

UNIVERSIDAD DE GRANADA
DEPARTAMENTO DE ELECTRÓNICA Y TECNOLOGÍA
DE COMPUTADORES



DESARROLLO DE NUEVAS ESTRATEGIAS DE
SINCRONIZACIÓN PARA VLSI EN SISTEMAS DE
PROCESAMIENTO RNS

TESIS DOCTORAL

DANIEL GONZÁLEZ CASTRO

Editor: Editorial de la Universidad de Granada
Autor: Daniel González Castro
D.L.: GR 755-2013
ISBN: 978-84-9028-438-4

**DESARROLLO DE NUEVAS ESTRATEGIAS DE
SINCRONIZACIÓN PARA VLSI EN SISTEMAS DE
PROCESAMIENTO RNS**

DANIEL GONZÁLEZ CASTRO

TESIS DOCTORAL

UNIVERSIDAD DE GRANADA

**DEPARTAMENTO DE ELECTRÓNICA Y
TECNOLOGÍA DE COMPUTADORES**

Granada, julio de 2012

D. Antonio García Ríos, Profesor Titular de Universidad, y D. Antonio Lloris Ruiz, Catedrático de Universidad y Profesor Colaborador Extraordinario, ambos del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada

CERTIFICAN que el trabajo de investigación que se recoge en la presente Memoria, titulada “Desarrollo de nuevas estrategias de sincronización para VLSI en sistemas de procesamiento RNS” y presentada por D. Daniel González Castro para optar al grado de Doctor por la Universidad de Granada, ha sido realizado en su totalidad bajo su dirección en el Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

Granada a 16 de Julio de 2012

Fdo. Antonio García Ríos

Fdo. Antonio Lloris Ruiz

Directores de la Tesis

AGRADECIMIENTOS

Deseo expresar mi agradecimiento en primer lugar a los profesores Dr. D. Antonio García Ríos y Dr. D. Antonio Lloris Ruiz por su dirección y asesoramiento, gracias a los cuales ha sido posible la realización de la investigación recogida en esta memoria. Asimismo quiero agradecer su amistad, su apoyo y la estrecha colaboración que hemos mantenido durante estos años.

A la profesora Dra. Dña. Encarnación Castillo Morales por su amistad, su ayuda, y los valiosos consejos que me han ayudado a finalizar esta investigación.

Al profesor Dr. D. Luis Parrilla Roure por sus consejos, su asesoramiento y sus orientaciones durante la investigación.

Al resto de los miembros del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada por el ánimo y el apoyo recibido durante todos estos años.

Agradezco especialmente a mi esposa María del Mar, a mi hijo Juan y al resto de mi familia su constante apoyo, su paciencia y su generosa cesión de tiempo sin la cual este trabajo no hubiera sido posible.

Finalmente, destacar que este trabajo ha sido financiado en parte gracias a los proyectos del Plan Nacional TIC2002-02227 y TEC2007-68074-C02-01

A mi familia

ÍNDICE

Índice	i
Índice de figuras	vii
Índice de tablas	xiii
Abreviaturas y siglas	xvii
Prólogo	xxiii
CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Evolución y estado actual de las prestaciones en sistemas digitales	2
1.2 Problemática en la sincronización de circuitos VLSI	5
1.3 Estrategias de sincronización de sistemas VLSI	8
1.4 Nueva estrategia de sincronización	10
1.5 Conclusión	11
CAPÍTULO 2: FUNDAMENTOS DEL SISTEMA NUMÉRICO DE RESIDUOS	13
2.1. Los Sistemas Numéricos de Residuos	14
2.1.1. Representación en RNS de enteros	17
2.1.2. El sistema numérico de residuos simétrico (SRNS)	18
2.1.3. El sistema numérico de residuos cuadrático (QRNS)	18
2.2. Operaciones complicadas de realizar en el RNS	19
2.2.1 Comparación de magnitud y división	19
2.2.2 Detección de desbordamiento	20
2.2.3 Detección de errores	21

2.3. Técnicas de conversión para el RNS	22
2.3.1. Conversión de binario a RNS	22
2.3.2. Conversión de RNS a binario	25
2.4. Módulos aritméticos para el RNS	27
2.4.1. Sumadores en aritmética de residuos	28
2.4.2. Multiplicadores en aritmética de residuos.	31
2.4.2.1. Multiplicadores basados en tablas de consulta	31
2.4.2.2. Multiplicadores de ley cuadrática.....	32
2.4.2.3. Multiplicadores basados en aritmética de índices	33
2.4.2.4. Multiplicadores que no hacen uso de tablas de consulta	35
2.5 Conclusión	38
CAPÍTULO 3: SINCRONIZACIÓN EN CIRCUITOS DIGITALES VLSI.....	41
3.1. Introducción	42
3.1.1. Sistemas síncronos	42
3.1.2. Problemas que afectan a la señal de reloj	44
3.2. Distribución de la señal de reloj.....	47
3.2.1. Dificultades en la distribución de la señal de reloj	48
3.2.1.1. Retardo (<i>delay</i>)	48
3.2.1.2. Pérdida de sincronía (<i>skew</i>)	51
3.2.1.3. Dispersión en el flanco (<i>clock jitter</i>)	56
3.2.1.4. Electromigración	60
3.2.2. Topologías de la red de distribución de la señal de reloj.....	61
3.2.2.1. Distribución no restringida	61
3.2.2.2. Distribuciones simétricas.....	64
3.2.2.3. Distribución en columna central (<i>central spine</i>)	68
3.2.2.4. Mallas de distribución (<i>grid</i>)	71
3.2.2.5. Soluciones híbridas.....	72
3.2.2.6. Otras técnicas de sincronización.....	74
3.2.2.6.1. Sincronización óptica	74
3.2.2.6.2. Sincronización mediante osciladores rotatorios (<i>rotary clocking</i>).....	76
3.2.2.6.3. Supresión controlada del reloj (<i>clock gating</i>).....	79
3.2.2.6.4. Oscilación en baja amplitud (<i>low swing</i>)	80
3.3. Elementos de almacenamiento síncronos.....	81
3.3.1. Sincronización de CSEs controlados por nivel (<i>latches</i>).....	88
3.3.2. Sincronización de circuitos controlados por flanco	92

3.4 Conclusión	94
CAPÍTULO 4: NUEVA ESTRATEGIA DE SINCRONIZACIÓN.....	97
4.1. Introducción	98
4.2. Metodología alternativa de sincronización de sistemas basados en RNS	99
4.2.1. Requisitos para la aplicación de la nueva estrategia de sincronización.....	100
4.2.2. Características de las señales de reloj levemente desfasadas	101
4.2.3. Generación de las señales desfasadas:	
celda parametrizable dCLK_cell	103
4.2.4. Análisis de la celda parametrizable dCLK_cell.....	105
4.2.4.1. Efecto de L_d y W_d	107
4.2.4.2. Efecto de W_b	109
4.2.4.3. Análisis del consumo	111
4.3. Diseño y aplicación de las estrategias de sincronización a tres sistemas.....	112
4.3.1. Diseño de la red de distribución del reloj	114
4.3.2. Filtro CIC para diezmado de Hogenauer	123
4.3.2.1. Descripción del sistema.....	123
4.3.2.2. Implementación para simulación.....	125
4.3.2.2.1. Sincronización del sistema.....	126
4.3.2.2.2. Conversión de binario a RNS.....	130
4.3.2.2.3. Integradores.....	133
4.3.2.2.4. Peines	134
4.3.2.2.5. Reductor de frecuencia.....	135
4.3.2.2.6. Conversión y escalado: módulo ε -CRT	135
4.3.2.3. Resultados de simulación	137
4.3.3. Controlador PID	144
4.3.3.1. Descripción del sistema.....	145
4.3.3.2. Implementación para simulación.....	146
4.3.3.2.1. Sincronización del sistema.....	148
4.3.3.2.2. Conversión de binario a RNS.....	150
4.3.3.2.5. Canales del controlador PID	151
4.3.3.2.6. Conversión y escalado: módulo ε -CRT	151
4.3.3.3. Resultados de simulación	152
4.3.4. Transformada 1D-DWT por transformación al dominio de los índices	158
4.3.4.1. Descripción del sistema.....	158
4.3.4.2. Implementación para simulación.....	161
4.3.4.2.1. Sincronización del sistema.....	163

4.3.4.2.2. Conversión de binario a RNS	165
4.3.4.2.5. Canales del sistema 1D-DWT	166
4.3.4.2.6. Conversión y escalado: módulo ε -CRT	167
4.3.3.3. Resultados de simulación	168
4.3.5. Comparación de resultados	173
4.4 Conclusión	178
CAPÍTULO 5: APLICACIÓN EN DISPOSITIVOS LÓGICOS PROGRAMABLES	181
5.1 Introducción	182
5.1.1. Dispositivos lógicos programables. FPGAs	183
5.1.1.1. Sincronización en FPGAs.....	185
5.1.2. Protección de módulos IP	191
5.2. Aplicación de la nueva estrategia de sincronización a dispositivos lógicos programables.....	192
5.2.1. Descripción del sistema	193
5.2.1.1. Sincronización del sistema	194
5.2.1.2. Canales RNS de procesamiento y conversión más escalado	195
5.2.1.3. Lógica de extracción de la firma	196
5.2.2. Implementación del sistema.....	198
5.2.2.1. Generación de las señales de reloj.....	200
5.3. Resultados experimentales	203
5.3.1. Efecto de la nueva estrategia de sincronización sobre el consumo.....	206
5.3.2. Efecto de la nueva estrategia de sincronización sobre el proceso de extracción de la firma	211
5.4 Conclusión	214
CAPÍTULO 6: CONCLUSIONES.....	217
6.1 Introducción	218
6.2 Principales aportaciones.....	220
6.3 Líneas de investigación futuras.....	224
6.4 Conclusión	225
APÉNDICE A: GENERACIÓN DE LA SEÑAL DE RELOJ	227
A.1. Generación de la señal de reloj	228
APÉNDICE B: ANÁLISIS DE LOS DISPOSITIVOS Y MÓDULOS USADOS EN LAS SIMULACIONES	237

B.1. Transistores	238
B.2. Elementos de memoria.....	240
B.2.1. Biestables	240
B.2.2. Registros	241
B.2.3. Memorias ROM	241
B.3 Sumadores	246
A.3.1 Sumador binario de n bits	246
A.3.2 Sumador módulo m	247
B.4 Otros módulos	248
B.4.1. Multiplexores	249
B.4.2. Puertas lógicas	249
B.4.3. Contador de 32 bits	250
B.5 Generador de la señal de reloj externa	250
APÉNDICE C: PROGRAMA GENERADOR DE MEMORIAS	253
C.1. Organigrama.....	254
C.2. Programa generador de librerías	255
Bibliografía	259

ÍNDICE DE FIGURAS

CAPÍTULO 1: INTRODUCCIÓN	1
Figura 1.1. Evolución del número de dispositivos integrados en un solo microprocesador.....	2
Figura 1.2. Evolución del tamaño característico de los transistores.....	3
Figura 1.3. Evolución de la frecuencia de funcionamiento diferentes.....	4
CAPÍTULO 2: SISTEMA NUMÉRICO DE RESIDUOS	13
Figura 2.1. Conversión por bloques de binario a RNS.	24
Figura 2.2. Sumadores basados en memorias ROM.....	30
Figura 2.3. Sumadores modulares basados en sumadores binarios.	31
Figura 2.4. Multiplicador módulo m basado en la ley cuadrática.....	33
Figura 2.5. Multiplicador basado en aritmética de índices.	34
Figura 2.6. Multiplicador módulo m que no hace uso de tablas.	37
CAPÍTULO 3: SINCRONIZACIÓN EN CIRCUITOS DIGITALES VLSI	41
Figura 3.1. Esquema de funcionamiento de un sistema síncrono.....	42
Figura 3.2. <i>Skew</i> negativo (a) y positivo (b).	52
Figura 3.3. Problema de carreras o <i>double clocking</i> debido a un valor del <i>skew</i> negativo elevado.....	53
Figura 3.4. Definición de <i>jitter</i>	56
Figura 3.5. Filtro RC para corrección del <i>jitter</i> [KUR01].	59
Figura 3.6. Red de distribución no restringida.....	62
Figura 3.7 Distribución no simétrica [RES98].	63
Figura 3.8. Procesador Itanium® 2 y red de distribución del reloj de la caché de 3 ^{er} nivel [CHA05].....	64

Figura 3.9. Red de distribución en <i>H-tree</i>	65
Figura 3.10. Distribuciones en <i>X-tree</i> y <i>H-tree</i> con acoplamiento de impedancia.	65
Figura 3.11. Redes de distribución en árbol binario (a) y árbol binario con conexiones intermedias (b).....	66
Figura 3.12. Reducción del <i>skew</i> mediante interconexiones.....	67
Figura 3.13. Distribución de reloj del PowerPC 64b [HOF00].....	68
Figura 3.14. Distribución en columna central (a) y con interconexiones equilibradas (b).....	69
Figura 3.15. Distribución mediante columnas centrales en modelos de procesador Alpha® [GRO98].....	69
Figura 3.16. Columnas de distribución de reloj en Pentium® 4 [KUR01].....	70
Figura 3.17. Rejilla bidimensional (a) y unidimensional (b).	72
Figura 3.18. Distribución del reloj y retardos a través del chip en el procesador Power4 [RES02].	73
Figura 3.19. Regiones de reloj y distribución en el procesador Xeon® [TAM06].....	74
Figura 3.20. Red de distribución global óptica [TOS07].	75
Figura 3.21. RTWO (a) y matriz de 25 RTWOs (b).	77
Figura 3.22. Registro con habilitación de carga síncrona (a). Aplicación de <i>clock gating</i> (b).....	79
Figura 3.23. Retardos principales en CSEs (a). Dependencia de t_{CQ} con t_{DC} y t_{CD} (b).....	82
Figura 3.24. Relación entre t_{CQ} y t_{DQ} en función de t_{su}	83
Figura 3.25. Definiciones de t_{su} y t_{hold}	85
Figura 3.26. Diferentes estrategias de sincronización mediante <i>latches</i> : a) lógica C ² CMOS con NPTC; b) NORA CMOS dinámica; c) TSPC CMOS dinámica.....	89
Figura 3.27. <i>Latch</i> tipo <i>master-slave</i> [WAR06].....	90
Figura 3.28. Diagrama de captura de datos en <i>latch</i> : dato anticipado (D_1) y dato retardado (D_2).	91
Figura 3.29. Cuatro ejemplos de <i>flip-flop</i> típicos sincronizados mediante TSPC: a) <i>flip-flop</i> basado en biestable tipo D; b) SAFF; c) HLFF; d) Sdff	93
Figura 3.30. Cronograma de <i>flip-flop</i>	94

CAPÍTULO 4: NUEVA ESTRATEGIA DE SINCRONIZACIÓN	97
Figura 4.1. Esquema de línea de generación de las señales de reloj desfasadas:	
(1) inversor de entrada, con una gran capacidad de entrada;	
(2) inversor de salida, de dimensiones mínimas;	
(3) inversor restaurador, genera la señal de reloj efectiva.	103
Figura 4.2. Detalle de la celda parametrizable dCLK_cell.....	104
Figura 4.3. Representación de la señal de reloj de	
entrada original junto con las señales levemente desfasadas.	106
Figura 4.4. Retardo por celda en función de L_d y W_d	108
Figura 4.5. Retardo en función de L_d , W_d y W_b	110
Figura 4.6. Consumo de la celda parametrizable en función de L_d , W_d y W_b	111
Figura 4.7. Sistema de prueba para apreciar el efecto del perfil de la	
señal de reloj sobre el consumo	115
Figura 4.8. <i>Buffer</i> regenerador de señal de reloj.....	116
Figura 4.9. Consumo global de un sistema en función	
de la anchura del canal del <i>buffer</i>	117
Figura 4.10. Consumo de un sistema, sin incluir la red de distribución de la señal	
de reloj, en función del tamaño del <i>buffer</i> que lo sincroniza.	119
Figura 4.11. Tiempos de subida y bajada en función del tamaño del <i>buffer</i>	120
Figura 4.12. Biestable sincronizado por flanco de bajada.	122
Figura 4.13. Filtro CIC de tres etapas.....	124
Figura 4.14. Esquema de un integrador (a) y un peine ($D=1$) (b) para el procesamiento	
de señales discretas basados en sumadores modulares.	124
Figura 4.15. Esquema del filtro CIC sincronizado mediante un	
único reloj (a) y celdas dCLK_cell (b).....	125
Figura 4.16. Reductores de frecuencia para las dos estrategias de sincronización.....	126
Figura 4.17. Cronograma con la señal que sincroniza el reductor de	
frecuencia y las cinco señales generadas por éste.....	127
Figura 4.18. Señales de reloj desfasadas generadas mediante	
las celdas dCLK_cell.....	129
Figura 4.19. Módulo de conversión de binario a RNS.	131
Figura 4.20. Sumador modular con cauce segmentado y salida registrada.	132
Figura 4.21. Implementación de integradores de 6 y 8 bits (a y b,	
respectivamente) y peines para los canales de 6 y 8 bits (c y d).	134

Figura 4.22. Módulo de conversión más escalado ε -CRT para el filtro CIC.....	136
Figura 4.23. Señales de reloj y corriente instantánea con reloj único (a) y reloj distribuido (b).....	138
Figura 4.24. Incremento del consumo en sistemas sincronizados por celdas dCLK_cell.	142
Figura 4.25. Implementación discreta de un controlador PID.	146
Figura 4.26. Implementación para simulación de un controlador PID basado en RNS de cinco canales mediante las dos metodologías de sincronización.	147
Figura 4.27. Módulo de conversión de binario a RNS del controlador PID.....	150
Figura 4.28. Sumador modular combinacional.	150
Figura 4.29. Canal PID implementado mediante tablas de consulta.....	151
Figura 4.30. Módulo de conversión más escalado ε -CRT para el controlador PID.....	152
Figura 4.31. Comparación de los valores máximos de las demandas de corriente.....	153
Figura 4.32. Comparación de corrientes con reloj distribuido a 40MHz (a) y 125MHz (b).	154
Figura 4.33. Arquitectura para la realización de la transformada 1D-DWT por transformación al dominio de los índices.	160
Figura 4.34. Implementación de un filtro basado en RNS para la transformada 1D-DWT <i>wavelets</i> mediante las dos estrategias de sincronización.	162
Figura 4.35. Divisor de la señal de reloj en sus fases y señales de reloj más relevantes.....	163
Figura 4.36. Módulo de conversión y escalado ε -CRT en el sistema 1D-DWT <i>wavelets</i>	166
Figura 4.37. Comparación de las corrientes máximas en el sistema 1D-DWT a 40MHz y 125MHz.	168
Figura 4.38. Variación porcentual del incremento de consumo al aplicar varias señales desfasadas.	171
Figura 4.39. Variación porcentual del consumo dinámico en el sistema 1D-DWT.....	172
Figura 4.40. Reducción de los picos de corriente en los sistemas estudiados.....	173
Figura 4.41. Incremento del consumo debido a la inclusión de la cadena de celdas dCLK_cell.	175

Figura 4.42. Consumo dinámico de los sistemas bajo las dos metodologías de sincronización.	178
---	-----

CAPÍTULO 5: APLICACIÓN EN DISPOSITIVOS

LÓGICOS PROGRAMABLES 181

Figura 5.1. Arquitectura general de una FPGA y detalle de un Bloque Lógico.	184
--	-----

Figura 5.2. Arquitectura de la distribución global de la señal de reloj (a) y detalle de la red de distribución de una región de reloj (b) en la serie 7 de Xilinx [XIL12a].	186
---	-----

Figura 5.3. Distribuciones de reloj global (a) y regional (b) en la FPGA Stratix V [XIL12].	187
--	-----

Figura 5.4. Estructura de alto nivel de la Virtex-5 XC5VLX30 [XIL12b].	188
---	-----

Figura 5.5. Estructura de un CLB (a) e interconexiones entre CLBs y <i>slices</i> (b) [XIL12b].	189
--	-----

Figura 5.6. Esquema general de un CTM en la FPGA Virtex-5 [XIL12b].	190
--	-----

Figura 5.7. Sistema 1D-DWT sincronizado mediante una señal de reloj única (a) y mediante DCMs (b).	194
---	-----

Figura 5.8. Lógica de extracción de la firma mediante un LFSR.	197
---	-----

Figura 5.9. Placa de prototipado Virtex-5 LX [XIL11].	198
--	-----

CAPÍTULO 6: CONCLUSIONES 217

APÉNDICE A: GENERACIÓN DE LA SEÑAL DE RELOJ 227

Figura A.1. Circuito oscilador de cristal.	228
---	-----

Figura A.2. Esquema general de un PLL.	229
---	-----

Figura A.3. Esquema general de un DLL.	230
---	-----

Figura A.4. Diagrama de bloques de un sistema ADPLL <i>current-starved ring oscillator</i>	232
---	-----

Figura A.5. PLL como multiplicador de frecuencia.	233
--	-----

Figura A.6. Oscilador basado en anillo (a) y en oscilador LC (b).	234
--	-----

Figura A.7 Inductores espirales de dos vueltas: a) cuadrado, b) octogonal y c) octogonal simétrico.	235
--	-----

APÉNDICE B: ANÁLISIS DE LOS DISPOSITIVOS Y	
MÓDULOS USADOS EN LAS SIMULACIONES.....	237
Figura B.1. Esquemático de un biestable tipo D activado por flanco de bajada.....	240
Figura B.2. Implementación de un registro de n bits.	241
Figura B.3. Visión de global de un módulo de memoria de 256×4	242
Figura B.4. Celda de selección de filas dec_64.....	243
Figura B.5. Celda de almacenamiento de datos cell_2x8.	244
Figura B.6. Esquemático de la celda sel2_1.....	244
Figura B.7. Esquemático de la celda amp_inv.	245
Figura B.8. Sumador completo de un bit.	246
Figura B.9. Sumadores modulares simple (a) y con <i>pipelinig</i> (b).....	247
Figura B.10. Celda básica del multiplexor.....	248
Figura B.11. Contador de 32 bits.	249
Figura B.12. Señal de reloj de entrada.	250
APÉNDICE C: PROGRAMA GENERADOR DE MEMORIAS	253
Figura C.1. Ejemplo de archivo de descripción de la memoria.	254
Figura C.2. Organigrama para la generación de la librería.	256

ÍNDICE DE TABLAS

CAPÍTULO 1: INTRODUCCIÓN	1
CAPÍTULO 2: SISTEMA NUMÉRICO DE RESIDUOS	13
CAPÍTULO 3: SINCRONIZACIÓN EN CIRCUITOS DIGITALES VLSI	41
CAPÍTULO 4: NUEVA ESTRATEGIA DE SINCRONIZACIÓN	97
Tabla 4.1. Consumo del sistema y de la red de distribución del reloj y perfil de la señal de reloj en función del tamaño de <i>buffer</i> regenerador de la señal de reloj.....	118
Tabla 4.2. Área total de puerta de los transistores sincronizados por la señal de reloj de un biestable estándar.....	122
Tabla 4.3. Dimensionado de salida de los <i>buffers</i> y celdas <i>dclk_cell</i> en filtro CIC.	128
Tabla 4.4. Retardo entre señales de reloj adyacentes y valor de los parámetros L_d y W_d	130
Tabla 4.5. Comparación de corrientes máximas con reloj único y distribuido en el filtro CIC.	139
Tabla 4.6. Comparación de la máxima derivada de la corriente en el filtro CIC.	140
Tabla 4.7. Comparación de consumos con ambas estrategias de sincronización.	141
Tabla 4.8. Consumo del filtro CIC sin actividad a la entrada de datos.	143
Tabla 4.9. Comparación del consumo dinámico en el filtro CIC.	144
Tabla 4.10. Dimensionado del <i>buffer</i> y de las celdas <i>dCLK_cell</i>	147
Tabla 4.11. Dimensionado de las celdas <i>dCLK_cell</i> para alcanzar los retardos requeridos.	148
Tabla 4.12. Corrientes máximas con reloj único y distribuido en el controlador PID.	153

Tabla 4.13. Comparación de la máxima derivada de la corriente en el controlador PID.	155
Tabla 4.14. Comparación de consumos en el controlador PID	156
Tabla 4.15. Consumo del controlador PID sin actividad en la entrada de datos.	157
Tabla 4.16. Comparación del consumo dinámico en el controlador PID.....	158
Tabla 4.17. Dimensionado del <i>buffer</i> y de las celdas dCLK_cell.	163
Tabla 4.18. Dimensionado de las celdas dCLK_cell para alcanzar los retardos requeridos.....	164
Tabla 4.19. Dimensiones de los buffers que generan las señales internas de los filtros 1D-DWT.	165
Tabla 4.20. Corrientes máximas con reloj único y distribuido en el sistema 1D-DWT.	168
Tabla 4.21. Comparación de la máxima derivada de la corriente en el sistema 1D-DWT.	169
Tabla 4.22. Comparación de consumos en el sistema 1D-DWT.....	170
Tabla 4.23. Consumo del filtro 1D-DWT sin actividad en la entrada de datos.....	171
Tabla 4.24. Comparación del consumo dinámico en el sistema 1D-DWT.	172
Tabla 4.25. Descenso porcentual de la derivada de la corriente al aplicar la nueva estrategia de sincronización	174

CAPÍTULO 5: APLICACIÓN EN DISPOSITIVOS LÓGICOS

PROGRAMABLES	181
Tabla 5.1. Recursos empleados y frecuencia máxima de funcionamiento.....	200
Tabla 5.2. Desfase generado por cada DCM en función de la frecuencia.....	202
Tabla 5.3. Consumo del sistema con SEF manual.	203
Tabla 5.4. Consumo del sistema con SEF generada mediante LFSR.	204
Tabla 5.5. Consumo estático del sistema.....	205
Tabla 5.6. Influencia sobre el consumo de la inclusión de los DCMs y de la aplicación de la nueva estrategia de sincronización.	206
Tabla 5.7. Consumo dinámico del sistema con SEF manual.....	208
Tabla 5.8. Consumo dinámico del sistema con SEF generada mediante LFSR.....	209
Tabla 5.9. Reducción porcentual del consumo debida a los DCMs.....	210
Tabla 5.10. Variaciones en el consumo entre los modos de funcionamiento normal y de extracción de la firma.	211

Tabla 5.11. Variaciones en el consumo dinámico entre los modos de funcionamiento normal y de extracción de la firma.....	212
CAPÍTULO 6: CONCLUSIONES	213
APÉNDICE A: GENERACIÓN DE LA SEÑAL DE RELOJ	227
APÉNDICE B: ANÁLISIS DE LOS DISPOSITIVOS Y MÓDULOS USADOS EN LAS SIMULACIONES	237
Tabla B.1. Dimensiones de los transistores NMOS y PMOS.....	239
Tabla B.2. Especificaciones para las dos señales de reloj ideales.....	250
APÉNDICE C: PROGRAMA GENERADOR DE MEMORIAS	253

ABREVIATURAS Y SIGLAS

A continuación se recogen las abreviaturas y siglas que aparecen a lo largo de esta tesis para facilitar su lectura.

A/D Analógico/digital.

ADPLL *All Digital Phase-Locked Loop*, ciclo de fijación de fase completamente digital.

ASIC *Application Specific Integrated Circuit*, Circuito integrado para aplicaciones específicas.

CIC *Cascade Integrator Comb*, integradores y peines en cascada.

CLB *Configurable Logic Block*, bloque lógico configurable.

CMC Canadian Microelectronic Corporation, Corporación Canadiense para la Microelectrónica.

CMOS *Complementary Metal-Oxide-Semiconductor*, proceso metal-óxido-semiconductor complementario.

CPA *Carry Propagate Adder*, sumador con propagación de acarreo.

- CPLD** *Complex Programmable Logic Device*, dispositivo lógico programable complejo.
- CRT** *Chinese Remainder Theorem*, teorema del resto chino.
- CSA** *Carry Save Adder*, sumador con ahorro de propagación de acarreo.
- CSE** *Clocked storage element*, elemento almacenador síncrono.
- C²MOS** *Clocked CMOS Logic*, lógica de sincronización con dispositivos CMOS.
- DCM** *Digital Clock Manager*, administrador digital de la señal de reloj.
- DCO** *Digital Controlled Oscillator*, oscilador controlado digitalmente.
- DLL** *Delay-Locked Loop*, ciclo de fijación de fase por retardo.
- DOE** *Diffraction Optical Element*, elemento óptico difractor.
- DSP** *Digital Signal Processing*, procesamiento digital de señales (también Digital Signal Processor, procesador digital de señales).
- DWT** *Discrete Wavelets Transform*, Transformada Discreta de Wavelets.
- EPLD** *EPLD technology-based Programmable Logic Device*, Dispositivos Lógicos Programables basados en tecnología EPROM.
- EPROM** *Erasable Programmable Read Only Memory*, Memoria de Solo Lectura Borrable y Programable.
- EEPROM** *Electrically Erasable Programmable Read Only Memory*, Memoria de Solo Lectura Borrable y Programable Eléctricamente.

- FF** *Flip-flop*, biestable.
- FFT** *Fast Fourier Transform*, transformada rápida de Fourier.
- FIR** *Finite Impulsive Response*, respuesta a impulso finita.
- FPGA** *Field Programmable Gate Array*, matriz de puertas programables.
- FPL** *Field Programmable Logic*, Lógica programable.
- FSM** Finite-State Machine, máquina de estados finitos.
- GALS** *Globally Asynchronous Locally Synchronous*, globalmente asíncrono localmente síncrono.
- GQRNS** *Galois Enhanced QRNS*, QRNS mejorado mediante aritmética sobre cuerpos de Galois.
- HDL** *Hardware Description Language*, lenguaje de descripción de *hardware*.
- IP** *Intellectual Property*, propiedad intelectual.
- LFSR** *Linear Feedback Shift Register*, registro de desplazamiento con retroalimentación lineal.
- LSI** *Large Scale of Integration*, integración a gran escala.
- LUT** *Look-up table*, tabla de consulta.
- MOS** *Metal-Oxide-Semiconductor*, metal-óxido-semiconductor.
- MOSFET** *Metal-Oxide-Semiconductor Field Effect Transistor*, Transistor de Efecto Campo Metal Óxido Semiconductor.

- MRC** *Mixed Radix Conversion*, conversión por bases mixtas.
- MSI** *Medium Scale of Integration*, integración a media escala.
- MSL** *Master-slave latch*, elemento de almacenamiento síncrono controlado por nivel tipo maestro-esclavo.
- MCD** Máximo Común Divisor.
- N-C²MOS** *N-devices Clocked CMOS Logic*, lógica de sincronización CMOS con dispositivos de canal N.
- nETDF** *Negative Edge-Triggered D Flip-Flop*, biestable controlado por flanco de bajada.
- NMOS** *N-channel Complementary Metal-Oxide-Semiconductor*, proceso metal-óxido-semiconductor de canal N.
- NORA** *No Races*, sin carreras.
- NPTC** *Non-Overlapping Pseudo Two Phase Clock*, reloj de pseudo-dos fases no solapadas.
- PLL** *Phase-Locked Loop*, ciclo de fijación de fase.
- POD** *Point of Divergence*, punto de divergencia.
- QRNS** *Quadratic Residue Number System*, sistema numérico de residuos cuadrático.
- P-C²MOS** *P-devices Clocked CMOS Logic*, lógica de sincronización CMOS con dispositivos de canal P.
- PID** Proporcional, integral, diferencial.

PLD	<i>Programmable Logic Device</i> , dispositivo lógico programable.
PMOS	<i>P-channel Complementary Metal-Oxide-Semiconductor</i> , proceso metal-óxido-semiconductor de canal P.
RC	<i>Resistive-Capacitive</i> , resistivo-capacitivo.
RDR	Red de Distribución del Reloj.
RNS	<i>Residue Number System</i> , sistema numérico de residuos.
ROA	<i>Rotary Oscillatos Array</i> , matriz de osciladores rotatorios.
ROM	<i>Read Only Memory</i> , memoria de sólo lectura.
RTWO	<i>Rotary traveling-wave oscillator</i> , oscilador rotatorio de ondas viajeras.
SPDM	<i>Single poly double metal</i> , una capa de polisilicio y dos capas de metal.
SPLD	<i>Simple Programmable Logic Device</i> , Dispositivo lógico programable simple.
SRAM	<i>Static Random Acces Memory</i> , Memoria de Acceso Aleatorio Estática.
SSI	<i>Small Scale of Integration</i> , integración a pequeña escala.
SoC	<i>System on chip</i> , sistemas integrados en el mismo circuito integrado.
SOI	<i>Silicon On Insulator</i> , silicio sobre aislante.
SRNS	<i>Symmetric Residue Number System</i> , sistema numérico de residuos simétrico.
TSPC	<i>True Single Phase Clock</i> , reloj de fase única.

- VCO** *Voltage Controlled Oscillator*, oscilador controlado por tensión.
- VHDL** *Very High Speed Integrated Circuit Hardware Description Language*, lenguaje de descripción de *hardware* para circuitos integrados de alta velocidad.
- VLSI** *Very Large Scale of Integration*, muy alta escala de integración.

PRÓLOGO

La sincronización es uno de los aspectos más importantes a considerar en el diseño de sistemas electrónicos VLSI (*Very Large Scale of Integration*), ya que las prestaciones y el rendimiento de un sistema dependen considerablemente de la exactitud y la calidad de la señal de reloj. Los últimos avances en la fabricación de circuitos integrados han tenido como consecuencia directa un notable incremento de los niveles de integración y de las velocidades de funcionamiento de los sistemas. Estos dos factores dificultan de forma extraordinaria la sincronización de los circuitos integrados. En los sistemas síncronos la señal de reloj controla el flujo de información a lo largo de todo el circuito, por lo tanto, la calidad de la misma repercute de forma crítica en el rendimiento del sistema.

La técnica de sincronización más frecuente en los sistemas digitales actuales es la denominada TSPC (*True Single Phase Clock*) [KAR88, AFG90], que consiste en transmitir una única señal de reloj a lo largo de todo el sistema de modo que todos los elementos síncronos sean activados simultáneamente por el flanco de dicha señal. Esto significa que una señal eléctrica debe ser transmitida de manera simultánea y con calidad aceptable por todo el circuito hasta cada módulo operativo. La señal de reloj es básicamente una señal de control y, como tal, es la que más distancia debe recorrer a lo largo del circuito y la que más rápidamente debe conmutar; de ahí que su generación y distribución sean un aspecto prioritario en el diseño de sistemas digitales.

Los problemas más comunes en la generación y distribución de la señal de reloj son el ruido, los retardos y la atenuación. Dichos problemas afectan a la calidad de la señal y al instante en que ésta conmuta, por lo que pueden causar una pérdida de sincronía en el sistema, o restricciones en la máxima frecuencia de reloj alcanzable debido a la necesidad de establecer márgenes de funcionamiento seguros. Por otro lado, el funcionamiento perfectamente síncrono de sistemas con un elevado número de dispositivos puede provocar problemas de distribución de corriente. La conmutación simultánea de gran cantidad de dispositivos requiere elevados niveles de corriente, de manera que es necesario el aumento del tamaño de las líneas de alimentación y el diseño de fuentes de alimentación capaces de suministrar rápidamente elevados niveles de corriente al sistema manteniendo constantes los niveles de referencia de tensión. El aumento del tamaño de las líneas de distribución de corriente provoca un incremento de las capacidades asociadas y, por lo tanto, del retardo producido y del consumo, mientras que el diseño de fuentes de alimentación más complejas complican y encarecen el diseño del sistema.

El principal objetivo de esta investigación es el estudio de una estrategia de sincronización para sistemas basados en el sistema numérico de residuos (RNS: *Residue Number System*) que distribuya varias señales de reloj ligeramente desfasadas a lo largo del periodo de la señal de reloj de referencia. Cada una de estas señales controlará cada uno de los canales de procesamiento de datos en los que se descomponen de forma natural los sistemas basados en el RNS. De esta manera se distribuye el funcionamiento del sistema, de modo que los valores máximos de corriente y el ritmo de crecimiento de dichas demandas disminuyan significativamente, lo que permitirá el escalado de las líneas de alimentación del circuito integrado y la simplificación de las fuentes de alimentación. Al mismo tiempo, se eliminará la necesidad de transmitir una única señal de reloj perfectamente síncrona a lo largo de todo el sistema, con lo que se simplificará significativamente la labor de diseño de la red de distribución del reloj. Para llevar a cabo esta investigación se han realizado simulaciones eléctricas sobre sistemas realistas y también se han implementado sobre dispositivos lógicos programables.

La memoria este trabajo está estructurada en seis capítulos y tres apéndices, tal como se describe a continuación:

CAPÍTULO 1. INTRODUCCIÓN

En el primer capítulo se introduce el problema de la sincronización de sistemas VLSI y se exponen las principales motivaciones para el desarrollo de este trabajo. Más concretamente se realiza una revisión histórica de la evolución del tamaño de los circuitos integrados, de la reducción del tamaño de los dispositivos y del aumento de la frecuencia de los sistemas digitales. Además, se plantean una serie de problemas derivados del aumento de la frecuencia y de las dimensiones de los sistemas digitales actuales. Para finalizar se plantea una estrategia de sincronización para sistemas basados en el RNS que aprovecha la modularidad de dichos sistemas.

CAPÍTULO 2. FUNDAMENTOS DEL SISTEMA NÚMÉRICO DE RESIDUOS

En el segundo capítulo se realiza una revisión del RNS. En este capítulo se introducen los Sistemas Numéricos de Residuos y las operaciones más comunes en la aritmética de residuos. Además, se presentan algunas otras características del RNS, tales como la posibilidad de detección y corrección de errores, nociones sobre conversión, etc. También se describe la principal ventaja del RNS consistente en que es un sistema de representación numérica en el que cada número es representado por un conjunto de módulos de tal manera que las operaciones de suma, diferencia y producto de varios números se realizan de modo independiente en cada canal de procesamiento.

CAPÍTULO 3. SINCRONIZACIÓN EN CIRCUITOS DIGITALES VLSI

En el tercer capítulo se hace una revisión de diferentes métodos para la sincronización de circuitos digitales VLSI. Se analiza la propagación de la señal de reloj, así como su efecto en los elementos síncronos que componen un sistema. Se estudian los problemas que aparecen como consecuencia del aumento en tamaño, complejidad y velocidad de los circuitos integrados, así como los cada vez más exigentes requerimientos de fiabilidad. Adicionalmente, se presentan diferentes alternativas de sincronización que pretenden soslayar dichos problemas, al tiempo que pueden aumentar las prestaciones del sistema.

CAPÍTULO 4. NUEVA ESTRATEGIA DE SINCRONIZACIÓN

En este capítulo define, analiza y caracteriza una estrategia de sincronización basada en la aplicación de varias señales de reloj levemente desfasadas que controlan los diferentes canales que componen los sistemas basados en el RNS. Se compara esta estrategia de sincronización con la metodología de sincronización basada en una señal de reloj única, que es la más frecuente en la sincronización de sistemas digitales. Ambas estrategias se aplican a una serie de sistemas de prueba y se analiza el efecto de la aplicación de las señales desfasadas, sobre las demandas de corriente y el consumo de los sistemas.

CAPÍTULO 5. APLICACIÓN EN SISTEMAS PROGRAMABLES

En el capítulo quinto se implementa un sistema basado en el RNS sobre un dispositivo lógico programable FPGA (*Field Programmable Gate Array*) protegido mediante una firma digital. Se estudia el efecto de la aplicación de la nueva estrategia de sincronización sobre el consumo de un sistema real en contraste con el sistema sincronizado mediante una única señal de reloj. Además se va a estudiar el efecto de la aplicación de varias señales desfasadas en las diferencias entre los modos de funcionamiento normal y de extracción de la firma con las dos metodologías de protección analizadas.

CAPÍTULO 6. CONCLUSIONES

Por último, el capítulo sexto contiene las conclusiones obtenidas y una valoración crítica de los resultados. Se recogen las aportaciones de esta memoria y se sugieren nuevas líneas de trabajo.

A final de la memoria se añaden tres apéndices con los siguientes contenidos:

APÉNDICE A. GENERACIÓN DE LA SEÑAL DE RELOJ

En el primer apéndice se analiza la generación de las señales de reloj así como los diferentes métodos para acondicionar la señal de reloj de referencia dentro del chip. Se estudian diversas configuraciones para conseguir la sincronización entre la señal de reloj externa y la interna, y la multiplicación de frecuencia dentro del chip.

APÉNDICE B. ANÁLISIS DE LOS DISPOSITIVOS Y MÓDULOS USADOS EN LAS SIMULACIONES

En el segundo apéndice se detallan los dispositivos y circuitos empleados en las simulaciones eléctricas y se justifican algunas decisiones de diseño en cuanto al dimensionado de dispositivos. Además se describe en detalle la arquitectura y funcionamiento de las memorias ROM que se han usado asiduamente en los sistemas simulados.

APÉNDICE C. PROGRAMA GENERADOR DE LIBRERÍAS DE MEMORIAS ROM

En el último apéndice se describe un programa desarrollado para este trabajo para la implementación de las memorias ROM, usadas en el diseño de los sistemas simulados, a partir de la descripción de la memoria y de los datos contenidos en las mismas.

Como finalización de este prólogo se van a dar algunas indicaciones para facilitar la lectura de esta memoria:

- la memoria está estructurada en capítulos y éstos se organizan en diferentes secciones y éstas en apartados y subapartados, que se numeran con varios dígitos. El primer dígito hace referencia al capítulo y los posteriores a la sección y diferentes apartados en el orden de inclusión en que van apareciendo. Ejemplo 4.5.2.3;
- las figuras y tablas se numeran con dos números, el primero hace referencia al capítulo y el segundo al orden de aparición dentro del capítulo. Además la memoria incluye índices de tablas y figuras para facilitar la búsqueda concreta de alguno de estos elementos. Ejemplo: Tabla 4.6, Figura 5.2;
- las expresiones matemáticas y ecuaciones se referencian de modo similar a las tablas y las figuras con dos números entre paréntesis, que representan el capítulo y al orden de aparición dentro del mismo. Ejemplo (2.1);

- Por último las referencias bibliográficas se designan entre corchetes con tres letras, correspondientes al apellido del primer autor y dos números relativos al año de publicación. En el caso de coincidencia en la denominación de las referencias se incluirá una letra al final. Las referencias se hallan ordenadas alfabéticamente al final de la memoria. Ejemplos [GAR99], [CAS07], [GON02a].

CAPÍTULO 1

INTRODUCCIÓN

En este primer capítulo se exponen las principales motivaciones que justifican la realización de esta tesis. Se realiza un estudio preliminar sobre la evolución de las características y prestaciones de los sistemas digitales y se comentan las dificultades que surgen durante el diseño de sistemas cada vez más extensos que funcionan a frecuencias cada vez mayores. Después se hace una breve introducción a la evolución de diferentes metodologías de sincronización y se propone una estrategia alternativa consistente en la aplicación de varias señales de reloj ligeramente desfasadas. También se hace una breve reseña al sistema numérico de residuos (RNS: *Residue Number System*), cuyas características motivaron su elección como base de los sistemas analizados durante el desarrollo de este trabajo.

