

MULTIPLICACIÓN SECUENCIAL EN DISPOSITIVOS LÓGICOS PROGRAMABLES

Ing. Marcos Funes

Este Trabajo de Tesis fue presentado al Departamento de Electrónica
de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata
el 8 de Octubre de 2007, como requisito parcial para la obtención del título de

Doctor en Ingeniería. Mención Electrónica

Director: Dr. Daniel Carrica

Co-Director: Ing. Mario Benedetti



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

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

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



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

MULTIPLICACIÓN SECUENCIAL EN DISPOSITIVOS LÓGICOS PROGRAMABLES

Ing. Marcos Funes

Este Trabajo de Tesis fue presentado al Departamento de Electrónica
de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata
el 8 de Octubre de 2007, como requisito parcial para la obtención del título de
Doctor en Ingeniería. Mención Electrónica

Director: Dr. Daniel Carrica

Co-Director: Ing. Mario Benedetti

A mis Padres.

A Andrea.

A Nicolás, Alejandro y Fernando

Índice general

Agradecimientos	xvii
Resumen	xviii
Nomenclatura	xx
1. Introducción	1
1.1. Estructuras de Multiplicadores	2
1.2. Objetivos de la tesis	4
1.3. Organización de la tesis	4
2. Arquitecturas de Multiplicadores	6
2.1. Introducción	6
2.2. Algoritmo de la Multiplicación	7
2.2.1. Operandos en punto fijo	7
2.2.2. Operandos en punto flotante	9
2.3. Adición	12
2.3.1. Semisumador	12
2.3.2. Sumador Total	13
2.3.3. Suma por Propagación de Acarreo	15
2.3.4. Suma Carry-Save	16

2.3.5. Suma Multi-operandos	17
2.4. Multiplicadores Paralelos	19
2.4.1. Multiplicador por Tabla de Look-up	19
2.4.2. Multiplicador Ripple Carry	20
2.4.3. Multiplicador Carry Save	22
2.4.4. Multiplicador Guild	23
2.4.5. Multiplicador McCanny-McWhinter	25
2.5. Consumo de recursos lógicos	25
2.5.1. Operandos en Punto Fijo	25
2.5.2. Operandos en punto flotante	31
2.6. Conclusiones	32
3. Nuevas Arquitecturas de Multiplicadores	33
3.1. Introducción	33
3.2. Multiplicación Secuencial	34
3.2.1. Algoritmo	34
3.2.2. Implementación	37
3.2.3. Multiplicador Secuencial de Base 4	45
3.3. Arquitecturas Propuestas en Punto Fijo	56
3.3.1. Multiplicador Secuencial Sin Entradas Registradas	56
3.3.2. Multiplicador Secuencial Fraccionado	61
3.3.3. Multiplicador de Sumas Consecutivas	67
3.4. Comparación de los multiplicadores	75
3.4.1. Multiplicadores optimizados en consumo de recursos	75
3.4.2. Multiplicadores optimizados en velocidad	76
3.4.3. Performance de los multiplicadores	78
3.5. Arquitecturas Propuestas en Punto Flotante	80

3.5.1. Variante Multiplicador Secuencial Sin Entradas Registradas	80
3.5.2. Variante Multiplicador Secuencial de Sumas Consecutivas	86
3.6. Conclusiones	90
4. Resultados Experimentales	92
4.1. Introducción	92
4.2. Parámetros de interés	92
4.3. Multiplicación en Punto Fijo	93
4.3.1. Consumo de recursos lógicos	93
4.3.2. Comportamiento temporal	98
4.3.3. Comparación de los multiplicadores	101
4.4. Multiplicación en Punto Flotante	114
4.4.1. Consumo de recursos lógicos	114
4.4.2. Comportamiento temporal	115
4.4.3. Performance	116
4.5. Conclusiones	119
5. Conclusiones	121
5.0.1. Trabajos futuros	127
Bibliografía	129
A. Dispositivos Lógicos Programables	138
A.1. Introducción	138
A.2. PLD	139
A.3. CPLD	140
A.4. FPGA	142
A.4.1. Celdas Lógicas	144
A.4.2. Recursos dedicados	146

A.5. Conclusiones	150
B. Sistemas numéricos	151
B.1. Introducción	151
B.2. Representación numérica	151
B.2.1. Punto fijo	152
B.2.2. Punto flotante	155
B.2.3. No-convencionales	158
C. Multiplicación Secuencial aplicada al Control de Movimiento	160
C.1. Introducción	160
C.2. Accionamientos paso a paso	162
C.2.1. Generación de perfiles de velocidad Off-line	163
C.2.2. Generación de perfiles de velocidad On-line	163
C.3. Generación de perfiles de velocidad mediante FPGA	168
C.3.1. Implementación del producto $V_r \cdot n_k$	170
C.4. Resultados experimentales	173
C.4.1. Modo de funcionamiento en pasos	173
C.4.2. Modo de funcionamiento en micropasos	175
C.5. Conclusiones	179
D. Publicaciones	180
D.1. Introducción	180
D.2. Multiplicadores secuenciales en FPGA: Evaluación y Comparación	
de Parámetros	183
D.3. Estudio comparativo de multiplicadores secuenciales implementa-	
dos en FPGA	195
D.4. Floating Point Multipliers with Reduced FPGA Area	201

D.5. Performance Evaluation of FPGA Floating Point Multipliers . . .	209
D.6. Novel FPGA based Floating Point Multiplier: Consecutive-Sums Sequential Multiplier	213
D.7. Novel Stepper Motor Controller Based on FPGA Hardware Imple- mentation	222
D.8. FPGA based stepper motor controller	227

Índice de Tablas

2.1. Producto de dos operandos de 4 bits	7
2.2. Implementación de multiplicadores en FPGA.	29
3.1. Producto de dos operandos de 4 bits, acumulación de productos parciales.	34
3.2. Multiplicación secuencial con desplazamiento a la izquierda para operandos de 4 bits.	35
3.3. Multiplicación con desplazamiento a la derecha para operandos de 4 bits.	36
3.4. Estimación de consumo de recursos lógicos de un SM.	40
3.5. Retardos de una FPGA Xilinx Spartan-4.	43
3.6. Producto de dos operandos en base 4	46
3.7. Multiplicación base 4 de dos operandos de 4 bits.	46
3.8. Codificación de los dígitos, alternativa (-X).	47
3.9. Multiplicación base 4, alternativa (-X).	48
3.10. Estimación de consumo de recursos de un SMB4(3X).	49
3.11. Estimación de consumo de recursos lógicos SMB4(-X).	50
3.12. Estimación de consumo de recursos lógicos de un SMSR.	57
3.13. Estimación de consumo de recursos lógicos SMF(SM).	63
3.14. Estimación de consumo de recursos lógicos de un SMF(SMSR).	64

