

INDICE

Agradecimientos.....	vii
Resumen	ix
Abstract.....	xi
Resum.....	xiii
Índice.....	xv
Lista de figuras.....	xix
Lista de tablas.....	xxiii

Capítulo 1: Introducción	1
1.1. Presentación y Contexto	2
1.2. Objetivos	4
1.3. Estructura de la Memoria	5

I. APROXIMACION DE FUNCIONES ELEMENTALES

Capítulo 2: Métodos para la Aproximación de Funciones Elementales	9
2.1. Introducción	10
2.1.1. Reducción de Rango	12
2.2. Aproximación de Funciones Elementales	13
2.2.1. CORDIC	14
2.2.2. Métodos de Convergencia Lineal.....	17
2.2.2.1. Normalización Multiplicativa	17
2.2.2.2. Normalización Aditiva.....	18
2.2.3. Métodos Basados en Recurrencia de Dígitos	21
2.2.4. Métodos Basados en Tablas	22
2.2.4.1. Tablas Directas	22
2.2.4.2. Interpolación Lineal	24
2.2.4.3. Tablas Bipartidas.....	26
2.2.4.4. Tablas Bipartidas Simétricas (SBTM)	29
2.2.4.5. Tablas Multipartidas - (STAM)	32
2.2.5. Aproximaciones por Polinomios	35
2.2.6. Aproximaciones por División de Polinomios	37
2.2.7. Aproximaciones por Series de Taylor	38
2.3. Conclusiones	40

Capítulo 3: Aproximación de la $\text{Atan}(y/x)$ basada en Métodos de Tablas	43
3.1. Introducción.....	44
3.2. Arquitectura Propuesta	45
3.2.1. Etapa de Pre-procesado	46
3.2.2. Etapa de Post-procesado	47
3.2.3. Cálculo de la División.....	47
3.2.4. Cálculo de la $\text{atan}(z)$	51
3.3. Implementación y Resultados	51

3.3.1. Comparación de Consumos por las Distintas Implementaciones	57
3.4. Conclusiones	58
Capítulo 4: Aproximación de la $\text{Atan}(y/x)$ basada en la División Logarítmica.	61
4.1. Introducción	62
4.2. Arquitectura Propuesta	62
4.2.1. Aproximación del Logaritmo.....	64
4.2.1.1. Métodos para la Corrección de la Aproximación de Mitchell	65
4.2.1.1.a. Métodos basados en interpolación lineal mediante desplazamientos y sumas.....	65
4.2.1.1.b. Métodos basados en LUTs.....	68
4.2.1.2. Método Propuesto.....	68
4.2.1.3. Resultados de Implementación de la Aproximación del Logaritmo Propuesta y Comparación de Resultados con los Distintos Métodos	71
4.2.1.4. Aproximación del $\log(1+x)$ mediante LUTs	73
4.3. Aproximación de la $\text{Atan}(y/x)$	75
4.3.1. Etapas de Pre-procesado y Post-procesado.....	76
4.3.2. Divisor Logarítmico.....	76
4.3.3. Aproximación de la $\text{Atan}(2^n)$	77
4.3.3.1. Aproximación basada en LUTs con Segmentación No-uniforme.....	77
4.3.3.2. Aproximación basada en Rectas y Error-LUT.....	81
4.4. Implementación y Resultados.....	84
4.4.1. Comparación de Resultados Obtenidos de la Implementación Propuesta con Diferentes Diseños	89
4.5. Conclusiones	92

II. GENERACION DE NUMEROS ALEATORIOS GAUSIANOS

Capítulo 5: Generadores de Números Aleatorios Gaussianos.....	97
5.1. Introducción.....	98
5.1.1. Caracterización de un Canal de Ruido Aditivo Blanco Gaussiano	100
5.2. Generación de Variables Aleatorias Gaussianas	103
5.2.1. Método de la Inversión de la Función de Distribución Acumulada (ICDF).....	104
5.2.2. Método Box-Muller.....	105
5.2.3. Método Polar.....	106
5.2.4. Método Ziggurat.....	107
5.2.5. Método Iterativo o de Wallace.....	108
5.2.6. Método CLT (Central Limit Theorem)	110
5.3. Generación de Variables Aleatorias Uniformemente Distribuidas.....	110
5.3.1. Generadores Congruenciales Lineales.....	111
5.3.2. Generadores Congruenciales de Fibonacci.....	111
5.3.3. Generador suma-con-acarreo, resta-con-acarreo, multiplicación-con-acarreo.....	112
5.3.4. Lineal Feedback Shift register (LFSR).....	112
5.4. Test Estadísticos	114
5.4.1. Test Chi-cuadrado χ^2	115
5.4.2. Test Kolgomorov-Smirnov	116
5.4.3. Test Anderson-Darling.....	118
5.4.4. Paquetes de Test	119
5.5. Conclusiones	119