1.1. Evolución y estado actual de las prestaciones en sistemas digitales

Desde el inicio de la fabricación de dispositivos integrados la tendencia dominante ha sido la disminución del tamaño de los transistores y el incremento en los niveles de integración de los circuitos y, como consecuencia, un aumento de la frecuencia de funcionamiento y la cantidad de dispositivos incluidos en un mismo circuito integrado. En 1965 Moore escribió un artículo [MOO65] en el que afirmaba: “*La complejidad de los componentes se ha multiplicado aproximadamente por 2 cada año. A corto plazo, se puede esperar que esta tasa se mantenga, o incluso que aumente. A largo plazo, la tasa de aumento es un poco más incierta, aunque no hay razón para creer que no permanecerá constante por lo menos durante 10 años. Esto significa que para 1975, el número de componentes en cada circuito integrado de mínimo coste será de 65.000. Creo que un circuito tan grande puede construirse en una sola oblea*”. La predicción de Moore, enunciada tres años antes de la aparición del microprocesador, se mantuvo hasta principios de los 70; a mediados de esa misma década se constató que dicho ritmo había

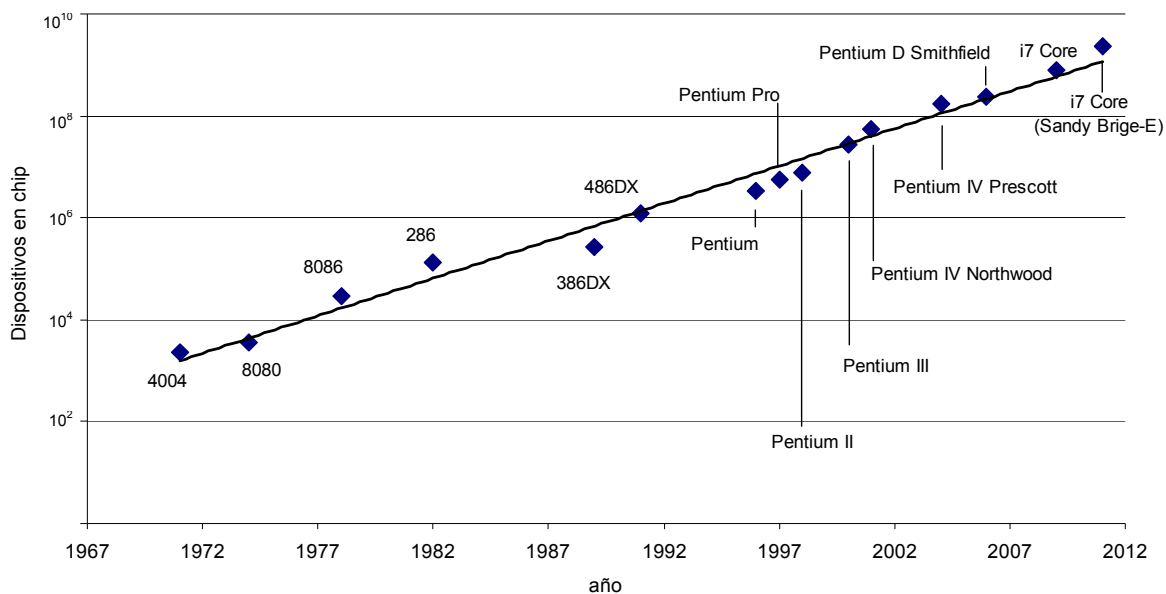


Figura 1.1. Evolución del número de dispositivos integrados en un solo microprocesador.

disminuido hasta ser necesarios aproximadamente dos años para que se duplicase el número de componentes en un sólo chip. Esta tendencia, conocida como Ley de Moore, se ha mantenido vigente hasta la actualidad, tal como se puede apreciar en la Figura 1.1, que muestra la evolución del número de dispositivos integrados en un solo chip de diferentes microprocesadores de los últimos 40 años. Este incremento está directamente relacionado con la disminución del tamaño característico de los dispositivos del chip. En la Figura 1.2 se representa la evolución de la longitud del canal de los transistores y se puede apreciar la tendencia exponencial de disminución de la misma. Los últimos valores de la gráfica muestran que el estado actual de la tecnología permite la fabricación de dispositivos con una longitud de canal de 32nm. A modo de ilustración sobre la evolución de la tecnología en la fabricación de dispositivos integrados se podría comparar el primer PC, el *Altair computer* de Intel producido en 1974, que incorporaba el microprocesador 8080 con 6.000 transistores implementados con una tecnología de 6µm capaz de funcionar a 2MHz, con el procesador de Intel *i7-390X Processor Extreme Edition (Sandy Bridge-E)* (2011) de séxtuple núcleo, con capacidad de 12 procesos en paralelo, que contiene 2.270 millones de transistores fabricados mediante un proceso de 32nm, con una frecuencia de funcionamiento de 3.30GHz capaz de alcanzar los 3.90GHz en modo turbo. Otros ejemplos del impresionante desarrollo de la industria electrónica hasta el momento son el hecho de que desde el año 2004 el sector de los semiconductores produce más transistores (y a un menor coste) que granos de arroz en todo el mundo, según la Asociación del Sector de Semiconductores de los Estados

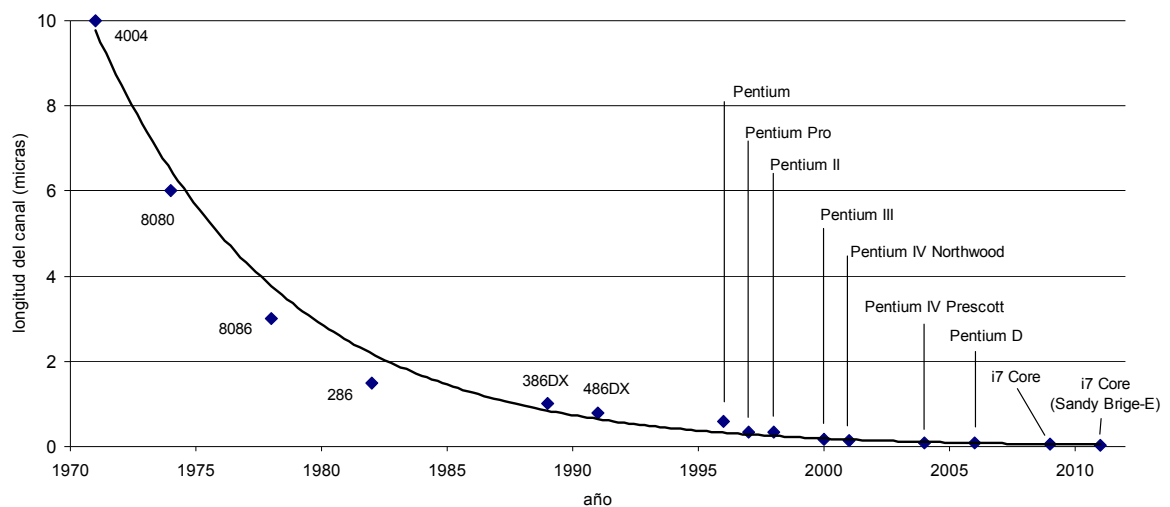


Figura 1.2. Evolución del tamaño característico de los transistores.

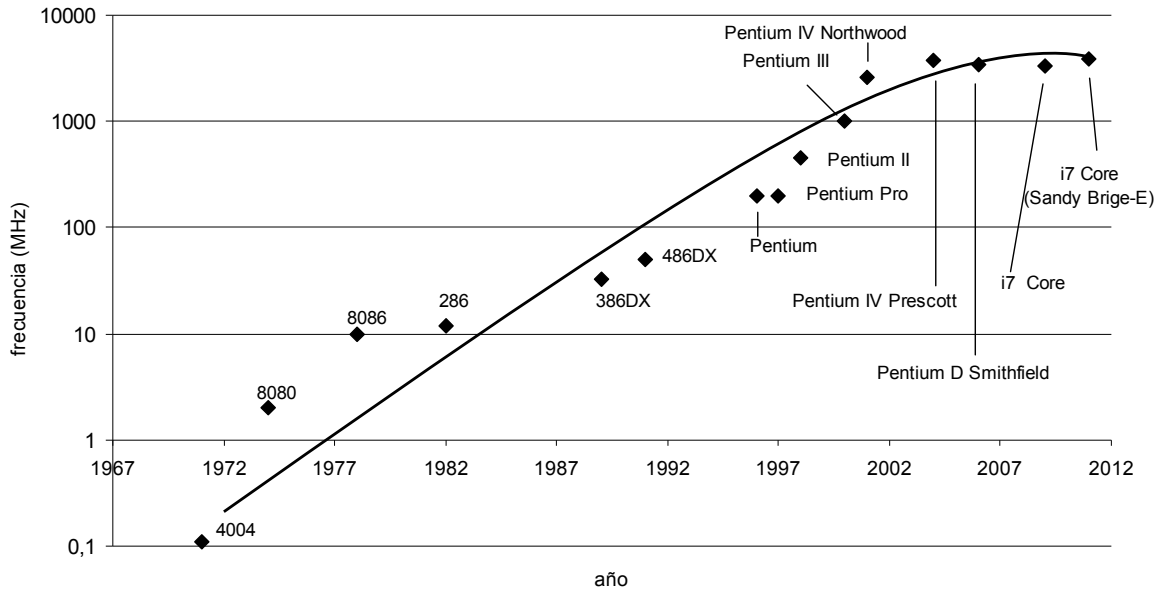


Figura 1.3. Evolución de la frecuencia de funcionamiento de microprocesadores.

Unidos (*U.S. Semiconductor Industry Association*), o que si el desarrollo de la industria aeronáutica hubiera sido similar al de los semiconductores, actualmente se podría viajar de Nueva York a París por un céntimo de euro y en menos de un segundo.

De seguir la progresión marcada por la ley de Moore en el año 2012 deberían existir chips de 10 GHz con 2.000 millones de transistores. Sin embargo, el mercado del microprocesador comercial se ha estancado en frecuencias de reloj de en torno a 4GHz; la disipación térmica de cientos de millones de transistores a tan elevadas frecuencias es muy complicada y los efectos capacitivos, inductivos y de emisión electromagnética cobran una mayor relevancia complicando extraordinariamente el diseño de circuitos integrados. La figura 1.3 muestra la evolución de la frecuencia de funcionamiento de los microprocesadores de los últimos 40 años. La alternativa actual parece dirigir sus esfuerzos hacia sistemas multiprocesador. En el primer tercio del año 2005 se comenzaron a comercializar los sistemas biprocesador. Los *Pentium Dual-Core* de Intel y los *Athlon X2* de AMD abrieron el camino a sus inmediatos sucesores: los procesadores *Core 2 Duo* de Intel, optimizados y capaces de un mayor rendimiento incluso a frecuencias inferiores (2.4GHz en la primera mitad de 2007). Actualmente se han desarrollado los procesadores de séxtuple núcleo, *i7 core Sandy Bridge-E* de Intel y el procesador *AMD FX 8150 Black Edition* de óctuple núcleo capaz de alcanzar los

4.2GHz de AMD. La conclusión es que se ha alcanzado un límite rentable en la frecuencia del reloj de los microprocesadores y los diseñadores incrementan la capacidad de procesamiento mejorando la arquitectura y aprovechando las ventajas del procesamiento paralelo. En sistemas supercomputador se ha producido la misma situación. Los supercomputadores más potentes, dedicados bien a la atención de múltiples usuarios o al cálculo masivo (meteorología, aeronáutica, prospección minera y petrolífera, etc.) mejoran su rendimiento mediante la paralelización masiva de microprocesadores. Como ejemplo, el supercomputador *Roadrunner* de IBM que funciona con de 12.240 procesadores *PowerXCell 8i* en paralelo y 6.912 procesadores *Opteron* de AMD de doble núcleo conectados mediante 92 km de fibra óptica y está diseñado para realizar alrededor de 1.7 petaFlops (*peta Floating-point Operations Per Second*, 10^{15} operaciones en punto flotante por segundo) con un consumo de 3 megawattios.

1.2. Problemática en la sincronización de circuitos VLSI

La disminución del tamaño característico de los transistores que forman los circuitos integrados actuales ha sido la causa del gran incremento tanto en el nivel de integración de los sistemas electrónicos como en la velocidad de operación de los mismos. La combinación de estos dos factores lleva a la implementación de sistemas VLSI (*Very Large Scale of Integration*) mayores, más complejos y más rápidos, lo que se traduce en que la influencia de una serie de efectos, hasta ahora de segundo orden, se haga más significativa. La espectacular disminución en el tamaño de los dispositivos y del periodo de la señal de reloj no se ha producido de manera proporcional en el error asociado a retardos, ruido o variaciones de proceso. A medida que los tiempos de conmutación se reducen al rango de los nanosegundos (y por lo tanto las frecuencias se incrementan hasta los GHz), las líneas de distribución de las diferentes señales comienzan a presentar efectos inductivos [KAR10, MAH06, ZHO91] que atenúan y retardan de manera apreciable la propagación de dichas señales. El incremento de la frecuencia de la señal de reloj tiene como resultado que se deban propagar componentes de alta frecuencia que sufren mayor atenuación, por lo que la señal de reloj se degrada, pasando

de un perfil abrupto a otro más suave. Este perfil suavizado es la causa de dos problemas de importancia capital a los que los diseñadores dedican gran parte de tiempo y recursos. Por un lado se producen incertidumbres en el instante que conmutan las señales, con lo que no se sabe con exactitud en momento en el que los sistemas cambian de estado. A medida que la frecuencia de funcionamiento aumenta se hace más significativa dicha incertidumbre frente al periodo del reloj. En el Capítulo 3 se abundará en esta cuestión y en dos de los principales efectos que esto provoca: el *skew* y el *jitter*. Por otro lado, se produce un incremento del consumo dinámico ya que, en la tecnología CMOS, los dispositivos conducen únicamente durante la fase de conmutación. Otro problema derivado de los altos niveles de integración y, consecuentemente del aumento en el número de dispositivos, es el brusco incremento de las demandas de corriente de las fuentes de alimentación en instantes muy concretos. Esto provoca variaciones en las tensiones de alimentación de los sistemas con el consiguiente aumento de las incertidumbres en la señal de reloj.

De modo general se puede asumir que la señal más crítica en una sistema digital síncrono es la señal de reloj; es la que se debe propagar a lo largo de más dispositivos, debe llegar a todos los módulos operativos del sistema y, por lo tanto, ha de propagarse a través de largas líneas de interconexión, es la señal que más rápidamente debe conmutar y su calidad determina el funcionamiento del resto del sistema. En sistemas digitales de altas prestaciones el consumo asociado a la distribución de la señal de reloj puede alcanzar hasta el 60% del consumo total del chip, por lo que se debe poner especial énfasis en el diseño de la red de distribución de la señal de reloj.

Algunos de los problemas de mayor relevancia, y que más pueden afectar al rendimiento de los sistemas, relacionados con la generación y distribución de la señal de reloj son los siguientes:

- dificultad para que una única señal de reloj se transmita y actúe de forma simultánea en todos los módulos de un circuito extenso o con elevados requerimientos de velocidad (*skew*) [NIC12, KIM10, ESM10, KHL04, CEL99, KEE91, AFG90, YUA89];
- retardo en la propagación de la señal de reloj a lo largo de una línea,

ocasionado por aumento del tamaño de los circuitos integrados junto con el incremento de la frecuencia de operación, que provoca la limitación de la velocidad de funcionamiento global [MOH11, HIG11, CEL99, ZHO91];

- dispersión en el periodo de la señal de reloj (*clock jitter*), ocasionado por efectos de ruido que se hacen más significativos cuando disminuyen las corrientes de funcionamiento y el periodo [JAN12, SHI10, JIA10, YAM01, HEY00, HER99];
- la demanda de corriente puede alcanzar niveles elevados en determinados instantes muy concretos, mientras que en el resto del tiempo el consumo es sensiblemente más bajo. Esto es consecuencia del funcionamiento síncrono, que requiere que las diferentes partes del circuito funcionen simultáneamente [SOM11, BHA98];
- la necesidad de conducir corrientes elevadas supone el uso de líneas de interconexión de gran sección, que a su vez implican grandes capacidades que ralentizan la carga y descarga de dichas líneas, producen efectos de acoplamiento entre líneas y ocupan grandes áreas [KAR10, BAI04, LEE98];
- aparición de efectos inductivos en las interconexiones debido al incremento de las frecuencias de funcionamiento, con la consiguiente degradación en el perfil de la señal de reloj y la aparición de señales reflejadas en las interconexiones [KAR10, MAH06, ZHO91].

La falta de sincronía entre diferentes módulos del sistema (*skew*) se produce cuando el tiempo de respuesta de los módulos de proceso se hace comparable a las diferencias entre los retardos de la señal en cada ruta de distribución, de tal forma que la señal de reloj adquiere valores sensiblemente diferentes en distintas partes del circuito en un mismo instante. Este efecto se hace relevante en circuitos de gran tamaño y con frecuencias de funcionamiento elevadas. Una estrategia para la reducción del *skew* en un circuito requiere el uso de la teoría de estructuras RC en árbol (*RC-trees*), consistente en modelar las líneas de distribución como una estructura ramificada en la que cada

línea de distribución es una asociación de una resistencia y una capacidad, y cada módulo operativo una capacidad terminal [RAM94]. El análisis del sistema de distribución según su modelo RC permite calcular el retardo a lo largo de cada línea y, de esta manera, establecer una distribución de los módulos operativos y elementos de retardo adicionales, de forma que el retardo en la distribución de reloj sea lo más similar posible a lo largo de las diferentes líneas de distribución. Éste es un método de resolución de peor caso, ya que conduce a realizaciones seguras pero implica un aumento en el periodo del reloj como consecuencia de la inclusión de *buffers* cuya única finalidad es producir un retardo adicional en una ruta para igualarla al resto [RAM94]. Una alternativa consiste en realizar una aproximación estadística que proporciona situaciones en las que la probabilidad de error es muy pequeña, pero no nula, y produce periodos de reloj menores [JIA01]. Otros enfoques distribuyen los módulos operativos de forma simétrica (en forma de H o X, *H-trees* o *X-trees*, respectivamente), de manera que las señales de reloj sufran los mismos retardos en todos los caminos [FRI01]. Estos métodos tienden a igualar el retardo a lo largo de las diferentes líneas de distribución del reloj evitando diferencias en la sincronía, pero no resuelven el problema del retardo global del sistema y acentúan el problema del consumo de potencia, ya que, en definitiva, como resultado los módulos funcionan al mismo tiempo, haciendo que la demanda de corriente en determinados instantes se incremente.

1.3. Estrategias de sincronización de sistemas VLSI

Existen dos metodologías diferentes de sincronización en los sistemas VLSI, dependiendo de si los elementos de memoria están controlados por nivel (*latches* o cerrojos) o por flanco (*flip-flops*). En los sistemas controlados por nivel, los bloques combinacionales que realizan el procesamiento de los datos están separados por estructuras que pueden estar en dos estados dependiendo del nivel (alto o bajo) de la señal de reloj. En estado abierto, se establece conexión (y por lo tanto transferencia de datos) entre los bloques y, en el estado cerrado, se bloquea el paso de datos aislando una etapa de la otra. En los sistemas gobernados por flanco los elementos de memoria son biestables del tipo D que transfieren el valor de la entrada al de la salida cuando ocurre un flanco en la señal de reloj.

Los sistemas sincronizados por nivel son más rápidos y requieren menos recursos hardware, pero su funcionamiento es más complicado y su optimización más compleja. Por otro lado los sistemas controlados por flanco tienen un funcionamiento menos complejo y se pueden optimizar de forma más sencilla, pero son más lentos y requieren más dispositivos.

Los sistemas controlados por nivel se pueden sincronizar utilizando diferentes estrategias que pueden requerir el uso de una o más señales de reloj. Una de las primeras estrategias de sincronización en este tipo de sistemas fue la NPTC (*Non-overlapping Pseudo Two Phase Clock*) [YUA89] que requiere la distribución de cuatro señales de reloj no solapadas. Los recursos necesarios y la condición necesaria de no-solapamiento tienen como consecuencia fuertes restricciones en cuanto a la calidad de las señales y los límites permisibles de *skew* y retardo. En este mismo tipo de sistemas la técnica de sincronización NORA (*No Race*) [GON83] simplificaba la labor de diseño limitando a dos las señales necesarias (señal de reloj y complementaria), pero seguía presentando la dificultad de tener que distribuir dos señales de reloj. Por último la metodología TSPC (*True Single Phase Clock*) [KAR88, AFG90] sólo necesita de una única señal de reloj y es actualmente la técnica de sincronización en uso. Esta técnica es la más adecuada en los sistemas controlados por flanco, ya que lo único que se necesita de la señal que controla el sistema es su cambio de nivel, es decir, su flanco. En el tercer capítulo se analizan las técnicas de distribución de reloj basadas en TSPC más frecuentes en la actualidad.

Todas las metodologías de sincronización son en una u otra medida sensibles a los mismos problemas. Las estrategias que requerían más señales de reloj eran más vulnerables al *skew* y al *clock jitter*; la técnica TSPC presenta mejor comportamiento frente al *skew*, pero éste se vuelve a hacer patente a medida que aumenta la frecuencia de funcionamiento. En lo concerniente a esta memoria, se empleará un sistema controlado por flanco de bajada, y por lo tanto se aplicará una única señal de reloj en la que se inducirá un *skew* controlado para aplicar diferentes señales levemente desfasadas a los diferentes canales que formarán el sistema analizado.

1.4. Nueva estrategia de sincronización

Como alternativa a los circuitos con una única señal de reloj se ha propuesto [GAR99] una estructura que a partir de una única señal de reloj genera otras señales ligeramente desfasadas respecto de la original. Cada una de estas señales controla diferentes partes del circuito de forma que éstas funcionan en diferentes instantes, dentro del mismo ciclo de reloj, distribuyéndose de esta manera el consumo y reduciéndose los picos de corriente. Esta estrategia también elimina la necesidad de una sincronía global. El análisis detallado de las restricciones que deben cumplir las señales, de su diseño, de los circuitos que las generan y los resultados de su aplicación a sistemas concretos se realizará en el capítulo cuarto.

La idea de usar una señal levemente desfasada (como consecuencia del *skew*) para mejorar el rendimiento de los sistemas, aparece en la bibliografía en diversos trabajos [FISH90, FRI93, FRI01]. En concreto, se plantea la posibilidad de disminuir el periodo de funcionamiento y, consecuentemente, aumentar la frecuencia, haciendo llegar a registros adyacentes señales desfasadas (la más retrasada al primero de ellos). La aplicación de esta estrategia está limitada por la aparición del problema de carreras si el desfase resulta ser superior al tiempo necesario para realizar el proceso de los datos tal como se analizará en el tercer capítulo. Otros autores han propuesto sistemas que producen un *skew* sobre una única señal para conseguir un funcionamiento distribuido [YOO99] y limitar los valores de pico de las corrientes [BEN96]. Posteriormente se ha propuesto [GAR99] la generación de diferentes señales levemente desfasadas para controlar diferentes canales en un sistema basado en el RNS. En esta memoria se parte de esta última idea y se aplica y analizan los resultados sobre varios sistemas, un filtro CIC para diezmado de Hogenauer, un controlador PID y un sistema para el cálculo de la transformada *wavelet* 1D-DWT por transformación al dominio de los índices.

Para poder aplicar varias señales de reloj en un circuito digital es necesario que el diseño permita el funcionamiento paralelo e independiente de módulos diferentes. De esta manera, tan sólo es necesario que los datos estén disponibles de forma síncrona a la salida. Los circuitos de cálculo basados en el Sistema Numérico de los Residuos cumplen con esta característica. Una característica del RNS [SZA67] es que en las operaciones de suma no existe un acarreo que se transmita de un residuo a otro, por lo

que el sistema consta de una serie de canales que pueden operar de forma independiente. El RNS es un sistema de representación numérica muy eficaz en el cálculo de sumas y productos, pero presenta dificultades en operaciones de división, comparación y detección de signo, y desbordamiento, por lo que el uso de esta alternativa queda restringido a sistemas en los que sea necesario calcular gran cantidad de sumas y productos. Sin embargo, existen multitud de aplicaciones en las que estas limitaciones no son un obstáculo, entre las que se incluyen la transformada rápida de Fourier (FFT: *Fast Fourier Transform*) y otras transformadas discretas y, en general, la mayoría de aplicaciones de procesamiento digital de señales (DSP: *Digital Signal Processing*).

Es objetivo de este trabajo profundizar más en las posibilidades y características de esta nueva estrategia de sincronización, así como comprobar su comportamiento en sistemas reales.

1.5. Conclusión

En este capítulo se han planteado las diferentes técnicas que se han usado hasta el momento para sincronizar sistemas VLSI y se han sugerido los principales problemas que pueden surgir en la sincronización de circuitos extensos que operan a elevadas frecuencias de reloj. Se ha comentado una estrategia de sincronización para sistemas controlados por flanco, consistente en generar diferentes señales de reloj desfasadas entre sí de manera que cada señal controla una parte del circuito. La plena aplicación de esta técnica requiere sistemas cuyo funcionamiento se pueda separar en diferentes módulos o canales independientes entre sí, de forma que cada canal esté gobernado por una de estas señales desfasadas. Los sistemas basados en el RNS presentan esta característica, por lo que serán un campo más que adecuado para la aplicación de la estrategia en estudio.

CAPÍTULO 2

FUNDAMENTOS DEL SISTEMA NUMÉRICO DE RESIDUOS

En este capítulo se introducen los Sistemas Numéricos de Residuos y las operaciones más comunes en la aritmética de residuos, tales como suma, resta, multiplicación y escalado. Además, se presentan algunas otras características del RNS, tales como la posibilidad de detección y corrección de errores, nociones sobre conversión, etc. También se describe la principal ventaja del RNS consistente en que es un sistema de representación numérica en el que cada número es representado por un conjunto de módulos de tal manera que las operaciones de suma, diferencia y producto de varios números no implica la propagación de acarreo entre los diferentes módulos que representan a cada uno de los números, con lo que dichas las operaciones se resuelven en paralelo para cada canal de manera independiente.

2.1. Los Sistemas Numéricos de Residuos

Los sistemas numéricos usuales son lineales, posicionales y pesados, es decir, el valor de cada dígito depende de la posición que ocupe. Los sistemas basados en el RNS, en cambio, parten de un conjunto de bases previamente establecidas y cada número queda representado por el conjunto de residuos en dichas bases, esto es, los restos de las divisiones enteras de dicho número entre cada una de las bases que definen el sistema numérico. Cada número queda unívocamente representado por el conjunto de sus residuos siempre que sea menor que el producto de las bases tal y como establece el Teorema del Resto Chino (CRT: *Chinese Remainder Theorem*) [TAY84, SOD86].

Teorema 2.1. Considérese el conjunto de módulos $\{m_1, m_2, \dots, m_n\}$ todos ellos primos relativos entre sí. Cualquier número X tal que $X < M$ siendo:

$$M = \prod_{i=1}^n m_i \quad (2.1)$$

tiene una representación única dada por $X = [x_1, x_2, \dots, x_n]$ donde:

$$x_i = X \bmod m_i \quad (2.2)$$

Demostración. Supóngase que existen dos números X e Y que tengan la misma representación en residuos, es decir, $x_i = y_i$ para todo i , siendo:

$$\begin{aligned} x_i &= X \bmod m_i \\ y_i &= Y \bmod m_i \end{aligned}$$

Entonces se cumple que $X-Y$ debe ser múltiplo de cada m_i y, por lo tanto, también del mínimo común múltiplo de los m_i . Como los m_i son primos entre sí, su mínimo común múltiplo es M y al ser $X-Y$ múltiplo de M , no pueden ser ambos simultáneamente mayores que M .

De este modo, el RNS definido por los módulos $\{m_1, m_2, \dots, m_n\}$ permite representar M enteros diferentes en el rango $[0, M-1]$, formando un grupo cíclico.

Por otro lado, el anillo de enteros módulo M , $Z(M)$, es isomorfo a la suma directa de anillos de los enteros m_1, m_2, \dots, m_n :

$$Z(M) = Z(m_1) \oplus Z(m_2) \oplus \dots \oplus Z(m_n) \quad (2.3)$$

De esta última conclusión deriva el interés por el RNS, ya que la aritmética queda definida sobre el anillo de los enteros módulo M y éste, a su vez, es isomórfico a la suma directa de los anillos correspondientes a los diferentes módulos que definen el RNS. Esto significa que la representación en RNS de la suma (o resta o producto) de dos números pertenecientes a $Z(M)$ se puede obtener como las sumas (o restas o productos) de las representaciones en RNS correspondientes a los números operados en $Z(m_i)$. Por lo tanto, la aritmética en el RNS se realiza sobre n canales independientes de rango dinámico inferior al definido por el conjunto de los mismos sin propagación de acarreo entre ellos. Si \diamond representa suma, resta o multiplicación, y $0 \leq X, Y < M$, entonces se satisface:

$$Z = (X \diamond Y) \bmod M \equiv [z_1, z_2, \dots, z_n] \quad (2.4)$$

$$z_i \equiv (x_i \diamond y_i) \bmod m_i \quad (i = 1, \dots, n)$$

Es decir, la aritmética en el RNS se realiza sobre n canales independientes de rango dinámico menor que el definido por el conjunto de los mismos; además, no hay propagación de acarreos entre cada uno de los residuos que representa a cada número y la aritmética es cerrada en cada una de estas posiciones. Esto es lo que permite realizar sumas y multiplicaciones de números grandes a la misma velocidad que para números pequeños, ya que la velocidad estará limitada por el canal más lento.

Respecto a la resta, dado que:

$$(X) \bmod m - (Y) \bmod m = (X - Y) \bmod m \quad (2.5)$$

esta operación no presenta ningún problema en aritmética de residuos y las operaciones en los diferentes módulos se llevan a cabo sobre números positivos; además, para representar números negativos sólo es necesaria una traslación del rango representado por el RNS.

Ejemplo, Operaciones de suma, diferencia y producto en RNS definido por $\{5, 3, 2\}$, con $M=30$. Resultados obtenidos mediante aplicación directa de (2.4) y (2.5). Nótese que cada pareja de residuos se suma, resta o multiplica de forma independiente, eliminando cualquier acarreo entre posiciones.

$$\begin{array}{r}
 21 \quad [1, 0, 1] \\
 \begin{array}{r}
 |_{+30} 14 \\
 \hline
 5
 \end{array}
 \quad
 \begin{array}{r}
 |_{+30} [4, 2, 0] \\
 \hline
 [0, 2, 1]
 \end{array}
 \end{array}
 \qquad
 \begin{array}{r}
 6 \quad [1, 0, 0] \\
 \begin{array}{r}
 |_{-30} 11 \\
 \hline
 25
 \end{array}
 \quad
 \begin{array}{r}
 |_{-30} [1, 2, 1] \\
 \hline
 [0, 1, 1]
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 8 \quad [3, 2, 0] \\
 \begin{array}{r}
 |_{\times 30} 5 \\
 \hline
 10
 \end{array}
 \quad
 \begin{array}{r}
 |_{\times 30} [0, 2, 1] \\
 \hline
 [0, 1, 0]
 \end{array}
 \end{array}$$

Como ya se mencionó antes, la velocidad a la que se puede operar viene determinada sólo por el más lento de los diversos canales en los que se realiza la operación para cada módulo. Es preciso, por tanto, buscar en la elección de los módulos un compromiso entre la velocidad, rango dinámico a representar y recursos a utilizar.

Otra cuestión que entra en juego en la elección de los diferentes módulos es la existencia o no de circuitos aritméticos convencionales para estos módulos. A este respecto, y dadas las aplicaciones de la representación en residuos, es preciso realizar las operaciones a la mayor velocidad posible, por lo que la sustitución de la lógica tradicional por circuitos de memoria ROM [BAY87] en las que se almacena el resultado de la operación para cada combinación de entradas es una alternativa muy interesante, cada día más, a medida que aumenta la velocidad y el nivel de integración de los diferentes dispositivos de memoria. En consecuencia, y según se ha comentado

anteriormente, las operaciones que implica el RNS van adquiriendo mayor importancia a medida que aumenta la capacidad de memoria que facilita la tecnología actual.

2.1.1. Representación en RNS de enteros

El RNS también permite representar cualquier intervalo numérico de enteros mediante una traslación de rango, lo cual resulta práctico para la representación de enteros. Es habitual tomar rangos simétricos en cuyo caso los intervalos de números representables dependen de si M es par o impar:

- si M es par $\rightarrow \left[-\frac{M}{2}, \dots, 0, \dots, \frac{M}{2} - 1 \right]$
- si M es impar $\rightarrow \left[-\frac{M-1}{2}, \dots, 0, \dots, \frac{M-1}{2} \right]$

En este caso el cálculo del residuo de un número entero se realiza mediante la expresión:

$$x_i = \begin{cases} X \bmod m_i & \text{si } X \geq 0 \\ (M - |X|) \bmod m_i & \text{si } X < 0 \end{cases} \quad (2.6)$$

Ejemplo. Si se consideran las bases $m_1=2$ y $m_2=3$ y se aplican directamente (2.1), (2.2) y (2.6) se obtiene $M=6$ y las representaciones;

Rango $\{0,5\}$	Rango $\{-3,2\}$	RNS
0	0	[0,0]
1	1	[1,1]
2	2	[0,2]
3	-3	[1,0]
4	-2	[0,1]
5	-1	[1,2]

2.1.2. El sistema numérico de residuos simétrico (SRNS)

Otra alternativa de representación modular es el RNS simétrico (SRNS: *Symmetric Residue Number System*) [TAY84], donde el residuo $(x_i \bmod X)$ es un número entero con signo en el conjunto:

- $\left[-\frac{m_i}{2}, \dots, 0, \dots, \frac{m_i}{2} - 1 \right]$ si la base m_i es par;
- $\left[-\frac{m_i - 1}{2}, \dots, 0, \dots, \frac{m_i - 1}{2} \right]$ si la base m_i es impar;

y el número de enteros representables de manera única sigue siendo el producto de los módulos M . Cualquier entero $X \in [0, M-1]$ se codifica de manera única en el SRNS con la n -tupla $[x_1, x_2, \dots, x_n]$, siendo:

$$x_i = |X|_{m_i} = X - \left\lfloor \frac{X}{m_i} \right\rfloor m_i \quad (2.7)$$

2.1.3. El sistema numérico de residuos cuadrático (QRNS)

En el caso de los números complejos, de frecuente uso en procesamiento digital de señales y comunicaciones, el RNS convencional no aporta ventaja alguna en el caso de la multiplicación, que necesita de cuatro multiplicadores y dos sumas. Para operar con números complejos es más apropiado el uso del sistema de residuos cuadrático QRNS (*Quadratic Residue Number System*) [VAN78, LEU81, JEN83] que reduce considerablemente la complejidad de la multiplicación compleja al requerirse únicamente dos productos.

Si se considera un canal módulo m y un número complejo $z = x + iy$ de manera que $x, y \in \{0, 1, \dots, m-1\}$ y siendo j la unidad imaginaria ($j = \sqrt{-1}$), la representación de z en el QRNS queda definida por el par (c, c^*) :

$$\begin{aligned} c &= (x + \bar{j}y) \bmod m \\ c^* &= (x - \bar{j}y) \bmod m \end{aligned} \quad (2.8)$$

siendo $\bar{j} \in \{0, 1, \dots, m-1\}$ de tal modo que $\bar{j}^2 = (-1) \bmod m$.

Con estos cambios se desacoplan las componentes real e imaginaria de modo que el producto pasa de requerir cuatro operaciones con dos niveles a dos operaciones en un nivel. De este modo la aplicación del QRNS para el cálculo complejo reduce la cantidad de recursos e incrementa la velocidad de operación frente al uso del RNS convencional. Taylor [TAY85] lo utilizó para el cálculo de la FFT de 4 puntos. En esta aplicación el uso del QRNS supone reducciones de 34 a 22 operaciones de suma y de 4 a 3 niveles de operación.

2.2. Operaciones complicadas de realizar en el RNS

En apartados anteriores se ha mostrado la utilidad del RNS en operaciones de suma, resta y multiplicación. Sin embargo existen determinadas operaciones que presentan dificultades para ser llevadas a cabo mediante el RNS; concretamente las dificultades provienen del hecho que en el sistema RNS no existe un criterio que permita determinar si un número es mayor que otro. Diversos autores han planteando estrategias o metodologías para realizar estas operaciones, pero en todos los casos el RNS sigue siendo poco ventajoso en los ámbitos que requieren de dichas operaciones.

2.2.1. Comparación de magnitud y división

En los sistemas convencionales de representación con base fija cada dígito tiene un peso perfectamente determinado, lo que permite comparar dos números haciéndolo dígito a dígito; del mismo modo, es fácil la detección de signo, sin más que comprobar, en base 2 y para las diferentes formas de representación (signo y magnitud, y complementos a uno y a dos) el valor del bit más significativo. Desgraciadamente, esto no es posible en el RNS; no caben afirmaciones del tipo $[1, 2, 0] < [0, 3, 0]$ con el conjunto de módulos $\{3, 4, 5\}$. Esta dificultad para la comparación de magnitudes y la detección de signo limita las aplicaciones del RNS frente a las arquitecturas aritméticas tradicionales y lo restringe a aplicaciones en las que se requiere un elevado número de sumas y productos. Además, dado que la división es una sucesión de restas y comparaciones, es una operación difícil de implementar en el RNS. Al ser un sistema de representación de

enteros, el RNS no es cerrado para la división. Banerji propuso en 1981 [BAN81] una rutina de división aproximada a partir de la división estándar de enteros; Kinoshita lo había desarrollado para el SRNS en 1974 [KIN74] y más recientemente se han propuesto otros algoritmos de división en el RNS [HIT95, POS96, HIA97], pero la complejidad de esta operación sigue siendo mucho mayor que la de suma o multiplicación.

2.2.2. Detección de desbordamiento

Una cuestión relacionada con lo anterior es la detección de desbordamiento (*overflow*); en realidad, no existe desbordamiento en el RNS, ya que la aritmética queda definida sobre un grupo cíclico; sin embargo, cuando el RNS es utilizado para la resolución de algoritmos del tipo de la ecuación:

$$y = \sum_{i=1}^n \alpha_i x_i \quad (2.9)$$

es preciso asegurar que ningún resultado parcial supere el rango dinámico máximo, ya que esto conduciría a un resultado final erróneo. Si en los sistemas convencionales la detección de desbordamiento se realiza fácilmente inspeccionando el valor del dígito más significativo, en la representación con residuos habría de llevarse a cabo como comparación de magnitud, que como se ha visto, es una operación compleja. Este problema ha llevado a la introducción del escalado para evitar el desbordamiento durante el funcionamiento del sistema. El escalado es la composición de una variable con una constante de valor conocido; se trataría, por tanto, de una forma especial de división, lo que representa un problema importante para su realización. Sin embargo, el hecho de que el divisor sea constante simplifica el problema hasta el punto de hacerlo abordable en determinadas aplicaciones. Jullien propuso un método basado en la utilización de ROM [JUL78]; en concreto, su trabajo recoge dos propuestas: la primera consiste en modificar iterativamente el número a fin de conseguir una división exacta que proporcione un resultado truncado o redondeado; la segunda de las propuestas supone una mejora respecto del hardware necesario y se basa en la suma de ciertos vectores, estimados según una métrica, para obtener el número escalado. Sin embargo,

convierten a entero todas las posibles combinaciones de $N-1$ residuos; si existe error, una de estas combinaciones produce un entero en el rango $[0, M-1]$, y el resto de las $N-1$ restantes proporcionan valores en $[M, M_T-1]$. Sin embargo, si no hay errores, todos producen un valor inferior a M . De este modo, si hay un error, la combinación que produce el entero menor que M indica la localización y el valor del residuo incorrecto. Jenkins y Krozmeier [JEN83] desarrollaron un método similar para el Sistema Numérico de Residuos Cuadrático (QRNS).

2.3. Técnicas de conversión para el RNS

Ya se han visto las ventajas que representa el uso del RNS en la realización de operaciones aritméticas de sumas y productos. Sin embargo, los datos de entrada al sistema y la salida deben estar representados en el sistema binario convencional debido a que los sistemas que capturan o suministran los datos y aquellos que los deben presentar o almacenar utilizan en la gran mayoría de los casos dicha representación. Es por lo tanto vital para la utilidad del RNS disponer de técnicas de conversión eficientes y fiables en ambos sentidos, binario a RNS y RNS a binario. En los siguientes apartados se estudian y analizan las técnicas de conversión más frecuentes.

2.3.1. Conversión de binario a RNS

En este apartado se va a abordar la conversión de valores enteros a su representación en residuos, y viceversa. El primero de estos aspectos se ve condicionado una vez más por el tipo de aplicaciones del RNS; como ya se ha comentado, la complejidad de las soluciones implicando el tratamiento de residuos frente a esquemas aritméticos convencionales sólo se justifica en aplicaciones de alta velocidad, lo que requiere que la adquisición de datos y la conversión de decimal a residuos se ajusten a esta velocidad. Por tanto, si se trata con datos analógicos [RAD98], la conversión A/D debe realizarse de manera muy rápida, lo que requiere convertidores de tipo flash con un número de bits relativamente pequeño, por lo común entre 6 y 12 bits. Esta corta longitud de palabra permite que la salida del convertidor A/D sirva de dirección para chips de memoria en los que se hayan almacenados los diferentes residuos del valor de entrada para cada uno de los módulos m_i . Esto requiere el uso de tablas de consulta (LUT: *Look-*

Up Table) de $2^t \times \lceil \log_2(m_i) \rceil$ para longitudes de palabra de t bits. Para el caso de datos digitales, Szabo y Tanaka [SZA67] propusieron un método muy simple para generar los residuos a partir de la representación en complemento a dos; si X está representado por el número de t bits:

$$X = -B_{t-1}2^{t-1} + \sum_{j=0}^{t-2} B_j 2^j \quad (2.10)$$

donde B_{t-1} es el bit de signo y el residuo $x_i = X \bmod m_i$ se genera como:

$$x_i = \left(B_{t-1}(m_i - 2^{t-1} \bmod m_i) + \sum_{j=0}^{t-2} B_j(2^j \bmod m_i) \right) \bmod m_i \quad (2.11)$$

De este modo, si se define la función:

$$F_i(j) = \begin{cases} 2^j \bmod m_i & j = 0, \dots, t-2 \\ m_i - 2^j \bmod m_i & j = t-1 \end{cases} \quad (2.12)$$

es inmediato que:

$$x_i = \left(\sum_{j=0}^{t-1} B_j F_i(j) \right) \bmod m_i \quad (2.13)$$

y los diferentes residuos pueden obtenerse almacenando las funciones F_i en ROM y sumando los contenidos en sumadores módulo m_i . El tamaño de las tablas requeridas resulta ser $t \times \lceil \log_2 m_i \rceil$, lo cual supone una importante reducción en el tamaño de la LUT aunque este método requiere la inclusión de $t-2$ sumadores modulares.

Como se ha visto existen varias alternativas para la conversión de números binarios a sus representaciones en RNS y todas ellas requieren del uso de memorias ROM para ser usadas como tablas de consulta. En esta tesis se va a emplear un sistema de conversión por bloques que presenta la ventaja de sustituir una memoria ROM por varias de menor tamaño aprovechando el hecho de que, si se reducen a la mitad la

anchura de de la palabra de direcciones y la anchura del dato, la tabla resultante es cuatro veces menor que la original. Más aún, en la implementación mediante las FPGAs es frecuente encontrar que los elementos lógicos configurables (*CLB Configurable Logic Block*) están formados por generadores de funciones que pueden actuar como una LUT de cuatro entradas con lo que la implementación física es directa. Considérese un dato binario de n bits que se descompone en p bloques de b bits:

$$X = [x_{n-1}, x_{n-2}, \dots, x_0] = [\tilde{x}_{p-1}, \tilde{x}_{p-2}, \dots, \tilde{x}_0] \quad (2.14)$$

en ese caso se puede expresar:

$$X = 2^{(p-1)b} \tilde{x}_{p-1} + 2^{(p-2)b} \tilde{x}_{p-2} + \dots + 2^b \tilde{x}_1 + \tilde{x}_0 \quad (2.15)$$

donde es posible que el bloque $p-1$ tenga menos de b bits lo cual supondría que la última tabla podría ser menor. Una vez calculados los módulos de cada sumando de (2.15), se realiza la suma modular de todos los elementos para la obtención definitiva del residuo del dato de entrada. La transformación de (2.15) a módulo m se realiza

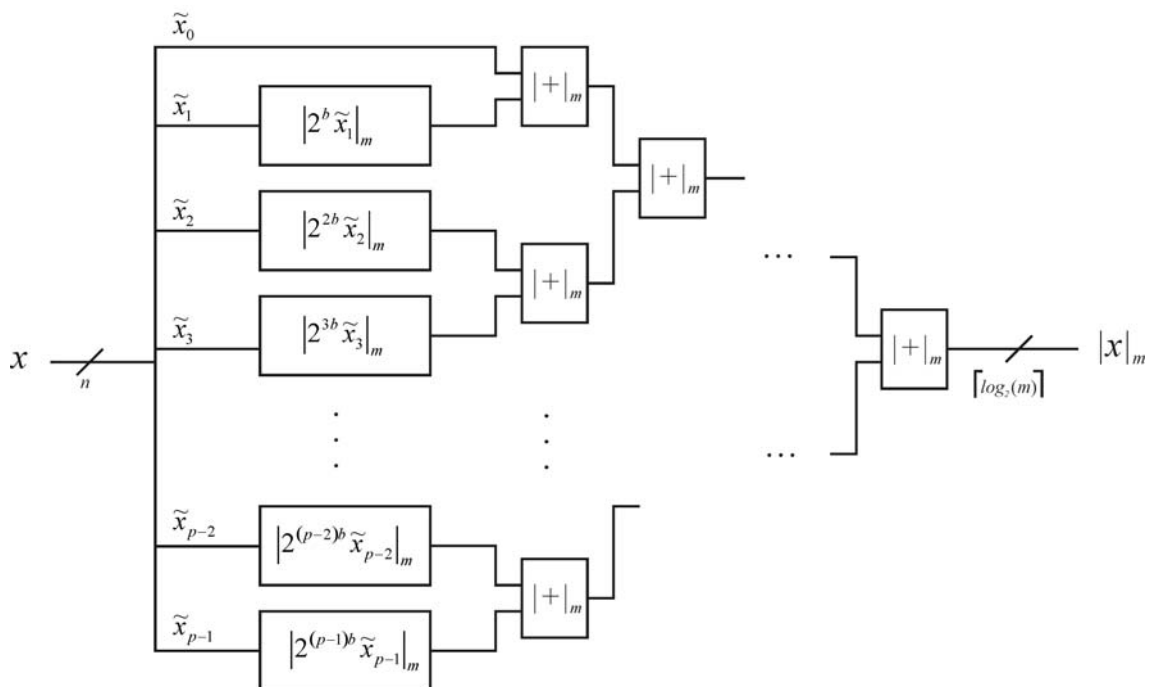


Figura 2.1. Conversión por bloques de binario a RNS.

según la expresión:

$$|X|_m = \left\| 2^{(p-1)b} \tilde{x}_{p-1} \right\|_m + \left\| 2^{(p-2)b} \tilde{x}_{p-2} \right\|_m + \left\| 2^b \tilde{x}_1 \right\|_m + \left\| \tilde{x}_0 \right\|_m \quad (2.16)$$

Una ventaja adicional de este método de cálculo consiste en que cualquier sumando de (2.15) que tenga en todo caso un valor inferior al módulo del canal correspondiente, coincide con su módulo y, por lo tanto, no necesita ser almacenado en una tabla, tan solo es necesario añadir la lógica necesaria para mantener la latencia en cada canal. En la Figura 2.1 se muestra la implementación física de (2.16) donde se ha supuesto que solamente el sumando ‘ \tilde{x}_0 ’ de (2.15) es inferior al módulo considerado m .

2.3.2. Conversión de RNS a binario

En lo que se refiere a la conversión desde el RNS a binario, una n -tupla de residuos puede convertirse a un valor entero de diversas maneras; en primer lugar, se puede realizar un producto interno que lleva a conversión por bases mixtas (MRC: *Mixed Radix Conversion*); este método genera unos coeficientes, de bases mixtas, que se calculan a través de algoritmos recursivos [JUL78, MIL98]. Otro posible método se basa en la aplicación directa del teorema del resto chino, CRT, de forma que:

$$X = \left(\sum_{i=1}^n s_i (x_i \cdot s_i^{-1}) \text{mod } m_i \right) \text{mod } M \quad (2.17)$$

donde:

$$s_i = \frac{M}{m_i} \quad (2.18)$$

y s_i^{-1} representa el inverso multiplicativo de s_i , es decir, el número entero que verifica:

$$(s_i \cdot s_i^{-1}) \text{mod } m_i = 1 \quad (2.19)$$

Ejemplo. A continuación se muestra la obtención del número decimal 53 a partir de su representación en residuos con las bases 2, 5 y 7.

Según (2.2) es claro que 53 queda representado unívocamente por la terna [1,3,4], para las bases que se han considerado.

Para la transformación inversa se tiene $M=30$, $s_1=6$, $s_2=10$ y $s_3=15$, mientras que $s_1^{-1}=1$, $s_2^{-1}=4$, $s_3^{-1}=5$ según (2.1), (2.18) y (2.19), con lo que:

$$\begin{aligned} X &= (35 \cdot (1 \cdot 1) \bmod 2 + 14(3 \cdot 4) \bmod 5 + 10 \cdot (4 \cdot 5) \bmod 7) \bmod 70 = \\ &= (35 + 28 + 60) \bmod 70 \\ &= (123) \bmod 70 = 53 \end{aligned}$$

El gran inconveniente de este algoritmo es la necesidad de un acumulador módulo M , siendo generalmente M un entero suficientemente grande como para que sea problemático su tratamiento. Por último, siempre está el recurso, para sistemas de alta velocidad, de almacenar directamente en una ROM la correspondencia entre la representación en residuos y los valores binarios correspondientes.