3.15. Ejemplo de una multiplicación mediante sumas consecutivas	68
3.16. Estimación de consumo de recursos lógicos de un SMSC(SM). . .	69
3.17. Estimación de consumo de recursos lógicos de un SMSC(SMSR). . .	70
3.18. Estimación de consumo de recursos lógicos de la variante SMSR con redondeo a cero.	84
3.19. Estimación de consumo de recursos lógicos de la variante SMSR con redondeo a $+\infty$	85
3.20. Estimación de consumo de recursos lógicos de la variante SMSC con redondeo a cero.	88
3.21. Estimación de consumo de recursos lógicos de la variante SMSC con redondeo a $+\infty$	90
4.1. Consumo de recursos de los multiplicadores en Spartan[CLB]. . .	94
4.2. Consumo Estimado de recursos de los multiplicadores [CLB]. . . .	94
4.3. Diferencia porcentual, Consumo Experimental vs Estimado [%]. . .	95
4.4. Consumo de recursos de los multiplicadores en Virtex [slices]. . . .	98
4.5. Consumo de recursos de los multiplicadores en Virtex II [slices]. .	98
4.6. Retardo de propagación de los multiplicadores en Spartan [ns]. . .	99
4.7. Retardo estimado de propagación de los multiplicadores $T_{rd} = 0$ [ns].	99
4.8. Diferencia porcentual, Retardo de propagación Experimental vs Estimado [%].	100
4.9. Velocidad de Procesamiento de los multiplicadores en Spartan [Mops/s].	100
4.10. Velocidad de Procesamiento de los multiplicadores en Virtex [Mops/s].	101
4.11. Velocidad de Procesamiento de los multiplicadores en Virtex II [Mops/s].	101
4.12. Consumo de recursos lógicos de los multiplicadores en punto flo- tante en Spartan [CLBs].	114

4.13. Consumo de recursos lógicos de las variantes en Virtex [slices]. . .	115
4.14. Consumo de recursos lógicos de las variantes en Virtex II [slices]. .	115
4.15. Comportamiento temporal de las variantes en Spartan [Mflop/s]. .	115
4.16. Comportamiento temporal de las variantes en Virtex [Mflop/s]. .	116
4.17. Comportamiento temporal de las variantes en Virtex II [Mflop/s].	116
5.1. Implementación de Multiplicadores Paralelos en FPGA.	122
5.2. Resultados experimentales de Multiplicadores Secuenciales	123
5.3. Resultados experimentales de los Multiplicadores Secuenciales pro-	
puestos.	124
5.4. Resultados experimentales de Multiplicadores en Punto Flotante .	125
B.1. Punto flotante estándar.	157
B.2. Punto flotante a medida.	158

Índice de figuras

2.1. Diagrama en bloques de un multiplicador en punto flotante	9
2.2. Semisumador a) Símbolo lógico, b) y c) arquitectura.	13
2.3. Sumador total, a) símbolo lógico b) y c) arquitecturas.	15
2.4. Símbolo lógico de un CPA e implementación de un RCA.	16
2.5. CSA a) Símbolo lógico y b) implementación con FAs.	17
2.6. Sumadores concatenados a) con CPA, b) con CSA.	18
2.7. Árbol de sumadores para $m=9$	19
2.8. Multiplicador por Tabla de Look-up.	20
2.9. Multiplicador Ripple Carry de 4 bits.	20
2.10. PE de un multiplicador Ripple Carry.	21
2.11. Esquema de un multiplicador Ripple Carry mediante PEs.	21
2.12. Multiplicador Carry Save de 4 bits.	22
2.13. PE de un multiplicador Carry Save.	22
2.14. Esquema de un multiplicador Carry Save mediante PEs.	23
2.15. PE de un multiplicador paralelo Guild.	23
2.16. Multiplicador paralelo Guild.	24
2.17. PE de un multiplicador Mccanny - Mcwhinter.	25
2.18. Multiplicador paralelo Mccanny - Mcwhinter.	26
2.19. Consumo de recursos lógicos de MP.	28
2.20. Multiplicadores proporcionados por Xilinx.	29

2.21. Implementación de Multiplicadores $n = 8$	30
2.22. Implementación de Multiplicadores $n = 16$	30
2.23. Implementación de un multiplicador optimizado en área de Xilinx	
$n = 32$	31
3.1. Multiplicador SM con desplazamiento a la derecha.	37
3.2. Multiplicador SM con desplazamiento a la derecha con registro	
compartido.	38
3.3. Multiplicador SM con desplazamiento a la izquierda.	39
3.4. Consumo de recursos del SM.	41
3.5. Esquema de retardos en un sumador.	42
3.6. Retardo de propagación estimado del SM.	44
3.7. Máxima frecuencia de reloj estimada aplicable al SM.	44
3.8. Velocidad máxima de procesamiento del SM.	45
3.9. Diagrama de un SMB4(3X).	47
3.10. Diagrama de un SMB4(-X).	49
3.11. Diagrama de la variante del SMB4(-X).	50
3.12. Consumo de recursos de multiplicadores SM.	52
3.13. Retardo de propagación estimado del SMB4(3X).	53
3.14. Máxima frecuencia de reloj estimada aplicable al SMB4(3X).	53
3.15. Velocidad máxima de procesamiento del SMB4(3X).	54
3.16. Retardo de propagación estimado del SMB4(-X).	55
3.17. Máxima frecuencia de reloj estimada aplicable al SMB4(-X).	55
3.18. Velocidad máxima de procesamiento del SMB4(-X).	56
3.19. Multiplicador SMSR.	57
3.20. Consumo de recursos del SMSR.	58
3.21. Retardo de propagación estimado del SMSR.	59

3.22. Máxima frecuencia de reloj estimada aplicable al SMSR.	60
3.23. Velocidad máxima de procesamiento del SMSR.	60
3.24. Diagrama de operación de un SMF.	62
3.25. Consumo de recursos de multiplicadores SMF.	65
3.26. Retardo de propagación estimado del SMF.	66
3.27. Máxima frecuencia de reloj estimada aplicable al SMF.	66
3.28. Velocidad máxima de procesamiento de los SMF.	67
3.29. Ejemplo de un SMSC(SM).	68
3.30. Ejemplo de un SMSC(SMSR).	69
3.31. Consumo de recursos estimado de un SMSC.	71
3.32. Ruta crítica de dos sumas consecutivas.	72
3.33. Retardo de propagación estimado del SMSC.	73
3.34. Máxima frecuencia de reloj estimada aplicable al SMSC.	74
3.35. Velocidad máxima de procesamiento del SMSC.	74
3.36. Consumo de recursos lógicos de un SM vs SMSR.	75
3.37. Velocidad de procesamiento de un SM vs SMSR.	76
3.38. Consumo de recursos lógicos de los multiplicadores optimizados en velocidad.	77
3.39. Velocidad de procesamiento de los multiplicadores optimizados en velocidad.	78
3.40. Índice de performance de los multiplicadores.	79
3.41. Índice de performance de los multiplicadores $20 < n < 32$.	80
3.42. Modificación del SMSR para el producto de las mantisas	81
3.43. Ejemplo del esquema de redondeo implementado	83
3.44. Multiplicación de las mantisas.	83
3.45. Consumo de recursos del PFPM(SMSR), $r = 8$.	85
3.46. Consumo de recursos del PFPM(SMSR), $r = 8$.	86