Capítulo 6: Generador de ruido AWGN de altas Prestaciones basado en el Método de Inversión	121
6.1. Introducción	122
6.2. Implementación de la Aproximación de la Inversa de la Función de Distribución Acumulada (ICDF)	124
6.2.1. Segmentación No-uniforme de la ICDF	126
6.3. Arquitectura Propuesta	135
6.3.1. Generador de Números Uniformemente Distribuidos (GNUD).....	136
6.3.2. Generador de Direcciones	137
6.3.3. Interpolación Polinómica.....	138
6.4. Resultados de Implementación	139
6.4.1. Arquitectura Propuesta Modificada.....	143
6.5. Verificación de las Muestras Generadas	148
6.6. Conclusiones	157
Capítulo 7: Conclusiones.....	159
7.1. Conclusiones	160
7.2. Líneas Futuras	162
7.3. Resultados Publicados durante la Tesis Doctoral.....	163
Bibliografía	165



LISTA DE FIGURAS

Figura 1.1 – Valores de la Función de Distribución de Probabilidad (PDF) de la distribución Gaussiana comprendida entre 6σ y 10σ . Representación en escala logarítmica del eje de las ordenadas	2
Figura 2.1 – Distribución del numero de etapas necesarias para la implementación del operador CORDIC mediante el Core-IP de Xilinx.....	16
Figura 2.2 – Diagrama de bloques de la aproximación de la $\text{atan}(y/x)$ mediante métodos de convergencia lineal.....	19
Figura 2.3 – Algoritmo básico para la realización de la división <i>non-restoring</i> en base 2.....	19
Figura 2.4 – Tamaño de las tablas necesarias para realizar la aproximación por LUTs de una función elemental.....	23
Figura 2.5 – Diagrama de bloques de la aproximación de la $\text{atan}(y/x)$ mediante el uso de interpolación lineal.....	25
Figura 2.6 – División de la palabra de entrada en tres partes para el direccionamiento de las dos LUTs necesarias.....	26
Figura 2.7 – Valores de las tablas a_0 y a_1 utilizadas en el cálculo de la aproximación del recíproco.....	27
Figura 2.8 – Estructura hardware para la implementación de la aproximación mediante LUTs bipartidas	28
Figura 2.9 – Implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUT bipartida.....	28
Figura 2.10 – Valores de las tablas a_0 y a_1 utilizadas en el cálculo del recíproco mediante el método SBTM.....	30
Figura 2.11 – Estructura hardware para la implementación de la aproximación mediante el método SBTM.....	31
Figura 2.12 – Implementación de la aproximación de la $\text{atan}(y/x)$ mediante el método SBTM.....	31
Figura 2.13 – División de la palabra de entrada en $m+1$ partes para el direccionamiento de las m LUTs necesarias.....	32
Figura 2.14 – Estructura hardware para la implementación de la aproximación mediante el método STAM	34
Figura 2.15 – Tamaño de las tablas utilizadas para la aproximación de la $\text{atan}(y/x)$	34
Figura 2.16 – Implementación de la aproximación de la $\text{atan}(y/x)$ mediante el método STAM.....	35
Figura 2.17 – Implementación de la aproximación de la $\text{atan}(y/x)$ mediante la serie de Taylor de dos variables	40

