMax}=2.56$ μ s ($\frac{2^N}{f_{clk}}$). Con la configuración del contador, cada ciclo de reloj incrementa en 1 el registro que lleva las cuentas, obteniendo $f_{counter} = f_{clk}$. Utilizando la propuesta presentada en [51] se implementa un contador que incrementa el registro de cuentas por flanco ascendente y descendente del reloj, de esta forma la $f_{counter} = 2 \cdot f_{clk}$, posibilitando un $T_{inMax}=1.28$ μ s ($\frac{2^N}{2 \cdot f_{clk}}$).

1.6. Implementación del one shot ADC de 8-bit en una FPGA

Se realiza la implementación del sistema en Simulink-Matlab con la herramienta System generator. Esta herramienta permite la integración del entorno Simulink con los elementos

configurados en la FPGA.

En la Fig. 1.13 se observan los diferentes bloques de la FPGA, además de los elementos externos del sistema. Se realiza la implementación del ADC de 8-bit, por lo que se realizan los cálculos teniendo en cuenta las limitaciones presentadas. En primer lugar, se halla $V_{Sumador}$ a partir de $V_{fs}=1$ V con la Eq. 1.3a, generando así un $V_{Sumador}=3.5$ V. Como se explicó en la sección 1.5 el contador debe realizar 2^N cuentas en T_{inMax} , por lo que T_{inMax} debe ser de 1.28 μ s.

Por último se obtiene $C=500$ pF, $R=6.2891$ k Ω y $t_D=22.24$ μ s, generando con ello una $f_{sampling}=42.518$ kSPS. Se muestra en la Fig. 1.14 el B resultante, a partir de la señal tipo rampa de V_{in} que va desde 0 V y 1 V.

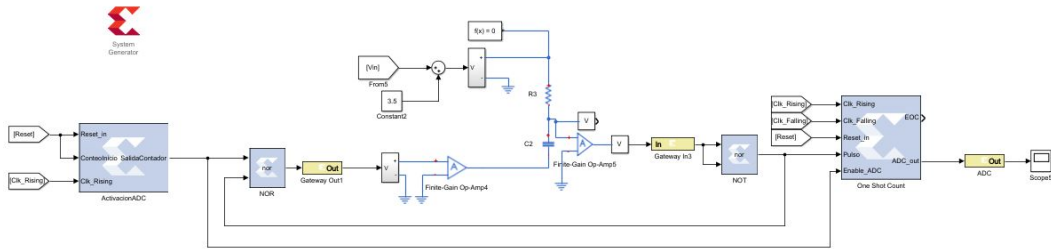


Figura 1.13: Esquemático de la simulación del one shot ADC en Simulink-Matlab

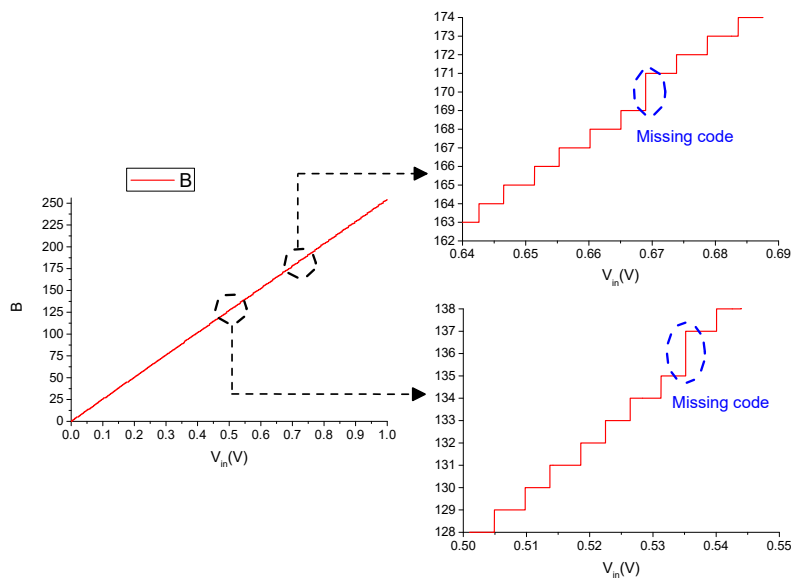


Figura 1.14: B de 8-bit con $f_{sampling}=42.518$ kSPS

Como se ve en la Fig. 1.14 se presentan varios casos de *missing code*. Para encontrar

la causa de dicho error, se realiza el cálculo teórico de los valores necesarios para que la respuesta sea $B=136$ y sus adyacentes en la Tabla 1.2. A partir de B se obtuvo el valor de V_{in} correspondiente, T_{in} se obtiene a partir de la Eq. 1.2, el número ideal de cuentas con $f_{counter} = 200$ MHz se obtiene dividiendo T_{in} por $\frac{1}{f_{counter}}$ y por último se muestra el número real de cuentas generadas en la FPGA.

En la Tabla 1.2 se distingue claramente que para los casos de $B=136$ y $B=137$ el número de cuentas generadas por la FPGA es igual, lo que conllevaría a un error en B . Este error es debido a la no linealidad entre el V_{in} y T_{in} que se abordó anteriormente.

B	V_{in} (V)	T_{in} (ns)	Contador ideal con $f_{counter}=200$ MHz	Cuentas generadas en la FPGA con $f_{counter}=200$ MHz
135	0.527344	599.3309	119.8662	119
136	0.531250	594.8303	118.9661	118
137	0.535156	590.3362	118.0672	118

Tabla 1.2: B , V_{in} , T_{in} y número de cuentas posibles para un one shot con $T_{inMax}=1.28$ μ s

Si la relación entre V_{in} y T_{in} fuese lineal (caso ideal), la diferencia entre dos T_{in} generados para dos B adyacentes, sería igual para todos los casos posibles. En la Fig. 1.15 se aprecia que, en el caso propuesto, esta diferencia varía a lo largo de B . La Fig. 1.15 se realizó con los valores hallados anteriormente, $V_{Sumador} = 3,5$ V, $C = 500$ pF y $R = 6,2891$ k Ω .

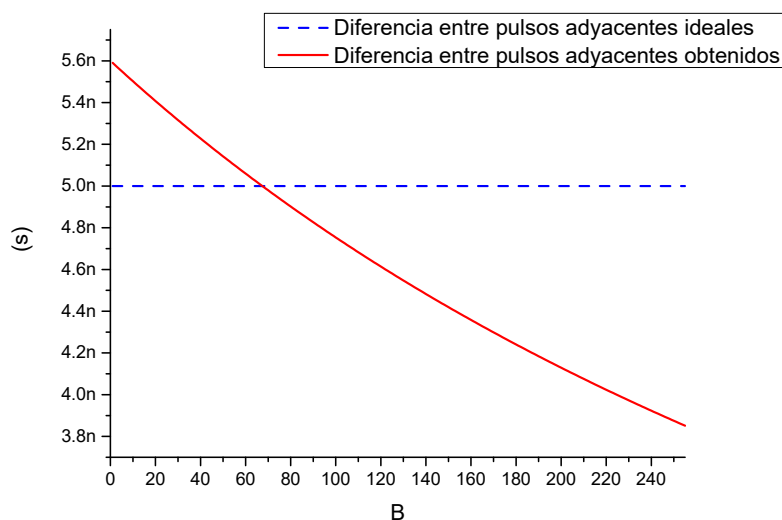


Figura 1.15: Diferencia entre los pulsos adyacentes de un one shot ADC de 8-bit con $T_{inMax} = 1,28$ μ s

1.7. Síntesis del diseño del ADC basado en one shot

Una de las mayores ventajas del one shot ADC es su versatilidad. Si se requiere un ADC con un número indicado de bits de resolución, sólo harían falta unas modificaciones para que pudiese ser implementado. Además de permitir ser implementado en otra FPGA con diferentes valores de V_{th} , ΔV_O y f_{clk} .

Para realizar el diseño del ADC con resolución N-bit, con los parámetros V_{th} , ΔV_O y f_{clk} de la FPGA, se realiza el siguiente procedimiento.

1. En primer lugar basándose en el rango de V_{in} deseado se obtiene $V_{Sumador}$ con la Eq. 1.3b.
2. A partir de $f_{Sampling}$ deseada y el $V_{Sumador}$ hallado, se obtienen los valores de R y C con la Eq. 1.9.
3. Una vez sean hallados los valores de R y C , se obtiene T_{inMax} con la Eq. 1.2 y t_D con la Eq. 1.7. Para obtener T_{inMax} se evalúa T_{in} en la Eq. 1.2 con V_{in} igual a la tensión mínima del ADC diseñado.
4. Al obtenerse T_{inMax} , se debe considerar el número de cuentas posible del contador en T_{inMax} . Si se implementa el contador con $f_{counter} = f_{clk}$ se utiliza la Eq. 1.4. Si se implementa con $f_{counter} = 2 \cdot f_{clk}$ sería el doble del resultado de la Eq. 1.4. Si es posible realizar 2^N cuentas en dicho tiempo, se puede implementar con los valores hallados. De lo contrario si no es posible realizar el total de las cuentas, se debe modificar R y C para cumplir dicho cometido. Este cambio en R y C generarían cambios en $f_{Sampling}$, T_{inMax} y t_D .

La $f_{Sampling}$ está limitada por T_{inMax} y t_D . Si estos parámetros aumentan la $f_{Sampling}$ disminuye. Además, si N aumenta, también lo hace T_{inMax} y t_D , por lo que la $f_{Sampling}$ disminuye.

1.7.1. ADC de 10-bit basado en el circuito one shot

Para ejemplificar la síntesis del diseño, se implementa un ADC de 10-bit, con un rango de V_{in} entre 0 V y 1 V y una $f_{counter} = 200$ MHz. Se obtienen los valores mostrados en la

Tabla 1.3.

$f_{Sampling}$	8,94 kSPS
T_{inMax}	5,12 μs
$V_{Sumador}$	3,5 V
ΔV_O	3,3 V
V_{th}	1,3 V
R	25,23 k Ω
C	500 pF
t_D	106,7 μs

Tabla 1.3: Características del one shot para ADC de 10 bits con $f_{Sampling} = 8,94$ kSPS

Para la implementación del one shot ADC de 10 bits hay que tener en cuenta algunos factores. En primer lugar, debido a que la resolución del ADC es mayor, se debe asegurar de que el tiempo de descarga sea el apropiado. Utilizando la Eq.1.7 se obtiene que el t_D para el one shot ADC de 10 bits es de $8,46 \cdot R \cdot C$.

Con la información anterior se obtiene la Fig. 1.16, donde se observa que el resultado es el esperado, aunque se presenta el mismo inconveniente del error en las cuentas. Se muestran los cálculos en la Tabla 1.4.

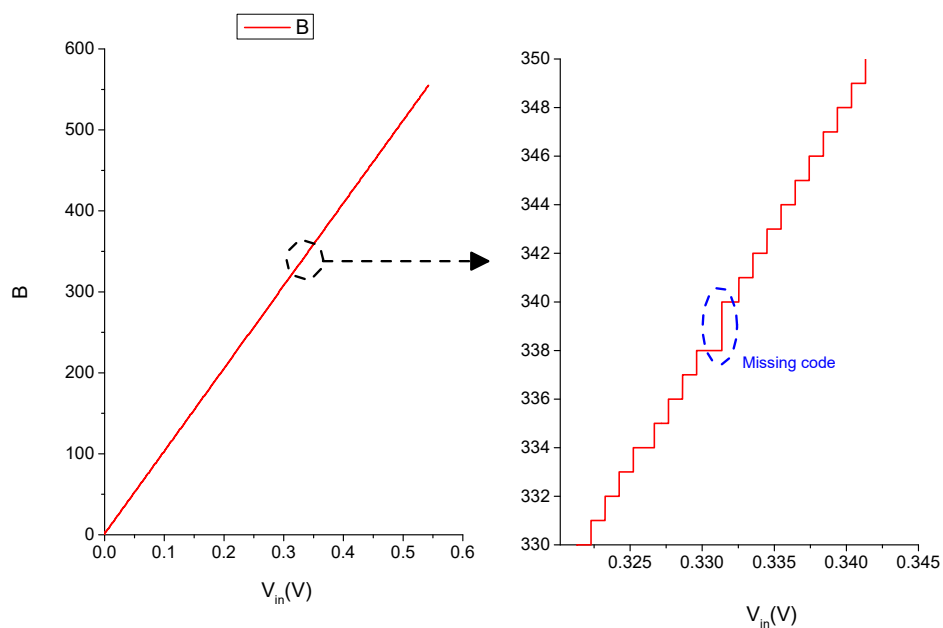


Figura 1.16: Resultado de la conversión del one shot de 10 bits con $f_{Sampling} = 8,94$ kSPS

B	$V_{in}(\text{V})$	$T_{in}(\mu\text{s})$	Contador ideal con $f_{counter}=200 \text{ Mhz}$	Cuentas generadas en la FPGA con $f_{counter}=200 \text{ Mhz}$
338	0.3300781	3.3499	669.97	669
339	0.3310547	3.3450	669.01	669
340	0,3320313	3.3401	668.025	668

Tabla 1.4: B , V_{in} , T_{in} y número de cuentas posibles para un one shot con $f_{Sampling} = 8,94 \text{ kSPS}$

1.8. Posibles métodos de mejora

El problema del t_D que debe tener el condensador para llegar a su estado estacionario, hace que se retarde el momento en que se realiza la siguiente conversión. Una posible solución para dicho problema es conocer el comportamiento de V_C durante la descarga, de este modo se sabría valor de $V_{in} + Error$ al momento de la siguiente conversión. Lo anterior requeriría una precisión alta en la operación matemática del comportamiento de V_C durante la descarga, además de una mayor f_{clk} , para obtener V_C de manera exacta.

1.9. Conclusiones

En este capítulo se analiza la implementación en una FPGA de un convertor analógico digital (ADC) a partir del circuito one shot, dónde se analizan los limites existentes del diseño, las ventajas y desventajas de este. El ADC a partir del circuito one shot es analizado debido a su facilidad de implementación y su bajo coste. El circuito one shot genera un pulso en la puerta lógica NOT (V_{NOT}), cuya anchura (T_{in}) es modulada por la tensión de entrada (V_{in}), a través de un circuito RC y puertas lógicas. Por lo tanto, a partir del T_{in} generado se obtiene el valor digital correspondiente a la tensión de entrada V_{in} .

Inicialmente se configura un contador con $f_{counter} = f_{clk}$, esto hace T_{inMax} sea $T_{inMax} = \frac{2^N}{f_{clk}}$. Si se configura que el contador tenga $f_{counter} = 2 \cdot f_{clk}$ T_{inMax} pasaría a ser $T_{inMax} = \frac{2^N}{2 \cdot f_{clk}}$. Esta reducción de T_{inMax} permite el aumento de la $f_{Sampling}$ del ADC.

T_{in} y V_{in} presentan una relación exponencial, por lo que es necesario modificar los parámetros y así poder trabajar en la zona con un comportamiento más lineal de esta

relación. Se modifica el circuito one shot, incrementando la constante de tiempo del condensador y añadiendo un $V_{Sumador}$. Como ventaja se obtiene un sistema con un comportamiento más lineal, pero como desventaja el tiempo de descarga del condensador (t_D) aumenta. Este aumento genera que la $f_{Sampling}$ disminuya.

La versatilidad al implementar un ADC con la técnica one shot, se ve limitada por el rango de V_{in} deseado y $f_{counter}$. Esto es debido a que, de ser necesario un rango de V_{in} más amplio del utilizado en la sección 1.6, se estaría trabajando en una zona que se comporta menos lineal en la curva de carga de V_C , generando así muchos errores en B . Por otro lado $f_{counter}$ limita el T_{inMax} posible, debido a que se deben realizar 2^N cuentas en ese periodo de tiempo.

Las simulaciones fueron realizadas para una FPGA de la familia Zynq 7000 con una f_{clk} de 100 MHz. Sin embargo, este método puede ser extrapolado a diferentes familias de FPGA, con diferentes f_{clk} , ΔV_O y V_{th} . Se puede implementar con el mismo método, cambiando los parámetros al realizar los cálculos mostrados en el capítulo.

2. Capítulo 2

2.1. Resumen

En este capítulo se propone el diseño de un convertor analógico digital (ADC) basado en successive approximation register (SAR), reemplazando el convertor digital analógico (DAC) del SAR con un modulador de ancho de pulso (PWM) y un filtro pasa bajo (LPF). Este método permite la implementación de un ADC con elementos de bajo coste, además de ser parametrizable y escalable. La lógica SAR será implementada en una FPGA y se analizarán las ventajas y desventajas del diseño.

2.2. ADC basado en SAR

El ADC basado en SAR es una arquitectura implementada en tres etapas consecutivas. La lógica SAR, un DAC y un comparador analógico. Se puede apreciar el esquemático en la Fig. 2.1.

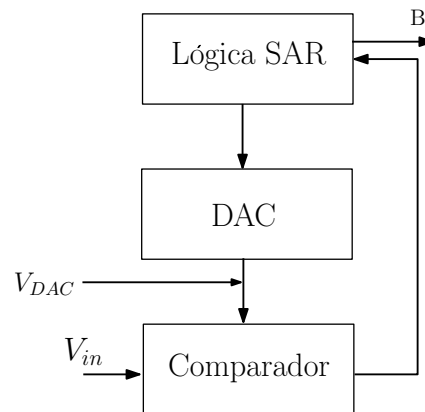


Figura 2.1: Esquemático de un ADC basado en SAR

La lógica SAR es utilizada en aplicaciones que requieren un cómputo complicado y velocidad de procesamiento muy alta. Por ejemplo, en [52, 53] se utiliza para realizar operaciones matemáticas como la división de decimales o raíz cuadrada, también es usado en el control de motores industriales [54].

La conversión del ADC basado en SAR es un proceso que se repite periódicamente, donde se testean diferentes valores digitales, con el objetivo de obtener la conversión digital (B) correspondiente a la tensión analógica de entrada (V_{in}).

Un ejemplo del funcionamiento de la arquitectura SAR se puede apreciar en la Fig. 2.2, donde se puede observar que la tensión de salida del convertor digital analógico (V_{DAC}) se acerca a V_{in} en base a la información obtenida por el comparador analógico.

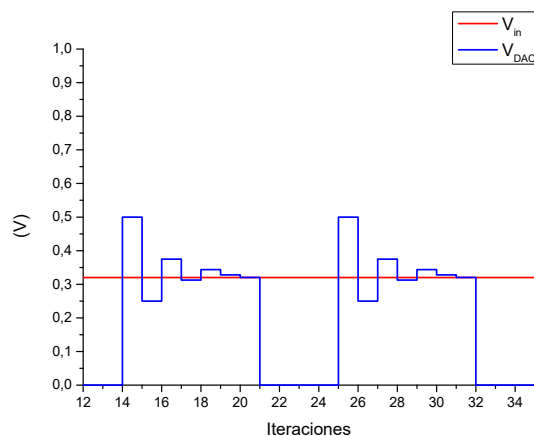


Figura 2.2: V_{DAC} y V_{in} en ADC basado en SAR de 7-bit

La idea general del ADC basado en SAR es comparar V_{in} con una tensión continua generada a partir de un número binario controlado por la lógica SAR. Esto permite evaluar los diferentes niveles de tensión, para finalmente converger a B [55–57]. En cada iteración se compara V_{DAC} con V_{in} comenzando desde el *MSB* (bit más significativo), permitiendo saber si se aumenta o disminuye V_{DAC} en la siguiente iteración.

El procedimiento de conversión es el siguiente. La lógica SAR indica el valor que debe tener V_{DAC} y este se compara con V_{in} . Basándose en la respuesta del comparador analógico ($V_{Comparador}$), la lógica SAR reescribe el valor de B .

Al inicio de la conversión, la lógica SAR inicializa todos los bits en '0'. En la primera iteración *MSB* pasa a ser '1', mientras el resto de los bits son '0'. Este valor es enviado al V_{DAC} y se realiza la comparación. Si $V_{Comparador}$ es '1' indica que V_{in} es mayor que V_{DAC} , en caso contrario, cuando $V_{Comparador}$ ='0' significa que V_{in} es menor que V_{DAC} .

Cuando se haya realizado la comparación, la lógica SAR escribe en *MSB* el resultado de $V_{Comparador}$, generando un nuevo V_{DAC} para la siguiente iteración. Una vez se haya almacenado el *MSB*, se realiza el mismo procedimiento con el siguiente bit más

significativo, teniendo en cuenta que *MSB* se mantiene en el valor obtenido anteriormente.

Se realiza el mismo procedimiento para cada bit hasta llegar al *LSB* y generar B , el cual se obtiene almacenando el último valor generado por la lógica SAR. Una vez hecho esto V_{DAC} vuelve a su estado inicial con todos los bits a '0', a la espera de una nueva conversión.

Para ejemplificar este procedimiento, la Fig. 2.3 muestra la conversión para $V_{in} = 0,35 V$ en un ADC de 4-bit. En la Eq. 2.1 se aprecia B obtenido teóricamente. El rango de V_{in} del ADC está entre $0 V$ y $V_{fs} = 1 V$.

$$B = \frac{V_{in} \cdot 2^N}{V_{fs}} = \frac{0,35 \cdot 16}{1} = 5,6 \quad (2.1)$$

$$B = 5 = 0101$$

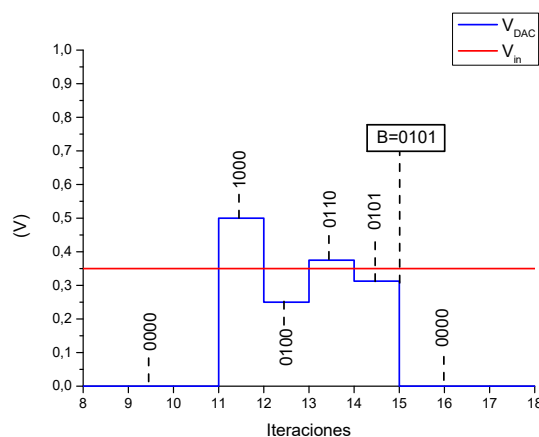


Figura 2.3: V_{DAC} y V_{in} para un ADC basado en SAR de 4-bit

Como se puede apreciar en la Fig. 2.3, B se obtiene en $N+1$ (siendo N el número de bits del ADC) iteraciones, desde que V_{DAC} se encuentra en su estado inicial, hasta que se obtiene *LSB*.

Se realiza una modificación en el proceso de conversión para mejorar las prestaciones del ADC. En el estado inicial del V_{DAC} del procedimiento explicado todos los bits son '0'. Una vez iniciada la conversión *MSB* pasa a ser '1' mientras el resto de los bits continúan en '0'. Este primer paso es una iteración más en la conversión, por lo que se modifica la lógica SAR de modo tal que, el estado inicial sea $MSB='1'$ y el resto de los bits en '0'.

Esta modificación permite disminuir el número de iteraciones necesarias, debido a que V_{DAC} con $MSB='1'$ y el resto de los bits '0', es el estado inicial y a su vez la primera comparación. Esta modificación se puede apreciar en la Fig. 2.4, ahora son necesarias N iteraciones para obtener B .

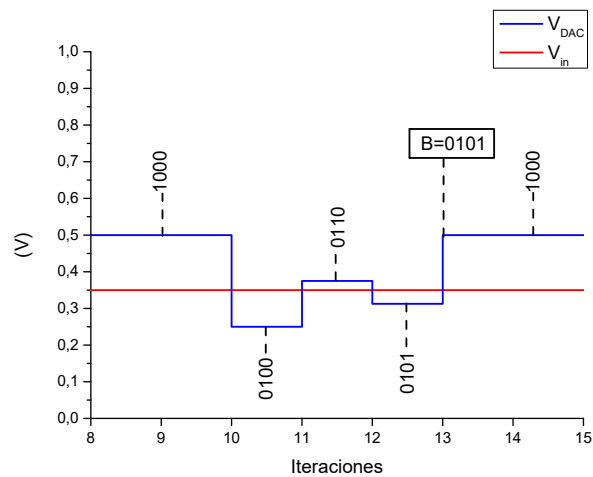


Figura 2.4: V_{DAC} y V_{in} para un ADC de 4-bit basado en SAR con modificación en el estado inicial

Este método es extrapolable a un ADC de diferente resolución. En la Fig. 2.5 se muestra el comportamiento de V_{DAC} para un ADC de 8-bit.

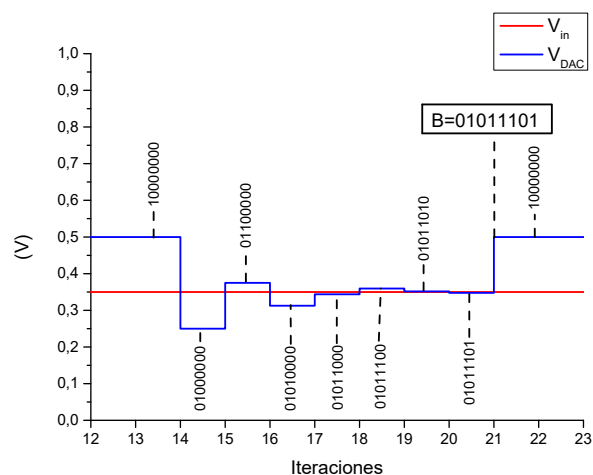


Figura 2.5: V_{DAC} y V_{in} para un ADC basado en SAR de 8-bit

2.2.1. DAC implementado con un LPF y un PWM

Con el objetivo de implementar un ADC con una menor cantidad de entradas de la FPGA, se pretende reemplazar el DAC, basándose en el método presentado en [58]. Se reemplaza el DAC utilizando un PWM y un LPF, quedando el esquemático mostrado en la Fig 2.6.

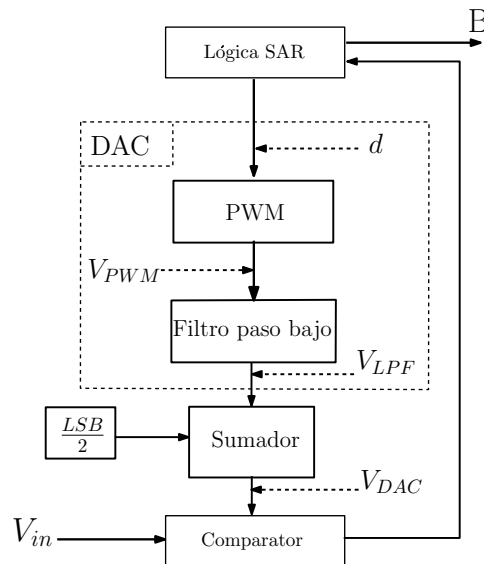


Figura 2.6: Esquemático del ADC basado en SAR con modificación del DAC

El DAC implementado con PWM y LPF permite que se genere una tensión en la salida del LPF (V_{LPF}), acorde al ciclo de trabajo (D) del PWM. D tiene el rango mostrado en la Eq. 2.2, variando entre 0 y 1.

$$D = \frac{d}{2^N} \quad (2.2)$$

$$0 \leq d \leq 2^N - 1$$

La variación de V_{LPF} con respecto D , permite obtener el V_{DAC} requerido por la lógica SAR. Para obtener V_{DAC} se debe implementar un sumador analógico con el objetivo de seguir la Eq. 2.3, de modo que el error de conversión pase de ser $\pm LSB$ a $\pm \frac{1}{2} LSB$.

$$V_{DAC} = V_{LPF} - \frac{LSB}{2} \quad (2.3)$$

El rango de V_{in} depende de las tensiones de salida de la FPGA, como se muestra en la Eq. 2.4, donde V_{OH} es la salida de la FPGA cuando el valor lógico es '1' y V_{OL} cuando es '0'.

$$V_{OL} \leq V_{in} < V_{OH}$$

$$\Delta V_O = V_{OH} - V_{OL} \quad (2.4)$$

$$V_{fs} = V_{OH}$$

La Fig 2.7 muestra, a modo de ejemplo, el resultado de la simulación de un PWM de 3-bit con frecuencia del PWM (f_{PWM}) de 12.5 MHz y un LPF con frecuencia de corte (f_C) de 4 MHz y orden del LPF (O_{LPF}) de 3. Se aprecia como varía V_{LPF} con respecto a D , permitiendo obtener el V_{DAC} necesario para la lógica SAR. Es posible observar que cada variación en V_{LPF} es equidistante a la otra.

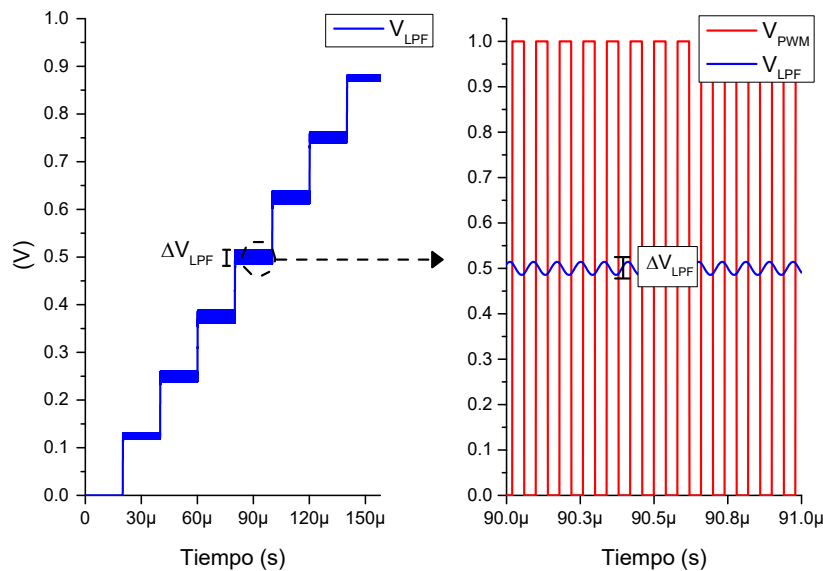


Figura 2.7: V_{LPF} para PWM de 3-bit y LPF con $f_C=4$ MHz y $O_{LPF}=3$, variando d entre 0 y $2^N - 1$.

V_{LPF} se obtiene a partir de la Eq. 2.5, donde su tensión media ($\overline{V_{LPF}}$) depende de D y ΔV_O . Debido a que el LPF no es ideal V_{LPF} presenta un rizado (ΔV_{LPF}) asociado a la atenuación en f_C y al O_{LPF} del LPF. Para que el ADC pueda ser implementado con un PWM y el LPF, ΔV_{LPF} debe ser menor a LSB . Para obtener LSB utiliza la Eq. 2.6.

$$V_{LPF}(t) = \overline{V_{LPF}} + \Delta V_{LPF}(t) \quad (2.5)$$

$$\overline{V_{LPF}} = D \cdot \Delta V_O$$

$$\begin{aligned} \Delta V_{LPF} &\leq LSB \\ LSB &= \frac{\Delta V_O}{2^N} \end{aligned} \tag{2.6}$$

2.2.2. Diseño e implementación del PWM

La implementación del PWM de N-bit, se realiza con una máquina de estados regida por un contador ascendente que cuenta desde 0 hasta $2^N - 1$ en cada periodo del PWM. La máquina de estados se muestra en la Fig. 2.8.

Se configura un contador que aumenta en 1 el registro de cuentas en cada flanco ascendente del reloj. D varía entre 0 y 1, donde $D = \frac{d}{2^N}$ y d varía entre 0 y $2^N - 1$. Para obtener la f_{PWM} se hace a través de la Eq. 2.7.

$$f_{PWM} = \frac{f_{clk}}{2^N} \tag{2.7}$$

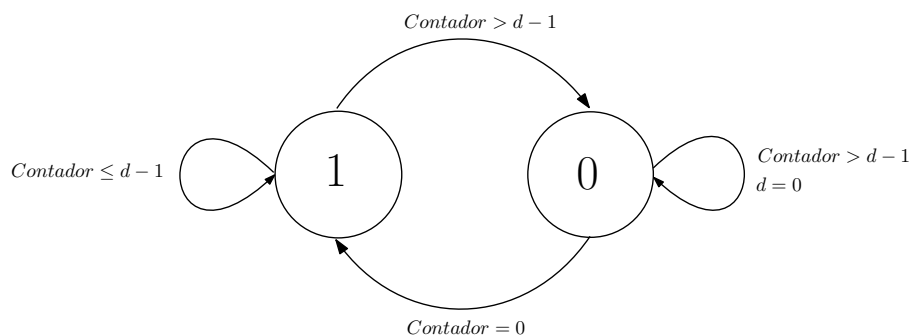


Figura 2.8: Máquina de estados del PWM

Las cuentas del contador aumentan en 1 en cada flanco ascendente del reloj. Una vez el contador alcanza $2^N - 1$, este vuelve a 0 y empieza un nuevo periodo del PWM. El PWM pasa a '1' cuando el Contador=0 y $d \neq 0$, y se mantiene en '1' hasta que el Contador= $d - 1$. El funcionamiento del contador ascendente que rige el PWM de 2-bit con $D = \frac{2}{4}$ se muestra en la Fig. 2.9.

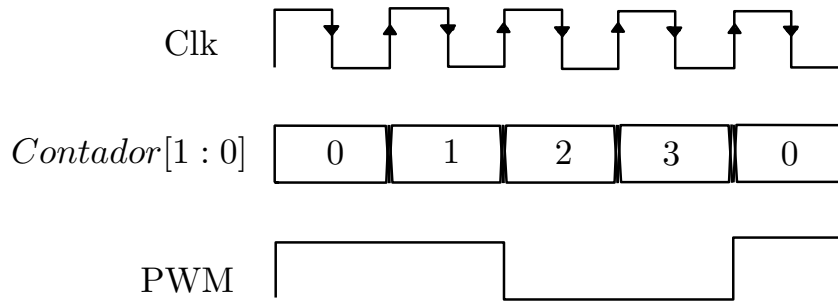


Figura 2.9: Comportamiento del contador ascendente con un PWM de 3 bits

La implementación del contador ascendente se realiza a través de un circuito sumador, N flip-flops y una máquina de estados (FSM). El contador genera '1' lógico en la salida durante un periodo de reloj (T_{Reloj}), indicando que se han realizado las cuentas requeridas, es decir cuando el contador es igual al parámetro ingresado por el usuario, que en el caso del PWM es el valor de d . El contador es el encargado de indicarle al PWM cuando debe cambiar de estado, es decir, cuando el $Contador = d - 1$. Se muestra el esquemático de la implementación del contador en la Fig. 2.10.

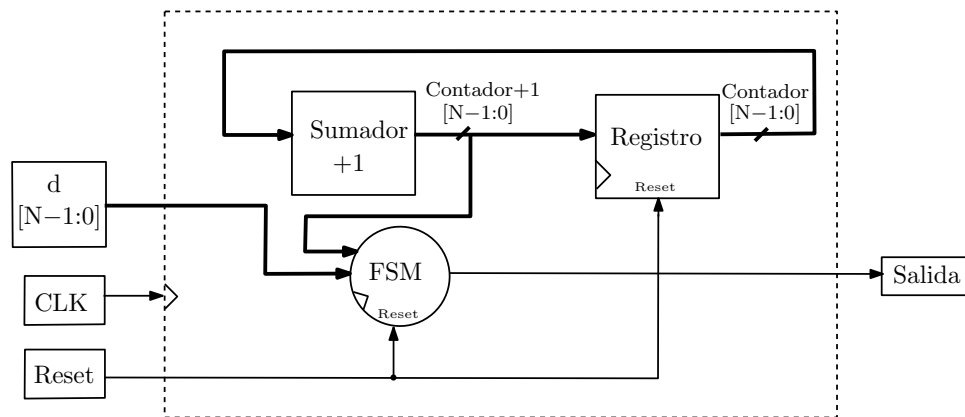


Figura 2.10: Esquemático de la implementación de un contador ascendente para el PWM

2.2.3. Filtro pasa bajo

Uno de los objetivos del LPF es obtener un V_{LPF} con un ΔV_{LPF} menor a un LSB . De esta manera, el ΔV_{LPF} quedaría incluido dentro del error de precisión del ADC, que es $\pm \frac{LSB}{2}$. Para lograr esto, se modifica f_C y/o O_{LPF} .

Si se requiere un ADC de 8-bit, con un PWM de $\Delta V_O = 1\text{ V}$ ($\Delta V_O = V_{OH} - V_{OL}$), es

necesario que ΔV_{DAC} sea menor a $LSB=3.906$ mV. Se debe encontrar una f_C y O_{LPF} , capaz de obtener dicho resultado.

Teniendo en cuenta que la f_{PWM} es constante, cuanto menor sea f_C respecto a f_{PWM} , menor será ΔV_{LPF} . Otra forma de disminuir ΔV_{LPF} es aumentando O_{LPF} . Cada unidad que se aumenta O_{LPF} , la pendiente de la atenuación aumenta asintóticamente en $20 \frac{dB}{dec}$.

A continuación, se analizarán las diferentes arquitecturas para la implementación del LPF con sus respectivas ventajas y desventajas, haciendo énfasis en su respuesta en frecuencia, atenuación y tiempo de establecimiento ($t_{Settling}$). Se analizarán las arquitecturas Butterworth, Bessel, Chebyshev, Chebyshev inverso y Elíptica.

En primer lugar, se descartan las arquitecturas Elíptica y Chebyshev inverso, debido a que ofrecen respuestas al escalón muy lentas con respecto al resto de arquitecturas [59, 60], haciendo que la $f_{Sampling}$ sea menor.

Cada iteración de la lógica SAR requiere un $t_{Settling}$ del DAC debido al LPF y los componentes analógicos. Por lo tanto, en un ADC de N-bit la $f_{Sampling}$ se obtiene según la Eq. 2.8.

$$f_{Sampling} = \frac{1}{N \cdot t_{Settling}} \quad (2.8)$$

Se analizan las arquitecturas Butterworth, Bessel y Chebyshev, las cuales presentan una buena respuesta al escalón. En la Fig. 2.11 se muestra el diagrama de Bode normalizado para una $f_C=1$ Hz y $O_{LPF}=3$ para cada arquitectura. Como se puede observar, el LPF Chebyshev es el que mejor atenuación tiene, pero presenta una resonancia en su banda no eliminada, generando un rizado en el dominio del tiempo.

Los LPFs Bessel y Butterworth presentan una respuesta de fase lineal en la banda no eliminada, por lo que no distorsiona la señal. El Butterworth presenta mejor transición entre la banda no eliminada y la banda eliminada que el Bessel.

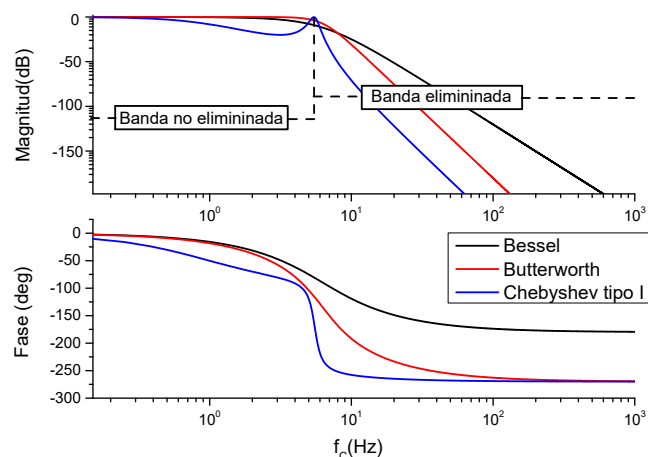


Figura 2.11: Diagrama de Bode para las diferentes arquitecturas del LPF

Ahora se estudia la respuesta al escalón de las diferentes arquitecturas. Para esto se utilizan las gráficas normalizadas con $f_C=1$ Hz y $O_{LPF}=3$ y se muestran en la Fig. 2.12. El LPF Bessel tiene un $t_{Settling}$ menor al resto de arquitecturas, mientras el LPF Chebyshev es el que más tarda en establecerse.

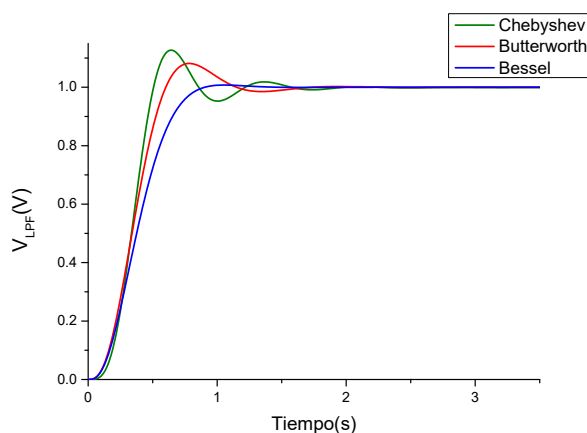


Figura 2.12: Comparación de la respuesta al escalón de diferentes arquitecturas de LPF, con $f_C=1$ Hz y $O_{LPF}=3$.

Como hemos observado el filtro Bessel es el LPF que tiene la transición más lenta de la banda no eliminada a la banda eliminada, pero es el que ofrece menor $t_{Settling}$ [61]. Esto lo hace ideal para ciertas aplicaciones como transmisión digital de señales [62], transmisión de señales de TV y procesamiento de señal de vídeo [63]. El LPF Bessel es el que ofrece menor tiempo de establecimiento, por lo que se escoge para la implementación del DAC. Los polos del LPF Bessel se localizan sobre un círculo y se distancian en su parte imaginaria

por $\frac{2}{p}$, siendo p el número de polos del LPF. La distancia entre los últimos polos y el radio del círculo es $\frac{1}{p}$. Se muestra en la Fig. 2.13 los polos resultantes en un LPF de $O_{LPF}=5$.

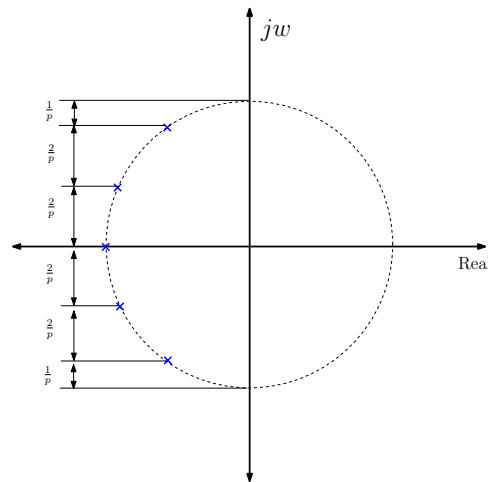


Figura 2.13: Ubicación de los polos del LPF de arquitectura Bessel de $O_{LPF}=5$

A medida que f_C disminuye, el $t_{Settling}$ del LPF es mayor. Esto se puede observar en la Fig. 2.14, donde se compara la respuesta al escalón de diferentes f_C con un $O_{LPF}=4$. Por otra parte, al disminuir f_C , ΔV_{LPF} también disminuye [64]. Teniendo en cuenta esto se debe escoger una f_C que cumpla la Eq. 2.6, pero con el menor $t_{Settling}$ posible.