3.47. SMSC modificado para el producto de las mantisas	87
3.48. Consumo de recursos del PFFPM(SMSC), $r = 8$.	89
3.49. Consumo de recursos del PFFPM(SMSC), $r = 8$.	91
4.1. Síntesis de un CLB del multiplicador SM.	95
4.2. Consumo de recursos lógicos del SM en Spartan.	96
4.3. Consumo de recursos lógicos del SM en Spartan.	97
4.4. Consumo de recursos lógicos: SM vs SMSR en Spartan.	102
4.5. Consumo de recursos lógicos: SM vs SMSR en Virtex.	103
4.6. Consumo de recursos lógicos: SM vs SMSR en Virtex II.	103
4.7. Velocidad de procesamiento: SM vs SMSR en Spartan.	104
4.8. Velocidad de procesamiento: SM vs SMSR en Virtex.	104
4.9. Velocidad de procesamiento: SM vs SMSR en Virtex II.	105
4.10. Consumo de recursos lógicos de multiplicadores optimizados en velocidad en Spartan.	106
4.11. Consumo de recursos lógicos de multiplicadores optimizados en velocidad en Virtex.	106
4.12. Consumo de recursos lógicos de multiplicadores optimizados en velocidad en Virtex II.	107
4.13. Velocidad de procesamiento de los multiplicadores optimizados en velocidad en Spartan.	108
4.14. Velocidad de procesamiento de multiplicadores optimizados en ve- locidad en Virtex.	108
4.15. Velocidad de procesamiento de multiplicadores optimizados en ve- locidad en Virtex II.	109
4.16. Índice de performance de los multiplicadores en Spartan, $8 \leq n \leq 20$.	110

4.17. Índice de performance de los multiplicadores en Spartan, $20 \leq n \leq 32$	110
4.18. Índice de performance de los multiplicadores en Virtex, $8 \leq n \leq 20$	111
4.19. Índice de performance de los multiplicadores en Virtex, $20 \leq n \leq 32$	112
4.20. Índice de performance de los multiplicadores en Virtex II, $8 \leq n \leq 20$	113
4.21. Índice de performance de los multiplicadores en Virtex II, $20 \leq n \leq 32$	113
4.22. Índice de performance de los multiplicadores con redondeo a cero en Spartan.	117
4.23. Índice de performance de los multiplicadores con redondeo a $+\infty$ en Spartan.	118
4.24. Índice de performance de los multiplicadores con redondeo a $+\infty$ en Virtex.	118
4.25. Índice de performance de los multiplicadores con redondeo a $+\infty$ en Virtex II.	119
A.1. Diagrama en bloques de un CPLD	141
A.2. Distribución de bloques de una FPGA de Xilinx	142
A.3. Matriz de interconexión de una FPGA de Xilinx	144
A.4. Descripción de un LE de una FPGA FLEX10K de Altera	145
A.5. Descripción de un CLB de una FPGA XC4000 de Xilinx	146
A.6. Recursos lógicos de algunas FPGAs de Xilinx	147
A.7. Lógica de acarreo dedicada de una FPGA XC4000 de Xilinx	149
B.1. Esquema de formatos numéricos	152
C.1. Esquema de motores de una sección del CLIC.	162

C.2. Sistema de control de movimiento de motores de accionamiento incremental.	163
C.3. Diagrama de flujo de un algoritmo on-line.	164
C.4. Perfil de velocidad trapezoidal.	165
C.5. Perfil de velocidad, deseado, cuantizado y la temporización resultante.	167
C.6. (Izquierda) Posición y velocidad con el algoritmo iterativo. (Derecha) Detalle del perfil de posición y velocidad del perfil iterativo .	168
C.7. Arquitectura hardware del algoritmo de generación de perfil de velocidad.	169
C.8. Implementación del multiplicador <i>Ripple Carry</i> en una FPGA XC4006E.	171
C.9. Implementación de un multiplicador optimizado en área de Xilinx en una FPGA XC4006E.	172
C.10. Perfil de velocidad y posición del accionamiento controlado por un sistema basado en FPGA. $15 \frac{\text{rev}}{\text{s}} \equiv 6000 \frac{\text{pasos}}{\text{s}}$	174
C.11. Perfil de velocidad del accionamiento controlado por un sistema basado en FPGA. $62,5 \frac{\text{rev}}{\text{s}} \equiv 25000 \frac{\text{pasos}}{\text{s}}$	175
C.12. Perfil de velocidad del accionamiento utilizando el modo de micropasos.	177
C.13. Perfiles de posición y velocidad para un sistema de accionamiento en modomicropaso.	178

Agradecimientos

A Mario

A Daniel

A mis colegas en el Laboratorio de Instrumentación y Control

Resumen

El objetivo principal de esta tesis fue desarrollar arquitecturas de multiplicadores binarios de menor consumo de recursos lógicos que las existentes, manteniendo las prestaciones de velocidad de cálculo. El propósito particular fue implementar las mismas en los dispositivos digitales denominados FPGA (Field Programmable Gate Array).

Para ello se investigaron las estructuras de multiplicación del tipo secuencial que se caracterizan por poseer un consumo de recursos reducido y una velocidad de procesamiento baja.

Se modelizaron los multiplicadores a partir del desarrollo de ecuaciones que permiten estimar el consumo de recursos lógicos y el desempeño temporal de los mismos. Se desarrollaron nuevas arquitecturas que se compararon con las existentes. Se introdujo un nuevo indicador denominado índice de performance, que permite cuantificar el costo de un multiplicador para una dada FPGA. Los resultados teóricos se corroboraron con un exhaustivo trabajo experimental mediante el cual se han validado las ecuaciones obtenidas.

A partir de la validación de los modelos de los multiplicadores, se determinaron los esquemas de mayor desempeño que cumplen con los objetivos deseados. Así se arribó a multiplicadores en punto fijo con los que se obtuvo un consumo de recursos hasta 8 veces menor y una velocidad comparable a la de una arquitectura existente. Los avances se extendieron a la multiplicación en punto

flotante, obteniéndose esquemas de bajo consumo de recursos y buena velocidad de procesamiento.

Nomenclatura

1C	Complemento a 1
2C	Complemento a 2
ALP	Arreglo Lógico Programable
ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
CPA	Carry Propagate Adder
CPLD	Complex Programmable Logic Device
CSA	Carry Save Adder
DSP	Digital Signal Processor
FA	Full Adder
FF	Flip Flop
FG	Function Generator
FPGA	Field Programmable Gate Array
GAL	Generic Array Logic
HA	Half Adder
LSB	Least Significant Bit
LUT	Look-up Table

MAC	Multiplicador/Acumulador
MP	Multiplicador Paralelo
PAL	Programmable Array Logic
PE	Procesador Elemental
PLD	Programmable Logic Device
PROM	Programmable Read Only Memory
PS	Processing Speed
RCA	Ripple Carry Adder
SM	Multiplicador Secuencial
SMB4	Multiplicador Secuencial de base 4
SMF	Multiplicador Secuencial Fraccionado
SMSC	Multiplicador Secuencial de Sumas Consecutivas
SMSR	Multiplicador Secuencial Sin entradas Registradas
SR	Shift Register
VHDL	Very high speed integrated circuit Hardware Description Language
f_c	Máxima frecuencia de reloj
p_i	Bit i del operando P
P	Operando Producto
T_{BYP}	C_{IN} to C_{OUT} bypass delay
T_c	Tiempo de cálculo
T_{CK}	Período de reloj
T_{CKO}	Clock-to-output delay

T_{ILO}	F/G inputs to X/Y outputs delay
T_{OCPY}	Operand inputs (F1, F2, G1, G4) to C_{OUT} delay
t_{pd}	Propagation delay
T_{rd}	Route delay
T_{SUM}	C_{IN} through function generator to X/Y output delay
X, Y	Operando Multiplicando y multiplicador respectivamente
x_i, y_i	Bit i del operando Multiplicando y Multiplicador respectivamente
V_{min}	Velocidad mínima
V_{max}	Velocidad máxima
$V_r(k)$	Velocidad de referencia para el paso k -esimo
$\Delta t(k)$	Período de tiempo entre el paso actual y el próximo

Capítulo 1

Introducción

En el Procesamiento Digital de Señales (PDS) se recurre, como operación elemental, al producto de dos operandos y la posterior adición de un tercero, estructura conocida como MAC (Multiplicador/Acumulador).