Una alternativa al CRT es el ε -CRT, propuesto por Griffin *et al.* [GRI88], [GRI89], que lleva a cabo una conversión a binario y el escalado en un solo paso; este método es muy sencillo y transforma la ecuación (2.17) de una conversión entre la n -tupla de residuos y un valor módulo M a un escalado con constante de escalado V , siendo V un número real, y una conversión de la representación en residuos a un valor módulo $\lfloor M/V \rfloor$. En concreto, se puede demostrar [GRI88] que la versión escalada de la ecuación (2.17) es:

$$X^s \equiv \left\lfloor \frac{X}{V} \right\rfloor = \left(\sum_{j=1}^n \lfloor \phi(x_j) / V \rfloor \right) \bmod \lfloor M / V \rfloor \quad (2.20)$$

donde:

$$\phi(x_i) = s_i (x_i s_i^{-1}) \bmod m_i \quad (2.21)$$

Las ecuaciones (2.20) y (2.21) conducen a un esquema sencillo de conversión, ya que:

- se pueden precalcular los valores de $\lfloor \phi(x_i)/V \rfloor$ y almacenarlos en una tabla de consulta; de este modo, sólo son necesarias n tablas para generar los n términos de la ecuación (2.20); cada una de estas tablas es de $2^{k_i} \times k_{\lfloor m/V \rfloor}$ bits, siendo k_i el número de bits necesarios para representar valores módulo m_i , y $k_{\lfloor m/V \rfloor}$ el número de bits necesarios para representar valores módulo $\lfloor M/V \rfloor$;
- por tanto, sólo se requiere un acumulador módulo $\lfloor M/V \rfloor$ para finalizar el cálculo de (2.20); esto supone una gran ventaja frente al algoritmo CRT tradicional de la ecuación (2.17), al pasar de un acumulador de k_M bits a uno de $k_{\lfloor m/V \rfloor}$ bits. Además, dado que V puede ser cualquier valor real, sólo hay que elegir V de forma que $\lfloor M/V \rfloor$ sea una potencia de dos, lo que simplifica de manera sustancial el problema.

La gran ventaja del ε -CRT frente al CRT convencional, además de realizar el escalado al mismo tiempo que la conversión de RNS a binario, es la sustitución de un acumulador módulo M por sumadores convencionales para implementar un acumulador módulo para una potencia de dos. Por otra parte, dado que en la ecuación (2.20) se utiliza la parte entera de (2.21) en lugar del valor exacto, el e-CRT, a través de la ecuación (2.20), introduce un pequeño error al acercarnos al límite del rango dinámico M , con respecto a la ecuación (2.21); sin embargo, esto no constituye inconveniente alguno si consideramos que el rango dinámico seleccionado ha de ser suficiente para no ser rebasado o bien se realizan las operaciones de escalado necesarias para evitar el desbordamiento.

2.4. Módulos aritméticos para el RNS

Tal como se ha visto en las ecuaciones (2.4) del apartado 2.1, la suma y el producto en el RNS se pueden realizar de forma paralela en cada módulo con el resultado de una significativa reducción del rango dinámico respecto del dato completo. La eficacia en velocidad y área de los diseños basados en el RNS dependen en gran medida de la

elección de los circuitos aritméticos. En los siguientes apartados se analizarán diferentes alternativas para la suma y el producto indicando sus ventajas e inconvenientes.

2.4.1. Sumadores en aritmética de residuos

La suma módulo m se puede realizar mediante una gran variedad de diseños surgidos en los últimos 30 años. Los primeros diseños estaban basados en la utilización de puertas lógicas integradas en SSI o MSI [BAN74] ya que eran una alternativa más ventajosa en velocidad y coste que el uso de memorias. Posteriormente se hicieron algunas propuestas basadas en métodos híbridos [SOD86], consistentes en emplear unidades básicas MSI o LSI tales como sumadores completos o PLAs. A mediados de los 80 Bayoumi, Jullien y Miller [BAY87] estudiaron el diseño y la realización práctica de sumadores modulares a nivel de circuito.

Uno de los primeros métodos de suma modular fue propuesto por Banerji [BAN74] y consistía en un sistema de rotación-selección. Este método se fundamenta en que el conjunto $\{0, 1, 2, \dots, m-1\}$ es cíclico respecto de la suma, de tal modo que para sumar dos números $(x + y)$ primero se desplazan los datos hacia la izquierda x veces y después se selecciona el elemento y -ésimo.

Ejemplo. Cálculo de $(4 + 6) \bmod 9$

1° Inicialmente el registro está ordenado en orden decreciente.

8	7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---

2° Desplazamiento de cuatro posiciones a la izquierda.

4	3	2	1	0	8	7	6	5
---	---	---	---	---	---	---	---	---

3° Selección del sexto elemento del registro.

4	3	2	1	0	8	7	6	5
---	---	---	---	---	---	---	---	---

Y en efecto: $|4 + 6|_9 = 1$

La implementación de este mecanismo requiere una red de $n \cdot m^2$ (con $n = \lceil \log_2 m \rceil$) puertas NAND de dos entradas para la rotación y un línea de $n \cdot m$ puertas para la selección. Existen otras alternativas que reducen el número de puertas pero reducen la velocidad de funcionamiento. Este sistema requiere, además, lógica de puesta a cero del registro para cada suma. Una característica interesante de este sistema de cálculo es que la operación de resta se puede implementar de forma muy sencilla sin más que añadir la lógica necesaria para producir el desplazamiento del registro hacia la derecha.

En general, la operación de suma módulo m se puede expresar como:

$$|x + y|_m = \begin{cases} x + y & \text{si } x + y < m \\ x + y - m & \text{si } x + y \geq m \end{cases} \quad (2.22)$$

Existen varias alternativas para la realización de (2.22). La primera (Figura 2.2a) se basa en una tabla de consulta donde están almacenados los valores precalculados de la suma modular de los datos. El tamaño de la tabla ($2^{2n} \times n$ con $n = \lceil \log_2 m \rceil$) limita la utilidad de esta alternativa a valores reducidos del tamaño de palabra, típicamente $n \leq 4$, es decir, para módulos iguales o inferiores a 16. Este método resulta bastante ineficaz por dos motivos:

- esta alternativa aprovecha completamente la memoria ROM únicamente si se verifica que $m=2^n$, ya que, en caso contrario, se desperdician posiciones de memoria;
- teniendo en cuenta la propiedad conmutativa de la suma, la mitad de la tabla contiene los mismos datos que la otra mitad, por lo que la información aparece duplicada.

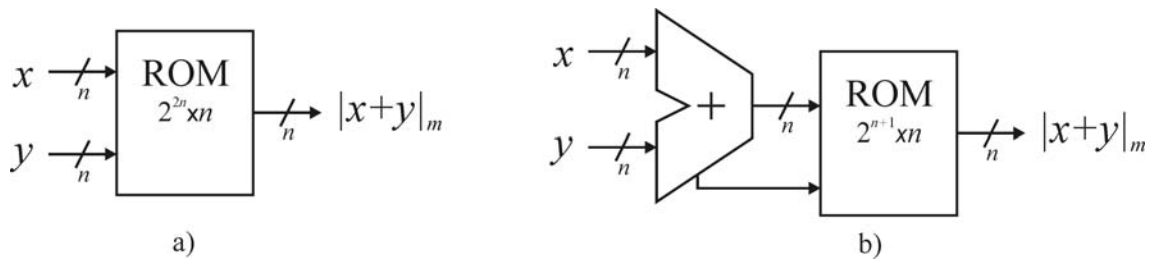


Figura 2.2. Sumadores basados en memorias ROM.

Para módulos mayores se pueden emplear sistemas híbridos (Figura 2.2b) combinando un sumador binario y una memoria ROM; la suma de los datos junto con el bit de acarreo como dirección de la memoria de tamaño $2^{n+1} \times n$. La reducción del tamaño de la tabla permite emplear módulos mayores, y además, se resuelve el problema de la redundancia en los datos contenidos en la tabla.

Una alternativa al uso de memorias ROM es la implementación mediante sumadores binarios (Figura 2.3), de aplicación más general porque permite el uso de módulos mayores. El sistema más general consiste en emplear dos sumadores binarios que implementan las dos posibilidades de (2.22) y un multiplexor que decide entre ambas alternativas. La segunda opción de (2.22) se realiza sumando a $(x + y)$ el complemento de m , es decir $2^n - m$. Un análisis de todos los casos posibles [BAY87, DUG92] muestra que la solución correcta es directamente la suma de los datos $(x + y)$ siempre que no se produzcan desbordamientos en ninguno de los dos sumadores binarios, en cuyo caso hay que restar m a la suma de datos. Por lo tanto, la señal de control del multiplexor es simplemente la función OR de los acarrees de ambos sumadores. Dos situaciones interesantes en la implementación mediante sumadores binarios consisten en que el módulo cumpla que $m = 2^n$ o $m = 2^n - 1$. En el primer caso la suma modular se puede implementar únicamente con un sumador binario en el que se desprecia el acarreo, con lo que se consiguen significativas ventajas en velocidad y reducción de coste (Figura 2.3b). En el segundo caso la suma módulo m se puede obtener sumando directamente los dos datos y, en caso de producirse acarreo, sumando éste al resultado obtenido previamente (Figura 2.3c).

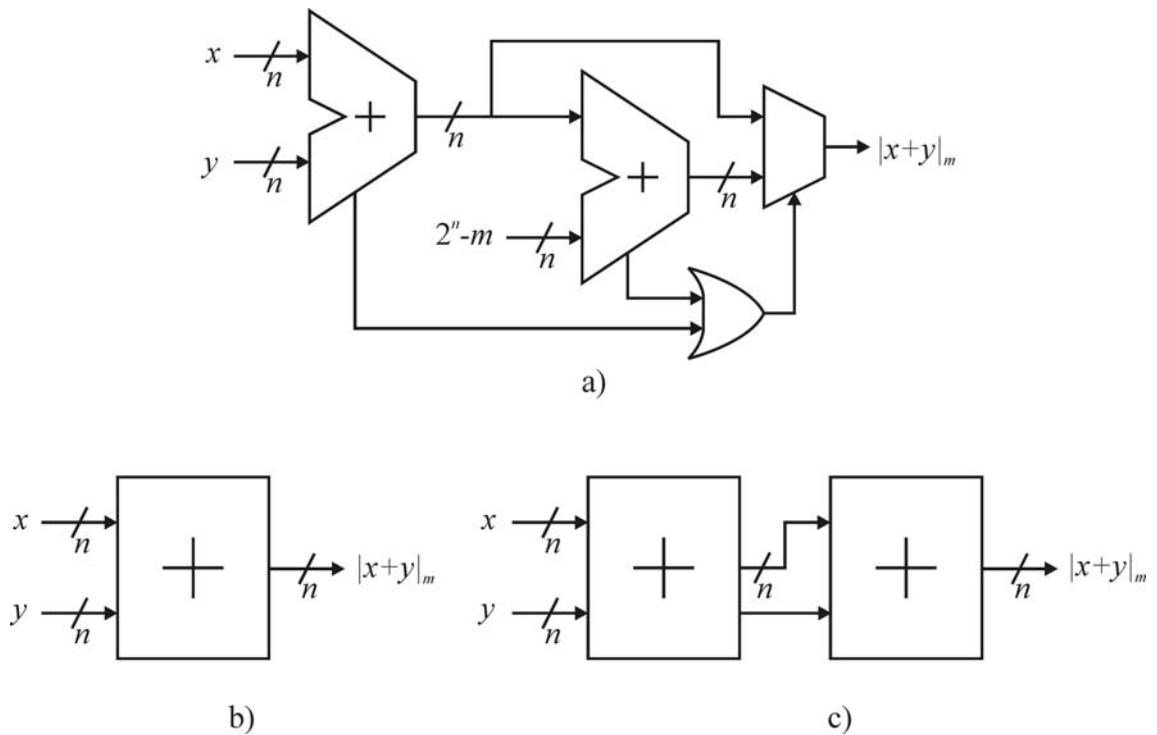


Figura 2.3. Sumadores modulares basados en sumadores binarios.

2.4.2. Multiplicadores en aritmética de residuos

En el RNS la operación de multiplicación se puede realizar mediante una gran cantidad de métodos. La estrategia más simple es el uso de tablas de consulta que contienen los valores precalculados del producto de ambas entradas. El tamaño de la tabla limita la eficacia de esta alternativa a módulos pequeños. Otras alternativas que permiten el uso de módulos mayores son el multiplicador de ley cuadrática y el multiplicador por transformación al dominio de los índices. Por último la mejor opción para módulos medios y grandes son los multiplicadores que no hacen uso de tablas de consulta.

2.4.2.1. Multiplicadores basados en tablas de consulta

Los multiplicadores basados en tablas de consulta consisten en una memoria ROM que almacena los productos de las entradas, que actúan como selectores de direcciones. El tamaño de estas tablas es $2^{2n} \times n$ siendo n la longitud en bits del módulo m . Esta alternativa presenta los mismos inconvenientes de no aprovechamiento de todas las posiciones de la tabla y redundancia en los productos que afectaban a la suma, además de que el tamaño de la tabla crece exponencialmente con la longitud de bits necesarios

para representar el módulo m . Por lo tanto no es una alternativa interesante para módulos superiores a 16 ($n=4$ bits) a menos que se emplee la tabla para realizar el producto por una constante, en cuyo caso el tamaño de la misma es de $2^n \times n$ y el rango de módulos se puede ampliar hasta 64 ($n=8$ bits) de forma eficiente.

2.4.2.2. Multiplicadores de ley cuadrática

Ya se ha visto en apartados anteriores que los sistemas basados en el RNS presentan dificultades para la división y la comparación de magnitud. Esto supone un problema cuando se quiere evitar el desbordamiento de los datos. Existen dos alternativas para evitar el desbordamiento; la ampliación del rango dinámico mediante la inclusión de más módulos y el escalado. La primera de las alternativas eleva el coste del sistema pero, en ocasiones, es preferible a la complejidad que supone la inclusión de una etapa de escalado. El multiplicador de ley cuadrática [POL76, NUS76, JOH80, TAY81] fue propuesto como un método para elevar el rango dinámico de un sistema a un valor que permite reducir el número de operaciones de escalado o incluso eliminarlo por completo.

El multiplicador basado en la ley cuadrática se fundamenta en la igualdad:

$$xy = \left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2 \quad (2.23)$$

donde x e y son números naturales inferiores al módulo considerado m .

Dependiendo de si x e y son números pares o impares se pueden tener dos situaciones:

- x e y son ambos pares o ambos impares, en cuyo caso los cocientes dados por (2.23) proporcionan directamente números enteros.
- x e y son uno par y el otro impar. En este caso las partes fraccionarias de los dos términos de (2.23) son iguales tras elevar al cuadrado y, en la resta de ambos términos, dicha parte fraccionaria se cancela.

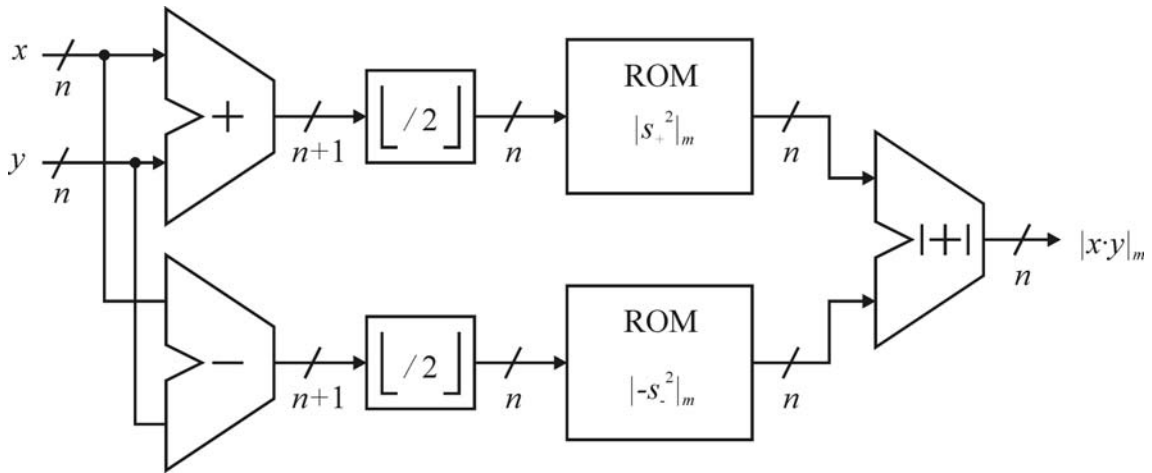


Figura 2.4. Multiplicador módulo m basado en la ley cuadrática.

De este modo, se puede realizar el producto modular según:

$$|xy|_m = \left| \left(\frac{x+y}{2} \right)^2 - \left(\frac{x-y}{2} \right)^2 \right|_m \quad (2.24)$$

Para la realización física de (2.24) se requieren un sumador y un restador, de las salidas de ambos se desprecia el bit menos significativo para realizar la división entera entre dos, dos memorias ROM de $2^n \times n$ ($n = \lceil \log_2(m) \rceil$) donde se almacenan el cuadrado módulo m y el cuadrado cambiado de signo módulo m de las entradas y un sumador modular a la salida de ambas tablas tal como muestra la Figura 2.4 en la que se han definido $s_+ = \lfloor (x+y)/2 \rfloor$ y $s_- = \lfloor (x-y)/2 \rfloor$. El resultado es un multiplicador de alta velocidad que permite el uso de mayores módulos con reducida complejidad. Taylor [TAY81] propuso usar el conjunto de módulos $\{2^n-1, 2^n, 2^n+1\}$ para reducir aun más la complejidad del sumador módulo m de la etapa final del sumador.

2.4.2.3. Multiplicadores basados en aritmética de índices

El multiplicador basado en aritmética de índices [JUL80, RAD91] es una alternativa para la realización de productos cuando los módulos empleados son números primos. Se basa en transformar los factores de un producto al dominio de los índices donde son representados como potencias de base común. De este modo el producto se transforma en la suma de los exponentes o índices. Estos multiplicadores son de complejidad y velocidad similares a los multiplicadores basados en la ley cuadrática.

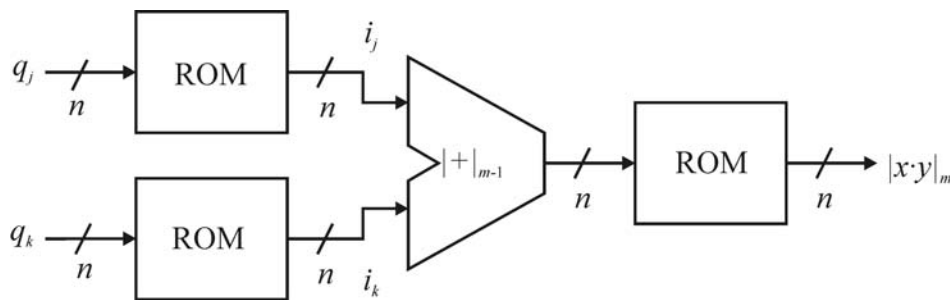


Figura 2.5. Multiplicador basado en aritmética de índices.

El fundamento matemático del multiplicador en el dominio de los índices se asienta en una de las propiedades de los cuerpos de Galois [KRI94, KRI98], que consiste en que todos los elementos no nulos se pueden generar a partir de un elemento generador o raíz g . Por lo tanto, si m es primo se puede establecer un isomorfismo entre el grupo multiplicativo $Q=\{1, 2, \dots, m-1\}$ y el grupo aditivo $I=\{0, 1, \dots, m-2\}$ consistente en que para todo $q \in Q$ existe un $i \in I$ tal que:

$$q = g^i \quad (2.25)$$

De este modo la multiplicación módulo m sobre Q se realiza como una suma módulo $m-1$ sobre I con la equivalencia:

$$|q_j q_k|_m = g^{|i_j + i_k|_{m-1}} \quad (2.26)$$

Con esto, el producto módulo m de dos números se puede obtener mediante la transformación al dominio de los índices de ambos números, la suma módulo $m-1$ de los índices y la transformación inversa del resultado, tal como se muestra en la Figura 2.5. El principal inconveniente de esta alternativa consiste en el caso que uno de los factores sea nulo, ya que la transformación (2.25) no es posible. En este caso se debe añadir la lógica necesaria para la detección del valor nulo a la entrada y para hacer cero el valor de la salida con la latencia adecuada. Existen alternativas que consisten en acelerar el proceso de la suma de índices usando submódulos primos relativos entre sí cuyo producto sea mayor que $2m$ [JUL80]. Otra estrategia consiste en emplear un conjunto de

módulos que constituyan una asociación isomórfica del grupo $GF(m)$ en suma directa de un conjunto de grupos aditivos más pequeños [RAD91]. Con esto se consigue dividir el sumador módulo $m-1$ en un conjunto de sumadores modulares paralelos que mejoran las prestaciones del sistema en velocidad y área.

2.4.2.4. Multiplicadores que no hacen uso de tablas de consulta

Los multiplicadores que se han estudiado en apartados anteriores son eficaces para módulos pequeños o medios. Para módulos pequeños las estructuras basadas en tablas de consulta son las más prácticas [SOD80, JUL80, RAD91, RAD92, DUG94] y para módulos medios las alternativas de ley cuadrática [POL76, NUS76, JOH80, TAY81] y aritmética de índices [JUL80, RAD91] son más eficaces. Para módulos de gran longitud [WAL93, DIC95, ALI91, HIA96, ELL95, WRZ96] se emplean multiplicadores formados por módulos aritméticos, memorias ROM de pequeño tamaño y lógica combinatorial.

Uno de los multiplicadores modulares más eficientes que no hacen uso de tablas de consulta es el propuesto por Hiasat [HIA00]. Está compuesto por dos multiplicadores binarios, tres sumadores binarios y lógica combinatorial de reducido tamaño. Sean x e y dos números enteros de n bits.

$$\begin{aligned} x &= \sum_{j=0}^{n-1} 2^j x_j \\ y &= \sum_{j=0}^{n-1} 2^j y_j \end{aligned} \tag{2.27}$$

con $n = \lceil \log_2(m) \rceil$.

Suponiendo que el módulo m no es potencia de 2, se puede expresar:

$$m = 2^n - \alpha \tag{2.28}$$

donde se verifica que:

$$1 \leq \alpha < 2^{n-1} \tag{2.29}$$

El número α se puede representar con k bits donde $k = \lceil \log_2(\alpha) \rceil$. Es inmediato a partir de (2.29) que $1 \leq k < n-1$. Por otro lado, es obvio que $2^{n-1} < m < 2^n$.

Los datos x e y se multiplican usando un multiplicador binario $n \times n$ dando como resultado z :

$$z = \sum_{i=0}^{2n-2} 2^i z_i \quad (2.30)$$

Si se definen las variables a , b , c y d como:

$$\begin{aligned} a &= \sum_{i=0}^{n-2} 2^i z_i \\ b &= z_{n-1} \\ c &= \sum_{i=n}^{2n-2-k} 2^{i-n} z_i \\ d &= \sum_{i=2n-1-k}^{2n-1} 2^{i-(2n-1-k)} z_i \end{aligned} \quad (2.31)$$

se puede escribir z como:

$$z = a + 2^{n-1} b + 2^n c + 2^{2n-1-k} d \quad (2.32)$$

Según (2.28) se cumple que $|2^n|_m = \alpha$, por lo tanto $|z|_m$ se puede evaluar en términos de la ecuación:

$$|z|_m = \left| a + \alpha c + \left| 2^{2n-1-k} d + 2^{n-1} b \right|_m \right|_m \quad (2.33)$$

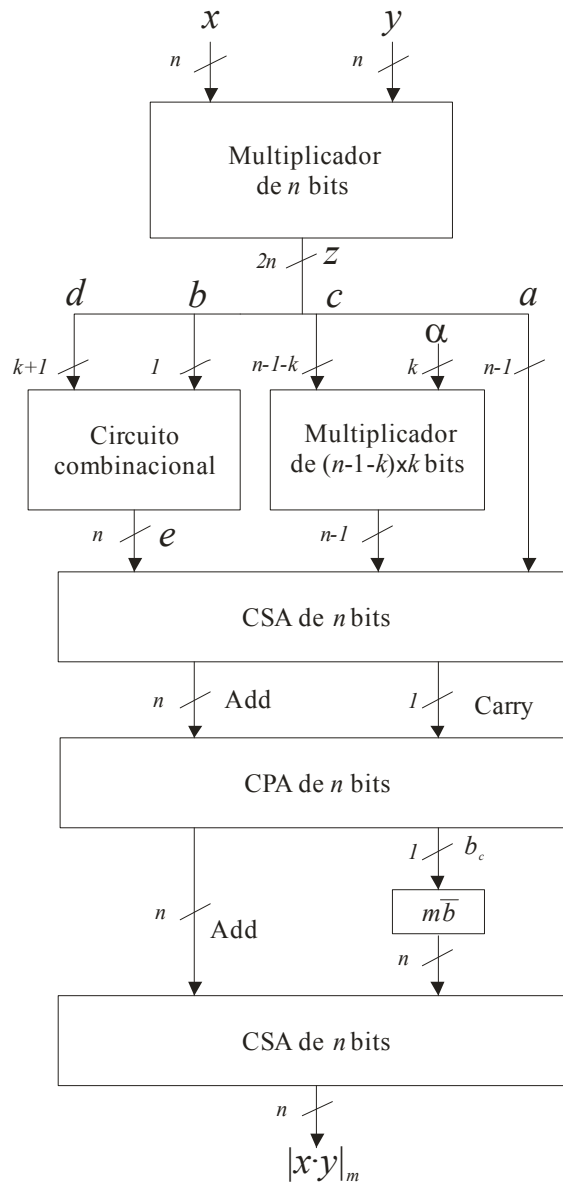


Figura 2.6. Multiplicador módulo m que no hace uso de tablas.

Se puede demostrar que $|a + \alpha c|_m = a + \alpha c$, y por lo tanto el cálculo de este término se ha simplificado. La implementación VLSI del multiplicador que realiza (2.33) se muestra en la Figura 2.6, donde se tienen los siguientes bloques combinatoriales:

- Multiplicador binario de n bits de entrada que realiza el producto $x \cdot y$ y produce la salida z de $2n$ bits.

- Multiplicador binario de $(n-1-k) \times k$ bits que realiza el producto de c por la constante α .
- Circuito combinacional que realiza el cálculo $e = |2^{2^{n-1-k}} d + 2^{n-1} b| - m$ y cuya salida es negativa expresada en complemento a dos.
- Sumador con ahorro de propagación de acarreo, CSA (*Carry Save Adder*), de n bits que realiza la suma de a , αc y e .
- Sumador con propagación de acarreo, CPA (*Carry Propagated Adder*), de n bits que suma la salida más el acarreo del sumador anterior. El bit de acarreo de este sumador b_c indica si el resultado ha sido positivo o negativo.
- Sumador con ahorro de acarreo de n bits que suma m al resultado del sumador anterior en el caso que $b_c=0$ y no suma nada en caso contrario. La salida de este módulo es directamente el producto modular $|x \cdot y|_m$.

2.5. Conclusión

En este capítulo se ha introducido el fundamento teórico de RNS y se han comentado diferentes alternativas de aplicación. Además, se ha revisado el comportamiento del RNS frente a las operaciones aritméticas y, como consecuencia, se puede concluir que su principal campo de aplicación es el procesamiento digital de señales, con transformadas, filtrado, etc. y, en general, procesos en los que sea necesario realizar grandes cantidades de sumas y productos, que es donde el RNS presenta mayores ventajas.

Las principales ventajas del RNS son: ausencia de acarreo entre canales, lo que posibilita dividir un rango dinámico elevado en otros más pequeños y, por lo tanto, más fáciles de diseñar, implementar y que ofrecen mejores prestaciones; también hay que destacar la facilidad para la realización de sumas y productos, operaciones muy frecuentes en algunas aplicaciones y, sobre todo, que la independencia de canales

permite aplicar estrategias de sincronización que mejoran las prestaciones del sistema y hacen más fácil su diseño. Entre los inconvenientes están las dificultades para realizar las operaciones de división, comparación y detección de signo. Sin embargo, estos inconvenientes no suponen restricción alguna para la realización de este trabajo, mientras que las ventajas permiten la aplicación de las estrategias de sincronización que son objeto de estudio en esta memoria.

CAPÍTULO 3

SINCRONIZACIÓN EN CIRCUITOS DIGITALES VLSI

En este capítulo se hace una revisión de diferentes métodos para la sincronización de circuitos digitales VLSI. El estudio se realiza en dos niveles: la propagación de la señal de reloj a lo largo del chip y el comportamiento de los elementos síncronos. Se estudian los problemas que aparecen como consecuencia del aumento en tamaño, complejidad y velocidad de los circuitos integrados, así como los cada vez más exigentes requerimientos de fiabilidad. Adicionalmente, se presentan diferentes alternativas de sincronización que pretenden soslayar dichos problemas, al tiempo que pueden aumentar las prestaciones del sistema.

3.1. Introducción

Antes de abordar el estudio de la propagación y aplicación de la señal de reloj en un sistema digital, que es el principal objetivo del presente capítulo, se van a analizar los principales parámetros que definen los márgenes funcionales del proceso de transmisión y recepción de datos por parte de los elementos síncronos, en relación a la señal de reloj que los controla. Además, se plantearán las principales dificultades que se presentan fundamentalmente en la propagación de la señal de reloj a lo largo de un circuito integrado. El apéndice I contiene una introducción a las diferentes técnicas de generación de a señal de reloj así como las ventajas e inconveniente de cada una de ellas.

3.1.1. Sistemas síncronos

Un sistema síncrono es aquel cuyo funcionamiento está determinado por la presencia de una señal de control (señal de reloj) que determina los instantes en los que el sistema cambia de estado [OKL03, RAB03, XAN09]. El sistema síncrono está concebido como una sucesión de elementos síncronos sobre los que actúa la señal del reloj (*latches* o *flip-flops*), y circuitos combinacionales (Figura 3.1a). La misión de los elementos síncronos es aislar las etapas de lógica combinacional, capturar el dato procesado por la etapa de lógica previa y mantener el valor de dicho dato para que sea evaluado por la lógica de la etapa posterior. La función de los circuitos combinacionales es procesar de

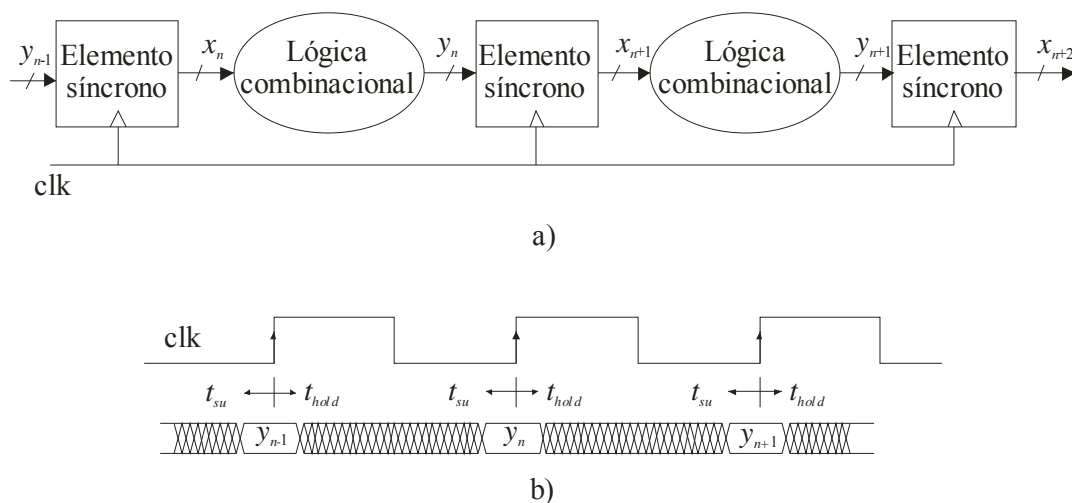


Figura 3.1. Esquema de funcionamiento de un sistema síncrono.

los datos. En el caso de los circuitos controlados por *latches* es la etapa combinacional anterior la que mantiene el dato para ser procesado por la etapa siguiente mientras el *latch* intermedio permite la comunicación entre etapas. Los elementos síncronos más frecuentes en los circuitos actuales son los biestables controlados por flanco y la opción más común la distribución de una única señal de reloj, por lo que las descripciones siguientes se basarán en ese tipo de sistemas. En el apartado 3.3 se describen estrategias de distribución de más de una señal de reloj, y en el 3.4 se describe el funcionamiento de los sistemas basados en el nivel de la señal del reloj.

La señal del reloj actúa sobre los elementos síncronos marcando el instante en el que la información pasa de una etapa a la siguiente, por lo que tanto la frecuencia como la precisión y la calidad de la señal de reloj son factores críticos para el funcionamiento y el rendimiento del sistema. El tiempo entre dos flancos consecutivos debe ser el suficiente para que el dato anterior (y_{n-1}) sea transmitido por el elemento síncrono (x_n), sea procesado por la lógica combinacional (y_n) y permanezca de forma estable y correcta el tiempo suficiente a la entrada del elemento síncrono siguiente (*setup time*, t_{su}) previamente a la llegada del flanco de la señal de reloj. Por otra parte el tiempo de procesamiento de la lógica entre dos elementos síncronos consecutivos debe ser suficiente como para no corromper un dato con el siguiente ya que, tras el flanco de reloj, un dato debe permanecer estable un tiempo (*hold time*, t_{hold}) para poder ser capturado correctamente (Figura 3.1b).

La primera condición, relativa al tiempo entre flancos, establece una restricción para el periodo mínimo (máxima frecuencia) a la que puede operar un sistema. El periodo debe ser lo bastante prolongado para permitir el correcto funcionamiento de la etapa más lenta (*critical path*). De este modo se puede definir un límite mínimo para el periodo de la señal de reloj requerido:

$$T \geq T_{m\acute{a}x} = t_{CQ} + t_D + t_{int} + t_{su} \quad (3.1)$$

donde $T_{m\acute{a}x}$ es el tiempo máximo necesario desde que la señal de reloj llega al elemento registro i hasta que el dato procesado sea almacenado de forma estable en el registro $i+1$, y se puede determinar como la suma del tiempo máximo necesario para que el dato sea propagado por el registro inicial desde que éste recibe la señal de reloj (t_{CQ}), más el

tiempo necesario para que el dato sea procesado por la lógica en la etapa más lenta (t_D), más el tiempo necesario para la propagación del dato y del resultado a través de las interconexiones (t_{int}), más el tiempo de *set-up* (t_{su}).

Por otro lado, la segunda condición, que asegura el tiempo necesario para que la lógica procese un dato, impone que el retardo de dicho procesamiento sea inferior al de *hold time* del biestable.

$$T_{min} \geq T_{hold} \quad (3.2)$$

donde T_{min} es el tiempo de procesamiento de la etapa más rápida.

Las expresiones (3.1) y (3.2) establecen las dos restricciones en cuanto a tiempos de procesamiento que cualquier sistema síncrono debe respetar en todos y cada uno de los tramos de las diferentes rutas de procesamiento. Tal y como están planteadas ambas expresiones se ha asumido que la señal de reloj es exacta y precisa; sin embargo hay una gran variedad de causas que pueden alterarla. Las expresiones (3.1) y (3.2) deben ser reformuladas para tener en cuenta las incertidumbres que pueden afectar a la señal de reloj. A continuación se introducen las causas más relevantes que provocan estas incertidumbres.

3.1.2. Problemas que afectan a la señal de reloj

La distribución de una señal de reloj a lo largo de un circuito de gran tamaño es una tarea difícil y problemática. El periodo de la señal de reloj generalmente está ajustado a valores mínimos para obtener una velocidad de funcionamiento lo más elevada posible. En estas condiciones, pequeños retrasos en la distribución de la señal pueden ocasionar errores de funcionamiento. En un sistema síncrono, la señal de reloj controla la transferencia de datos entre los diferentes módulos de procesamiento y define los instantes en los que es posible el cambio de estado. Las señales de reloj son básicamente señales de control y, debido a esta naturaleza, presentan una serie de características que hacen que su generación y distribución sean especialmente críticas para el rendimiento del sistema. Idealmente, en un sistema síncrono, la señal de reloj debe llegar a todos los elementos del circuito al mismo tiempo. Esto tiene como resultado que las señales de reloj:

- sufran niveles elevados de *fan-out*, es decir, la señal que se propaga por una línea debe actuar sobre una cantidad elevada de dispositivos;
- deban propagarse a lo largo de largas líneas de interconexión, que pueden incluso abarcar todo el chip;
- han de conmutar a la máxima velocidad permitida por la tecnología con la que está fabricado el circuito, ya que son las señales que más rápidamente deben conmutar y gobiernan el funcionamiento del resto del sistema;
- requieran una gran calidad en su perfil, porque este define los instantes en los que los bloques operativos realizan el procesamiento de los datos o producen sus resultados.

En los casos reales la señal de reloj ha de propagarse a lo largo de líneas conductoras, metálicas o polisilicio (que inherentemente llevan asociada una resistencia y una capacidad), elementos lógicos de distribución de la señal (*buffers*) e interconexiones. Todos estos elementos son causa de la degradación en el perfil y de la aparición de retardos en la distribución de la señal. Los principales problemas que puede plantearse son:

- que entre el instante en que ocurra un flanco, de subida, por ejemplo, y el instante en que dicho flanco llegue a su destino, transcurra un intervalo de tiempo apreciable (*delay* o retardo);
- que en un instante determinado, la señal de reloj adquiera valores diferentes en diferentes partes del circuito (*skew*);
- que no esté perfectamente definido el instante en el que se produce un flanco de subida o bajada, produciéndose un intervalo de incertidumbre (*clock jitter*);

- que los flancos sean atenuados y presenten un aspecto suavizado, e incluso se produzcan pérdidas apreciables de tensión (atenuación);
- que el circuito no sea capaz de establecer un nivel de referencia de tierra preciso y se produzcan variaciones en la tensión de referencia debido a una deficiente eliminación de los portadores de carga (*ground bounce*);
- que la salida de un biestable alcance la entrada del siguiente antes de que llegue la señal de reloj a este último, con lo que dicha señal atravesará dos elementos de almacenamiento síncronos en un ciclo de reloj (carrera o *double clocking*);
- que el retardo entre la salida de un dato y su llegada al siguiente biestable sea mayor que el periodo del reloj (*zero clocking*).

La disminución de las dimensiones características de los circuitos integrados viene acompañada de un aumento considerable de la frecuencia de reloj, por lo que los límites permisibles para el retardo, el *skew*, y el *jitter* se han visto reducidos críticamente. La mayor capacidad de integración tiene como resultado directo que la señal de reloj deba seguir diferentes caminos en los que debe atravesar gran cantidad de dispositivos y conexiones, por lo que algunas rutas sufrirán mayores retardos que otras. Es más, el tamaño cada vez mayor de los circuitos integrados requiere líneas de distribución de reloj de mayor longitud, que pueden producir retardos comparables al periodo de la señal.

Los problemas planteados en la distribución de la señal de reloj se pueden resolver mediante diferentes estrategias. La calidad de la señal puede ser restaurada mediante el uso de *buffers* aunque, como contrapartida, esto puede suponer el aumento del retardo en las líneas y la cantidad de dispositivos del sistema, con el consiguiente incremento del consumo. Los problemas de *skew* se pueden resolver mediante el análisis en *RC-tress* [FRI01], que se verá más adelante y que consiste en concebir las líneas de distribución de reloj como asociaciones de resistencias y capacidades, y los bloques operativos de destino como capacidades terminales. De esta manera, se puede estimar el retardo a lo largo de todas las líneas, e incluir elementos de retardo

adicionales de manera que se minimice el *skew*. Esta técnica aplica un criterio de peor caso, ya que aumenta el retardo en líneas donde éste es menor, por lo que si bien los problemas de *skew* son minimizados, aumenta el número de dispositivos y, por lo tanto, el área utilizada y el consumo. Otra alternativa de minimización del *skew* es la técnica de *H-trees*, que consiste en distribuir la señal de reloj de manera simétrica a lo largo del circuito en forma de H. Así se asegura que el retardo a lo largo de cada línea es el mismo, aunque esta técnica es plenamente aplicable y se obtienen mejores resultados en circuitos formados por estructuras regulares de módulos operativos. Una variante frecuente de la distribución en *H-trees* es la estructura en *X-trees* que, se fundamenta en el mismo principio que la anterior pero define las líneas de distribución de reloj oblicuamente en forma de X. Otras estrategias de distribución consisten en insertar líneas principales (*central spine*) de distribución del reloj de las que parten diferentes líneas secundarias, o bien en crear una red en rejilla (*grid*) de modo que se consigue una disminución de la resistividad y no se limita la propagación a una única línea. Con esto se consigue una distribución de la señal del reloj uniforme en un área.

En cuanto a las incertidumbres dinámicas asociadas al *jitter* los diseñadores deben tener en cuenta las dos principales causas que originan este efecto: las fluctuaciones en la tensión de alimentación y el ruido en el generador de la señal de reloj. Las inestabilidades de la tensión de alimentación se deben principalmente a la conmutación de un elevado número de dispositivos en las proximidades de los elementos responsables de la distribución de la señal de reloj como son los *buffers*. En cuanto al ruido en el generador de la señal de reloj, éste se produce principalmente en el oscilador controlado por tensión (*VCO Voltage Controlled Oscillator*) y también hay una componente importante en el ruido externo.

3.2. Distribución de la señal de reloj

Una vez generada, la señal de reloj debe llegar a todos los elementos síncronos en el instante preciso y con la calidad adecuada. En sistemas con un elevado número de dispositivos, esta tarea se complica a medida que la frecuencia del reloj aumenta. Generalmente se usan *buffers* que suministran la corriente necesaria para soportar el elevado *fan-out* que soporta la señal de reloj y, además, actúan como restauradores del

perfil que va degradándose a medida que la señal de reloj atraviesa largas interconexiones. Si la resistencia de la interconexión entre el *buffer* y el generador de reloj es menor que la resistencia de salida del *buffer* [FRI01], basta con un único *buffer* para sincronizar el circuito. Esta estrategia es adecuada cuando las líneas de distribución son enteramente metálicas, lo que conlleva que los efectos de carga sean menores. Es necesario que la corriente suministrada por el *buffer* a la línea de distribución de reloj sea suficiente como para superar por una parte los efectos capacitivos debidos a las líneas y a los dispositivos que se controlan y, por otra, el *fan-out*. En circuitos mayores se suelen usar más *buffers*, convenientemente distribuidos, para asegurar que la señal de reloj tiene la calidad apropiada para la sincronización del circuito. En este apartado se van a analizar, en primer lugar, las dificultades que supone la distribución de la señal de reloj y, después, diferentes metodologías de distribución que tratan de paliar dichas dificultades. Ya al final del apartado se hace una breve revisión de estrategias de sincronización basadas en técnicas no convencionales que emplean pulsos de luz, efectos de oscilación resonante en anillos o incluso la supresión o alteración de la propia señal de reloj.

3.2.1. Dificultades en la distribución de la señal de reloj

Los principales problemas derivados de la propagación de la señal de reloj consisten en pérdidas de sincronía, es decir, incertidumbres en el instante que ocurre el flanco de la señal de reloj. Estas incertidumbres se producen como consecuencia de los diferentes retardos que experimentan las diferentes señales de reloj a través de diferentes caminos o bien debido a variaciones del tensión, de proceso o de temperatura.

3.2.1.1. Retardo (*delay*)

El retardo de la señal de reloj está originado por grandes longitudes de propagación a lo largo del circuito y por las interconexiones entre las líneas de distribución y los dispositivos que controla la señal de reloj. Este efecto se hace patente cuando los tiempos de conmutación de los dispositivos son inferiores a 1ns y las dimensiones mínimas de los dispositivos se reducen por debajo de 1 μ m. El retardo en un sistema se puede analizar desde dos perspectivas [ZHO91]: la más tradicional del retardo debido a efectos capacitivos y resistivos, que modela los circuitos como asociaciones serie de resistencias y capacidades, y la perspectiva del retardo de líneas de transmisión [DEU95, RES98, ISM00], que incluye los efectos inductivos de las líneas y que tiene su

justificación en las elevadas frecuencias de operación de los circuitos integrados actuales junto con las cada vez más largas líneas de distribución del reloj.

La evaluación y el control del retardo en la propagación de las líneas de reloj son problemas de especial relevancia en el diseño de circuitos VLSI. Este problema es de difícil resolución debido fundamentalmente a dos motivos: en primer lugar, los simuladores de circuitos permiten evaluar un retardo de configuraciones específicas, pero la obtención de reglas generales para relacionar la geometría de las líneas de distribución y los dispositivos con el comportamiento eléctrico requieren un tratamiento analítico. Por otro lado, este tratamiento involucra complejos cálculos matemáticos y el tratamiento real del comportamiento eléctrico de los elementos que intervienen en la distribución de las señales supone la aplicación de modelos físicos de dispositivos y materiales que no admiten un tratamiento analítico sencillo. Se establece, pues, un compromiso entre la exactitud y la facilidad del tratamiento del modelo. Tradicionalmente, las interconexiones entre dispositivos se han modelado como asociaciones de una resistencia y una capacidad (líneas RC). Sin embargo, cuando los tiempos de conmutación se reducen, este modelo resulta poco adecuado, debiendo aplicarse un tratamiento de líneas de transmisión. El modelo tradicional de línea de transmisión alimentado por una fuente de tensión ideal no puede ser empleado al estar la línea alimentada por transistores con características I-V no lineales.