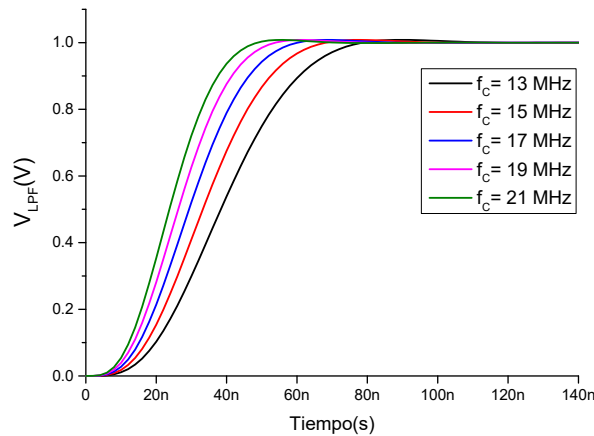


Figura 2.14: Comparación de la respuesta al escalón del LPF con $O_{LPF}=4$ y diferentes f_C .

Otro de los factores a tener en cuenta a la hora de diseñar e implementar el LPF es el O_{LPF} . Se muestra en la Fig. 2.15 la respuesta al escalón de diferentes O_{LPF} con $f_C=13$ MHz. Se aprecia que a medida que el O_{LPF} aumenta y teniendo f_C constante el $t_{Settling}$ es mayor. Otro inconveniente al aumentar el O_{LPF} es el aumento del número de

componentes al implementar el filtro, incrementando a su vez el coste del diseño y el área de implementación.

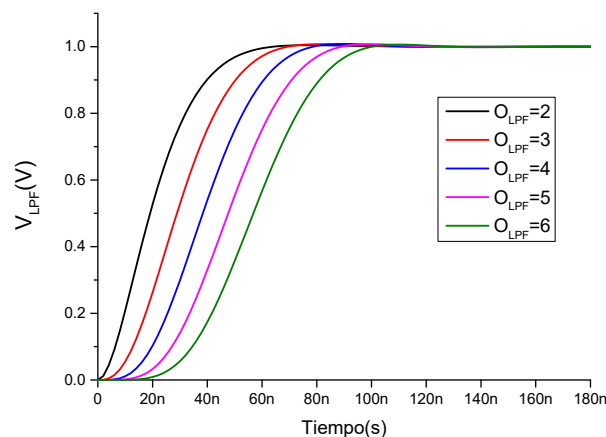


Figura 2.15: Comparación de la respuesta al escalón del LPF con $w_C = 2\pi \cdot 13 \cdot 10^6 \frac{rad}{s}$ para diferentes O_{LPF} .

Ahora bien, por cada O_{LPF} aumentado, la pendiente de la magnitud de la respuesta en frecuencia aumenta en $20 \frac{dB}{dec}$. Esto hace que ΔV_{LPF} disminuya, permitiendo escoger una f_C que cumpla con la condición 2.6 y tenga menor $t_{Settling}$

Para hallar el $t_{Settling}$ del LPF se utiliza la gráfica normalizada de la respuesta al escalón presentada en la Fig. 2.16, donde se divide el eje X de la gráfica por la f_C utilizada con su correspondiente O_{LPF} .

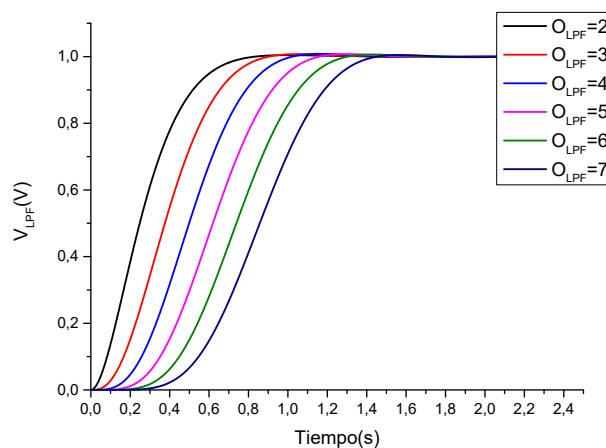


Figura 2.16: Respuesta al escalón normalizado del LPF Bessel para una $f_C=1$ Hz con diferentes O_{LPF} .

En la Tabla 2.1 se muestra el $t_{Settling}$ para cada O_{LPF} , con el que se realizarán los cálculos de los parámetros del LPF y el ADC. Como se explicó anteriormente, para obtener $t_{Settling}$

se divide la gráfica normalizada por f_C del O_{LPF} correspondiente, utilizando la Eq. 2.9. A partir de la Tabla 2.1 se puede representar gráficamente f_C vs. $t_{Settling}$ para los diferentes O_{LPF} , permitiendo encontrar rápidamente el valor de $t_{Settling}$ para cada f_C y de esta manera encontrar los parámetros del LPF a implementar. Se muestra la gráfica de f_C vs. $t_{Settling}$ para f_C desde 1 Hz hasta 20 Hz en la Fig. 2.17.

$$t_{Settling} = \frac{t_{Settling-Normalizado}}{f_C} \quad (2.9)$$

O_{LPF}	2	3	4	5	6	7	8	9	10
$t_{Settling}$ normalizado	0.65	1.00	1.25	1.52	1.75	1.90	2.10	2.21	2.32

Tabla 2.1: $t_{Settling}$ normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .

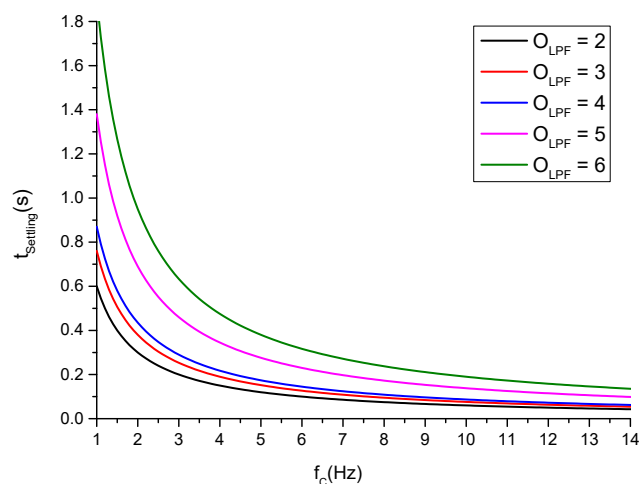


Figura 2.17: $t_{Settling}$ Vs. f_C para diferentes O_{LPF} con f_C desde 1 Hz hasta 20 Hz

En resumen, para la implementación del LPF se deben tener en cuenta varios factores. En primer lugar, la condición mostrada en la Eq. 2.6. Para disminuir ΔV_{LPF} se puede aumentar el O_{LPF} y/o disminuir f_C . En ambos casos el $t_{Settling}$ es mayor. En el caso contrario, si se aumenta f_C y se disminuye el O_{LPF} , se tendrá un $t_{Settling}$ menor haciendo que el ADC tenga una mayor $f_{Sampling}$. Este comportamiento se resume en la Tabla 2.2.

Parámetro	$t_{Settling}$	ΔV_{LPF}	Coste
Mayor f_C	Menor	Aumenta	Constante
Mayor O_{LPF}	Mayor	Disminuye	Aumenta

Tabla 2.2: Parámetros que afectan la dinámica del LPF

2.2.4. Implementación del filtro pasa bajo

Una clasificación de los LPFs es en activos y pasivos. Los filtros pasivos son lineales y están basados en resistencias, condensadores y bobinas. Estos filtros no dependen de una fuente externa y son difíciles de sintonizar por lo que se descartarían debido a la necesidad de mayor precisión [65]. Los filtros activos tienen la misma combinación de elementos que los filtros pasivos, pero con la inclusión de un elemento activo, el amplificador operacional (AO). Su sintonización es más simple, además que permite amplificar la señal en caso de ser necesario [66].

Para la implementación del LPF se utilizará el circuito Sallen-Key mostrado en la Fig. 2.18. Este circuito implementa un filtro de $O_{LPF}=2$ combinando dos resistencias y dos condensadores con un AO. Para implementar un filtro de mayor O_{LPF} se adicionan etapas en cascada.

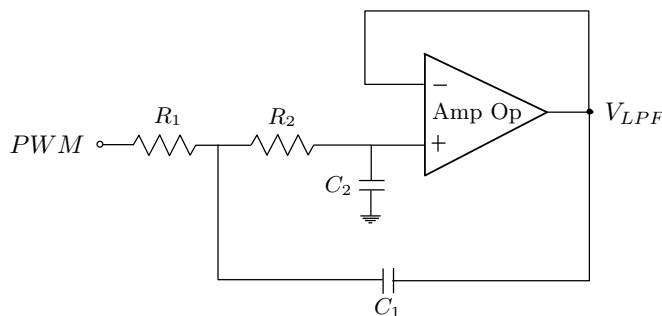


Figura 2.18: Arquitectura Sallen-Key para LPF de $O_{LPF}=2$

Una de las ventajas de la topología Sallen-Key es su facilidad de implementación, además de que el AO trabaja como no inversor. Esta topología es la adecuada para factores de calidad (Q) menores a 3 [67]. La arquitectura Bessel utiliza $Q < 2$ para O_{LPF} menores a 10 [68].

Para una ganancia unitaria de la topología Sallen-Key se obtiene la función de transferencia mostrada en la Eq. 2.10 y obtenida en [69], siendo $w_c = 2\pi \cdot f_C$

$$H(s) = \frac{1}{1 + w_c C_1 (R_1 + R_2) s + w_c^2 R_1 R_2 C_1 C_2 s^2} \quad (2.10)$$

Para diseñar el LPF con la topología Sallen-Key se deben tener en cuenta f_c , O_{LPF} y el tipo de filtro. Teniendo en cuenta que se implementa el LPF con arquitectura Bessel, se utiliza la Tabla 2.3. Esta tabla contiene los parámetros normalizados necesarios para implementar el LPF [70].

O_{LPF}	Q_1	f_{p1}	Q_2	f_{p2}	Q_3	f_{p3}	Q_4	f_{p4}
2	0.577	1.274						
4	0.52	1.432	0.81	1.606				
6	0.51	1.607	1.02	1.908	0.61	1.692		
8	0.51	1.781	1.23	2.192	0.71	1.956	0.56	1.835

Tabla 2.3: Valores normalizados a $f_c = 1$ Hz del LPF Bessel .

Para implementar el LPF de $O_{LPF}=2$, se utiliza la Tabla. 2.3. En esta tabla se encuentran los polos normalizados y el valor de Q para cada O_{LPF} . Los polos, generan la dinámica del filtro Bessel. Con f_c , f_{p1} (obtenido de la Tabla 2.3) y la Eq. 2.11a se obtiene el polo deseado. Con la Eq. 2.11b y 2.11c se obtienen los componentes pasivos del filtro, siendo $R = R_1 = R_2$

$$f_p = f_c f_{p1} \quad (2.11a)$$

$$f_p = \frac{1}{2\pi R \sqrt{C_1 C_2}} \quad (2.11b)$$

$$Q = 0,5 \sqrt{\frac{C_2}{C_1}} \quad (2.11c)$$

Para implementar un LPF de mayor O_{LPF} , se realiza a través de la conexión en cascada de LPF de $O_{LPF}=2$ y con los valores de la Tabla. 2.3. Por ejemplo, si se requiere un LPF de $O_{LPF}=4$, la primera etapa tendrá los valores normalizados de $Q=0.52$ y $f_p=1.432$, mientras que la segunda etapa tendrá los valores $Q=0.81$ y $f_p=1.606$.

2.2.5. Problemas prácticos de implementación de filtros

En las secciones anteriores se explicó teóricamente el comportamiento de los filtros con valores y elementos ideales. Una vez se requiera implementar los filtros, se debe tener en cuenta las posibles fuentes de error [71].

Los componentes pasivos son la principal fuente de errores por varias razones. La primera es que, al realizar los cálculos matemáticos del diseño, se obtienen valores de elementos que no están disponibles comercialmente. La forma de solventar este inconveniente está en la combinación de elementos en serie o paralelo según sea necesario. Ahora bien, esto último incrementaría el coste total del diseño, además de incrementar el área del circuito.

La segunda fuente de error está en la tolerancia de los elementos pasivos. Es recomendable el uso de elementos con una tolerancia baja, de modo que la respuesta sea lo más cercana a la ideal.

La tercera fuente de error son los efectos de la temperatura. En la Fig. 2.19 se puede apreciar cómo afecta la temperatura al valor nominal del condensador.

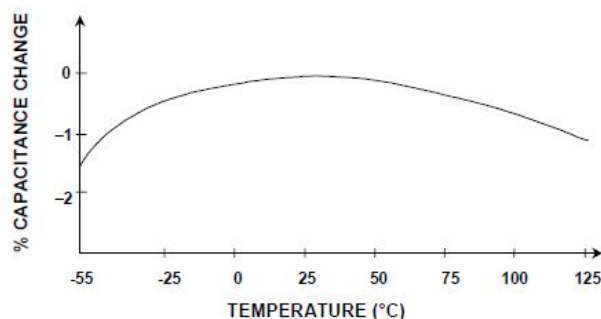


Figura 2.19: Efecto de la temperatura en un condensador de "thick film"

En cuanto a los valores de los condensadores se debe tener en cuenta que el valor nominal no debe ser menor a 10 pF o mayor a $10\text{ }\mu\text{F}$ [65]. En la tabla 2.4 se muestran las tecnologías de condensadores más comunes con sus ventajas, desventajas y su absorción dieléctrica (AD). La AD es la cantidad de energía indeseada que queda almacenada después de la descarga del condensador. Teniendo en cuenta lo anterior y la Tabla. 2.4, el condensador que más se adapta a la aplicación es el NP0 cerámico.

Tipo	AD	Ventajas	Desventajas
Poliestireno	0.001 % a 0.02 %	Económico Buena estabilidad	Gran tamaño Alta inductancia Baja disponibilidad
Polipropileno	0.001 % a 0.02 %	Económico Buena estabilidad	Gran tamaño Alta inductancia
NP0 cerámico	<0.1 %	Económico Tamaño pequeño Buena estabilidad	Valor máximo (10 nF)
Electrolítico	Muy alta	Rango de valores Altas corrientes Altas tensiones	Fugas altas Polarización Poca estabilidad Poca precisión

Tabla 2.4: Comparativa de diferentes tipos de condensadores.

Los valores de las resistencias también deben escogerse con cuidado, ya que si se utilizan valores menores a 100Ω aumentaría la corriente y se disiparía más potencia. Por otra parte, con valores mayores a $1 M\Omega$ es más propenso a capacidades parásitas [72].

El AO del filtro Sallen-Key es considerado ideal al diseñar el filtro. Tiene ganancia infinita, impedancia de salida cero e impedancia de entrada infinita [73]. La característica más importante a tener en cuenta es su limitación de ancho de banda.

Los AO incorporan de fabrica un condensador para evitar las oscilaciones indeseadas en la salida [74]. Lo anterior genera que al haber retroalimentación negativa la respuesta del AO sea un LPF de $O_{LPF}=1$.

En [75] se dan unos parámetros de diseño que permiten escoger correctamente el AO dependiendo del diseño. En primer lugar, se debe escoger la topología de la implementación del LPF, en este caso Sallen-Key. Además, se debe conocer f_C y la ganancia el LPF, en este caso ganancia unitaria. Por último, es necesario Q del LPF. Si $Q>1$ se utiliza la Eq. 2.12a, de lo contrario la Eq. 2.12b.

El GBP (gain bandwidth product) es el producto del ancho de banda del AO y la ganancia a la cual amplificará el mismo. El valor encontrado en la Eq. 2.12a y 2.12b es el GBP mínimo que debe tener el AO para la implementación del LPF.

$$GBP = 100 \cdot Gain \cdot f_C * Q^3 \quad (2.12a)$$

$$GBP = 100 \cdot Gain \cdot f_C \quad (2.12b)$$

Los AO que están diseñados como LPF de $O_{LPF}=1$, tienen GBP independiente de la ganancia, por lo que su GBP es igual al ancho de banda del propio amplificador [76].

2.2.6. Análisis en frecuencia del PWM

De manera ideal cuando la salida del PWM se conecta al LPF, se espera obtener una señal continua a la salida del LPF. Sin embargo, lo que se obtiene es un ΔV_{LPF} superpuesto a una señal continua, como se muestra en la Fig. 2.20. Para analizar dicho comportamiento, se hace a través de la transformada de Fourier.

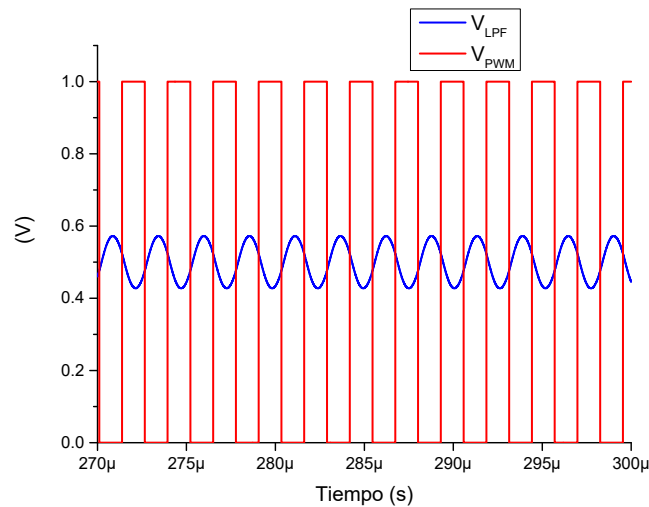


Figura 2.20: V_{PWM} y V_{LPF} utilizando un LPF Bessel con $f_C=200$ kHz, $O_{LPF}=3$, $f_{CLK}=100$ MHz, PWM de 8-bit y $D = \frac{128}{256}$

La serie de Fourier permite representar cualquier señal periódica como una serie de sumas de armónicos a diferentes frecuencias de la forma mostrada en la Eq. 2.13. El PWM se analiza como una señal cuadrada periódica, siendo A_0 la componente continua de la señal, A_n y B_n son los coeficientes de la serie de Fourier [77, 78]. Los coeficientes son hallados

con la Eq. 2.14 y la representación de la serie Fourier de la señal se muestra en la Eq. 2.15, donde D varía entre 0 y 1, siendo ΔV_O la amplitud de la señal del PWM.

$$f(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi t}{T}\right) + B_n \sin\left(\frac{2n\pi t}{T}\right) \quad (2.13)$$

$$\begin{aligned} A_0 &= \frac{1}{2T} \int_{-T}^T f(t) dt \\ B_n &= \frac{1}{T} \int_{-T}^T f(t) \cdot \sin\left(\frac{2n\pi t}{T}\right) \cdot dt \\ A_n &= \frac{1}{T} \int_{-T}^T f(t) \cdot \cos\left(\frac{2n\pi t}{T}\right) \cdot dt \end{aligned} \quad (2.14)$$

$$\begin{aligned} A_0 &= D \cdot \Delta V_O = \frac{d}{2^N} \cdot \Delta V_O \\ B_n &= 0 \\ A_n &= 2 \cdot \frac{\Delta V_O}{n\pi} \left[\sin(n\pi D) \right] \end{aligned} \quad (2.15)$$

Teniendo en cuenta que $\Delta V_O = V_{OH} - V_{OL}$, se obtiene que el valor medio de V_{LPF} es $\Delta V_O \cdot D$, con un rizado representado por los armónicos de orden superior a 1.

A partir de los componentes armónicos de la representación de Fourier del PWM ($g_n(t)$) mostrada en la Eq. 2.16, se pretende obtener el D que produce el máximo ΔV_{LPF} .

$$g_n(t) = A_n \cdot \cos\left(\frac{2n\pi t}{T}\right) \quad (2.16)$$

La energía contenida en una señal coseno es proporcional al cuadrado de la señal [79]. Por lo tanto, para obtener la máxima energía, se deriva parcialmente $g_n(t)$ con respecto a D y se iguala a 0 como se muestra en las Eq. 2.17 y 2.18.

$$\frac{\partial}{\partial D} (g_n^2(t)) = 0 \quad (2.17)$$

$$2 \cdot A_n \cos^2\left(\frac{2n\pi t}{T}\right) \cdot \frac{\partial(A_n)}{\partial D} = 0 \quad (2.18)$$

El término coseno de la Eq. 2.18 no depende de D , por lo que no se considera en la

igualdad. Este procedimiento se realiza para el primer armónico, ya que es el que más afecta al ΔV_{LPF} [80].

Primero se iguala el término $2 \cdot A_n$, como se muestra en la Eq. 2.19 y se obtiene $D=[0,1]$. Cuando $D = 0$ y $D = 1$ el V_{PWM} es constante en V_{OL} y V_{OH} respectivamente, por lo que no se genera ΔV_{LPF} .

$$2 \cdot A_n = 0 \quad (2.19)$$

$$2 \cdot \Delta V_O \frac{1}{n\pi} \cdot [\sin(n\pi D) - \sin(2n\pi(1 - \frac{D}{2}))] = 0$$

El siguiente término para evaluar es la derivada parcial de A_n con respecto a D . En la Eq. 2.20, se obtiene que $D = \frac{1}{2}$ es el ciclo de trabajo donde se produce la máxima energía. Este resultado indica que si se cumple la condición de la Eq. 2.6 para un $D = \frac{1}{2}$, lo será para el resto de D .

$$\frac{\partial}{\partial D}(A_n) = 0 \quad (2.20)$$

$$\cos(n\pi D) + \cos(2n\pi - n\pi D) = 0$$

Al evaluar $D = \frac{1}{2}$ en la Eq. 2.15, se obtiene la Eq. 2.21, donde $A_{1(max)}$ es la amplitud máxima generada por el primer armónico, el cual es el armónico más significativo para ΔV_{LPF} .

$$A_{1(max)} = 2 \cdot \frac{\Delta V_O}{\pi} \left[\sin\left(\frac{\pi}{2}\right) \right] = 2 \cdot \frac{\Delta V_O}{\pi} \quad (2.21)$$

El ΔV_{LPF} generado a la salida del LPF, se obtiene filtrando el primer armónico mostrado en la Eq. 2.22 a partir de la Eq. 2.21, donde A_{LPF} representa la ganancia en magnitud de f_{PWM} . Por lo tanto, utilizando la Eq. 2.6 y Eq. 2.21, se genera en la Eq. 2.23 el límite para la ganancia de la magnitud del LPF a f_{PWM} .

$$\Delta V_{LPF} > A_{LPF}(f_{PWM}) \cdot 4 \left(\frac{\Delta V_O}{\pi} \right) \quad (2.22)$$

$$A_{LPF}(max) < \frac{\pi}{4} \cdot \frac{1}{2^N} \quad (2.23)$$

2.2.7. Obtención de los parámetros del LPF para un ADC de N-bits basado en SAR

Después de estudiar los efectos de los parámetros que afectan el comportamiento del LPF, se busca un método para obtener f_C y el O_{LPF} , a partir de las características del ADC. Se presenta en la Fig. 2.21 el esquemático de la respuesta en frecuencia del LPF, indicando los parámetros necesarios para encontrar f_C .

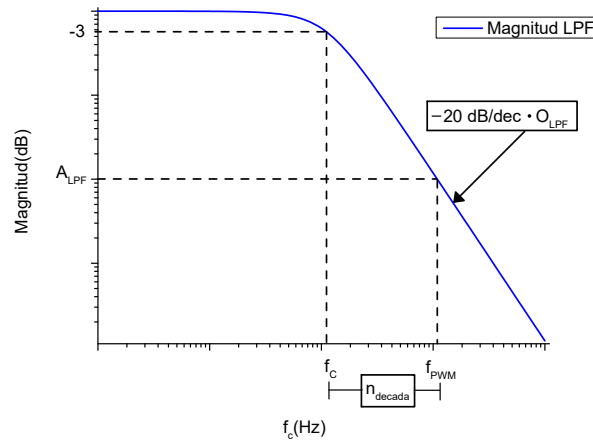


Figura 2.21: Esquemático de la magnitud de la respuesta en frecuencia del LPF

En primer lugar, con la Eq. 2.24 se computa el límite máximo para la ganancia del primer armónico del PWM, a partir de la Eq. 2.23 en escala logarítmica. En la Eq. 2.25 se presenta la salida del PWM filtrada con respecto a f_{clk} , f_C y O_{LPF} .

$$A_{LPF(max)} = 20 \cdot \log_{10} \left(\frac{\pi}{4} \cdot \frac{1}{2^N} \right) \quad (2.24)$$

$$A_{LPF}(f_{PWM}) = -20 \cdot O_{LPF} \cdot \log_{10} \left(\frac{f_{PWM}}{f_C} \right) = -20 \cdot O_{LPF} \cdot \log_{10} \left(\frac{f_{clk}}{2^N \cdot f_C} \right) \quad (2.25)$$

Para que ΔV_{LPF} sea menor a LSB se debe cumplir que $A_{LPF}(f_{PWM}) < A_{LPF(max)}$. Se combinan la Eq. 2.24 y 2.25 para generar la Eq. 2.26. Esta ecuación permite encontrar una serie f_C y O_{LPF} que satisfaga la atenuación requerida. El $t_{Settling}$ depende de f_C y O_{LPF} , por lo tanto, la solución deseada es que, con el mínimo O_{LPF} se encuentre una f_C que la $f_{Sampling}$ requiera.

$$-O_{LPF} \cdot \log_{10} \left(\frac{f_{clk}}{2^N \cdot f_C} \right) < \log_{10} \left(\frac{\pi}{4} \cdot \frac{1}{2^N} \right) \quad (2.26)$$

$$\left(\frac{f_C}{f_{clk}} \right) < \frac{O_{LPF} \sqrt{\pi}}{2^{\left(\frac{2+N+O_{LPF}}{O_{LPF}} \right)}}$$

Para ejemplificar el procedimiento, se diseña el LPF para un ADC de 6-bit, con los parámetros de la FPGA en $f_{clk}=100$ MHz y $\Delta V_O=1$ V. Se obtiene $LSB=15.625$ mV a partir de la Eq. 2.6, $f_{PWM}=1.56$ MHz a partir de la Eq. 2.7 y $A_{LPF}=-38.22$ dB aplicando la Eq. 2.24. A partir de la Eq. 2.26 se obtiene f_C para diferentes O_{LPF} y se muestran en la Tabla 2.5. Además, se muestra la respuesta en frecuencia de los LPF hallados en la Fig. 2.22, donde es posible observar la atenuación en decibeles del LPF para f_{PWM} .

O_{LPF}	f_C	$t_{Settling}$ del LPF.	$t_{Conversion}$ del ADC	$f_{Sampling}$
$O_{LPF}=2$	173.091 kHz	3.75 μ s	22.53 μ s	44.4 kSPS
$O_{LPF}=3$	360.404 kHz	2.77 μ s	16.65 μ s	60.1 kSPS
$O_{LPF}=4$	520.052 kHz	2.40 μ s	14.42 μ s	69.3 kSPS

Tabla 2.5: O_{LPF} , f_C , $t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 6-bit

En la Fig. 2.22 se muestra la respuesta en frecuencia de los LPFs con las características de la Tabla 2.5, donde se aprecia que la intersección de los LPFs, es el A_{LPF} hallado anteriormente.

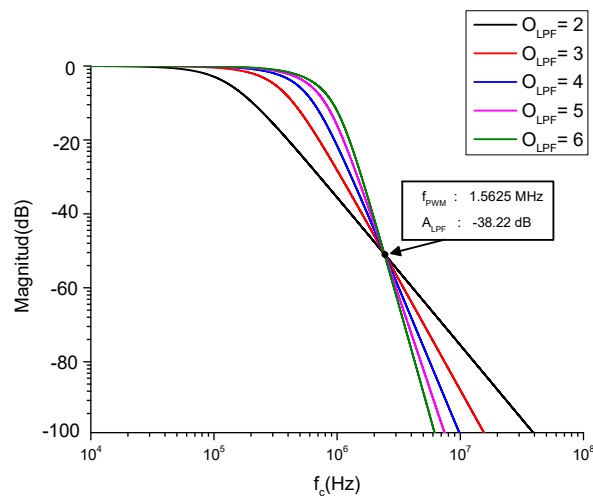


Figura 2.22: Magnitud de la respuesta en frecuencia del LPF con diferentes O_{LPF}

Con los parámetros de los LPFs de la Tabla 2.5 se obtiene las respuestas al escalón y se muestran en la Fig. 2.23. Se puede apreciar que ΔV_{LPF} es igual para cada LPF, debido a que se trabaja con la misma A_{LPF} .

Como se explicó anteriormente, si se mantiene el valor de f_C y se aumenta O_{LPF} el $t_{Settling}$ aumenta. Pero de acuerdo a la Fig. 2.23 a medida que O_{LPF} aumenta, es posible obtener una f_C que satisfaga la Eq. 2.6 y genere un menor $t_{Settling}$. Lo anterior es debido a que el incremento del $t_{Settling}$ al aumentar O_{LPF} , se ve compensado y mejorado por el aumento de la f_C , obteniéndose un menor $t_{Settling}$. El menor $t_{Settling}$ de la Fig. 2.23 es el LPF con $O_{LPF}=4$ y $f_C=520.052$ kHz.

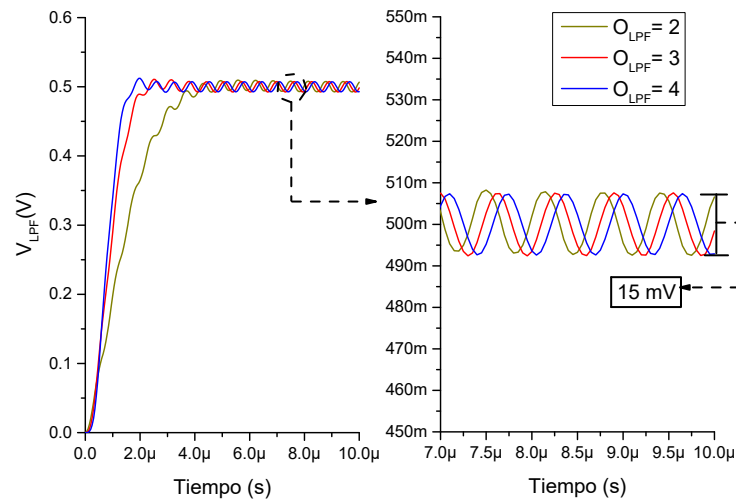


Figura 2.23: Respuesta al escalón de los LPFs con f_C y O_{LPF} hallados

Lo anterior permite gran versatilidad en la implementación. Si se requiere una mayor $f_{Sampling}$ del ADC, se implementa el LPF con un mayor O_{LPF} , que a su vez incrementaría el coste. Por el contrario, si la aplicación no requiere una $f_{Sampling}$ alta, podría usarse una implementación más económica.

Otro punto a tener en cuenta es la fase del LPF. En la Fig. 2.24 se aprecia como varía la fase de la respuesta. Para el $O_{LPF}=2$ hay un desfase respecto al PWM de aproximadamente de 180° , 270° para el $O_{LPF}=3$ y 360° para el $O_{LPF}=4$. La variación en la fase de la señal no afecta a la hora de implementar el DAC con un PWM y un LPF.

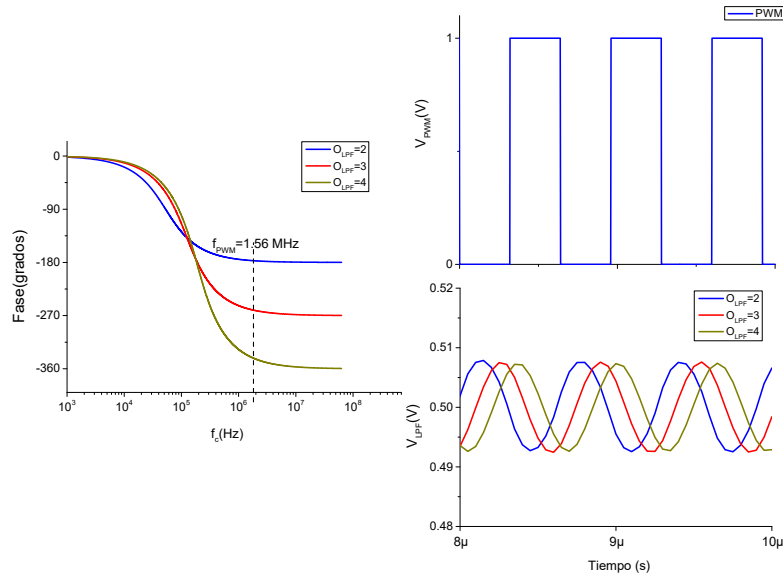


Figura 2.24: Fase y respuesta al escalón de los diferentes LPF con f_C y O_{LPF} hallados

2.2.8. Rizado en estado estacionario

Una vez obtenido el D crítico, se procede a encontrar f_C y O_{LPF} que cumpla con la condición de la Eq. 2.6, con el menor $t_{Settling}$ posible. Para esto se usará el método explicado en la sección 2.2.7. Una vez se obtenga f_C y O_{LPF} , se debe comprobar el ΔV_{DAC} para el D crítico y de ser necesario realizar ajustes para cumplir con la condición de la Eq. 2.6.

Cabe recordar que al aumentar f_C el $t_{Settling}$ del LPF es menor, pero presenta un mayor ΔV_{DAC} . También se debe tener en cuenta que al aumentar el O_{LPF} ΔV_{DAC} disminuye, pero el $t_{Settling}$ es mayor y el coste aumenta. En caso de una mala selección de f_C , se presentarían errores como el que se muestra en la Fig. 2.25, donde no se cumple la Eq. 2.6.

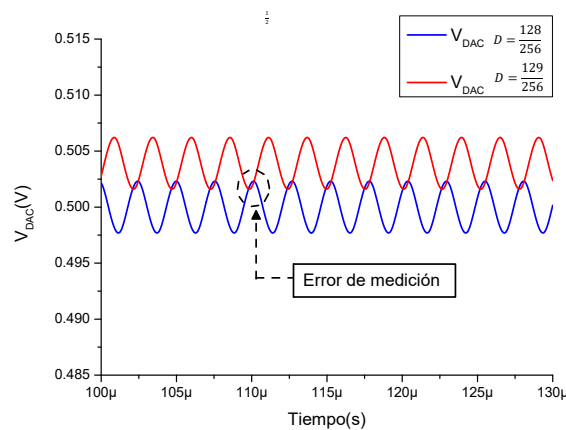


Figura 2.25: V_{DAC} de $f_C=60$ kHz, $O_{LPF}=3$ con ciclos de trabajo de $\frac{128}{256}$ y $\frac{129}{256}$

En la Fig. 2.26 se escogen dos f_C que cumplen con la condición de la Eq.2.6, para un ADC de 8-bit con $\Delta V_O=1$ V, $LSB=3.906$ mV. La diferencia radica en que se escogió una f_C que genera un ΔV_{DAC} menor, pero que produce un $t_{Settling}$ más lento, es decir existe una relación de compromiso entre reducir ΔV_{DAC} y reducir $t_{Settling}$. A la hora de diseñar el ADC se debe tener en cuenta lo anterior, donde debe existir un equilibrio entre el $t_{Settling}$ y ΔV_{DAC} , para encontrar un O_{LPF} y f_C que cumplan con la condición de la Eq. 2.6 y permita mayor $f_{Sampling}$.

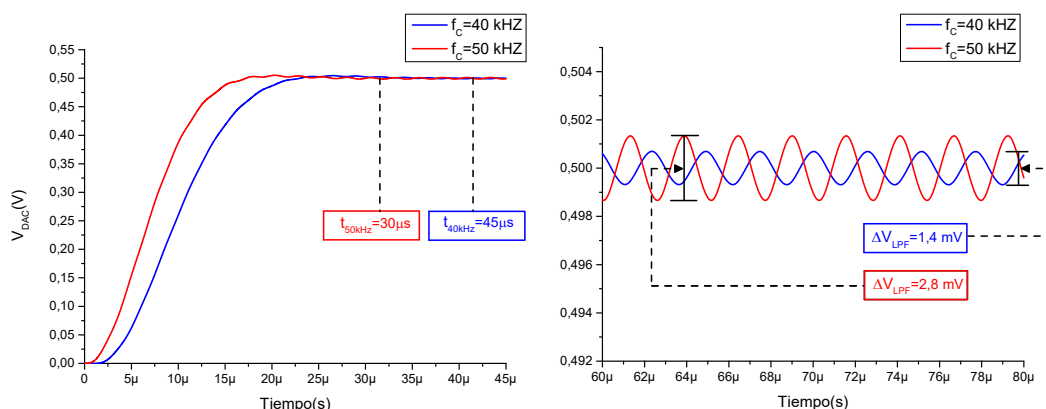


Figura 2.26: V_{DAC} con LPF de $O_{LPF}=3$ y con $f_C=40$ kHz y $f_C=50$ kHz, el PWM con $D = \frac{128}{256}$

2.3. ADC de 8-bit basado en SAR con PWM y LPF

Para validar el método de obtención de los parámetros del LPF para implementar el ADC en una FPGA, se utilizará una FPGA de la familia Zynq 7000 de Xilinx, con $f_{clk}=100$ MHz y $\Delta V_O=3.3$ V. Se pretende diseñar un ADC con una $f_{sampling}=8$ KSPS a modo de ejemplo. El ADC puede ser implementado en FPGAs de diferentes familias y diferentes características, siguiendo el mismo procedimiento. Teniendo en cuenta lo anterior se propone diseñar un ADC basado en SAR donde el DAC sea implementado con PWM y LPF.

Esta implementación se realizará con un ADC de 8-bit como primera aproximación. Para la implementación se deben tener claros los parámetros del ADC y la FPGA. En este caso, los parámetros mostrados en la Tabla. 2.6. El ΔV_O y f_{clk} dependen de la FPGA, mientras que f_{PWM} depende del número de bits y de la f_{clk} .

Parámetros	Valor
ΔV_O	3.3 V
N	8
LSB	12.89 mV
$f_{Sampling}$	8 KSPS
f_{clk}	100 MHz
f_{PWM}	390.625 kHz

Tabla 2.6: Parámetros para la implementación del ADC de 8-bit .

Una vez se tengan claras las características del ADC y con el método explicado en la sección 2.2.7 se obtiene f_C para diferentes O_{LPF} . Los resultados se muestran en la Tabla 2.7. En la Fig. 2.27a se puede observar gráficamente el comportamiento de los LPF hallados, donde se muestra como varía el $t_{Settling}$ con respecto a f_C para cada O_{LPF} . Se observa como al aumentar el O_{LPF} la f_C hallada también aumenta, por lo que $t_{Settling}$ disminuye. Los puntos seleccionados en la Fig. 2.27a para cada O_{LPF} indican la f_C máxima hallada, por lo que también es posible escoger f_C menores, pues siguen cumpliendo con la Eq. 2.6.

O_{LPF}	f_C	$t_{Settling}$ del LPF.	$t_{Conversion}$ del ADC	$f_{Sampling}$
2	21.64 kHz	30.04 μs	240.3 μs	4.16 kSPS
3	56.76 kHz	17.62 μs	140.9 μs	7.10 kSPS
4	91.93 kHz	13.60 μs	108.8 μs	9.19 kSPS
5	122.8 KHz	12.38 μs	99.04 μs	10.1 kSPS
6	148.9 kHz	11.75 μs	94.02 μs	10.6 kSPS

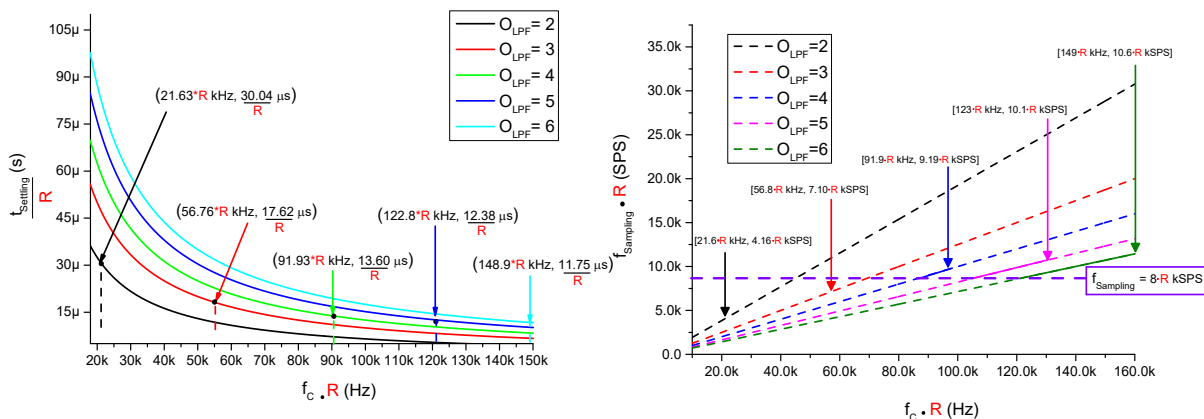
Tabla 2.7: O_{LPF} , f_C , $t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 8-bit.

Ahora bien, otra forma de presentar los resultados se muestra en la Fig. 2.27b, donde se muestra directamente la $f_{Sampling}$ obtenida para cada f_C con diferentes O_{LPF} . De esta forma se obtiene gráficamente las posibles soluciones para la implementación del ADC. La línea morada indica el punto en donde la $f_{Sampling}$ es igual a la deseada, por encima de esta línea también sería posible su implementación. Por otro lado, también hay una $f_{Sampling}$ máxima que se puede alcanzar con cada O_{LPF} , esta viene limitada por el LSB . Teniendo en cuenta lo anterior en cada O_{LPF} se señala este límite máximo.

Uno de los objetivos es obtener un método sistemático para implementar el ADC con un LPF y PWM en cualquier FPGA. Los resultados mostrados en al Fig. 2.27a y 2.27b están

normalizados para $f_{CLK}=100$ MHz, por lo que es posible obtener los parámetros para otras f_{CLK} multiplicando por un factor R, representado en la Eq. 2.27.

$$R = \frac{f_{clk}}{100 \text{ MHz}} \quad (2.27)$$



(a) f_C vs. $t_{Settling}$ para diferentes O_{LPF} (b) f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} normalizado con $O_{LPF}=100$ MHz. O_{LPF} normalizado con $O_{LPF}=100$ MHz.