Los dispositivos de cálculo más empleados en el PDS son los DSPs (Digital Signal Processors). Si bien los DSPs son baratos y flexibles, poseen sólo una unidad MAC por lo que un proceso aritmético demanda la ejecución secuencial de las operaciones. Como consecuencia, una operación algorítmica consume un determinado tiempo de ejecución y, si fuera necesario utilizar un tiempo de ejecución menor, convendría adoptar hardware a medida, alternativa proporcionada por los ASICs (Application-Specific Integrated Circuit) y las FPGAs (Field Programmable Gate Array). Tanto los ASICs como las FPGAs posibilitan la utilización de varias MACs en paralelo, logrando de este modo una notable reducción del tiempo de procesamiento. Los ASICs admiten implementar sistemas complejos y resguardan la propiedad intelectual de los procesos debido a que no son copiables. Sin embargo implican un gran costo de fabricación por lo que no aceptan errores en su desarrollo y, consecuentemente, su utilización representa un gran riesgo de producción.

En cuanto a las FPGAs, éstas combinan la flexibilidad de un DSP con la velocidad y la densidad de componentes de un ASIC. Las FPGAs poseen una gran cantidad de recursos lógicos, un bajo costo de desarrollo ya que son fáciles de depurar y, fundamentalmente, permiten al diseñador corregir errores y actualizar el diseño. Estas ventajas convierten a las FPGAs en dispositivos apropiados como procesadores de señales o aceleradores de cálculo.

1.1. Estructuras de Multiplicadores

Los multiplicadores se pueden clasificar según el modo en que ingresan los operandos [1] [2]. Si ambos operandos ingresan en forma serie se denominan serie/serie. La mayor ventaja de estos multiplicadores reside en el bajo número de entradas/salidas utilizadas y en la bajo consumo de recursos lógicos, sin embargo estos multiplicadores se encuentran limitados a aplicaciones de baja velocidad [3] [4] [5]. Si sólo uno de los operandos ingresa en forma serie, el multiplicador se denomina serie/paralelo [6]. Este tipo de multiplicadores presentan un consumo de recursos lógicos moderado y son ideales para aplicaciones de velocidad media donde los multiplicadores serie/serie son demasiado lentos. Si ambos operandos son ingresados en paralelo el tipo de multiplicador es denominado paralelo/paralelo. Estos últimos son los que se tratarán en esta tesis debido a su capacidad de procesar los datos a alta velocidad.

Los multiplicadores también se pueden clasificar en función del algoritmo de cálculo en: suma y desplazamiento, por árbol o contadores. Los multiplicadores por suma y desplazamiento utilizan el método conocido comúnmente como *lápiz y papel* para calcular el producto. En algunos casos esta tarea se realiza construyendo una celda de procesamiento básica que se repite en un arreglo determinado.

La diferencia de procesamiento de estos multiplicadores radica en el tipo de comunicación entre estas celdas, que puede ser local (solo existe comunicación entre celdas vecinas) o global (existe comunicación mas allá de la celda vecina). Con el objetivo de acelerar los productos parciales¹ C. Wallace [7] propuso originalmente una estructura de sumadores, conocida como *Árbol de Wallace*, donde el retardo de la suma es proporcional el logaritmo del número de sumandos. Otro modo de acelerar los productos parciales consiste en la utilización de *contadores* paralelos [8] para obtener la suma de los bits que forman las columnas de la matriz de productos parciales.

Los multiplicadores que realizan el algoritmo de suma y desplazamiento en forma simultánea son denominados habitualmente multiplicadores paralelos. Estos multiplicadores realizan el producto rápidamente, pero son difíciles de realizar cuando un dispositivo posee una cantidad de recursos limitada. Este problema se acentúa, cuando el formato de los operandos esta expresado en punto flotante. Los multiplicadores en punto flotante utilizan típicamente multiplicadores paralelos para el producto de las mantisas [9] [10]. A medida que se incrementa la complejidad de la operación, el consumo de recursos lógicos utilizados aumenta notablemente y consecuentemente los costos de implementación.

Un esquema de multiplicación de operandos en paralelo que utiliza una menor cantidad de recursos lógicos es el denominado *Shift and Add* [11] [12] o secuencial (SM). Esta reducción del consumo de recursos se realiza a costa de ejecutar tantas iteraciones como longitud de palabra posean los operandos [13] [14]. Esto representa una seria desventaja debido al excesivo tiempo de cálculo y por esta razón han sido excluidos de las aplicaciones típicas. Por otro lado, si se puede mejorar el desempeño en velocidad sin incurrir en la cantidad de recursos que requiere un

¹En la multiplicación de dos números enteros de n bits se denomina *producto parcial* al producto un bit del multiplicador por el multiplicando.

multiplicador paralelo, estos multiplicadores pueden resultar más ventajosos.

1.2. Objetivos de la tesis

El objetivo principal de esta tesis consiste en el desarrollo de multiplicadores con un reducido consumo de recursos lógicos, que mantengan un desempeño aceptable de la velocidad de procesamiento.

En función de este objetivo, se caracterizarán las arquitecturas de los multiplicadores existentes. Se propondrán variantes que incluyan la multiplicación de operandos en punto fijo y punto flotante.

1.3. Organización de la tesis

Esta tesis está dividida en cinco capítulos. El Capítulo [1](#) presenta el estado del arte e introduce aspectos de la temática de esta tesis. El Capítulo [2](#) presenta conceptos básicos sobre la multiplicación. Se presentan los esquemas de Multiplicadores Paralelos, los cuales se caracterizan y se ensayan con el objetivo de cuantificar la problemática de consumo de recursos.

El Capítulo [3](#) está dedicado al desarrollo de la propuesta que consiste en la aplicación de Multiplicadores Secuenciales al producto binario en punto fijo y en punto flotante. En este capítulo se desarrollan diferentes esquemas de Multiplicación Secuencial para los cuales se estima su desempeño en cuanto a la cantidad de recursos lógicos y velocidad de procesamiento.

El Capítulo [4](#) está dedicado a la verificación experimental. Para ello se explicita la implementación práctica de los diferentes esquemas desarrollados sobre algunas familias de FPGAs. El objetivo de este capítulo es la contrastación de las estimaciones obtenidas en el capítulo anterior con los resultados experimentales.

En el mismo se establecen criterios de comparación, en términos de performance, entre los multiplicadores propuestos y otros existentes.

Adicionalmente, se presentan cuatro apéndices, a saber:

El Apéndice **A** realiza una breve descripción de los Dispositivos Lógicos Programables y sus características.

El Apéndice **B** introduce los formatos numéricos utilizados en PDS.

El Apéndice **C** describe la aplicación específica de los multiplicadores secuenciales en el Procesamiento Digital utilizado en el área de Control de Movimiento. En el mismo se aprovechan las prestaciones que brindan los multiplicadores en la implementación hardware de un algoritmo de control de posición.

Finalmente, el Apéndice **D** presenta los trabajos publicados como resultado del trabajo de tesis.