Zhou *et al.* [ZHO91] han derivado expresiones analíticas para el tratamiento desde los dos modelos (RC y línea de transmisión) y han evaluado el efecto de la disminución del tamaño en los dispositivos. Además, establecen relaciones entre retardo y parámetros tecnológicos como resistencia, capacidad, inductancia, longitud y anchura de la línea y capacidad de carga de la misma. Se revisan a continuación algunas de las relaciones entre retardo y características físicas del circuito. Un factor que influye de forma determinante en el retardo es la cantidad de corriente que los dispositivos pueden suministrar a la línea, si bien este efecto no es relevante cuando la capacidad de carga es sensiblemente inferior a la capacidad de la interconexión. En líneas metálicas la relación entre la longitud de la línea y el retardo es prácticamente lineal en las longitudes consideradas (<5cm) y depende básicamente de la capacidad de la línea y de la resistencia de salida del dispositivo que suministra la corriente. En líneas de polisilicio la resistencia de la línea juega un papel importante y la relación retardo-longitud es

marcadamente no lineal. Las dimensiones de la línea también juegan un papel importante en la generación de retardo. Por una parte, desde un punto de vista de análisis RC, líneas excesivamente anchas tienen como resultado un aumento en la capacidad asociada, con lo que aumenta el retardo; sin embargo, mediante un análisis de líneas de transmisión, para una longitud dada de línea existe una anchura crítica por debajo de la cual el retardo de la línea se ve incrementado enormemente.

Celik y Pileggi [CEL99] realizan otra aproximación al problema del retardo desde el dominio de la frecuencia. Como ya se ha comentado, cuando las frecuencias alcanzan el rango de los GHz, es preciso un análisis de líneas de transmisión para analizar el comportamiento de las líneas de interconexión. Esta necesidad se debe fundamentalmente a dos motivos:

- en primer lugar es necesario tener en cuenta el efecto de atenuación de la señal, ya que las líneas presentan un comportamiento de filtro paso-baja. Es por este motivo que, cuando una señal cuadrada sufre atenuación debido a la naturaleza de la interconexión, se deteriora su perfil y la señal se aproxima a una función senoidal como consecuencia de la pérdida de componentes en frecuencia de orden superior;
- por otro lado, en el rango de altas frecuencias, las dimensiones del chip puedan dar lugar a desfases del orden de 2π , por lo que el análisis en largos periodos de tiempo deja de ser útil.

Celik y Pileggi proponen controlar en primer lugar la atenuación de la señal y después ajustar el retardo una vez la calidad de la señal está asegurada; de esta manera se incide principalmente en los puntos donde la atenuación es mayor, que son aquéllos donde el retardo se puede hacer más significativo. Aswatha y Basavaraju [ASW08] han presentado un modelo analítico RLC de interconexiones que permiten un cálculo más exacto de los retardos una vez se ha establecido la red de distribución del reloj mediante análisis de corriente. Las ecuaciones obtenidas también permiten el cálculo de la anchura óptima de las líneas de interconexión para la minimización del consumo según un análisis de señales de tensión. Además las ecuaciones permiten calcular con un error inferior al 0.05% respecto a soluciones numéricas, la ubicación y dimensiones de

buffers repetidores de modo que, teniendo en cuenta los efectos inductivos, se minimizan el retardo, el área y el consumo de dichos repetidores.

Vasseghi *et al.* [VAS96] diseñaron una red de distribución del reloj en la que sólo el primer nivel fue tratado como una línea de transmisión. Mediante esta técnica se pueden extraer y controlar fácilmente los parámetros de la línea de transmisión pero los requisitos de interconexión con las líneas de alimentación y tierra afectan significativamente al enrutado de la red de distribución cuando solamente se dispone de dos capas de metal. Gieseke *et al.* [GIE97] solventaron la dificultad en otro diseño añadiendo dos capas adicionales de metal dedicadas exclusivamente a alimentación y tierra; pero esta estrategia encarece el coste del chip a no ser que previamente fuesen requeridas para la red de distribución de energía a lo largo del circuito.

3.2.1.2. Pérdida de sincronía (*skew*)

El *skew* consiste en las variaciones en el momento en el que ocurre el flanco del reloj en dos lugares diferentes del circuito o entre dos señales de reloj derivadas de un mismo punto. Estas variaciones son constantes o varían muy lentamente con el tiempo, por lo que también se conocen como variaciones o incertidumbres estáticas. El problema del *skew* ha sido ampliamente tratado en la bibliografía desde la década de los 80 hasta la actualidad [NIC12, KIM10, ESM10, HON09, YAM04, KHL04, BLA92, NEK98, BAK86, FIS85].

El *skew* provocará problemas en la distribución del reloj si se verifica la expresión [GRO94]:

$$\frac{D}{v} > \frac{k}{f_{ap}} \quad (3.3)$$

donde k representa los requerimientos de *skew* en términos de tanto por uno del periodo de la señal de reloj ($k < 0.10$ típicamente), D es la dimensión típica del sistema, v es la velocidad de propagación nominal de la señal y f_{ap} es la frecuencia de reloj aplicada. Esta expresión representa, en términos de tamaño y frecuencia, el porcentaje de tiempo (k) del periodo de la señal de reloj ($T = f_{ap}^{-1}$) que una señal tardaría en cruzar todo el chip ($t_c = D/v$). Expresada en otros términos (3.3) quedaría:

$$\frac{t_c}{T} > k \quad (3.4)$$

lo cual indica que el *skew* podrá ocasionar problemas si el tiempo que la señal de reloj requiere para atravesar el chip supone más del 10% del periodo de la señal de reloj.

Como ya se ha visto, los sistemas síncronos consisten en una serie de registros separados por bloques combinatoriales que realizan el procesamiento de los datos. Los sistemas síncronos deben cumplir (3.1) y (3.2) en cuanto a los tiempos de procesamiento de cada bloque combinatorial entre dos elementos de almacenamiento. La influencia del *skew* en estas restricciones se puede expresar sumando un término T_{skew} que representa la diferencia de tiempo que transcurre desde que la señal de reloj

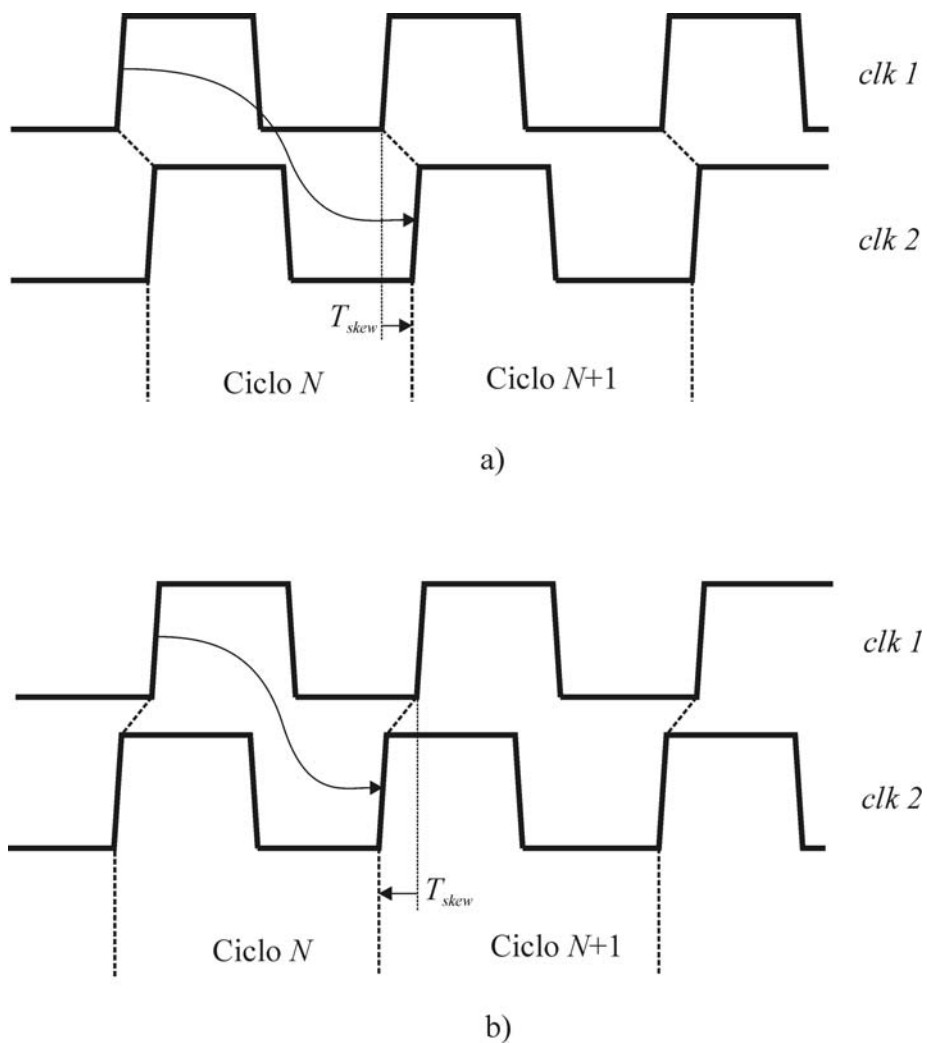


Figura 3.2. *Skew* negativo (a) y positivo (b).

llega al registro i hasta que llega al registro $i+1$ y representa la incertidumbre estática o constante en la señal de reloj. Las expresiones quedan:

$$T \geq T_{m\acute{a}x} = T_{PD} + T_{skew} \tag{3.5}$$

$$T_{min} \geq T_{hold} + T_{skew} \tag{3.6}$$

donde:

$$T_{PD} = T_{C-Q} + T_D + T_{int} + T_{su} \tag{3.7}$$

El *skew* se puede considerar negativo o positivo según la señal del reloj llegue antes al registro anterior o al posterior respectivamente, tal como queda indicado en la Figura 3.2. Sin embargo, este criterio [FRI01] no es absoluto y algunos autores [HAT88, RAB03] consideran el signo del *skew* al contrario. El efecto de un valor negativo de T_{skew} consiste en que el registro final recibe el flanco de reloj después que el inicial y, por lo tanto, el dato tiene más tiempo para ser transmitido y procesado (Figura 3.2a). Este tiempo adicional (T_{skew}) se puede restar del tiempo necesario para el procesamiento y transmisión del dato (T_{PD}) en (3.5) y de ahí el criterio de considerarlo negativo. Con esto se obtiene una disminución del periodo mínimo de funcionamiento (T_{min}) y, por lo tanto, un incremento de la frecuencia a la que opera el circuito. El resultado anterior se puede utilizar para mejorar el rendimiento del sistema, pero existe una limitación; si el valor de T_{skew} es mayor que el tiempo necesario para que el registro i cargue el dato D_{i+1} y este dato sea procesado por la lógica combinacional, el dato D_i

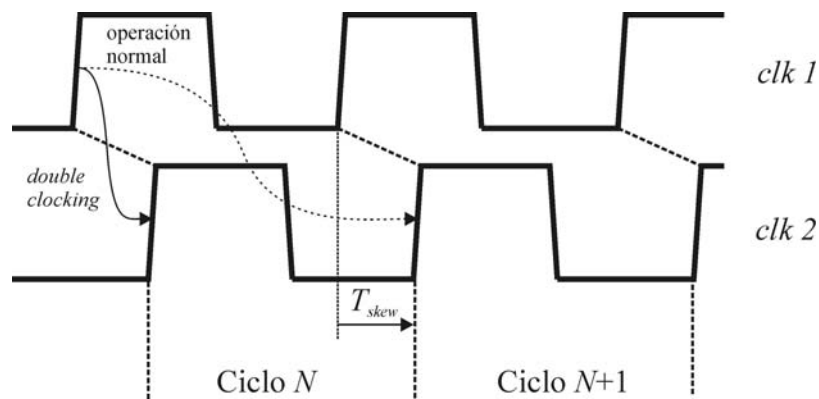


Figura 3.3. Problema de carreras o *double clocking* debido a un valor del *skew* negativo elevado.

puede ser sobrescrito en el registro posterior (Figura 3.3). Esto resultaría en una violación de la restricción (3.6) y provocaría un problema de carreras (*double clocking*). Por otra parte, un valor positivo de T_{skew} (Figura 3.2b) tiene como resultado el efecto contrario, un incremento en el periodo de la señal de reloj. Es el tiempo adicional que hay que añadir a $T_{m\acute{a}x}$ en (3.5) para obtener un valor suficiente del periodo de la señal de reloj para que el procesamiento de los datos sea el correcto. El resultado es una disminución de la frecuencia máxima de funcionamiento.

Las causas que provocan la aparición del *skew* en un sistema son muy variadas y abarcan desde las características propias del diseño del circuito tales como la topología de la red de distribución de la señal del reloj o las diferencias en las capacidades intermedias o terminales en cada línea de distribución, hasta las variaciones de proceso en la fabricación del chip o las fluctuaciones en la tensión de alimentación o en la temperatura. Desde el punto de vista del diseño del sistema, el *skew* es un resultado de las diferencias en las longitudes físicas de las rutas de distribución del reloj o de los diferentes *buffers* e interconexiones que dicha señal debe atravesar. Estas diferencias pueden estar originadas por:

- las restricciones en cuanto a la ubicación de los bloques funcionales del sistema;
- un diseño no riguroso de la red de distribución de la señal de reloj;
- un diseño intencionado con objeto de proporcionar más tiempo a rutas de procesamiento más lentas (*cycle borrowing*) o para conseguir una distribución del funcionamiento, y por lo tanto del consumo, más uniforme para evitar picos de corriente elevados.

En cuanto a las variaciones del proceso de fabricación, son causa del *skew* las diferencias en los parámetros de las interconexiones (dimensiones, resistividad, constante dieléctrica, capacidades, etc.) y de los elementos activos (tensión umbral de los dispositivos MOS, movilidades de los portadores en el canal, etc.). Más del 55% del valor del *skew* es debido a los dispositivos y a la topología de la red de distribución [GEA98], aproximadamente el 25% se debe a las fluctuaciones en la fuentes de

alimentación, en torno al 15% está originado por las variaciones en la carga y una pequeña fracción se debe a variaciones en la temperatura. El *skew* en diferentes sistemas que operan por encima del rango de los centenares de MHz hasta las decenas de GHz oscila entre el 2 y el 9% [CLA04, GOL06, DOR07, SAK07], siendo el promedio aproximadamente un 4.5%.

La tendencia constante en la reducción del tamaño de los sistemas junto con el mantenimiento de frecuencias en el rango de los GHz, ha provocado que el *skew* sea un efecto cada vez más significativo, por lo que los diseñadores deben poner especial énfasis en su control y limitación. Las principales estrategias de minimización del *skew* son las siguientes:

1. topologías de distribución de la señal de reloj que compensen el *skew*, mediante redes de distribución simétricas (*H-trees*, *binary-trees* y distribución en rejilla), líneas centrales de distribución [RAM89, JAC90, CON93, RAM94] o distribuciones basadas en *rotary clocking* [VEN07, HON09];
2. diseños más minuciosos en cuanto a ubicación de líneas y con mayor control de las dimensiones y ubicación de los dispositivos [NEK97, NEK98, NEK01, YI01, LIU01, VEL04];
3. incorporación a los chips de subsistemas adaptativos eficientes de test y/o compensación del *skew* durante el funcionamiento [MUE08, TAK08, XU09].

Las dos primeras estrategias tratan de compensar total o parcialmente el *skew* previamente a la implementación física, sin embargo un análisis exacto requiere una gran inversión en tiempo de computación. Como alternativa a la simulación eléctrica se han desarrollado diferentes modelos analíticos para la determinación del *skew*. Kugelmass y Steiglitz [KUG88] propusieron un modelo probabilístico que analiza el retardo de la señal de reloj de cada línea como la suma de los retardos de cada segmento de la línea y asume que cada segmento presenta un retardo según una ley probabilística. Asumiendo la independencia de cada segmento se obtiene que el *skew* sigue un modelo de distribución normal. Otro enfoque es el determinista propuesto por Afghahi y Svenson [AFG89], que representan el *skew* como el resultado de la dispersión de los

parámetros físicos del circuito (dimensiones geométricas) y de los parámetros de proceso. Jiang y Horiguchi [JIA01] proponen una nueva aproximación, que consiste en considerar que las rutas de distribución de la señal de reloj no tienen por qué ser idénticas. Se considera la correlación de los retardos debidos a solapamientos entre líneas de distribución. Este modelo es más general ya que en sistemas reales se pueden dar solapamientos en las rutas de distribución de la señal de reloj. La tercera estrategia para corregir el *skew* consiste en diseñar sistemas capaces de realizar compensaciones adaptativas durante el funcionamiento del circuito, lo que es más ventajoso en grandes volúmenes de fabricación.

3.2.1.3. Dispersión en el flanco (*clock jitter*)

Otro problema asociado a la generación y distribución de la señal de reloj es el *clock jitter*, que consiste en variaciones aleatorias en el instante en el que se produce la transición en la señal de reloj (Figura 3.4) [JAN12, SHI10, JIA10, YAM01, HEY00, HER99]. Estas variaciones ocurren rápidamente en el tiempo por lo que el *jitter* se denomina también variaciones o incertidumbres dinámicas. *Jitter* es la denominación de estas incertidumbres en el dominio del tiempo y ruido de fase en el dominio de la frecuencia. Existen diversas definiciones [YAM01, XAN09] de este efecto:

- Dispersión en la fase (*phase jitter*), definido como la diferencia entre el instante en que ocurre la señal de reloj y el instante en que teóricamente debería haber ocurrido. Idealmente las transiciones de una señal cuadrada ocurren en los instantes $0, T/2, T, 3T/2$, etc. Sin embargo, el flanco puede ocurrir con una cierta incertidumbre anterior o posterior al instante teórico. El

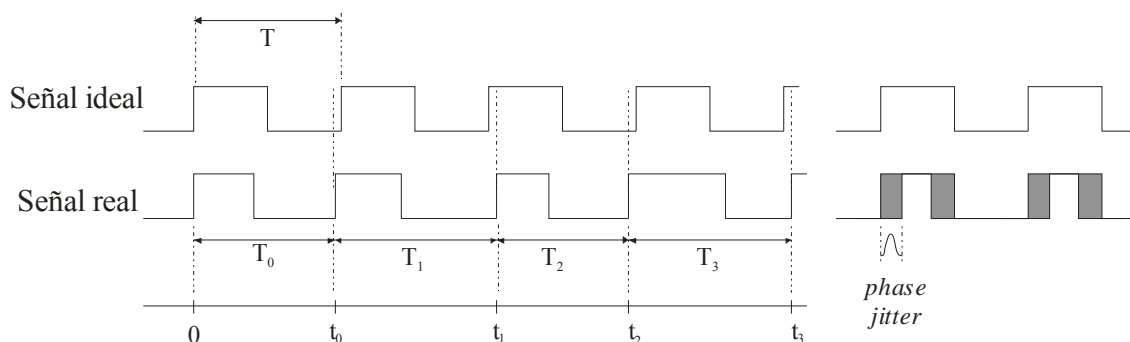


Figura 3.4. Definición de *jitter*.

phase jitter se define como:

$$\Phi_{ph-j}[n] = \Phi_{ph-j}(nT) = t_n - nT, \quad \left(n = 0, \frac{1}{2}, 1, \frac{3}{2}, \dots \right) \quad (3.8)$$

donde t_n es el instante en el que ocurre el flanco n -ésimo y T es el periodo de la señal de reloj.

- Dispersión en el periodo (*period jitter*), definido como la diferencia entre el periodo nominal de un sistema y el promedio del periodo de la señal de reloj.

$$\Phi_{per-j}[n] = (t_n - t_{n-1}) - T = T_n - T \quad (3.9)$$

que puede expresarse como:

$$\Phi_{per-j}[n] = \Phi_{ph-j}[n] - \Phi_{ph-j}[n-1] \quad (3.10)$$

Nótese que t_n es el instante absoluto en el que ocurre el flanco, por lo que su valor acumula las variaciones de los n flancos anteriores.

- Dispersión en el periodo ciclo a ciclo (*cycle-to-cycle jitter*), que indica la diferencia entre sucesivos periodos de la señal. Se puede expresar como:

$$\Phi_{c2c-j}[n] = T_n - T_{n-1} \quad (3.11)$$

o bien como:

$$\Phi_{c2c-j}[n] = \Phi_{per-j}[n] - \Phi_{per-j}[n-1] \quad (3.12)$$

El primer tipo (*phase jitter*) es la diferencia entre los valores propios de la señal de reloj real y la ideal, por lo que representa el valor básico conceptual del *jitter*. Este tipo de jitter es de interés en sistemas en los que la fase de la señal de reloj es crucial, como pueden ser los sistemas de transmisión de datos en serie. El segundo tipo (*period jitter*) se deriva del primero mediante la diferencia y representa las variaciones del periodo de la señal de reloj. Es por lo tanto un fenómeno de alta frecuencia. Es un parámetro de interés en los sistemas síncronos en general en los que el tiempo de

procesamiento de la ruta crítica está ajustado a la frecuencia de funcionamiento del sistema y variaciones en ésta pueden provocar errores de funcionamiento. El tercer tipo de *jitter* es la primera diferencia del *period jitter* y la segunda diferencia del *phase jitter*, por lo que es un fenómeno de muy alta frecuencia.

Elevados valores en el *clock jitter* llevan al funcionamiento del circuito a frecuencias inferiores de las posibles debido a la necesidad de establecer márgenes de funcionamiento seguros; en caso contrario se pueden producir errores de sincronización dentro del mismo chip y entre los diferentes chips que forman un sistema. El *jitter* suele suponer hasta el 5.5% [BOE98, BOE99, HOF00, KUR01, RES02] del ciclo total del sistema y tiene su origen en tres causas:

- Variaciones en el voltaje de alimentación debidas a caídas de tensión o fluctuaciones dinámicas del mismo.
- Fluctuaciones producidas en el propio generador del reloj, predominantemente en el VCO (ruido intrínseco) y en la señal de entrada (ruido extrínseco).
- Gradientes de temperatura en el chip debidos a variaciones de la actividad de diferentes zonas del mismo.

Las variaciones de voltaje globales y el ruido en el generador del reloj contribuyen al *jitter* de todo el sistema y afectan a las restricciones de *set-up* modificando la frecuencia de funcionamiento. El *jitter* debido a variaciones térmicas es el menos significativo puesto que dichas variaciones son muy lentas en comparación con el periodo de la señal de reloj.

El efecto dominante en estas incertidumbres dinámicas son las variaciones en el voltaje de alimentación y se deben principalmente a efectos de conmutación de diferentes zonas que afectan a la red de distribución del reloj de manera no uniforme. Otra causa que afecta a los niveles de la tensión de referencia es el ruido alimentación/tierra (*P/G noise*), consistente en caídas de tensión-corriente originadas por el carácter resistivo e inductivo de las líneas de alimentación y por las elevadas demandas de corriente debidas a la conmutación simultánea de un elevado número de

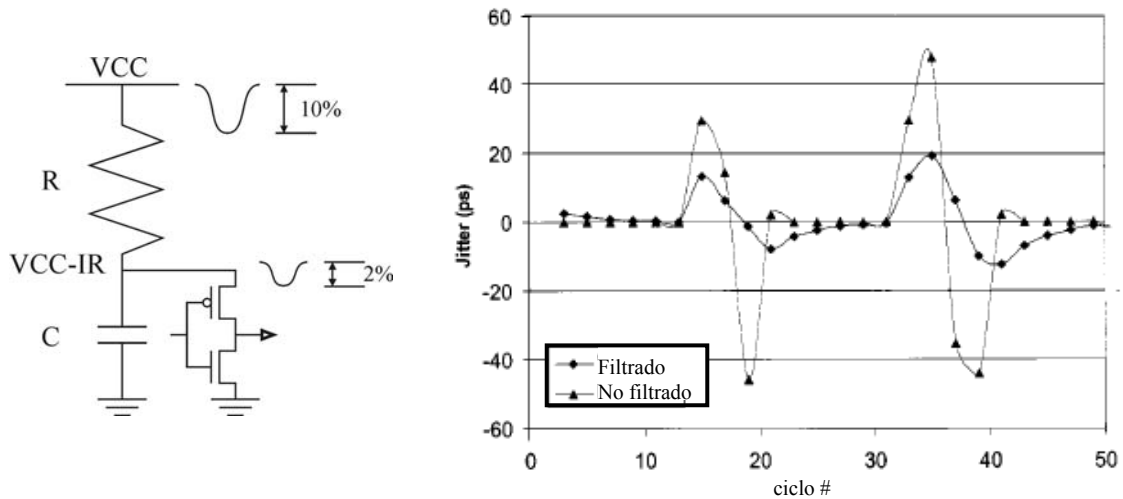


Figura 3.5. Filtro RC para corrección del *jitter* [KUR01].

dispositivos. Una estrategia para limitar el ruido de alimentación consiste en realizar un filtrado RC, tal como se muestra en la Figura 3.5. Este esquema se ha implementado en el procesador Pentium® 4 y consigue reducir el *jitter* del 10% al 2% [KUR01].

La segunda causa en importancia en la generación del *jitter* son los diferentes tipos de ruido que pueden afectar al PLL (*Phase-Locked Loop*: lazo cerrado de fijación de fase), que producen variaciones aleatorias en el periodo de la señal generada. El aumento de la frecuencia de reloj, la disminución de los tamaños de los dispositivos y la integración de todos los componentes en un mismo chip, tienen como consecuencia el aumento de la influencia del ruido. Las variaciones en los niveles de tensión de la fuente y tierra tiene como consecuencia la variación aleatoria de los niveles de referencia del VCO, produciéndose variaciones instantáneas y aleatorias en el periodo de la señal de reloj. Un cuidadoso diseño de los elementos constituyentes del PLL (detector de fase, VCO y filtro en lazo cerrado) puede reducir el ruido a niveles aceptables. En cualquier caso, las fuentes de ruido dominantes en los sistemas PLL son el ruido de fase del VCO y el ruido de la señal de entrada [HEY01], aunque también introducen ruido térmico y *shot* (aunque son menos significativos [HEY00]) los elementos pasivos de los filtros paso baja presentes en el PLL. Recientemente se han realizado diversos trabajos en la caracterización del ruido de fase en osciladores [HAJ98]. Herzel [HER99] analiza el *jitter* debido al ruido de la fuente de alimentación y del sustrato. El efecto del ruido es considerado como un VCO con diferente nivel de referencia de tensión de manera que el efecto del *jitter* aparece como una señal sinusoidal modulada en frecuencia. Este

tratamiento tiene la limitación de asumir al VCO como un sistema determinista en presencia del ruido. En otros trabajos se propone un modelo estocástico [HEY00] para el ruido P/G para diferentes valores de la capacidad de desacoplo del propio chip; sin embargo, no considera el caso general de múltiples frecuencias de reloj dentro del chip. Heydari [HEY01] usa también un modelo estocástico para predecir el *jitter* debido al ruido P/G en un sistema con un PLL de inyección de carga frecuente en los sistemas actuales. Kuppuswamy *et al.* [KUP01] han desarrollado diferentes técnicas para la detección y eliminación del jitter.

3.2.1.4. Electromigración

Un efecto directo del escalado de las dimensiones de los circuitos integrados es la disminución del grosor de las pistas de interconexión de aluminio. El desarrollo de la tecnología de película fina tiene como resultado el aumento de las probabilidades de fallo en la metalización, tanto en el proceso de fabricación, como en el funcionamiento del circuito. La electromigración consiste en una lenta degradación de las finas capas de aluminio, que actúan como interconexiones, debido al paso de corriente a través de ellas. Esta degradación consiste en el arrastre de los átomos de aluminio debido a las colisiones con los electrones que circulan por la red cristalina. Este problema toma cada vez mayor relevancia conforme los espesores de estas pistas de interconexión han ido disminuyendo y los valores de pico de corriente han aumentado.

En el MESA *Research Institute*, en la Universidad de Twente (Holanda), se ha desarrollado un modelo que reproduce los efectos de la electromigración [PET95]. Este modelo predice un cambio en la resistencia de finas capas de aluminio como consecuencia del paso de corrientes elevadas cuyo efecto final es un incremento permanente en la resistencia de la pista una vez las corrientes han finalizado. Este modelo está basado en el movimiento de huecos de la estructura cristalina del aluminio. Las metalizaciones de aluminio presentan una estructura granular en la que se encuentran huecos, que tienen una elevada movilidad en la zona externa de las pistas y están generados por el flujo de corriente, e imperfecciones en la metalización, que se pueden considerar estáticas. El efecto combinado de los huecos y las imperfecciones determinan la resistencia del material. La movilidad de los átomos de aluminio en las pistas tiene como consecuencia dos efectos antagónicos que provocan errores del funcionamiento del chip. En las pistas metálicas se forman dos zonas: regiones de

depleción de masa, que originan grietas y rupturas de las líneas y regiones de acumulación de masa, que pueden ocasionar contactos con líneas adyacentes.

Una solución al problema de la electromigración consiste en limitar los niveles de corriente que circularán a través de los dispositivos, siendo precisamente ésa una de las ventajas que se pueden obtener mediante la aplicación de la estrategia de sincronización propuesta más adelante en esta memoria.

Otra forma de abordar el problema consiste en estimar los valores máximos de corriente que puede ser demandada por un sistema. Este método presenta el inconveniente de que una búsqueda exhaustiva puede ser irrealizable si el número de entradas al sistema es elevado. Lam y Devadas [LAM96] proponen un simulador que calcula las combinaciones de entrada a sistemas combinatoriales de modo que se produzcan los máximos valores de pico en las corrientes demandadas de las fuentes. El algoritmo busca un conjunto de combinaciones de entradas que produzcan el máximo número de conmutaciones en los nodos internos del circuito y, de esta manera, estimar los valores máximos de pico de corriente de alimentación. Así se pueden estimar las dimensiones necesarias de las pistas del circuito para evitar el problema de la electromigración.

3.2.2. Topologías de la red de distribución de la señal de reloj

Como ya se ha mencionado anteriormente, la señal de reloj debe llegar a todos los elementos síncronos del sistema. Para conseguirlo se requiere una red de distribución de la señal que, en la bibliografía, se concibe como un árbol [FRI01]. En los siguientes apartados se van a revisar las estructuras de la red de distribución de la señal de reloj más frecuentes.

3.2.2.1. Distribución no restringida

En el caso más general, la topología del árbol de distribución está condicionada por la estructura del circuito concreto y la distribución que el diseñador ha elegido, por lo que ésta suele ser arbitraria (Figura 3.6). Este tipo de red de distribución es típica de sistemas sintetizados con herramientas automáticas y que no tienen ningún tipo de restricción en cuanto al número de *buffers* empleados, la distribución de las pistas ni a

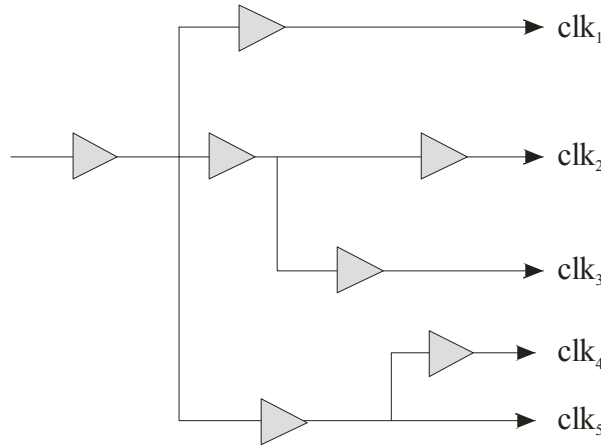


Figura 3.6. Red de distribución no restringida.

las diferencias de los retardos debidas a *buffers* o interconexiones. El diseño culmina cuando la función de coste (3.13) es minimizada. Esta función de coste se define como la suma al cuadrado de la diferencia entre los periodos medidos en cada nodo de la red de distribución y el periodo promediado de todos los nodos (3.14). El objetivo de utilizar esta función de coste en la minimización de los retardos en la distribución de la señal de reloj.

$$\mathfrak{G} = \sum_{i=1}^k (T_{Clk_i} - T_{Clk_promedio})^2 \quad (3.13)$$

$$T_{Clk_promedio} = \frac{1}{N} \sum_{i=1}^k T_{Clk_i} \quad (3.14)$$

Este tipo de diseños puede dar lugar a distribuciones muy descompensadas que hagan difícil un seguimiento o análisis de las variaciones de proceso, voltaje y temperatura. Jackson *et al.* [JAC90] propusieron un método de enrutamiento para conseguir que el retardo en todas las líneas de distribución fuese aproximadamente igual, pero restringido a estructuras muy regulares de celdas pequeñas, como sería el caso de las celdas estándar. Este algoritmo consistía en particionar recursivamente un sistema en dos partes iguales y establecer una conexión entre el punto central (desde el punto de vista del retardo de la señal) del sistema total con los puntos centrales de cada subsistema. Este algoritmo fue mejorado por Kahng [KAH91] para obtener exactamente el mismo retardo en las interconexiones, aunque aún estaba basado únicamente en las longitudes de las líneas. El algoritmo fue finalmente perfeccionado [TSA91] mediante

del uso de modelos RC de las interconexiones, con lo que se consigue la eliminación por completo del *skew*, aunque este algoritmo requiere que los sistemas no tengan demasiados dispositivos y, además, plena libertad para situar las líneas de distribución de reloj, por lo que su aplicación está limitada. Ninguna de las alternativas anteriores es válida para circuitos VLSI en general, donde los diferentes bloques operativos no son iguales, ni se distribuyen de forma regular y donde el circuito puede tener gran cantidad de dispositivos. Otros autores [LIU01, DHA91] combinan algoritmos de enrutamiento lo más simétrico posible con la inserción de *buffers* de tamaño variable cuyo propósito es doble: por un lado, disminuyen el retardo asociado a líneas largas de distribución, mientras que controlan el retardo de cada línea de modo que se equilibren los retardos de cada línea hasta eliminar el *skew*. Ramanathan, Dupont y Shin [RAM94] han desarrollado una alternativa más realista que las anteriores, que asume elementos de distinto tamaño situados según los requerimientos de área del sistema. Una alternativa a éstos métodos es la de realizar una aproximación estadística, que proporciona situaciones en las que la probabilidad de error es muy pequeña, pero no nula, y produce periodos de reloj menores [JIA01]. Herramientas de diseño más modernas incorporan algoritmos más sofisticados que generan redes de distribución no restringidas que permiten un seguimiento mejorado de las variaciones de estos parámetros. Debido a las limitaciones de este tipo de red de distribución su uso queda restringido a sistemas

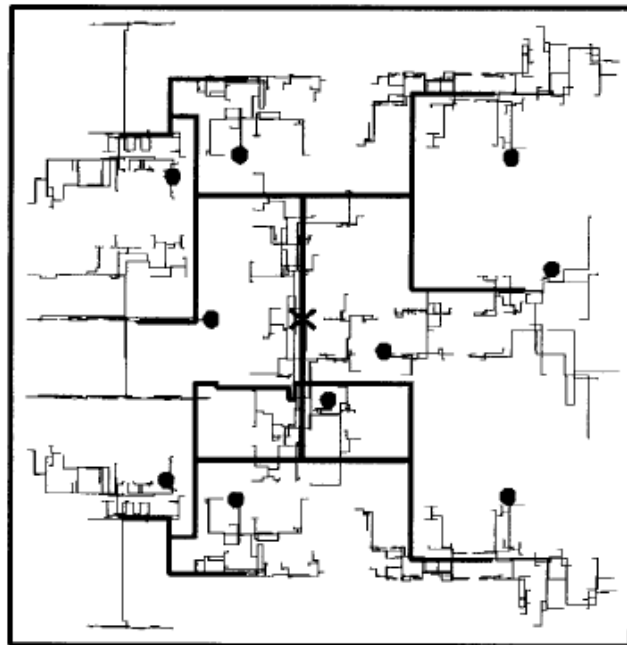


Figura 3.7 Distribución no simétrica [RES98].

reducidos o a pequeños bloques funcionales dentro de sistemas mayores.

Como ejemplos de este tipo de metodología de distribución se verán dos microprocesadores; el primero, el S/390 [RES98], opera a 400 MHz y en la Figura 3.7 se muestran los dos niveles (líneas gruesa y fina) de distribución del reloj. Durante el diseño de este sistema los autores desarrollaron herramientas CAD para el enrutamiento detallado y los grosores de las líneas para la minimización del *skew*, los retardos, el consumo, las longitudes de las líneas y la sensibilidad a las variaciones de proceso. El *skew* medido en los puntos marcados de la Figura 3.7 es inferior a 30 ps. El segundo sistema es el microprocesador Itanium® 2 que opera a 1.7 GHz con 592 millones de transistores implementados en un tecnología de 130 nm [CHA05]. En la Figura 3.8 se puede apreciar cómo el propio diseño es asimétrico en cuanto a la distribución de los bloques funcionales. La cache de tercer nivel ocupa la mayor parte del chip y la distribución del reloj se realiza mediante trece *buffers* repetidores situados de modo que la distribución sea lo más equilibrada posible.

3.2.2.2. Distribuciones simétricas

Las distribuciones simétricas tratan de igualar el retardo asociado a la distribución del reloj en todas sus líneas mediante topologías que presentan una fuerte simetría, de modo

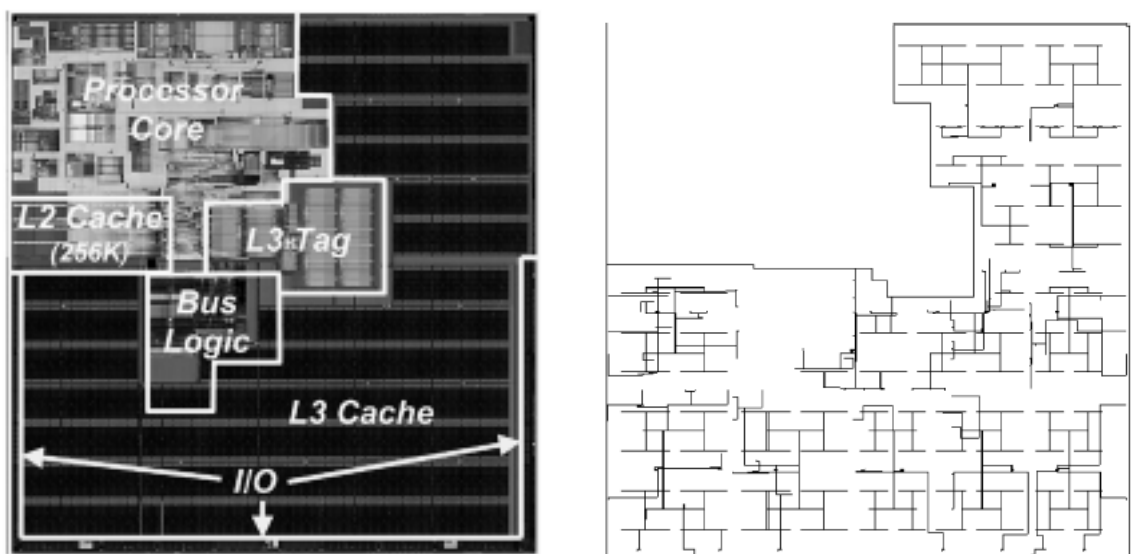


Figura 3.8. Procesador Itanium® 2 y red de distribución del reloj de la caché de 3^{er} nivel [CHA05].

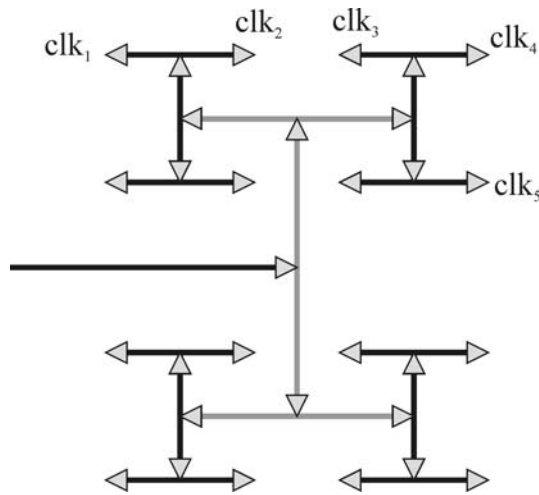


Figura 3.9. Red de distribución en *H-tree*.

que el *skew* nominal de la red de distribución sea cero. El caso más simple es la distribución en H (*H-tree*) (Figura 3.9). Diseñada con tramos de idéntica longitud e idénticos *buffers*, la distribución en *H-tree* resulta muy adecuada para el análisis y seguimiento de variaciones PVT, además de distribuir de modo eficiente la señal de reloj a cualquier parte del chip. Los extremos finales de esta red de distribución llevan la señal de reloj a pequeñas regiones locales de manera que el número de estas regiones determina la profundidad de la red de distribución.

Otras alternativas basadas en los *H-tree* son la distribución *X-tree* y la *H-tree* con acoplamiento de impedancia (Figura 3.10). Mientras que la primera sólo se

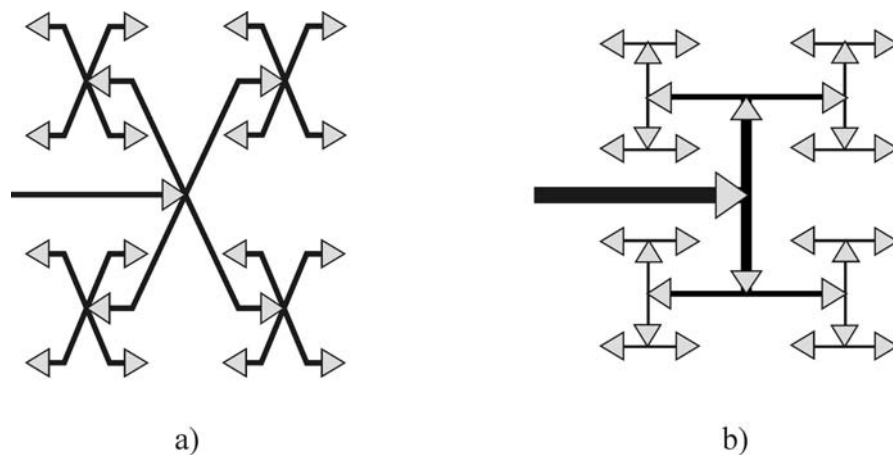


Figura 3.10. Distribuciones en *X-tree* y *H-tree* con acoplamiento de impedancia.

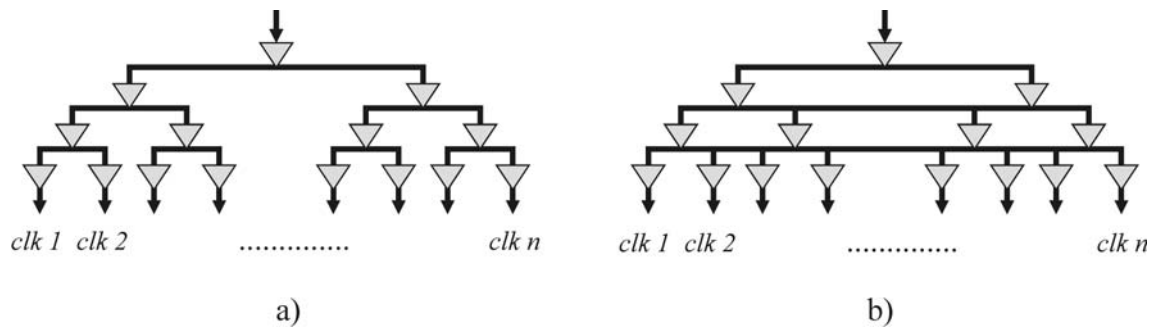


Figura 3.11. Redes de distribución en árbol binario (a) y árbol binario con conexiones intermedias (b).

diferencia en que las líneas de distribución son oblicuas, la segunda consiste en que, a medida que se va avanzando en la red de distribución, la anchura de las líneas disminuye de modo que en cada intersección se produce un acoplamiento de impedancias (el producto RC asociado a cada línea permanece constante) de modo que se evita el reflejo de la señal de reloj. Este tipo de distribución, conocido en la bibliografía como *tapered H-tree* (red de distribución con reducción de sección), se hace necesario a medida que la frecuencia de reloj aumenta y las líneas de distribución se comportan como líneas de transmisión. En estos tipos de red de distribución se consigue una propagación de la señal de reloj tanto en la dirección horizontal como en la vertical.

Una alternativa a las distribuciones anteriores en la que interesa que la señal de reloj se propague en un solo sentido es la distribución en árbol binario (Figura 3.11a). Todas las líneas de esta distribución presentan idénticos *buffers* e interconexiones, por lo que se consigue un *skew* estructural nulo y un sencillo seguimiento de las variaciones PVT. Las distribuciones en árbol binario presentan la ventaja de agrupar los *buffers* próximos entre sí, con lo que la sensibilidad a las variaciones de proceso disminuye, además de facilitar la distribución de los componentes en el chip. Las estructuras en *H-tree* son más complicadas de adaptar debido a las posiciones tan dispersas de los *buffers*.

A pesar de que las distribuciones simétricas presentan un *skew* estructural nulo, existe una incertidumbre en la señal de reloj asociada a los retardos de cada uno de los *buffers* que debe atravesar la señal de reloj. Tam ha demostrado en [XAN09] que tanto el *skew* como el *jitter* aumentan proporcionalmente con la raíz cuadrada del número de etapas de *buffers* y linealmente con el retardo nominal de cada etapa desde el punto de

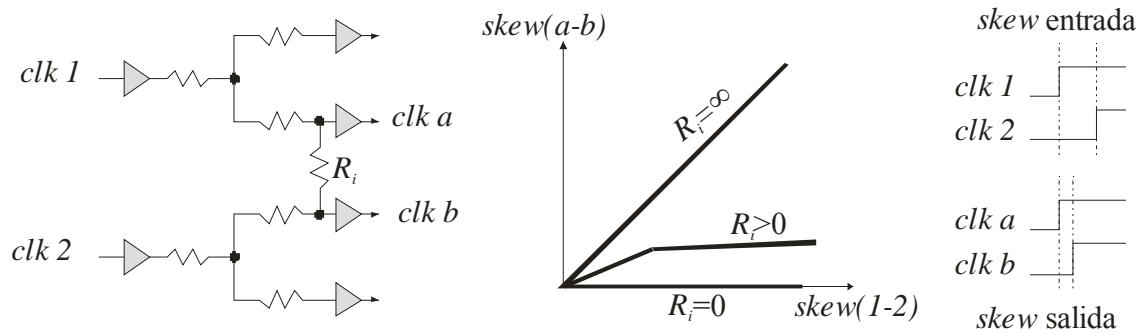


Figura 3.12. Reducción del *skew* mediante interconexiones.

divergencia (POD: *Point of Divergence*), último punto común a ambas líneas de distribución. El resultado es que se genera un *skew* en dos terminales de la red de distribución debido a la suma de los diferentes retardos desde el POD hasta cada uno de los terminales. Para paliar este problema existe la alternativa de los árboles binarios con conexiones intermedias (Figura 3.11b) que presentan la ventaja de reducir el *skew* asociado a los retardos de los *buffers* a lo largo de diferentes vías de distribución [BIN03]. En esta estructura se consigue que el POD esté lo más cerca posible de los terminales de las líneas de distribución. La estrategia de establecer interconexiones entre las líneas de distribución se ha mostrado como un método eficaz de minimizar e incluso corregir el *skew* debido a los diferentes retardos de *buffers* debidos a variaciones PVT. En efecto, si *clk 1* y *clk 2* son señales de reloj desfasadas, se puede reducir este desfase en etapas posteriores interconectando las diferentes líneas de distribución (Figura 3.12). Se define el factor de atenuación del *skew* α como la razón entre el *skew* observado a la entrada de los receptores de la señal de reloj (*clk a* y *clk b*) y el *skew* a la entrada de los *buffers* que transmiten la señal a lo largo de cada línea (*clk 1* y *clk 2*). Si no existe interconexión entre los puntos *a* y *b* se tiene $R_i = \infty$ ($\alpha = 1$) y si la interconexión es perfecta $R_i = 0\Omega$ ($\alpha = 0$). En el primer caso no habría ningún tipo de compensación del *skew* mientras que en el segundo se produciría una corrección total. En casos típicos se suelen tener factores de atenuación α entre 0.1 y 0.8, dependiendo de la topología del circuito y de las restricciones de posicionamiento, para distancias de unos centenares de micras [BIN03].