Figura 2.27: Parámetros del LPF con diferentes O_{LPF} implementando un ADC de 8-bit con $f_{Sampling} = 8$ kSPS y normalizado a $f_{clk} = 100$ MHz.

Basándose en la Tabla. 2.7 se escoge $O_{LPF}=4$ y $f_C=91.93$ kHz para realizar la implementación. Esta combinación cumple con el $t_{Settling}$ necesario para obtener una $f_{Sampling}=8$ KSPS. También era posible escoger O_{LPF} más altos, pero el coste de implementación aumenta sin necesidad alguna. La implementación del LPF Bessel, se realizó por medio de la Tabla 2.3, la Tabla normalizada de Bessel, siendo el punto de partida para el diseño [81]. Después de realizar el cálculo de los elementos del LPF como se indica en la sección 2.2.4, se implementa el LPF de $O_{LPF}=4$ como se muestra en la Fig. 2.28, donde se implementa mediante dos LPF de $O_{LPF}=2$ conectados en cascada.

El amplificador operacional utilizado en la implementación de los LPFs es el LT1364 de Analog Devices. Este AO tiene un ancho de banda suficiente para el LPF implementado. Además, su slew rate es suficiente para generar la respuesta adecuada.

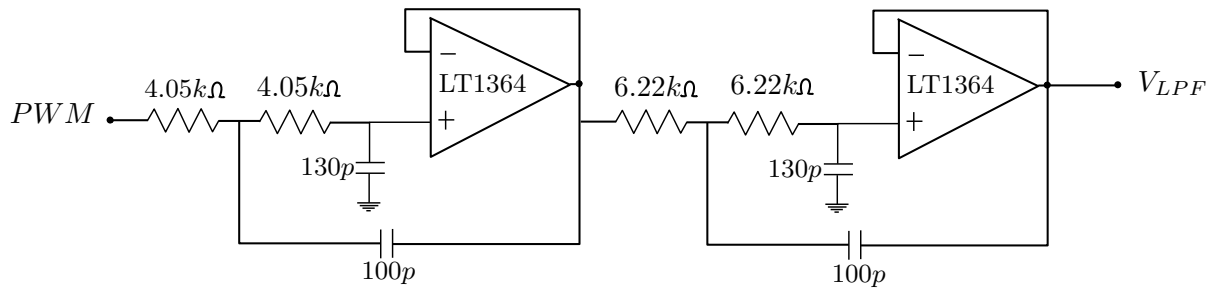


Figura 2.28: Implementación del LPF de $f_C=91.93$ kHz y $O_{LPF}=4$

Se realiza la simulación del ADC en Simulink-Matlab con la herramienta System Generator de Xilinx. Esta herramienta permite la simulación de los elementos implementados en VHDL como el contador, PWM y lógica SAR, además de los modelos Simulink-Matlab del LPF, sumador y el comparador analógico. En la Fig. 2.29 se muestra el esquemático de la simulación del ADC. Como se explicó anteriormente, para obtener V_{DAC} se resta a V_{LPF} el equivalente a $\frac{1}{2} LSB$, de modo que el error de conversión pase de $\pm 1 LSB$, a ser $\pm \frac{1}{2} LSB$.

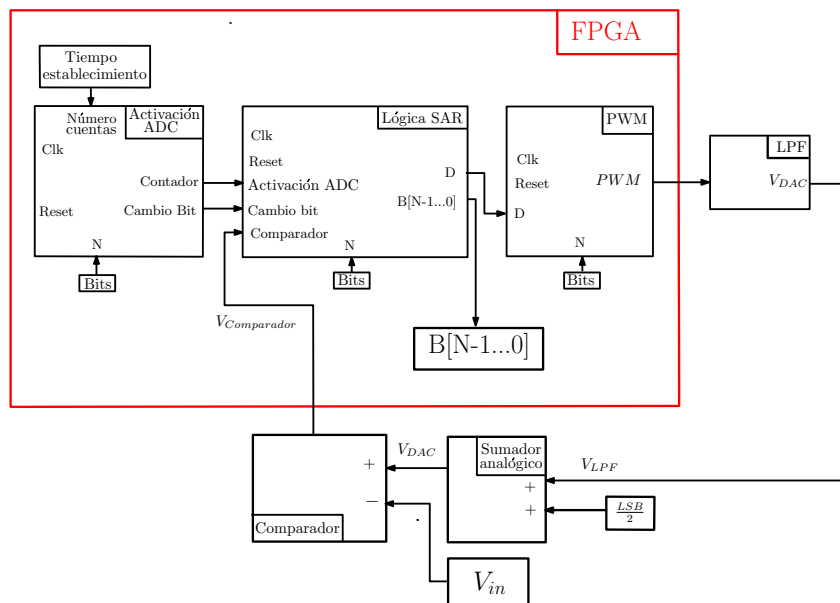


Figura 2.29: Esquemático de la simulación del ADC basado en SAR en una FPGA

El funcionamiento de los bloques implementados en la FPGA se explica a continuación. En primer lugar, el usuario indica el número de bits y el $t_{Settling}$ del LPF. Con esta información se configura el contador (Activación ADC), la lógica SAR y el PWM. El contador es el encargado de iniciar la conversión, además de generar un pulso cuando el $t_{Settling}$ del LPF haya pasado. El proceso es el siguiente, 'Activación ADC' inicia la conversión mediante

un pulso. Este primer pulso le indica a la lógica SAR que debe leer $V_{Comparador}$, donde se compara V_{DAC} y V_{in} . La lógica SAR basándose en $V_{Comparador}$ genera el d correspondiente y este es enviado al PWM. El PWM con el d recibido y el LPF generan V_{DAC} .

El contador inicia la cuenta nuevamente después de haber generado el primer pulso. Una vez termine la nueva cuenta, se genera un pulso nuevamente y se realiza todo el procedimiento explicado para la evaluación del siguiente bit. Cuando hayan sido evaluados todos los bits, la lógica SAR generará B , guardando el último valor de V_{DAC} .

En la Fig. 2.30 se observa el procedimiento para un ADC de 8-bit, donde el usuario indica el $t_{Settling}$ necesario para cada bit (15 μ s). En la sección 2.2 se explicó que V_{DAC} inicia con el bit más significativo en '1' y el resto en '0'. Para un rango de V_{in} del ADC entre 0 V y 3.3 V, se inicia en 1.65 V.

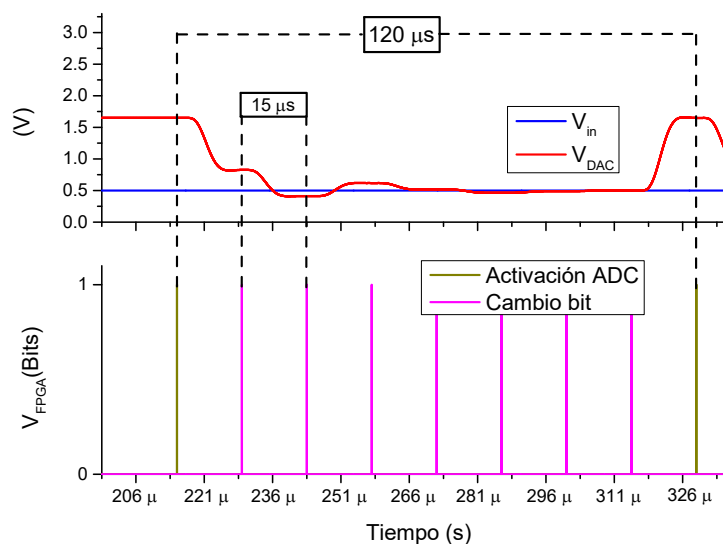


Figura 2.30: Comportamiento de V_{DAC} en el proceso de conversión del ADC

Para observar B para cada posible valor de V_{in} , se genera una rampa de valores de V_{in} desde 0 V hasta 3.3 V, con una pendiente tal que, incremente menos de un LSB en el tiempo de conversión del ADC ($t_{Conversion}$). En la Fig. 2.31 se observa como B es generada de manera correcta.

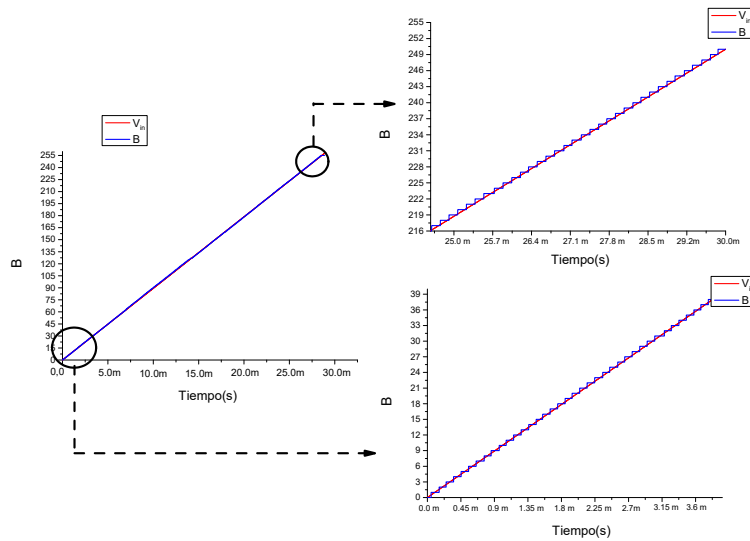
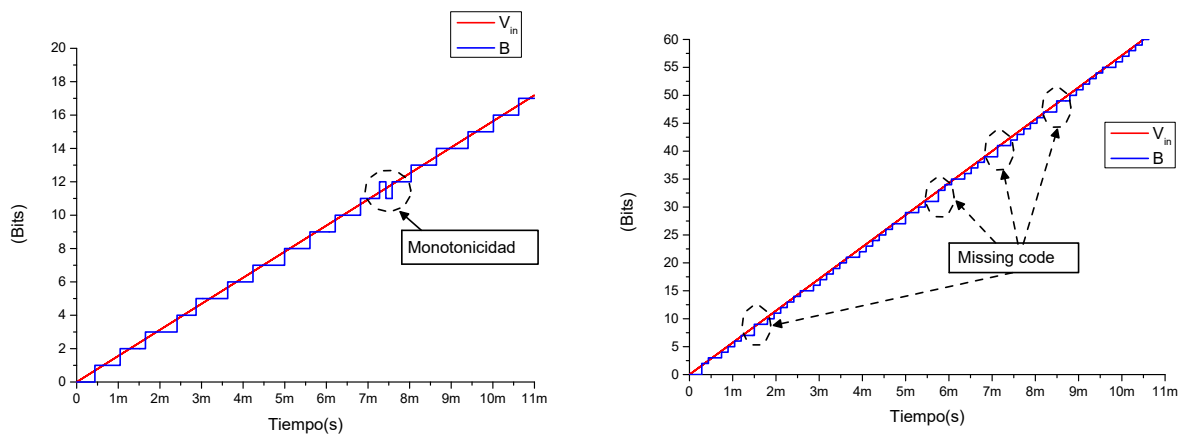


Figura 2.31: B obtenida a partir de V_{in}

En la Fig. 2.32a se muestra el error de monotonicidad generado por una mala selección de f_C . Se utilizó una f_C con un menor $t_{Settling}$ pero que no cumplió con la condición de la Eq. 2.6. El error de Missing code se aprecia en la Fig. 2.32b, el cual es generado por una selección errónea del $t_{Settling}$ del LPF. Es decir, se realizó la comparación entre V_{DAC} y V_{in} cuando V_{LPF} aún no estaba estabilizado.



(a) Error de monotonicidad causada por mala selección de $t_{Settling}$

(b) Error de missing code causada por mala selección de f_C

Figura 2.32: Errores causados por mala selección de parámetros

2.4. Síntesis del diseño del ADC basado en SAR

Una de las mayores ventajas del ADC basado con SAR, con el DAC implementado con PWM y LPF es su versatilidad. Si se requiere un ADC de N-bit, sólo es necesario unas modificaciones en el diseño para ser implementado. Para realizar el diseño de un ADC de N-bit, es necesario conocer ΔV_O y f_{clk} de la FPGA y realizar el procedimiento presentado a continuación. Hay que tener en cuenta que el diseño puede ser implementado en diferentes familias de FPGA con distintos valores de ΔV_O y f_{clk} .

1. En primer lugar, a partir de la $f_{Sampling}$, N y la Eq. 2.8 se obtiene $t_{Settling}$. Además, a partir de la Eq. 2.6 se obtiene LSB .
2. Una vez obtenidos LSB y $t_{Settling}$, el siguiente paso es encontrar f_C del LPF a partir de O_{LPF} y la Eq. 2.26, con el método presentado en la sección 2.2.7. Es posible hallar f_C para diferentes O_{LPF} , donde $t_{Settling}$ disminuye a medida que O_{LPF} aumenta. Como desventaja, al aumentar O_{LPF} el coste del diseño incrementa.
3. Con los f_C hallados a partir de los O_{LPF} , se debe encontrar el $t_{Settling}$ de cada LPF. Para esto se utiliza la gráfica normalizada de la respuesta al escalón de la Fig. 2.16, la Tabla 2.1 o f_C vs. $t_{Settling}$ de la Fig. 2.17. Una vez se encuentre los $t_{Settling}$ de los LPF hallados, se debe comparar con el $t_{Settling}$ necesario para cumplir con $f_{Sampling}$.
4. Una vez escogido que LPF adecuado, el siguiente paso es implementar V_{DAC} a partir de un sumador analógico y la Eq. 2.3. Esta tensión será comparada con V_{in} .
5. Por último se escogen los elementos para la implementación del sumador, el LPF y el comparador. Se deben tener en cuenta las limitaciones explicadas en la sección 2.2.5 al escoger el AO, las resistencias, condensadores y el tiempo de respuesta del comparador.
6. A medida que N aumenta, f_{PWM} y LSB disminuyen. Esto hace que f_C deba alejarse más respecto de f_{PWM} para poder cumplir con la condición de la Eq. 2.6, aumentando $t_{Settling}$ y disminuyendo $f_{Sampling}$. Esto hace que sea necesario aumentar O_{LPF} para encontrar una f_C que cumpla con $f_{Sampling}$ y $t_{Settling}$.

2.5. ADC de 10-bit basado en SAR con PWM y LPF

Para ejemplificar la versatilidad del diseño. se realiza el diseño de un ADC de 10-bit con una $f_{Sampling}=8$ KSPS, como el diseño realizado en la sección 2.3. Hay que tener en cuenta que, f_{PWM} ha disminuido en factor de 4 con respecto al ADC de 8-bit, debido a que f_{PWM} pasó de ser $\frac{f_{clk}}{28}$ a $\frac{f_{clk}}{210}$. Para la implementación del ADC de 10-bit, se seguirá el procedimiento explicado en la sección 2.4.

1. En primer lugar a partir de $f_{Sampling}$ y la Eq. 2.8 se obtiene $t_{Settling}$, además se obtiene LSB a partir de la Eq. 2.6.

$$LSB \leq 3,2227mV$$

$$t_{Settling} \leq 12,5\mu s$$

2. A continuación se halla f_C para diferentes O_{LPF} y se muestran en la Tabla 2.8.

O_{LPF}	f_C	$t_{Settling}$ del LPF.	$t_{Conversion}$ del ADC	$f_{Sampling}$
2	2.70 kHz	240 μs	2.40 ms	416 SPS
3	8.94 kHz	112 μs	1.12 ms	894 SPS
4	16.25 kHz	76.9 μs	769 μs	1.30 kSPS
5	23.26 KHz	65.3 μs	653 μs	1.53 kSPS
6	29.55 kHz	59.2 μs	592 μs	1.69 kSPS

Tabla 2.8: O_{LPF} , f_C , $t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 10-bit.

3. Como se puede apreciar en la Tabla 2.8 con los $O_{LPF}=2$ hasta $O_{LPF}=6$ no se pudo obtener un $t_{Settling}$ menor al necesario, por lo que se debe continuar aumentando el O_{LPF} con el objetivo de alcanzar los parámetros requeridos del LPF.

O_{LPF}	f_C	$t_{Settling}$ del LPF.	$t_{Conversion}$ del ADC	$f_{Sampling}$
7	35.048 kHz	54.2 μs	542 μs	1.84 kSPS
8	39.838 kHz	52.7 μs	527 μs	1.90 kSPS
9	44.011 kHz	50.2 μs	502 μs	1.99 kSPS
10	47.662 KHz	48.7 μs	487 μs	2.05 kSPS

Tabla 2.9: O_{LPF} , f_C , $t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$, para la implementación de un ADC de 10-bit.

4. Como se puede apreciar en la Tabla 2.9 con los $O_{LPF}=7$ hasta $O_{LPF}=10$ tampoco fue posible obtener el $t_{Settling}$ necesario. Es aquí donde el usuario debe escoger si se disminuye N o $f_{Sampling}$ para el ADC. También es posible modificar la forma de implementación del PWM, pero que será tratado en posteriores capítulos.
5. Se escogerá una $f_{Sampling}=1$ KSPS para ejemplificar el procedimiento de diseño e implementación, tanto del LPF como de la lógica SAR. En primer lugar, a partir de la Tabla 2.8 se obtiene la Fig. 2.33. Como se puede apreciar tanto graficamente como en la Tabla 2.8, solo es posible implementar el ADC de 10-bit con $O_{LPF}=4$ o mayor.

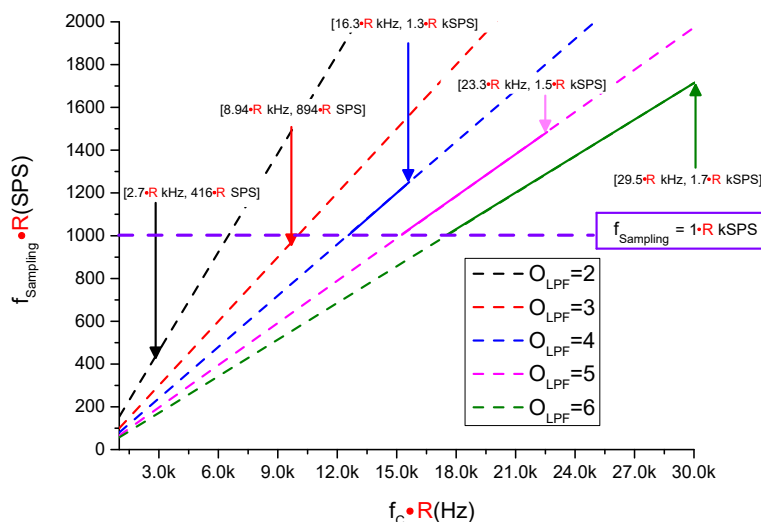


Figura 2.33: f_c vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 10-bit con $f_{Sampling}=1$ kSPS normalizado a $f_{CLK}=100$ MHz.

6. Se implementa el ADC con un LPF con $f_c=16.25$ kHz y $O_{LPF}=4$ y se simula su comportamiento en la Fig. 2.34. Cuando se inicia la conversión, con la señal Activación conversión, se realizan las 10 iteraciones mostradas en la Fig. 2.34 para obtener B .

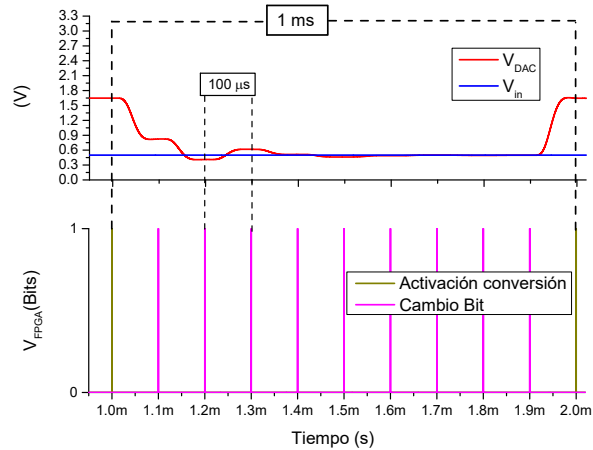


Figura 2.34: Comportamiento de V_{DAC} en el proceso de conversión del ADC

7. El LPF será implementado con la configuración Sallen-Key, utilizando 2 LPFs de $O_{LPF}=2$ en cascada para implementar el LPF con $O_{LPF}=4$, como se explicó anteriormente en la sección 2.2.4.
8. Por último, para obtener la Fig. 2.35, se almacena el último valor de V_{DAC} de cada conversión para obtener B . Una vez obtenido B , este es comparado con la entrada V_{in} . Se puede observar que la conversión es correcta para cada posible valor de B .

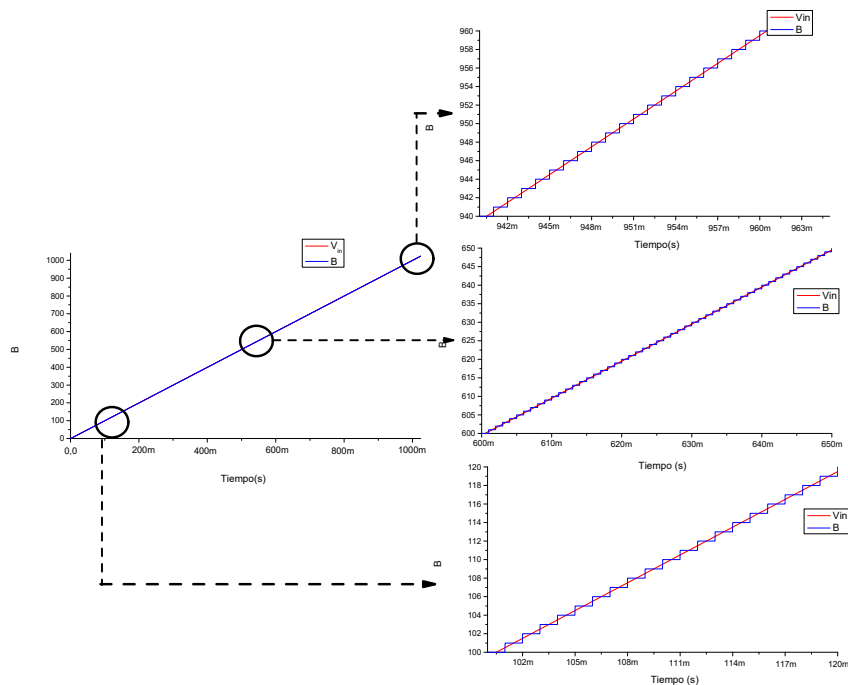


Figura 2.35: B obtenida a partir de V_{in}

2.6. Conclusiones

En este capítulo se analiza la implementación de un conversor analógico digital (ADC) basado en la lógica SAR, donde se reemplaza el conversor digital analógico (DAC) por la combinación de un modulador de anchura de pulsos (PWM) y un filtro pasa bajo (LPF). Se busca encontrar los límites del diseño, además de sus ventajas y desventajas.

Se estudió el efecto que tiene la frecuencia de corte (f_C) y el orden (O_{LPF}) en la respuesta al escalón del LPF. A mayor f_C el tiempo de establecimiento del LPF ($t_{Settling}$) es menor, pero el rizado (ΔV_{LPF}) es mayor. Esto indica que hay una relación de compromiso entre $t_{Settling}$ y cumplir con la condición de la Eq. 2.6. En el caso del O_{LPF} , a mayor O_{LPF} ΔV_{LPF} disminuye, pero el $t_{Settling}$ es mayor. Por último, a mayor O_{LPF} el coste de implementación aumenta.

Se estudia un método que permita encontrar f_C y O_{LPF} que cumplan con la condición de la Eq. 2.6. Para esto, se parte del análisis en frecuencia del PWM y el LPF con el objetivo de hallar una expresión generalizada, donde a partir de LSB, N, f_{CLK} y O_{LPF} se encuentre la f_C que cumpla con dicha condición.

Al variar O_{LPF} , la f_C del LPF varía con el objetivo de cumplir con la condición de la Eq. 2.6. Esto último, da la posibilidad de encontrar múltiples soluciones, que dan versatilidad al usuario. Si la aplicación requiere mayor $f_{Sampling}$ se escoge un O_{LPF} alto, que a su vez incrementa el coste de implementación. En caso contrario, si la aplicación permite $f_{Sampling}$ más pequeña, es posible disminuir costes de implementación.

Se presenta un método sistemático para la implementación de un ADC sobre FPGA para diferentes resoluciones basado en la lógica SAR, reemplazando el DAC por un PWM y un LPF. Esto fue posible gracias a la versatilidad que presenta el diseño. Para implementar un ADC de diferente resolución, se realizan cambios en el PWM, como también en f_C y O_{LPF} del LPF. El ADC puede ser implementado en una FPGA con distintos valores de ΔV_O y f_{clk} , sólo sería necesario realizar los cálculos explicados con estos nuevos parámetros.

3. Capítulo 3

3.1. Resumen

En este capítulo se estudia la forma de disminuir el rizado del convertor digital analógico (ΔV_{DAC}) aumentando la frecuencia del PWM (f_{PWM}), sacando provecho al "double data rate" (DDR) incorporado en la FPGA. De este modo es posible mejorar las prestaciones del convertor analógico digital (ADC) basado en successive approximation register (SAR) .

3.2. Diseño de un PWM por doble flanco de reloj

En el capítulo 2 se explicó el diseño de un PWM de N-bit regido por un contador ascendente. El contador en el capítulo 2 incrementa en 1 el registro de cuentas en cada flanco ascendente del reloj, por lo que la $f_{counter} = f_{clk}$. Se define D para cada periodo del PWM en la Eq. 2.2 del capítulo 2.

En [82, 83] se utiliza un sistema de adquisición de datos basado en 'double data rate' (DDR). Para esto se utiliza un registro dedicado 'Output DDR' (ODDR) de la FPGA para implementar salidas DDR [51, 84], disminuyendo el tiempo de propagación de la señal hasta la salida. Esto significa que los datos son adquiridos y procesados a mayor velocidad, por lo que es posible utilizar flip-flop de flanco ascendente y descendente del reloj para controlar la salida del PWM.

Teniendo en cuenta lo anterior, para implementar el PWM con DDR (PWM-DDR) se modifica el registro que lleva las cuentas del contador con el flanco ascendente y descendente del reloj, obteniéndose la $f_{counter} = 2 \cdot f_{clk}$. El requisito para que esto sea posible, es que la FPGA cuente con entradas y salidas DDR, como también flip-flops de flanco ascendente y descendente del reloj. Para obtener la frecuencia del PWM-DDR ($f_{PWM-DDR}$) se utiliza la Eq. 3.1.

$$f_{PWM-DDR} = \frac{2 \cdot f_{clk}}{2^N} \quad (3.1)$$

El contador, como se muestra en la Eq. 2.2, cuenta desde 0 hasta $2^N - 1$ y el registro de

las cuentas se lleva en un vector de N-bit de tipo standard logic. Para implementar el contador, el bit menos significativo será regido por los dos flancos del reloj, mientras el resto del vector será regido por el flanco ascendente del reloj. En la Fig. 3.1 se muestra el comportamiento del contador para un PWM-DDR de 3-bit, donde *LSB* es el bit menos significativo y *C_{MSB}* el contador con el resto de los bits.

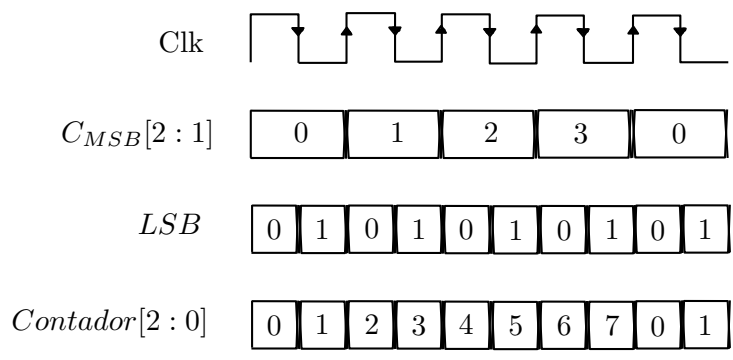


Figura 3.1: Contador de 3-bit implementado para el PWM-DDR

Al utilizando el PWM-DDR y el PWM en un mismo LPF, se puede apreciar que con el PWM-DDR el ΔV_{LPF} es menor. Esto es debido a que la f_C del LPF se aleja más de la $f_{PWM-DDR}$ que de la f_{PWM} , aumentando la atenuación del LPF. Para implementar el contador ascendente en la FPGA se realiza a través de un circuito sumador, una máquina de estados y flip-flops tanto ascendentes como descendentes. Al igual que con el contador ascendente implementado en el capítulo 2, el contador implementado con flip-flop de flanco ascendente y descendente de reloj, genera a la salida "1" de duración de $\frac{1}{f_{clk}}$ cuando $Contador = d - 1$. Se muestra el esquemático de la implementación de un contador de N-bit en la Fig. 3.2.

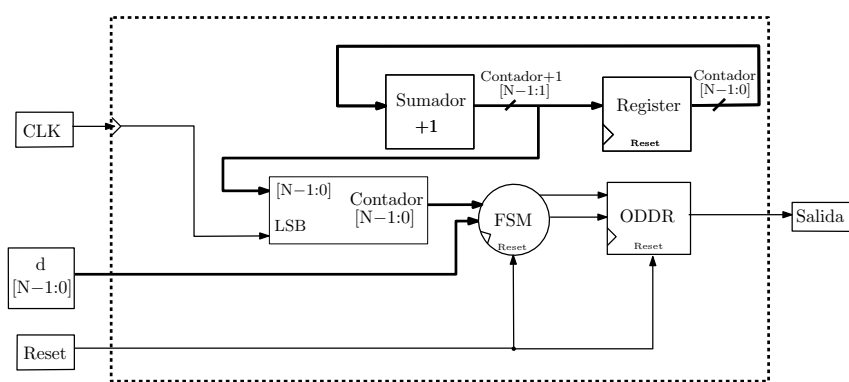


Figura 3.2: Esquemático de la implementación de un contador de N-bit

En la Fig. 3.3 se presenta V_{LPF} de un LPF con $f_C = 70$ kHz y $O_{LPF} = 4$ para un PWM con $f_{PWM} = 100$ kHz y un PWM-DDR con $f_{PWM-DDR} = 200$ kHz. Al obtener un menor ΔV_{LPF} con el PWM-DDR, es posible aumentar la f_C que cumpla con la condición de la Eq. 2.6 y con un $t_{Settling}$ menor, aumentando $f_{Sampling}$.

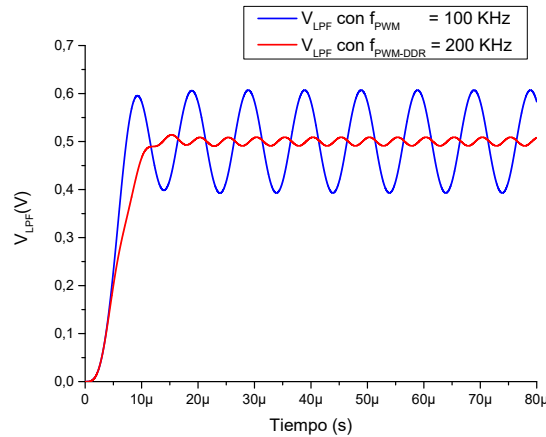


Figura 3.3: Comparación de los V_{LPF} con $f_C = 70$ kHz, $O_{LPF} = 4$ y $D = \frac{1}{2}$ con un PWM con $f_{PWM} = 100$ kHz y PWM-DDR con $f_{PWM} = 200$ kHz

3.3. Obtención parámetros del LPF

En el capítulo 2 se ejemplificó la obtención de los parámetros f_C y O_{LPF} para un ADC de 6-bit, teniendo $\Delta V_O = 1$ V y $f_{clk} = 100$ MHz en la FPGA. Se modifica la expresión 2.26 del capítulo 2 y se obtiene la Eq. 3.2.

$$-O_{LPF} \cdot \log_{10} \left(\frac{f_{PWM-DDR}}{f_C} \right) < \log_{10} \left(\frac{\pi}{4} \cdot \frac{1}{2^N} \right) \quad (3.2)$$

$$\left(\frac{f_C}{f_{clk}} \right) < \frac{2 \cdot O_{LPF} \sqrt{\pi}}{2^{\left(\frac{2+N+N \cdot O_{LPF}}{O_{LPF}} \right)}}$$

Se genera $f_{PWM-DDR} = 3.125$ MHz a partir de la Eq. 3.1 y $LSB = 15.625$ mV a partir de las Eq. 2.6 del capítulo 2, para luego obtener los parámetros del LPF con la Eq. 3.2. Para comparar con los valores hallados en la sección 2.2.7 del capítulo 2, se obtiene la Tabla. 3.1, donde se puede apreciar que para un mismo O_{LPF} , el aumento de $f_{PWM-DDR}$ con

respecto a f_{PWM} permite escoger una f_C con menor $t_{Settling}$. En la Fig. 3.4 se muestra la respuesta al escalón de los valores hallados y mostrado en la Tabla. 3.1. También se presenta en la Tabla 3.1 el número de resistencias, condensadores y AO necesarios para la implementación del LPF.

ADC=6-bit	PWM	PWM-DDR
$f_{PWM}/f_{PWM-DDR}$	1.56 MHz	3.125 MHz
f_C	520.05 kHz	1.040 MHz
O_{LPF}	4	
$t_{Settling}$	2.41 μ s	1.20 μ s
$f_{Sampling}$	69.3 kSPS	139 kSPS
R	8	
C	4	
AO	2	

Tabla 3.1: Comparación de las características de la implementación del LPF con PWM y PWM-DDR, para un ADC de 6-bit.

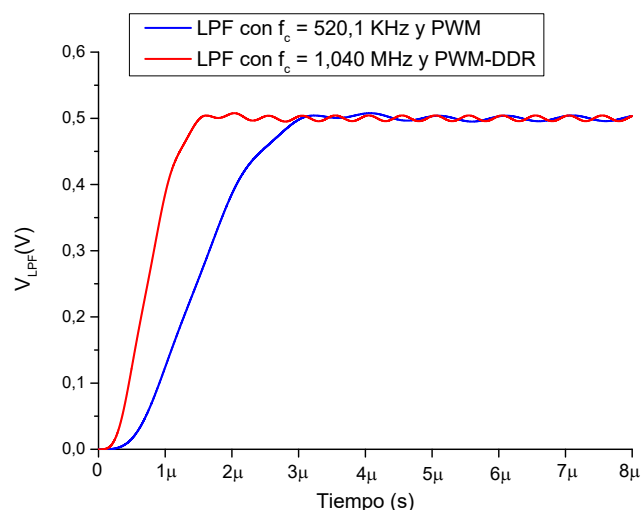


Figura 3.4: V_{LPF} de un PWM y PWM-DDR de 6-bit con los LPF de la Tabla 3.1

3.4. ADC de 8-bit basado en SAR con PWM-DDR y LPF

En el capítulo 2 se implementó un ADC de 8-bit con una $f_{Sampling}=8$ KSPS, con una FPGA de la familia Zynq 7000 de Xilinx, con $\Delta V_O=3.3$ V, $f_{clk}=100$ MHz, $f_{PWM}=390.625$ kHz y $LSB=12.89$ mV. Esta FPGA tiene incluido el registro DDR además de flip-flop de flanco ascendente y descendente [85], por lo que permite trabajar con el PWM-DDR. Se

obtiene $f_{PWM-DDR}=781.250$ kHz a partir de la Eq. 3.1, se hallan las f_C a partir distintos O_{LPF} y se muestran en la Tabla. 3.2.

O_{LPF}	f_C	$t_{Settling}$ del LPF.	$t_{Conversion}$ del ADC	$f_{Sampling}$
2	43.27 kHz	15.0 μ s	120.2 μ s	8.32 kSPS
3	113.52 kHz	8.81 μ s	70.47 μ s	14.2 kSPS
4	183.87 kHz	6.80 μ s	54.39 μ s	18.4 kSPS
5	245.56 KHz	6.19 μ s	49.52 μ s	20.2 kSPS
6	297.80 kHz	5.88 μ s	47.01 μ s	21.3 kSPS

Tabla 3.2: O_{LPF} , f_C , $t_{Settling}$ del LPF, $t_{Conversion}$ y $f_{Sampling}$ para un ADC de 8-bit.

A partir de la Eq. 2.8 se obtiene que $t_{Settling}=15$ μ s es el tiempo de establecimiento necesario para obtener $f_{Sampling}=8$ KSPS. Basándose en la Tabla 3.2, se deduce que a partir del $O_{LPF}=2$ es posible obtener dicho $t_{Settling}$. En la Fig. 3.5 se muestra la gráfica de f_C vs. $f_{Sampling}$, donde se puede ver el rango de f_C para la implementación para cada O_{LPF} para $f_{Sampling}=8$ KSPS.

Si se comparan los parámetros del LPF hallados para el PWM y PWM-DDR, se aprecia que para implementar el ADC de 8-bit con el PWM es necesario un $O_{LPF}=4$, mientras que para el PWM-DDR es necesario $O_{LPF}=2$. Esta disminución en el O_{LPF} se ve reflejada en menor coste y área de implementación.

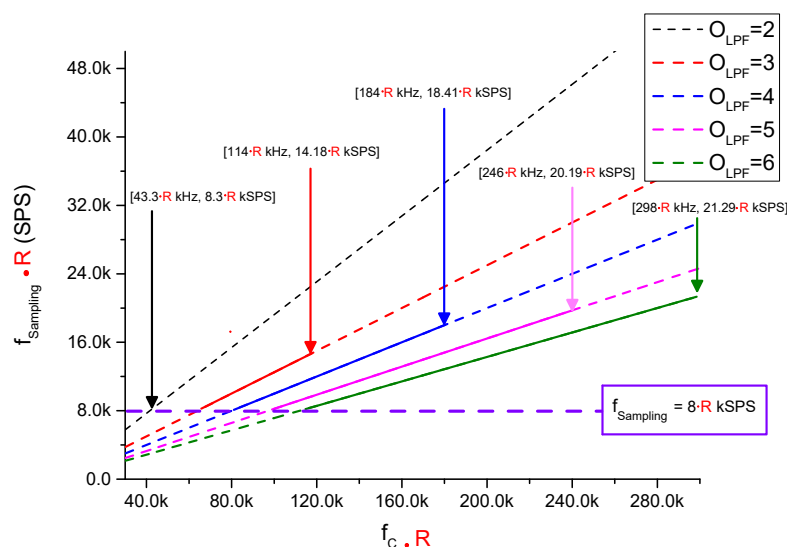


Figura 3.5: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=8$ kSPS con PWM-DDR normalizado a $f_{CLK}=100$ MHz.

Cabe destacar que a medida que O_{LPF} aumenta, el aumento de $f_{Sampling}$ no es lineal.

Es decir, entre los $O_{LPF}=2$ y $O_{LPF}=3$, $f_{Sampling}$ aumentó 70%, mientras que entre los $f_{Sampling}$ de los $O_{LPF}=5$ y $O_{LPF}=6$ el aumento fue inferior.

3.4.1. Simulación de un ADC de 8-bit basado en SAR con PWM-DDR y LPF

Una vez hallados los diferentes LPF que cumplan con la condición de la Eq. 2.6, el menor O_{LPF} con el que se cumple la condición de la Eq. 2.6 es $O_{LPF}=2$ y $f_C=43.27$ KHZ. La implementación del LPF con arquitectura Bessel, se ha realizado por medio de la tabla normalizada de Bessel, presentada en el capítulo 2.

El funcionamiento del sistema es el siguiente. En primer lugar, se configura el contador (Activación ADC), la lógica SAR y el PWM, para esto el usuario indica el número de bits y el $t_{Settling}$ del LPF.

La Fig. 3.7 muestra el comportamiento del V_{DAC} , donde una vez se inicia la conversión, 'Activación ADC' genera un pulso que le indica a la lógica SAR que debe leer $V_{Comparador}$. Basándose en $V_{Comparador}$ la lógica SAR indica el d correspondiente y este es enviado al PWM. Una vez d es enviado, el PWM y el LPF generan V_{LPF} , para luego obtener V_{DAC} , restando a V_{LPF} la tensión equivalente a $\frac{LSB}{2}$. Este proceso se realiza para cada bit del ADC, empezando por el MSB hasta el LSB . En la Fig. 3.6 se muestra el esquemático de la implementación del ADC.

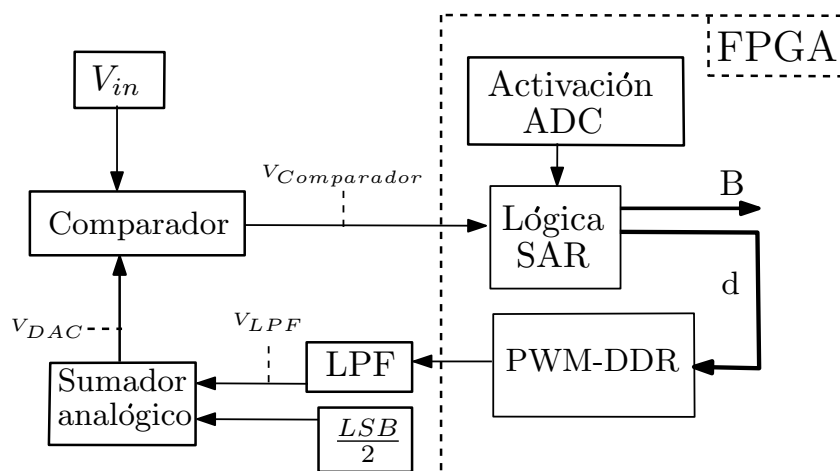


Figura 3.6: Esquemático de la implementación del en una FPGA con salida implementada con DDR

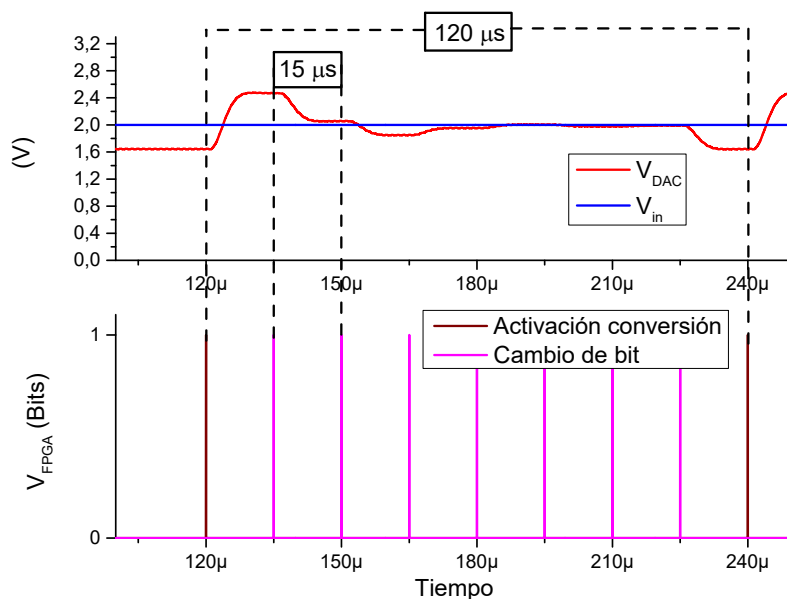


Figura 3.7: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit.