Capítulo 5

Conclusiones

Las conclusiones de esta Tesis son las siguientes:

1. Se relevaron las arquitecturas de multiplicadores existentes en la literatura, corroborándose el excesivo consumo de recursos lógicos de los mismos.
2. Se propuso la utilización de la Multiplicación Secuencial para reducir el consumo de recursos. Se modelaron los multiplicadores secuenciales y se concluyó que se debía mejorar su desempeño en velocidad.
3. Se propusieron nuevas arquitecturas de multiplicadores secuenciales con el objetivo de mejorar su desempeño en velocidad. Se modelaron las variantes y se las comparó con los multiplicadores existentes.
4. Se validaron experimentalmente los modelos de los multiplicadores a través de la implementación de los mismos en varias familias de FPGAs. Se obtuvieron mejoras importantes del desempeño de los mismos.
5. Se aplicaron las nuevas arquitecturas de Multiplicadores Secuenciales a la multiplicación en punto flotante. Se obtuvieron multiplicadores de punto flotante de reducido consumo lógico y buenas prestaciones de velocidad.

6. Se aplicó la multiplicación secuencial a un problema concreto de control de movimiento. El sistema desarrollado mostró notables mejoras con respecto al diseño convencional basado en DSPs.

Estas conclusiones se desarrollan con más detalle a continuación.

Arquitecturas existentes

Se analizaron y modelaron los multiplicadores paralelos Ripple Carry, Carry Save y las variantes propuestas por Guild y McCanny-McWhinter. Adicionalmente, se realizaron implementaciones en FPGA del multiplicador Ripple Carry y de otro propuesto por Xilinx optimizado en consumo de recursos. Los resultados de esta implementación obtenidos se resumen en la Tabla 5.1

Tabla 5.1: Implementación de Multiplicadores Paralelos en FPGA.

		Spartan		Virtex		Virtex II	
Bits	Tipo	CLB	Mops	Slices	Mops	Slices	Mops
8	MP	61	20.8	64	41.9	64	45.5
8	MX	52	17.5	36	76.3	36	62.6
16	MP	247	10.8	257	22.0	258	24.7
16	MX	213	11.2	140	59.0	141	47.2
32	MX	816	3.1	544	40.5	548	38.5

MP: Multiplicador Paralelo

MX: Multiplicador propuesto por Xilinx

Se concluyó que, si bien los multiplicadores paralelos pueden ejecutar productos rápidamente, presentan un elevado consumo de recursos que además aumenta cuadráticamente con la longitud de palabra de los operandos. Por ejemplo se pudo comprobar que, para 32 bits, no existen modelos de la serie Spartan, capaces de soportar la cantidad de recursos necesarios. Por lo tanto, la aplicación de estas arquitecturas se limita a FPGAs de gran tamaño.

Multiplicación Secuencial

Se investigó el esquema de la Multiplicación Secuencial, que consume una cantidad inferior de recursos. En particular se modelaron y implementaron experimentalmente tres tipos: SM, SMB4(3X) y SMB4(-X). La Tabla 5.2 resume el consumo de recursos y la velocidad de procesamiento de estas arquitecturas.

Tabla 5.2: Resultados experimentales de Multiplicadores Secuenciales

		Spartan		Virtex		Virtex II	
Bits	Tipo	CLB	Mops	CLB	Mops	CLB	Mops
8	SM	19	10.0	19	23.9	19	30.9
8	SMB4(3X)	24	11.9	24	28.0	24	34.8
8	SMB4(-X)	28	10.5	28	22.9	28	32.1
16	SM	35	4.4	35	12.2	35	14.7
16	SMB4(3X)	45	5.8	45	15.1	45	18.4
16	SMB4(-X)	51	5.2	51	12.5	51	17.9
32	SM	65	1.6	65	5.2	65	6.1
32	SMB4(3X)	86	2.5	86	6.7	86	8.6
32	SMB4(-X)	98	2.3	98	5.9	98	8.5

Se concluyó que, si bien el consumo de recursos lógicos es sustancialmente menor, la velocidad de procesamiento de estas arquitecturas se encuentra por debajo del 50% de la velocidad de los MPs ensayados. Por lo tanto, se concluyó que es necesario mejorar el desempeño en velocidad de los multiplicadores secuenciales.

Arquitecturas propuestas de Multiplicadores Secuenciales

Se realizaron propuestas para la optimización en velocidad de la Multiplicación Secuencial. Para cada uno de los multiplicadores se describió su estrategia y se lo modeló para poder comparar su desempeño. A partir del modelo de cada multiplicador, se expresó cada uno en Lenguaje de Descripción de Hardware (HDL) [28] [29] y se realizó la correspondiente implementación experimental. La Tabla 5.3 resume el consumo de recursos y la velocidad de procesamiento obtenida para estos multiplicadores.

Tabla 5.3: Resultados experimentales de los Multiplicadores Secuenciales propuestos.

		Spartan		Virtex		Virtex II	
Bits	Tipo	CLB	Mops	CLB	Mops	CLB	Mops
8	SMSR	19	11.4	19	27.5	21	36.1
8	SMSC(SM)	24	12.7	30	31.0	30	35.7
8	SMSC(SMSR)	28	16.1	28	38.2	28	45.5
8	SMF(SM)	32	14.6	37	35.5	36	44.3
8	SMF(SMSR)	35	18.1	35	42.1	35	53.8
16	SMSR	35	4.7	34	12.5	38	15.7
16	SMSC(SM)	45	6.1	55	15.9	55	18.0
16	SMSC(SMSR)	51	7.0	50	17.9	51	20.4
16	SMF(SM)	59	7.1	67	19.9	67	23.8
16	SMF(SMSR)	64	8.2	64	21.3	62	27.8
32	SMSR	65	1.7	64	5.4	71	6.5
32	SMSC(SM)	86	2.6	104	7.6	104	8.5
32	SMSC(SMSR)	98	2.7	96	8.1	96	9.1
32	SMF(SM)	111	2.7	127	9.4	128	10.9
32	SMF(SMSR)	122	3.0	122	9.6	122	11.6

Se concluye que, con las variantes propuestas, se pudo incrementar el desempeño en velocidad sin un sacrificio sustancial de recursos lógicos. En particular, con la variante SMF(SMSR) se logró un incremento de velocidad estimado entre un 60% (8 bits) y un 100% (32 bits) con respecto al SM. Con esta variante se estaría alcanzando la velocidad de procesamiento del MP para la serie Spartan con un consumo hasta 8 veces menor.

Nuevas Arquitecturas de Multiplicación Secuencial en Punto Flotante

Los resultados obtenidos con las multiplicadores propuestos se aplicaron a la multiplicación en punto flotante. Se seleccionaron dos de los multiplicadores desarrollados y se realizó el respectivo modelo en punto flotante. Las ecuaciones se validaron a través de la implementación de los mismos en las familias de FPGA antes mencionadas.

La Tabla 5.4 muestra el consumo de recursos lógicos de los multiplicadores SMSR y SMSC aplicados a punto flotante. A su vez, se muestran los ensayos realizados con redondeo a cero y redondeo a $+\infty$ (ver Sección 2.2.2).