Figura 3.1 – Arquitectura propuesta para la aproximación de la $\text{atan}(y/x)$ mediante LUTs.....	45
Figura 3.2 – Diagrama de bloques de la aproximación de la $\text{atan}(y/x)$ basada en la división de las dos entradas y la utilización de LUTs.	46
Figura 3.3 – Etapa de pre-procesado	46
Figura 3.4 – Etapa de post-procesado	47
Figura 3.5 – Cálculo de la división y'/x'	47
Figura 3.6 – Bloque normalizador de las entradas para el divisor.....	48
Figura 3.7.a – LZD de dos bits. Figura 3.7.b – LZD de ocho bits.....	48
Figura 3.8 – <i>Barrel-shifter</i> implementado mediante multiplicadores.....	49
Figura 3.9 – Esquema de memoria bipartida utilizada para la aproximación del recíproco.....	50
Figura 3.10 – Simulación del error cometido en la arquitectura propuesta expresado en bits	51
Figura 3.11 – Esquemas de las memorias multipartidas (3 LUTs) utilizadas en la aproximación propuesta. (a) recíproco, (b) $\text{atan}(z)$	55
Figura 3.12 – Esquemas de las memorias multipartidas (4 LUTs) utilizadas en la aproximación propuesta. (a) recíproco, (b) $\text{atan}(z)$	56
Figura 3.13 – Comparativa del consumo de potencia obtenido por las diferentes implementaciones y modificando el número de etapas de segmentación	57
Figura 3.14 – Comparativa del consumo de potencia obtenido por las diferentes implementaciones y modificando el número de etapas de segmentación y utilizando únicamente <i>slices</i>	58
Figura 4.1.a – Segmentación uniforme de la aproximación de la $\text{atan}(2^n)$.	
Figura 4.1.b – Segmentación no-uniforme de la aproximación de la $\text{atan}(2^n)$	64
Figura 4.2 – Error cometido por la aproximación de Mitchell.....	65
Figura 4.3 – Interpolación lineal utilizada en los métodos de dos regiones y su error cometido.....	66
Figura 4.4 – Interpolación lineal utilizada en los métodos de cuatro regiones y su error cometido	67
Figura 4.5 – Interpolación lineal utilizada en los métodos de seis regiones y su error cometido.....	68
Figura 4.6 – Interpolación lineal utilizada en el método de cuatro regiones propuesto y su error cometido.....	69
Figura 4.7 – Error cometido en la aproximación del logaritmo mediante interpolación rectas y error-LUT.....	70
Figura 4.8 – Arquitectura propuesta para la aproximación del $\log(x)$	70
Figura 4.9 – Arquitectura propuesta para la aproximación del $\log(1+x)$ mediante rectas para una entrada de 32 bits	71
Figura 4.10 – Arquitectura realizada para comparar resultados de implementación con [56]	73
Figura 4.11 – Arquitectura propuesta para la aproximación del $\log(x)$ mediante LUTs multipartidas	74
Figura 4.12 – Arquitectura propuesta para la aproximación $\text{atan}(y/x)$ mediante división logarítmica.....	75
Figura 4.13 – Diagrama de bloques de la implementación propuesta para la aproximación de la $\text{atan}(y/x)$	76
Figura 4.14.a – Etapa pre-procesado. Figura 4.14.b – Etapa post-procesado.....	76
Figura 4.15.a – Divisor Logarítmico. Figura 4.15.b – Divisor Logarítmico multiplexado	77
Figura 4.16.a – Emplazamiento óptimo de los segmentos para la aproximación de la $\text{atan}(2^n)$. Figura 4.16.b – Error cometido en la aproximación utilizando segmentación no-uniforme expresado en bits.....	78
Figura 4.17 – Esquema de memoria utilizado para la aproximación de la $\text{atan}(2^n)$ mediante LUTs	80
Figura 4.18 – Implementación generador direcciones de la aproximación basada en LUTs y segmentación no-uniforme.....	80
Figura 4.19 – Bloque utilizado para el cálculo segmento direccionado	81
Figura 4.20 – Diseño propuesto para la aproximación de la $\text{atan}(2^n)$ mediante rectas y error-LUT.....	82
Figura 4.21 – Rectas utilizadas para la aproximación de la $\text{atan}(2^n)$	82