En ocasiones se usan árboles binarios no simétricos que mantienen el mismo número de *buffers* por etapa pero cada línea difiere de las demás en las longitudes de las

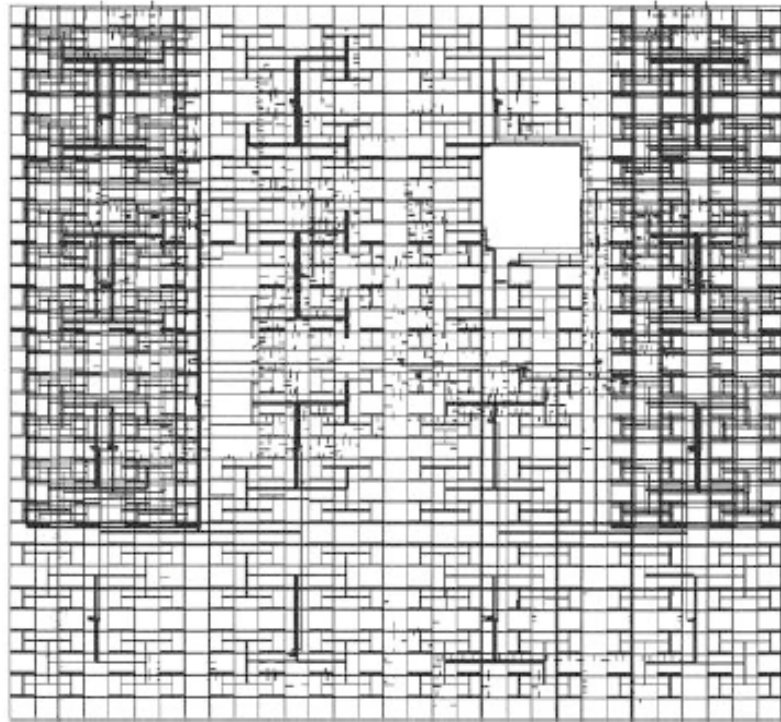


Figura 3.13. Distribución de reloj del PowerPC 64b [HOF00].

interconexiones. En estos casos se requiere un ajuste de los retardos de forma similar a las redes de distribución no restringidas lo cual implica detallados cálculos computacionales durante la etapa de diseño.

Como ejemplo de red de distribución en H, el procesador PowerPC 64b (Figura 3.13) [HOF00] distribuye la señal de reloj desde el PLL a través de un único *buffer* tras el cual hay dos niveles más de *drivers* y tres niveles de *H-trees* que distribuyen la señal de reloj a tres zonas de chip con un *skew* de 9.7 ps, 14.1 ps y 14.9 ps en cada región.

3.2.2.3. Distribución en columna central (*central spine*)

La distribución en columna central es una implementación específica de árbol binario. Todos los *buffers* y las primeras ramas de interconexiones se encuentran agrupados en una columna de la cual salen las líneas finales de distribución en ambos sentidos (Figura 3.14a). Esta alternativa presenta un leve *skew* estructural final debido a que la última etapa pierde el carácter simétrico al estar condicionada por las necesidades de la distribución de la señal de reloj. Generalmente, en sistemas grandes, se tienen varias columnas que definen diferentes regiones de reloj con el fin de minimizar la distancia que tiene que cubrir la última etapa de la red de distribución. Adicionalmente es

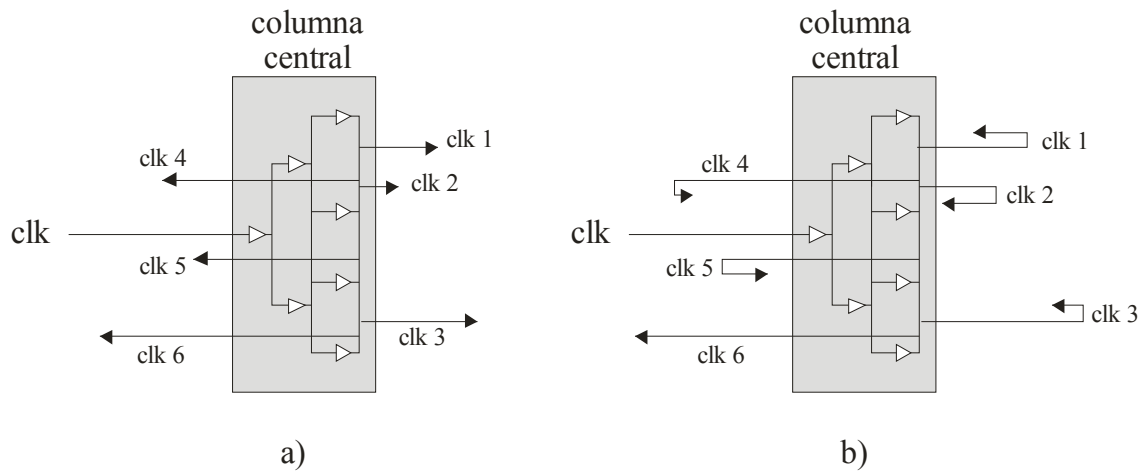


Figura 3.14. Distribución en columna central (a) y con interconexiones equilibradas (b).

frecuente minimizar el *skew* residual de las últimas interconexiones ajustando las longitudes de las mismas para igualarlas con la mayor de ellas (Figura 3.14b).

Los procesadores Alpha® incorporaban en su red de distribución de reloj columnas centrales [GRO98]. Las tres generaciones constituidas por los modelos 21064, 21164 y 21264 fueron sucesivamente diseñadas enfatizando la alta frecuencia de

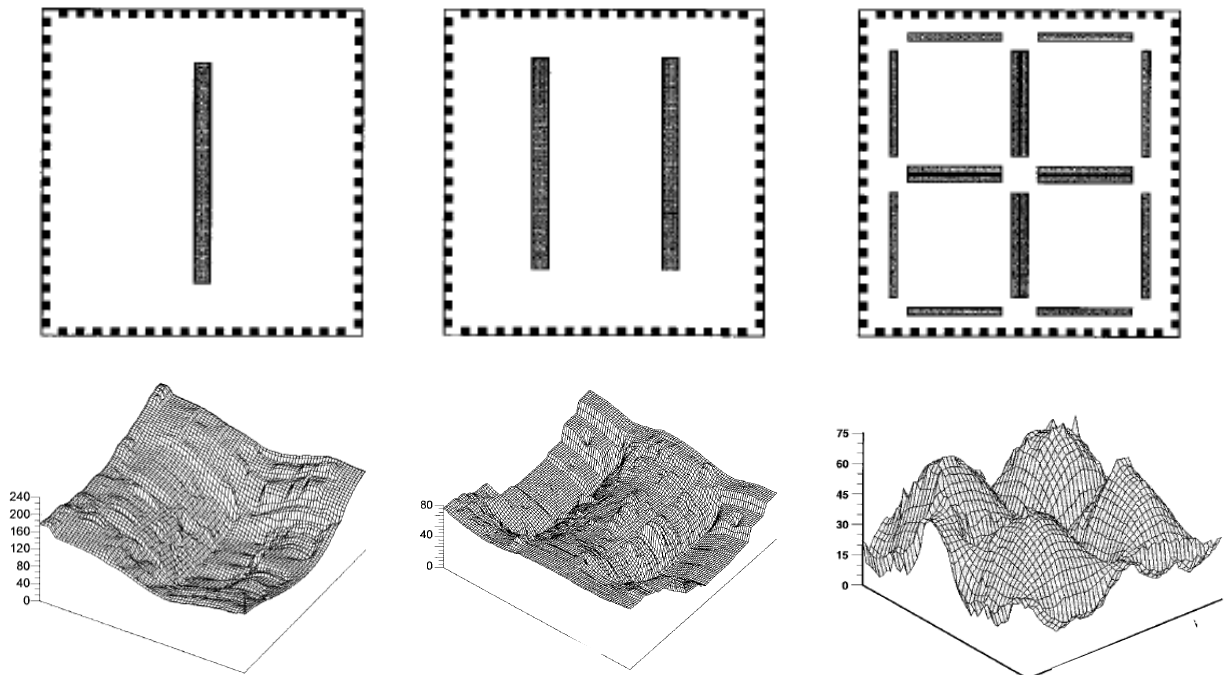


Figura 3.15. Distribución mediante columnas centrales en modelos de procesador Alpha® [GRO98].

funcionamiento. El primer modelo contaba con una única columna central (Figura 3.15), estaba constituido por 1.68 millones de transistores implementados en una tecnología de $0.75\ \mu\text{m}$ y operaba a 200 MHz. El segundo modelo estaba constituido por tres columnas de distribución en dos niveles; la columna central de distribución primaria (no mostrada en la figura) lleva la señal de reloj hacia las dos columnas laterales que distribuyen efectivamente el reloj al sistema. Estaba implementado mediante 9.3 millones de transistores en tecnología de $0.5\ \mu\text{m}$ y su frecuencia nominal de funcionamiento eran 300 MHz. El tercer modelo distribuía la señal de reloj mediante 16 columnas en cuatro regiones además de usar una distribución final de rejilla que se verá en el siguiente apartado. Este modelo implementado en tecnología de $0.35\ \mu\text{m}$ contaba con 15.2 millones de transistores y alcanzaba una frecuencia de 600 MHz. La

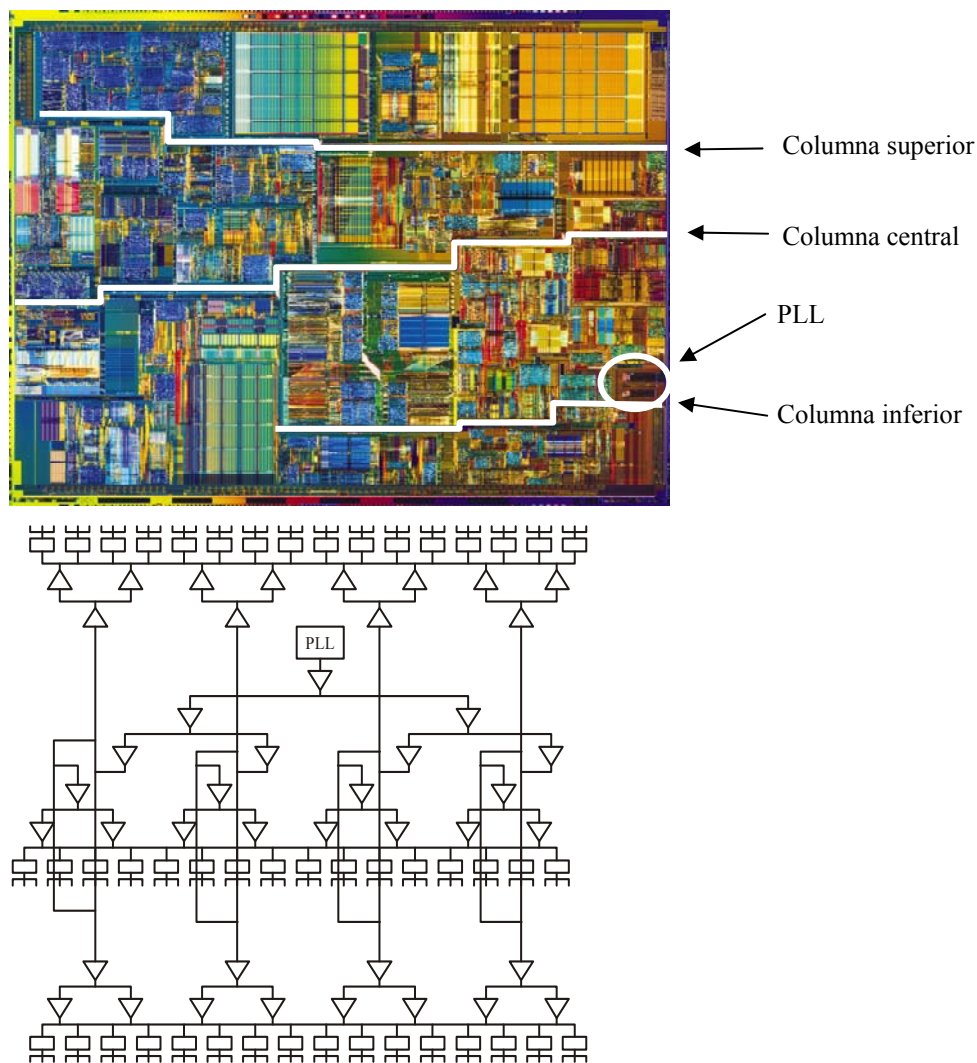


Figura 3.16. Columnas de distribución de reloj en Pentium® 4 [KUR01].

Figura 3.15 muestra las columnas de distribución de los tres modelos así como el *skew* en ps a lo largo de las dos dimensiones del chip.

Un ejemplo de distribución en columna central con interconexiones equilibradas es el Pentium® 4 [KUR01] (Figura 3.16) que distribuye la señal de reloj a partir de tres columnas. La señal de reloj parte del PLL hacia la columna central desde donde es enviada mediante *buffers* que la acondicionan para que llegue en similares condiciones hacia las columnas superior e inferior. El sistema cuenta con dos PLLs, uno para los buses de entrada/salida, que funcionan a 100MHz, el bus común, a 200 MHz el de direcciones, y a 400 MHz, el de datos, y otro para el núcleo, donde la frecuencia se multiplica a tres valores: 1 GHz para los bloques no críticos, como el interfaz de entrada/salida, 2 GHz para la mayoría de los bloques operativos, y 4 GHz para el núcleo de ejecución rápida. Esta configuración mantiene un compromiso entre potencia de cálculo y disipación térmica. Cada columna consta de un árbol binario para la distribución de la señal de reloj, y cada terminal del árbol controla de modo independiente una de la 47 regiones donde se aplica dicha señal. El sistema global consta de 42 millones de transistores implementados en tecnología CMOS de 0.18 μm con seis capas de metal.

3.2.2.4. Mallas de distribución (*grid*)

El fin de la red de distribución es llevar la señal de reloj a los elementos síncronos, generalmente biestables, donde dicha señal es aplicada. La gran cantidad de biestables de cualquier sistema complica esta tarea de modo que las interconexiones terminales de la red de distribución tienden a aumentar en número, complejidad y profundidad. A medida que aumenta la complejidad y longitud de las interconexiones finales, aumentan los retardos y por lo tanto las incertidumbres debidas a POD lejanos a los biestables. El resultado final es el incremento del *skew* y la degradación de la señal de reloj.

La solución a este problema consiste dividir el sistema en reducidas regiones de reloj y aplicar a cada una de ellas una malla de interconexiones, esto es, una rejilla de interconexiones cortocircuitadas que conectan a los *buffers* que alimentan todos los elementos síncronos de la zona (Figura 3.17). Esta interconexión distribuida minimiza los retardos y los iguala, con lo que disminuye significativamente el *skew* y los retardos debidos a los POD al ser regiones de tamaño reducido. Los parámetros críticos de

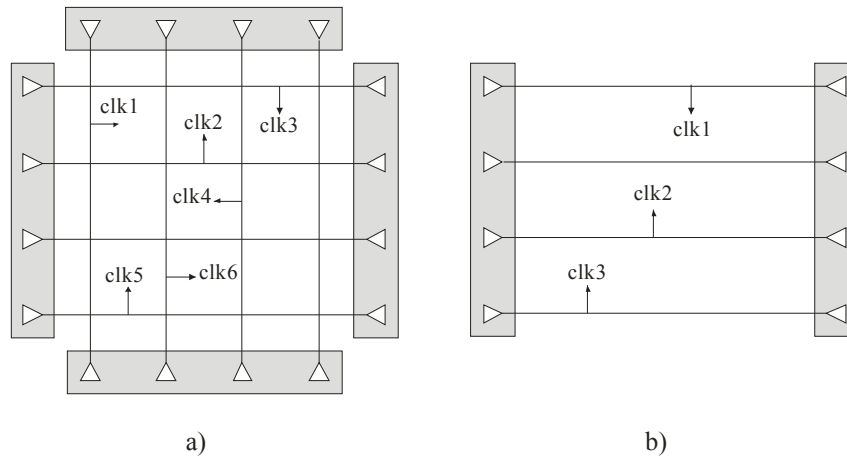


Figura 3.17. Rejilla bidimensional (a) y unidimensional (b).

diseño de la malla son el posicionamiento de los *buffers*, la distancia entre ellos y la separación y grosor de las líneas que la forman. En la Figura 3.17 se muestran dos distribuciones en malla, bidimensional y unidimensional, donde se puede apreciar cómo se pueden extraer diversas señales de reloj en función de las necesidades de los circuitos que sincronizan. El principal inconveniente de la distribución en malla es el incremento del consumo y el aumento del retardo global debido a la gran capacidad asociada a la extensa red metálica que componen las mallas del sistema.

3.2.2.5. Soluciones híbridas

Los sistemas más complejos de sincronizar generalmente implementan conjuntamente varias de las estrategias de distribución que se han explicado en los apartados anteriores. Las combinaciones más frecuentes son las mallas de distribución como elementos terminales combinadas con distribuciones tipo *H-tree* o con árboles de distribución binarios implementados en forma de columna central [GER08].

Como ejemplo de combinación eficaz de *H-tree* con malla se tiene el microprocesador Power4 de IBM [RES02]. Es un chip de doble núcleo basado en tecnología SOI capaz de alcanzar frecuencias superiores a 1.3 GHz cuya red de distribución del reloj destaca por su simplicidad. Los diseñadores implementaron la red de distribución en dos etapas: descendente para llevar la señal de reloj desde el PLL hasta cada una de las mallas de los 6 módulos operativos de cada núcleo, y ascendente desde cada uno de los elementos síncronos de cada módulo hasta la malla de distribución correspondiente. En la primera etapa todas las interconexiones y árboles de

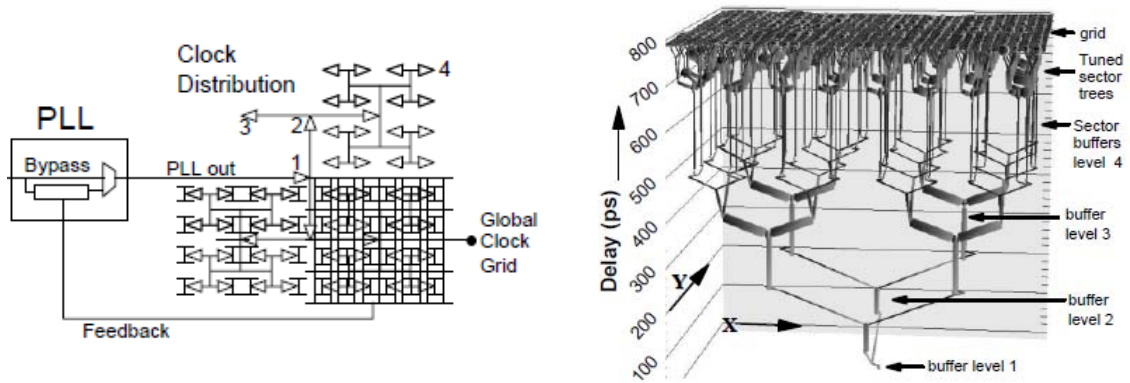


Figura 3.18. Distribución del reloj y retardos a través del chip en el procesador Power4 [RES02].

distribución fueron cuidadosamente diseñadas e implementadas a nivel de circuito. En la segunda etapa un ajuste muy cuidadoso conecta 15.200 pines de reloj desde los elementos síncronos hasta los buffers terminales de la malla. Las herramientas de diseño optimizaron estas conexiones con condiciones de interconexiones muy próximas a los *buffers*, de modo que el efecto de dichas interconexiones en las señales fuera el mínimo posible. El resultado final es una red de distribución sencilla sin regiones de reloj diferenciadas y sin necesidad de realizar ajustes de corrección de *skew* entre dichas regiones. La Figura 3.18 muestra la red de distribución y un diagrama tridimensional del retardo a través del chip. Un análisis de peor caso del *skew* originado por retardos diferentes en los *buffers*, debidos a variaciones de proceso, muestra que el máximo *skew* entre los elementos terminales de la red de distribución es de 70 ps.

Otro ejemplo de solución híbrida es el procesador Xeon® *dual-core* que aprovecha las ventajas de la columna central de distribución junto con la distribución terminal en malla [TAM06]. Al contrario que el sistema anterior, este circuito contiene varias regiones de reloj, dos núcleos que funcionan con una frecuencia de reloj MCLK, circuitería auxiliar para los *cores* sincronizados por un reloj SCLK a mitad de frecuencia del principal, y el sistema de entrada/salida sincronizado mediante ZCLK a un cuarto de la frecuencia principal. La Figura 3.19 muestra las diferentes regiones de reloj y las columnas centrales vertical y horizontal que distribuyen el reloj a lo largo de todo el chip. Las regiones de los núcleos y los circuitos auxiliares son totalmente independientes en cuanto a reloj con el objetivo de aumentar la flexibilidad del sistema.

3.2.2.6. Otras técnicas de sincronización

Hasta ahora se han revisado los métodos más frecuentes de distribución de una señal de alta frecuencia con la mejor calidad posible. Sin embargo, existen otras técnicas que permiten la transmisión de esta señal bien mediante otros tipos de fenómenos físicos diferentes, como pueden ser los pulsos láser o las ondas electromagnéticas viajando por el espacio libre, la generación una onda dentro de un conductor cerrado o la interrupción controlada de la propagación de la señal de reloj en diferentes dominios de reloj a lo largo del chip.

3.2.2.6.1. Sincronización óptica

Ya se ha visto la dificultad de propagar con calidad una señal eléctrica de alta frecuencia a través de pistas metálicas. Una interesante y prometedora alternativa es la utilización de pulsos láser en los niveles superiores de la red de distribución del reloj y la conversión a señal eléctrica en los niveles inferiores. Además, el uso de pulsos de luz puede emplearse para comunicar diferentes chips en la misma o diferentes placas, ya que la gran capacidad asociada a pistas metálicas tan extensas limita enormemente la velocidad de transmisión. La implementación de los elementos ópticos y optoelectrónicos es compatible con la tecnología existente [CHO03, ATM03, TOS04] y su integración en el circuito integrado posible.

La transmisión de señales ópticas dentro del circuito integrado se puede realizar

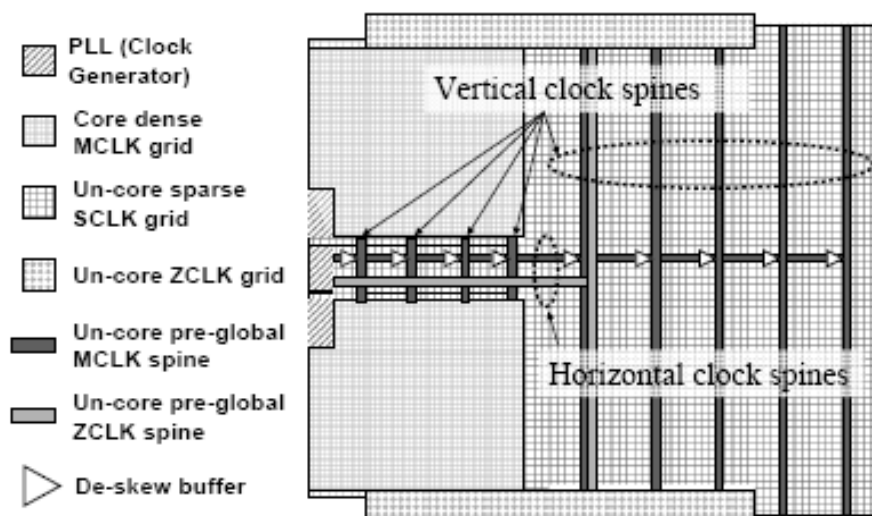


Figura 3.19. Regiones de reloj y distribución en el procesador Xeon® [TAM06].

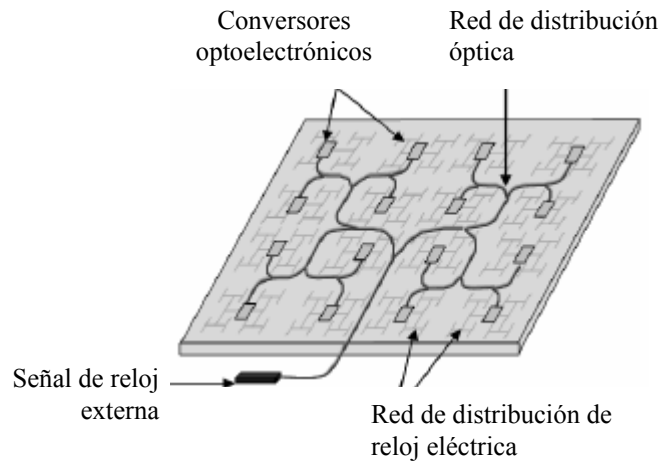


Figura 3.20. Red de distribución global óptica [TOS07].

mediante dos alternativas: la emisión del pulso láser al espacio libre y su distribución sobre el chip mediante un único elemento difractor (DOE: *diffractive optical element*), o la distribución de la señal láser mediante guías de ondas planas insertadas en el propio chip (Figura 3.20). En el primer caso la red de difracción se debe diseñar de modo que envíe la señal óptica a los diferentes puntos del chip que se requiera. Es un mecanismo muy eficiente ya que con un único DOE en forma de matriz bidimensional se puede generar hasta un millón de puntos de luz diferenciados a partir de un haz láser. La alternativa de las guías de ondas son plenamente realizables en las obleas [TOS04] ya que el núcleo de la fibra se realiza directamente con Si (índice de refracción $n=3.5$) mientras que el recubrimiento se implementa con SiO_2 ($n=1.5$). Ambos materiales son transparentes para longitudes de onda del infrarrojo cercano ($1.30\text{-}1.55\ \mu\text{m}$). Los principales inconvenientes de las guías de ondas son la importante fracción de la energía perdida en bifurcaciones y el *skew* debido a gradientes térmicos que afectan al índice de refracción de las guías. Tosik *et al.* han demostrado [TOS07] que asumiendo un gradiente térmico de 50°C en un chip el *skew* es del 4.41% en tecnología de 65 nm a 9 GHz y comienza a ser comparable al *skew* eléctrico (10% aproximadamente) a partir de los 20 GHz en tecnologías de 45 nm.

La conversión del pulso de fotones a señal eléctrica se lleva a cabo mediante un sistema formado por un detector más un receptor electrónico. El receptor convierte la pequeña corriente generada por los fotones en el detector en una señal adecuada para la lógica. El convertidor es la mayor fuente de *skew* y *jitter* del sistema y debe diseñarse

cuidadosamente. Debaes *et al.* [DEB03] diseñaron un tipo de conversor local para generar la señal de reloj directamente en pequeños bloques operativos sin la necesidad de un conversor global. En este caso la capacidad del detector debe ser mínima para conseguir la máxima variación en la tensión de salida del mismo. Dicho receptor electrónico debe tener una alta impedancia capacitiva para, a partir de la señal del detector, generar una señal adecuada para conmutar pequeños bloques de circuitos digitales.

En cuanto a la generación de los pulsos del reloj existen dos alternativas, la inserción desde el exterior de un haz láser modulado o la generación *in situ* de pulsos láser de fase constante (*phase-locked* o *mode-locked laser*) mediante la interferencia de los diferentes modos de la cavidad resonante donde se genera el láser. La primera alternativa requiere un modulador de alta velocidad para acondicionar la señal para ser usada dentro del chip; el modulador es una fuente importante de *jitter*. La segunda alternativa [DEB03] genera pulsos de un alto factor de calidad y muy alta frecuencia. Las ventajas de esta opción son que la frecuencia no induce pérdidas energéticas adicionales, el *jitter* se reduce a unos pocos femptosegundos y está provocado por emisiones espontáneas y variaciones mecánicas en la longitud de la cavidad resonante, y los pulsos emitidos tienen un valor energético del orden de los kilowatios.

Las ventajas de la transmisión [DEB03, ACK05] óptica consisten en bajos requisitos energéticos, gran ancho de banda, baja latencia, acoplamiento mínimo entre líneas e inmunidad a la interferencia electromagnética. El consumo energético de la red de distribución óptica está asociado a los conversores optoelectrónicos y a la fuente emisora de láser, y este consumo es muy poco dependiente de la frecuencia, al contrario que en los sistemas convencionales, en los que el consumo está asociado a la carga y descarga de líneas y capacidades de los dispositivos y en el consumo estático de los *buffers*.

3.2.2.6.2. Sincronización mediante osciladores rotatorios (*rotary clocking*)

Esta es una estrategia de generación y distribución de la señal de reloj que se basa en el carácter inductivo de las líneas de distribución del reloj a altas frecuencias. Los osciladores basados en ondas viajeras (RTWO *Rotary traveling-wave oscillator*) aprovechan la estabilidad de la característica LC de las interconexiones dentro del chip

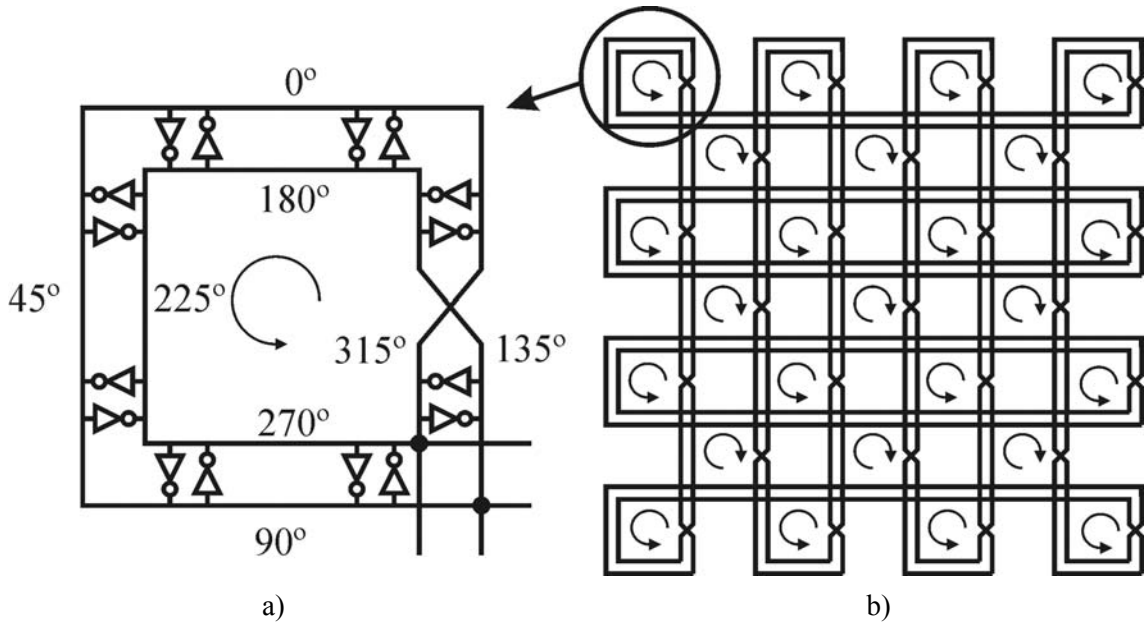


Figura 3.21. RTWO (a) y matriz de 25 RTWOs (b).

para implementar un oscilador distribuido de baja impedancia [WOO01]. El RTWO funciona creando una onda viajera rotatoria dentro de una línea de transmisión diferencial cerrada. Las capacidades de los sistemas conectados a la línea de transmisión forman parte del sistema oscilador y diversos inversores CMOS distribuidos a lo largo de la línea diferencial inician y mantienen la oscilación, aportando la energía necesaria para compensar las pérdidas, al tiempo que generan una señal cuadrada en lugar de la onda sinusoidal típica que se produce en este tipo de líneas de transmisión. Las estructuras RTWO son bastante compactas y se pueden combinar varias de ellas para producir matrices de osciladores rotatorios (ROA *Rotary Oscillator Array*) para ampliar la zona de actuación del reloj en chips extensos. La Figura 3.21 muestra un RTWO en detalle y una matriz formada por 25 RTWOs. En dichas matrices los osciladores individuales se sincronizan entre sí mediante un fenómeno de fijación de frecuencia basado en el acoplamiento de impedancias entre los RTWO, que impiden la reflexión de las señales, con lo que a cada unión llegan simultáneamente dos pulsos que son transmitidos al oscilador adyacente. La frecuencia obtenida en el RTWO es:

$$f = \frac{1}{2\sqrt{LC}} \tag{3.15}$$

donde L y C son la inductancia y la capacidad total del lazo, respectivamente, y el factor 2 representa el hecho que la onda debe recorrer el lazo dos veces por ciclo de reloj. En estos sistemas no existe el reloj de referencia externo y la frecuencia está fijada según (3.15) por las características LC del sistema, aunque dicha frecuencia puede ajustarse situando bancos de varactores a lo largo de la línea de transmisión. Esta estrategia permite obtener fácilmente frecuencias de oscilación en el rango de los GHz.

Combinar la generación y distribución de la señal de reloj permite diseñar sistemas con bajo *skew* y *jitter* [HON09, XU09]. Además, es una técnica muy apropiada para sistemas digitales que operen con dos o más fases del reloj, puesto que dependiendo donde se realice la conexión al oscilador se tendrá una fase u otra. Eso puede suponer un inconveniente en cuanto a la ubicación de las interconexiones, ya que sólo un punto de cada oscilador es adecuado para sistemas que funcionan con una sola fase de reloj (TSPC: *True Single Phase Clock*), que es la estrategia de sincronización más frecuente.

Otro aspecto importante de esta técnica es su carácter adiabático: la energía es reciclada en un alto porcentaje gracias a la característica LC del circuito. Los sistemas convencionales de distribución del reloj se basan en la carga y descarga de las capacidades conectadas a la red de distribución (líneas, *buffers* y elementos síncronos); sin embargo, en los sistemas basados en *rotary clocking* la mayor parte de la energía es reciclada en cada oscilación, con lo que esta técnica resulta muy interesante en sistemas donde el bajo consumo sea prioritario. Yu y Liu [YU05] han probado que el ahorro medio en consumo de estos sistemas es del orden del 60% en comparación con sistemas de distribución convencional en diferentes frecuencias entre 0.6 y 3.5 GHz. Posteriormente han propuesto un método para minimizar el consumo [YU07] y controlar la frecuencia de funcionamiento mediante el diseño de los RTWO, es decir, sus dimensiones y el número y ubicación de los pares inversores que inician y mantienen la señal cuadrada.

Los principales inconvenientes de esta estrategia son la necesidad de determinar exactamente la posición de la interconexión de los diferentes subsistemas a los RTWO y el acoplamiento entre líneas. Existen trabajos [VEN06, VEN07] que proponen métodos

para flexibilizar la ubicación de posibles interconexiones con el objeto de minimizar el *skew* y la capacidad de carga en los anillos.

3.2.2.6.3. Supresión controlada del reloj (*clock gating*)

La técnica del *clock gating* [BEN94, HUD09] consisten en detener la señal de reloj en bloques operativos que no van a ser usados o cuya entrada permanece constante y, por lo tanto, la salida no va a variar. El objetivo es obvio; minimizar el consumo del sistema reduciendo su actividad. Actualmente es una de las técnicas más empleadas en sistemas portátiles donde la reducción del consumo es crítica.

Las herramientas de síntesis implementan, en general, una asociación de lógica combinacional con registros posteriores empleando un multiplexor para mantener un estado en el que la salida no cambie, tal como muestra la Figura 3.22a. El conjunto de señales *en* (*enable*) permiten la carga desde la lógica o el mantenimiento del dato. Este sistema es energéticamente ineficiente puesto que se está cargando continuamente en el registro un dato que ya está establecido, lo que supone la carga de las capacidades internas de los registros en cada ciclo de reloj. Más aún, en los modernos sistemas digitales la distribución de la señal de reloj supone entre un 40% y 60% del consumo total por lo que, la propagación de la señal de reloj a lo largo de los registros, es una fracción importante del consumo de los mismos. La filosofía del *clock gating* consiste en detener la propagación de la señal de reloj dentro de los módulos síncronos, tal como se muestra en la Figura 3.22b. La aplicación de esta estrategia puede suponer un ahorro de entre el 30% y el 50% del consumo dependiendo del sistema considerado y en torno a un 70% sólo en la red de distribución del reloj [BUT07, MAH09 y HUD09].

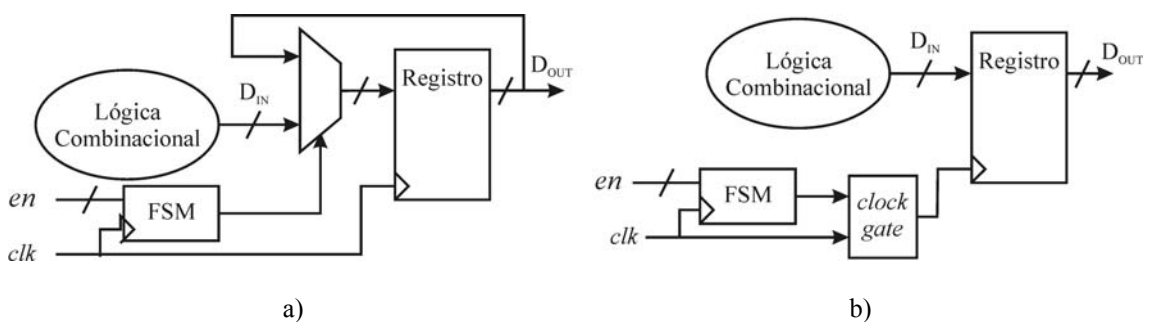


Figura 3.22. Registro con habilitación de carga síncrona (a). Aplicación de *clock gating* (b).

La aplicación de esta técnica implica la inserción de lógica adicional en la red de distribución del reloj por lo que su aplicación supone un incremento tanto en el *skew* como en el *jitter*. Chang *et al.* [CHI08] han propuesto un algoritmo para sintetizar redes de distribución de *skew* nulo controlando el tipo y ubicación de los elementos que implementan el *clock gating*, principalmente puertas AND, OR y *buffers*.

3.2.2.6.4. Oscilación en baja amplitud (*low swing*)

Esta técnica consiste en hacer oscilar a la señal de reloj con una amplitud inferior a la requerida por la lógica durante la propagación. El consumo de potencia en la red de distribución (al igual que en cualquier sistema CMOS) del reloj se puede agrupar en tres categorías [ASG04]:

- consumo dinámico normal: es el mayor componente debido a la alta frecuencia de conmutación de la señal de reloj;
- transiciones de la lógica CMOS: disminuye minimizando los tiempos de subida y bajada de la señal de reloj;
- las corrientes de fuga (*leakage*), que es un efecto poco relevante en la mayoría de los sistemas.

La potencia disipada (P) por la red de distribución del reloj durante el régimen normal de funcionamiento se puede expresar como el producto:

$$P = f \cdot C_L \cdot V_{DD} \cdot V_{sw} \quad (3.16)$$

donde f es la frecuencia, C_L es la carga total de la línea, V_{DD} tensión de alimentación y V_{sw} tensión de oscilación de salida del *buffer*. Normalmente $V_{sw}=V_{DD}$, con lo que:

$$P = f \cdot C_L \cdot V_{DD}^2 \quad (3.17)$$

La estrategia del *low swing* consiste en disminuir el valor de V_{SW} en (3.18) [ASG04] para obtener una disminución lineal en la potencia disipada del mismo factor o la disminución en toda la red de distribución del reloj de V_{DD} [ZHA10] para obtener una disminución cuadrática en (3.17).

La reducción de la tensión de oscilación se puede realizar de dos maneras: mediante fuentes de alimentación duales [GOL94, MOI00] o a partir de una fuente de alimentación única [NAK93, RJO99, ZHA99]. El primer caso requiere una fuente de alimentación independiente para las tensiones reducidas. Usualmente se generan dentro del mismo chip, requieren circuitería extra y suponen un incremento de área pero, como ventaja, reducen el número de transistores en las líneas de distribución de la señal de reloj, lo que mejora el consumo. En cuanto a la fuente de alimentación única se requieren diseños más complicados en la línea de distribución de reloj. Un método es intercambiar los papeles de los transistores n y p, haciendo que los primeros transmitan los valores en alta y los segundos los valores en baja, pero el deterioro en los tiempos de subida y bajada hacen impracticable su uso en sistemas de alto rendimiento.

3.3. Elementos de almacenamiento síncronos

Una de las decisiones más importantes en el diseño de un sistema digital es la elección del tipo y características de los elementos de almacenamiento síncronos (CSEs: *clocked storage elements*). La finalidad de los CSEs es capturar y almacenar temporalmente los datos con los que opera el sistema al tiempo que aíslan las etapas combinacionales. Las características y el funcionamiento de los CSEs dependen de modo crucial de la señal de reloj, por lo que la elección del tipo y características de los mismos es una de las decisiones críticas que se deben tener en cuenta en las fases iniciales del diseño de un sistema. La elección de los CSEs afectará a todos los aspectos del diseño como la frecuencia de funcionamiento, el consumo, la testeabilidad, el proceso de fabricación o las herramientas de diseño desde las simulaciones iniciales a alto nivel hasta las herramientas para el ajuste detallado a nivel de dispositivo.

En los sistemas actuales se pueden encontrar del orden de hasta 1.5 millones de CSEs [KRU08], que suponen entre el 30% y el 40% del consumo total del sistema

[NAF02, FRI07]. En cuanto al retardo, los CSE suelen suponer entre 2 y 3 FO4 (*fan-out-of-four*) [WAR06, GIA07], lo que suele suponer entre el 10% y el 25% del ciclo de reloj en sistemas con retardos por etapa, de entre 10 FO4 en sistemas de alto rendimiento, y 30 FO4 para sistemas de bajo consumo [ZYU04]. El FO4 es una unidad de medida de retardo independiente del proceso para tecnología CMOS. El retardo de una puerta está en función del nivel de escalado, por lo que resulta interesante la definición de una unidad de retardo relativa. Un FO4 se define como el retardo de un inversor cuya entrada está conectada a un inversor cuatro veces menor, que tiene a su salida otro inversor cuatro veces mayor (o equivalentemente un *fan-out* de cuatro inversores idénticos a él). El fundamento de la definición consiste en la verificación de que, para conmutar una carga fijada C_L mediante inversores de capacidad menor, el retardo óptimo se alcanza con una cadena de N inversores (con $N \approx \log_4(C_L/C_{in})$) cada uno cuatro veces mayor que el anterior, donde C_{in} es la capacidad de entrada del primer inversor [SUT99].

Existen básicamente dos tipos de CSEs dependiendo de si los elementos de memoria están controlados por el nivel (*latches*) o por el flanco (*flip-flops* o *FF*) de la señal de reloj. Las características más representativas de los CSEs son los retardos asociados a su funcionamiento. La definición y tipo de estos retardos difieren entre los *latches* y los *FF*, sin embargo, se pueden establecer similitudes entre ellos. La Figura 3.23a muestra los principales retardos definidos en el funcionamiento de los CSEs. Se

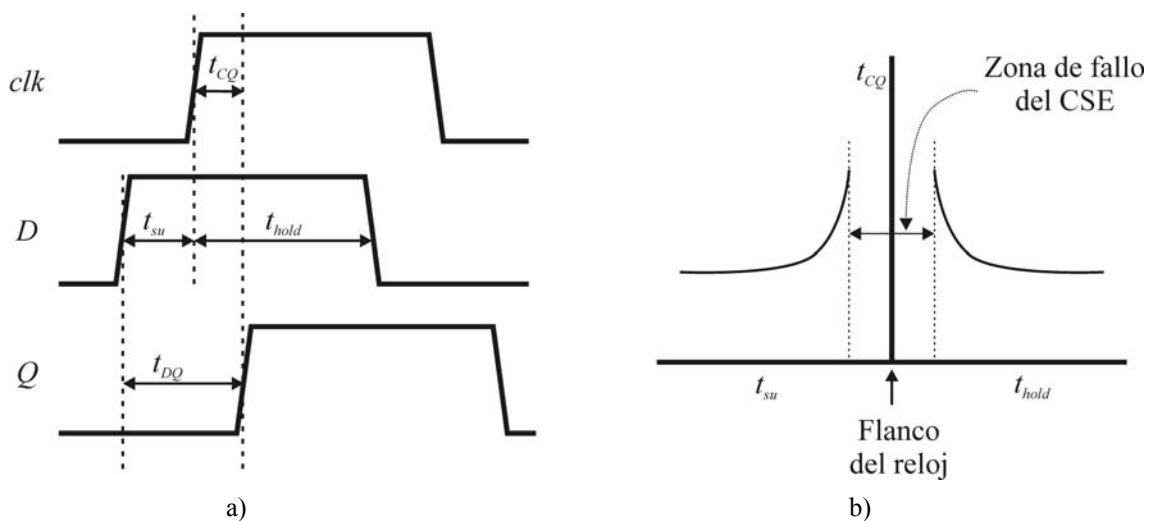


Figura 3.23. Retardos principales en CSEs (a). Dependencia de t_{CQ} con t_{DC} y t_{CD} (b).

define el tiempo que transcurre desde la llegada del dato hasta la propagación a la salida de dicho dato como t_{DQ} y el tiempo desde que ocurre el flanco del reloj hasta la salida del dato como t_{CQ} . El tiempo que transcurre desde que el dato permanece estable previamente a la señal del reloj se define como *setup time*, t_{su} , mientras que el tiempo que el dato de entrada D permanece estable tras el flanco de reloj se define como *hold time*, t_{hold} . Estos retardos están relacionados entre sí de modo que existen unos márgenes de funcionamiento que, de no cumplirse, pueden llevar a un incremento del retardo global o incluso al fallo del CSE.