Una vez haya sido evaluado LSB , se almacena el valor de V_{DAC} para generar B correspondiente a la tensión analógica V_{in} . En la Fig. 3.8 se observa que B es generado correctamente con una entrada V_{in} de tipo rampa, donde se evalúa cada posible valor de V_{in} en el rango de entrada del ADC.

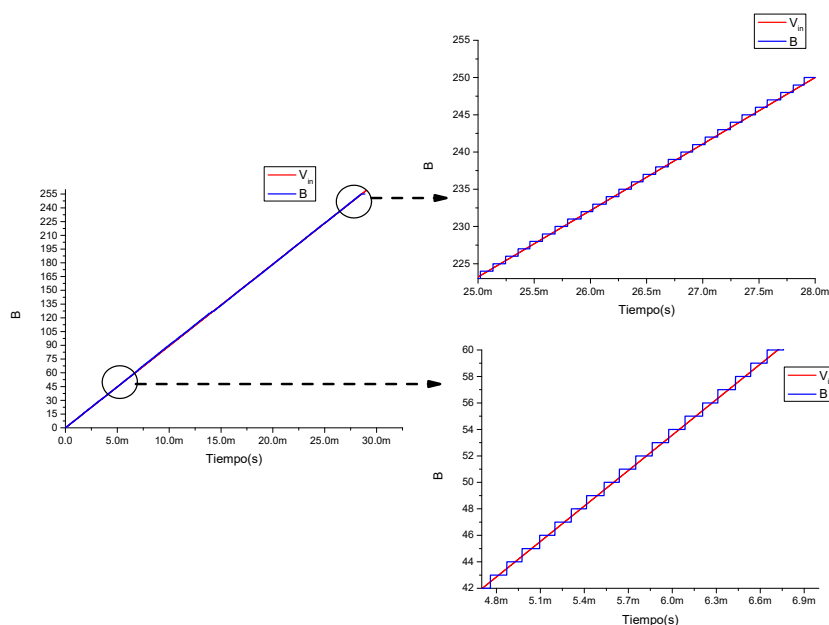


Figura 3.8: B obtenida a partir de V_{in} para un ADC de 8-bit

3.4.2. Implementación ADC de 8-bit basado en SAR con PWM-DDR y LPF

Para la implementación del ADC de 8-bit con $f_{Sampling}=8$ KSPS se utilizará la FPGA de la familia Zynq 7000, con $f_{clk}=100$ MHz, $\Delta V_O=3.3$ V y $LSB=12.89$ mV que se obtiene a partir de la Eq. 2.6 y se configura la salida de la FPGA con el registro DDR.

La implementación del LPF se realiza con el AO LT1364 de Analog Devices, cuyo ancho de banda es suficiente para la f_C , O_{LPF} y ganancia utilizados [86]. Por otra parte, su slew rate es suficiente para que se genere el V_{LPF} con el $t_{Settling}$ deseado. En la sección 2.2.5 del capítulo 2 se explican las limitaciones a tener en cuenta para escoger el AO.

Debido a que se está trabajando con señales sensibles al ruido, es necesario reducir al máximo las posibles fuentes de ruido, ya sea por cableado, pistas mal diseñadas, componentes susceptibles al ruido etc. Teniendo en cuenta lo anterior en la Fig. 3.9 se muestra la placa donde fue implementado el sistema, utilizando el menor camino posible para las señales, pistas suficientemente anchas, planos de masa, condensadores de desacople y conectores especiales para la placa de desarrollo ZedBoard Zynq 7000.

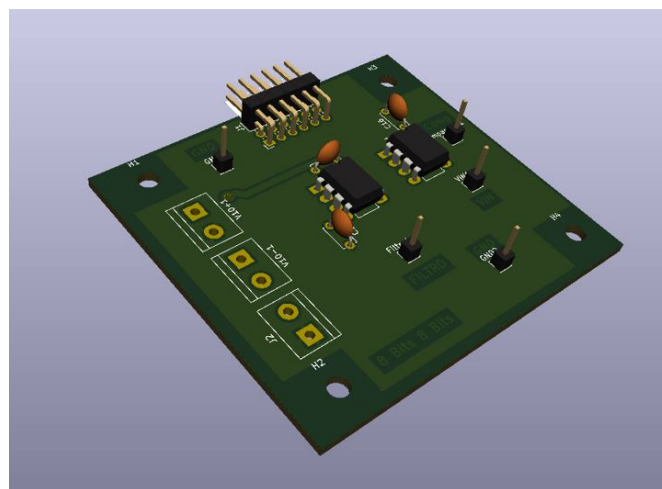
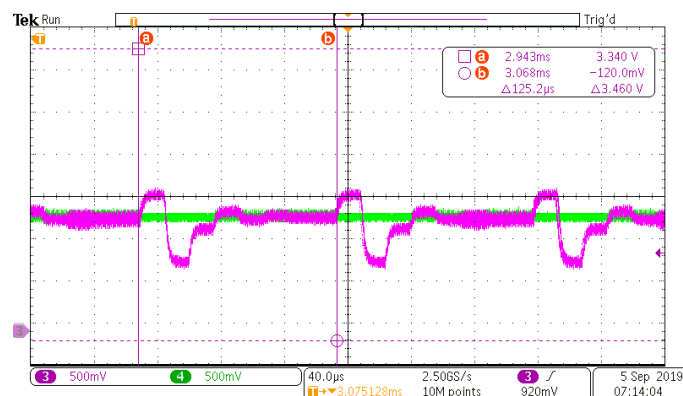
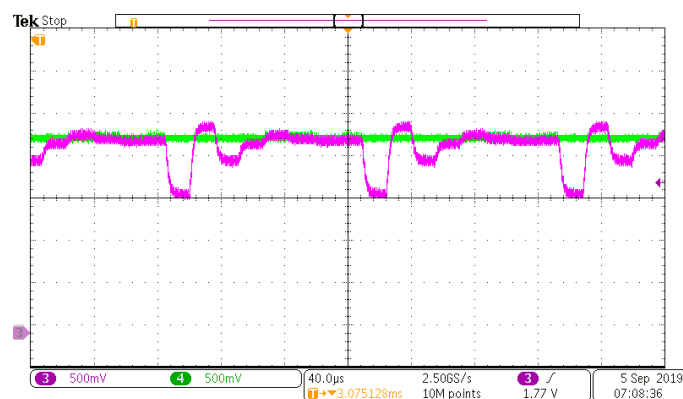


Figura 3.9: Placa de implementación del ADC de 8-bit

En primer lugar, se comprueba el comportamiento de V_{DAC} con un V_{in} continuo. En la Fig. 3.10a $V_{in}=1.2$ V y en la Fig. 3.10b $V_{in}=2.2$ V. La $f_{Sampling}$ del ADC es de 7.98 kSPS.

(a) V_{DAC} de un ADC de 8-bit para $V_{in}=1.2$ V(b) V_{DAC} de un ADC de 8-bit para $V_{in}=2.2$ V**Figura 3.10:** V_{DAC} de un ADC de 8-bit con V_{in} continuo

A continuación, se realiza la conversión analógica digital para una señal triangular de V_{in} con el rango del ADC, que permita obtener todos los posibles valores de B . Además, se introduce un DAC que permite visualizar B obtenido por el ADC. Se muestra el esquemático de la implementación en la Fig. 3.11. Una vez la lógica SAR indique que se ha obtenido B , este valor es enviado al 'DAC visualización' para poder ser mostrado en el osciloscopio.

En la Fig. 3.12 se muestra B (señal verde) resultante a partir de una entrada triangular de V_{in} (señal azul). V_{in} varía entre 0 V y 3.3 V. Es posible observar que B está desplazada ligeramente con respecto a V_{in} debido al $t_{Conversion}$ del ADC.

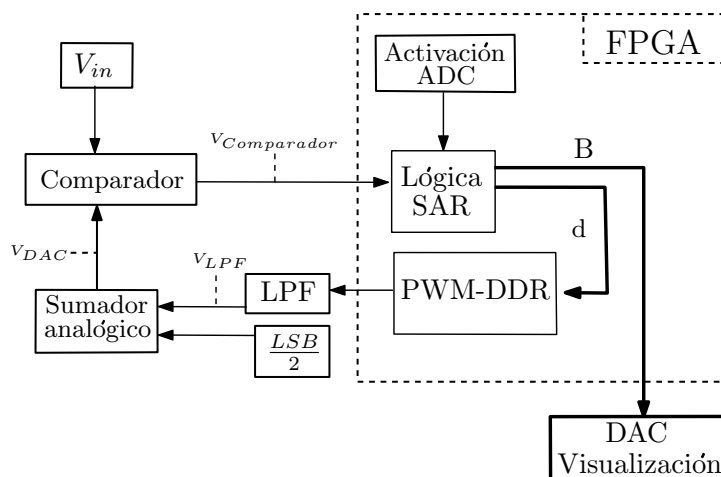


Figura 3.11: Esquemático de la implementación del ADC basado en SAR y DAC adicional para visualización

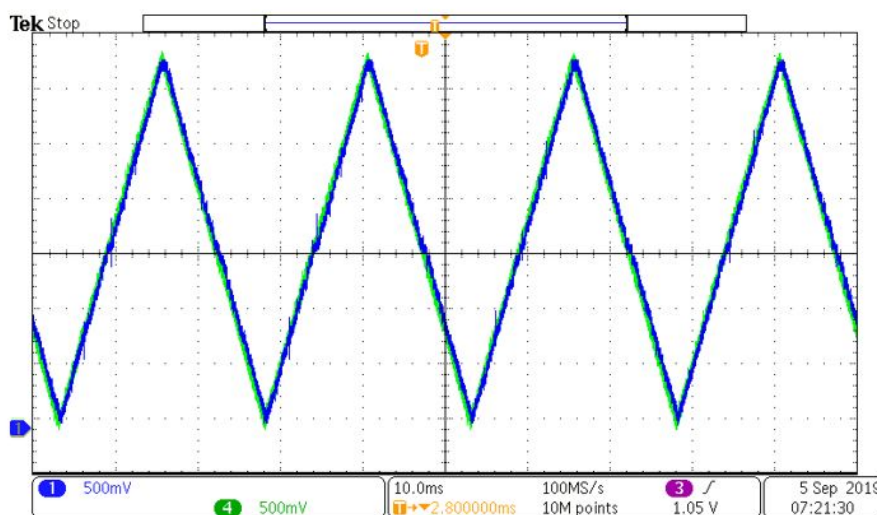
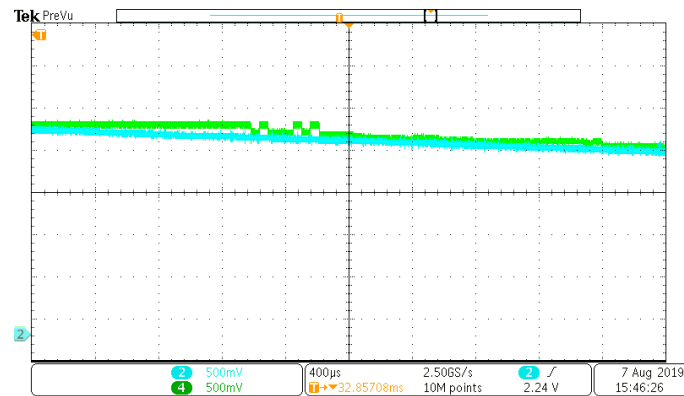
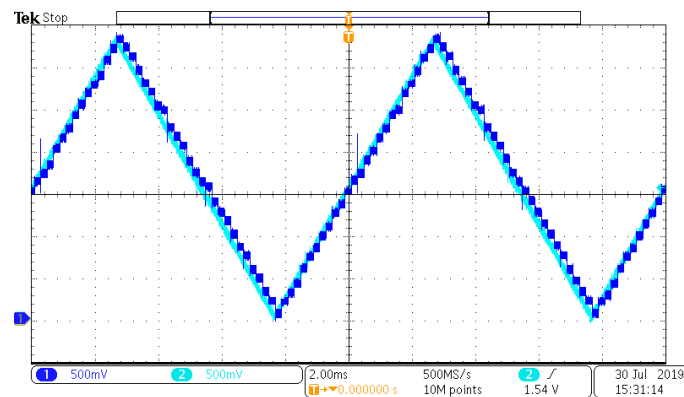


Figura 3.12: V_{in} y B obtenido por un ADC de 8-bit

Al realizar la implementación del ADC, se pueden presentar errores al escoger f_C y O_{LPF} , que impedirían cumplir con la condición de la Eq. 2.6. Una vez sea implementado es difícil corregir dicho problema. También se generan errores si se configura mal el momento de realizar la comparación entre V_{in} y V_{DAC} . En la Fig. 3.13a se muestra el error de monotonicidad, debido a que realiza la comparación antes de que LPF alcanzara el estado estacionario. En la Fig. 3.13b se muestra el error en la conversión debido a que la señal triangular de V_{in} cambia en más de un LSB antes de generar B , por lo que el resultado sería erróneo.



(a) V_{in} y B con error de monotonicidad debido a mala selección del momento de realizar la comparación



(b) V_{in} y B con error de medición debido a que la frecuencia de V_{in} es mayor a $f_{Sampling}$

Figura 3.13: Errores en B debido a escoger de manera errónea parámetros del LPF

3.5. ADC de 10-bit basado en SAR con PWM-DDR y LPF

Para comprobar el diseño del ADC basado en SAR con el PWM-DDR se diseña un ADC de 10-bit con $f_{Sampling}=8$ KSPS, como se realizó en el capítulo 2 en la sección 2.5. El método para encontrar f_C y O_{LPF} del LPF que cumplan con la condición de la Eq. 2.6 es el mismo.

Los parámetros de la FPGA son $f_{clk}=100$ MHz y $\Delta V_O=3.3$ V y a partir de ellos se halla LSB , f_C y O_{LPF} . A partir de la Eq. 3.1 se halla $f_{PWM}=195.31$ kHz y a partir de la Eq. 2.6 $LSB=3.22$ mV, por último, se obtiene f_C y O_{LPF} que cumplan con la condición de la Eq. 2.6. En la Fig. 3.14 se muestra $t_{Settling}$ Vs f_C , para diferentes O_{LPF} .

Como se puede apreciar en la Fig. 3.14, no se pudo encontrar un LPF que permita obtener $t_{Settling} < 12.5 \mu s$, por lo que no es posible implementar un ADC de 10-bit de esta manera. El usuario debe escoger si disminuir N o $f_{Sampling}$ para implementar el ADC. También es posible modificar la forma de implementación del PWM, pero esto sera tratado en capítulos posteriores.

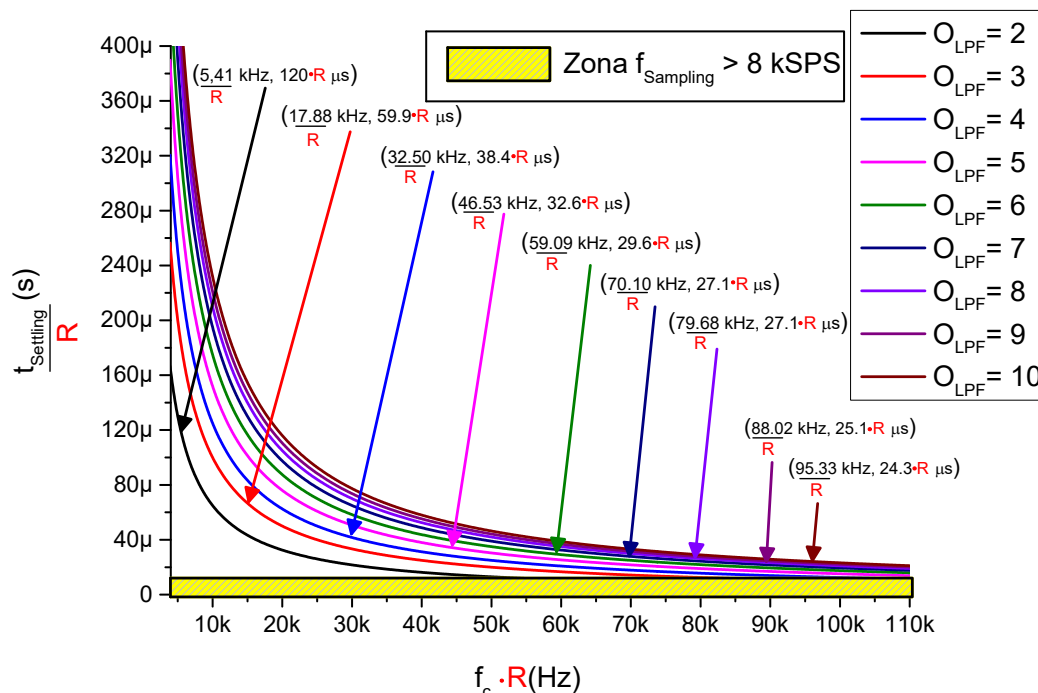


Figura 3.14: f_c vs. $t_{Settling}$ para un ADC de 10-bit con $f_{Sampling} = 8$ KSPS normalizado con $f_{CLK} = 100$ MHz.

3.6. Conclusiones

En este capítulo se estudió la implementación de un modulador de anchura de pulsos (PWM-DDR) regido por un contador con flanco ascendente y descendente del reloj, aprovechando la salida "double data rate" (DDR) incorporada en la FPGA, que disminuye el tiempo de propagación de la señal hasta la salida de la FPGA. Esto permite aumentar al doble la frecuencia del PWM (f_{PWM}).

Al aumentar f_{PWM} , esta se aleja de la frecuencia de corte (f_c) del filtro pasa bajo (LPF), aumentando la atenuación del LPF. Lo último se traduce en la posibilidad de obtener una f_c que cumpla con la condición de que el rizado (ΔV_{DAC}) sea menor a un LSB , además de un tiempo de establecimiento del LPF ($t_{Settling}$) menor.

Al comparar los parámetros del LPF hallados para el ADC de 8-bit con un PWM y los hallados con un PWM-DDR para una $f_{Sampling}=8$ KSPS, se pudo disminuir el O_{LPF} de 4 para el PWM a 2 para el PWM-DDR. Esto se ve traducido en una disminución de los costes y área de implementación. La implementación del ADC con esta mejora puede realizarse en cualquier FPGA, siempre y cuando cuente con salida DDR y con flip-flops de flanco ascendente y descendente.

4. Capítulo 4

4.1. Resumen

En este capítulo se estudia una optimización en el convertor digital analógico (DAC) sustituyendo el modulador de anchura de pulsos (PWM) y el filtro pasa bajo (LPF) por un conjunto de PWMs y LPFs. Se estudia la división del PWM de N-bit en un conjunto de 2 PWMs de $\frac{N}{2}$ -bit que permite incrementar la frecuencia máxima de muestreo, reduciendo el rizado a la salida de los LPFs (ΔV_{LPF}). Este método de división del PWM es generalizable en K PWMs de $\frac{N}{K}$ -bit. Se estudian las ventajas y desventajas del diseño, como también los límites de este.

4.2. División del PWM de N-bit en 2 PWMs de $\frac{N}{2}$ -bit

En el capítulo 2 se explicó el funcionamiento del ADC basado en SAR, con la implementación del DAC con un PWM y un LPF. Al variar D del PWM a través del LPF se genera V_{LPF} , que es restada con una tensión equivalente a $\frac{LSB}{2}$ para así obtener V_{DAC} . V_{DAC} es comparado con V_{in} para finalmente obtener B correspondiente a V_{in} , una vez es evaluado cada bit.

Para realizar el diseño del ADC se encuentra f_C y O_{LPF} del LPF que cumplan con la condición de la Eq. 2.6 del capítulo 2. La lógica SAR genera d con base a $V_{Comparador}$ y se envía al PWM. El esquemático del ADC basado en SAR con DAC reemplazado con un PWM y LPF se muestra en la Fig. 4.1.

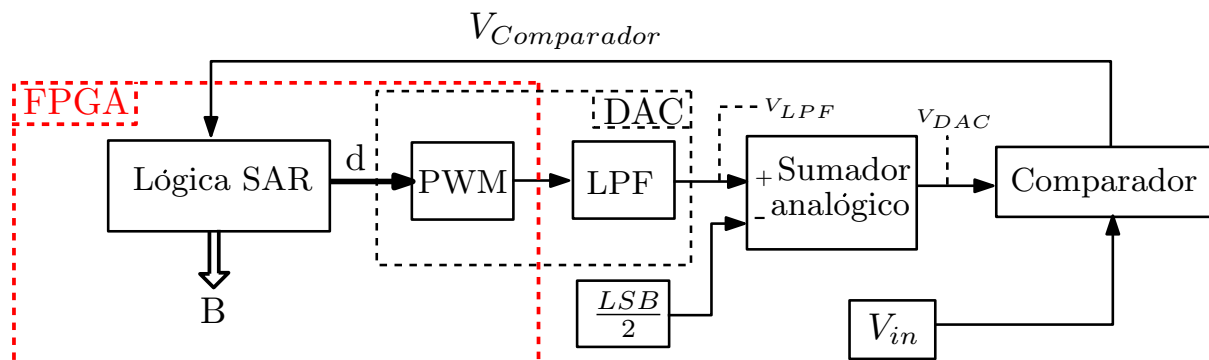


Figura 4.1: Esquemático ADC basado en SAR con DAC reemplazado con PWM y LPF

Como se ha explicado en anteriores capítulos, el tiempo de conversión del ADC ($t_{Conversion}$) es el producto del número de ciclos de bit por el $t_{Settling}$. Una estrategia para disminuir $t_{Settling}$ es aumentar la f_{PWM} , como se mostró en el capítulo 3 con la implementación del PWM-DDR.

Para aumentar f_{PWM} se propone paralelizar el PWM de N-bit en 2 PWMs de $\frac{N}{2}$ -bit. Con esto la frecuencia de modulación del PWM resultante (f_{PWM-2}) es regida por la Eq. 4.1

$$f_{PWM-2} = \frac{f_{clk}}{2^{\frac{N}{2}}} \quad (4.1)$$

Como se muestra en la Eq. 4.2 si se compara la f_{PWM} utilizada en el capítulo 2 y la f_{PWM-2} generada a partir de la paralelización, se obtiene que la f_{PWM-2} es $2^{\frac{N}{2}}$ veces mayor que la f_{PWM} . Si se utiliza el PWM-DDR, $f_{PWM-DDR-2}$ también sería $2^{\frac{N}{2}}$ veces mayor que $f_{PWM-DDR}$ como se puede ver en la Eq. 4.3.

$$f_{PWM-2} = \frac{f_{clk}}{2^{\frac{N}{2}}} \quad (4.2)$$

$$f_{PWM-2} = f_{PWM} \cdot 2^{\frac{N}{2}}$$

$$f_{PWM-DDR-2} = \frac{2 \cdot f_{clk}}{2^{\frac{N}{2}}} \quad (4.3)$$

$$f_{PWM-DDR-2} = f_{PWM-DDR} \cdot 2^{\frac{N}{2}}$$

Para obtener los PWMs de $\frac{N}{2}$ -bit, se divide $d[N-1...0]$ en $d_1 = d[N-1... \frac{N}{2}]$ y $d_0 = d[\frac{N}{2}-1...0]$. d_1 contiene los bits más significativos y d_0 contiene los bits menos significativos de d , donde d_1 es $2^{\frac{N}{2}}$ veces mayor que d_0 . Se puede ver la forma de dividir d en la Fig.4.2.

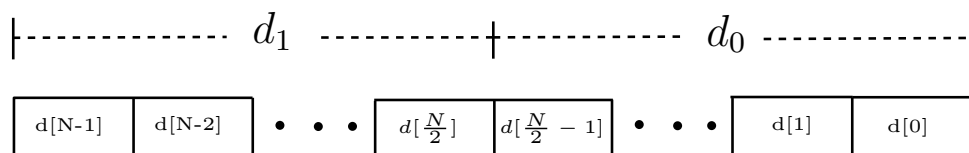


Figura 4.2: Esquemático de la división de d en d_s de $\frac{N}{2}$ -bit

Una vez se divide d en d_1 y d_0 , es necesario obtener la ponderación de cada V_{LPF} resultante de la división, para obtener la ecuación que rige V_{DAC} . Partiendo de la Eq. 4.4a, con la que se genera V_{DAC} a partir de d , se obtiene la Eq. 4.4b, donde se reemplaza d con

su división en d_1 y d_0 . Por último, en la Eq. 4.4c se muestra V_{DAC} a partir de los V_{LPF} generados por los PWMs resultantes, además de tener en cuenta el equivalente a $\frac{LSB}{2}$.

$$V_{DAC} = \Delta V_O \left(\frac{d}{2^N} \right) - \frac{LSB}{2} \quad (4.4a)$$

$$V_{DAC} = \Delta V_O \left(\frac{d_1 \cdot 2^{\left(\frac{N}{2}\right)} + d_0}{2^N} \right) - \frac{LSB}{2} \quad (4.4b)$$

$$V_{DAC} = \Delta V_O \left(V_1 + \frac{V_0}{2^{\left(\frac{N}{2}\right)}} \right) - \frac{LSB}{2} \quad (4.4c)$$

En la Fig 4.3 se muestra el esquemático de la paralelización del PWM, donde se necesitaría una salida más en la FPGA y un LPF extra, además de un sumador con ponderación para obtener V_{DAC} . Esto último incrementa los costes y área de implementación, pero tiene la ventaja de que se pueden obtener $f_{Sampling}$ más altas.

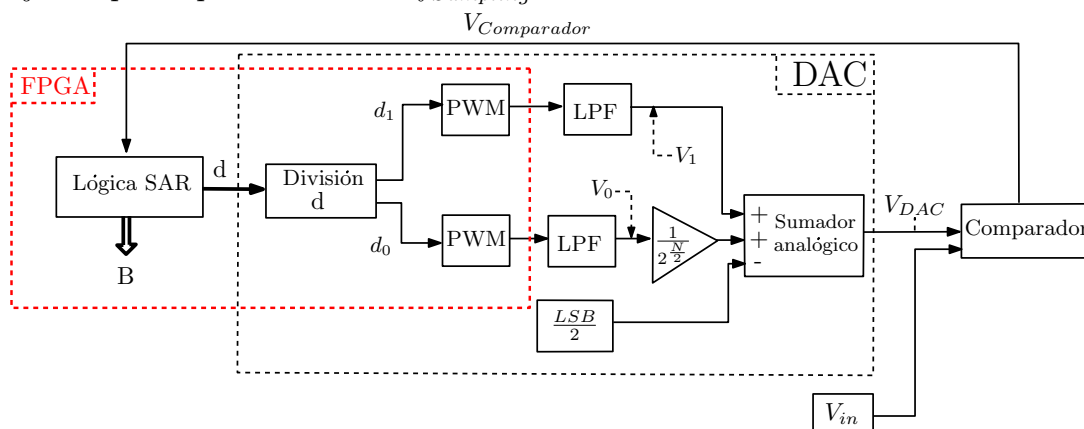


Figura 4.3: Esquemático del ADC basado en SAR con DAC reemplazado por 2 PWMs de $\frac{N}{2}$ -bit y 2 LPFs

4.2.1. Obtención parámetros del LPF para un ADC de N-bit con 2 PWMs de $\frac{N}{2}$ -bit

En los capítulos 2 y 3 se mostró la expresión que permite encontrar f_C a partir de las características del ADC, los parámetros de la FPGA y O_{LPF} . Al paralelizar el PWM o PWM-DDR, es necesario modificar la expresión para encontrar los parámetros del LPF. Para eso se modifica la Eq. 2.25 del capítulo 2 y se obtienen la Eq. 4.5a y 4.5b para el PWM y PWM-DDR respectivamente.

$$A_{LPF}(f_{PWM}) = -20 \cdot O_{LPF} \cdot \log_{10} \left(\frac{f_{PWM}}{f_C} \right) = -20 \cdot O_{LPF} \cdot \log_{10} \left(\frac{f_{clk}}{2^{(\frac{N}{2})} \cdot f_C} \right) \quad (4.5a)$$

$$A_{LPF}(f_{PWM-DDR}) = -20 \cdot O_{LPF} \cdot \log_{10} \left(\frac{2 \cdot f_{clk}}{2^{(\frac{N}{2})} \cdot f_C} \right) \quad (4.5b)$$

A partir de la Eq. 2.22 y la Eq. 2.6 en escala logarítmica se obtiene la Eq. 4.6a para el PWM, mientras para el PWM-DDR se obtiene la Eq. 4.6b, partiendo de la condición $A_{LPF}(f_{PWM}) < A_1(max)$ explicada en el capítulo 2.

$$\left(\frac{f_C}{f_{clk}} \right) < \frac{O_{LPF} \sqrt{\pi}}{2^{(\frac{N}{2} + \frac{N+2}{O_{LPF}})}} \quad (4.6a)$$

$$\left(\frac{f_C}{f_{clk}} \right) < \frac{2 \cdot O_{LPF} \sqrt{\pi}}{2^{(\frac{N}{2} + \frac{N+2}{O_{LPF}})}} \quad (4.6b)$$

Para mostrar la optimización obtenida al paralelizar el PWM o PWM-DDR, se obtendrá los parámetros del LPF para el ADC de 6-bit y se muestra la comparación en la Fig. 4.4. Se puede observar que al dividir el PWM y el PWM-DDR para un mismo O_{LPF} , las prestaciones del ADC mejoran notablemente con respecto a los otros diseños. Los parámetros son hallados a partir de las Eq. 4.6a y 4.6b. Se incluye el factor R , con el objetivo de que el diseño sea extrapolable a otras familias de FPGA.

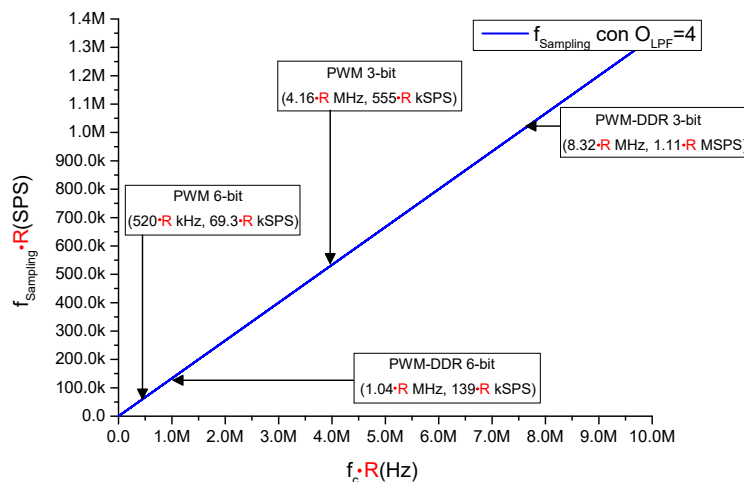


Figura 4.4: f_C vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.

4.2.2. ADC de 8-bit con 2 PWMs de 4-bit

En los capítulos 2 y 3 se implementó un ADC de 8-bit con una $f_{Sampling}=8$ kSPS en una FPGA de la familia Zynq 7000, con $\Delta V_O=3.3$ V, $f_{clk}=100$ MHz y con la posibilidad de implementar el PWM-DDR. Para poder analizar las prestaciones de la división del PWM-DDR se implementará un ADC de 8-bit utilizando dos PWM-DDRs de 4-bit. Se realiza el cálculo de f_C para distintos O_{LPF} a partir de la Eq. 4.6b y se muestran en la Fig. 4.5. Cabe recordar que se multiplica los valores hallados por un factor R , con el objetivo de que sea extrapolable a FPGAs de diferentes familias.

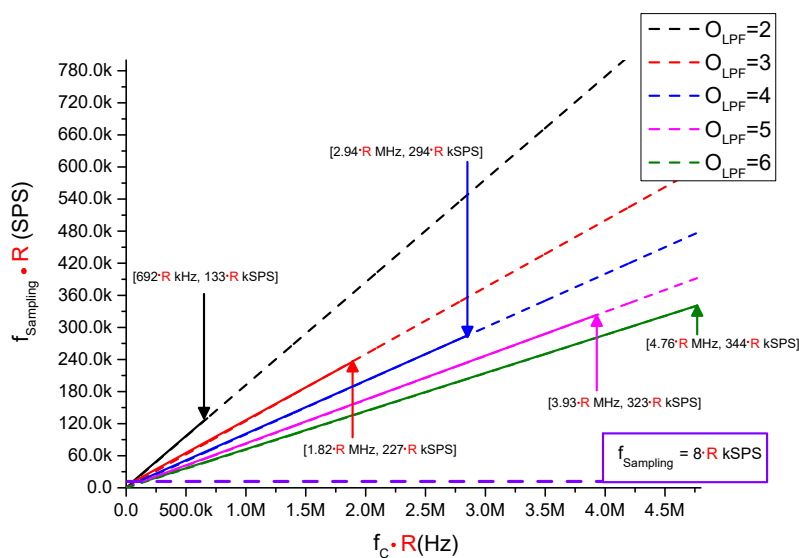


Figura 4.5: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=8$ kSPS con 2 PWM-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.

Hay que tener en cuenta que al dividir el PWM, en PWMs de $\frac{N}{2}$ -bit, es necesario implementar dos PWMs y dos LPFs, incrementando el área de la implementación, además del uso de una salida extra de la FPGA. En cuanto a los elementos analógicos de los LPFs dependerá de los parámetros escogidos, pues al dividir el PWM es posible que disminuya el O_{LPF} de cada LPF.

Si se compara los valores del LPF necesario para implementar el ADC de 8-bit con $f_{Sampling}=8$ kSPS, tanto con un PWM-DDR (Fig. 3.5 del capítulo 3), como con 2 PWM-DDR de 4-bit (Fig. 4.5), es necesario un $O_{LPF}=2$ para ambos casos. Lo anterior significa que para este caso en específico, saldría más rentable (tanto en coste, área de implementación y número de elementos analógicos) implementar el ADC con un LPF y un PWM-DDR.

Para la implementación del ADC de 8-bit con el PWM-DDR dividido en 2 PWMs de 4-bit, se deben realizar unas modificaciones en la implementación como se muestra en el esquemático de la Fig. 4.6. El ADC consta de los módulos 'Lógica SAR', 'División D', dos PWM-DDR, dos LPFs, un sumador analógico y un comparador.

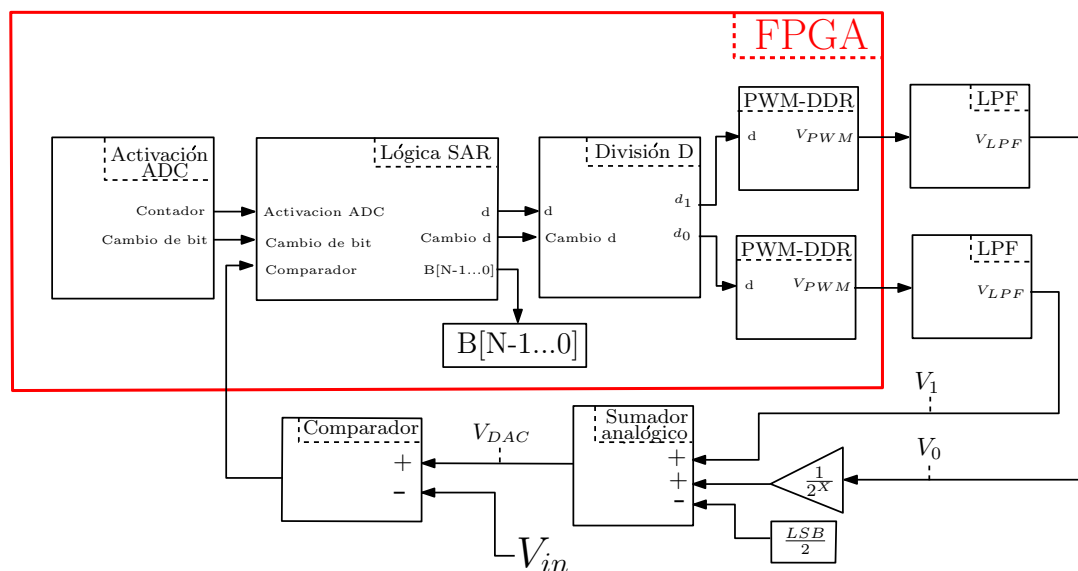


Figura 4.6: Esquemático de la implementación del ADC con 2 PWM-DDR de $\frac{N}{2}$ -bit.

Se obtiene V_{DAC} a partir de la Eq. 4.4c, donde se suma V_0 atenuado $2^{(\frac{N}{2})}$ veces y V_1 , además se resta el equivalente a $\frac{LSB}{2}$. Para esto se implementa un sumador analógico a través de un AO como se muestra en la Fig.4.7, donde se incluye una resistencia de ajuste del circuito (R_T) [87].

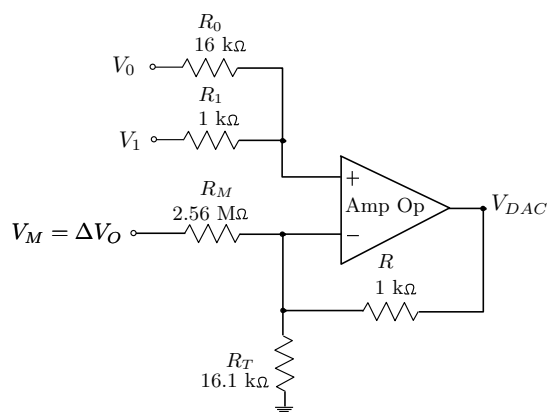


Figura 4.7: Circuito sumador-restador con AO para un ADC de 8-bit.

En primer lugar, se obtienen las resistencias R_0 y R_1 correspondientes a las tensiones V_0 y V_1 . Las resistencias se obtienen a partir de la resistencia de retroalimentación R con la

Eq. 4.7. V_M es la tensión continua con la que se pretende generar el equivalente a $\frac{LSB}{2}$ y su resistencia es R_M .

$$\begin{aligned} R_1 &= R \\ R_0 &= \frac{R}{2^{(\frac{N}{2})}} \\ R_M &= \frac{R}{2^{N+1}} \end{aligned} \quad (4.7)$$

Para obtener R_T se utiliza la Eq. 4.8, donde A_{Suma} es la suma de las ganancias de V_1 y V_0 de la Eq. 4.4c y A_{Resta} es la ganancia del equivalente de $\frac{LSB}{2}$. Esta resistencia se conecta entre la entrada no inversora y masa [88] como se muestra en la Fig. 4.7.

$$\begin{aligned} A_{Suma} &= 1 + \frac{1}{2^{(\frac{N}{2})}} \\ A_{Resta} &= \frac{1}{2^{N+1}} \\ R_T &= \frac{R}{A_{Resta} + 1 - A_{Suma}} \end{aligned} \quad (4.8)$$

A partir de las Eq. 4.7 y 4.8 se obtienen los valores de las resistencias de la Fig. 4.7 para el sumador analógico del ADC de 8-bit. En este ejemplo se fijó $R=100 \Omega$ y se obtuvo $R_1=100 \Omega$, $R_0=1.6 \text{ k}\Omega$, $R_T=1.610 \text{ k}\Omega$ y por último $R_M=256 \text{ k}\Omega$ con $V_M=3.3 \text{ V}$.

Aprovechando la posibilidad de aumentar la $f_{Sampling}$, se implementará el ADC de 8-bit con $f_{Sampling}=340 \text{ kSPS}$. Se presenta en la Fig. 4.8 f_C vs. $f_{Sampling}$, donde se aprecia que solo es posible implementarse con $O_{LPF}=6$ y $f_C=4.76 \text{ MHz}$. En la Tabla 4.1 se muestran los elementos analógicos necesarios para implementar el ADC de 8-bit. Para implementar dos LPFs con $O_{LPF}=6$ es necesario 6 AO, 12 resistencias y 12 condensadores, además el AO del sumador y sus respectivas resistencias (mostradas en la Fig. 4.7). En cuanto a las salidas de la FPGA, son necesarias 2 salidas para los PWM-DDR y una para el comparador.

Se muestra en la Fig. 4.9 el comportamiento de V_{DAC} durante la conversión. Por último, en la Fig. 4.10 los valores de conversión B para una entrada V_{in} de tipo rampa, con rango entre 0 V a 3,3 V.

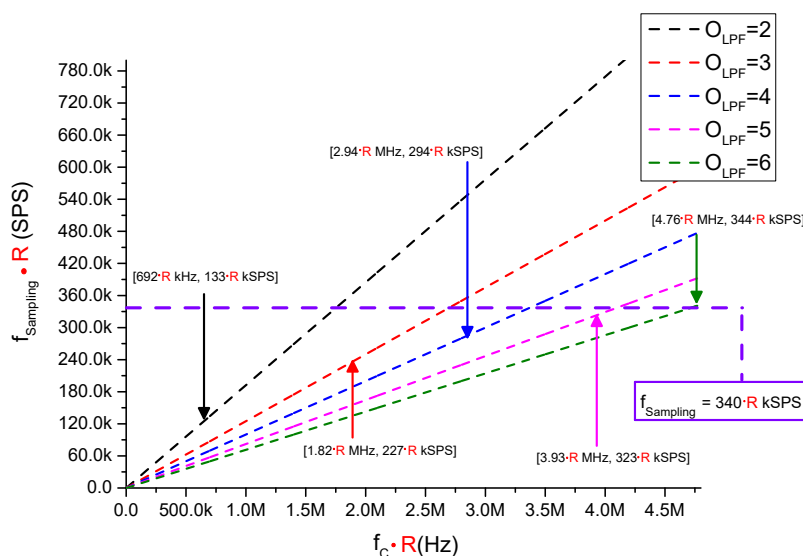


Figura 4.8: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=340$ kSPS con 2 PWM-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.