Figura 4.22 – Error cometido en la aproximación por rectas y que será almacenado en la error-LUT.....	83
Figura 4.23 – Error cometido por la aproximación por rectas y rectas más error-LUT expresado en bits.....	83
Figura 4.24 – Simulación completa de la aproximación de la $\text{atan}(y/x)$ mediante aproximación del logaritmo por rectas y aproximación de la $\text{atan}(2^n)$ mediante LUT multipartida y segmentación no-uniforme	84
Figura 4.25 – Comparativa de área y velocidad de los distintos métodos propuestos para la aproximación $\text{atan}(y/x)$	86
Figura 5.1 – Elementos básicos de un sistema de Comunicaciones Digital.....	98
Figura 5.2 – Modelo de un canal AWGN utilizado en comunicaciones.....	100
Figura 5.3 – Función Distribución Probabilidad de la distribución gaussiana.....	101
Figura 5.4 – Función Distribución Acumulada Probabilidad de la distribución Gaussiana	102
Figura 5.5 – Funcionamiento del método de la Inversión	104
Figura 5.6 – División de la distribución gaussiana por medio de rectángulos, cuñas y la región de cola	108
Figura 5.7 – Pseudocódigo del algoritmo para la realización del método Ziggurat	108
Figura 5.8 – Pseudocódigo del algoritmo para la realización del método Wallace.....	109
Figura 5.9 – Implementación hardware de un LFSR de 9 bits.....	113
Figura 5.10 – Implementación hardware de un GFSR de 8 bits.....	113
Figura 5.11 – Generador Tausworthe de 32 bits y periodo 2^{88}	114
Figura 5.12 – Comparación entre la CDF empírica (ECDF) y la CDF ideal de la distribución Gaussiana.....	118
Figura 6.1 – Generación de muestras aleatorias Gaussianas mediante el método de la inversión.....	124
Figura 6.2 – Esquema de implementación y validación del método de inversión propuesto.....	125
Figura 6.3 – Inversa de la Función Distribución Acumulada de la distribución Gaussiana.....	126
Figura 6.4 – Segmentación no-uniforme utilizada para la aproximación de la ICDF de la distribución Gaussiana.....	128
Figura 6.5 – Relación entre la desviación máxima (σ) de la distribución Gaussiana y el número de bits necesarios en el GNUD.....	128
Figura 6.6 – Pseudocódigo del algoritmo de cálculo de la segmentación no-uniforme de la función ICDF.....	130
Figura 6.7 – Esquema de direccionamiento utilizado en la aproximación de la función ICDF.....	131
Figura 6.8 – Error cometido (ulp) en la aproximación no-uniforme de la función ICDF.....	133
Figura 6.9 – Arquitectura propuesta para la aproximación de la función ICDF mediante segmentación no-uniforme y con polinomio interpolador de grado dos	133
Figura 6.10 – Variación del numero de segmentos para diferentes precisiones de salida y distintos grados de polinomio interpolador.....	134
Figura 6.11 – Variación del numero de segmentos para diferentes valores de desviación y distintos grados de polinomio interpolador.....	134
Figura 6.12 – Arquitectura propuesta para la aproximación de la función ICDF de la distribución Gaussiana.....	135
Figura 6.13 – Generador Tausworthe de 64 bits.....	136
Figura 6.14 – Diagrama de bloques del generador de direcciones utilizado para la segmentación no-uniforme.....	137
Figura 6.15 – Implementación de la interpolación lineal y cuadrática de la función ICDF.....	138
Figura 6.16 – Esquema de memoria utilizados para la interpolación cuadrática en distintas FPGAs	139
Figura 6.17 – Distribución de las etapas de segmentado en los distintos bloques de la aproximación propuesta.....	140
Figura 6.18 – Arquitectura del GNAG propuesto. Arq. modificada A.....	144
Figura 6.19 – Generador de direcciones utilizado en la arquitectura B.....	145

Figura 6.20 – Esquema asignación Bits para direccionamiento e interpolación arquitectura C: (a) el CMS aparece entre los bits del 127 al 20; (b) el CMS aparece dentro de (b) los bits del 19 al 3.....	145
Figura 6.21 – Circuito generador de la máscara utilizado en la arquitectura C.....	146
Figura 6.22 – Generador de direcciones utilizado en la arquitectura C.....	146
Figura 6.23 – Comparación de la PDF de las muestras generadas con respecto a la PDF ideal de la distribución Gaussiana.....	149
Figura 6.24.a – PDF de las muestras generadas comparadas con la distribución ideal en el tramo 6.2σ a 7.8σ	150
Figura 6.24.b – PDF de las muestras generadas comparadas con la distribución ideal en el tramo 7.8σ a 9.2σ	150
Figura 6.24.c – PDF de las muestras generadas comparadas con la distribución ideal en el tramo 9.2σ a 10.6σ	151
Figura 6.24.d – PDF de las muestras generadas comparadas con la distribución ideal en el tramo 10.6σ a 12σ	151
Figura 6.24.e – PDF de las muestras generadas comparadas con la distribución ideal en el tramo 12σ a 13.1σ	151
Figura 6.25 – Simulación del error cometido en la implementación del GNAG propuesto en términos de ulp.....	152
Figura 6.26.a – Diagrama de dispersión de las muestras generadas por el GNUD.	
Figura 6.26.b – Ampliación del diagrama de dispersión.....	152
Figura 6.27 – Diagrama de dispersión de las muestras generadas por el GNAG.....	153
Figura 6.28 – Diagrama de Bloques para la implementación de la medida del BER.....	156
Figura 6.29 – Comparación entre la grafica del BER obtenido por el GNAG propuesto y el BER de la codificación BPSK ideal.....	157