Tabla 5.4: Resultados experimentales de Multiplicadores en Punto Flotante

Bits	Tipo	Spartan		Virtex		Virtex II	
		CLB	Mflops	CLB	Mflops	CLB	Mflops
18	SMSR trunc	32	7.6	30	20.3	32	25.6
18	SMSR $+\infty$	35	6.2	32	16.2	35	20.9
18	SMSC trunc	45	10.3	46	27.0	46	29.3
18	SMSC $+\infty$	51	10.2	51	26.9	53	29.4
32	SMSR trunc	51	2.5	52	7.2	53	9.1
32	SMSR $+\infty$	53	2.0	54	6.0	57	7.5
32	SMSC trunc	85	3.8	89	10.5	89	12.1
32	SMSC $+\infty$	100	3.7	100	10.3	102	11.9

Se concluye que, para multiplicadores de 18 bits, se ha obtenido un desempeño hasta 8 veces superior a la de la arquitectura propuesta por Shirazi *et al* [9]. Comparando los resultados con los reportados por Aty *et al* [25], el desempeño obtenido es hasta 6 veces mejor para la serie Spartan, y hasta 5 veces para la serie Virtex II en 18 bits. Para el caso de 32 bits, se concluye que los multiplicadores propuestos obtienen un desempeño al menos 3 veces superior a los reportados por estos autores¹.

Por otra parte, por ejemplo el SMSC propuesto en esta tesis consume 8 veces menos que el propuesto por Jaenicke and Luk [23], con una velocidad tan sólo 2,5 veces menor. Es decir, que el desempeño logrado con los multiplicadores propuestos es al menos 3 veces mayor.

¹Estos autores hacen uso del multiplicador embebido de 18 bits de la serie Virtex II

Multiplicación Secuencial aplicada al Control de Movimiento

El controlador desarrollado permitió el control de posición de máquinas incrementales a alta velocidad en paso completo o en modo micropaso². Se utilizó el multiplicador SMSR para el modo paso y el SMSC para el modo micropaso. Con el sistema desarrollado, se alcanzaron velocidades hasta 5 veces mayores que las obtenidas mediante DSPs en modo paso y hasta 12 veces en modo micropaso.

Publicaciones

- “Multiplicadores secuenciales en FPGA: evaluación y comparación de parámetros”, 8th Argentine Symposium on Computing Technology (36 JAIIO) ISBN 1850-2776. Mar del Plata 2007.
- “Estudio comparativo de multiplicadores secuenciales implementados en FPGA”, XII Reunión de Trabajo en Procesamiento de la Información y Control. Río Gallegos 2007.
- “Floating Point Multipliers with Reduced FPGA Area”, II Southern Conference on Programmable Logic. ISBN 84-609-8998-4. Mar del Plata 2006.
- “Performance evaluation of Floating Point Multipliers”, XX Congreso Argentino de Control Automático. ISBN 978-950-99994-4-2. Buenos Aires 2006.
- “Novel FPGA based Floating Point Multiplier: Consecutive-Sums Sequential Multiplier”, 8th Argentine Symposium on Computing Technology (36 JAIIO) ISBN 1850-2776. Mar del Plata 2007.

²El modo micropaso consiste en el accionamiento del movimiento del motor incremental en fracciones de un paso.

- “Novel stepper motor controller based on FPGA hardware implementation”, IEEE/ASME Transactions on Mechatronics, Nro. 1 Vol 8 ISSN 1083-4435 March 2003, pg 120-124.
- “FPGA based stepper motor controller”, II Southern Conference on Programmable Logic. ISBN 84-609-8998-4. Mar del Plata 2006.

5.0.1. Trabajos futuros

La implementación hardware de algoritmos de control puede resultar de suma utilidad en el campo de la Electrónica de Potencia. Los convertidores de potencia presentan permanentemente un compromiso entre la velocidad del sistema global y la complejidad del control. La evolución de los dispositivos digitales de las últimas décadas permitió el avance en el control de estos sistemas, sin embargo los requerimientos actuales también son mayores.

En el campo de lógica programable, la capacidad de paralelización de tareas permitiría aumentar la eficiencia del control de los convertidores, ya sea aumentando la velocidad de procesamiento del sistema utilizando algoritmos de control y/o modulación más sofisticados que mejoraría la performance del convertidor. Adicionalmente, el sistema implementado en FPGA permitiría la generación de seguridad y protección en la conmutación de las llaves de los convertidores en forma independiente de la ejecución del algoritmo.

En el marco de la investigación realizada hasta el momento, se propone expandir los conocimientos adquiridos para el desarrollo e implementación de los algoritmos de control destinados a Convertidores de Potencia. Entre las herramientas necesarias, se encuentra la necesidad de desarrollar sumadores en punto flotante que posean las mismas características que los multiplicadores propuestos

en el Capítulo 3. Los sumadores en punto flotante consumen considerables cantidades de recursos lógicos, comparables a la de los multiplicadores. Por otro lado, los algoritmos de control de los convertidores de potencia requieren usualmente el cálculo de funciones trigonométricas. Se pretende desarrollar avances en este campo mediante la aplicación de los conocimientos en la ejecución de algoritmos, entre los que se encuentra el de CORDIC [30] [31].

Bibliografía

- [1] E. Boemo, E. Juárez, and J. Meneses, "Taxonomía de multiplicadores," *Proc. VIII DCIS Conference. Univ. of Malaga.*, pp. 185–190, 1993.
- [2] D. Teixeira, A. Susim, and L. Carro, "Comparación de multiplicadores en fpga," *V Workshop Iberchip*, vol. 1, pp. 182–187, Enero 1999.
- [3] O.Nibouche, A. Bouridarie, and M.Nibouche, "New architectures for serial-serial multiplication," *IEEE*, pp. 705–708, 2001.
- [4] A. Aggoun, A. Ashur, and M. K. Ibrahim, "Area-time efficient serial-serial multipliers," in *IEEE International Symposium on Circuits and Systems*, 2000.
- [5] A. Aggoun, A. Farwan, M. Ibrahim, and A. Ashur, "Radix-2n serialserial multipliers," *IEE Proc.-Circuits Devices Syst.*, vol. 151 No. 6, pp. 503–509, 2004.
- [6] A. Bouridane, M.Ñibouche, D. Crookes, and B. Albesh, "A low latency bi-directional serial-parallel multiplier architecture," *IEEE ISCAS 2000*, vol. 5, pp. 593–596, May 2000.
- [7] C. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. on Electronic Comuters*, pp. 14–17, 1964.
- [8] L. Dadda, "Some schemes for parallel multipliers," *Alta Frecuenza*, vol. XX-XIV, N 5, 1965.
- [9] N. Shirazi, A. Walters, and P. Athanas, "Quantitative analysis of floating point arithmetic on FPGA based custom computing machines," *Proceedings. IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 155–162, 1995.
- [10] I. W. B. Ligon, S. McMillan, G. Monn, K. Schoonover, F. Stivers, and K. D. Underwood, "A re-evaluation of the practicality of floating-point operations on FPGAs," *Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 206–215, 1998.