PWM-DDR	O_{LPF}	R	C	Número AO	Entradas/Salidas en la FPGA
4-bit	6	17	12	7	3

Tabla 4.1: Características de la implementación del ADC de 8-bit con 2 PWM-DDR

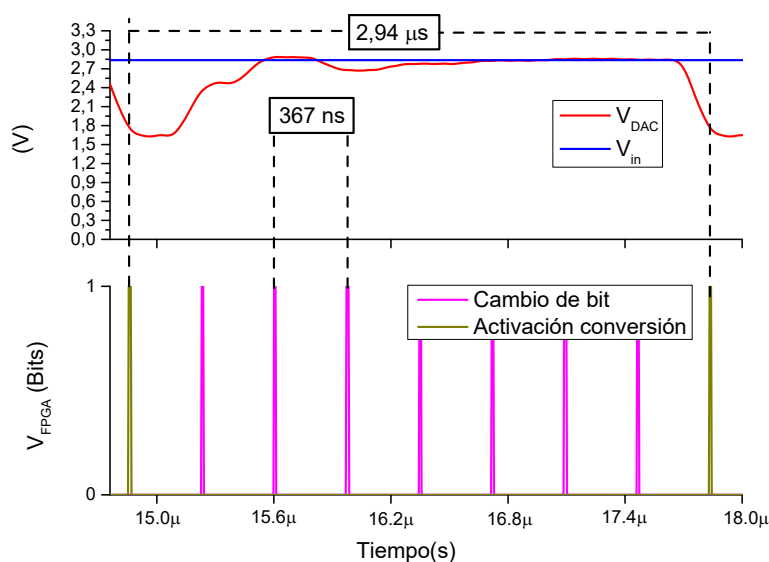


Figura 4.9: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con $f_{Sampling}=340$ kSPS, implementado con 2 PWM-DDR de 4-bit

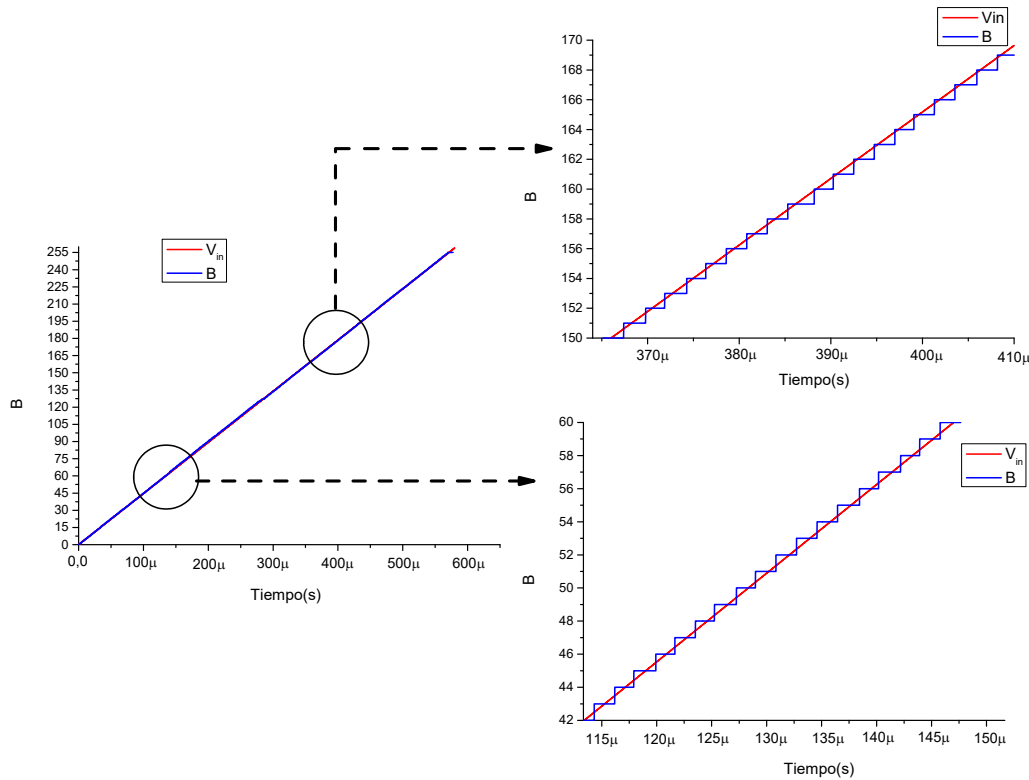


Figura 4.10: B a partir de V_{in} con un ADC de 8-bit con $f_{Sampling}=340$ kSPS implementado con 2 PWM-DDR de 4-bit

4.3. División del PWM en K PWMs de $\frac{N}{K}$ -bit

En la sección 4.2 el DAC fue modificado para aumentar f_{PWM} . Para esto se reemplazó el DAC que en principio estaba implementado por un PWM y un LPF, por dos PWMs y dos LPFs, dividiendo el PWM de N -bit en 2 PWMs de $\frac{N}{2}$ -bit. A continuación, se propone generalizar la división del PWM N -bit en K PWMs de $\frac{N}{K}$ -bit. Para dividir el PWM en K PWMs de $\frac{N}{K}$ -bit, se obtienen los ds como se muestra en la Fig. 4.11.

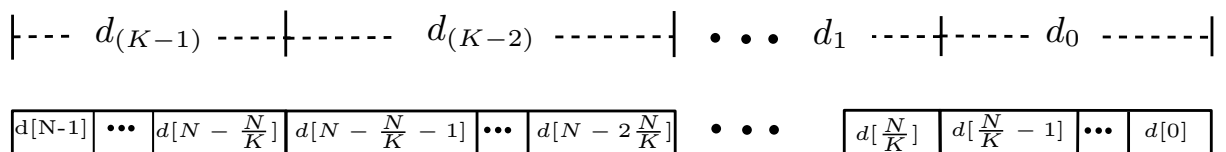


Figura 4.11: Esquemático de la división de d en K ds de $\frac{N}{K}$ -bit

Si se compara la f_{PWM} utilizada en el capítulo 2 y la f_{PWM-k} generada a partir de la

paralelización del PWM en K PWMs, se obtiene que la f_{PWM-K} es $2^{\frac{N}{K} \cdot (K-1)}$ veces mayor que la f_{PWM} como se muestra en la Eq. 4.9 . Si se utiliza el PWM-DDR, $f_{PWM-DDR-K}$ también sería $2^{\frac{N}{K} \cdot (K-1)}$ veces mayor que $f_{PWM-DDR}$ como se puede ver en la Eq. 4.10.

$$f_{PWM-K} = \frac{f_{clk}}{2^{\frac{N}{K}}} \quad (4.9)$$

$$f_{PWM-K} = f_{PWM} \cdot 2^{\frac{N}{K} \cdot (K-1)}$$

$$f_{PWM-DDR-K} = \frac{2 \cdot f_{clk}}{2^{\frac{N}{K}}} \quad (4.10)$$

$$f_{PWM-DDR-K} = f_{PWM-DDR} \cdot 2^{\frac{N}{K} \cdot (K-1)}$$

Para obtener V_{DAC} a partir de los V_{LPF} generados por los PWMs resultantes de la división, se parte de la Eq. 4.4a. En la Eq. 4.11 se reemplaza d por su división en ds de $\frac{N}{K}$ -bit. Por último, en la Eq. 4.13 se muestra la ecuación generalizada de V_{DAC} obtenida a partir de las Eq. 4.11 y 4.12.

$$V_{DAC} = \Delta V_O \left(\frac{d_{K-1} \cdot 2^{\frac{N}{K} \cdot (K-1)}}{2^N} + \dots + \frac{d_1 \cdot 2^{\frac{N}{K}}}{2^N} + \frac{d_0}{2^N} \right) - \frac{LSB}{2} \quad (4.11)$$

$$V_{DAC} = \Delta V_O \cdot \sum_{j=0}^{K-1} \left(\frac{2^{\frac{N}{K} \cdot j}}{2^N} \cdot d_j \right) - \frac{LSB}{2}$$

$$V_j = A_{0-j} = \Delta V_O \cdot \frac{d_j}{2^{\frac{N}{K}}} \quad (4.12)$$

$$V_{DAC} = \sum_{j=0}^{K-1} \left(\frac{2^{\frac{N}{K} \cdot (j+1)}}{2^N} \cdot V_j \right) - \frac{LSB}{2} = \sum_{j=0}^{K-1} \left(\frac{2^{\frac{N}{K} \cdot j}}{2^{N-\frac{N}{K}}} \cdot V_j \right) - \frac{LSB}{2} \quad (4.13)$$

$$V_{DAC} = V_{(K-1)} + \sum_{j=0}^{K-2} \left(\frac{2^{\frac{N}{K} \cdot j}}{2^{N-\frac{N}{K}}} \cdot V_j \right) - \frac{LSB}{2}$$

En la Fig. 4.12 se muestra el esquemático de la implementación del ADC con la optimización presentada. El PWM de N-bit se divide en K PWMs de $\frac{N}{K}$ -bit, los ds de $\frac{N}{K}$ -bit obtenidos

son enviados a los PWMs, que junto con los LPFs y un sumador analógico regido por la Eq. 4.13, generan V_{DAC} . Esta implementación brindaría al usuario una mayor versatilidad y adaptabilidad a los requerimientos del ADC.

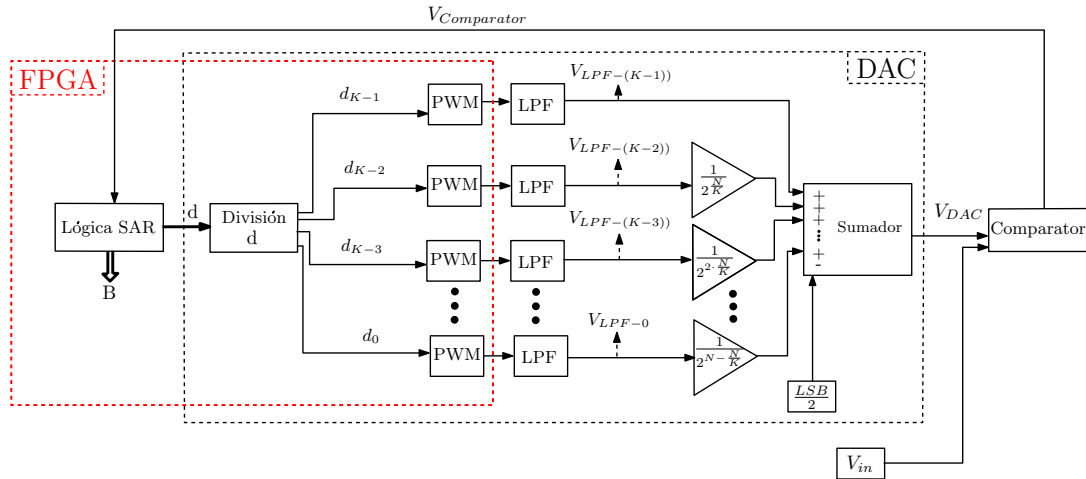


Figura 4.12: Esquemático del ADC basado en SAR con DAC reemplazado por PWMs de $\frac{N}{K}$ -bit

El procedimiento anterior se realiza para la división del PWM de N-bit en K PWMs de $\frac{N}{K}$ -bit, siendo $\frac{N}{K}$ un número entero. Si se requiere dividir el PWM en K PWMs de $\frac{N}{K}$ -bit y $\frac{N}{K}$ no es un número entero se debe realizar una modificación con el objetivo de obtener V_{DAC} .

Al dividir N entre K, el resultado se debe aproximar al siguiente número entero (P_1). Se divide el PWM en $K - 1$ PWMs de P_1 -bit y un último PWM de P_2 -bit, siendo $P_2 = N - P_1 \cdot (K - 1)$. Se muestra la forma de dividir d en K d_s en la Fig. 4.13. Partiendo de la Eq. 4.4a y de la forma de división de d mostrada en la Fig. 4.13, se obtiene la Eq. 4.14.

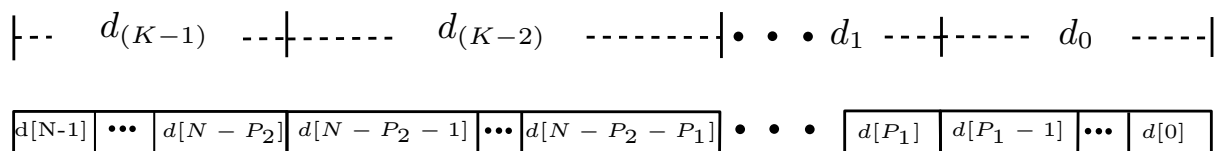


Figura 4.13: Esquemático de la división de d en K d_s de $\frac{N}{K}$ -bit con $\frac{N}{K}$ inexacto

$$\begin{aligned}
V_{DAC} &= \frac{2^{P_2+P_1 \cdot (K-1)}}{2^N} \cdot V_{(K-1)} + \sum_{j=0}^{K-2} \left(\frac{2^{P_1 \cdot (j+1)}}{2^N} \cdot V_j \right) - \frac{LSB}{2} \\
V_{DAC} &= V_{(K-1)} + \sum_{j=0}^{K-2} \left(\frac{2^{P_1 \cdot (j+1)}}{2^N} \cdot V_j \right) - \frac{LSB}{2}
\end{aligned} \tag{4.14}$$

El inconveniente que presenta la optimización de dividir el PWM en K PWMs de $\frac{N}{K}$ -bit radica en que, al aumentar K, aumenta el número de salidas utilizadas en la FPGA y cada PWM tiene su respectivo LPF, incrementando el coste y el área de implementación del circuito.

Si al dividir el PWM en K PWMs, se mantiene el mismo O_{LPF} , el coste de implementación incrementa a medida que K lo hace. Ahora bien, si al hallar los parámetros del LPF de los PWMs de $\frac{N}{K}$ -bit, es posible disminuir O_{LPF} , el número de elementos analógicos puede disminuir. Esto dependerá de la $f_{sampling}$ que se necesite, pues a mayor $f_{sampling}$ mayor O_{LPF} .

Una vez se obtenga V_{DAC} a partir de V_{LPF} se debe implementar el sumador analógico con un AO. En la Fig. 4.14 se muestra la forma de implementación de dicho sumador analógico. Para obtener las resistencias relacionadas con cada V_{LPF} se utiliza la Eq. 4.15 cuando $\frac{N}{K}$ es entero, o las Eq. 4.16a y 4.16b cuando $\frac{N}{K}$ no es entero. Para obtener R_M se utiliza las Eq. 4.8.

$$R_{(K-1-j)} = \sum_{j=0}^{K-1} \left(\frac{R}{2^{\left(\frac{N}{K} \cdot j\right)}} \right) \tag{4.15}$$

$$R_{(K-1)} = \left(\frac{R}{2^{P_2}} \right) \tag{4.16a}$$

$$R_{(K-2-j)} = \sum_{j=1}^{K-1} \left(\frac{R}{2^{(P_2+P_1 \cdot j)}} \right) \tag{4.16b}$$

Para obtener R_T se utiliza las Eq. 4.17d, donde A_{Suma} es la ganancia de cada uno de los V_{LPF} y se obtiene con la Eq. 4.17a o la Eq. 4.17b, mientras que A_{Resta} es la ganancia del equivalente a $\frac{LSB}{2}$.

$$A_{Suma} = 1 + \sum_{j=1}^{K-1} \left(\frac{1}{2^{\frac{N}{K} \cdot j}} \right) \quad (4.17a)$$

$$A_{Suma} = 1 + \frac{1}{2^{P_2}} + \sum_{j=1}^{K-2} \left(\frac{1}{2^{P_2+P_1 \cdot j}} \right) \quad (4.17b)$$

$$A_{Resta} = \frac{1}{2^{N+1}} \quad (4.17c)$$

$$R_T = \frac{1}{A_{Resta} + 1 - A_{Suma}} \quad (4.17d)$$

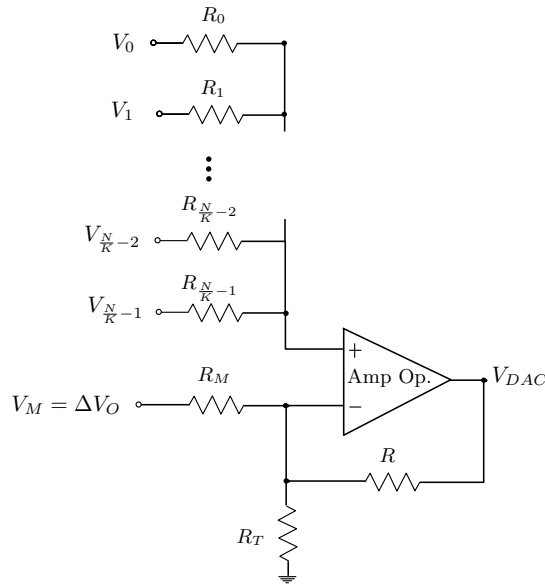


Figura 4.14: Circuito sumador analógico con AO para un ADC de N-bit con el PWM dividido en K PWMs de $\frac{N}{K}$ -bit.

4.3.1. Obtención de los parámetros del LPF para un ADC de N-bits con K PWMs de $\frac{N}{K}$ -bit

En la sección 4.2.1 se obtuvo la expresión generalizada de f_C a partir de O_{LPF} para un ADC de N-bit dividiendo el PWM o el PWM-DDR en 2 de $\frac{N}{2}$ -bit, esta expresión se muestra en la Eq. 4.6a para el PWM y la Eq. 4.6b para el PWM-DDR. Ahora se presenta la expresión para hallar f_C para K PWMs de $\frac{N}{K}$ -bit en la Eq. 4.18a y la Eq. 4.18b para K PWM-DDR de $\frac{N}{K}$ -bit.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{O_{LPF}\sqrt{\pi}}{2\left(\frac{N}{K} + \frac{N+2}{O_{LPF}}\right)} \quad (4.18a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2 \cdot O_{LPF}\sqrt{\pi}}{2\left(\frac{N}{K} + \frac{N+2}{O_{LPF}}\right)} \quad (4.18b)$$

Para comparar la mejora generada por la división del PWM en K PWMs de $\frac{N}{K}$ -bit, se realiza el diseño de un ADC de 6-bit, con PWM-DDR y con los siguientes parámetros de la FPGA, $f_{clk}=100$ MHz y $\Delta V_O=3.3$ V. Se muestra los resultados del LPF en la Fig. 4.15, donde se obtienen los parámetros del LPF con $O_{LPF}=4$, además en la Tabla 4.2 se muestra los componentes analógicos necesarios para implementar el ADC.

A medida que aumenta K la $f_{Sampling}$ obtenida es mayor, a cambio de un mayor número de elementos analógicos (si se mantiene O_{LPF}), mayor número de entradas/salidas de la FPGA y mayor área de implementación.

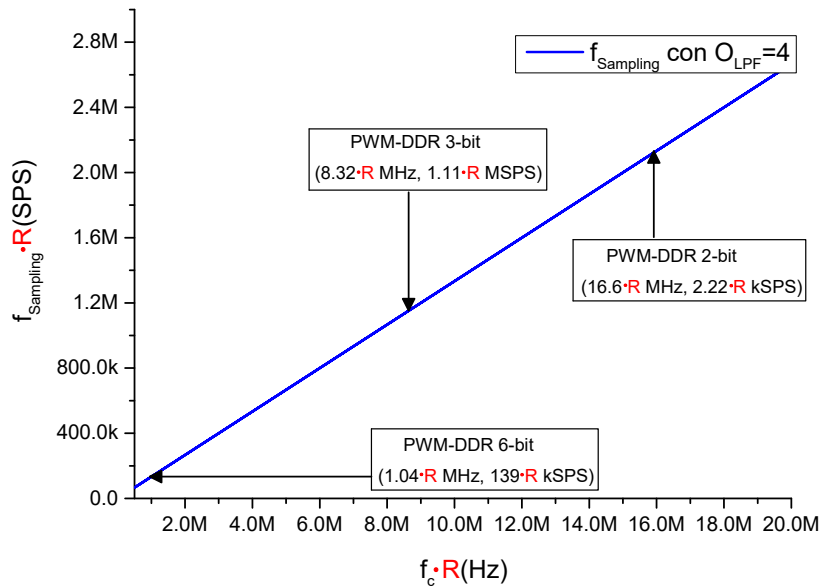


Figura 4.15: f_C vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.

ADC=6-bit	PWM-DDR 6-bit	PWM-DDR 3-bit	PWM-DDR 2-bit
O_{LPF}	4		
Número de R	8	13	23
Número de C	4	8	16
Número de AO	2	5	7

Tabla 4.2: Comparación de las características de implementación del LPF para un ADC de 6-bit con PWM-DDR y diferentes K.

4.3.2. ADC de 8-bit con 4 PWM-DDR de 2-bit

En los capítulos 2 y 3, además de la sección 4.2.2 de este capítulo, se realizó la implementación de un ADC de 8-bit con una $f_{Sampling} = 8$ kSPS con $\Delta V_o = 3.3$ V y $f_{clk} = 100$ MHz. En la sección 4.2.2 se obtuvo los parámetros del LPF con muy buenas prestaciones, mejorando la $f_{Sampling}$ requerida, por lo que se decidió simular un ADC con $f_{Sampling} = 340$ kSPS. A continuación, se realiza el diseño del ADC dividiendo el PWM-DDR en 4 PWM-DDR de 2-bit. Se obtiene f_C vs. $f_{Sampling}$ para diferentes O_{LPF} y se muestran en la Fig. 4.16.

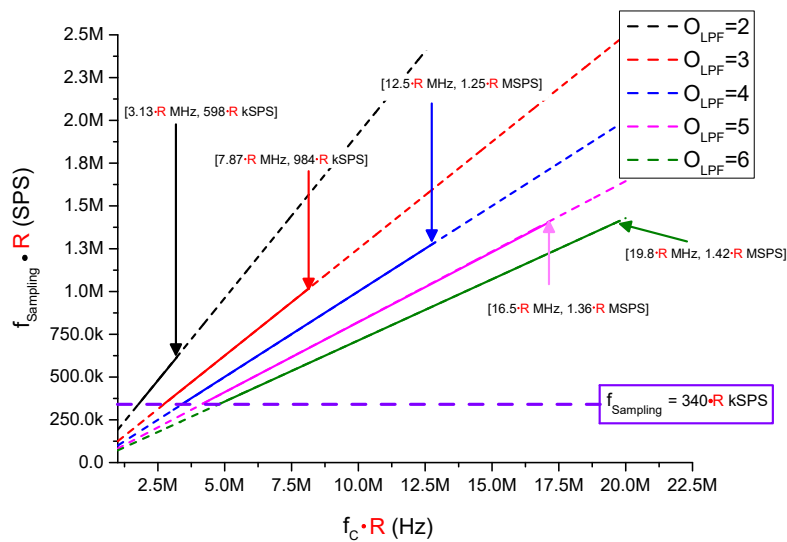


Figura 4.16: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 4 PWM-DDR de 2-bit normalizado a $f_{CLK} = 100$ MHz.

Para obtener V_{DAC} se utiliza la Eq. 4.13, que es la ecuación generalizada de V_{DAC} a partir de V_{LPF} . Se obtiene V_{DAC} en la Eq. 4.19 para el ADC de 8-bit con 4 PWM-DDR de 2-bit. Además de la suma de los V_{LPF} , se resta el equivalente a $\frac{LSB}{2}$. Se obtienen las

resistencias del sumador analógico a partir de la Eq. 4.7, 4.8 y se muestra el sumador analógico implementado en la Fig. 4.17.

$$V_{DAC} = V_3 + \frac{V_2}{2^2} + \frac{V_1}{2^4} + \frac{V_0}{2^6} - \frac{LSB}{2} \quad (4.19)$$

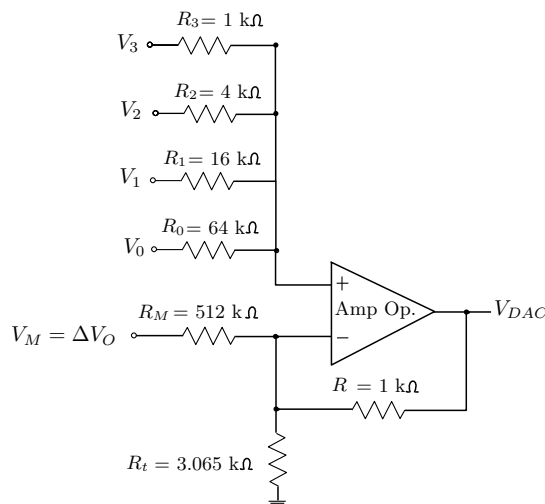


Figura 4.17: Circuito sumador con AO para un ADC de 8-bit con el PWM-DDR dividido en 4 PWMs de 2-bit.

Se muestra el funcionamiento de V_{DAC} en el proceso de conversión del ADC en la Fig. 4.18, donde se obtiene un $t_{Conversion}$ menor al requerido. El momento en donde se realiza la comparación entre V_{in} y V_{DAC} es modificado por el usuario, por lo que se puede implementar un ADC tanto con $f_{Sampling}=533$ kSPS, como uno de $f_{Sampling}=340$ kSPS.

Para obtener la $f_{Sampling}=340$ kSPS presentada en la sección 4.2.2 fue necesario un $O_{LPF}=6$, mientras que dividiendo el PWM-DDR en 4 PWM-DDR de 2-bit fue solo necesario $O_{LPF}=2$. Como se puede apreciar en la Tabla 4.3 si se divide en 2 PWM-DDR son necesarios más elementos analógicos. La ventaja que existe de dividir el PWM-DDR en 2, está en el número de entradas y salidas necesarias en la FPGA que son menos.

PWM-DDR	O_{LPF}	R	C	Número de AO	$t_{Settling}$ del LPF	Entradas/Salidas en la FPGA	$f_{Sampling}$
4-bit	6	17	12	7	367 ns	3	340 kSPS
2-bit	2	15	8	5	234 ns	5	533 kSPS

Tabla 4.3: Comparación de las características de la implementación del ADC de 8-bit con 2 PWM-DDR y 4 PWM-DDR

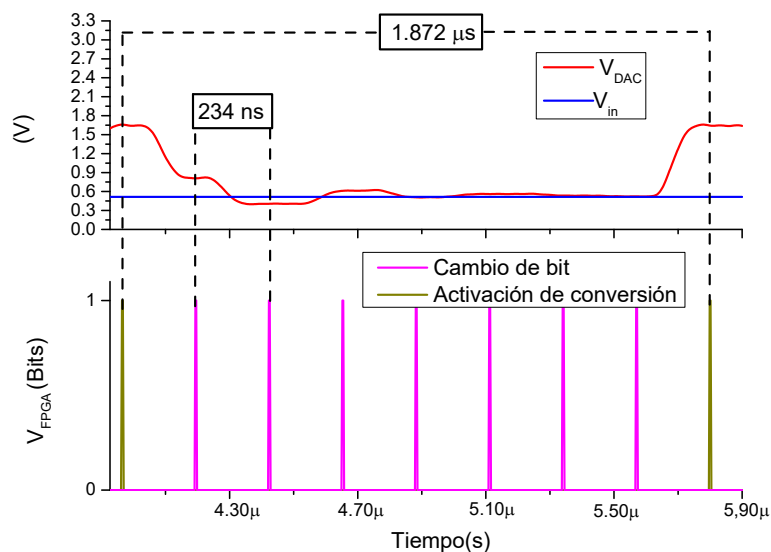


Figura 4.18: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con 4 PWMs de 2-bit

4.3.3. ADC de 8-bit con 8 PWMs de 1-bit

A continuación, se busca aumentar la $f_{sampling}$ dividiendo el PWM en 8 PWM-DDR de 1-bit, siendo el equivalente a implementar un DAC. De esta manera no sería necesaria la implementación de LPFs que generen V_{DAC} a partir de V_{LPF} , sino que se genera V_{DAC} a partir de una red $R - 2R$, de la forma mostrada en la Fig. 4.19.

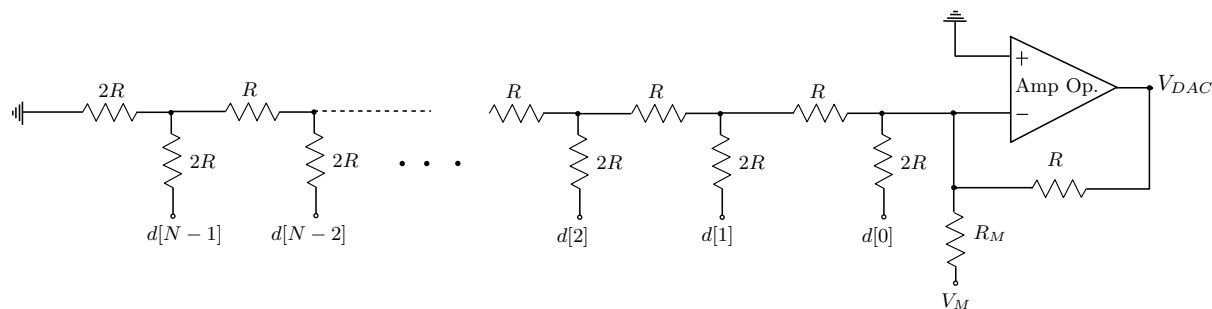


Figura 4.19: Circuito de la red R-2R para implementar DAC.

Lo anterior trae como ventaja el incremento en la $f_{sampling}$ y la disminución de los elementos analógicos. Como desventaja se incrementa el número de salidas utilizadas en la FPGA. El esquemático de la implementación se muestra en la Fig. 4.20.

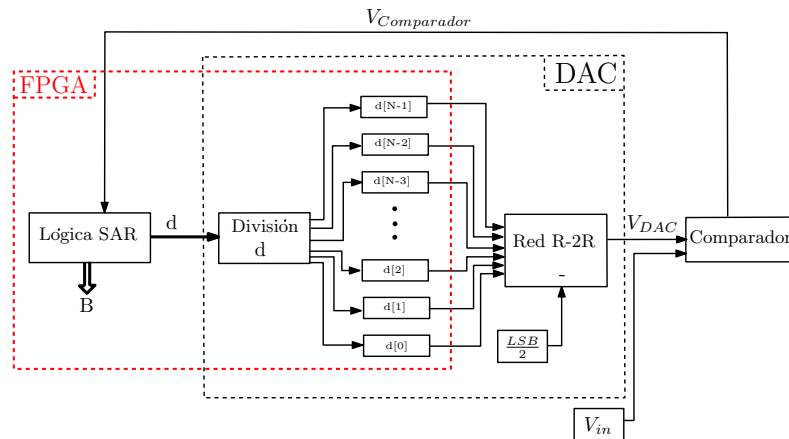


Figura 4.20: Esquemático del ADC basado en SAR con DAC implementado con red R-2R

Al no implementarse ningún LPF el tiempo de establecimiento depende del AO de la red R-2R. En el caso del LT 1364 el tiempo de establecimiento es de 80 ns [86]. De ser necesario mayor f_{Sampling} , se puede implementar con AO que permita menor tiempo de establecimiento, pero que usualmente son más costosos. Un ejemplo sería con el AO THS320 de Texas Instruments, con tiempo de establecimiento es de 50 ns [89]. En la Fig. 4.21 se muestra el comportamiento de V_{DAC} implementado con la red R-2R con el AO LT 1364.

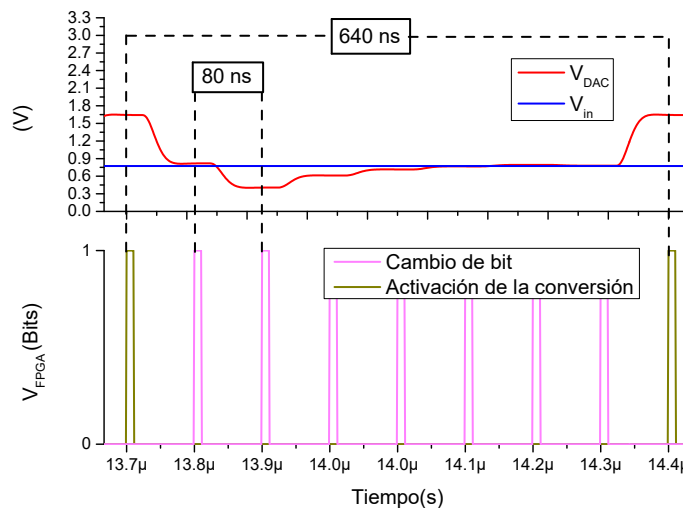


Figura 4.21: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R

En la Fig. 4.22 se muestra los parámetros del LPF con diferentes diseños para un $O_{LPF}=6$ y en la Tabla 4.4 se muestran los elementos analógicos necesarios para su implementación, además de la cantidad de entradas y salidas necesarias en la FPGA. También se muestra

la $f_{Sampling}$ con un PWM-DDR dividido en PWM-DDR de $\frac{N}{K}$ -bit con $K=2$, $K=4$ y $K=8$. Para este último se presenta con dos diferentes AO.

Para obtener las resistencias y AO hay que tener en cuenta los usados en el sumador analógico, mientras que para las entradas y salidas de la FPGA se debe tener en cuenta la entrada utilizada por el comparador.

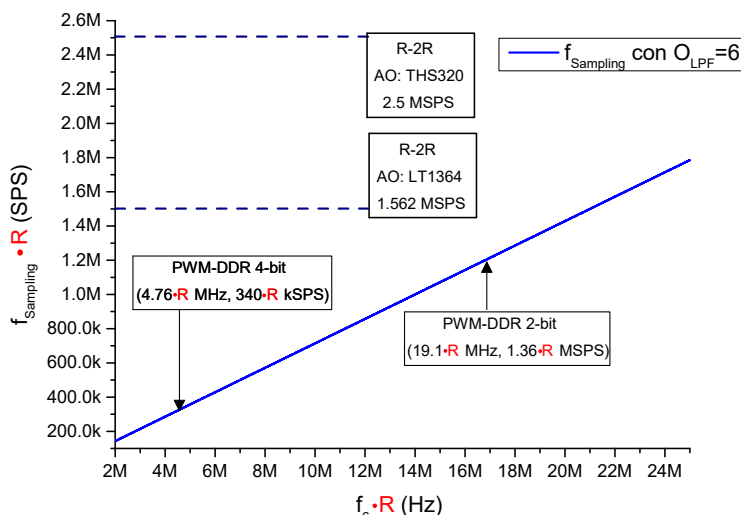


Figura 4.22: f_C vs. $f_{Sampling}$ con los parámetros hallados para un ADC de 8-bit con diferentes diseños, $O_{LPF}=6$ y normalizado con $f_{CLK}=100$ MHz

8-bit $f_{clk}=100$ MHz. PWM-DDR	4-bit	2-bit	1-bit AO: LT1364	1-bit AO: TMS320
K	2	4	8	
O_{LPF}	6			
Número de R	17	31	18	
Número de C	12	24	0	
Número de AO	7	13	1	
Entradas/Salidas de la FPGA	3	5	9	

Tabla 4.4: Comparación de las características de implementación del ADC de 8-bit con PWM-DDR y diferentes diseños

4.3.3.1 Implementación ADC de 8-bit con 8 PWMs de 1-bit

Para la implementación del ADC de 8-bit se utiliza la FPGA de la familia Zynq 7000, con una $f_{clk}=100$ MHz y $\Delta V_O=3.3$ V. Se implementará el ADC con 8 PWM-DDR de 1-bit

implementado con la red $R - 2R$ [90–92]. Se muestra el esquemático de la implementación en la Fig. 4.23

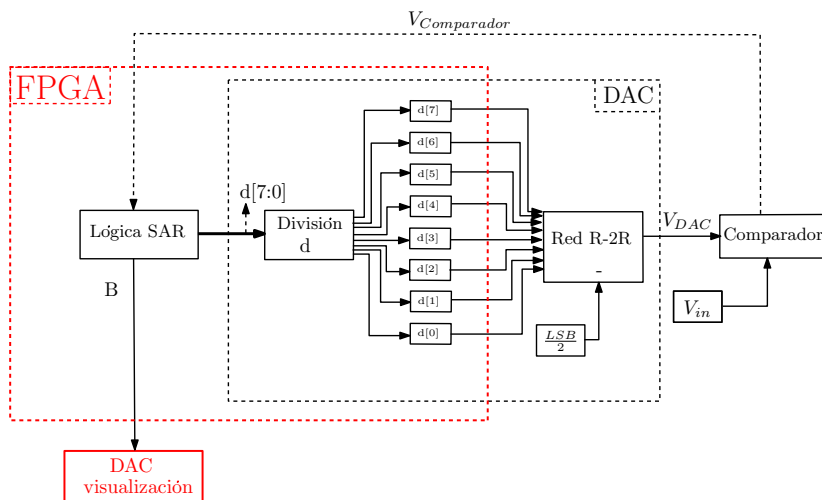


Figura 4.23: Esquemático de la implementación del ADC de 8-bit con 8 PWMs de 1-bit

En primer lugar, se comprueba el comportamiento de V_{DAC} para V_{in} continuo y se muestra en la Fig. 4.24 donde $V_{in}=0.7$ V y la $f_{Sampling}=1.497$ MSPS.

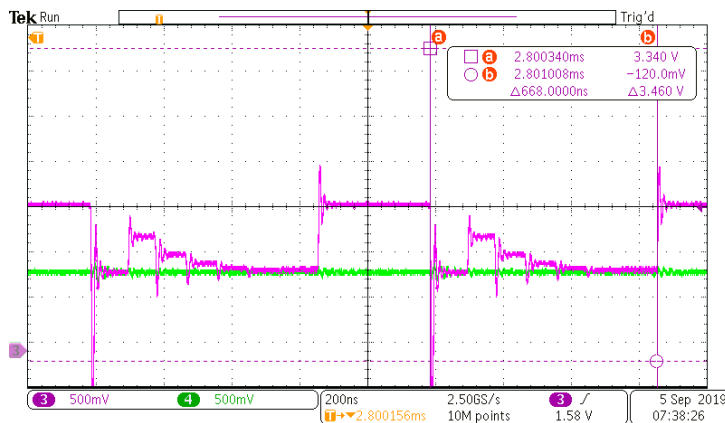


Figura 4.24: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R

Una vez se comprueba el correcto funcionamiento de V_{DAC} , se obtiene B para una señal triangular de V_{in} que permita obtener todos los posibles valores de B .

Para visualizar B se introduce un DAC, para que una vez se haya realizado la conversión, B obtenido sea mostrado y comparado con V_{in} . En la Fig. 4.25 se muestra la señal de entrada V_{in} (señal azul) y B (señal verde).

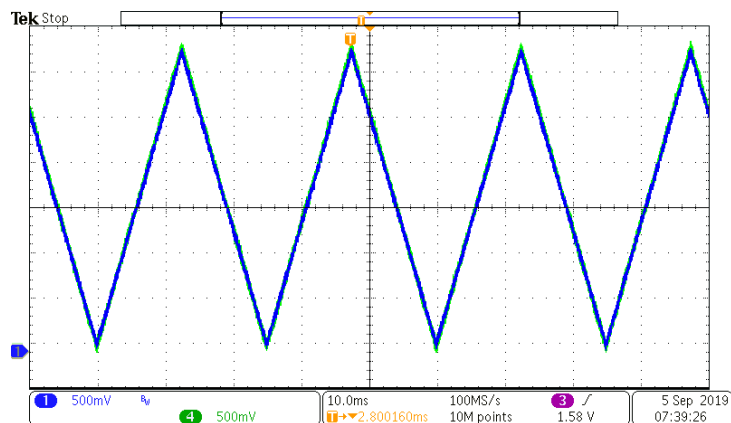


Figura 4.25: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con DAC en red R-2R

4.4. Síntesis del diseño del ADC de N-bit con DAC con K PWMs de $\frac{N}{K}$ -bit

Los diseños presentados en este capítulo tienen la ventaja de brindar al usuario versatilidad a la hora de diseñar e implementar el ADC. Para implementar un ADC de N-bit, es necesario conocer los parámetros de la FPGA como ΔV_O y f_{clk} . También es necesario conocer si la FPGA permite las salidas con el registro DDR (PWM-DDR) o solo con PWM. A continuación, se presentan los pasos necesarios para implementar un ADC con los diseños presentados en este capítulo.

1. A partir de la $f_{Sampling}$, N y la Eq. 2.8 se obtiene $t_{Settling}$, mientras que con los parámetros de la FPGA se obtiene LSB a partir de la Eq. 2.6.
2. El usuario debe escoger a partir de las necesidades del diseño que tipo de implementación desea realizar. Si el usuario tiene entradas y salidas de sobra en la FPGA, puede dividir el PWM-DDR en un mayor número de PWM-DDR de $\frac{N}{K}$ -bit pudiendo obtener mejores $f_{Sampling}$. Por el contrario, si no dispone de muchas entradas y salidas de la FPGA, es necesario disminuir la cantidad de divisiones del PWM-DDR y aumentar el O_{LPF} .
3. En primer lugar se halla f_C a partir de la Eq. 2.26 si se utiliza un PWM o con la Eq. 3.2 si es PWM-DDR. Se obtiene f_C para diferentes O_{LPF} con el objetivo de encontrar el $t_{Settling}$ que permita el $f_{Sampling}$.

4. Si al aumentar el O_{LPF} no se obtiene el $t_{Settling}$ necesario, se debe implementar con otro método. En primer lugar, se debe obtener f_C para diferentes O_{LPF} dividiendo el PWM en 2 PWMs de $\frac{N}{2}$ -bit, para esto se utiliza la Eq. 4.6a. También es posible hacerlo con el PWM-DDR utilizando la Eq. 4.6b.
5. Si dividiendo el PWM o PWM-DDR en 2, no es posible encontrar un LPF que permita la $f_{Sampling}$ deseada, es necesario aumentar la K al dividir el PWM. Se divide el PWM en K PWMs de $\frac{N}{K}$ -bit, permitiendo encontrar mejores prestaciones del LPF. Si con $K=2$ no se consigue el $t_{Settling}$ deseado, se aumenta a $K=3$.

De este modo es posible mejorar las prestaciones del LPF, pero se debe utilizar más salidas de la FPGA, más elementos analógicos y más área de implementación. Si al aumentar a $K=3$ aún no se obtiene la $f_{Sampling}$ deseada, se aumenta a $K=4$ y así sucesivamente hasta llegar a $K=N-1$. Para encontrar f_C a partir de O_{LPF} se realiza con la Eq. 4.18a o 4.18b si es con PWM o PWM-DDR respectivamente.

6. Una vez se halle f_C y O_{LPF} de los LPF, se obtiene V_{DAC} a partir de las Eq. 4.13 o 4.14, dependiendo si $\frac{N}{K}$ es un número entero o no.
7. Una vez obtenido V_{DAC} a partir de V_{LPF} se implementa el sumador analógico con AO. Para esto, se hallan las resistencias de asociadas a cada V_{LPF} basándose en la ponderación de la suma.

En primer lugar, se decide la resistencia de retroalimentación (R) y a partir de esta se obtienen R_M que es la resistencia que permite restar el equivalente a $\frac{LSB}{2}$ y las resistencias asociadas a los V_{LPF} , para esto se utilizan las Eq. 4.17d y 4.7 respectivamente.