LISTA DE TABLAS

Tabla 2.1 – Rango de valores de entrada óptimos para la aproximación de las funciones elementales más utilizadas	12
Tabla 2.2 – Valores de las constantes utilizadas en el operador CORDIC.....	15
Tabla 2.3 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante CORDIC y modificando el numero de etapas de segmentación.....	16
Tabla 2.4 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante CORDIC y modificando la precisión de salida de la aproximación	17
Tabla 2.5 – Resultados de implementación del divisor <i>non-restoring</i> para distintas precisiones de salida.....	20
Tabla 2.6 – Resultados de implementación del divisor modificando el número divisiones por ciclo	20
Tabla 2.7 – Valores almacenados para la aproximación de la $\text{atan}(2^i)$ para una precisión de 10 bits.....	21
Tabla 2.8 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante métodos de convergencia lineal y modificando la precisión de salida de la aproximación	21
Tabla 2.9 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUT directas	23
Tabla 2.10 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante el uso de interpolación lineal.....	26
Tabla 2.11 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUT bipartidas	29
Tabla 2.12 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante el método SBTM.....	32
Tabla 2.13 – Resultados de implementación de la aproximación $\text{atan}(y/x)$ mediante el método STAM	35
Tabla 2.14 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante polinomios y modificando la precisión de salida de la aproximación.....	37
Tabla 2.15 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante el método de Taylor y modificando la precisión de salida de la aproximación	40
Tabla 2.16 – Comparativa de resultados de las diferentes implementaciones de la aproximación de la $\text{atan}(y/x)$ para una salida de baja aproximación.....	41
Tabla 2.17 – Comparativa de resultados de las diferentes implementaciones de la aproximación de la $\text{atan}(y/x)$ para una salida de baja aproximación.....	42
Tabla 3.1 – Particionado óptimo de la LUT bipartida utilizada para la aproximación del recíproco... 50	
Tabla 3.2 – Particionado óptimo de la LUT bipartida utilizada en la aproximación de la $\text{atan}(z)$	51
Tabla 3.3 – Resultados de implementación de la arquitectura propuesta modificando el número de etapas de segmentación	52
Tabla 3.4 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUTs y modificando la precisión de la aproximación generada	52
Tabla 3.5 – Resultados de la implementación de alta velocidad de la $\text{atan}(y/x)$ mediante LUTs y modificando la precisión de la aproximación generada	53
Tabla 3.6 – Resultados de la implementación para altas precisiones de la $\text{atan}(y/x)$ basada en LUTs y utilizando LUTs multipartidas	53
Tabla 3.7 – Resultados de las medidas consumo de la aproximación de la $\text{atan}(y/x)$ mediante LUTs y modificando el número de etapas de segmentación y	