Desde el punto de vista del funcionamiento la cuestión primordial es ¿cuál es la configuración óptima para minimizar el retardo del CSE? Algunos autores [MAR01] consideran que el retardo del CSE queda determinado por t_{CQ} de modo que, si se minimiza dicho retardo, se obtiene la respuesta más rápida del CSE tras el flanco de reloj. En la Figura 3.23b se puede apreciar cómo a medida que la variación en el dato ocurre cada vez más cerca del flanco del reloj, se produce un incremento en el tiempo que tarda la salida Q en estabilizarse hasta el punto de fallo en la captura. Por otro lado, si se establece un margen de seguridad amplio y el dato se mantiene estable durante demasiado tiempo se consume una parte importante del periodo de la señal de reloj, lo que afecta negativamente a la frecuencia de funcionamiento del sistema. Una

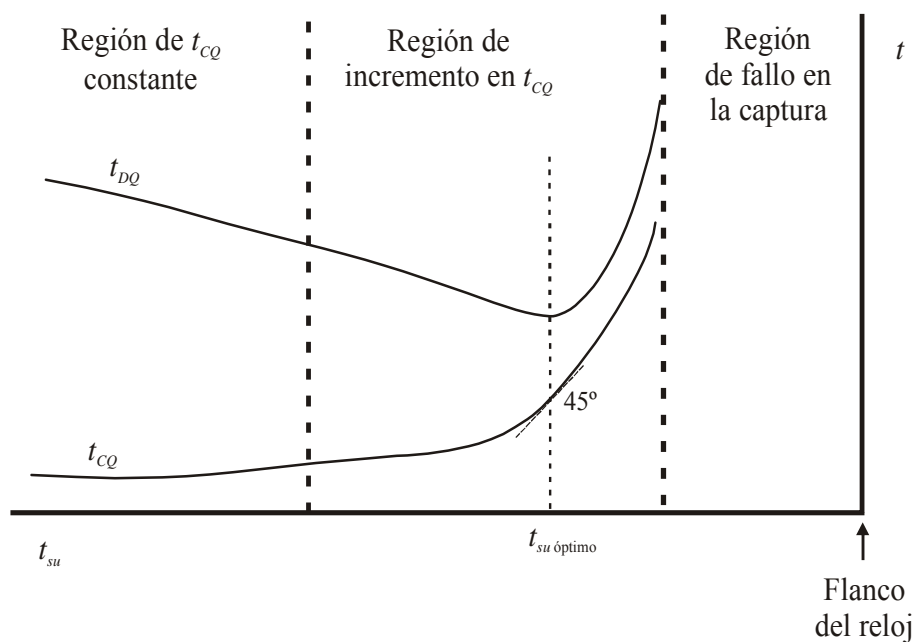


Figura 3.24. Relación entre t_{CQ} y t_{DQ} en función de t_{su} .

visualización más detallada de la Figura 3.23b se tiene en la Figura 3.24. Se pueden distinguir tres regiones en el retardo t_{CQ} en función del tiempo de llegada del dato (t_{su}): región de t_{CQ} constante, región de t_{CQ} variable y región de fallo del CSE. En la primera región el retardo t_{CQ} permanece constante con independencia de la antelación, respecto a la señal de reloj, con la que el dato llegue al CSE. En la segunda región el retardo t_{CQ} aumenta al aproximarse la llegada del dato al flanco del reloj, hasta el punto de fallo en la captura (tercera región). Desde la perspectiva de Marcovick [MAR01], t_{su} se establece de modo que el retardo necesario para que el parámetro t_{CQ} se incremente en un 5% sobre su valor estable. Este criterio establece un buen margen de seguridad frente al *skew* y *jitter* ya que t_{su} se encuentra en una zona de poca variación en t_{CQ} y se mantiene relativamente alejado de la región de fallo del CSE. Este es un criterio típico de sistemas sintetizados a partir de celdas estándar. Otros autores [STO99, OKL03] consideran que, desde el punto de vista de minimizar el tiempo de respuesta del CSE respecto del ciclo de reloj, el verdadero indicador del retardo es t_{DQ} y, por lo tanto, se deben considerar conjuntamente t_{su} , t_{CQ} y su interdependencia. La Figura 3.24 muestra las relaciones entre estos retardos. En la región de t_{CQ} estable, el retardo t_{DQ} disminuye a medida que la llegada del dato se aproxima al flanco de reloj. En la segunda región el retardo t_{CQ} comienza a incrementarse, sin embargo el retardo global t_{DQ} sigue disminuyendo, es decir, el tiempo extra que necesita la salida Q en estabilizarse es aún inferior al tiempo extra que se obtiene al aproximar el dato de entrada D al flanco de reloj. El valor mínimo de t_{DQ} se obtiene en esta región y coincide [OKL03] con la pendiente de 45° de la curva característica de t_{CQ} . De esta manera se puede establecer el tiempo de *setup* óptimo para la respuesta más rápida del CSE frente al ciclo de reloj completo. Esta condición tan precisa del *setup time* no es fácil de alcanzar debido a las incertidumbres en la señal de reloj que, en este caso, tienen un efecto más significativo que en el anterior tanto en la degradación del retardo como en el posible fallo en la captura del dato. Sin embargo, si los CSEs están próximos, tal como ocurre en la ruta de máximo retardo (*critical path*), es posible realizar un ajuste muy fino de la señal de reloj y reducir el tiempo de proceso de esa ruta concretamente para mejorar la frecuencia de funcionamiento del sistema global.

La definición del *hold time* está relacionada con el *setup time* que se haya considerado; t_{hold} es el tiempo que el dato se debe mantener estable tras el flanco de reloj de modo que el retardo t_{CQ} sea el mismo que el obtenido con t_{su} (Figura 3.25). El

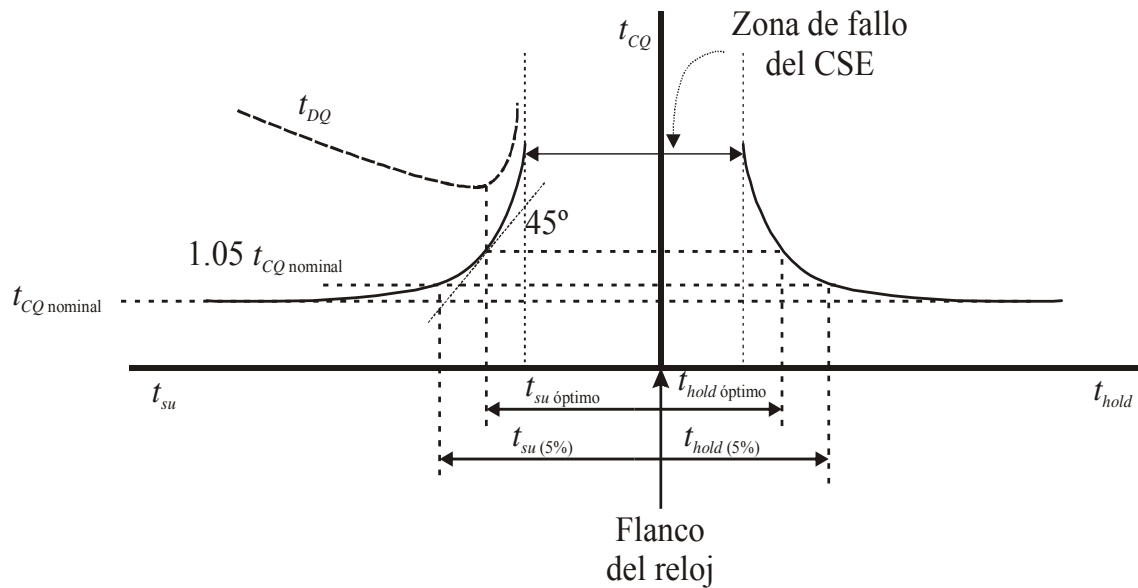


Figura 3.25. Definiciones de t_{su} y t_{hold} .

efecto del valor de t_{hold} sobre el sistema es igual de crítico que el de t_{su} . En la etapa de diseño se debe tener muy en cuenta el retardo de la lógica combinatorial entre dos etapas de CSEs ya que en rutas de procesamiento rápidas, si el tiempo de procesamiento de la lógica es menor que t_{hold} se puede producir un problema de carreras al sobrescribir un dato con el siguiente antes que el primero sea capturado. La suma de t_{su} y t_{hold} determina el tiempo total que el dato de entrada debe mantenerse estable, por lo que todos los elementos de procesamiento combinatoriales deben estar ajustados dentro de los márgenes establecidos por estos parámetros. La ruta de procesamiento más lenta debe proporcionar su salida como máximo t_{su} antes del flanco del reloj que determina la captura en los *FF* o el fin de la fase de captura en los *latches*, mientras que la etapa más rápida debe tener un retardo de procesamiento como mínimo igual a t_{hold} después de dicho flanco. En general t_{su} y t_{hold} tienen valores diferentes ya que el proceso de captura de un dato no es simétrico respecto al flanco de la señal de reloj, tal como se muestra en la Figura 3.25.

Para los circuitos controlados por flanco, el retardo asociado a una línea entre dos *FFs* debe ser menor que el periodo de reloj, mientras que para los circuitos disparados por nivel el retardo puede ser mayor. Esto es así porque en los sistemas sincronizados por nivel existe más flexibilidad en la llegada y captura del dato que en los *FFs*, en los que el instante de captura está muy definido por el flanco del reloj. Esta

flexibilidad se aprovecha mediante una estrategia conocida como *cycle stealign* o *cycle borrowing* [LIN92, HAR97] y consiste en proporcionar más tiempo de cálculo a las etapas más lentas a costa del tiempo de las etapas mas rápidas. En cuanto a la velocidad, los circuitos disparados por nivel son más rápidos y requieren menos elementos de memoria, pero su operación es más complicada y su optimización más difícil [MAH99]. En los circuitos controlados por nivel existen diversas estrategias de sincronización que pueden requerir el uso de una, dos o cuatro señales de reloj (no solapadas en los dos últimos casos), mientras que en los sistemas sincronizados por flanco se requiere una única señal de reloj.

En cuanto al consumo del CSE, éste se puede calcular mediante la expresión 3.18:

$$P_{total} = \int_t^{t+T} V_{DD} \cdot i_{VDD}(\tau) \cdot d\tau \quad (3.18)$$

en la que se supone que en el periodo T deben haber ocurrido transiciones en el dato de entrada, el reloj y en el dato de salida. El consumo de un CSE se puede desglosar en cuatro términos [OKL03]: conmutación, cortocircuito, fugas y consumo estático:

$$P_{total} = P_{conmutación} + P_{cortocircuito} + P_{fugas} + P_{estático} \quad (3.19)$$

El consumo de conmutación se define como:

$$P_{conmutación} = \sum_{i=1}^N \alpha_{tran}(i) \cdot C_i \cdot V_{SW}(i) \cdot V_{DD} \quad (3.20)$$

donde N es el número total de nodos del CSE, $\alpha_{tran}(i)$ es la probabilidad de que ocurra una transición en el nodo i , C_i es la capacidad del nodo, $V_{SW}(i)$ es la tensión de oscilación en el nodo y V_{DD} es la tensión de alimentación del CSE. En general ésta es la mayor componente en el consumo total y depende de la actividad del CSE. Una

reducción en cualquiera de los términos de (3.20) puede suponer una disminución significativa del consumo de todo el sistema.

El término de consumo en cortocircuito hace referencia a la corriente consumida durante las transiciones de la lógica CMOS. Un sistema basado en esta lógica normalmente mantiene en activo los transistores tipo N o los tipo P, de modo que la conducción de unos supone el corte de otros. Con esto la corriente solamente fluye desde las fuentes hasta la capacidad de carga de un nodo o desde dicho nodo o hacia tierra. Sin embargo, si la tensión que alimenta las puertas de los transistores experimenta transiciones lentas, con tiempos de subida o bajada relativamente largos, se produce la conducción simultánea de ambos tipos de transistor, con lo que se establece momentáneamente un cortocircuito entre alimentación y tierra. La señal de reloj suele ser la que determina las transiciones de los transistores, de ahí la importancia de que dicha señal presente la mejor calidad posible con rápidas transiciones. En general esta componente supone menos del 10% del consumo total.

La energía perdida en fugas proviene de dos tipos de corriente: la corriente de diodo en inversa en los drenadores de los transistores y la corriente subumbral a través del canal de un dispositivo en corte. La primera proviene de las uniones PN inversamente polarizadas entre drenador y substrato, o fuente y substrato, en un transistor en corte. Esta componente normalmente es despreciable frente a la otra componente de la corriente de fugas. La corriente subumbral se debe a la difusión de portadores entre drenador y fuente cuando la tensión aplicada a la puerta es inferior a la tensión umbral, es decir el transistor se encuentra en el rango correspondiente a la región de inversión débil [RAB03]. Esta corriente se hace más importante a medida que se reducen las dimensiones de los dispositivos, como muestra el hecho que el procesador Power6TM formado por 700 millones de transistores implementados mediante un proceso de 65nm en tecnología SOI que operan a 5 GHz, en el que las corrientes de fugas suponen el 42% del consumo total.

La energía estática disipada es poco frecuente en CSEs y puede provenir de la aplicación de bajos niveles de tensión aplicados a circuitos CMOS o circuitos con componente de corriente continua como pueden ser los basados en pseudo-nMOS.

3.3.1. Sincronización de CSEs controlados por nivel (*latches*)

Los *latches* son circuitos que transmiten el dato de entrada D hasta la salida Q mientras el reloj mantenga el nivel alto (*latches* positivos) o bajo (*latches* negativos), y mantienen su salida durante la fase contraria. Durante la fase en la que se permite el paso de información el *latch* está en modo transparente. Existen en la bibliografía una gran variedad de tipos de *latches* [SUZ73, GON83, JIR87, YUA89, RAB03, XAN09].

Los primeros *latches* se diferenciaban de los *flip-flops* en que, mientras que éstos almacenaban efectivamente la información, aquéllos simplemente actuaban como puertas de paso, permitiendo la comunicación entre etapas o aislándolas, debiendo ser la lógica precedente la que mantuviese el valor de la información para la etapa siguiente. La Figura 3.26 muestra algunos de estos *latches*. El primer caso [SUZ73] (Figura 3.26a) estaba basado en la técnica *Clocked CMOS* (C^2 MOS) y fue un *latch* de uso bastante frecuente, simple de implementar, sincronizado mediante la técnica NPTC (*Non-Overlapping Pseudo-Two-Phase Clock*). El punto débil de este *latch* era el requisito de cuatro señales de reloj no solapadas, que lo hacía muy sensible al *skew*, la existencia de tiempos muertos entre las fases en alta de las cuatro señales consumía parte del ciclo de los relojes, y presentaba muchos inconvenientes cuando se requería incrementar la velocidad de funcionamiento. La segunda estrategia, NORA (*NO RAcés*) *dynamic CMOS*, [GON83] (Figura 3.26b) solamente requería dos fases de reloj y era muy resistente a las carreras debidas al *skew* mediante ciertas restricciones en la implementación lógica. En una construcción típica se tenían dos secciones denominadas sección ϕ y sección $\bar{\phi}$ y en cada sección se aplicaba una técnica de precarga de datos. Las secciones ϕ y $\bar{\phi}$ se situaban alternativamente de manera que entre dos secciones hubiera un número par de inversiones del dato que se iba a transmitir. Esta estrategia supuso un avance en cuanto a fiabilidad frente al *skew* y velocidad de funcionamiento. La tercera estrategia mostrada [JIR87] esta basada en la técnica TSPC (*True Single Phase Clock*), es decir, la propagación de una única señal de reloj (Figura 3.26c). En esta estrategia se tenían etapas N y P de inversores dobles alternadas que funcionaban con precarga N y P respectivamente. La distribución de una única señal de reloj hacía a este sistema más resistente al *skew* y permitía frecuencias superiores a las anteriores estrategias.

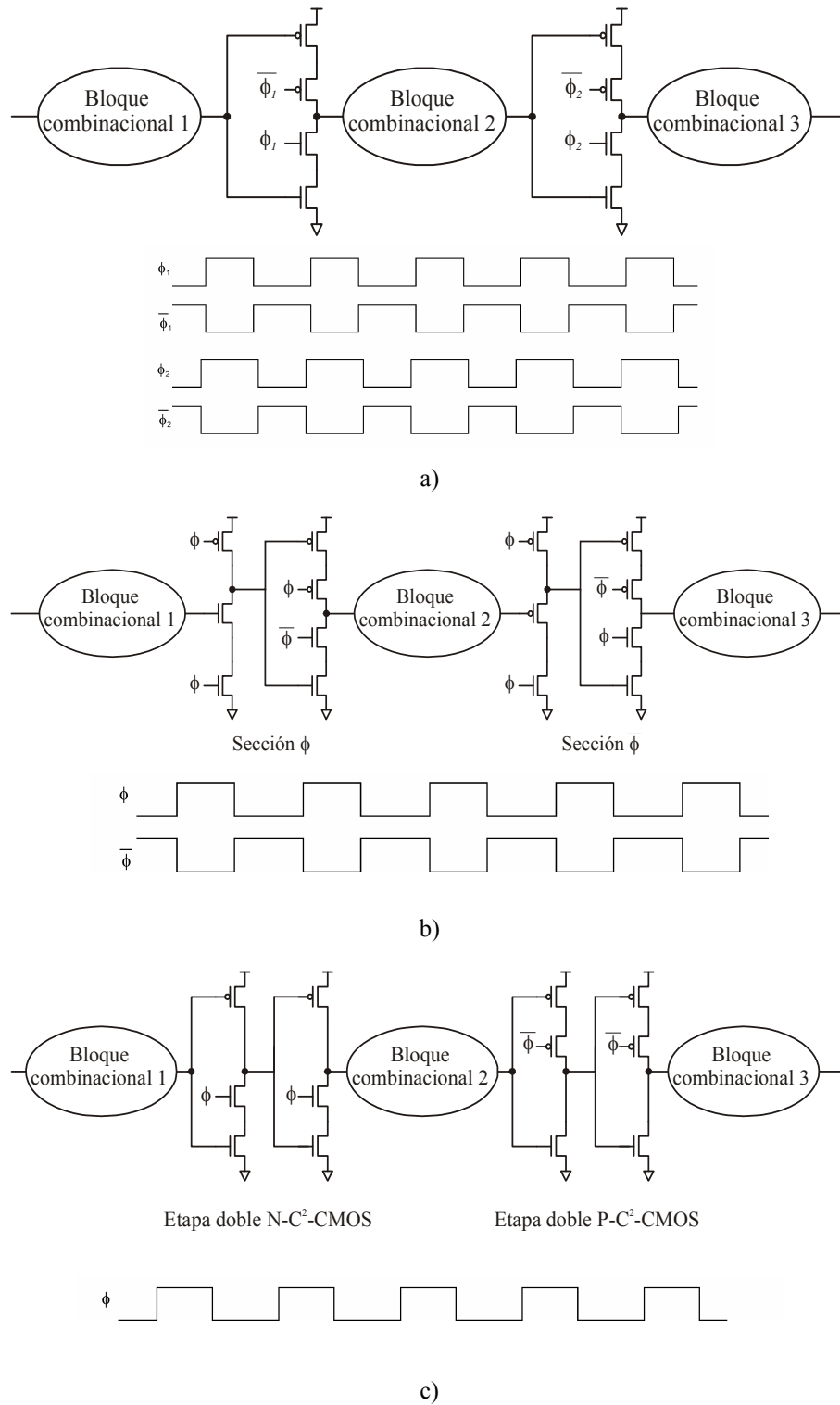


Figura 3.26. Diferentes estrategias de sincronización mediante *latches*:
 a) lógica C²CMOS con NPTC;
 b) NORA CMOS dinámica;
 c) TSPC CMOS dinámica.

Como ejemplo representativo de uno de los diseños actuales más frecuentes la Figura 3.27 muestra un latch tipo MSL (*Master-Slave Latch*) [WAR06]. Este tipo de *latch* es capaz de capturar y almacenar el dato de entrada. El MSL requiere dos señales de reloj, clk_d (*data capture clock*) que es la señal de captura del dato y clk_l (*launch clock*) que es la señal de transmisión a la salida que, salvo en diseños orientados a alto rendimiento, no se deben solapar. Mientras clk_d permanece en nivel alto, el dato D es capturado por la primera etapa del *latch*, al tiempo que la segunda etapa permanece aislada de la primera manteniendo el dato anterior Q , al estar clk_l en bajo. Al pasar clk_d a nivel bajo y clk_l a nivel alto el primer *latch* se aísla de la entrada D al tiempo que el dato que almacena es transmitido al *latch* de salida. En diseños orientados al alto rendimiento se permite el solapamiento en fase alta de ambas señales de reloj, de modo que sea posible la transmisión de datos de etapas más lentas a través del *latch* a costa de consumir parte del ciclo de la etapa siguiente (*cycle borrowing*). En estos casos el dato debe llegar en la mitad del periodo de tiempo en el que ambas señales están en alta para evitar interferencias con el flanco de la señal de reloj. Este esquema de funcionamiento hace al sistema más tolerante al *skew*. El inconveniente de esta estrategia consiste en

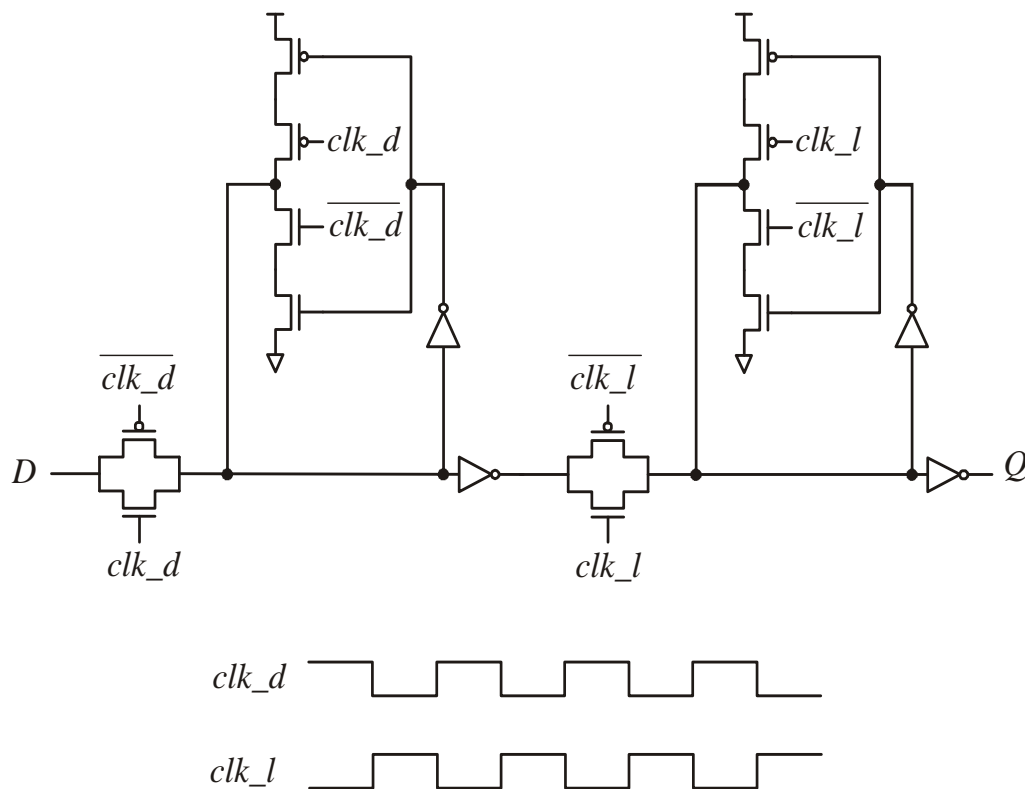


Figura 3.27. Latch tipo master-slave [WAR06].

que, al retardar el paso a nivel bajo de clk_d , se incrementa el *hold time* creándose situaciones de carreras entre *latches* en rutas de procesamiento rápidas. La solución a este problema es añadir elementos de retardo adicionales (*delay padding*) a estos bloques combinacionales.

Como ya se ha dicho, los *latches* son dispositivos sincronizados por nivel. Esto lleva a la posibilidad de dos posibles escenarios de captura en función de si la llegada del dato ocurre antes de que el *latch* pase a modo de captura o después (Figura 3.28). Los tiempos t_{su} y t_{hold} se definen en los *latches* a partir del flanco que determina el final de la etapa de muestreo (*sampling window*). El dato de entrada puede llegar antes que el *latch* entre en dicha fase y en este caso el retardo de la salida será t_{cQ} (D_1 en la Figura 3.28) y no tiene influencia alguna la restricción de *setup*. Por otro lado el dato puede

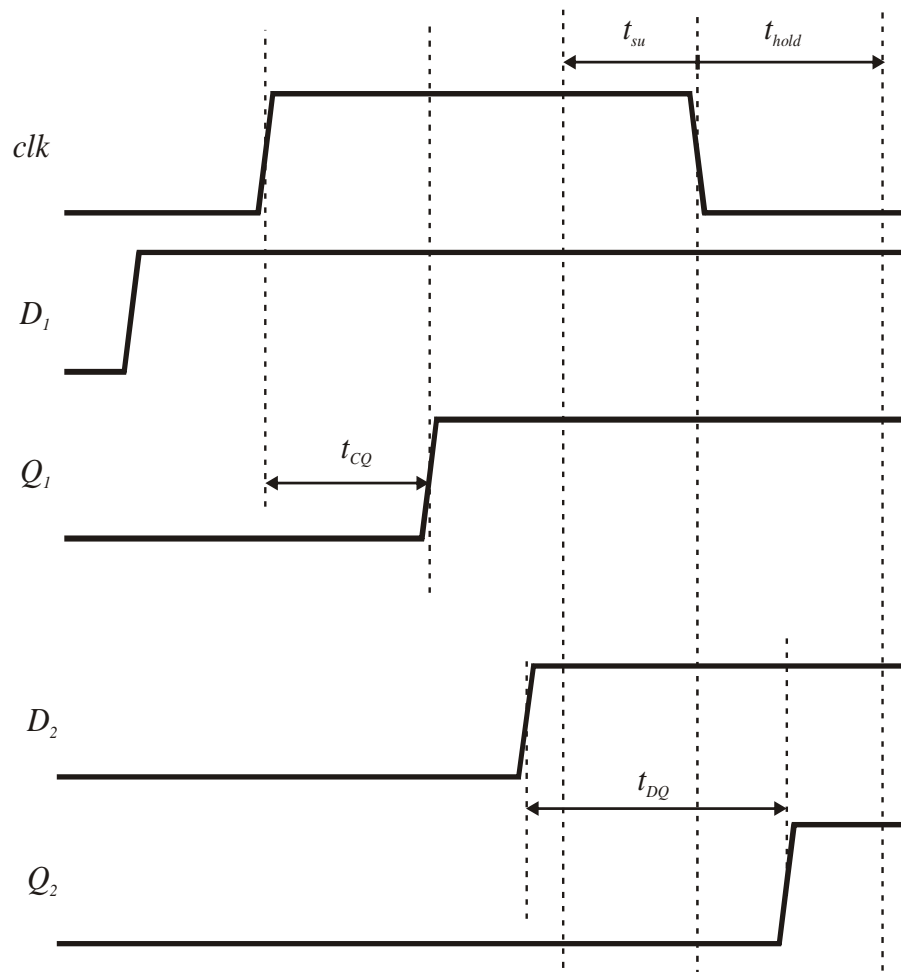


Figura 3.28. Diagrama de captura de datos en *latch*: dato anticipado (D_1) y dato retardado (D_2).

llegar durante la fase de captura (D_2 en la Figura 3.28) y deberá hacerlo respetando el tiempo t_{su} . En este caso el retardo del *latch* respecto al dato de salida será t_{DQ} que es independiente de la señal de reloj. En ambos casos los datos de entrada deben mantenerse estables hasta cumplir la restricción de *hold time*.

3.3.2. Sincronización de circuitos controlados por flanco

Los circuitos controlados por flanco son la alternativa a los circuitos controlados por nivel. El flanco (de subida o de bajada) de la señal de reloj indica el instante en el que el dato será capturado y transferido inmediatamente a la salida del *flip-flop* hasta la llegada del siguiente flanco [LLO92]. Durante ese tiempo los bloques combinatoriales deben realizar el procesamiento de los datos para que éstos sean cargados en el siguiente registro cuando se vuelva a producir un flanco en la señal de reloj.

Los circuitos controlados por flanco presentan varios inconvenientes respecto de los disparados por nivel: en primer lugar son más lentos y tienen mayores requerimientos de *hardware*, por lo que el consumo y el área se incrementan. Por otro lado, el procesamiento de los datos debe completarse durante el ciclo de reloj, mientras que en los circuitos de tipo *latch* el retardo concreto de un módulo puede ser mayor que el periodo de reloj (aunque este retardo debe ser compensado en etapas sucesivas); además, son más vulnerables a las incertidumbres del reloj puesto que su funcionamiento está determinado por el flanco de la misma. Como ventajas los circuitos controlados por flanco operan de forma más sencilla y su optimización es más fácil. Dado que este tipo de circuito sólo necesita la transición de la señal de reloj, la sincronización puede ser del tipo TSPC, con lo que solamente es necesario distribuir una señal de reloj a lo largo de todo el circuito, aunque, al estar la sincronización supeditada a la transición de la señal de reloj, se requiere gran calidad en el perfil de la señal en cada CSE.

De igual modo que los *latches*, existe una gran variedad de diseños de *FF*. Los ejemplos más típicos son los *FF* basados biestables tipo D [AFG90, LLO92], los basados en amplificación de pequeña señal, SAFF (*Sense-Amplifier Flip-Flop*) [MAT94, RAB03], los *FF* híbridos, que combinan el modo de funcionamiento del *latch* y *FF*, HLFF (*Hybrid Latch Flip-Flop*) [PAR96] y los *FF* semidinámicos, Sdff (*Semi-*

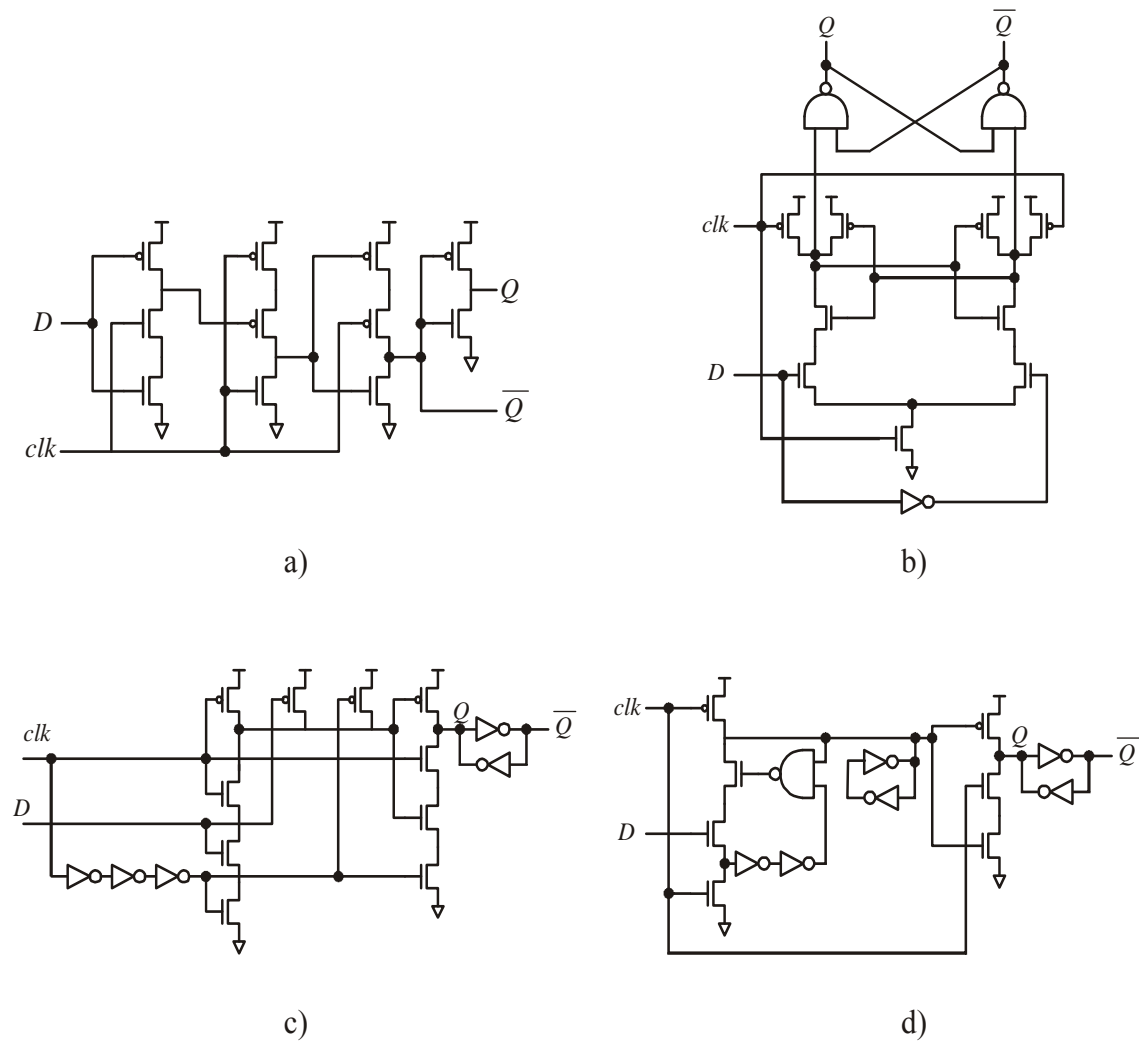


Figura 3.29. Cuatro ejemplos de *flip-flop* típicos sincronizados mediante TSPC:
 a) *flip-flop* basado en biestable tipo D;
 b) SAFF
 c) HLFF
 d) SDFF

Dynamic Flip-Flop) [KLA99, XAN09]. La Figura 3.29 muestra un ejemplo de cada uno de estos biestables. El *FF* basado en biestables tipo D (Figura 3.29a) es muy frecuente debido a su sencillez y robustez frente a las incertidumbres de la señal de reloj. El SAFF (Figura 3.29b) es capaz de aceptar una señal de entrada de baja amplitud y amplificarla hasta el valor adecuado para ser útil como señal digital. Se usan con frecuencia en circuitos de memoria o como etapa de adaptación en estrategias de *low swing* (apartado 3.2.6.4) para reducir el consumo o mejorar el rendimiento del sistema. El *FF* híbrido (Figura 3.29c) se emplea para aprovechar en un único CSE las ventajas de los *FF*, como son el funcionamiento basado en flanco, baja latencia y baja capacidad de entrada, y la principal ventaja de los *latches*, flexibilidad en los tiempos de llegada de los datos

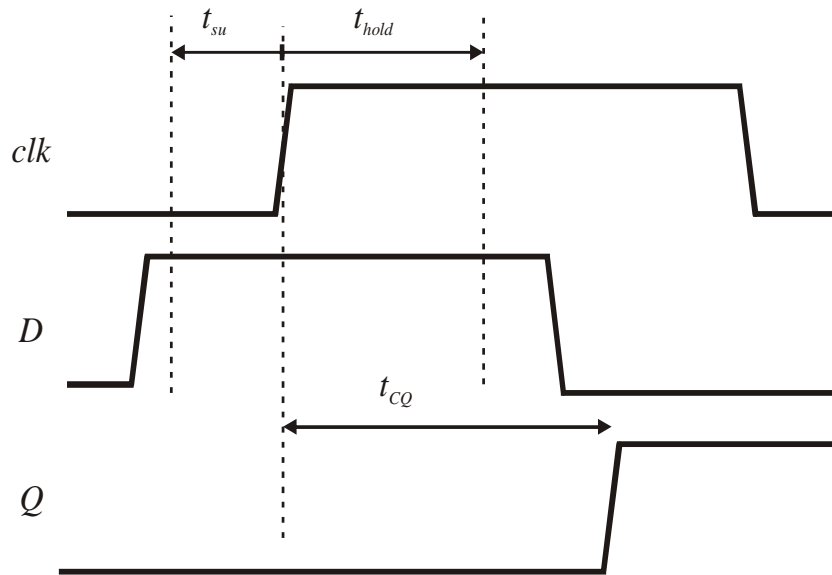


Figura 3.30. Cronograma de *flip-flop*.

para su captura. Por último el SDFF tiene como principal ventaja la capacidad de incluir funciones lógicas embebidas con poco retardo en su funcionamiento de modo que son interesantes en sistemas de altas prestaciones.

Las restricciones de *setup* y *hold* de los *FF* están referidas al flanco de reloj que define el funcionamiento del mismo. En la Figura 3.30 se representa el cronograma de funcionamiento de un *FF* que funciona en flanco de subida.

3.4. Conclusión

En este capítulo se ha realizado una completa revisión de la sincronización de sistemas digitales VLSI. Se han definido los principales parámetros que intervienen en la propagación y aplicación de la señal de reloj, se han comentado las principales dificultades que surgen en el proceso de sincronizar sistemas con grandes cantidades de dispositivos repartidos en regiones extensas del chip a frecuencias cada vez mas elevadas.

Posteriormente se han recorrido las tres fases de la sincronización de sistemas: la generación, la distribución y la aplicación de la señal de reloj. En la generación se han

explicado someramente las diferentes alternativas de producción de la señal de reloj comentando diferentes estructuras y metodologías con sus ventajas e inconvenientes. Después se ha realizado una introducción a las dificultades que se plantean en la propagación de una señal de alta frecuencia a través de interconexiones y *buffers* y cómo diferentes estrategias de distribución atenúan o corrigen estos efectos indeseados. El análisis de las diferentes topologías de distribución del reloj se ha ilustrado con ejemplos de sistemas reales y se han comentado las ventajas, inconvenientes y los casos más eficientes de aplicación de cada una de las estrategias de distribución.

Por último se han revisado en detalle las características de los elementos receptores de la señal de reloj, CSEs, y los parámetros que relacionan y condicionan el funcionamiento de dichos elementos síncronos con respecto a la señal de reloj recibida. Se han analizado las dos grandes familias de elementos síncronos, los controlados por nivel o los controlados por flanco de la señal de reloj, comentando las características de cada familia y las situaciones en que cada uno es más interesante.

CAPÍTULO 4

NUEVA ESTRATEGIA DE SINCRONIZACIÓN

En este capítulo se va a definir, analizar y caracterizar una estrategia de sincronización basada en la aplicación de varias señales de reloj levemente desfasadas que controlan los diferentes canales que componen los sistemas basados en el RNS. Esta estrategia de sincronización se va a comparar con la metodología de sincronización basada en una señal de reloj única, que es la más frecuente en la sincronización de sistemas digitales. Tras un estudio de las características de ambas estrategias, se van a aplicar a tres sistemas de prueba con diferentes frecuencias de funcionamiento, y bajo dos regímenes de operación: un funcionamiento normal con un patrón de entrada predefinido, y funcionamiento con entrada nula, de modo que se puedan realizar medidas de consumo dinámico y estático.

4.1. Introducción

Como ya se ha visto en el Capítulo 2, el RNS se ha empleado para obtener sistemas más rápidos que los basados en aritmética convencional, por su habilidad para realizar operaciones de suma y multiplicación de manera más eficiente que los sistemas tradicionales. La principal ventaja del RNS reside en su capacidad para realizar operaciones aritméticas sobre canales independientes sin propagación de acarreo entre canales. Esto permite realizar operaciones sobre un elevado rango dinámico, descomponiendo éste en una serie de canales independientes de menor rango dinámico, capaces de operar en paralelo. Sin embargo, las aplicaciones clásicas no han hecho un uso exhaustivo de esta propiedad de independencia entre canales; en concreto, esta independencia sólo ha sido aprovechada para la limitación de la propagación de acarreo, aumentando así las prestaciones de los sistemas. Pero esta independencia entre canales es completa, de manera que es posible sincronizar cada canal de manera independiente al resto; más concretamente, se puede sincronizar cada canal mediante una señal de reloj levemente desfasada respecto al resto de canales, siendo solamente necesaria la sincronización global a la salida de datos de todos los canales. De este modo se consigue un funcionamiento distribuido, que reduce los picos de corriente en las líneas de alimentación, y evita la necesidad de la sincronización global del sistema a lo largo de todo el circuito integrado.

En este capítulo se presentará una estrategia para plasmar este nuevo esquema de sincronización para sistemas basados en el RNS mediante diferentes señales de reloj. Dichas señales desfasadas se generarán a partir de la señal de reloj global mediante las celdas parametrizables `dCLK_cell`. Una cadena de estas celdas propaga la señal de reloj con un retardo controlado en cada una de ellas. Además, cada celda es capaz de generar de modo adecuado la señal de reloj que va a sincronizar cada uno de los canales RNS del sistema. Se estudiará en detalle el funcionamiento y las características del consumo y retardo de las celdas `dCLK_cell`.

Posteriormente se van a realizar una serie de simulaciones sobre sistemas realistas, sincronizados mediante una red de distribución de reloj tradicional y mediante esta estrategia de sincronización, todas ellas a diferentes frecuencias y en diferentes

condiciones de funcionamiento. Previamente a la realización de las simulaciones se ha realizado un estudio en profundidad de la red de distribución del reloj, de modo que se han dimensionado los elementos de dicha red para producir las señales de reloj más adecuadas para la sincronización de los sistemas. El análisis de los resultados mostrará las ventajas en la aplicación de la estrategia analizada. Se compararan los resultados de los diferentes sistemas para tener una visión global de los efectos de la aplicación de la estrategia de sincronización en estudio.

4.2. Metodología alternativa de sincronización de sistemas basados en RNS

Los sistemas basados en el RNS realizan el procesamiento de los datos transformando éstos en sus residuos y operando cada uno de los canales independientemente y en paralelo con los demás. Esta independencia de canales se ha aprovechado para conseguir sistemas más rápidos, aprovechando la reducción del rango dinámico en cada canal respecto del sistema total. El rendimiento de un sistema convencional se degrada notablemente a medida que la longitud de palabra aumenta; por ejemplo en un sumador convencional se requiere que el acarreo se propague a lo largo de toda la longitud de palabra hasta obtener el resultado correcto y, en el caso de los multiplicadores, la complejidad es proporcional al cuadrado de la longitud de palabra. Sin embargo, la independencia entre los canales que procesan cada residuo es completa, por lo que cada canal puede funcionar con una señal de reloj diferente a los demás, a condición que a la entrada se capturen los datos de modo sincronizado con el exterior y a la salida se restablezca la sincronización al componer el resultado a partir de sus residuos.

Por otro lado, y tal como se vio en el Capítulo 3, la propagación de la señal de reloj a lo largo de un circuito es una tarea tanto más compleja cuanto más extenso es el circuito y mayor es la frecuencia. Uno de los principales problemas en la propagación de la señal de reloj es el *skew*, es decir, el desfase que experimenta la señal de reloj en diferentes lugares del sistema. Los principales efectos del *skew* son dos: el posible fallo del sistema, al no estar bien definido el instante en el que ocurre el flanco de reloj, y la disminución de la frecuencia de funcionamiento, debido al incremento del periodo de la

señal de reloj por los márgenes de seguridad necesarios para asegurar la correcta sincronización del sistema.

4.2.1. Requisitos para la aplicación de la nueva estrategia de sincronización

El reto principal de esta tesis consiste en combinar las dos ideas anteriores, de modo que un sistema formado por canales totalmente independientes sea sincronizado por señales de reloj afectadas por un *skew* controlado por el diseñador. Esta tarea requiere:

- del diseño de una red de distribución de la señal de reloj capaz de generar varias señales de reloj, a partir de una única señal de referencia, entre las que debe existir un leve desfase controlado por el diseñador;
- del diseño de sistemas RNS que, a pesar de funcionar de un modo distribuido en el espacio y el tiempo, mantengan la coherencia aceptando datos de entrada y generando datos de salida de un modo totalmente sincronizado con el resto del sistema.

El primer requisito se puede alcanzar mediante el uso de las celdas parametrizables `dCLK_cell`, capaces de transmitir una señal de reloj, produciendo un retardo en dicha propagación y, al mismo tiempo, generar señales de reloj de calidad adecuada para sincronizar cada uno de los canales del sistema RNS. A lo largo de este capítulo se analizará en detalle la estructura, el funcionamiento, las prestaciones y cómo se diseñan estas celdas para controlar el retardo y generar las señales reloj adecuadas.

En cuanto al segundo requerimiento del diseño de un sistema globalmente síncrono, pero capaz de funcionar con señales de reloj internas desfasadas, los sistemas basados en el RNS tienen una estructura que permite de modo natural este tipo de sincronización. El esquema general de un sistema RNS consiste en una conversión inicial de la entrada binaria en n residuos, n canales de procesamiento paralelo de cada residuo, y una etapa final de conversión de los residuos en dato binario. La captura del

dato binario de entrada se realiza mediante la señal de reloj desfasada más adelantada, de modo que, a medida que ocurran los flancos de las señales levemente desfasadas posteriores, el dato ya esté disponible para su conversión al residuo correspondiente por parte de cada canal. La generación de todos y cada uno de estos residuos ocurre dentro de un ciclo de reloj por lo que, cuando llega el dato siguiente, la transformación del dato anterior en sus residuos ya se ha completado. El funcionamiento completo de cada canal está sincronizado con cada una de las señales desfasadas, tal como ya se ha dicho. Por último, la conversión de los datos de salida de cada canal en un dato binario se realiza en un módulo controlado por la señal de reloj más retrasada, de modo que cada canal ya ha producido su resultado cuando el módulo final realiza la conversión. Este esquema de funcionamiento requiere que todas las señales desfasadas estén dentro del ciclo de reloj, de manera que la cadencia de salida de datos sea la misma que la de la entrada y el circuito basado en RNS esté globalmente en sincronía con el resto del sistema.