A continuación, se obtienen A_{Suma} y A_{Resta} a partir de las Eq.4.17a y 4.17c respectivamente, para finalmente con la la Eq. 4.17d obtener la resistencia de ajuste R_T .

8. Los anteriores pasos funcionan cuando $K \leq N-1$, debido a que cuando $K=N$ no es necesaria la implementación de un LPF para generar V_{DAC} . Para generar V_{DAC} se realiza con la red R-2R mostrada en la Fig. 4.19.

4.5. Diseño de un ADC de 10-bit

Para ejemplificar la síntesis del ADC basado en SAR se realiza el diseño de un ADC de 10-bit con una FPGA de la familia Zynq 7000, con un $\Delta V_o=3.3$ V, $f_{clk}=100$ MHz y PWM-DDR. Se mostrará el diseño del ADC con las diferentes variantes presentadas en este capítulo con el objetivo de comparar el tipo de implementación y los resultados de los parámetros obtenidos.

En primer lugar, utilizando la optimización presentada en la sección 4.2, se divide el PWM-DDR de 10-bit en 2 PWM-DDR de 5-bit. Se obtuvo V_{DAC} a partir de la Eq. 4.13, debido a que $\frac{N}{2}$ es un número entero y se muestra en la Eq. 4.20. En la Fig. 4.26 se presenta el esquemático de la implementación.

$$V_{DAC} = V_1 + \frac{V_0}{2^5} - \frac{LSB}{2} \quad (4.20)$$

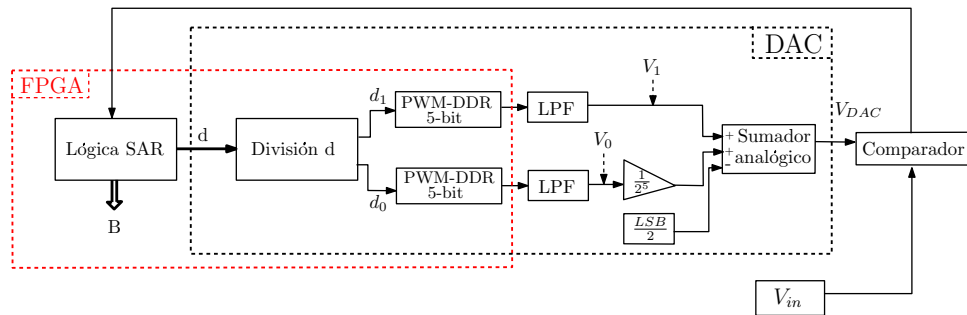


Figura 4.26: Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 2 PWM-DDR de 5-bit

Si se realiza la implementación dividiendo el PWM-DDR de 10-bit con $K=3$, es decir con 2 PWM-DDR de 4-bit y un PWM-DDR de 2-bit, obteniéndose $P_1=4$ -bit y $P_2=2$ -bit. V_{DAC} se rige por la Eq. 4.21 y se obtiene a partir de la Eq. 4.14, debido a que $\frac{N}{K}$ no es número entero. Se muestra el esquemático de la implementación en la Fig. 4.28.

$$V_{DAC} = V_2 + \frac{V_1}{2^2} + \frac{V_0}{2^6} - \frac{LSB}{2} \quad (4.21)$$

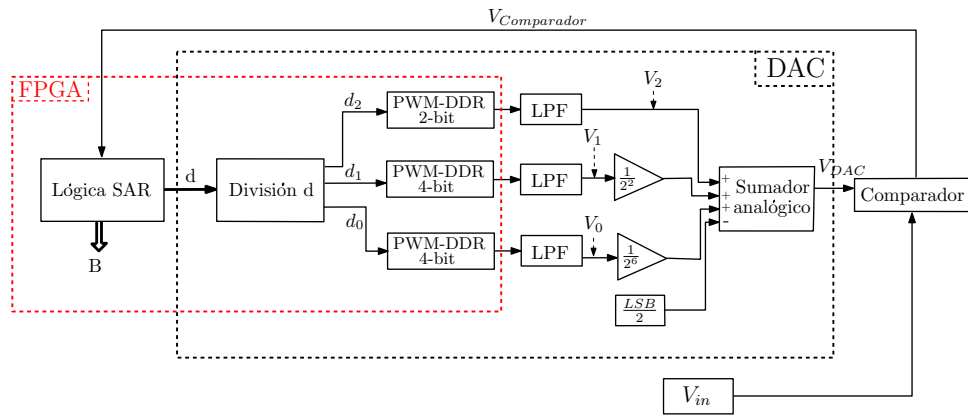


Figura 4.27: Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 2 PWM-DDR de 4-bit y un PWM-DDR de 2-bit

A continuación, se presenta en la Fig. 4.28 el diseño del ADC de 10-bit, dividiendo el PWM-DDR en 5 PWM-DDR de 2-bit. El V_{DAC} se rige por la Eq. 4.22.

$$V_{DAC} = V_{V_4} + \frac{V_3}{2^2} + \frac{V_2}{2^4} + \frac{V_1}{2^6} + \frac{V_0}{2^8} - \frac{LSB}{2} \quad (4.22)$$

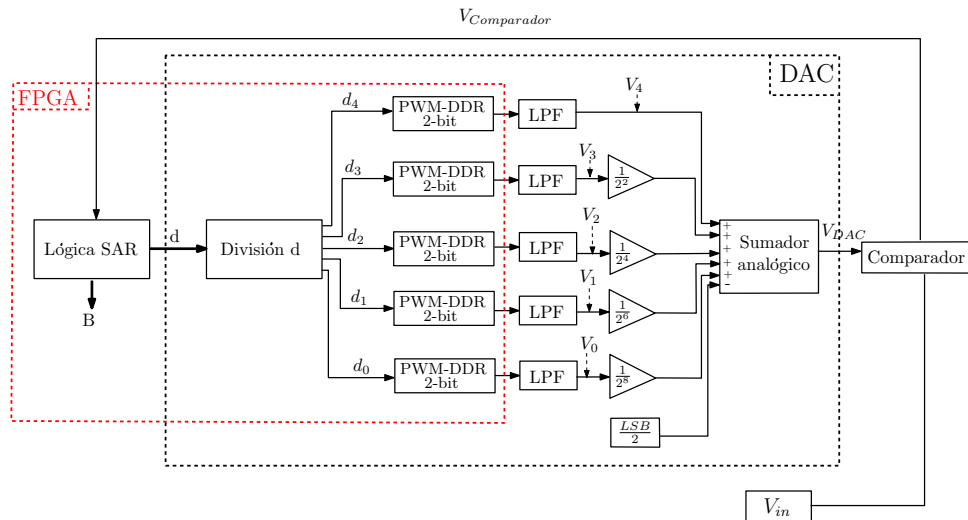


Figura 4.28: Esquemático de la implementación del ADC de 10-bit con el DAC implementado con 5 PWM-DDR de 2-bit

Por último, se realiza el diseño del ADC de 10-bit dividiendo el PWM-DDR en 10 PWM-DDR de 1-bit. Para obtener V_{DAC} no se realiza como se ha hecho hasta ahora, sino que se implementa una Red R-2R. Se muestra el esquemático del diseño en la Fig. 4.29.

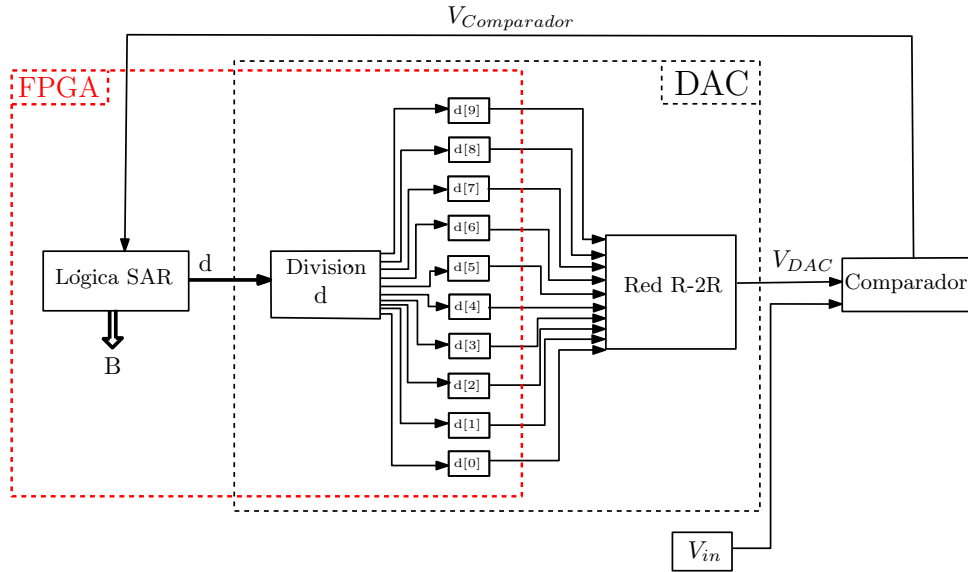


Figura 4.29: Esquemático de la implementación del ADC de 10-bit con el DAC implementado red R-2R

Una vez mostrados la forma de implementación de cada diseño, se obtienen los parámetros del LPF para implementar el ADC de 10-bit. Para efectos de comparación de los diseños, se halla f_C de los LPF con $O_{LPF} = 6$, mientras que para el diseño de 10 PWM-DDR de 1-bit, se tomara el tiempo de establecimiento de los AOs LT1364 y THS320. En la Fig. 4.30 se muestra los parámetros del LPF y para los PWM-DDR de 1-bit se muestra el $f_{Sampling}$. En la Tabla 4.5 se muestran los elementos analógicos necesarios para implementar el ADC, además de la cantidad de salidas de la FPGA.

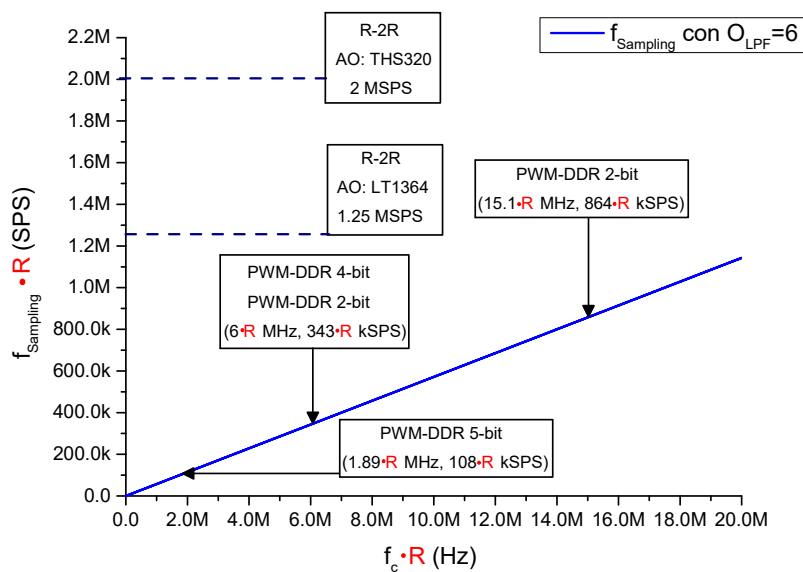


Figura 4.30: f_C vs. $f_{Sampling}$ con los parámetros hallados para un ADC 10-bit con diferentes diseños, $O_{LPF}=6$ y normalizado con $f_{CLK}=100$ MHz

4.6 Conclusiones

ADC=10-bit $f_{clk}=100$ MHz. PWM-DDR	$\frac{N}{2}$ -bit	$P_1=4$ -bit y $P_2=2$ -bit	X=2-bit	X=1-bit AO: LT1364/THS320
K	2	3	5	10
AO	7	10	16	1
R	17	24	38	22
C	12	18	30	0
Entradas/Salidas de la FPGA	3	4	6	11

Tabla 4.5: Comparación de las características de implementación del ADC de 10-bit con los diferentes diseños con $O_{LPF}=6$

La Fig. 4.30 y la Tabla 4.5 muestran la versatilidad que brinda las optimizaciones presentadas en este capítulo. Dependiendo de la $f_{Sampling}$ necesario el usuario escoge la forma de implementar el ADC. Si posee un número limitado de salidas de la FPGA, de ser posible se utiliza el ADC con $K=2$ y se aumenta los O_{LPF} , por el contrario, si no es posible alcanzar la $f_{Sampling}$ el usuario debe aumentar K aumentando los costes y área de implementación. Si el área del diseño es crucial el ADC con $K=N$ es el más indicado, debido a que no es necesario implementar ningún LPF o sumador analógico, en contra parte se utilizan más entradas de la FPGA.

4.6. Conclusiones

En este capítulo se presenta una optimización en el diseño del ADC basado en SAR, con el objetivo de aumentar la frecuencia de muestreo ($f_{Sampling}$) del ADC, aumentando la frecuencia de modulación del ancho de pulso (f_{PWM}), para así disminuir el rizado (ΔV_{DAC}) de la tensión de salida del DAC (V_{DAC}).

Se ha presentado una optimización en el diseño del DAC que consiste en dividir el PWM o PWM-DDR de N -bit en 2 de $\frac{N}{2}$ -bit. Cada PWM o PWM-DDR resultante junto a un filtro paso bajo (LPF) generan una tensión de salida del filtro (V_{LPF}). Estos V_{LPF} junto a un sumador analógico generan V_{DAC} . El aumento de la f_{PWM} hace que el LPF atenué más, por lo que es posible escoger una frecuencia de corte (f_C) y el orden del LPF (O_{LPF}) que cumplan con la condición de $\Delta V_{DAC} < LSB$ y tenga un menor tiempo de establecimiento ($t_{Settling}$).

Se presentó una optimización que consiste en dividir el PWM de N-bit en K PWMs de $\frac{N}{K}$ -bit. Los PWMs resultantes junto a los LPFs y un sumador analógico generan V_{DAC} . La ponderación de cada V_{LPF} para obtener V_{DAC} viene regida por la cantidad de bits de los PWMs resultante. A medida que K aumenta, es posible escoger un LPF con mejores prestaciones, haciendo que $f_{Sampling}$ aumente. El inconveniente radica en que, al aumentar K, aumentan las entradas/salidas necesarias en la FPGA, además de aumentar el coste y área de implementación del circuito.

La versatilidad del diseño se ve mejorado gracias a las optimizaciones presentadas en este capítulo. Si se necesita una $f_{Sampling}$ en específico, es posible implementar el ADC de diferentes maneras según las necesidades del diseño. De no ser posible alcanzar el $t_{Settling}$ necesario, el usuario tiene la posibilidad de aumentar K y mejorar las prestaciones del LPF.

La escalabilidad del diseño también se ve mejorada con las optimizaciones presentadas. De ser necesario el aumento de la resolución del ADC, sería necesario agregar un PWM o PWM-DDR con los bits adicionales, modificar la f_C de los LPF y utilizar salida más en la FPGA, por último, se agrega una entrada al sumador analógico.

5. Capítulo 5

5.1. Resumen

En este capítulo se presenta una optimización para aumentar la frecuencia de muestreo ($f_{Sampling}$) del conversor analógico digital (ADC) basado en el registro de aproximaciones sucesivas (SAR), donde se sustituye el conversor digital analógico (DAC) por un modulador de anchura de pulsos (PWM) y un filtro pasa bajo (LPF). Para esto, se reemplaza el PWM por un 'Distributed Dutty cycle PWM' (PWM-DDC) con el objetivo de disminuir el rizado del DAC (ΔV_{DAC}) y a su vez aumentar las prestaciones del ADC. Se estudian las ventajas y desventajas del diseño, como también los límites del mismo.

5.2. PWM con ciclo de trabajo distribuido

En capítulos anteriores se implementó el PWM de N-bit con un contador ascendente con cuentas que van desde 0 hasta $2^N - 1$. En el capítulo 2 este contador incrementaba en 1 el registro de cuentas con el flanco ascendente del reloj, y en el capítulo 3 el contador incrementaba en 1 tanto en flanco ascendente como descendente aprovechando el registro DDR, generando el PWM-DDR.

En el PWM implementado en los capítulos anteriores, si $D = \frac{1}{2}$, V_{PWM} se mantenía en V_{OH} la mitad del periodo del PWM (T_{PWM}) y la otra mitad estaba en V_{OLH} , como se muestra en la Fig. 5.1a. Para disminuir ΔV_{DAC} generado por el PWM y el LPF se propone reemplazar el PWM por el PWM-DDC implementado en [93–95]. La diferencia entre el PWM-DDC y el PWM radica en la forma de distribuir la tensión del PWM-DDC ($V_{PWM-DDC}$) y V_{PWM} están en V_{OH} y V_{OL} , durante el tiempo en que el contador realiza las 2^N cuentas. Para obtener el ciclo de trabajo del PWM-DDC (D_{DDC}) se utiliza la Eq. 5.1.

$$D_{DDC} = \frac{d}{2^N} \tag{5.1}$$

$$0 \leq d \leq 2^N - 1$$

Con el PWM-DDC, el tiempo en que V_{DDC} se encuentra en V_{OH} y V_{OL} se distribuye a lo largo del tiempo en que se realizan las 2^N cuentas. V_{DDC} cambia entre V_{OH} y V_{OL} repetidamente hasta completar el número de veces que V_{DDC} deba estar en V_{OH} , acorde al valor indicado por d . Se muestra en la Fig. 5.1a un PWM de 3-bit y en la Fig. 5.1b un PWM-DDC de 3-bit, ambos con D y D_{DDC} igual $\frac{1}{2}$.

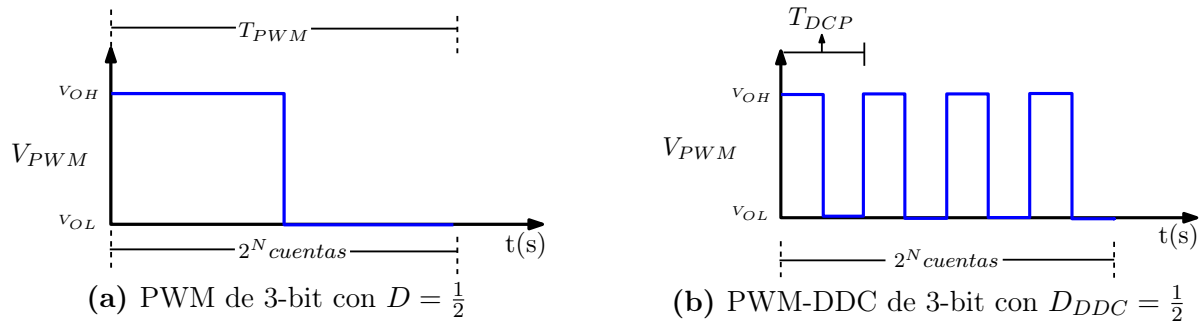


Figura 5.1: PWM y PWM-DDC de 3-bit con D y D_{DDC} igual a $\frac{1}{2}$

Al implementar PWM-DDC, cuando varia D lo que se está modificando es la frecuencia de modulación (f_{DDC}) del PWM-DDC [96–100]. Se muestra un PWM-DDC de 3-bit en las figuras 5.2a, 5.2b y 5.2c con un $D_{DDC} = \frac{2}{8}$, $\frac{3}{8}$ y $\frac{4}{8}$ respectivamente.

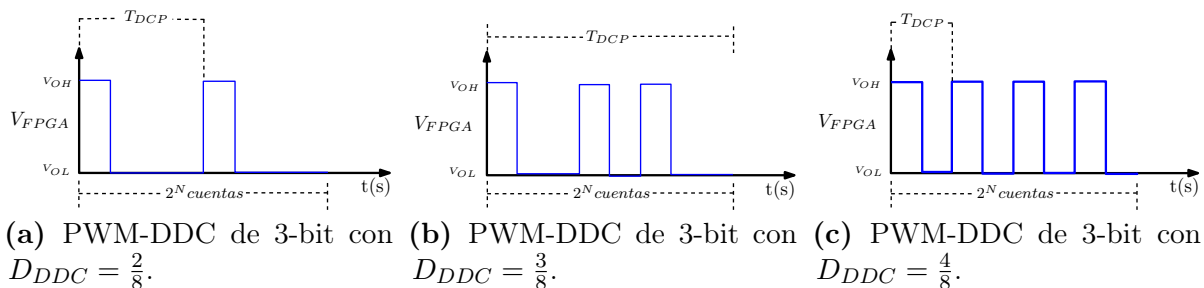


Figura 5.2: PWM-DDC de 3-bit variando D_{DDC} .

Partiendo desde $D_{DDC} = \frac{1}{2}$ donde la f_{DDC} es máxima, f_{DDC} disminuye a medida que lo hace d , llegando a $f_{DDC} = f_{PWM}$ cuando $D = \frac{d}{2^N} = \frac{1}{2^N}$. Esto último también sucede cuando d aumenta partiendo desde $D_{DDC} = \frac{1}{2}$ hasta llegar a $D_{DDC} = \frac{2^N-1}{2^N}$, donde $f_{DDC} = f_{PWM}$.

En la Fig. 5.3 se muestra la V_{LPF} con un PWM y un PWM-DDC de 3-bit con un D y un D_{DDC} de $\frac{3}{8}$. El rizado del V_{LPF} generado por el PWM-DDC es mucho menor que el generado con un PWM. Esta mejora permite escoger una f_C y O_{LPF} que cumpla con la condición de la Eq. 2.6, además de una $f_{Sampling}$ mayor.

A partir de los PWM-DDC presentados en la Fig. 5.1 y la Fig. 5.2 es posible ver que para los D_{DDC} de $\frac{2}{8}$ y $\frac{4}{8}$ el V_{DDC} puede distribuirse uniformemente a lo largo de las 2^N cuentas. Esta distribución uniforme de V_{DDC} hace que f_{DDC} aumente. En el caso de $D_{DDC} = \frac{3}{8}$ esta distribución no es uniforme, por lo que $f_{DDC} = f_{PWM}$. Sin embargo, aunque $f_{DDC} = f_{PWM}$ se puede apreciar en la Fig. 5.3 que ΔV_{LPF} generado por el PWM-DDC es menor.

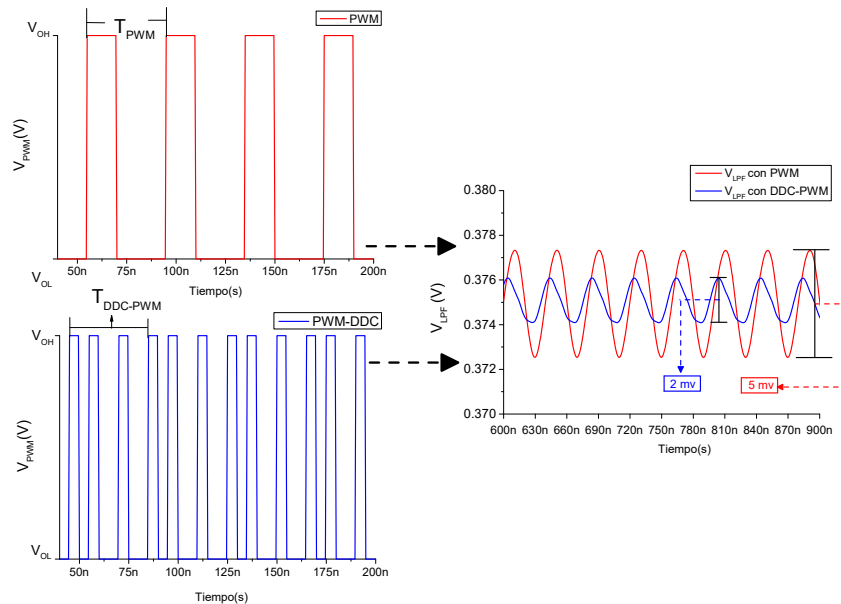


Figura 5.3: V_{LPF} con un PWM y PWM-DDC de 3-bit con D y D_{DDC} de $\frac{3}{8}$, con LPF de $O_{LPF}=3$ y $f_C=4$ MHz

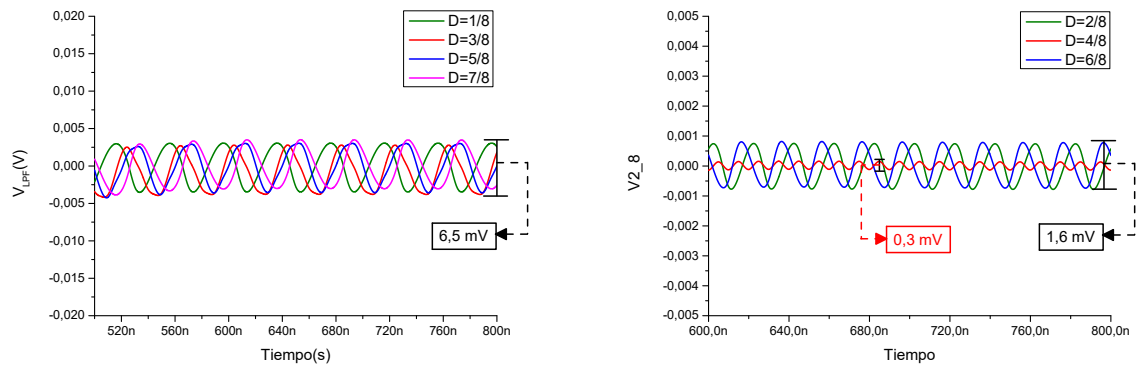
En la Fig. 5.4a se muestra V_{LPF} a partir de un PWM-DDC de 3-bit y un LPF con $f_C=4$ MHz y $O_{LPF}=3$ para los d impares, los cuales generan $f_{DDC}=f_{PWM}$ como se explicó anteriormente. Para obtener ΔV_{LPF} se sigue la Eq. 5.2, donde a cada V_{LPF} se le resta su parte continua ($\overline{V_{LPF}}$). Se aprecia que con el PWM-DDC y los d impares se genera un mismo ΔV_{LPF} , a diferencia del PWM donde cada d generaba un ΔV_{LPF} diferente.

$$V_{LPF}(t) = \overline{V_{LPF}} + \Delta V_{LPF}(t) \quad (5.2)$$

$$\overline{V_{LPF}} = D \cdot \Delta V_O$$

Se realiza el mismo procedimiento para obtener ΔV_{LPF} para los d pares mostrados en la Fig. 5.4b. Se aprecia una diferencia en los ΔV_{LPF} resultantes al variar los d pares. Esto es debido a que al poder distribuir uniformemente V_{DDC} , f_{DDC} aumenta. En la Fig. 5.4b se

aprecia que para $D = \frac{2}{8}$ y $D = \frac{6}{8}$ se presenta un mismo ΔV_{LPF} . Lo anterior se debe a que f_{DDC} es la misma para los d mencionados, ya que la distribución de V_{DDC} es simétrica.



(a) ΔV_{LPF} con un PWM-DDC de 3-bit con d impares (b) ΔV_{LPF} con un PWM-DDC de 3-bit con d pares

Figura 5.4: ΔV_{LPF} con un PWM-DDC de 3-bit con d pares e impares, con LPF de $O_{LPF} = 3$ y $f_C = 4$ MHz

Otra cuestión a tener en cuenta en la implementación del PWM-DDC para los d impares es el comportamiento de V_{DDC} a lo largo del tiempo en que se realizan las 2^N cuentas. En la Fig. 5.5 se muestra V_{LPF} para 2 PWM-DDC de 3-bits cada uno con $D = \frac{3}{8}$, distribuyendo V_{DDC} de manera diferente en cada PWM-DDC. Se puede apreciar en la Fig. 5.5 que V_{LPF} es el mismo para las dos distribuciones, solo se ve afectado en su fase.

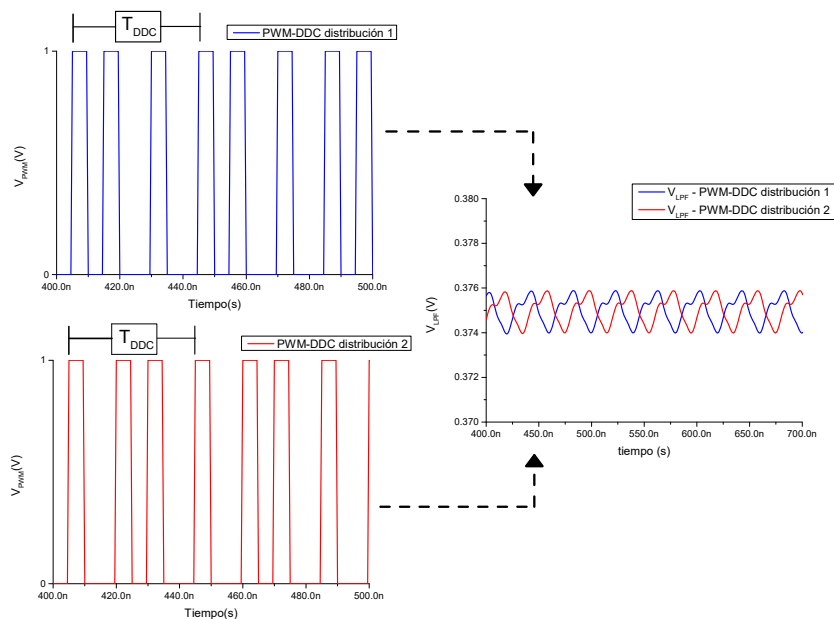


Figura 5.5: V_{LPF} con un PWM-DDC 3-bit y $D = \frac{3}{8}$, con LPF de $O_{LPF}=3$ y $f_C=4$ MHz

5.3. Obtención de los parámetros del LPF con PWM-DDC

En el capítulo 2 con el objetivo de representar la señal del PWM y analizar el comportamiento del PWM, se obtuvo los coeficientes de la serie de Fourier y se vuelven a mostrar en la Eq. 5.3 donde D varía entre 0 y 1.

$$\begin{aligned} A_0 &= D \cdot \Delta V_O = \frac{d}{2^N} \cdot \Delta V_O \\ B_n &= 0 \\ A_n &= 2 \cdot \frac{\Delta V_O}{n\pi} \left[\sin(n\pi D) \right] \end{aligned} \quad (5.3)$$

También se halló el D que produce el máximo ΔV_{LPF} a partir de la energía contenida en una señal y el resultado fue que $D = \frac{1}{2}$ generaba el mayor rizado. Como se explicó en la sección 5.2, debido a que el PWM-DDC distribuye el V_{DDC} , f_{DDC} es máxima cuando $D = \frac{1}{2}$, por lo que a diferencia del PWM, en el PWM-DDC el ciclo de trabajo que genera el menor ΔV_{LPF} es $D = \frac{1}{2}$.

Se debe tener en cuenta que con los d pares, la f_{DDC} es mayor que la f_{PWM} , por lo que cualquier d par genera un menor ΔV_{LPF} que los d impares. En cuanto a los d impares como se mostró en la Fig. 5.4a generan el mismo ΔV_{LPF} . Debido a lo anterior se evalúa D con $d = 1$ en la Eq. 5.4 para el primer armónico ($n = 1$).

$$A_n = 2 \cdot \frac{\Delta V_O}{n\pi} \left[\sin(n\pi D) \right] \quad (5.4)$$

$$A_{1-DDC(max)} = 2 \cdot \frac{\Delta V_O}{\pi} \left[\sin\left(\frac{\pi}{2^N}\right) \right]$$

El ΔV_{LPF} generado a la salida del LPF, se obtiene filtrando el primer armónico mostrado en la Eq. 5.5 a partir de la Eq. 5.4, donde A_{LPF} representa la ganancia en magnitud de f_{DDC} . Por lo tanto, utilizando la ecuación del *LSB* es decir la Eq. 2.6 y la Eq. 5.4, se genera en la Eq. 5.6 el límite para la ganancia de la magnitud del LPF a f_{DDC} .

$$\Delta V_{LPF} > 2 \cdot A_{LPF}(f_{DDC}) \cdot A_{1-DDC}(max) \quad (5.5)$$

$$\frac{\Delta V_O}{2^N} > 4 \cdot A_{LPF}(f_{DDC}) \cdot \frac{\Delta V_O}{\pi} \left[\sin\left(\frac{\pi}{2^N}\right) \right]$$

$$A_{LPF}(f_{DDC}) < \frac{\pi}{4} \cdot \frac{1}{2^N} \cdot \frac{1}{\sin\left(\frac{\pi}{2^N}\right)} \quad (5.6)$$

Se realiza el mismo procedimiento utilizado en el capítulo 2 para obtener f_C a partir de los parámetros de la FPGA y O_{LPF} . Para esto se compara la Eq. 5.6 en escala logarítmica, con la atenuación generada por el LPF a una f_{DDC} mostrada en la Eq. 5.7a. Para que ΔV_{LPF} sea menor a LSB, se debe cumplir que $A_{LPF}(f_{DDC}) < A_{LPF}(max)$. Se combinan la Eq. 5.7a y 5.7b para generar la Eq. 5.8.

$$A_{LPF}(f_{DDC}) = -20 \cdot O_{LPF} \cdot \log_{10}\left(\frac{f_{DDC}}{f_C}\right) = -20 \cdot O_{LPF} \cdot \log_{10}\left(\frac{f_{clk}}{2^N \cdot f_C}\right) \quad (5.7a)$$

$$A_{LPF}(max) = 20 \cdot \log_{10}\left(\frac{\pi}{4} \cdot \frac{1}{2^N} \cdot \frac{1}{\sin\left(\frac{\pi}{2^N}\right)}\right) \quad (5.7b)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{1}{\left[\sin\left(\frac{\pi}{2^N}\right)\right]^{\left(\frac{1}{O_{LPF}}\right)}} \cdot \frac{O_{LPF}\sqrt{\pi}}{2^{\left(\frac{2+N+N \cdot O_{LPF}}{O_{LPF}}\right)}} \quad (5.8)$$

Para simplificar la expresión de la Eq. 5.8, se utiliza la aproximación para ángulos pequeños del seno. Esta aproximación indica que $\sin x \approx x$ tiene una buena precisión a medida que el ángulo tiende a cero. De esta manera se obtiene la Eq. 5.9 para obtener f_C a partir de O_{LPF} y N.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{1}{2^{\left(\frac{2+N \cdot O_{LPF}}{O_{LPF}}\right)}} \quad (5.9)$$

La Eq. 5.9 se obtiene para hallar f_C para un PWM-DDC sin implementarlo con DDR. Cuando se implementa el PWM-DDC con DDR (PWM-DDC-DDR) se utiliza la Eq. 5.10.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2}{2^{\left(\frac{2+N \cdot O_{LPF}}{O_{LPF}}\right)}} \quad (5.10)$$

Como se propuso en el capítulo 4, donde se dividió el PWM en K PWMs de $\frac{N}{K}$ -bit con el objetivo de aumentar la f_{PWM} y poder obtener mayores $f_{Sampling}$. Para el PWM-DDC se puede realizar la misma optimización con el objetivo de obtener mejores prestaciones, para esto se modifica la Eq. 5.4 y se obtiene la Eq. 5.11. Realizando el mismo procedimiento explicado para hallar f_C , se genera la Eq. 5.12.

$$A_{1-DDC}(max) = 2 \cdot \frac{\Delta V_O}{\pi} \left[\sin\left(\frac{\pi}{2\left(\frac{N}{K}\right)}\right) \right] \quad (5.11)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{1}{\left[\sin\left(\frac{\pi}{2\left(\frac{N}{K}\right)}\right) \right] \left(\frac{1}{O_{LPF}}\right)} \cdot \frac{O_{LPF}\sqrt{\pi}}{2^{\left(\frac{N}{K} + \frac{N+2}{O_{LPF}}\right)}} \quad (5.12)$$

Si se realiza la aproximación de ángulos pequeños del seno, donde $\sin x \approx x$, se obtiene la Eq. 5.13a para el PWM-DDC dividido en K PWMs de $\frac{N}{K}$ -bit, mientras que en la Eq. 5.13b para el PWM-DDC-DDR.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2^{\left(\frac{N}{O_{LPF} \cdot K}\right)}}{2^{\left(\frac{N+2}{O_{LPF}} + \frac{N}{K}\right)}} \quad (5.13a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2^{\left(1 + \frac{N}{O_{LPF} \cdot K}\right)}}{2^{\left(\frac{N+2}{O_{LPF}} + \frac{N}{K}\right)}} \quad (5.13b)$$

En los capítulos anteriores se obtuvo los parámetros f_C y O_{LPF} del LPF para un ADC de 6-bit, teniendo $\Delta V_O = 1$ V y $f_{clk} = 100$ MHz. Para comparar las optimizaciones presentadas, se obtienen los parámetros con las expresiones para f_C halladas en cada capítulo, además de las expresiones para los PWM-DDC y PWM-DDC-DDR.

En primer lugar, se obtienen los parámetros del LPF para $O_{LPF} = 4$ para un PWM, PWM-DDC y sus variantes con DDR. Se muestran en la Fig. 5.6, donde se puede apreciar que al implementar el PWM-DDC ya sea con DDR o sin él, hay un aumento del $f_{Sampling}$. En

la Tabla 5.1 se presentan los componentes analógicos necesarios para la implementación del ADC.

ADC=6-bit	PWM	PWM-DDC	PWM-DDR	PWM-DDC-DDR
K			1	
O_{LPF}			4	
Entradas/Salidas de la FPGA			2	
Número de R			8	
Número de C			4	
Número de AO			3	

Tabla 5.1: Comparación de las características de implementación del LPF para un ADC de 6-bit con PWM y PWM-DDC de 6-bit

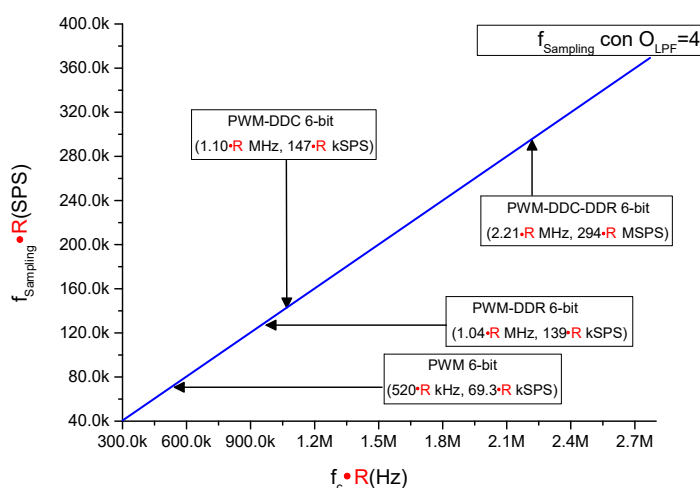


Figura 5.6: f_c vs. $f_{Sampling}$ con los parámetros hallados para diferentes diseños para ADC de 6-bit con $O_{LPF}=4$ normalizado con $f_{CLK}=100$ MHz.

En el capítulo 4 se presentó una optimización que permite reducir la f_{PWM} al dividir el PWM en K PWMs de $\frac{N}{K}$ -bit. Con el PWM-DDC también es posible realizar dicha optimización con el objetivo de mejorar las prestaciones del ADC. En la Fig. 5.7 se muestra los parámetros del LPF para $O_{LPF}=4$ cuando se divide el PWM y el PWM-DDC con K=2 y K=3, presentando mejores prestaciones del LPF. En la Tabla 5.2 se muestran los elementos analógicos necesarios para la implementación del PWM o PWM-DDC con K=2 y K=3.

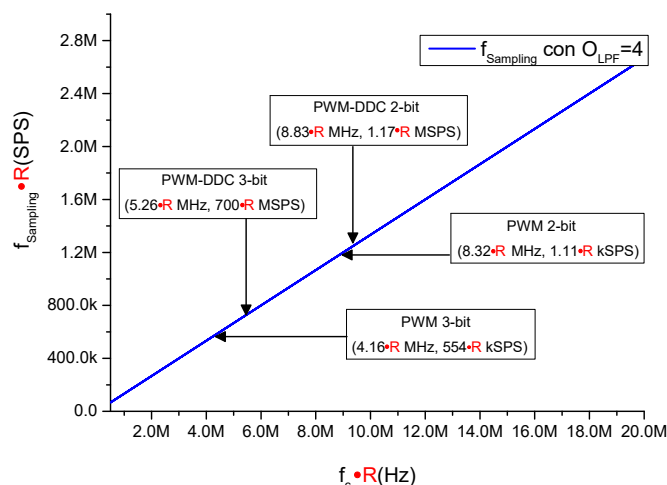


Figura 5.7: f_c vs. f_{Sampling} con los parámetros hallados para un ADC de 6-bit con diferentes diseños con $O_{LPF}=4$, dividiendo el PWM y PWM-DDC en $K=2$ y $K=3$

ADC=6-bit	PWM 3-bit	PWM-DDC 3-bit	PWM 2-bit	PWM-DDC 2-bit
K		2		3
O_{LPF}			4	
Entradas/Salidas de la FPGA		3		4
Número de R		13		18
Número de C		8		12
Número de AO		5		7

Tabla 5.2: Comparación de las características de implementación del LPF para un ADC de 6-bit con PWMs y PWM-DDC de 3-bit.

Para obtener el t_{Settling} se hace a través de la respuesta al escalón normalizada del la Fig. 2.16 del capítulo 2. En la Tabla 5.3 se muestra el tiempo de establecimiento del LPF normalizado a diferentes O_{LPF} , por lo que para obtener el t_{Settling} de cada O_{LPF} , se divide el t_{Settling} normalizado por f_c del O_{LPF} correspondiente.

O_{LPF}	2	3	4	5	6	7	8	9	10
t_{Settling} normalizado	0.65	1.00	1.25	1.52	1.75	1.90	2.10	2.21	2.32

Tabla 5.3: t_{Settling} normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .

Existe una diferencia al encontrar las f_c a partir de los O_{LPF} cuando se implementa con PWM o PWM-DDC. Para el PWM, al aumentar el O_{LPF} se obtiene una f_c con menor t_{Settling} , mientras que con el PWM-DDC esto no sucede. Para ejemplificar lo comentado,

se obtienen los parámetros del LPF para el ADC de 6-bit con PWM-DDC de 6-bit y se muestra en la la Fig. 5.8. A medida que O_{LPF} va aumentando la $f_{Sampling}$ disminuye, siendo el $O_{LPF}=2$ el que tiene las mejores prestaciones. En la Fig. 5.9 se presenta los parámetros del LPF para un ADC de 6-bit con 2 PWM-DDC de 3-bit. En este caso el O_{LPF} que genera el mayor $f_{Sampling}$ es $O_{LPF}=4$.