utilizando elementos embebidos de la FPGA	54
Tabla 3.8 – Resultados de las medidas consumo de la aproximación de la $\text{atan}(y/x)$ mediante LUTs y modificando el número de etapas de segmentación y únicamente con slices.....	54
Tabla 3.9 – Particionado óptimo de la LUT multipartida utilizada para la aproximación del recíproco.	55
Tabla 3.10 – Particionado óptimo de la LUT multipartida utilizada para la aproximación de la $\text{atan}(z)$	55
Tabla 3.11 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUTs multipartidas (3 LUT) y modificando el numero de etapas de segmentación	56
Tabla 3.12 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ mediante LUTs multipartidas (4 LUT) y modificando el numero de etapas de segmentación	57
Tabla 4.1 – Constantes utilizadas para la aproximación logaritmo por los métodos de dos regiones	66
Tabla 4.2 – Constantes utilizadas para la aproximación logaritmo por los métodos de cuatro regiones	67
Tabla 4.3 – Resultados de implementación de la aproximación del $\log(x)$ para diferentes precisiones de salida	71
Tabla 4.4 – Comparación de resultados de implementación para las distintas aproximaciones del $\log(1+x)$ basadas en rectas	72
Tabla 4.5 – Tamaño de las memorias utilizadas por la aproximación $\log(1+x)$ para altas precisiones.....	72
Tabla 4.6 – Comparación de los resultados obtenidos por la arquitectura propuesta y la arquitectura propuesta en [56].....	72
Tabla 4.7 – Tamaño de las memorias utilizadas para la aproximación $\log(1+x)$ mediante LUTs multipartidas para diferentes precisiones	73
Tabla 4.8 – Resultados de implementación de la aproximación del $\log(x)$ basada en LUTs multipartidas para diferentes precisiones de salida	74
Tabla 4.9 – Tamaños de las tablas utilizadas por los distintos segmentos de la aproximación $\text{atan}(2^n)$	79
Tabla 4.10 – Tamaño memoria error-LUT utilizada aproximación $\text{atan}(y/x)$ mediante rectas.....	82
Tabla 4.11 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ basada en LUTs y segmentación no-uniforme.....	85
Tabla 4.12 – Resultados implementación aproximación $\text{atan}(y/x)$ basada en rectas y error-LUT	85
Tabla 4.13 – Resultados implementación aproximación $\text{atan}(y/x)$ basada en LUTs multipartidas.....	85
Tabla 4.14 – Resultados de implementación de la $\text{atan}(y/x)$ mediante LUTs y segmentación no-uniforme. El operador logaritmo esta implementado con LUTs multipartidas	87
Tabla 4.15 – Resultados de implementación de la $\text{atan}(y/x)$ mediante rectas y error-LUT. El operador logaritmo esta implementado con LUTs multipartidas	87
Tabla 4.16 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ basada en LUTs y segmentación no-uniforme. Operador logaritmo multiplexado.....	88
Tabla 4.17 – Resultados de implementación de la aproximación de la $\text{atan}(y/x)$ basada en rectas y error-LUTs. Operador logaritmo multiplexado	88
Tabla 4.18 – Resultados de implementación aproximación alta velocidad de la $\text{atan}(y/x)$ basada en LUTs y segmentación no-uniforme	89
Tabla 4.19 – Resultados implementación aproximación alta velocidad de la $\text{atan}(y/x)$ basada en rectas y error-LUT	89
Tabla 4.20 – Comparativa de resultados de implementación de la aproximación de la $\text{atan}(y/x)$ para baja precisión de salida.....	90

Tabla 4.21 – Comparativa de resultados de implementación de la aproximación de la $\text{atan}(y/x)$ para alta precisión de salida.....	91
Tabla 4.22 – Comparativa de resultados de implementación de la aproximación de la $\text{atan}(y/x)$ para baja precisión de salida. Barrel-shifter implementado mediante multiplicadores embebidos.	91
Tabla 4.23 – Comparativa de resultados de implementación de la aproximación de la $\text{atan}(y/x)$ para baja precisión de salida. Barrel-shifter implementado mediante multiplicadores embebidos.	92
Tabla 5.1 – Valores calculados de la distribución χ^2 para los diferentes grados de libertad (k,α)	117
Tabla 5.2 – Valores críticos del test Anderson-Darling para $\alpha=0.05$	119
Tabla 6.1 – Resultados de implementación del GNAG propuesto utilizando interpolación cuadrática para diferentes dispositivos	140
Tabla 6.2 – Resultados de implementación del GNAG propuesto utilizando interpolación lineal para diferentes dispositivos	140
Tabla 6.3 – Recursos hardware utilizados por los distintos bloques del GNAG propuesto y utilizando interpolación cuadrática.....	141
Tabla 6.4 – Resultados implementación del GNAG propuesto modificando la precisión de las muestras generadas a la salida y un valor de desviación fijo.....	142
Tabla 6.5 – Resultados implementación del GNAG propuesto modificando el valor de desviación máxima y un tamaño de las muestras generadas fijo.....	142
Tabla 6.6 – Resultados de la implementación modificando el tipo de recursos de la FPGA	143
Tabla 6.7 – Resultados de implementación de los diferentes GNAG propuestos utilizando interpolación cuadrática.....	147
Tabla 6.8 – Recursos hardware utilizados por la arquitectura C propuesta y utilizando interpolación cuadrática.....	147
Tabla 6.9 – Comparativa de los resultados de implementación obtenidos por diversas implementaciones de GNAG	148
Tabla 6.10 – Resultados obtenidos por el GNAG en las simulaciones del test $\chi^2_{0.05,247}$	154
Tabla 6.11 – Resultados obtenidos por el GNAG en las simulaciones del test $\chi^2_{0.05,247}$. Arquitecturas modificadas	154
Tabla 6.12 – Resultados obtenidos por el GNAG en las simulaciones del test A-D	155
Tabla 6.13 – Resultados obtenidos por el GNAG en las simulaciones del test A-D. Arquitecturas modificadas	155