- [11] J. L. Hennessy and D. Patterson, *Computer Architecture: A Quantitative Approach*. San Francisco, California ISBN: 1-55860-329-8: Morgan Kaufmann Publishers, Inc., 1996.
- [12] J. P. Hayes, *Introducción al Diseño Lógico Digital*. Wilmington, Delaware, E.U.A: Addison-Wesley Iberoamericana, 1996.
- [13] G. E. Ordoñez-Fernandez, L. López-López, and J. Velazco-Medina, “Diseño de multiplicadores paralelos de 16 bits en fpgas,” in *Proceedings X Taller Iberchip*, 2004.
- [14] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Design*. New York, Oxford: Oxford University Press, 2000.
- [15] E. Jamro and K. Wiatr, “Constant coefficient convolution implemented in fpgas,” in *Digital System Design, 2002. Proceedings. Euromicro Symposium on*, 4-6 Sept. 2002, pp. 291–298.
- [16] J. B. Evans, C.-J. Chou, and S. Mohanakrishnan, “FPGA implementation of digital filters,” *Proc. ICSPAT’93*, pp. 1–9, 1993.
- [17] R. J. Landry, V. Calmettes, and E. Robin, “High speed IIR filter for xilinx FPGA,” *Proceedings of the 1998 Midwest Symposium on Systems and Circuits*, 1998.
- [18] I. S. 754, *IEEE standard for binary floating-point arithmetic*. New York: The Institute of Electrical and Electronics Engineers Inc., 1985.
- [19] H. Guild, “Fully iterative fast array for binary multiplication and addition,” *Electronic Letters*, vol. 5, p. 263, 1969.
- [20] J. McCanny and J. McWhirter, “Completely iterative, pipelined multiplier array suitable for VLSI,” *IEE Proceedings, Part G - Electronic Circuits and Systems*, vol. 129, pt. G, no. 2, Apr. 1982, p. 40-46., vol. 129, pp. 40–46, Apr. 1982.
- [21] “Core generator,” <http://www.xilinx.com/products/logicore/coregen/index.htm>, 2000.
- [22] L. Louca, T. A. Cook, and W. H. Johnson, “Implementation of IEEE single precision floating point addition and multiplication on FPGAs,” *Proceedings IEEE Symposium on FPGAs for Custom Computing Machines*, pp. 107–116, 1996.
- [23] A. Jaenicke and W. Luk, “Parameterised floating-point arithmetic on fpgas,” in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP ’01). 2001 IEEE International Conference on*, vol. 2, 7-11 May 2001, pp. 897–900vol.2.

- [24] M. A. Jiménez, N. G. Santiago, and D. T. Rover, “Development of a scalable FPGA-based floating point multiplier,” *Proceedings of the Fifth Canadian Workshop on Field-Programmable Devices*, pp. pp. 145 – 150, 1998.
- [25] G. Aty, A. Hussein, I. Ashour, and M. Mones, “High-speed, area-efficient fpga-based floating-point multiplier,” in *Microelectronics, 2003. ICM 2003. Proceedings of the 15th International Conference on*, 9-11 Dec. 2003, pp. 274–277.
- [26] B.Ñew, “Estimating the performance of XC4000E adders and counters,” Xilinx, Tech. Rep. Xapp 018, 1996.
- [27] *The Programable Logic Data Book 1999*. Xilinx, 1999.
- [28] S. Ghosh, *Hardware Description Languages Concepts and Principles*. 445 Hoes Lane, P.O. Box 1331 Piscataway, NJ 08855-1331: IEEE Press, 1999.
- [29] E. Villar, L. Terés, S. Olcoz, and Y. Torroja, *VHDL Lenguaje Estandar de Diseño Electrónico*. Madrid: McGraw Hill, 1997.
- [30] J. E. Volder, “The CORDIC Trigonometric Computing Technique,” *IRE Transactions on Electronic Computers*, vol. EC-8, pp. 330–334, 1959.
- [31] M. Kharrat, M. Loulou, and N. M. and L. Kamoun, “A new method to implement cordic algorithm,” in *The 8th IEEE International Conference on Electronics, Circuits and Systems*, vol. 2, 2001, pp. 715–718.
- [32] [Online]. Available: www.cypress.com
- [33] [Online]. Available: www.actel.com
- [34] [Online]. Available: www.quicklogic.com
- [35] [Online]. Available: www.xilinx.com
- [36] [Online]. Available: www.altera.com
- [37] [Online]. Available: <http://www.alcatel-lucent.com/wps/portal>
- [38] [Online]. Available: www.atmel.com
- [39] R. Bossart, H. Braun, F. Chautard, M. Comunian, J. Delahaye, J.C.Godot, I. Kamber, J. Madsen, L. Rinolfi, S. Schreiber, G. Suberlucq, I. Wilson, and W. Wuensch, “Performances obtained with the cern linear collider test facility,” in *Proc. European Particle Accelerator Conference-EPAC’94*, London, UK, July 1994, pp. 680–682. [Online]. Available: <http://accelconf.web.cern.ch/AccelConf/>

- [40] W. Coosemans and H. Mainaud, “Pre-alignment of clic using the double-wire method,” European Organization of Nuclear Research (CERN), Tech. Rep. 343, July 1997, CLIC-NOTE 343.
- [41] P. Poirier, “L’alignement dynamique submicrometrique de sections accelétrices,” Memoire de soutenance du Diplome de Recherche Specialisee en Sciences, Universite Louis Pasteur de Strasbourg, September 1991.
- [42] D. Carrica, R. Pittin, and M. Benedetti, “Control subsystem hardware of the high performance electronics for alignment regulation on the CLIC 30GHz modules,” *CERN Publication (European Laboratory for Particle Physics), Ginebra, Suiza.*, vol. CTF 99-09, March 1999.
- [43] J. Crook, “Low cost microprocessor based speed control,” in *Proceedings of the Thirteenth Annual Symposium, Incremental Motion Control System and Devices*, B. C. Kuo, Ed., Incremental Motion Control System Society. Champaign, Illinois: Board of Trustees of the University of Illinois, May 1984, pp. 25–30.
- [44] Z. Kang and S. Qu, “A new methodology for using single microprocessor to control DC stepper motors,” in *Proceedings of the IEEE International Conference on Industrial Technology*, I. of Electrical & Electronics Engineers Staff, Ed. Piscataway, NJ, USA: IEEE Press, Dec. 1994, pp. 543–545.
- [45] D. O. Carrica and S. A. González, “Algoritmo eficiente para la generación de perfiles de velocidad en el accionamiento simultáneo de múltiples motores paso a paso,” in *ANALES VIII Reunión de Trabajo en Procesamiento de la Información y Control, RPIC’99*, vol. 1. Mar del Plata, Argentina: Facultad de Ingenieria, U.N.M.d.P., Sept. 23–25, 1999, pp. 118–122.
- [46] P. Acarnley, *Stepping Motors: A Guide to Modern Theory and Practice*, 4th ed., ser. IEE Control Engineering Series, no. 63. London, U.K.: Peter Peregrinus Ltd, Dec. 2002.
- [47] T. Kenjo and A. Sugawara, *Stepping Motors and their Microprocessor Controls*, 2nd ed., ser. Monographs in Electrical and Electronic Engineering. Oxford, U.K.: Oxford University Press, 1995.
- [48] B. Robert, F. Alin, and C. Goedel, “Aperiodic and chaotic dynamics in hybrid step motor-new experimental results,” in *Proceedings of the IEEE International Symposium on Industrial Electronics-ISIE’2001*, 2001, pp. 2136–2141.
- [49] M.-C. Pera, B. Robert, and C. Goedel, “Nonlinear dynamics in electromechanical systems-application to a hybrid stepping motor,” *Electromotion*, vol. 7, no. 1, pp. 31–42, Jan./Mar. 2000.