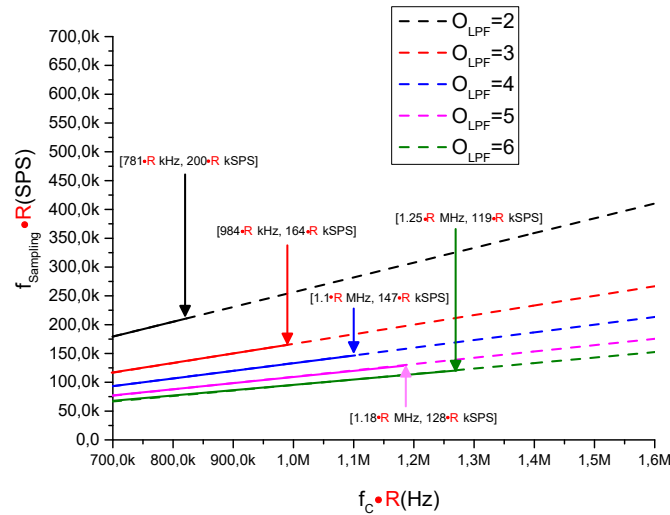


Figura 5.8: f_c vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 6-bit con PWM-DDC normalizado a $f_{CLK}=100$ MHz.

Basándose en lo anterior, a la hora de diseñar el ADC de N-bit con PWM-DDC, hay que realizar los cálculos de los parámetros del LPF con los diferentes O_{LPF} con el objetivo de encontrar el O_{LPF} y f_c que generen el mayor $f_{Sampling}$, a diferencia de con el PWM que siempre a mayor O_{LPF} la $f_{Sampling}$ mejoraba.

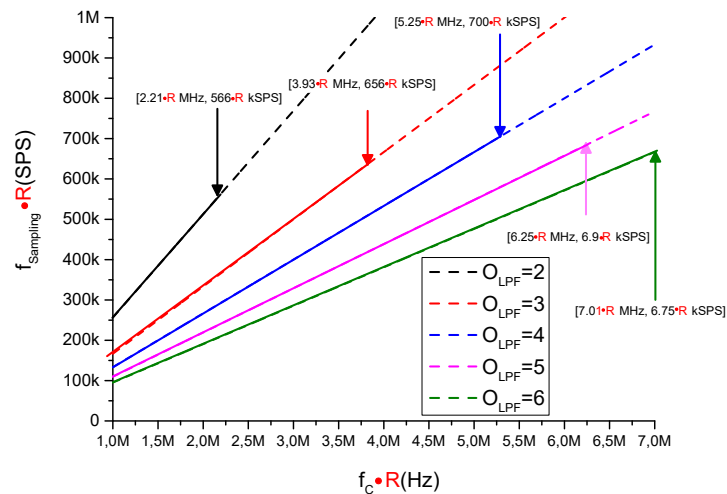
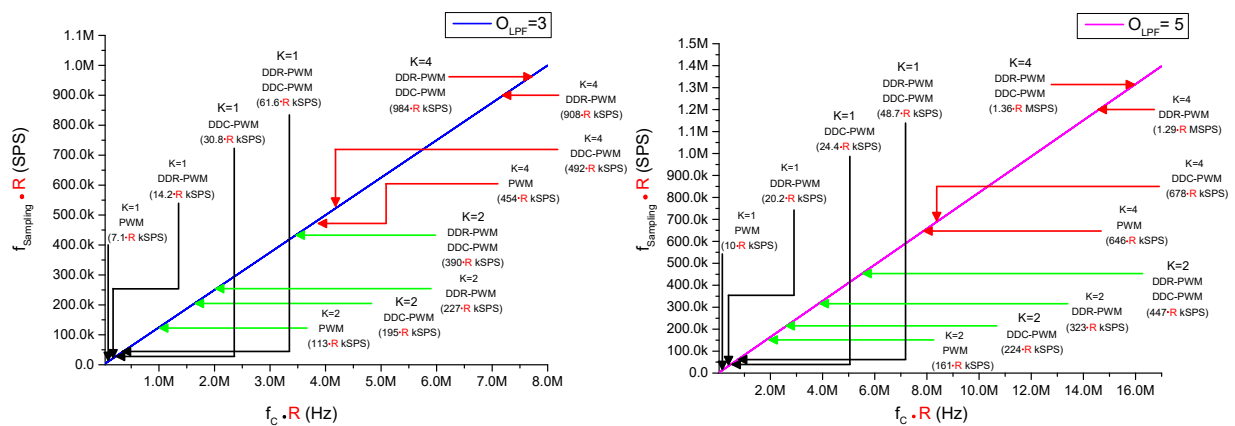


Figura 5.9: f_c vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 6-bit con 2 PWM-DDC de 3-bit normalizado a $f_{CLK}=100$ MHz.

5.4. ADC de 8-bit con PWM-DDC para diferentes optimizaciones

En los anteriores capítulos se implementó un ADC de 8-bit, con una FPGA de la familia Zynq 7000 de Xilinx, con $\Delta V_O=3.3$ V y $f_{clk}=100$ MHz, teniendo en cuenta que puede ser extrapolable a otras familias de FPGA. Se realiza el diseño de un ADC basado en SAR reemplazando el DAC con un LPF y PWM-DDC, con las formas de implementación mostradas en capítulos anteriores.

Con el objetivo de comparar los parámetros hallados con cada optimización, Se hallará f_C para un PWM, PWM-DDC y sus implementaciones con DDR. Se muestra en la Fig. 5.10a los parámetros del LPF para $O_{LPF}=3$ y en la Fig. 5.10a para $O_{LPF}=5$. Como se puede apreciar en las Fig. 5.10a y 5.10b comparando el PWM, PWM-DDR y el PWM-DDC-DDR, se obtienen mejores prestaciones con el PWM-DDC-DDR para cada K .



(a) f_C Vs. $f_{Sampling}$ para un ADC 8-bit con $O_{LPF}=3$ con diferentes implementaciones del PWM

(b) f_C Vs. $f_{Sampling}$ para un ADC 8-bit con $O_{LPF}=5$ con diferentes implementaciones del PWM

Figura 5.10: f_C Vs. $f_{Sampling}$ para un ADC 8-bit con diferentes implementaciones del PWM

Ahora bien, algo importante a tener en cuenta es que, si se compara el PWM-DDR y el PWM-DDC (sin DDR) los resultados son bastante similares. Esto permite que para FPGAs que no tengan la posibilidad de implementar el DDR, el implementar el PWM-DDC sea la mejor opción para aumentar la $f_{Sampling}$.

5.4.1. Simulación de un ADC de 8-bit con PWM-DDC-DDR de $\frac{N}{2}$ -bit

Se realizará la simulación del ADC de 8-bit basado en SAR reemplazando el DAC con dos LPF y dos PWM-DDC-DDR de $\frac{N}{2}$ -bit para una $f_{Sampling}=230$ kSPS. En primer lugar, se obtienen los f_C a partir de los O_{LPF} a partir de la Eq. 5.13b. En la Fig. 5.11 se muestra f_C Vs. $f_{Sampling}$ para diferentes O_{LPF} , además del límite donde se cumple $f_{Sampling}=230$ kSPS. A partir de la Fig. 5.11 se obtiene que con $f_C=1.56$ MHz y $O_{LPF}=2$ es posible implementar el ADC. Se obtiene la Tabla 5.4 con los parámetros del ADC y el LPF. Para obtener V_{DAC} se sigue la Eq. 4.13 del capítulo 4 y se muestra en la Eq. 5.14.

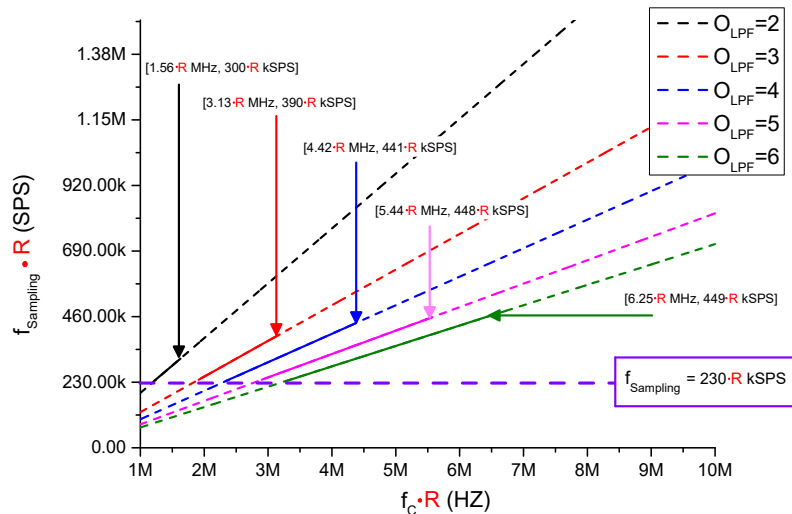


Figura 5.11: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=230$ kSPS con 2 PWM-DDC-DDR de 4-bit normalizado con $f_{CLK}=100$ MHz.

ADC	O_{LPF}	R	C	Número AO	$t_{Settling}$ del LPF	$t_{Conversion}$ del ADC	$f_{Sampling}$
8-bit	2	9	6	3	543 ns	4.64 μ s	230 kSPS

Tabla 5.4: Características del ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit.

$$V_{DAC} = V_1 + \frac{V_0}{2^4} - \frac{LSB}{2} \quad (5.14)$$

5.4.2. Implementación de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit

A continuación, se realiza la implementación del ADC de 8-bit con $f_{Sampling}=230$ kSPS con 2 PWM-DDC-DDR de 4-bit. Para la implementación se utilizará el AO LT1364 de Analog devices tanto para los LPF como para el sumador analógico. El esquemático de la implementación se presenta en la Fig. 5.12, donde se adiciona un DAC para visualizar B obtenido por el ADC. Al implementar los LPF y el sumador analógico se tuvo en cuenta las restricciones mencionadas en la sección 2.2.5 del capítulo 2, como los planos de masa, condensadores de desacople, el menor cableado posible etc.

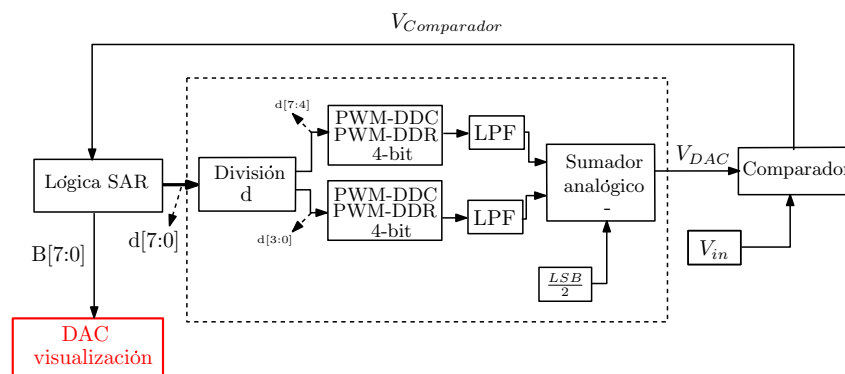
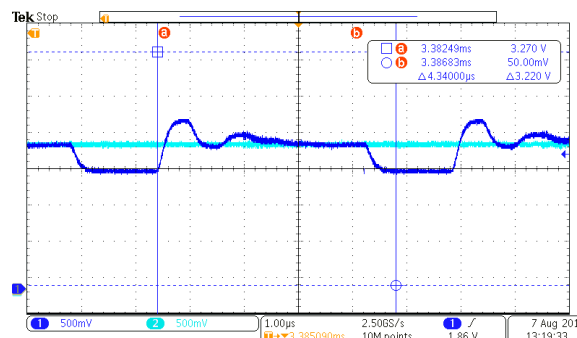
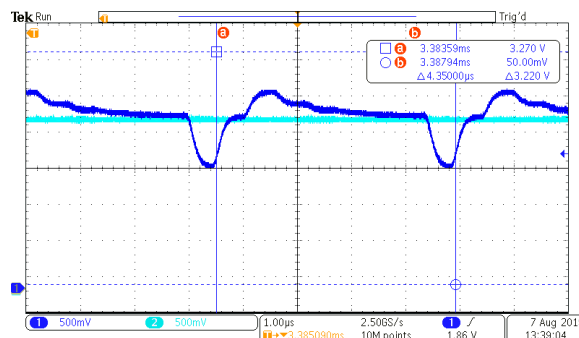


Figura 5.12: Esquemático de la implementación del ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit

En la Fig. 5.18a se muestra el comportamiento de V_{DAC} ante una entrada V_{in} continua de 1,7 V, mientras en la Fig. 5.18b con $V_{in} = 2,2$ V. Para comprobar B con los todos posibles valores de V_{in} , se utiliza una señal triangular que va desde 0 V hasta 3.3 V y se muestran en la Fig. 5.14 donde B es la señal rosa y V_{in} triangular la señal azul marino.



(a) V_{DAC} de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit para $V_{in}=1.7$ V.



(b) V_{DAC} de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit para $V_{in}=2.2$ V.

Figura 5.13: V_{DAC} de un ADC de 8-bit con 2 PWM-DDC-DDR de 4-bit para V_{in} continuo

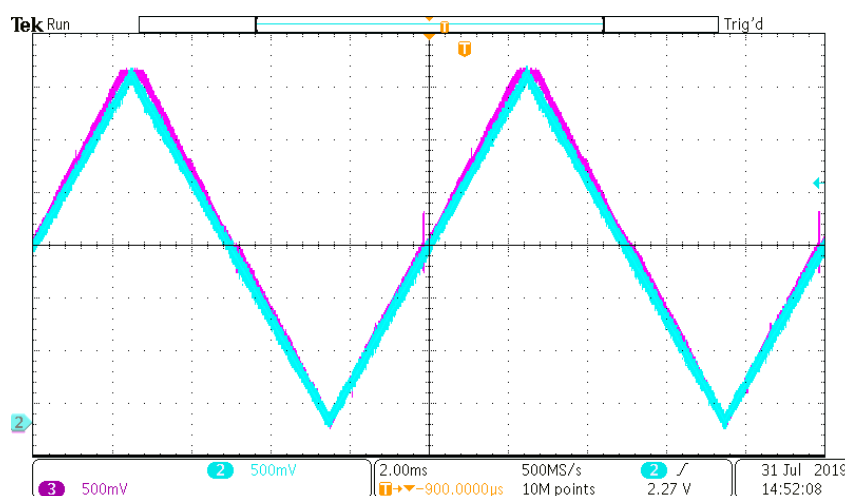


Figura 5.14: V_{in} y B obtenido por un ADC de 8-bit con PWM-DDC-DDR de 4-bit

5.4.3. Simulación de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit

Ahora se realiza la simulación de un ADC con 4 PWM-DDC-DDR de 2-bit y $f_{Sampling}=500$ kSPS. En primer lugar, se obtiene f_C para diferentes O_{LPF} a partir de la Eq. 5.13b y se muestran en la Fig 5.15. Se escoge $f_C=3.13$ MHz y $O_{LPF}=2$ y se muestra las características del ADC implementado en la Tabla 5.5.

Se obtiene V_{DAC} a partir de la Eq. 4.13 del capítulo 4 y se muestra en la Eq. 5.15. En la Fig. 5.16 se presenta el comportamiento de V_{DAC} antes una entrada continua de V_{in} .

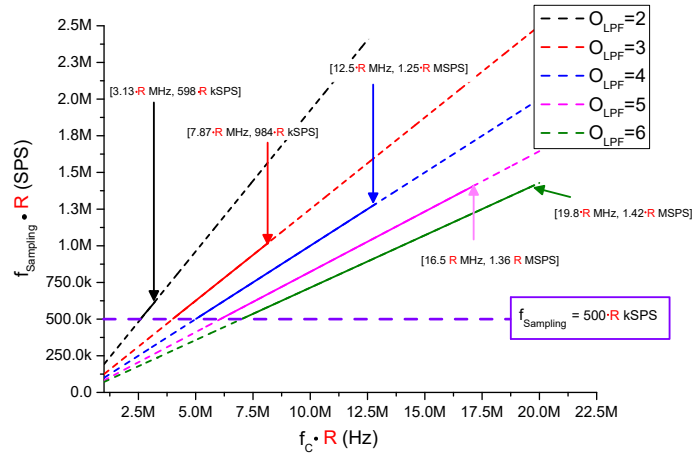


Figura 5.15: f_c vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con $f_{Sampling}=500$ kSPS con 4 PWM-DDC-DDR de 2-bit normalizado con $f_{CLK}=100$ MHz

ADC	O_{LPF}	R	C	Número AO	$t_{Settling}$ del LPF	$t_{Conversion}$ del ADC	$f_{Sampling}$
8-bit	2	15	8	5	250 ns	2 μ s	500 kSPS

Tabla 5.5: Características del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.

$$V_{DAC} = V_3 + \frac{V_2}{2^2} + \frac{V_1}{2^4} + \frac{V_0}{2^6} - \frac{LSB}{2} \tag{5.15}$$

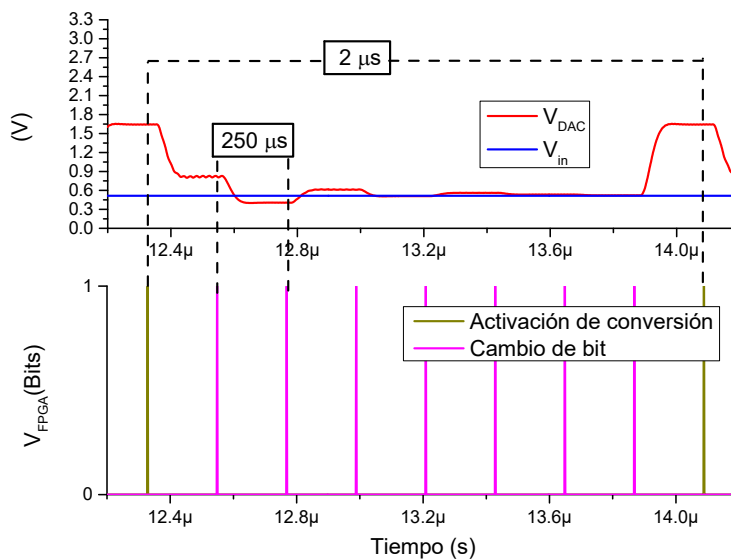


Figura 5.16: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.

5.4.4. Implementación de un ADC de 8-bit con $K=4$ y PWM-DDC-DDR de 2-bit

A continuación, se mostrara la implementación del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit. En la Fig. 5.17 se muestra el esquemático de la implementación. En la Fig. 5.18a se muestra el comportamiento de V_{DAC} ante un V_{in} de 1 V y en la Fig. 5.18b se muestra para V_{in} de 1.8 V.

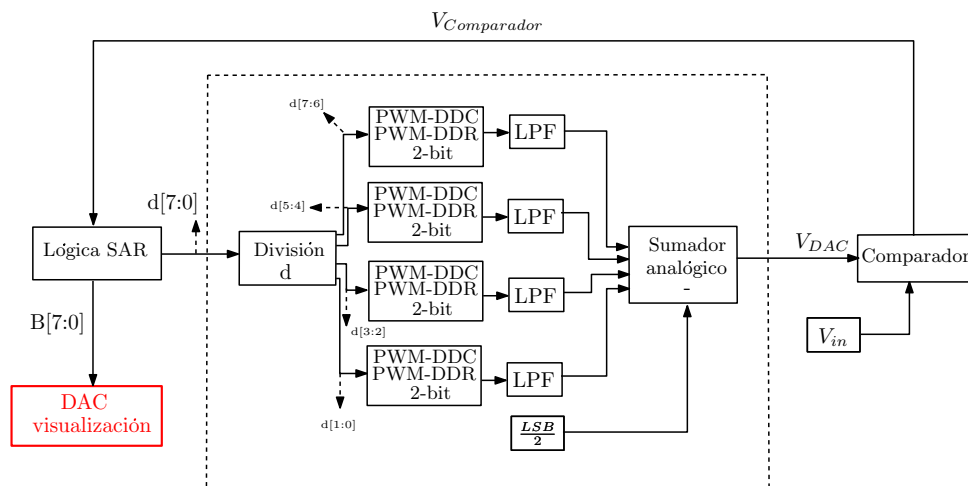
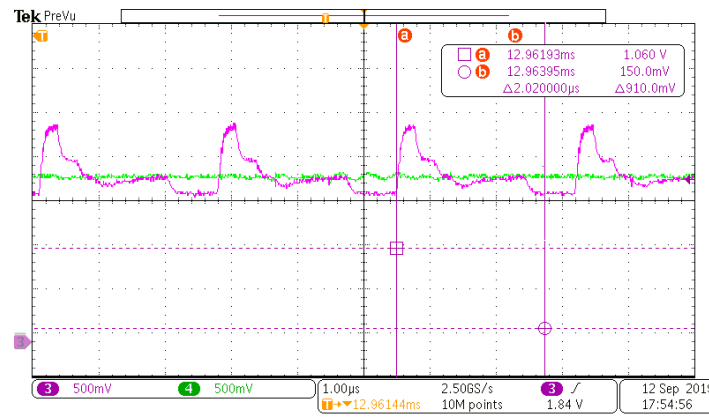


Figura 5.17: Esquemático de la implementación del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit

Para obtener todos los posibles valores de B se utilizó una señal de diente de sierra que va desde 0 V hasta 3.3 V. Una vez el ADC genere B , este es enviado a un DAC adicional para poder ser mostrado en el osciloscopio. Se muestra B obtenida a partir de V_{in} en la Fig. 5.19, donde V_{in} es la señal azul y B la señal rosa.



(a) V_{DAC} de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit para $V_{in} = 1$ V



(b) V_{DAC} de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit para $V_{in} = 1,8 V$

Figura 5.18: V_{DAC} de un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit para V_{in} continuo

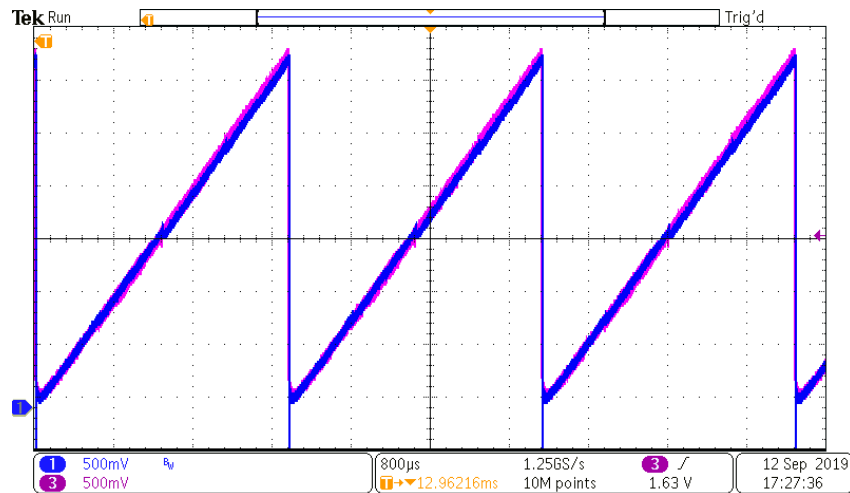


Figura 5.19: V_{in} y B obtenido por un ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit

5.5. Conclusiones

En este capítulo se presenta una optimización en la implementación del DAC, donde se reemplazó el modulador por ancho de pulsos (PWM) por un PWM de ciclo de trabajo distribuido (PWM-DDC) con el objetivo de mejorar las prestaciones del ADC. El PWM-DDC distribuye su tensión de salida (V_{DDC}) a lo largo del tiempo en que se realizan las 2^N cuentas, es decir el tiempo en que V_{DDC} esta en V_{OH} y V_{OL} no es continuo, sino distribuido en dicho tiempo acorde al ciclo de trabajo (d). Esta diferencia en la forma de implementación permite disminuir el rizado (ΔV_{LPF}) de la salida del filtro pasa bajo (LPF).

Cuando d es un número par, la frecuencia del PWM-DDC (f_{DDC}) varía a medida que d lo hace, debido a que V_{DDC} se puede distribuir de manera uniforme. Si la f_{DDC} aumenta, es posible escoger LPFs que permitan aumentar $f_{Sampling}$. Cuando d es un número impar, al no poder distribuirse de manera uniforme V_{DDC} , la f_{DDC} se mantiene constante, siendo igual a la frecuencia del PWM (f_{PWM}). Aunque la $f_{DDC} = f_{PWM}$, ΔV_{LPF} del LPF con el PWM-DDC es menor, por lo que es posible escoger un LPF que aumente $f_{Sampling}$.

Al reemplazar el PWM por el PWM-DDC en cada optimización, es posible escoger LPFs que permitan $f_{Sampling}$ mayores. Además, el PWM-DDC es compatible con la optimización del double data rate (DDR) y con la paralelización del PWM.

6. Capítulo 6

6.1. Resumen

En este capítulo se presenta la generalización de la implementación del convertor analógico digital (ADC) basado en successive approximation register (SAR), reemplazando el DAC con la combinación de PWM y filtro paso bajo (LPF). A partir de la frecuencia de muestreo del ADC deseada ($f_{Sampling}$) se escoge el tipo de implementación necesario para alcanzar dicho objetivo, priorizando el coste de los elementos analógicos de la implementación y la menor cantidad de salidas de la FPGA.

6.2. Generalización sistemática de la arquitectura del ADC basado en SAR

Una de las ventajas del ADC basado en SAR donde se reemplaza el DAC por la combinación del PWM con un LPF es la versatilidad que presenta. Basándose en las especificaciones de la aplicación, puede ser necesario un ADC con $f_{Sampling}$ alta con altos costes o por el contrario, un ADC con $f_{Sampling}$ bajas que permita menores costes, en ambos casos puede ser implementado con este ADC.

A continuación, se presentan los pasos a seguir para obtener los parámetros de implementación del ADC y el tipo de implementación del DAC necesario para obtener la $f_{Sampling}$ deseada. Todos los valores se presentan en base a los parámetros de la FPGA, tales como f_{clk} y ΔV_O , además de tener la posibilidad de implementar salidas DDR. Los resultados están normalizados para $f_{CLK}=100$ MHz, por lo que se incluye un factor R presentado en la Eq. 6.1, con el objetivo de obtener los parámetros de implementación en otras familias de FPGA.

$$R = \frac{f_{clk}}{100 \text{ MHz}} \quad (6.1)$$

1. El primer paso para implementar el ADC es definir la resolución y la $f_{Sampling}$ deseada. Con estos datos y las características de la FPGA se puede hallar LSB , con el objetivo de obtener ΔV_{DAC} máxima. Para hallar LSB se utilizan las Eq. 6.2 y 6.3. Con estos valores se genera la condición mostrada en la Eq. 6.4.

La ΔV_O viene dada por la FPGA, a partir de la Eq. 6.2, donde V_{OH} es la tensión de salida de la FPGA cuando el valor es '1' o 'high' y V_{OL} cuando la tensión de salida es '0' o 'low'.

$$\Delta V_O = V_{OH} - V_{OL} \quad (6.2)$$

$$LSB = \frac{\Delta V_O}{2^N} \quad (6.3)$$

$$\Delta V_{DAC} \leq LSB \quad (6.4)$$

2. A partir de la $f_{Sampling}$ se obtiene $t_{Settling}$ con la Eq. 6.5, donde $t_{Settling}$ es el tiempo de establecimiento del LPF, siendo el tiempo necesario para que el LPF llegue a estado estacionario y se pueda realizar la comparación entre la tensión de entrada (V_{in}) y V_{DAC} en cada iteración.

$$f_{Sampling} = \frac{1}{N \cdot t_{Settling}} \quad (6.5)$$

3. Se obtiene V_{DAC} a partir de la tensión de salida del LPF (V_{LPF}) y $\frac{LSB}{2}$ en la Eq. 6.6. Se resta la tensión equivalente a $\frac{LSB}{2}$, con el objetivo de que el error de conversión pase de $\pm 1 LSB$, a ser $\pm \frac{1}{2} LSB$.

$$V_{DAC} = V_{LPF} - \frac{LSB}{2} \quad (6.6)$$

4. Se presentaron dos alternativas para implementar el PWM del DAC. Una de ellas es el PWM de N-bit regido por un contador ascendente que va desde 0 hasta $2^N - 1$, aumentando en el flanco ascendente del reloj. La salida del PWM (V_{PWM}) está en V_{OH} desde Contador=0 hasta Contador=d, después $V_{PWM} = V_{OL}$ hasta que Contador= $2^N - 1$ y vuelve a empezar. Para obtener f_{PWM} se utiliza la Eq. 6.7.

$$f_{PWM} = \frac{f_{clk}}{2^N} \quad (6.7)$$

La segunda alternativa es el PWM-DDC de N-bit, también controlado por un contador ascendente que va desde 0 hasta $2^N - 1$, aumentando en el flanco ascendente del reloj. La salida del PWM-DDC (V_{DDC}) se distribuye a lo largo del tiempo en que se realizan las 2^N cuentas, donde el valor de d indica cuentas veces V_{DDC} esta en V_{OH} a lo largo del tiempo mencionado. El PWM-DDC se implementa con el objetivo de aumentar f_{DDC} con respecto a f_{PWM} y así disminuir ΔV_{LPF} . Para obtener la $f_{DPC-Maxima}$ se utiliza la Eq. 6.8

$$f_{DDC-Maxima} = \frac{f_{clk}}{2^N} \quad (6.8)$$

5. Si la FPGA permite la implementación de salidas DDR y flip-flop ascendente y descendente, es posible implementar el PWM-DDR o PWM-DDC-DDR, generando mejores prestaciones del ADC. Al implementar las salidas DDR, es posible implementar el contador para que aumente tanto en el flanco ascendente como descendente del reloj, incrementando al doble f_{PWM} o f_{DDC} . En la Fig. 6.1 se aprecia un PWM-DDR y PWM-DDC-DDR implementado con $D = \frac{1}{2}$.

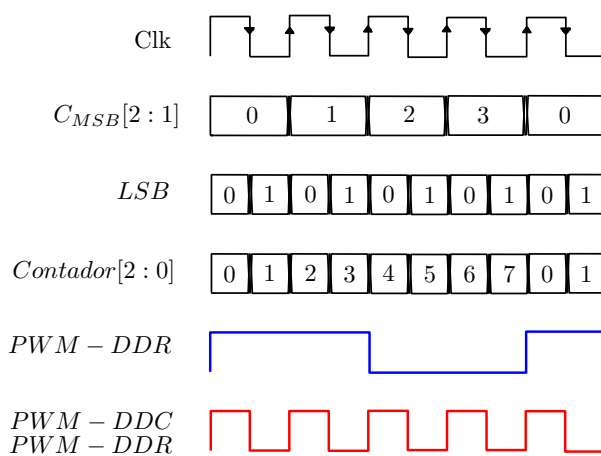


Figura 6.1: Comportamiento del contador implementado con registro DDR

6. Una vez se decida el tipo de implementación del PWM, se debe encontrar los parámetros del LPF. Se pueden hallar diferentes f_C a partir de diferentes O_{LPF} con la Eq. 6.9a para el PWM y Eq. 6.9b para el PWM-DDC.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{O_{LPF}\sqrt{\pi}}{2^{\left(\frac{2+N+N\cdot O_{LPF}}{O_{LPF}}\right)}} \quad (6.9a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{1}{2^{\left(\frac{2+N\cdot O_{LPF}}{O_{LPF}}\right)}} \quad (6.9b)$$

Si la FPGA que se utiliza permite las salidas DDR y flip-flop ascendente y descendente, se encuentra f_C a partir de O_{LPF} con la Eq. 6.10a para el PWM-DDR y la Eq. 6.10b.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2 \cdot O_{LPF}\sqrt{\pi}}{2^{\left(\frac{2+N+N\cdot O_{LPF}}{O_{LPF}}\right)}} \quad (6.10a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2}{2^{\left(\frac{2+N\cdot O_{LPF}}{O_{LPF}}\right)}} \quad (6.10b)$$

7. Una vez se obtiene f_C para diferentes O_{LPF} , es necesario conocer el $t_{Settling}$ de cada LPF. Para esto se utilizan las gráficas normalizadas para diferentes O_{LPF} con $f_C=1$ Hz de la respuesta al escalón mostradas en la Fig. 6.2. Para obtener $t_{Settling}$ se divide el eje X de gráfica por f_C del O_{LPF} correspondiente.

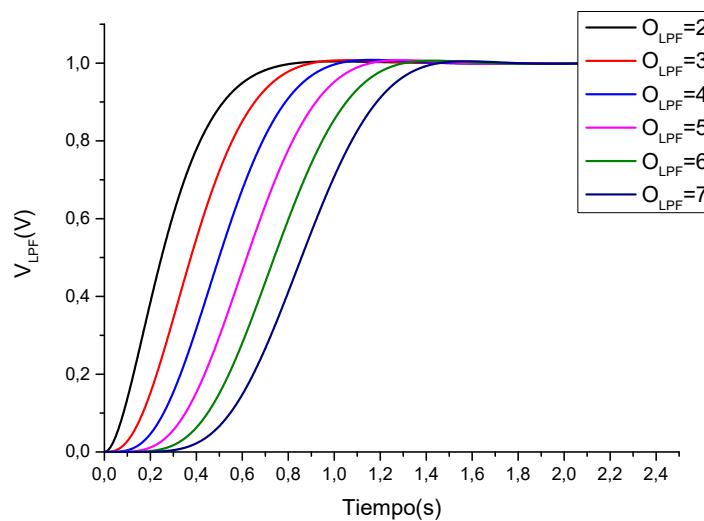


Figura 6.2: V_{LPF} para LPFs de arquitectura Bessel con $f_C = 1$ Hz para diferentes O_{LPF} .

En la Tabla 6.1 se muestra el $t_{Settling}$ para cada O_{LPF} , con el que se realizaran los cálculos de los parámetros del LPF y el ADC. De este modo para obtener el $t_{Settling}$ de los f_C hallados se utiliza la Eq. 6.11.

$$t_{Settling} = \frac{t_{Settling-Normalizado}}{f_C} \quad (6.11)$$

O_{LPF}	2	3	4	5	6	7	8	9	10
$t_{Settling}$ normalizado	0.65	1.00	1.25	1.52	1.75	1.90	2.10	2.21	2.32

Tabla 6.1: $t_{Settling}$ normalizado del LPF con arquitectura Bessel con diferentes O_{LPF} .

8. Si los $t_{Settling}$ obtenidos de los diferentes O_{LPF} son menores que el obtenido en la Eq. 6.5, se debe escoger el LPF con menor O_{LPF} , pues los costes son menores.

Por el contrario, si no se pudo alcanzar el $t_{Settling}$ deseado, se debe modificar la forma de implementación del LPF. En el capítulo 4 se presentó una optimización que permitía paralelizar el PWM de N-bit en K PWMs de $\frac{N}{K}$ -bit. Esta optimización también es aplicable para el PWM-DDC o sus variantes DDR.

Para obtener los parámetros de los LPF se utiliza la Eq. 6.13a para el PWM y la Eq. 6.13b para el PWM-DDC.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{O_{LPF}\sqrt{\pi}}{2^{\left(\frac{N}{K} + O_{LPF}\right)}} \quad (6.12a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2^{\left(\frac{N}{O_{LPF}\cdot K}\right)}}{2^{\left(\frac{N+2}{O_{LPF}} + \frac{N}{K}\right)}} \quad (6.12b)$$

Si la FPGA a utilizar permite las salidas en DDR y los flip-flop de doble flanco de reloj, se utilizaría la Eq. 6.13a para el PWM y la Eq. 6.13b para el PWM-DDC.

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2 \cdot O_{LPF}\sqrt{\pi}}{2^{\left(\frac{N}{K} + O_{LPF}\right)}} \quad (6.13a)$$

$$\left(\frac{f_C}{f_{clk}}\right) < \frac{2^{\left(1 + \frac{N}{O_{LPF}\cdot K}\right)}}{2^{\left(\frac{N+2}{O_{LPF}} + \frac{N}{K}\right)}} \quad (6.13b)$$

Al aumentar K , se obtiene mejores prestaciones puesto que aumenta la f_{PWM} o f_{DDC} , pero en contra parte al aumentar K , aumenta el número de elementos analógicos necesarios para la implementación, además del número de salidas de la FPGA.

Si con $K=2$ no se obtiene la $f_{Sampling}$ deseada, hay la posibilidad de aumentar a $K=3$ y comprobar si es posible implementar el ADC. K puede ser aumentada hasta un máximo de $K=N-1$. Si este último valor de K no es suficiente se puede implementar $K=N$ que sería equivalente a un DAC. Este DAC se puede implementar con la red R-2R, sin necesidad de LPFs, pero utilizando $K+1$ salidas de la FPGA.

9. Una vez se hallen los parámetros de los LPFs, se obtiene V_{DAC} a partir de V_{LPF} y $\frac{LSB}{2}$. Si se utiliza el PWM o PWM-DDC de N -bit o sus variantes DDR, se utiliza la Eq. 6.5. Si se paraleliza el PWM en K PWMs de $\frac{N}{K}$ -bit se utiliza la Eq. 6.14. Esta misma ecuación se utiliza para el PWM-DDC o las variantes con DDR.

$$V_{DAC} = V_{(K-1)} + \sum_{j=0}^{K-2} \left(\frac{2^{\frac{N}{K} \cdot j}}{2^{N-\frac{N}{K}}} \cdot V_j \right) - \frac{LSB}{2} \quad (6.14)$$

Cabe la posibilidad que al dividir el PWM en K PWMs de $\frac{N}{K}$ -bit, esta división no sea exacta. En este caso V_{DAC} se obtiene con la Eq. 6.15, donde $N = P_2 + P_1 \cdot (K - 1)$, donde hay un PWM de P_2 -bit y $K - 1$ PWMs de P_1 -bit.

$$V_{DAC} = V_{(K-1)} + \sum_{j=0}^{K-2} \left(\frac{2^{P_1 \cdot (j+1)}}{2^N} \cdot V_j \right) - \frac{LSB}{2} \quad (6.15)$$

10. Una vez sean hallados los parámetros del LPF y V_{DAC} , el siguiente paso es la implementación del LPF. El LPF será implementado con la topología Sallen-Key mostrada en la Fig. 6.3, donde se implementa un LPF con $O_{LPF}=2$. Si se requiere un LPF con mayor O_{LPF} , se conectan en cascada varios LPF de $O_{LPF}=2$. Cada uno de los K LPFs será conectado a un PWM o PWM-DDC de $\frac{N}{k}$ -bit.

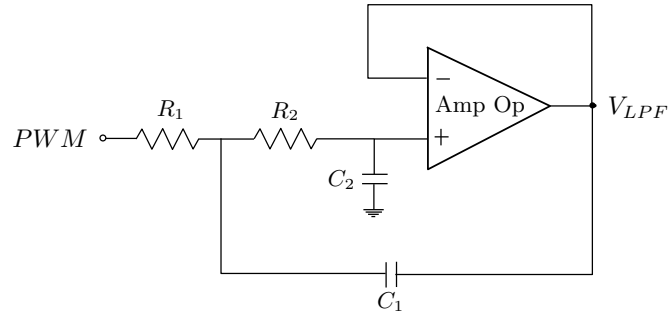


Figura 6.3: Arquitectura Sallen-Key para LPF de $O_{LPF}=2$

Para obtener los valores de R y C se hace a través de la Tabla 6.2, junto a las Eq. 6.16a, 6.16b y 6.16c. Dependiendo del O_{LPF} utilizado, se utilizan las constantes de la Tabla 6.2 para hallar R y C, teniendo en cuenta que $R_1=R_2$ y $C_1=C_2$.

O_{LPF}	Q_1	f_{p1}	Q_2	f_{p2}	Q_3	f_{p3}	Q_4	f_{p4}
2	0.577	1.274						
4	0.52	1.432	0.81	1.606				
6	0.51	1.607	1.02	1.908	0.61	1.692		
8	0.51	1.781	1.23	2.192	0.71	1.956	0.56	1.835

Tabla 6.2: Valores normalizados a $f_c=1$ Hz del LPF Bessel .

$$f_p = f_c \cdot f_{p1} \quad (6.16a)$$

$$f_p = \frac{1}{2\pi R \sqrt{C_1 C_2}} \quad (6.16b)$$

$$Q = 0,5 \cdot \sqrt{\frac{C_2}{C_1}} \quad (6.16c)$$

11. Para implementar V_{DAC} a partir de los V_{LPF} , se realiza a través de un sumador analógico con un amplificador operacional, como se muestra en la Fig. 6.4.

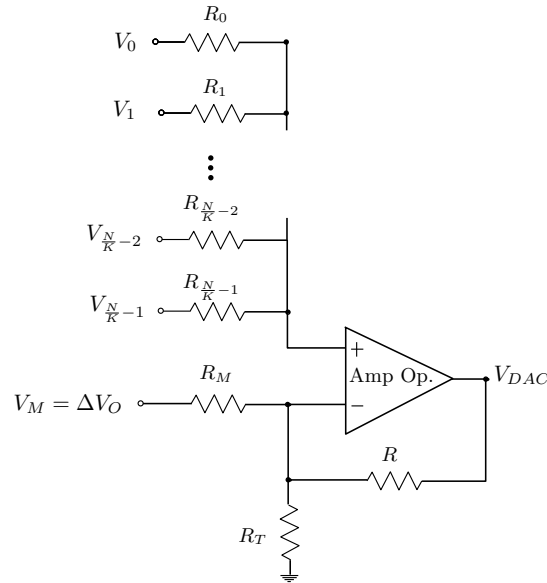


Figura 6.4: Circuito sumador analógico con AO para un ADC de N-bit con el PWM dividido en K PWMs de $\frac{N}{K}$ -bit.

Para obtener las resistencias de cada LPF se utiliza la Eq. 6.17 cuando $\frac{N}{K}$ es entero, o las Eq. 6.18a y 6.18b cuando $\frac{N}{K}$ no es entero. Se parte de la resistencia de retroalimentación R . Para obtener la resistencia correspondiente a $\frac{LSB}{2}$ (R_M) se utiliza las Eq. 6.19.

$$R_{(K-1-j)} = \sum_{j=0}^{K-1} \left(\frac{R}{2^{(\frac{N}{K} \cdot j)}} \right) \quad (6.17)$$

$$R_{(K-1)} = \left(\frac{R}{2^{P_2}} \right) \quad (6.18a)$$

$$R_{(K-2-j)} = \sum_{j=1}^{K-1} \left(\frac{R}{2^{(P_2+P_1 \cdot j)}} \right) \quad (6.18b)$$

$$R_M = \frac{R}{2^{N+1}} \quad (6.19)$$

Para obtener R_T se utiliza las Eq. 6.20d, donde A_{Suma} es la ganancia de cada uno de los V_{LPF} y se obtiene con la Eq. 6.20a o la Eq. 6.20b, mientras que A_{Resta} es la ganancia del equivalente a $\frac{LSB}{2}$.

$$A_{Suma} = 1 + \sum_{j=1}^{K-1} \left(\frac{1}{2^{\frac{N}{K} \cdot j}} \right) \quad (6.20a)$$

$$A_{Suma} = 1 + \frac{1}{2^{P_2}} + \sum_{j=1}^{K-2} \left(\frac{1}{2^{P_2+P_1 \cdot j}} \right) \quad (6.20b)$$

$$A_{Resta} = \frac{1}{2^{N+1}} \quad (6.20c)$$

$$R_T = \frac{1}{A_{Resta} + 1 - A_{Suma}} \quad (6.20d)$$

12. Para escoger los elementos analógicos, tales como el AO, las resistencias, condensadores y comparador se deben tener en cuenta los problemas prácticos de la implementación que son explicados en el capítulo 2 en la sección 2.2.5.