4.2.2. Características de las señales de reloj levemente desfasadas

Las señales de reloj levemente desfasadas que van a sincronizar un sistema RNS deben cumplir los siguientes requisitos:

- los flancos de estas señales deben estar lo suficientemente separados como para que el solapamiento de las respuestas de cada canal al flanco correspondiente sea mínimo;
- los flancos de las señales deben estar lo bastante próximos como para que la señal más retrasada permita el procesamiento del dato a lo largo del canal y que el resultado esté disponible, de modo que se respete la correspondencia de datos a la salida de los diferentes canales;
- la calidad de todas las señales debe ser adecuada para actuar como señal de reloj.

Las ventajas que se obtienen de la aplicación de esta estrategia de sincronización son las siguientes:

- al estar todos los dispositivos gobernados por diferentes señales de reloj, su respuesta no es simultánea, con lo que se reducen los picos de corriente en las líneas de alimentación del circuito;
- como consecuencia de lo anterior, las dimensiones de las líneas de alimentación se pueden escalar, simplificando la labor de diseño;
- las especificaciones de diseño de las fuentes de alimentación se simplifican debido a que disminuye la cantidad máxima demandada de corriente, así como el ritmo de variación de la misma;
- se elimina la distribución de una señal global por todo el chip; por el contrario, cada canal es gobernado por su propio reloj, con lo que sólo es necesario distribuir cada uno de estos relojes de manera local en el canal correspondiente;
- una correcta estrategia de asignación de las primeras señales desfasadas a los canales más lentos puede incrementar la velocidad del sistema en los casos en los que las etapas de salida de los canales tengan diferentes tiempos de procesamiento;
- el hecho de existir canales independientes e independientemente sincronizados facilita la testeabilidad de sistema, ya que cada canal realiza un procesamiento totalmente independiente de los demás.

Los recursos necesarios para conseguir las señales de reloj que posibilitan la aplicación de la estrategia de sincronización descrita son muy sencillos, por lo que esta técnica de sincronización va a permitir unas mejoras significativas en los sistemas en los que se va a aplicar, con un coste mínimo. Como ya se ha dicho, la red de distribución de la señal de reloj que posibilita la implementación de esta estrategia de sincronización se basa en la celda parametrizable `dCLK_cell`. En los siguientes apartados se realizará un análisis exhaustivo de dicha celda y se caracterizarán sus prestaciones, su consumo y el modo en el que se puede configurar para producir el retardo requerido y, al mismo

tiempo, generar las señales de reloj desfasadas con la calidad adecuada para sincronizar la parte del sistema correspondiente.

4.2.3. Generación de las señales desfasadas: celda parametrizable `dCLK_cell`

La generación de varias señales levemente desfasadas se puede conseguir mediante una cadena de inversores. El retardo asociado a esta cadena se controla alternando inversores de tamaño configurable con inversores de tamaño mínimo, de modo que exista un retardo apreciable y, sobre todo, controlable en la propagación de la señal de reloj a lo largo de la línea. Las señales intermedias obtenidas no tienen la calidad ni la energía suficiente para actuar como señales de reloj, por lo que se requiere otro inversor que restaure el perfil de la señal y proporcione suficiente corriente como para actuar sobre el sistema que debe sincronizar. La Figura 4.1 muestra esta estrategia de sincronización [GON02a, GON05]. El inversor 1 (de entrada) tiene una capacidad de entrada configurable por el diseñador mediante las dimensiones de los transistores que lo forman. El inversor 2 (de salida) es de dimensiones mínimas por lo que, por un lado conmuta rápidamente la señal que establece el inversor 1 previo pero, por otra parte, proporciona una cantidad de corriente limitada al inversor 1 de la etapa posterior cuya capacidad de entrada es significativamente mayor. El resultado es que la transición de estado de los inversores 1 se ve ralentizada por los bajos niveles de corriente suministrados por los inversores 2 de la etapa anterior, generándose de este modo un retardo en la propagación de la señal de reloj que se transmite a lo largo de la línea. Las

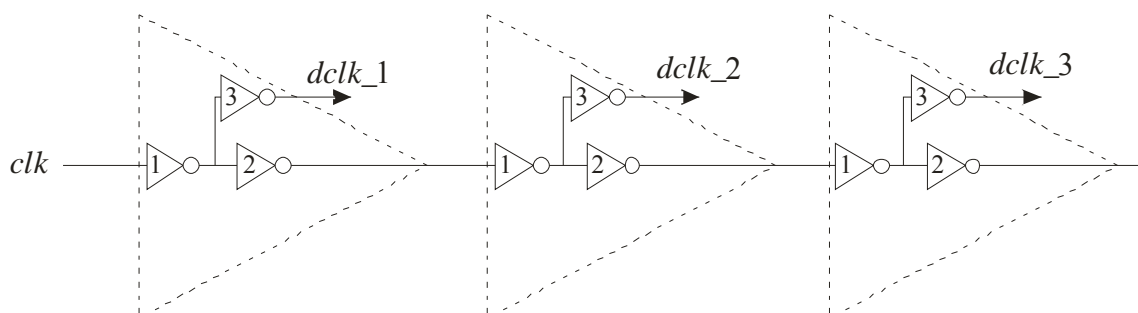


Figura 4.1. Esquema de línea de generación de las señales de reloj desfasadas:

- (1) inversor de entrada, con una gran capacidad de entrada;
- (2) inversor de salida, de dimensiones mínimas;
- (3) inversor restaurador, genera la señal de reloj efectiva.

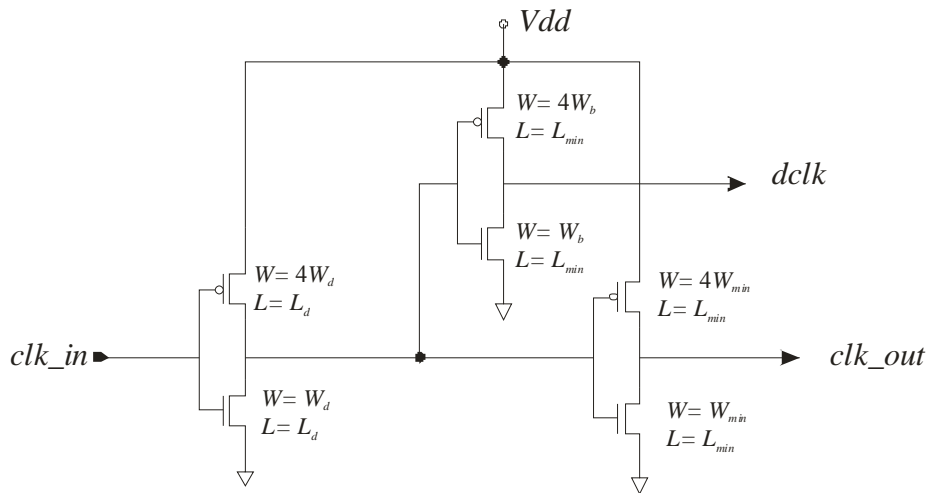


Figura 4.2. Detalle de la celda parametrizable dCLK_cell.

señales así generadas no son apropiadas para actuar como señal de reloj, por lo que es necesaria la inclusión del inversor 3 (restaurador), que restaura y acondiciona las señales para su aplicación a las diferentes regiones del circuito que sincronizan. Este inversor se puede configurar para adaptarlo a las necesidades de los circuitos que debe controlar, por lo que se requiere un análisis previo del sistema para establecer los valores adecuados en el tamaño de los transistores que forman este inversor de salida. Con este esquema de funcionamiento se tienen las señales *dclk_1*, *dclk_2* y sucesivas, que están levemente desfasadas entre sí y son las que se aplican efectivamente al sistema como señales de reloj.

La Figura 4.2 muestra en detalle la celda dCLK_cell con las dimensiones de los transistores que la constituyen indicadas. La capacidad de entrada del primer inversor se establece mediante los parámetros L_d y W_d que, como ya se ha explicado, condicionan el retardo producido en la celda. El inversor 2 se construye con las dimensiones mínimas que permita la tecnología (W_{min} y L_{min}) con objeto que se requiera un tiempo apreciable para cargar el inversor 1 de la celda siguiente. Por último, el parámetro W_b representa la anchura del canal del transistor NMOS del inversor de salida y su valor está condicionado por el canal que debe sincronizar. La alimentación de la celda (Vdd) es independiente del resto del sistema, de modo que se puede realizar un análisis por separado del consumo de la cadena de celdas dCLK_cell, o bien integrar éstas dentro del sistema completo. Esta posibilidad permite un estudio muy detallado del impacto en el consumo de la aplicación de esta estrategia de sincronización. En todo

caso, y para asegurar la igualdad en los flancos de subida y bajada de las señales que transmite la celda, se establece la relación 4 entre las anchuras de los canales de los transistores NMOS y PMOS, motivada por la diferencia entre las movilidades de los portadores de carga en ambos transistores y analizada en el Apéndice B. Dicha relación es aplicable a los transistores utilizados para la realización de este trabajo y debe ser calculada para las simulaciones realizadas en otros procesos.

El proceso de diseño de la celda comienza estableciendo el parámetro W_b , que está condicionado por el canal que se va a sincronizar y cuyo valor suele ser elevado debido al gran *fan-out* de esta salida de la celda. En segundo lugar se determina el retardo de la celda comenzando por el valor de L_d ya que, como se verá más adelante, su contribución al retardo es la más significativa. Para ello es necesario tener en cuenta el valor de W_b ya que, según se muestra en las Figuras 4.1 y 4.2, el inversor 1 debe cargar a los inversores 2 y 3, siendo el inversor 3 de dimensiones considerablemente superiores al inversor 1. Esto genera un retardo adicional en la cadena de inversores por lo que el valor L_d está condicionado por el valor W_b . Por último, se elige el valor de W_d para realizar el ajuste fino del retardo requerido. Todos estos aspectos y la cuantificación del efecto de cada uno de estos parámetros en el retardo se van a estudiar en detalle en los siguientes apartados.

4.2.4. Análisis de la celda parametrizable `dCLK_cell`

La finalidad de la cadena de celdas `dCLK_cell` es doble; por un lado producir un retardo en la señal de reloj, y por otra parte generar de modo adecuado las diferentes señales desfasadas para sincronizar los diferentes canales en que se ha desglosado un sistema. En la Figura 4.3 se muestra un sistema de prueba formado por cinco celdas `dCLK_cell`, cada una de ellas con una carga de cinco registros de ocho bits, y también se ilustran las señales de reloj levemente desfasadas generadas para dicho sistema. Se ha elegido este sistema de prueba por dos motivos: en primer lugar la regularidad tanto del sistema y como del patrón de entrada asegura que las medidas realizadas no dependen de la actividad del sistema ni de variaciones en el flujo de datos a lo largo del mismo; por otro lado, la elección de cinco canales asegura una propagación de la señal de reloj de referencia a lo largo de un número representativo de celdas `dCLK_cell`. Se ha

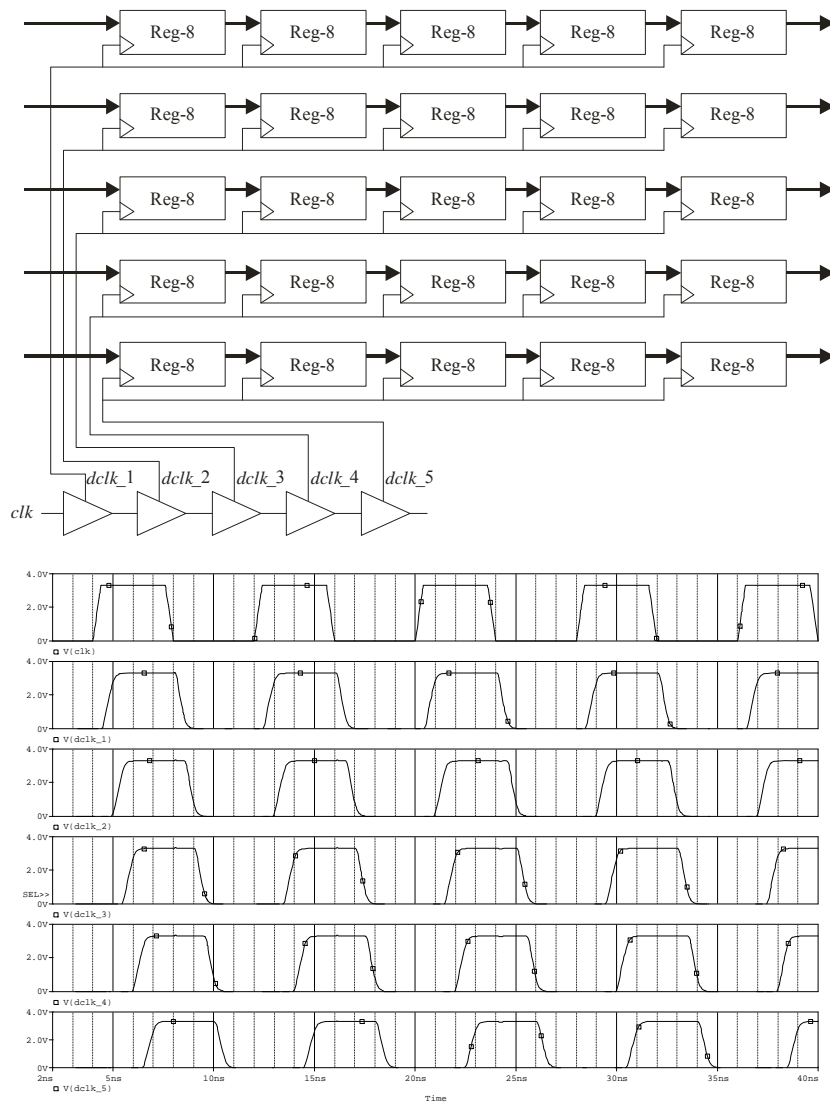


Figura 4.3. Representación de la señal de reloj de entrada original junto con las señales levemente desfasadas.

aplicado una señal de reloj ideal (*clk*) a la entrada de la cadena de celdas y a partir de ella se han generado las cinco señales (*dclk1* a *dclk5*) que actúan sobre cada canal de registros. La señal de reloj original se muestra en primer lugar y las cinco señales restantes corresponden a cada una de las cinco señales de reloj desfasadas. En la gráfica queda patente tanto el retardo producido en cada señal de reloj, como la leve degradación de su perfil debida a la carga aplicada a la salida de cada celda *dCLK_cell*, lo cual supone una situación realista de funcionamiento. Nótese cómo todas las señales levemente desfasadas están contenidas dentro del mismo ciclo de reloj de la señal original.

En todas las aplicaciones se debe tener en cuenta la relación entre el retardo aplicado, el número de etapas y el periodo de la señal de reloj, relación que se debe respetar para que el sistema funcione correctamente. Con esta metodología de sincronización, cada canal funciona independientemente; sin embargo, es necesaria la sincronización simultánea de los datos a la salida de todos los canales. Esto supone que todas las señales desfasadas deben estar comprendidas dentro de un mismo ciclo de reloj, ya que de no ser así serían necesarios más de un ciclo de reloj para adquirir los datos de salida. El máximo retardo permisible por etapa $t_{d_máx}$ para un sistema con N etapas y cuyo periodo de reloj es T viene dado por la expresión:

$$t_{d_máx} \leq \frac{T}{N} \quad (4.1)$$

En el caso de no cumplirse esta condición, la adquisición de los datos a la salida de los canales requerirá de dos o más ciclos de reloj, con lo que las prestaciones del sistema se verán afectadas. La expresión (4.1) asume que el retardo en todos los canales va a ser el mismo, aunque esta condición no ha de satisfacerse en sistemas en los que los canales fueran diferentes. Es posible que, en un sistema general, determinados canales requieran más tiempo para realizar el procesamiento de los datos. En esas situaciones se pueden modificar los retardos de modo que (4.1) no se cumpla en todas las señales aunque el incremento adicional de unas señales debe ser compensado por otras. De esta forma se proporciona más tiempo de procesamiento a los canales más lentos, aunque globalmente todas las señales deben estar dentro del mismo ciclo de reloj. Otra alternativa es asignar las primeras señales desfasadas a los canales que más tiempo requieran y sincronizar la salida de datos con la última señal desfasada. En definitiva, se utiliza el *skew* negativo para incrementar la velocidad del sistema (tal como se vio en el Capítulo 3) que, de otra manera, tendría que funcionar con un periodo de reloj mayor. A continuación se estudiarán los efectos de los diferentes parámetros en el funcionamiento de las celdas `dCLK_cell`.

4.2.4.1. Efecto de L_d y W_d

Como antes se ha dicho, los parámetros L_d y W_d controlan el retardo generado por la celda `dCLK_cell`. Para cuantificar el efecto que tienen estos parámetros sobre el retardo, se han realizado una serie de simulaciones sobre el mismo sistema de la Figura 4.3 en las que se han modificado L_d y W_d , y se ha medido el retardo generado entre dos

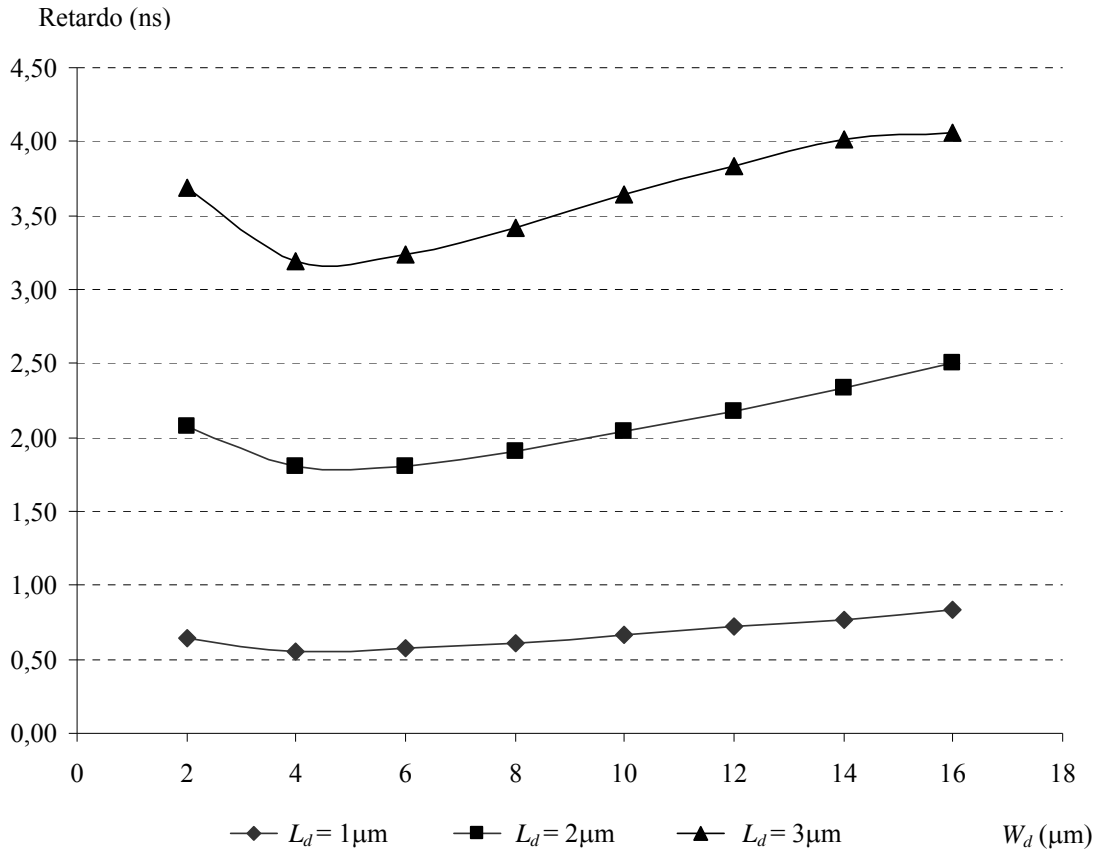


Figura 4.4. Retardo por celda en función de L_d y W_d .

señales de reloj *dclk* consecutivas. Los resultados obtenidos se han representado en la Figura 4.4. Como se puede apreciar en la gráfica, las variaciones en L_d influyen más significativamente en el retardo generado que las de W_d . Este resultado era previsible: un incremento en la anchura o en la longitud de la puerta de los transistores que forman el inversor de entrada supone un incremento de la capacidad de entrada de la celda `dCLK_cell`; sin embargo, si ese aumento se produce en la anchura de la puerta, la capacidad de conducción de los transistores del inversor de entrada se ve incrementada, con lo que este inversor conmuta rápidamente y, además, suministra grandes cantidades de corriente al inversor restaurador de la señal (el que genera la señal `dCLK`), que también cambiará de estado rápidamente. En estas condiciones, el retardo se produce mayoritariamente en el inversor de entrada. Por otro lado, un incremento en la longitud de la puerta, además de incrementar la capacidad de entrada, disminuye la conductividad de los transistores y, al disminuir la corriente que suministra el inversor de entrada, la conmutación de dicho inversor se hace más lenta, con lo que el retardo aumenta. Además, al suministrar el inversor de entrada bajos niveles de corriente

también se ve ralentizada la conmutación del inversor restaurador y, en menor medida, el inversor de salida. En esta situación el retardo es la suma del retardo del inversor de entrada más el producido en el inversor restaurador, ya que el inversor de salida es de dimensiones mínimas y su aportación por conmutación al retardo es muy baja. Este comportamiento permite una amplia gama de combinaciones en la selección del retardo. Ajustando el valor de L_d se establece el retardo de manera aproximada, mientras que con el parámetro W_d se puede ajustar de manera más precisa. La Figura 4.4 también muestra un efecto de incremento de retardo para valores bajos de W_d . El motivo es el explicado anteriormente: un valor muy bajo de W_d tiene como resultado una menor conductividad del inversor de entrada que se traduce en que el inversor restaurador tarde más en modificar su estado y el retardo se acumule adicionalmente en el inversor restaurador.

Existe un límite superior para la selección de los valores de W_d y L_d ; si el producto $W_d \times L_d$ alcanza un valor excesivo, determinado por la frecuencia de funcionamiento y por las características propias de la celda, el inversor de salida de una etapa no es capaz de cargar el inversor de entrada de la etapa siguiente a tiempo, y la señal de reloj no se transmite a lo largo de la cadena de celdas. En estas condiciones, una o varias partes del circuito quedarían sin sus correspondientes señales de reloj. Por todo lo explicado anteriormente, desde el punto de vista del diseño interesa tomar el valor de L_d lo menor posible y ajustar el retardo con W_b .

4.2.4.2. Efecto de W_b

Como se ha mostrado anteriormente, el parámetro W_b controla la cantidad de corriente que se suministra a la parte del sistema que va a estar sincronizada por la salida $dCLK$ de una celda $dCLK_cell$ determinada. La elección del valor de W_b debe cumplir un doble compromiso:

1. valores pequeños de W_b proporcionan señales de reloj de baja calidad con excesivos tiempos de subida y bajada;
2. valores elevados de W_b producen señales de reloj de buena calidad pero aumentan el retardo en la celda $dCLK_cell$ e incrementan el consumo.

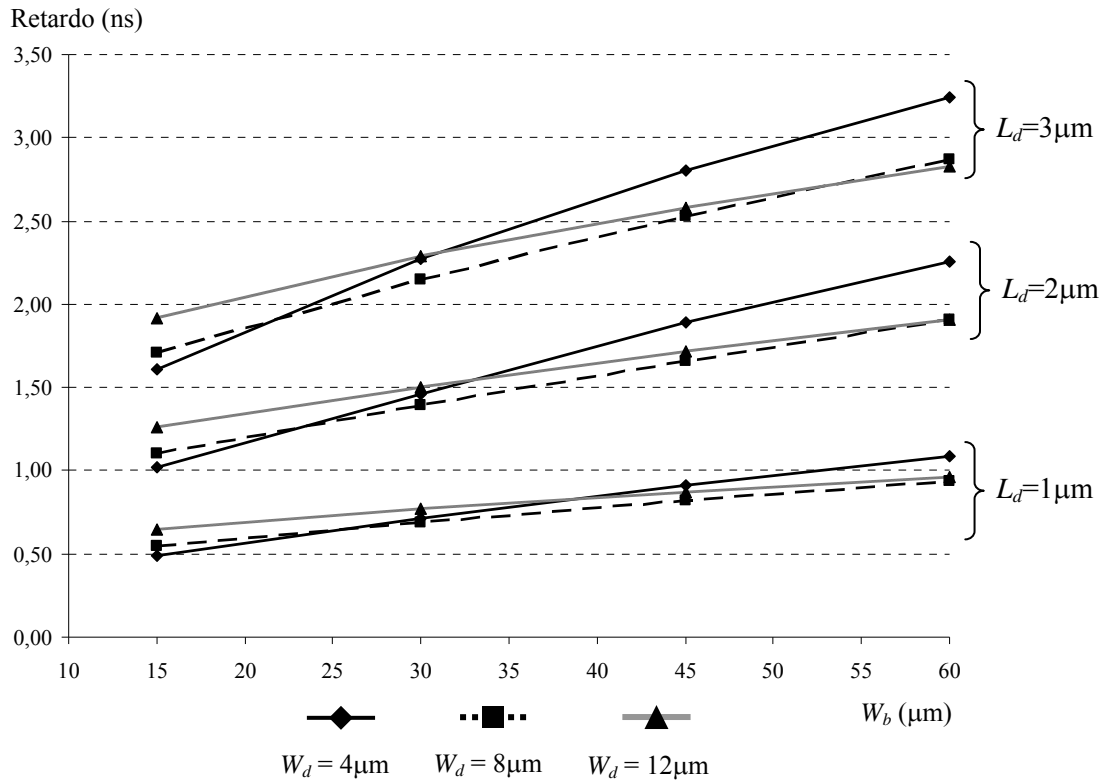


Figura 4.5. Retardo en función de L_d , W_d y W_b .

Así pues, la elección de este parámetro queda principalmente determinada por las características del canal que se va a controlar y, además, es necesario tener en cuenta el efecto que tendrá sobre el retardo generado por la celda.

Desde el punto de vista del retardo, el valor de L_d se debe fijar teniendo en cuenta el valor de W_b obtenido. Un análisis del retardo generado por la celda parametrizable en función de W_b , para diferentes valores de L_d y W_d , se muestra en la Figura 4.5, donde se ha representado el retardo generado en una cadena de celdas `dCLK_cell` actuando sobre varios sistemas de prueba formados por cuatro canales de registros de 8 bits sincronizados por cuatro celdas `dCLK_cell`. Las simulaciones se realizan sobre diferentes sistemas con canales de 4, 8, 12 y 16 registros de 8 bits, que suponen la carga adecuada a los valores tomados para W_b : 15 μm , 30 μm , 45 μm y 60 μm , respectivamente. Para cada uno de estos valores de W_b se han fijado tres valores de L_d (1 μm , 2 μm y 3 μm) y otros tres para W_d (4 μm , 8 μm y 12 μm) de modo que en la Figura 4.5 se puede obtener el retardo generado para cada configuración de los tres parámetros

que definen la celda `dCLK_cell`. Se ha medido el desfase entre la segunda y tercera señales de reloj, ya que ambas celdas están cargadas y cargan otras celdas similares y, por lo tanto, su funcionamiento es el típico de una celda intermedia de una cadena. La Figura 4.5 muestra cómo efectivamente un aumento en W_b incrementa el retardo producido en la celda. Este efecto es tanto mayor cuanto menor es W_d y cuanto mayor es L_d , es decir, W_b afecta al retardo generado por la celda tanto más cuanto menor es la cantidad de corriente que el inversor de entrada es capaz de proporcionar, tal como ya se ha explicado.

4.2.4.3. Análisis del consumo

Como conclusión al estudio de la celda `dCLK_cell`, se va a analizar la repercusión de los parámetros que la describen en su consumo. Para ello se ha tomado un sistema de prueba formado por tres canales de registros de 8 bits. Se ha considerado la celda central y se ha alimentado con una fuente de tensión independiente, de modo que la corriente suministrada por esta fuente alimenta exclusivamente a la celda en estudio. La cantidad de registros de los canales (4, 8 y 12) está en función del valor del parámetro

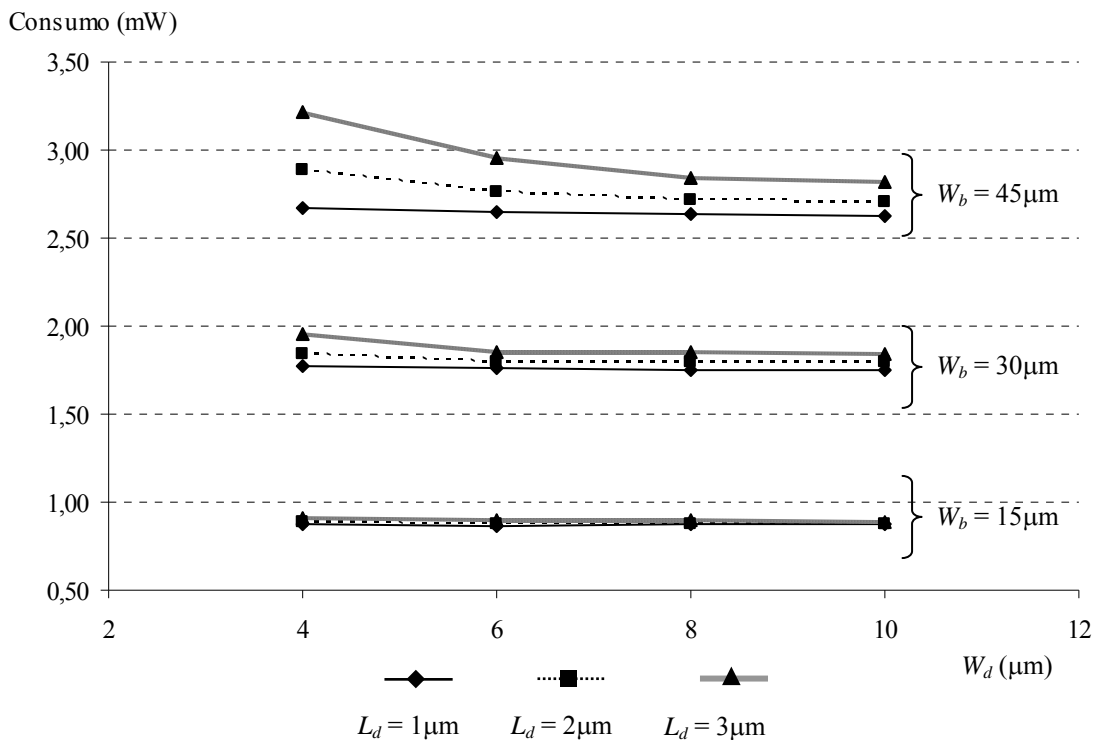


Figura 4.6. Consumo de la celda parametrizable en función de L_d , W_d y W_b .

W_b (15 μm , 30 μm y 45 μm), al igual que en las simulaciones realizadas en el apartado anterior. La Figura 4.6 muestra el consumo de una sola celda en función de L_d , W_d y W_b . Como se puede apreciar, el consumo está determinado principalmente por el valor de W_b , ya que este parámetro define las dimensiones del transistor de mayor tamaño de la celda debido a los elevados valores de *fan-out* a su salida. Para valores pequeños de W_b , las variaciones en L_d y W_d apenas afectan al consumo, ya que el inversor de entrada no es cargado en exceso por el inversor restaurador. A medida que el valor de W_b aumenta, el inversor de entrada debe proporcionar más corriente para hacer conmutar al inversor restaurador y, cuanto mayor es L_d o menor es W_b , es decir, cuanto menor es la conductividad del inversor de entrada, más progresiva es la variación de la entrada del inversor restaurador y más tiempo permanece en cortocircuito, por lo que el consumo es mayor. A medida que W_d aumenta, el consumo tiende a disminuir hasta un valor estable por el incremento en la conductividad del inversor de entrada.

Desde el punto de vista del consumo interesa que el valor de L_d sea el menor posible y el de W_b el mayor posible, de modo que el inversor restaurador realice transiciones rápidas y definidas. La Figura 4.6 también muestra que el consumo de cada una de ellas oscila entre 0.8 y 3.3mW, en función principalmente del parámetro W_b , que está relacionado con la cantidad de dispositivos que deba sincronizar la celda.

4.3. Diseño y aplicación de las estrategias de sincronización a tres sistemas

En este apartado se va a aplicar la estrategia de sincronización analizada a tres sistemas basados en el RNS. Los sistemas son un filtro CIC (*Cascade Integrator Comb*: integradores y peines en cascada) para diezmado de Hogenauer [GON02b], un controlador PID (Proporcional Integral Diferencial) [GON05] y un sistema para realizar la transformada *wavelet* unidimensional 1D-DWT (Transformada Discreta de *Wavelet*: *Discrete Wavelet Transform*) [RAM01, PAR03] por transformación al dominio de los índices. La selección de estos sistemas está motivada porque cada uno de ellos es de diferente complejidad, tamaño y con distintas características de distribución de la señal de reloj. De este modo se podrán comprobar los efectos de la aplicación de la nueva

estrategia de sincronización bajo diferentes escenarios de funcionamiento y distribución de la señal de reloj y los resultados obtenidos se podrán generalizar o matizar de un modo más objetivo. Todos los sistemas se han simulado bajo dos diferentes estrategias de sincronización: una señal de reloj única, distribuida mediante *buffers* amplificadores, y varias señales de reloj desfasadas generadas por una cadena de celdas `dCLK_cell`. De este modo se va a medir el efecto de la aplicación de la estrategia de sincronización propuesta, contrastando diversos parámetros de funcionamiento de los sistemas con respecto al esquema de sincronización convencional. Además, se han realizado otras dos simulaciones sobre todos los sistemas de prueba; la primera está orientada a comprobar el efecto de la aplicación de la estrategia de sincronización sobre el consumo estático del sistema, es decir, el consumo cuando no se aplica ninguna entrada de datos, y la segunda para analizar el consumo por separado de la red de distribución de la señal de reloj y el resto del sistema. Estas simulaciones permiten apreciar de un modo más detallado el impacto sobre el consumo del sistema de la circuitería necesaria para producir el efecto de *skew* controlado. Todas estas simulaciones se han realizado aplicando cuatro frecuencias de funcionamiento para apreciar el efecto de la frecuencia, sobre los parámetros analizados.

Previamente a la descripción y análisis de los sistemas de prueba se van a estudiar las características de la red de distribución de la señal de reloj que se requieren para la correcta sincronización de todo el sistema. Las características de los subsistemas encargados de distribuir la señal de reloj a lo largo de todo el circuito van a influir de modo notable en el funcionamiento y el rendimiento del mismo, por lo que la elección de un criterio objetivo para el diseño de la red de distribución de la señal de reloj es crucial para el estudio comparativo del impacto de la metodología de sincronización aplicada.

Posteriormente se van a describir cada uno de los sistemas de prueba; en primer lugar se realizará una introducción teórica sobre la arquitectura y el funcionamiento del sistema en cuestión, posteriormente se mostrará cómo ese sistema se ha adaptado para operar en el RNS, cómo se aplica la estrategia de sincronización en estudio y cómo se ha implementado para ser simulado mediante PSpice. Por último, se mostrarán y analizarán los resultados obtenidos mediante la simulación. En el análisis de los resultados se tomará como criterio general de referencia denominar 'CLK' a los

resultados obtenidos en el caso de una única señal de reloj, mientras que, los resultados derivados del caso de señal de reloj distribuída mediante la cadena de celdas `dCLK_cell`, serán referidos como ‘dCLK’. Del mismo modo, cuando se analice la corriente o el consumo relativo al sistema global será referenciado como ‘Sistema total’, se empleará el término ‘RDR’ (Red de Distribución del Reloj) para indicar resultados relativos a la lógica encargada de generar y propagar las señales de reloj a lo largo de todo el sistema y se empleará ‘Sistema sin la RDR’ para indicar el sistema sin incluir la red de distribución de la señal de reloj. En el Apéndice B se describen de modo detallado todos los modelos de librería creados para la realización de las simulaciones con PSpice, así como su funcionamiento.

4.3.1. Diseño de la red de distribución del reloj

Una de las decisiones más importantes en el diseño de un sistema VLSI son las características de la red de distribución de reloj. Como se ha visto en el Capítulo 3, la señal de reloj es crucial para el funcionamiento del sistema y afecta significativamente a las prestaciones y el consumo del mismo. Un perfil de la señal de reloj con una pendiente excesivamente suave tendrá como consecuencia una mayor incertidumbre en la conmutación de los dispositivos e incrementará el consumo del sistema debido a que mantiene durante más tiempo en estado de conducción a ambas fases en lógica CMOS de los circuitos sincronizados. Por otro lado, la obtención de un perfil más abrupto requiere el uso de *buffers* de mayor tamaño, con lo que el consumo de la red de distribución de reloj aumenta. En este apartado se va a definir un criterio objetivo para el diseño de la red de distribución del reloj, de modo que tanto el *buffer* amplificador como la celda `dCLK_cell` estén dimensionados según el mismo criterio y se puedan comparar ambas estrategias de sincronización objetivamente.

Para medir el efecto del perfil de la señal de reloj se ha realizado una serie de simulaciones sobre un sistema de prueba, en las que se ha modificado el tamaño del *buffer* que distribuye la señal del reloj, y se ha medido el consumo como indicador de la calidad de las transiciones de la lógica CMOS. El sistema de prueba consiste en dos columnas de 12 registros de 8 bits, sincronizados por una única señal de reloj distribuida por un *buffer* amplificador (Figura 4.7). Se ha diseñado este sistema de modo que haya un número considerable de dispositivos, 2.112 transistores de los cuales 768 están

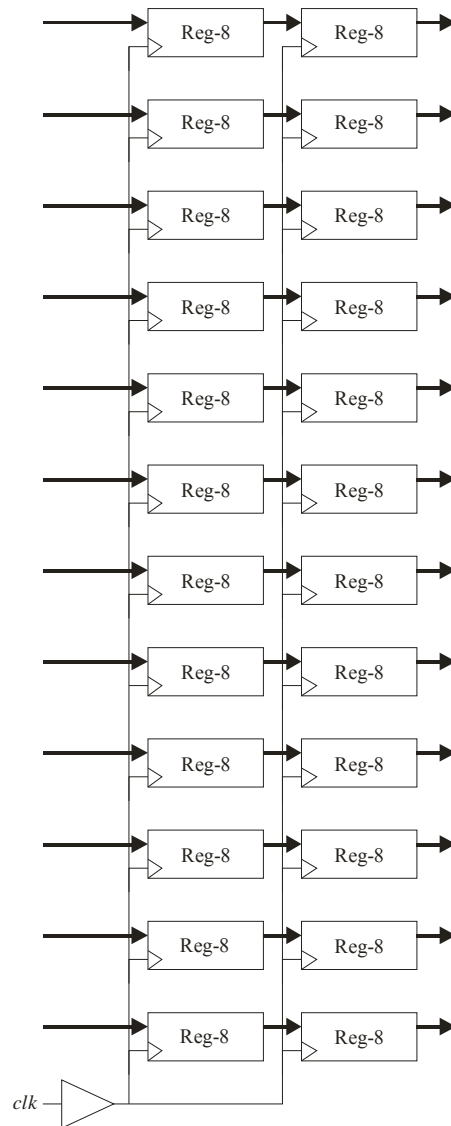


Figura 4.7. Sistema de prueba para apreciar el efecto del perfil de la señal de reloj sobre el consumo.

directamente controlados por la señal de reloj, para que se magnifiquen los efectos de consumo como consecuencia de las variaciones en la calidad de la señal de reloj. Además, se ha hecho predominar la dimensión vertical (12 registros en columna) sobre la horizontal (dos registros en profundidad) tal como se muestra en la Figura 4.7, para que los datos que se transmiten entre los biestables no sufran excesivos retardos de propagación que podrían enmascarar el efecto que se pretende medir, que es la dependencia del consumo con la calidad de la señal de reloj. El patrón de entrada de datos al sistema consiste en una secuencia que cambia todos los bits en cada ciclo de

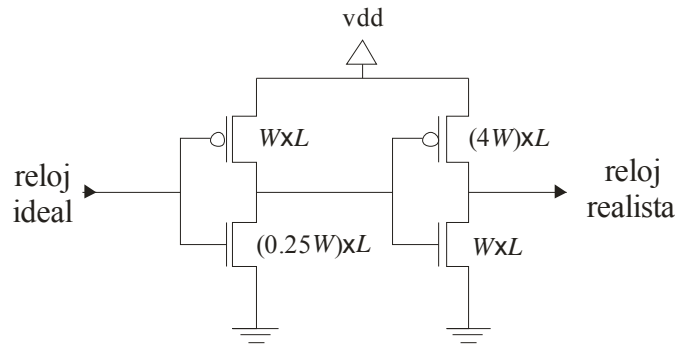


Figura 4.8. Buffer regenerador de señal de reloj.

reloj, de manera que todos los biestables cambien de estado en cada flanco de bajada, y el efecto de la calidad de la señal de reloj sobre el consumo del sistema sea máximo.

El *buffer* amplificador (Figura 4.8) que se empleará para modelar el esquema de sincronización convencional está constituido por dos inversores, de modo que las anchuras de los canales de los transistores que constituyen el inversor de salida sean cuatro veces mayores que las del inversor de entrada, ya que esa configuración permite la conmutación más rápida cuando se requiere que aumente la capacidad de conducción de corriente de una cadena de inversores [SUT99]. Esta característica hace adecuado a este tipo de *buffer* para amplificar la corriente requerida por las señales de reloj, con un *skew* inducido mínimo. Además, se ha mantenido la anchura de los transistores PMOS cuatro veces superior a la de los transistores NMOS, de modo que las pendientes de subida y bajada sean iguales, tal como se justifica en el Apéndice B. La longitud del canal de los transistores es mínima ($L=0.6\mu\text{m}$), mientras que las anchuras se pueden modificar mediante el parámetro W para adaptar el *buffer* al *fan-out* requerido. El modelo empleado en simulación permite alimentar el *buffer* mediante una fuente independiente del resto del sistema para, al igual que en el caso de las celdas `dCLK_cell`, poder medir el consumo propio de la red de sincronización independientemente del resto del sistema.

Si se representa el consumo global del sistema en función del tamaño del *buffer* se obtiene la curva mostrada en la Figura 4.9. Como se puede apreciar, el consumo aumenta a medida que W disminuye por debajo de $50\mu\text{m}$ y aumenta para valores superiores a dicho valor apreciándose cómo este incremento es más acentuado para

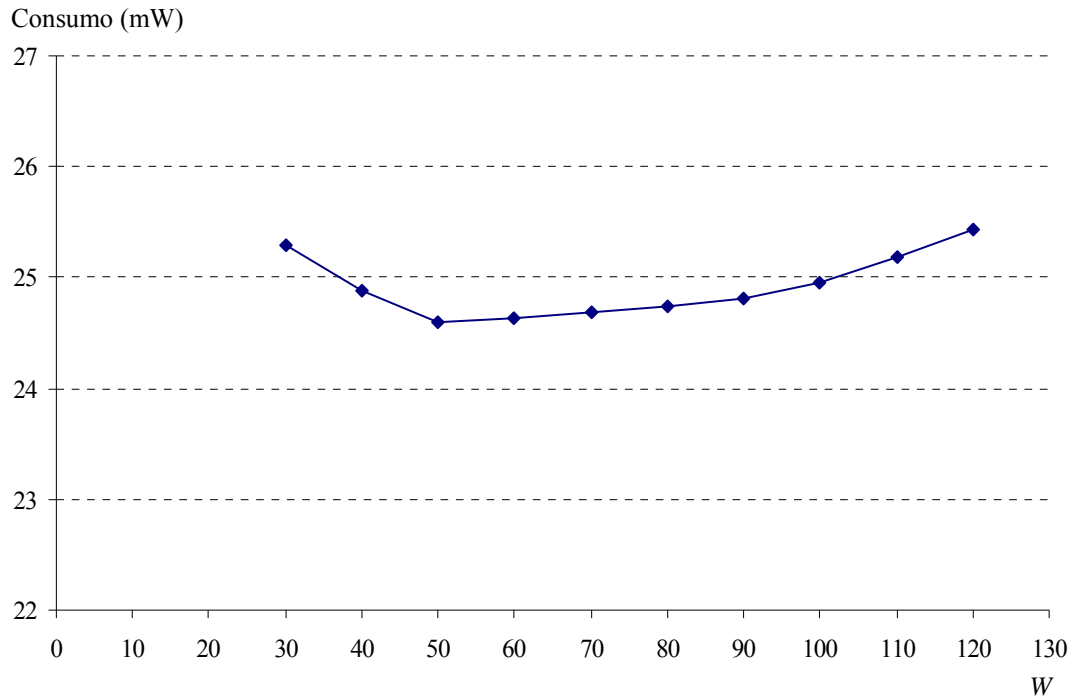


Figura 4.9. Consumo global de un sistema en función de la anchura del canal del *buffer*.

valores por encima de $90\mu\text{m}$. Entre $50\mu\text{m}$ y $90\mu\text{m}$ el consumo experimenta un ligero incremento inferior al 1%

Para explicar este comportamiento se han realizado las mismas simulaciones, pero analizando por separado los consumos del sistema y del *buffer* regenerador de la señal de reloj. Los resultados de las simulaciones quedan recogidos en la Tabla 4.1 y en la Figura 4.10. En la tabla se detallan el tamaño del parámetro W del *buffer*, los consumos separados del sistema y de la red de distribución de la señal de reloj, y los tiempos de subida y bajada de la señal de reloj junto con el perfil de la señal correspondiente. La Figura 4.10 muestra el consumo exclusivo del sistema en función del parámetro W . De los resultados obtenidos se pueden extraer una serie de conclusiones importantes:

- En primer lugar, para valores del parámetro W del *buffer* inferiores a $30\mu\text{m}$ se producen errores en la captura del dato por parte de los registros. Esto demuestra la importancia de la calidad de la señal de reloj con respecto al correcto funcionamiento de un sistema electrónico.