- [50] N. Acosta, C. Collado, and K. Larsen, "Métricas para la generación de multiplicadores paralelos en FPGA." *Proc. of VII Workshop Iberchip.*, 2001.
- [51] N. Acosta, E. Todorovich, C. Collado, and K. Larsen, "Multiplicadores paralelos: Estado del arte y análisis de su materialización en FPGA." *Proc. of VI Workshop Iberchip.*, pp. 158–168, 2000.
- [52] A. Booth, "A signed binary multiplication technique," *Quartelley J. of Mech. Appl. Math.*, vol. 4 Part 2, 1951.
- [53] S. Brown, "FPGA architectural research: A survey," *Design & Test of Computers, IEEE*, vol. 13, pp. 9–15, 1996.
- [54] S. Brown and J. Rose, "FPGA and CPLD architectures: A tutorial," *Design & Test of Computers, IEEE*, vol. 13, pp. 42–57, 1996.
- [55] D. Carrica, R. Pittin, W. Coosemans, and M. Benedetti, "Active alignment electronic system for CLIC 30 GHz modules in CTF2," *CERN Publication, CLIC Nro. 361*, November 1998.
- [56] T.-S. Chang and C.-W. Jen, "Hardware-efficient implementations for discrete function transforms using LUT-based FPGAs," *Computers and Digital Designs - IEE Proceedings*, vol. 146, Issue 6, p. 309, 1999.
- [57] C. Dick and F. Harris, "FPGA signal processing using sigma-delta modulation," *F. IEEE Signal Processing Magazine*, vol. Volume: 17 Issue: 1, pp. 20–35, Jan. 2000.
- [58] F. Elguibaly, "A fast parallel multiplier-accumulator using the modified booth algorithm," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 47, no. 9, pp. 902–908, September 2000.
- [59] M. Funes, D. Carrica, and M. Benedetti, "Floating point multipliers with reduced fpga area," in *Proceedings II Southern Programmable Logic Conference ISBN 84-609-8998-4*, vol. 1, 2006, pp. 59–66.
- [60] M. Funes, D. Carrica, and M. Benedetti, "Multiplicadores de punto flotante para estructuras FPGA," in *Proceedings XVIII Congreso Argentino de Control Automático*, vol. 1, 2002, p. 40.
- [61] M. Funes, D. Carrica, and M. Benedetti, "Multiplicadores secuenciales para estructuras FPGA," in *IX Reunión de Trabajo en Procesamiento de la Información y Control.*, vol. Vol. II, 2001, pp. 646–651.
- [62] A. A. Gray and S. D. Hoy, "Parallel adaptive equalizer employing subconvolution: VLSI architecture realized in a field programmable gate array," *2003 MAPLD International Conference*, 2003.

-
- [63] R. W. Hix and R. L. Haggard, "Comparative design methodologies for FPGA based computer arithmetic," *Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory*, pp. 374–378, 1997.
- [64] B. Lee and N. Burgess, "Parameterisable floating-point operations on FPGA," *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, pp. 1064–1068, 2002.
- [65] L. Mintzer, "Large FFT's in a single FPGA," *ICSPAT*, 1996.
- [66] K. Rajagopalan and P. Sutton, "A flexible multiplication unit for an FPGA logic block," *The 2001 IEEE International Symposium on Circuits and Systems, 2001. ISCAS 2001*, vol. vol. 4, pp. 546–549, 2001.
- [67] C. Souani, M. Abid, and R. Tourki, "An FPGA implementation of the floating point addition," *Industrial Electronics Society IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, vol. 3, pp. 1644–1648, 1998.
- [68] P. F. Stelling, C. U. Martel, V. G. Oklobdzija, and R. Ravi, "Optimal circuits for parallel multipliers," *IEEE Transactions on Computers*, vol. 47, no. 3, pp. 273–285, March 1998.
- [69] P. F. Stelling and V. G. Oklobdzija, "Optimal designs for multipliers and multiply-accumulators," *Proceedings of the 15th IMACS World Congress on Scientific Computation, Modeling, and Applied Mathematics*, vol. 4, pp. 739–744, August 1997.
- [70] M. Thornton, J. Gaiche, and J. Lemieux, "Tradeoff analysis of integer multiplier circuits implemented in FPGAs," *Communications, Computers and Signal Processing IEEE Pacific RIM Conference*, pp. 301–304, 1999.
- [71] E. Todorovich, G. Sutter, N. Acosta, E. Boemo, and S. López-Buedo, "Relación entre velocidad y consumo en FPGAs," *VII Workshop Iberchip*, pp. 1–6 CD Browser, 2001.
- [72] B. Von Herzen, "Signal processing at 250mhz using high-performanceFPGA's," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, Issue 2, pp. 238–246, 1998.
- [73] *The Programmable Logic Data Book*. Xilinx, 2000.
- [74] R. Zimmermann, *Binary Adder Architectures for Cell-Based VLSI and their Synthesis*. Swiss Federal Institute of Technology, 1997.
- [75] M. Lu, *Arithmetic and Logic in Computer Systems*. Wiley-Interscience, 2004.

- [76] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*. Springer-Verlag, 2001.
- [77] A. Gaffar, W. Luk, P. Cheung, and N. Shirazi, "Customising floating-point designs," in *Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2002, pp. 315 – 317.
- [78] J. Knopman, E. Fernandes, and A. Mesquita, "Implementation of hardware structures through configurable logic," in *EUROMICRO 94. System Architecture and Integration. Proceedings of the 20th EUROMICRO Conference.*, 5-8 Sept. 1994, pp. 125–130.
- [79] M. Louie and M. Ercegovac, "Multiplication and inner product with field programmable gate arrays," in *VLSI Signal Processing, VII, 1994., [Workshop on]*, 26-28 Oct. 1994, pp. 246–255.
- [80] R. Lysecky and F. Vahid, "A configurable logic architecture for dynamic hardware/software partitioning," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 1, 16-20 Feb. 2004, pp. 480–485Vol.1.
- [81] T. Miyamori and U. Olukotun, "A quantitative analysis of reconfigurable coprocessors for multimedia applications," in *FPGAs for Custom Computing Machines, 1998. Proceedings. IEEE Symposium on*, 15-17 April 1998, pp. 2–11.
- [82] T.Ñoll, "Carry-save arithmetic for high-speed digital signal processing," in *Circuits and Systems, 1990., IEEE International Symposium on*, 1-3 May 1990, pp. 982–986vol.2.
- [83] S. Shah, A. Al-Khalili, and D. Al-Khalili, "Comparison of 32-bit multipliers for various performance measures," in *Microelectronics, 2000. ICM 2000. Proceedings of the 12th International Conference on*, 31 Oct.-2 Nov. 2000, pp. 75–80.
- [84] N. Vassiliadis, S.Ñikolaidis, S. Siskos, and D. Soudris, "The effect of the interconnection architecture on the fpga performance and energy consumption," in *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, vol. 1, 12-15 May 2004, pp. 213–216Vol.1.
- [85] J. S. Walther, "A Unified Algorithm for Elementary Functions," in *Proceedings of the Spring Joint Computer Conference*, 1971.
- [86] X. Wang and S. Ziavras, "A configurable multiprocessor and dynamic load balancing for parallel lu factorization," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, 26-30 April 2004, p. 234.

-
- [87] L. Wayne and J. Allan, "Parameterised floating-point arithmetics on FPGAs," in *International Conference on Speech, and Signal Processing, Vol.* IEEE, 2001, pp. 897–900.
- [88] W. W. Yu and S. Xing, "Fixed-point multiplier evaluation and design with fpga," in *Proc. SPIE Vol. 3844, p. 153-161, Reconfigurable Technology: FPGAs for Computing and Applications*, Aug. 1999, pp. 153–161.