6.3. Diseño de un ADC de 8-bit con $f_{Sampling}=800$ kSPS

Para ejemplificar la metodología para el diseño e implementación de un ADC basado en SAR reemplazando el DAC con LPF y PWM, se realiza el diseño y simulación del ADC de 8-bit con $f_{Sampling}=800$ kSPS. Se utiliza la FPGA de la familia Zynq 7000, con $\Delta V_O=3.3$ V y $f_{clk}=100$ MHz, además serán utilizadas las salidas DDR.

En primer lugar, se obtienen $LSB=12.89$ mV y $t_{Settling}=156$ ns como se muestra a continuación.

$$LSB = \frac{\Delta V_O}{2^N} = 12,89 \text{ mV}$$

$$t_{Settling} = \frac{1}{N \cdot f_{Sampling}} = 156 \text{ ns}$$

Una vez obtenidos LSB y $t_{Settling}$ se decide la forma de implementación del PWM. Se utilizará el PWM-DDC-DDR, puesto que es el que mejores prestaciones genera. El siguiente paso sería encontrar los parámetros del LPF a partir de la Eq. 6.10b para diferentes O_{LPF} . Como se puede apreciar en la Fig. 6.5, con el PWM-DDC-DDR de 8-bit no se alcanza la

$f_{Sampling}$ deseada, por lo que es necesario paralelizar el PWM-DDC-DDR.

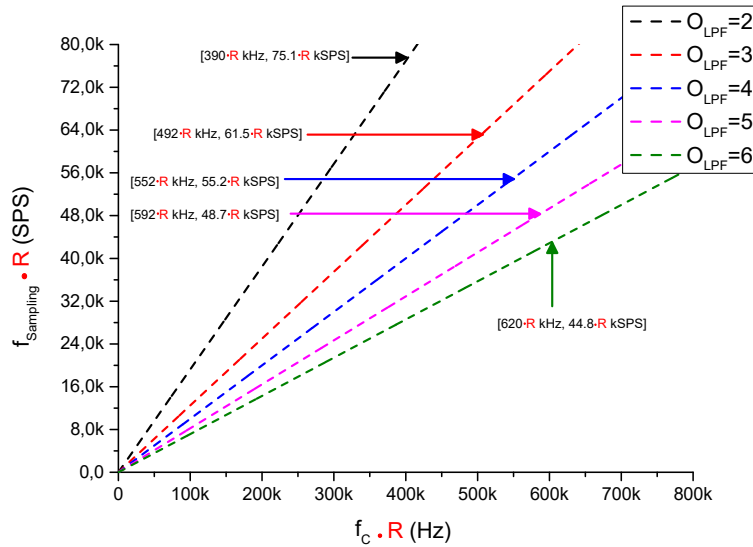


Figura 6.5: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con PWM-DDC-DDR de 8-bit normalizado a $f_{CLK}=100$ MHz.

Se paraleliza el PWM-DDC-DDR de 8-bit en 2 PWM-DDC-DDR de 4-bit y se obtienen los parámetros del LPF. Como se ve en la Fig. 6.6 la $f_{Sampling}$ no es alcanzada con esta implementación, por lo que se debe aumentar la K.

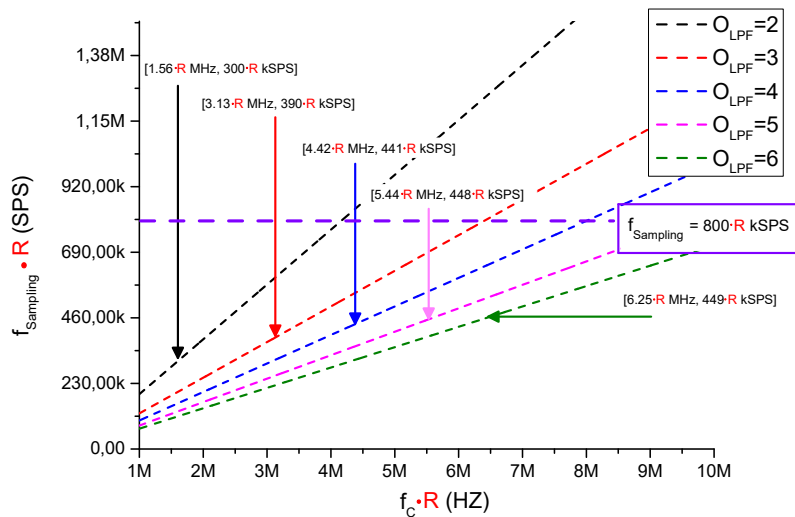


Figura 6.6: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 2 PWM-DDC-DDR de 4-bit normalizado a $f_{CLK}=100$ MHz.

Se obtiene la Fig. 6.7 donde se paraleliza el PWM-DDC-DDR en 3 PWM-DDC-DDR, 2 PWM-DDC-DDR de 3-bit y uno de 2-bit. Como se puede apreciar con $O_{LPF}=4$ es posible alcanzar la $f_{Sampling}=800$ kSPS, esto quiere decir que se necesitan 3 salidas de la FPGA

para los PWM-DDC-DDR y una entrada para el comparador. En la Tabla 6.3 se presenta los elementos analógicos necesarios para la implementación.

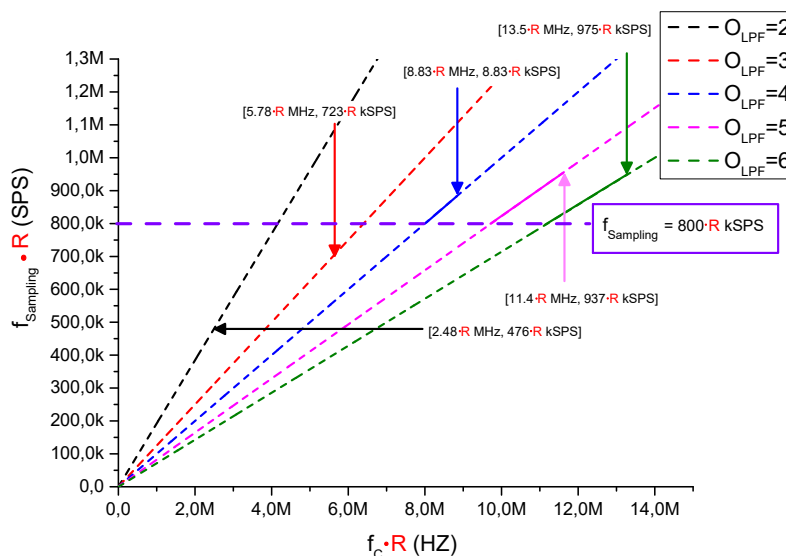


Figura 6.7: f_c vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 3 PWM-DDC-DDR, 2 PWM-DDC-DDR de 3-bit y uno de 2-bit normalizado a $f_{CLK}=100$ MHz.

ADC	Número entradas	O_{LPF}	R	C	Número AO	$t_{Settling}$ del LPF	$f_{Sampling}$
8-bit	4	4	18	12	7	156 ns	800 kSPS

Tabla 6.3: Características del ADC de 8-bit con 2 PWM-DDC-DDR de 3-bit y uno de 2-bit.

Se obtiene la Fig. 6.8 donde se paraleliza el PWM-DDC-DDR en 4 PWM-DDC-DDR de 2-bit. Como se puede apreciar es posible alcanzar $f_{Sampling}=800$ kSPS con $O_{LPF}=3$. En la Tabla 6.4 se muestra los elementos analógicos necesarios para la implementación.

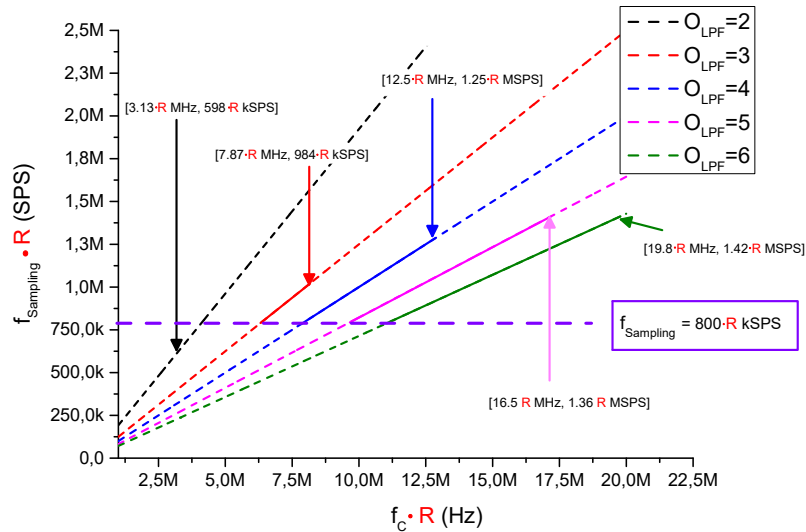


Figura 6.8: f_C vs. $f_{Sampling}$ para LPFs con diferentes O_{LPF} implementando un ADC 8-bit con 4 PWM-DDC-DDR de 2-bit normalizado a $f_{CLK}=100$ MHz.

ADC	Número de entradas	O_{LPF}	R	C	Número AO	$t_{Settling}$ del LPF	$f_{Sampling}$
8-bit	5	4	19	12	5	156 ns	800 kSPS

Tabla 6.4: Características del ADC de 8-bit con 4 PWM-DDC-DDR de 2-bit.

Las Fig. 6.7 y 6.8 representan la versatilidad del ADC, puesto que se puede acoplar a las necesidades del diseño. Si se dispone de entradas disponibles en la FPGA, es preferible utilizar $K=4$ puesto que utiliza menos componentes analógicos. Por el contrario, si las entradas de la FPGA son limitadas, conviene utilizar $K=3$. Para efectos de explicación se escogerá $K=3$ para el diseño del ADC con $f_{Sampling}=800$ kSPS.

El siguiente paso es hallar V_{DAC} a partir de los V_{LPF} y $\frac{LSB}{2}$ siguiendo la Eq. 6.15 y se muestra en la Eq. 6.21. Una vez obtenida V_{DAC} se realiza el diseño del sumador analógico y se muestra en la Fig. 6.9

$$V_{DAC} = V_2 + \frac{V_1}{2^2} + \frac{V_0}{2^5} - \frac{LSB}{2} \quad (6.21)$$

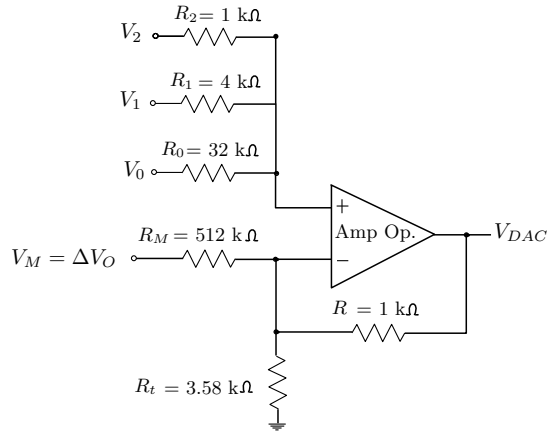


Figura 6.9: Circuito sumador-restador con AO para un ADC de 8-bit con $K=3$.

En la Fig. 6.10 se presenta el esquemático de la implementación y en la Fig. 6.11 se muestra el comportamiento de V_{DAC} en el proceso de conversión.

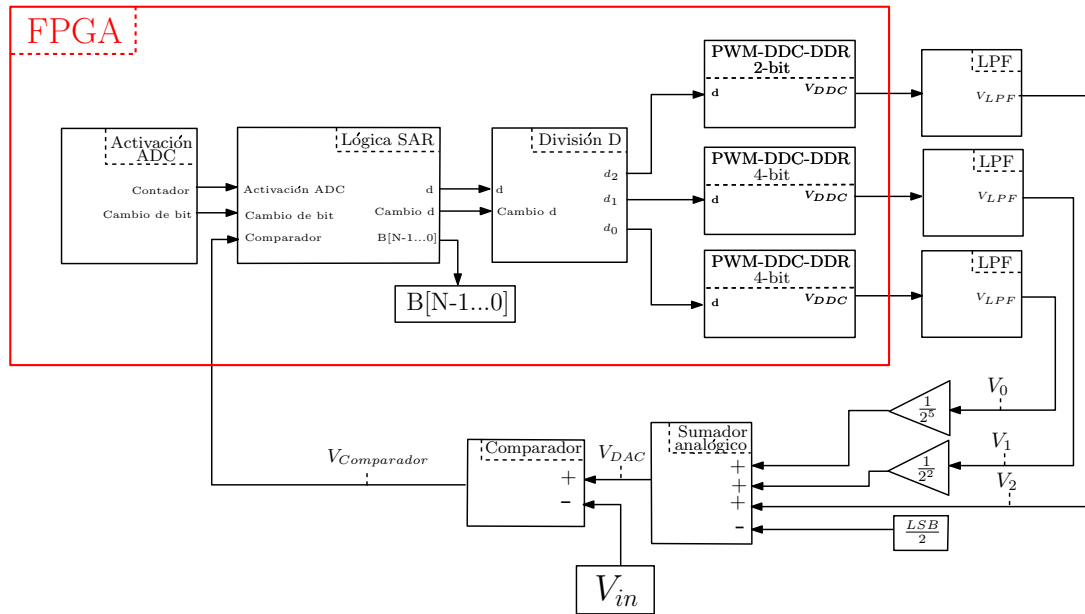


Figura 6.10: Esquemático de la implementación del ADC con 2 PWM-DDR de 3-bit y uno de 2-bit.

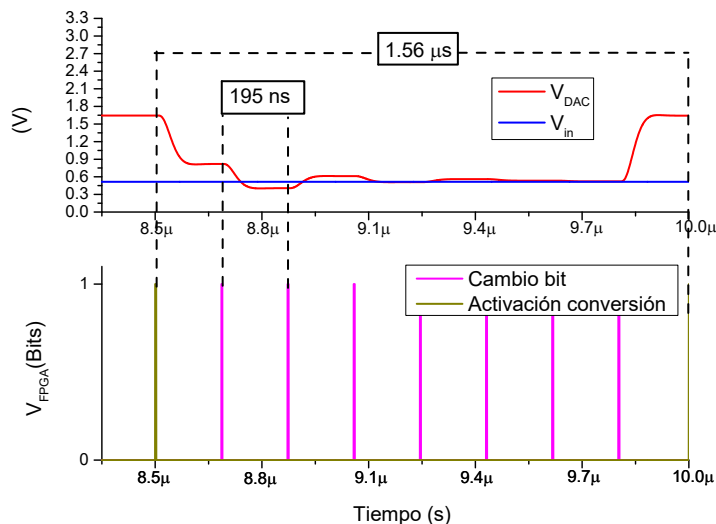


Figura 6.11: Comportamiento de V_{DAC} en el proceso de conversión del ADC de 8-bit con $f_{Sampling}=800$ kSPS.

6.4. Conclusiones

En este capítulo se presenta paso a paso la metodología para el diseño e implementación de un ADC de N-bit basado en SAR a partir de la frecuencia de muestreo del ADC ($f_{Sampling}$), la frecuencia del reloj de la FPGA (f_{CLK}) y la resolución del ADC. Se reemplaza el módulo del convertor digital analógico (DAC) por un módulo de modulador de ancho de pulsos (PWM) y filtro paso bajo (LPF). En esta metodología se presenta los criterios necesarios para escoger el tipo de implementación tanto del PWM, como del módulo del DAC.

Se presentan diferentes modos de implementación para un mismo ADC de N-bit. donde se divide el PWM en K PWMs de $\frac{N}{K}$ -bit. Con el objetivo de brindar versatilidad a la hora de diseñar el ADC, es posible obtener la $f_{Sampling}$ deseada a partir de diferentes K, priorizando costes de implementación, área de implementación o número de salidas de FPGA, según el diseño lo requiera.

7. Conclusiones

En esta tesis doctoral se presenta dos arquitecturas de implementación de un convertor analógico digital (ADC) escalable y parametrizable a partir de las entradas y salidas de la FPGA. La primera arquitectura presentada es la implementación de un ADC a partir de un circuito RC y puertas lógicas, donde se genera un pulso correspondiente a la tensión de entrada (V_{in}) del ADC. La segunda alternativa presentada es la implementación de un ADC basado en successive approximation register (SAR), donde se reemplaza el convertor digital analógico (DAC) por un modulador de ancho de pulso (PWM) y un filtro paso bajo (LPF). Las arquitecturas presentadas fueron estudiadas con una FPGA de la familia Zynq 7000 con características específicas, pero puede ser extrapolado a a diferentes familias de FPGA, con diferentes f_{clk} , ΔV_O y V_{th} .

En el capítulo 1 se analiza la implementación de un convertor analógico digital a partir de la estrategia del one shot ADC en una FPGA, dónde se analizan los límites existentes del diseño, las ventajas y desventajas de este. El convertor analógico digital a partir del circuito one shot ADC es analizado debido a su facilidad de implementación y su bajo coste. El circuito one shot genera un pulso en la puerta lógica NOT (V_{NOT}), cuya anchura (T_{in}) es modulada por V_{in} . A partir del T_{in} generado se obtiene el valor digital correspondiente a la tensión de entrada V_{in} . Para obtener T_{in} se implementa un contador ascendente con $f_{counter} = f_{clk}$ o $f_{counter} = 2 \cdot f_{clk}$ dependiendo si se cuenta o no la característica de flip-flop de doble flanco de reloj. T_{in} y V_{in} presentan una relación exponencial, por lo que es necesario modificar los parámetros y así poder trabajar en la zona con un comportamiento más lineal de esta relación. Se modifica el circuito one shot ADC, incrementando la constante de tiempo del condensador y añadiendo un $V_{Sumador}$. Como ventaja se obtiene un sistema con un comportamiento más lineal, pero como desventaja el tiempo de descarga (t_D) aumenta, disminuyendo $f_{sampling}$.

En el capítulo 2 se presentó el análisis de la implementación de un ADC basado en la lógica SAR, donde se reemplaza el convertor digital analógico (DAC) por la combinación de un modulador de anchura de pulsos (PWM) y un filtro pasa bajo (LPF). Se estudió el efecto que tiene la frecuencia de corte (f_C) y el orden (O_{LPF}) en la respuesta al escalón del LPF, donde a mayor f_C , el tiempo de establecimiento del LPF ($t_{Settling}$) es menor,

pero el rizado del LPF (ΔV_{LPF}) es mayor. En el caso del O_{LPF} , a mayor O_{LPF} e igual f_C , ΔV_{LPF} disminuye y aumenta $t_{Settling}$, además de aumentar el coste de implementación.

Se presenta en método sistemático para la implementación de un ADC de N-bit en una FPGA a partir de $f_{Sampling}$ deseada. Teniendo en cuenta las características de la FPGA (f_{clk} y ΔV_O) se obtienen los parámetros del LPF necesarios para la implementación del ADC. El método presentado puede ser extrapolado a diferentes familias de FPGA con diferentes características.

En el capítulo 3 se estudió la implementación del PWM con el sistema de adquisición de datos basado en 'double data rate' (DDR). Para esto se utiliza un registro dedicado 'Output DDR' (ODDR) de la FPGA para implementar salidas DDR. Esto permite disminuir el tiempo de propagación de la señal hasta la salida de la FPGA y así poder aumentar la velocidad de procesado del sistema. Al implementar el PWM con el DDR es posible utilizar los flip-flop de flanco ascendente y descendente de reloj, por lo que la frecuencia del PWM (f_{PWM}) es incrementada, permitiendo encontrar parámetros del LPF que generan mayor $f_{Sampling}$.

En el capítulo 4 se presentó una optimización en la implementación del DAC del ADC basado en SAR. Esta optimización consiste en paralelizar el PWM N-bit dividiéndolo en K PWMs de $\frac{N}{K}$ -bit. Cada PWM resultante con su respectivo LPF generan K V_{LPF} , que junto a un sumador analógico generan V_{DAC} . Esta optimización permite el aumento de la f_{PWM} , lo que genera mejores prestaciones para el ADC. La versatilidad del diseño se ve mejorado gracias a la división del PWM de N-bit en K PWMs de $\frac{N}{K}$ -bit. Si el usuario requiere una mayor $f_{Sampling}$, aumenta K, permitiendo que el PWM sea dividido en un mayor número de PWMs. Como desventaja está que al aumentar K es necesario un mayor número de salidas en la FPGA. Esta optimización puede ser combinada con la optimización de las salidas DDR.

En el capítulo 5 se presentó una modificación en la forma de implementación del PWM con el objetivo de disminuir ΔV_{DAC} , donde se utiliza el Distributed Duty cycle PWM (PWM-DDC). El PWM-DDC distribuye su tensión de salida (V_{DDC}) a lo largo del tiempo en que se realizan las 2^N cuentas, es decir el tiempo en que V_{DDC} esta en V_{OH} y V_{OL} no es continuo, sino distribuido en dicho tiempo acorde al ciclo de trabajo (d). Esta diferencia en la forma de implementación permite disminuir el ΔV_{LPF} de la salida LPF. El

PWM-DDC genera mejores prestaciones comparado con el PWM debido a la disminución de ΔV_{LPF} . La implementación del PWM-DDC puede ser combinada de con la salida DDR y la paralelización del PWM-DDC en K PWM-DDCs de $\frac{N}{K}$ -bit.

En el capítulo 6 se presenta una metodología para implementar un ADC basado en SAR de N-bit reemplazando el DAC por la combinación de PWM o PWM-DDC y un LPF. A partir de la $f_{Sampling}$ deseada se escoge la implementación más conveniente en la FPGA. Se prioriza los costes de implementación, por lo que se escoge la implementación que alcance $f_{Sampling}$ con el menor número de elementos analógicos y salidas de la FPGA.

Referencias

- [1] J. Zhang, Y. Kuai, S. Zhou, G. Hou, and M. Ren. Improved minimum entropy control for two-input and two-output networked control systems. In *2016 UKACC 11th International Conference on Control (CONTROL)*, pages 1–5, 2016.
- [2] M. Ren, J. Zhang, M. Jiang, M. Yu, and J. Xu. Minimum (h, ϕ) – entropy control for non-gaussian stochastic networked control systems and its application to a networked dc motor control system. *IEEE Transactions on Control Systems Technology*, 23(1):406–411, 2015.
- [3] E. E. Cruz-Miguel, J. R. García-Martínez, J. Rodríguez-Reséndiz, and R. V. A Carrillo-Serrano. New methodology for a retrofitted self-tuned controller with open-source fpga. *Sensors*, 2020.
- [4] T. Sakanushi, K. Yamada, S. Matsuura, T. Hagiwara, Y. Ando, and I. Murakami. A design method for two-degree-of-freedom simple repetitive control systems for multiple-input/multiple-output plants. In *2010 World Automation Congress*, pages 1–6, 2010.
- [5] G. J. N. Ang, W. Z. Lim, and C. M. Lim. A proposed two-input two-output self-tuning control scheme. In *2018 IEEE Symposium on Computer Applications Industrial Electronics (ISCAIE)*, pages 102–107, 2018.
- [6] R. R. Sudharsan. Synthesis of fir filter using adc-dac: A fpga implementation. In *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*, pages 18–20, India, pp. 1–3, December 2019. Krishnankoil.
- [7] Y.-T. Yau, K.-I. Hwu, and J.-J. Shieh. Applying fpga control with adc-free sampling to multi-output forward converter. *Electronics*, 2021.
- [8] J. Rodríguez-Reséndiz, F. Mendoza-Mondragón, R. A. Gómez-Loenzo, M. A. Martínez-Hernández, and V. H. Mucino. An approach to motion control applications based on advanced programmable devices. *Int. J. Electr. Eng. Educ*, 2012:243–259.
- [9] E. E. Cruz-Miguel, J. Rodríguez-Reséndiz, J. R. García-Martínez, K. A. Camarillo-Gómez, and G. I. Pérez-Soto. Field programmable gate array-based laboratory oriented to control theory courses. *Comput. Appl. Eng. Educ*, 2019:1253–1266.
- [10] N. Sahu, N. D. Londhe, and G. B. Fpga Kshirsagar. Fpga applications in inverter and converter circuits: A review on technology, benefits and challenges. pages 17–18. 2017 International Conference on Innovations in Information, March 2017.
- [11] O. S. Dhede and S. K. A review Shah. Hardware implementation of aes using minimal resources on fpga. In *Proceedings of the 2015 International Conference on Pervasive Computing (ICPC)*, pages pp. 1–3, Pune, India, January 2015.
- [12] M. Ni, X. Wang, F. Li, W. Rhee, and Z. A Wang. 13-bit 2-gs/s time-interleaved adc with improved correlation-based timing skew calibration strategy. *IEEE Trans. Circuits Syst. I Regul. Pap*, 69(2):481–494, February 2021.

- [13] J. Nam, M. Hassanpourghadi, A. Zhang, and M. S. A. 12-Bit 1.6 Chen. 3.2, and 6.4 gs/s 4-b/cycle time-interleaved sar adc with dual reference shifting and interpolation. *IEEE J. -Solid-State Circuits*, 2018:1765–1779.
- [14] S. Devarajan, L. Singer, D. Kelly, T. Pan, J. Silva, J. Brunsilius, D. Rey-Losada, F. Murden, C. Speir, J. Bray, et al. A 12-b 10-gs/s interleaved pipeline adc in 28-nm cmos technology. *J. -Solid-State Circuits*, 2017:3204–3218.
- [15] Y. Zhou, B. Xu, and Y. A Chiu. 12-b 1-gs/s 31.5-mw time-interleaved sar adc with analog hpf-assisted skew calibration and randomly sampling reference adc. *IEEE J. -Solid-State Circ*, 2019:2207–2218.
- [16] V. Monteiro, J. C. Ferreira, A. A. N. Melendez, J. A. Afonso, C. Couto, and J. L. Afonso. Experimental validation of a bidirectional three-level dc-dc converter for on-board or off-board ev battery chargers. *45th Annual Conference of the IEEE Industrial Electronics Society, Lisbon, Portugal*, 2019:14–17, October 2019.
- [17] B. Kim, H. Kim, and S. Choi. Three-phase on-board charger with three modules of single-stage interleaved soft-switching ac-dc converter. *IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 3405–3410, March 2018.
- [18] A. S. Kamble and P. S. Swami. On-board integreted charger for electric vehicle based on split three phase insuction motor. In *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*, pages 1–13, India, July 2018. Ernakulam.
- [19] J. Mooney, M. Halton, and A. E. Mahdi. Specialized digital signal processor for control of multi-rail/multi-phase high switching frequency power converters. *2010 Twenty-Fifth Annual IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2207–2211, February 2010.
- [20] J. Mooney, S. Effler, M. Halton, and A. E. Mahdi. Dsp-based control of multi-rail dc-dc converter systems with non-integer switching frequency ratios. *2010 1st International Conference on Energy, Power and Control (EPC-IQ)*, pages 203–207, November 2010.
- [21] A. A. Rahimi, L. B. Carreira, and S. Gupta. Synchronous multi-signal acquisition for wbsns using gold-code based joint-compressive sensing. *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 236–239, October 2016.
- [22] X. Man, L. Wu, X. Zhang, T. Ma, and W. A Jia. High precision multi-cell battery voltage detecting circuit for battery management systems. *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2016.
- [23] K. S. Amitkumar and G. Narayanan. Simplified implementation of space vector pwm strategies for a three level inverter. *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, pages 1–6, August 2012.
- [24] N. N. Lopatkin. Voltage source multilevel inverter voltage quality comparison under multicarrier sinusoidal pwm and space vector pwm of two delta voltages. *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 439–444, September 2017.
- [25] V. S. K. Devi and S. G. Srivani. Modified phase shifted pwm for cascaded h bridge

- multilevel inverter. *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pages 89–94, February 2017.
- [26] M.-K. Jeon, W.-J. Yoo, C.-G. Kim, and C. A Yoo. Stochastic flash analog-to-digital converter linearized by reference swapping. *IEEE Access*, 2017:23046–23051.
- [27] A. Fahmy, J. Liu, T. Kim, and N. Maghari. An all-digital scalable and reconfigurable wide-input range stochastic adc using only standard cells. *IEEE Trans. Circuits Syst. II Express Briefs*, 2015:731–735.
- [28] S. Weaver, B. Hershberg, and U. Moon. Digitally synthesized stochastic flash adc using only standard digital cells. *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61:84–91, 2014.
- [29] V. Unnikrishnan, S. R. Pathapati, and M. A Vesterbacka. fully synthesized all-digital vco-based analog-to-digital converter. *2015 Nordic Circuits and Systems Conference (NORCAS) NORCHIP & International Symposium on System-on-Chip (SoC)*, pages 1–4, October 2015.
- [30] V. Unnikrishnan and M. A Nand Vesterbacka. gate based standard cell vco for use in synthesizable adcs. *2015 Nordic Circuits and Systems Conference (NORCAS) NORCHIP & International Symposium on System-on-Chip (SoC)*, pages 1–4, October 2015.
- [31] S. Ghozzy, H. F. Ragai, and M. A El-Nozahi. Two-step vco-based adc with pwm pre-coded coarse quantizer. *2020 IEEE 3rd International Conference on Electronics Technology (ICET)*, pages 262–265, May 2020.
- [32] P. A. H. Vardhini. Analysis of integrator for continuous time digital sigma delta adc on xilinx fpga. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 2689–2693, March 2016.
- [33] G. P. Rajesh, C. K. Deepthy, M. M. Jermila, K. P. Raghunath, and V. N. Aparna. Force rebalance with multi-bit sigma-delta implemented with fpga for precision rate sensor. *2019 IEEE Region 10 Conference (TENCON)*, pages 654–657, October 2019.
- [34] P. S. Kumar, D. L. Narayana, V. R. Tejesvi, and M. K. Fpga Chandrasen. based sigma-delta analog to digital converter for power sensing. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 4912–4914, March 2016.
- [35] H. Homulle, F. Regazzoni, and E. Charbon. 200 ms/s adc implemented in a fpga employing tdc. *2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 228–235, November 2015.
- [36] H. Homulle, S. Visser, and E. A Cryogenic 1 GSa/s Charbon. Soft-core fpga adc for quantum computing applications. *IEEE Trans. Circ. Syst. I Regul. Pap*, 2016:1854–1865.
- [37] Z. Xiang, T. Wang, T. Geng, T. Xiang, X. Jin, and M. Soft-Core Herbordt. Multiple-lane, fpga-based adcs for a liquid helium environment. *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6, September 2018.

- [38] L. Leuenberger, D. Amiet, T. Wei, and P. Zbinden. An fpga-based 7-enob 600 msample/s adc without any external components. *2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 240–250, October 2021.
- [39] P. S. Croveti. All-digital high resolution d/a conversion by dyadic digital pulse modulation. *IEEE Trans. Circ. Syst. I Regul. Pap.*, 2017:573–584.
- [40] O. Aiello, P. Croveti, and M. Alioto. Fully synthesizable low-area analogue-to-digital converters with minimal design effort based on the dyadic digital pulse modulation. *IEEE Access*, 2020:70890–70899.
- [41] Aiello; Croveti, P. S.; Alioto, and M. Fully. Synthesizable low-area digital-to-analog converter with graceful degradation and dynamic power-resolution scaling. *IEEE Trans. Circ. Syst. I Regul. Pap.*, 2019:2865–2875.
- [42] Aiello; Croveti, P.; Alioto, and M. Standard. Cell-based ultra-compact dacs in 40-nm cmos. *IEEE Access*, pages 26479–12648, 2019.
- [43] Z. Xu, N. Ojima, S. Li, and T. Iizuka. An all-standard-cell-based synthesizable sar adc with nonlinearity-compensated rdac. *IEEE Trans. Very Large Scale Integr. (Vlsi) Syst.*, pages 2153–2162, 2021.
- [44] E. Abramov, T. Vekslender, O. Kirshenboim, and M. M. Peretz. Fully integrated digital average current-mode control voltage regulator module ic. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 6(2):485–499, June 2018.
- [45] J. Dondo Gazzano. Integrating fpgas: A dynamically reconfigurable fpga-based grid for high performance computing. In *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, pages 1–4, 2016.
- [46] I. Firmansyah, Y. Yamaguchi, and T. Boku. Performance evaluation of stratix v de5-net fpga board for high performance computing. In *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pages 23–27, 2016.
- [47] T. Vekslender, E. Abramov, Y. Lazarev, and M. Peretz. Fully-integrated digital average current-mode control 12v-to-1.xv voltage regulator module ic. In *2017 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2043–2050, March 2017.
- [48] A. V. Peterchev, Jinwen Xiao, and S. R. Sanders. Architecture and ic implementation of a digital vrm controller. *IEEE Transactions on Power Electronics*, 18(1):356–364, Jan 2003.
- [49] J. Monson, M. Wirthlin, and B. L. Hutchings. Implementing high-performance, low-power fpga-based optical flow accelerators in c. In *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, pages 363–369, 2013.
- [50] B. Belean, S. Pogacian, and A. Bot. Fpga based hardware architectures for high performance computing applications. In *2012 5th Romania Tier 2 Federation Grid, Cloud High Performance Computing Science (RQLCG)*, pages 11–14, 2012.
- [51] A. Tiwari. A low power high speed dual data rate acquisition system using fpga.

- In *2012 International Conference on Communication, Information Computing Technology (ICCICT)*, pages 1–4, Oct 2012.
- [52] R. Rodriguez Ponce and D. V. Talavera. Fpga implementation of decimal division for motor control drives. *IEEE Latin America Transactions*, 14(9):3980–3985, Sep. 2016.
- [53] M. D. Ercegovac and R. McIlhenny. Design and fpga implementation of radix-10 combined division/square root algorithm with limited precision primitives. In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 87–91, Nov 2010.
- [54] I. Takahashi and T. Noguchi. A new quick-response and high-efficiency control strategy of an induction motor. *IEEE Transactions on Industry Applications*, IA-22(5):820–827, Sep. 1986.
- [55] G. Wang and Y. Chiu. Fast fpga emulation of background-calibrated sar adc with internal redundancy dithering. *IEEE 2013 Custom Integrated Circuits Conference*, pages 1–4, September 2013.
- [56] M. Chen, C. Zhou, W. He, F. Ye, and R. A Junyan. Split-based neural network calibrator for sar-pipelined adc on fpga. *2020 IEEE 15th International Conference on Solid-State Integrated Circuit Technology (ICSICT)*, pages 1–3, November 2020.
- [57] L. Y. Hooi, L. H. Hiung, M. Drieberg, and P. Sebastian. Configurable 2 bits per cycle successive approximation register for analog to digital converter on fpga. *2016 6th International Conference on Intelligent and Advanced Systems (ICIAS)*, pages 1–5, August 2016.
- [58] G. D. Colletta, L. H. C. Ferreira, T. C. Pimenta, and P. C. Crepaldi. A successive approximation a/d converter using a pwm modulator dac. In *2012 IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference*, pages 333–337, Nov 2012.
- [59] Ian Hickman. Chapter 6 - active signal processing in the frequency domain. In Ian Hickman, editor, *Analog Electronics (Second Edition)*, pages 125 – 146. Newnes, Oxford, second edition edition, 1999.
- [60] M. T. Kyu, Z. M. Aung, and Z. M. Naing. Design and implementation of active filter for data acquisition system. In *2009 International Conference on Information Management and Engineering*, pages 406–410, April 2009.
- [61] A. Principios de electronica Malvino. Mcgraw-hill. 2007, February 2022.
- [62] W. Menzel and F. Bogelsack. Bessel low pass filter in mixed planar waveguide techniques. In *1999 29th European Microwave Conference*, volume 3, pages 191–194, Oct 1999.
- [63] A. P. James, Y. Tolegen, and A. Nandakumar. Frequency analysis of memristor based low pass bessel filter. In *2018 International Conference on Computing and Network Communications (CoCoNet)*, pages 185–189, Aug 2018.
- [64] John Hamburger Bob Dobkin. *Design Note Collection*, page 1136. Elsevier, 2015.

- [65] George S. Moschytz. *Analog Circuit Theory and Filter Design in the Digital World*. Springer, 2019.
- [66] Robert A. Pease. *World Class Designs*. Elsevier Inc, 2008.
- [67] Bruce Carter and Ron Mancini. Chapter 16 - active filter design techniques. In Bruce Carter and Ron Mancini, editors, *Op Amps for Everyone (Fifth Edition)*, pages 199 – 258. Newnes, fifth edition edition, 2018.
- [68] Thomas Kugelstadt. *Op Amps for Everyone*. Elsevier, 2017.
- [69] J. Karki. Analysis of the sallen-key architecture. Technical report, Texas Instrument, 2002.
- [70] Hank Zumbahlen and Analog Devices Inc. *Linear Circuit Design Handbook*. Newnes, USA, 2008.
- [71] Hank Zumbahlen. *Linear Circuit Design Handbook*. O’Reilly Media, Inc, 2011.
- [72] Michael Steffes. Design methodology for mfb filters in adc interface application. Technical report, Texas Instrument, 2006.
- [73] R. Jacob Baker. *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 3rd edition, 2010.
- [74] Igor M. Filanovsky Vadim V. Ivanov. *Operational Amplifier Speed and Accuracy Improvement*. Kluwer Academic Publishers, 2004.
- [75] Texas Instruments Incorporated. *FilterPro™ User’s Guide*. Texas Instruments Incorporated.
- [76] H. Ahmed and P. J. Spreadbury. *Negative feedback*, page 164–189. Electronics Texts for Engineers and Scientists. Cambridge University Press, 2 edition, 2011.
- [77] M. A. David. Using pwm output as a digital-to-analog converter on a tms320f280x digital signal controller. *Texas Instrument*, 2008, February 2022.
- [78] B. Pwm Dac Nisarga. Using msp430 high-resolution timer. *Texas Instrument*, 2013, February 2022.
- [79] M. Mitchell. Using pwm timer b as a dac. *Texas Instrument*, 2008, February 2022.
- [80] Z. Lukic, N. Rahman, and A. Prodie. Multibit $\sigma \delta$ pwm digital controller ic for dc-dc converters operating at switching frequencies beyond 10 mhz. *IEEE Trans. Power Electron*, pages 1693–1707, 2007.
- [81] W. Menzel and F. Bogelsack. Bessel low pass filter in mixed planar waveguide techniques. In *1999 29th European Microwave Conference*, volume 3, pages 191–194, Oct 1999.
- [82] R. H. Mehta, K. Jasani, D. Israni, and A. Shah. Fpga realization of novel techniques for ddr based data acquisition system. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 511–515, 2016.
- [83] A. Tiwari. A low power high speed dual data rate acquisition system using fpga. In *2012 International Conference on Communication, Information Computing Technology (ICCICT)*, pages 1–4, 2012.

- [84] H. HAMIL, M. S. AZZAZ, S. SAKHI, A. MAALI, Z. Zidelmal, and D. O. Abdeslam. Design and fpga real-time implementation of pwm and ppm modulation for ultra wide band applications. In *2019 6th International Conference on Image and Signal Processing and their Applications (ISPA)*, pages 1–5, Nov 2019.
- [85] Xilinx. *7 Series FPGAs Select IO Resources User's Guide*. Xilinx.
- [86] LINEAR TECHNOLOGY CORPORATION. *LT1364 Dual and Quad 70MHz, 1000V/ μ s Op Amps*, 4 2013. Rev. 3.
- [87] Luis Enrique Avendaño. *Sistemas electrónicos analógicos-Un enfoque matricial*. Universidad Tecnologica de Pereira, 2007.
- [88] Muhammad H. Rashid. *Microelectronic Circuits: Analysis and Design*. Cengage learning, 2016.
- [89] Texas Instruments. *Amplificador operacional THS3201DBVT Realimentación de corriente, 9 V, 12 V 1.8GHz SOT-23, 5 pines*, 4 2010. Rev. 2.
- [90] Vesselina Atanasova Barzinska and Emilian Stefanov Georgiev. High precision rc-oscillator using r-2r ladder for digital calibration. In *2016 XXV International Scientific Conference Electronics (ET)*, pages 1–4, 2016.
- [91] N V Uma Reddy, Rajesh Babu C, and M V Chaitanyakumar. A 4 bit 5ghz r-2r digital to analog converter using hetrojunction hemt. In *2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA)*, pages 1–6, 2013.
- [92] Chun-Chieh Chen and Nan-Ku Lu. Nonlinearity analysis of r-2r ladder-based current-steering digital to analog converter. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 833–836, 2013.
- [93] V. Radhika and K. Baskaran. Fpga based dpwm/dpfm architecture for digitally controlled dc-dc converters. In *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, pages 78–82, 2016.
- [94] S. Debnath, J. Qin, and M. Saeedifard. A distributed pwm strategy for modular multilevel converter. In *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pages 1014–1020, 2014.
- [95] L. Hernandez and E. Prefasi. Analog to digital conversion using a pulse width modulator and an irregular sampling decoder. In *2006 13th IEEE International Conference on Electronics, Circuits and Systems*, pages 716–719, 2006.
- [96] V. Radhika and K. Baskaran. Hybrid digital pulse width modulator architecture using fpga. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 123–126, 2016.
- [97] V. Sabarinath and K. Sivanandam. Design and implementation of fpga based high resolution digital pulse width modulator. In *2013 International Conference on Communication and Signal Processing*, pages 410–414, 2013.
- [98] K. Shimazoe, H. Takahashi, B. Shi, T. Furumiya, J. Ooi, Y. Kumazawa, and H. Murayama. Novel front-end pulse processing scheme for pet system based on

- pulse width modulation and pulse train method. *IEEE Trans. Nucl. Sci.*, 2010:782–786.
- [99] D. G. Zrilic. Alternative approach to use of pulse width modulation. *2006 World Automation Congress*, pages 24–26, July 2006.
- [100] H. Lin, R. Chen, R. Li, L. Zhu, H. Yan, and Z. Shu. A flexible and fast space vector pulse width modulation technique for multilevel converters. *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, pages 1–4, August 2019.