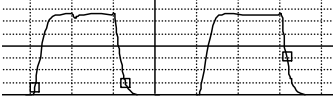
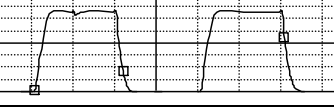
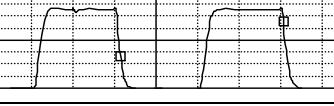
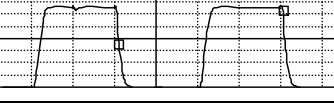
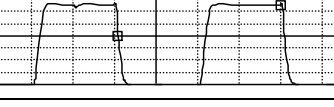
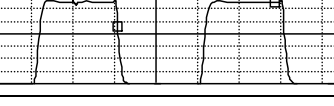
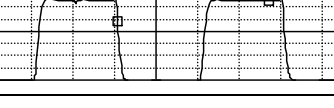
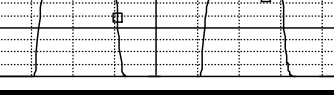
W_{buffer} (μm)	Consumo sistema (mW)	Consumo buffer (mW)	t_r (ns)	t_f (ns)	Perfil de la señal de reloj
30	27.49	12.08	0.958	0.872	
40	26.38	12.40	0.749	0.663	
50	25.83	12.67	0.623	0.551	
60	25.56	12.96	0.546	0.474	
70	25.41	13.24	0.459	0.400	
80	25.25	13.54	0.415	0.359	
90	25.01	13.79	0.364	0.308	
100	25.06	14.04	0.332	0.284	
110	25.07	14.30	0.310	0.262	
120	25.04	14.59	0.286	0.247	

Tabla 4.1. Consumo del sistema y de la red de distribución del reloj y perfil de la señal de reloj en función del tamaño de *buffer* regenerador de la señal de reloj.

- Para valores inferiores a $90\mu\text{m}$ el consumo del sistema es mayor cuanto menor es el tamaño del *buffer*. Esto se debe a que el flanco de la señal de reloj es poco abrupto y el tiempo de conmutación de la lógica, es decir, el tiempo que los

transistores PMOS y NMOS están en cortocircuito, se prolonga, por lo que el consumo aumenta.

- A partir de $90\mu\text{m}$ el consumo del sistema se estabiliza y deja de estar influido por la señal de reloj. La transición de estado de los biestables ahora depende fundamentalmente de su estructura interna y cualquier mejora en la calidad de la señal de reloj no afecta significativamente a su consumo. En cualquier diseño es importante llegar a este punto, puesto que para valores inferiores del parámetro W el sistema realiza transiciones de baja calidad, y para valores superiores la red de distribución está sobredimensionada y el consumo se ve incrementado.
- Una serie de simulaciones para valores de W en torno a $90\mu\text{m}$ demuestran que justamente para $W=90\mu\text{m}$ se obtiene el mínimo en el consumo del sistema en función a la anchura del canal del *buffer*, tal como se muestra en la Figura 4.10. Este resultado contrasta con el valor óptimo de $W=50\mu\text{m}$ desde el punto de

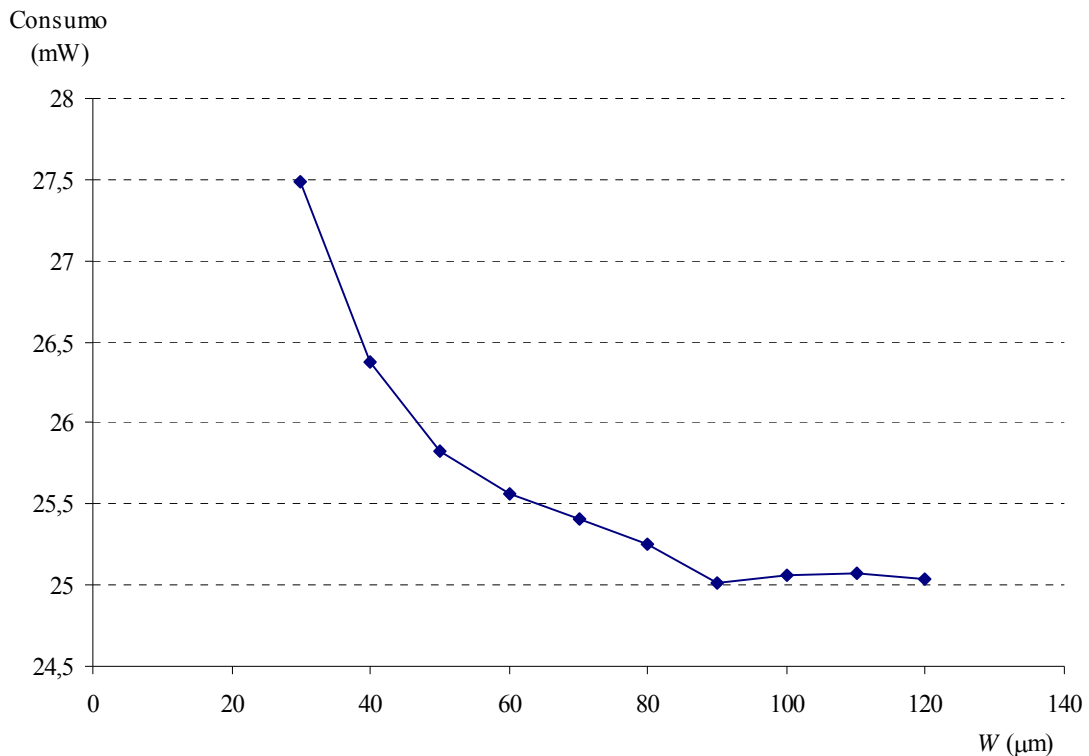


Figura 4.10. Consumo de un sistema, sin incluir la red de distribución de la señal de reloj, en función del tamaño del *buffer* que lo sincroniza.

vista del consumo global. La explicación es que entre $50\mu\text{m}$ y $90\mu\text{m}$ el sistema va reduciendo su consumo a medida que W aumenta, mientras que el *buffer* incrementa el suyo. El descenso en el consumo del sistema se ve levemente superado por el aumento de consumo del *buffer* entre los dos valores de W considerados.

- Los tiempos de subida (t_r) y de bajada (t_f) de la señal de reloj disminuyen significativamente (Tabla 4.1 y Figura 4.11) a medida que aumenta el tamaño del *buffer*, como consecuencia de la mayor conductividad de los transistores que lo constituyen. Dicha disminución es cada vez menor a medida que aumenta el parámetro W . Como ya se ha comentado antes, esta mejora en los tiempos de subida y bajada se consigue a costa del incremento del consumo del *buffer*, sin que se obtenga ninguna mejora en la conmutación de los biestables del sistema. También se puede apreciar cómo el tiempo de subida es superior al de bajada. Esto se debe a que la relación establecida en el Apéndice B para igualar las conductividades de los transistores NMOS y PMOS es ligeramente

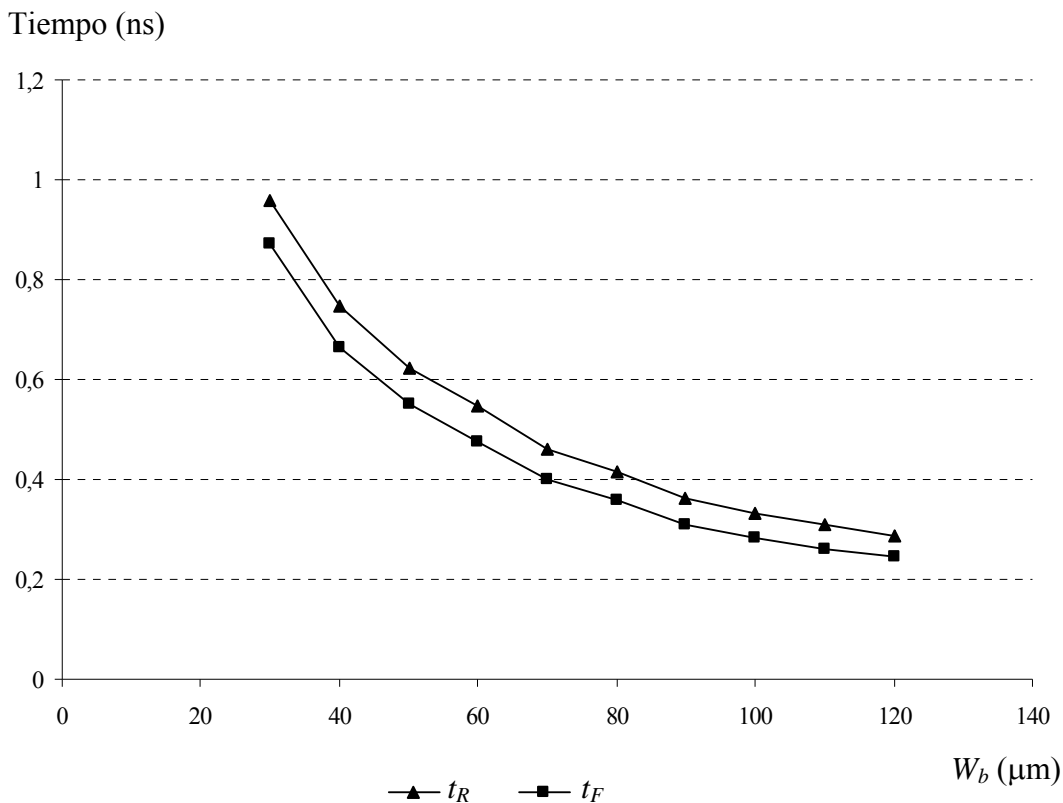


Figura 4.11. Tiempos de subida y bajada en función del tamaño del *buffer*.

superior a 4, y por lo tanto la conductividad de los transistores NMOS es levemente superior a la de los PMOS. A $90\mu\text{m}$ la diferencia es de 56ps.

- La Tabla 4.1 también muestra el perfil de la señal de reloj y su variación a medida que el parámetro W aumenta. Se puede apreciar cómo el efecto de mejora en el perfil de la señal se hace cada vez menos significativo a medida que aumenta el tamaño del *buffer*.
- Como cabía esperar, el consumo de la red de distribución de la señal de reloj aumenta monótonamente al aumentar el tamaño del *buffer* que la genera.

Como conclusión de todo lo anterior, se ha determinado que el tamaño del *buffer* más adecuado para sincronizar el sistema corresponde a $W=90\mu\text{m}$, que es el valor que optimiza las transiciones en la lógica CMOS, aunque no es el valor que minimiza el consumo global del sistema. Esta decisión se fundamenta en dos puntos:

1. Desde el punto de vista de la sincronización, la señal de reloj ideal es aquella que produce las transiciones más eficaces en velocidad y exactitud. Una transición lenta de la señal de reloj provoca incertidumbres en el momento de la conmutación de los dispositivos, con lo que se pueden ocasionar fallos de funcionamiento, se reduce la máxima frecuencia de operación y se incrementa el consumo de los dispositivos sincronizados.
2. La diferencia de consumo global entre el valor mínimo (para $W=50\mu\text{m}$) y el que se alcanza con $W=90\mu\text{m}$ es inferior al 1%, por lo que la penalización global es perfectamente asumible.

Desde un punto de vista puramente eléctrico, el *buffer* es un suministrador de corriente cuya misión es cargar, del modo más eficaz posible, las capacidades de puerta de los transistores sobre los que actúa. Partiendo de esta perspectiva, el diseño del *buffer* se va a realizar a partir de la carga, en términos de área, de las puertas de todos los dispositivos que la señal de reloj controla de modo directo. Para ello se va a establecer una relación entre el parámetro W del *buffer*, que determina la anchura de los inversores

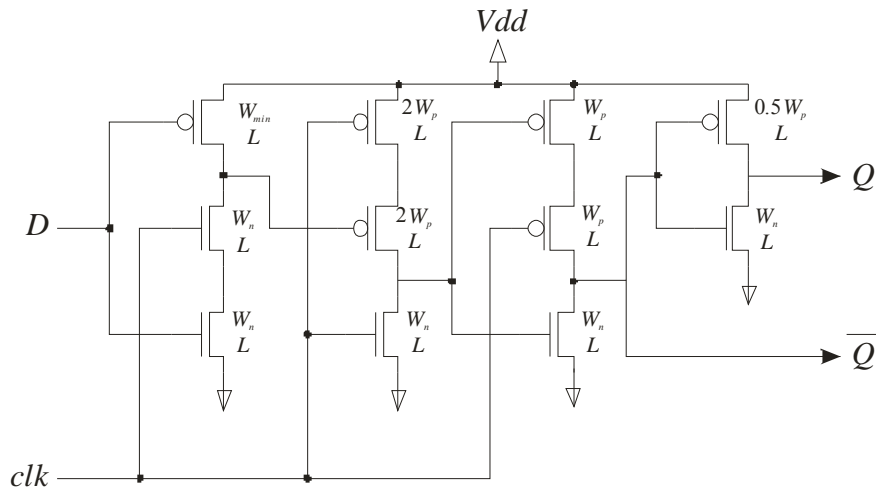


Figura 4.12. Biestable sincronizado por flanco de bajada.

que lo constituyen y, por lo tanto, está directamente relacionado con su conductividad, y el área de puerta de los dispositivos conectados a la señal de reloj.

El sistema de prueba descrito en la Figura 4.7 consta de 192 biestables que, como se ha demostrado, son sincronizados eficazmente con un *buffer* de $90\mu\text{m}$. Los biestables empleados en las simulaciones están constituidos por once transistores (Figura 4.12) cuatro de los cuales están controlados por la señal de reloj. La tabla 4.2 muestra el área total de puerta de estos transistores: 2 NMOS del mismo tamaño de puerta y 2 PMOS de dimensiones diferentes. El área total de puerta que debe ser cargada a través de la señal de reloj queda calculada en la Tabla 4.2 y resulta ser $13.44\mu\text{m}^2$.

	W (μm)	L (μm)	$W \times L$ (μm^2)
2 NMOS	1.6	0.6	1.92
1 PMOS	6.4	0.6	3.84
1 PMOS	12.8	0.6	7.68
	Total		13.44

Tabla 4.2. Área total de puerta de los transistores sincronizados por la señal de reloj de un biestable estándar.

La relación entre la longitud del canal del *buffer* y el área total de puerta óptimamente sincronizada con ese *buffer* resulta ser:

$$\frac{90\mu\text{m}}{192\text{ biestables}\times 13.44\mu\text{m}^2} = 0.03488\mu\text{m}/\mu\text{m}^2 \quad (4.2)$$

En las simulaciones realizadas sobre ejemplos de sistemas se empleará esta relación para determinar el valor del parámetro W de los *buffers* que deben sincronizar las diferentes partes del sistema. Más aún, tal como se ha visto en el apartado 4.2, la estrategia de sincronización basada en señales levemente desfasadas incorpora las celdas que, además de generar el desfase requerido, proporcionan a la señal de reloj resultante la corriente necesaria para sincronizar una parte del sistema total. Estas celdas emplean una etapa de salida formada por un inversor similar al *buffer*, por lo que la relación obtenida en (4.2) es plenamente aplicable a este estilo de sincronización.

4.3.2. Filtro CIC para diezmado de Hogenauer

El primer sistema de prueba consiste en un filtro CIC (*Cascade Integrator Comb*) para diezmado de Hogenauer [HOG81], que ha sido exitosamente adaptado al RNS [GAR99] y posteriormente sincronizado mediante señales levemente desfasadas [GON02b].

4.3.2.1. Descripción del sistema

Los filtros CIC de Hogenauer se caracterizan porque proporcionan filtros FIR empleando sólo sumadores y elementos de retardo; aunque la respuesta en frecuencia fuera de la banda de paso no es suficiente para algunas aplicaciones, esto se puede corregir mediante filtros convencionales de baja velocidad. La función de transferencia de un sistema CIC de S etapas referida a la frecuencia de muestreo superior viene dada por:

$$H(z) = \left(\frac{1 - z^{-RD}}{1 - z^{-1}} \right)^S = \left(\sum_{k=0}^{RD-1} z^{-k} \right)^S \quad (4.3)$$

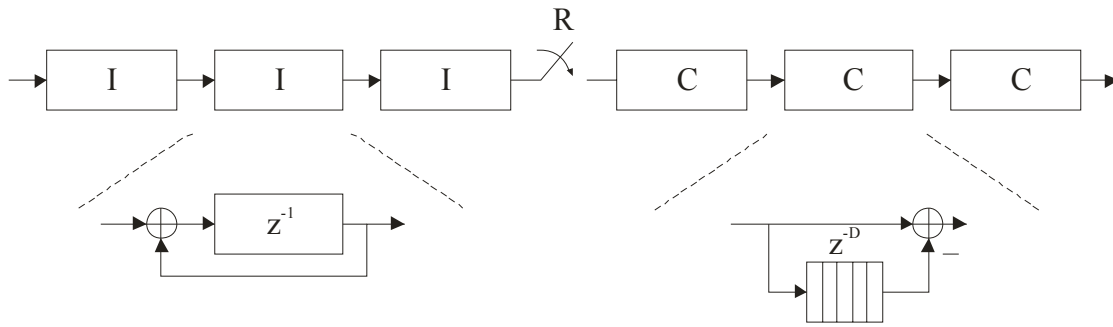


Figura 4.13. Filtro CIC de tres etapas.

La Figura 4.13 muestra un filtro CIC de tres etapas para diezmado; consta de tres integradores en cascada, una reducción de la frecuencia de muestreo en un factor R y tres filtros peine. La estructura del filtro para interpolación es equivalente, salvo que las etapas peine se anteponen a la cascada de integradores, aunque estos siguen funcionando a una frecuencia de muestreo superior en un factor R a la de las etapas peine.

La función de transferencia para un integrador de señales discretas se puede expresar como:

$$y[n] = x[n] + y[n-1] \quad (4.4)$$

El circuito que implementa esta operación en módulo m es el mostrado en la Figura 4.14a, donde se puede apreciar cómo un dato se suma con la salida anterior del integrador. Por otro lado, el peine requerido para la implementación del filtro CIC realiza la función siguiente:

$$y[n] = x[n] - x[n-D] \quad (4.5)$$

que se realiza con el esquema mostrado en la Figura 4.14b para $D=1$. Para otros valores

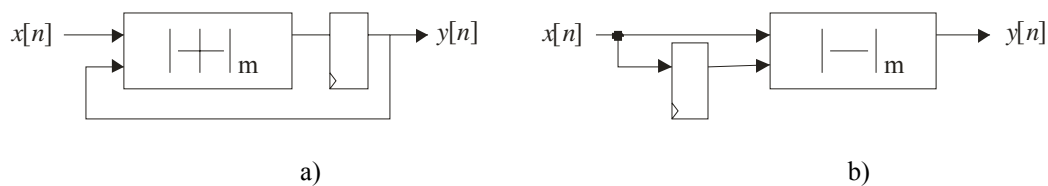


Figura 4.14. Esquema de un integrador (a) y un peine ($D=1$) (b) para el procesamiento de señales discretas basados en sumadores modulares.

de D se añaden tantos registros en serie a la entrada del sumador como indique dicho parámetro.

4.3.2.2. Implementación para simulación

Para un diseño efectivo de la estructura CIC es necesario calcular el máximo crecimiento de los datos. Se puede demostrar que para un filtro de tres etapas con $D=2$, $R=32$ y una entrada de 8 bits, un rango dinámico de 26 bits asegura que no se producirá desbordamiento durante el procesamiento. El sistema que se va a simular, que consta de alrededor de 30.000 MOSFETs, está formado por cuatro canales, de 8 bits el primero y de 6 bits los tres restantes, que son suficientes para asegurar dicho rango dinámico. El primer canal de 8 bits funciona en módulo 256 y los tres restantes de 6 bits son módulo 63, 61 y 59 respectivamente. La Figura 4.15 muestra el esquema general del filtro CIC. El primer módulo convierte un dato binario a su representación RNS y después cada uno de los residuos son procesados por tres etapas de integradores. Posteriormente, un reductor de frecuencia selecciona un dato de cada 32, que es enviado a tres etapas de peines. Finalmente los cuatro canales convergen en un módulo de conversión más

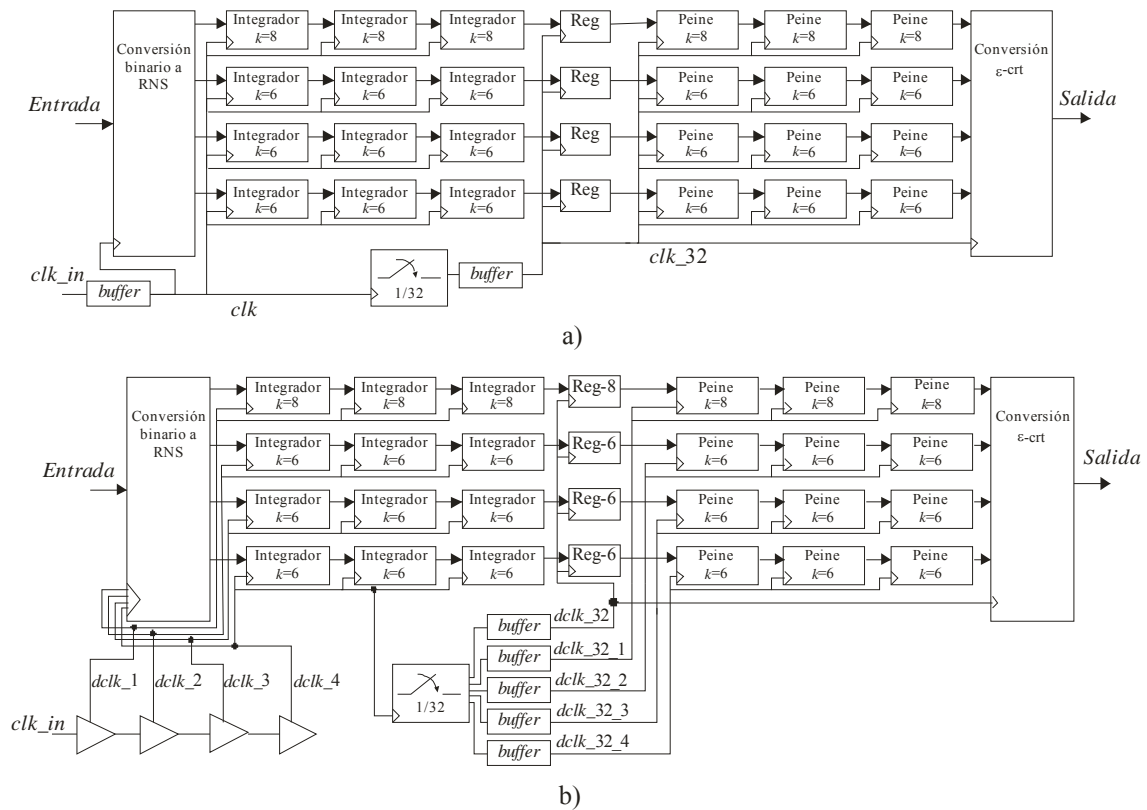


Figura 4.15. Esquema del filtro CIC sincronizado mediante un único reloj (a) y celdas dCLK_cell (b).

escalado ε -CRT (ver Capítulo 2), que proporciona un dato resultante en formato binario de 10 bits. El sistema se ha sincronizado mediante dos estrategias de sincronización: un único reloj (Figura 4.15a) y una señal de reloj distribuida en cuatro fases mediante celdas `dCLK_cell` (Figura 4.15b). En la Figura 4.15b se puede apreciar cómo cada canal está sincronizado mediante una señal de reloj diferente tanto en las etapas de integradores como en las de peines.

4.3.2.2.1. Sincronización del sistema

Los sistemas de prueba están sincronizados a partir de una señal de reloj externa e ideal generada mediante una fuente de tensión, la cual no se ve afectada por efectos de carga en la línea de distribución. Para modelar el comportamiento más realista de la red de distribución de la señal de reloj y poder medir su consumo, se emplea el *buffer* amplificador analizado en el apartado 4.3.1. El *buffer* se puede instanciar en diversos lugares del sistema adaptando sus dimensiones, y consecuentemente su consumo, a los requisitos de *fan-out* locales. En el sistema gobernado por las señales de reloj levemente desfasadas son las propias celdas `dCLK_cell` las que acondicionan las señales de reloj mediante el parámetro W_b , para que actúen sobre los diferentes módulos operativos del sistema.

En el circuito con un único reloj (Figura 4.15a) son necesarios dos *buffers*, uno que sincroniza los canales de integradores, y otro a la salida del reductor de frecuencia para generar la señal `clk_32` que controla la columna de registros del reductor de frecuencia, los canales de peines y el módulo final de conversión y escalado de RNS a binario. La reducción de frecuencia se consigue empleando un contador de cinco bits

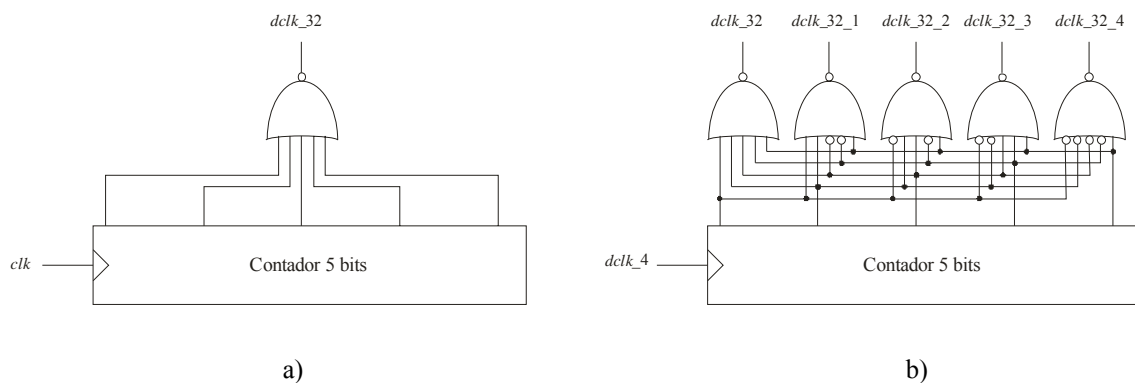


Figura 4.16. Reductores de frecuencia para las dos estrategias de sincronización.

cuya entrada es la señal de reloj y mediante una puerta NOR de cinco entradas (Figura 4.16a). Cada vez que se alcanza una combinación concreta en el contador, la puerta emite un pulso que es acondicionado por el *buffer*, de modo que se genera la señal *clk_32* que reduce la frecuencia de la señal de entrada en el factor 32 requerido. En el sistema sincronizado mediante señales desfasadas se emplean cuatro celdas *dCLK_cell* para sincronizar cada uno de los canales de integradores. La señal más retrasada de las anteriores (*dclk_4*) controla al contador del reductor de frecuencia y las salidas de éste actúan sobre cinco puertas NOR con combinaciones diferentes en sus entradas (Figura 4.16b). Así se puede controlar en qué momento se generará un pulso en las señales *dclk_32* y *dclk_32_1* a *dclk_32_4* de la Figura 4.15b y que se muestran en la Figura 4.17. De este modo se consigue un funcionamiento distribuido en los canales de peines. Cada una de estas señales es acondicionada por un *buffer* de las dimensiones adecuadas para los dispositivos que debe controlar. La aplicación de esta alternativa no supone una carga significativa al sistema en cuanto al número de dispositivos y, en cuanto a las dimensiones de los *buffers*, el tamaño requerido en cada canal es inferior a la alternativa de un único reloj, tal como se verá a continuación, puesto que la carga del *buffer* se ve significativamente reducida.

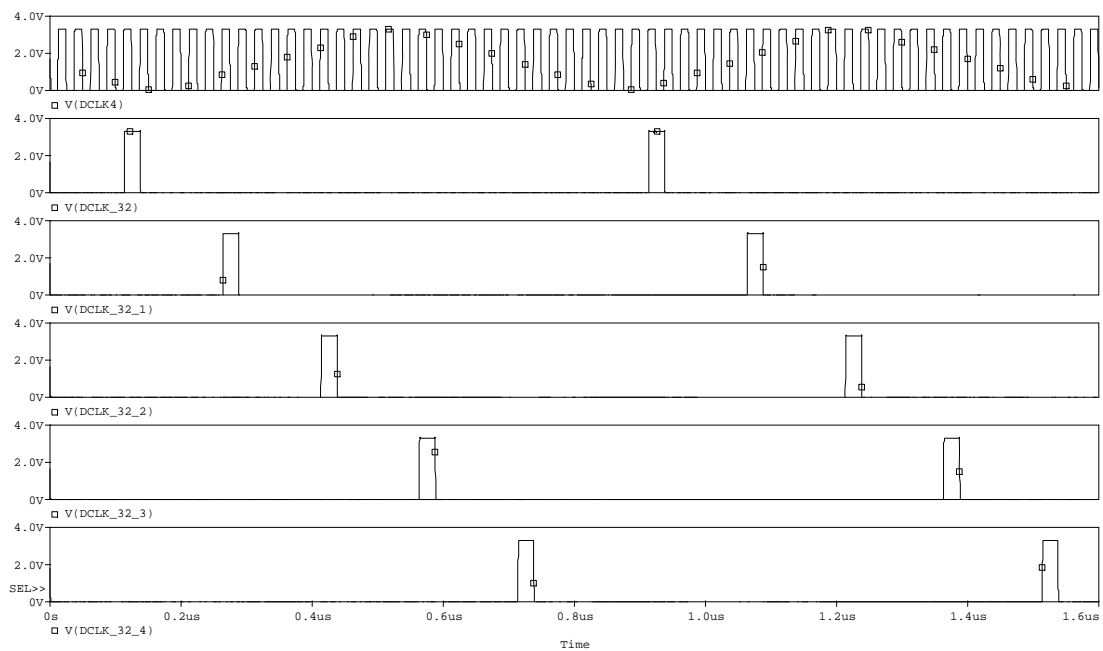


Figura 4.17. Cronograma con la señal que sincroniza el reductor de frecuencia y las cinco señales generadas por éste.

La Figura 4.17 muestra la señal de reloj $dclk_4$ y las cinco señales generadas para el procesamiento de los peines y la etapa de conversión y escalado final. Como se puede apreciar en la Figura 4.14b la señal $dclk_4$ controla el canal más retrasado de los integradores y funciona a la frecuencia del sistema. Cada 32 ciclos de reloj de $dclk_4$ se genera un pulso en la señal $dclk_32$ y se produce la carga de la columna de registros mostrada en la Figura 4.15 y la captura de los resultados de los cuatro canales de peines por parte del módulo final de conversión y escalado. El hecho que la señal clk_32 se genere a partir de la señal $dclk_4$, que es la más retrasada, garantiza la correcta presencia de los datos procesados por cada canal de integradores a la entrada de los registros. Esos datos permanecen en los registros durante 32 ciclos del reloj de entrada y a lo largo de ese tiempo son progresivamente capturados por los canales de peines. Tras la captura de la salida de los integradores se generan de modo uniformemente distribuido las señales $dclk_32_1$ hasta la $dclk_32_4$, que sincronizan respectivamente los cuatro canales de peines. Como ya se ha comentado, la señal clk_32 , además de controlar la carga de los datos producidos por los integradores, sincroniza el módulo final de conversión y escalado ε -CRT, de modo que la salida global del sistema ocurre en uno de cada 32 ciclos de la señal de reloj de entrada.

	Señal de reloj		Parámetros	
	<i>Buffer</i>	<i>dclk_cell</i>	<i>Buffer</i>	<i>dclk_cell</i>
			W (μm)	W_b (μm)
Sistema con reloj único	<i>clk</i>		116.3	
	<i>clk_32</i>		186.5	
Sistema con reloj distribuido		<i>dclk_1</i>		36.0
		<i>dclk_2</i>		26.0
		<i>dclk_3</i>		26.0
		<i>dclk_4</i>		28.4
		<i>dclk_32</i>	36.1	
		<i>dclk_32_1</i>	45.0	
		<i>dclk_32_2</i>	35.2	
		<i>dclk_32_3</i>	35.2	
	<i>dclk_32_4</i>	35.2		

Tabla 4.3. Dimensionado de salida de los *buffers* y celdas *dCLK_cell* en filtro CIC.

Para el dimensionado de los *buffers* y celdas `dCLK_cell` se han realizado detalladas medidas de las áreas de puerta de todos los transistores controlados por las señales de reloj en ambas estrategias de sincronización y, teniendo en cuenta la relación (4.2), se han determinado los valores adecuados para los parámetros W del buffer y W_b de las celdas `dCLK_cell`. La Tabla 4.3 muestra los valores así calculados para cada una de las señales de reloj mostradas en la Figura 4.15.

En cuanto al retardo generado en las celdas `dCLK_cell`, se ha tomado como decisión de diseño que la señal más retrasada este desfasada un cuarto de periodo respecto de la más adelantada (Figura 4.18). De este modo se consigue un funcionamiento claramente distribuido y, al mismo tiempo, un margen de seguridad adecuado para el momento en el que los datos de cada canal deban ser capturados simultáneamente por un señal de reloj común. Esta relación de desfase se mantiene constante en todas las frecuencias consideradas, por lo que los parámetros L_d y W_d de las celdas `dCLK_cell` se deben adaptar para cada frecuencia. El criterio elegido requiere que el desfase entre dos señales consecutivas sea la doceava parte del periodo de la señal de reloj. La Tabla 4.4 muestra, para cada frecuencia, el desfase entre señales

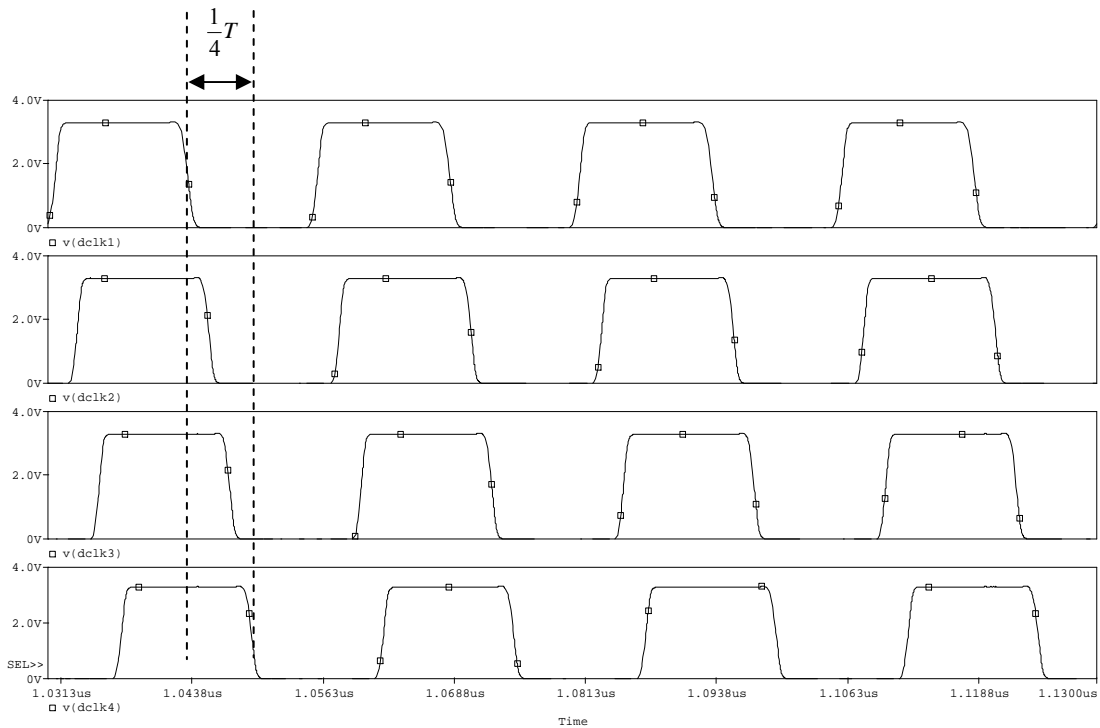


Figura 4.18. Señales de reloj desfasadas generadas mediante las celdas `dCLK_cell`.

Frecuencia	Periodo	Desfase por señal	L_d	W_d
40MHz	25ns	2.08ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 3.0 \mu\text{m}$ $L_{d3} = 2.8 \mu\text{m}$ $L_{d4} = 2.8 \mu\text{m}$	$W_{d1} = 3.8 \mu\text{m}$ $W_{d2} = 12.3 \mu\text{m}$ $W_{d3} = 9.8 \mu\text{m}$ $W_{d4} = 9.5 \mu\text{m}$
64MHz	15.63ns	1.30ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.8 \mu\text{m}$ $L_{d3} = 1.8 \mu\text{m}$ $L_{d4} = 1.8 \mu\text{m}$	$W_{d1} = 3.8 \mu\text{m}$ $W_{d2} = 12.1 \mu\text{m}$ $W_{d3} = 10.2 \mu\text{m}$ $W_{d4} = 9.5 \mu\text{m}$
100MHz	10ns	0.83ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.1 \mu\text{m}$ $L_{d3} = 1.1 \mu\text{m}$ $L_{d4} = 1.1 \mu\text{m}$	$W_{d1} = 3.8 \mu\text{m}$ $W_{d2} = 11.5 \mu\text{m}$ $W_{d3} = 11.5 \mu\text{m}$ $W_{d4} = 10.8 \mu\text{m}$
125MHz	8ns	0.67ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 0.9 \mu\text{m}$ $L_{d3} = 0.9 \mu\text{m}$ $L_{d4} = 0.9 \mu\text{m}$	$W_{d1} = 3.8 \mu\text{m}$ $W_{d2} = 10.2 \mu\text{m}$ $W_{d3} = 10.7 \mu\text{m}$ $W_{d4} = 10.1 \mu\text{m}$

Tabla 4.4. Retardo entre señales de reloj adyacentes y valor de los parámetros L_d y W_d .

de reloj consecutivas generadas por la cadena de celdas `dCLK_cell` y el valor de los parámetros L_d y W_d que generan ese desfase. Los resultados mostrados en la Tabla 4.4 para los valores de L_d y W_d dependen, además del desfase requerido, del parámetro W_b que, como se ha visto en apartados anteriores, afecta al retardo global generado en cada celda. En todos los casos se ha considerado que no se requiere que la primera celda genere desfase alguno; más aún, si los datos de entrada llegan sincronizados con el reloj de entrada al sistema, interesa que el desfase generado por la celda sea mínimo, para que la captura de los datos de entrada se produzca cuanto antes. Por estos motivos los parámetros L_d y W_d de la primera celda `dCLK_cell` se eligen de modo que produzcan transiciones óptimas de los inversores de salida y restaurador de la celda en cuestión con un consumo y retardo mínimos.

4.3.2.2.2. Conversión de binario a RNS

Como ya se ha dicho, el dato de entrada es de 8 bits y la primera acción es convertir cada dato en su representación en el sistema RNS con los módulos $\{256, 63, 61, 59\}$. La Figura 4.19 muestra cómo se realiza dicha conversión [RAM05]. En primer lugar, el

dato de entrada de 8 bits se divide en dos buses: por un lado los cinco bits menos significativos y por otro los tres bits más significativos. Esta división está motivada por el hecho que se puede descomponer un dato como una suma de dos números de manera que uno de ellos sea menor que el módulo del canal correspondiente. Así, el módulo coincide con dicha parte del dato y no es necesario su procesamiento. Solamente hay que transformar la parte más significativa del dato, con lo que se reduce considerablemente el tamaño de la tabla de consulta. Efectivamente, si se tiene el dato de 8 bits:

$$X[n] = [x_7, x_6, x_5, x_4, x_3, x_2, x_1, x_0] \tag{4.6}$$

es claro que:

$$X[n] = [x_7, x_6, x_5, 0, 0, 0, 0, 0] + [0, 0, 0, x_4, x_3, x_2, x_1, x_0] \tag{4.7}$$

$$X[n] = [x_7, x_6, x_5] \times 2^5 + [x_4, x_3, x_2, x_1, x_0] \tag{4.8}$$

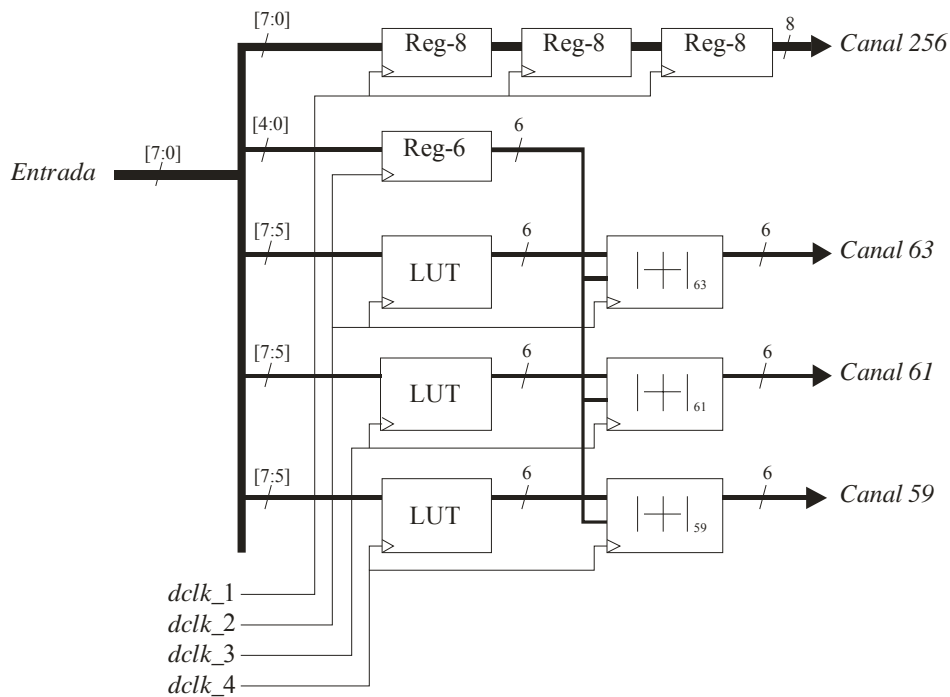


Figura 4.19. Módulo de conversión de binario a RNS.

con lo que:

$$|X[n]_m = \left\| [x_7, x_6, x_5] \times 2^5 \right\|_m + \left\| [x_4, x_3, x_2, x_1, x_0] \right\|_m \quad (4.9)$$

El segundo término de (4.9) es de 5 bits, con lo que está asegurado que es en todo caso inferior a cualquiera de los módulos considerados y el residuo coincide con el propio dato, por lo que no se requiere procesamiento alguno y basta con añadir un registro para mantener la latencia. Tan sólo es necesario almacenar en una LUT el valor del primer sumando de (4.9), lo que supone una memoria de 8 palabras de 6 bits y realizar la suma modular con el segundo término de (4.9). Por otra parte, el primer canal de módulo 256 no requiere modificación en los datos de entrada, por lo que tan sólo es necesario añadir tres registros como elementos de retardo para igualar la latencia de los canales de 6 bits. La señal de reloj desfasada más adelantada (*dclk_1*) sincroniza este canal mientras que los tres canales restantes son sincronizados por las otras tres señales de reloj. La segunda señal (*dclk_2*), además de sincronizar el segundo canal, controla un registro que almacena los cinco bits menos significativos del dato, que son ampliados a seis bits mediante la adición de un cero en la posición más significativa. Dichos bits son comunes a los tres canales de seis bits, por lo que su contenido es sumado modularmente a las salidas de las tres tablas de consulta correspondientes. El hecho que la señal *dclk_2* este más adelantada que las dos restantes, *dclk_3* y *dclk_4*, asegura que el dato contenido en el registro estará disponible para ser procesado por los tres canales. En simulaciones con reloj único todas las entradas de reloj están conectadas al reloj del sistema, de manera que el funcionamiento del módulo es perfectamente síncrono.

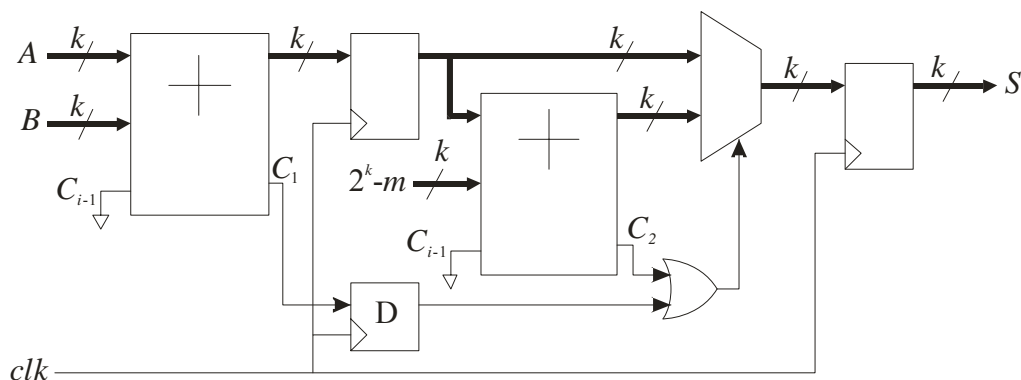


Figura 4.20. Sumador modular con cauce segmentado y salida registrada.

Las memorias empleadas en simulación (Apéndice B) actualizan su salida en flanco de subida, por lo que el registro de 6 bits común a los tres últimos canales está constituido con biestables D activados por flanco de subida. El registro de 8 bits del canal 256 funciona por flanco de bajada, como el resto de los registros del sistema. Los sumadores modulares empleados incluyen registros intermedios, lo que permite un funcionamiento con cauce segmentado (Figura 4.20 y Apéndice B) y además tienen un registro a la salida para mantenimiento del dato producido, por lo que la latencia de cada canal es tres ciclos de reloj y todas las salidas del módulo de conversión son actualizadas en el flanco de bajada del reloj.

4.3.2.2.3. Integradores

Los integradores en los canales de 6 bits se implementan directamente, tal como muestra la Figura 4.21a, gracias a que el sumador modular (Figura 4.20) incluye un registro a la salida, lo que hace innecesario el registro mostrado en el esquema general de un integrador modular (Figura 4.13a). El canal módulo 256 está implementado directamente con sumadores binarios convencionales, ya que la suma modular de datos de n bits en módulo 2^n se puede realizar de modo directo como una suma binaria despreciando el bit de acarreo. La Figura 4.21b muestra cómo se añaden dos registros para igualar la latencia de los integradores implementados mediante sumadores modulares. La configuración de los integradores empleados en todos los canales lleva a una función de transferencia:

$$y[n] = x[n] + y[n - 2] \quad (4.10)$$

que difiere de la mostrada en (4.4). Esta discrepancia no supone problema [MEY01] ya que la inclusión de los registros que implementan el *pipeline* introducen un polo a la mitad de la frecuencia de muestreo, que puede ser compensado por un peine de retardo unidad ($D=1$). En aplicaciones CIC de elevados niveles de diezmado se puede asumir que un filtro *antialiasing* proporciona suficiente supresión de las componentes en mitad de la frecuencia de muestreo como para que el efecto de (4.10) no sea significativo [GAR98].

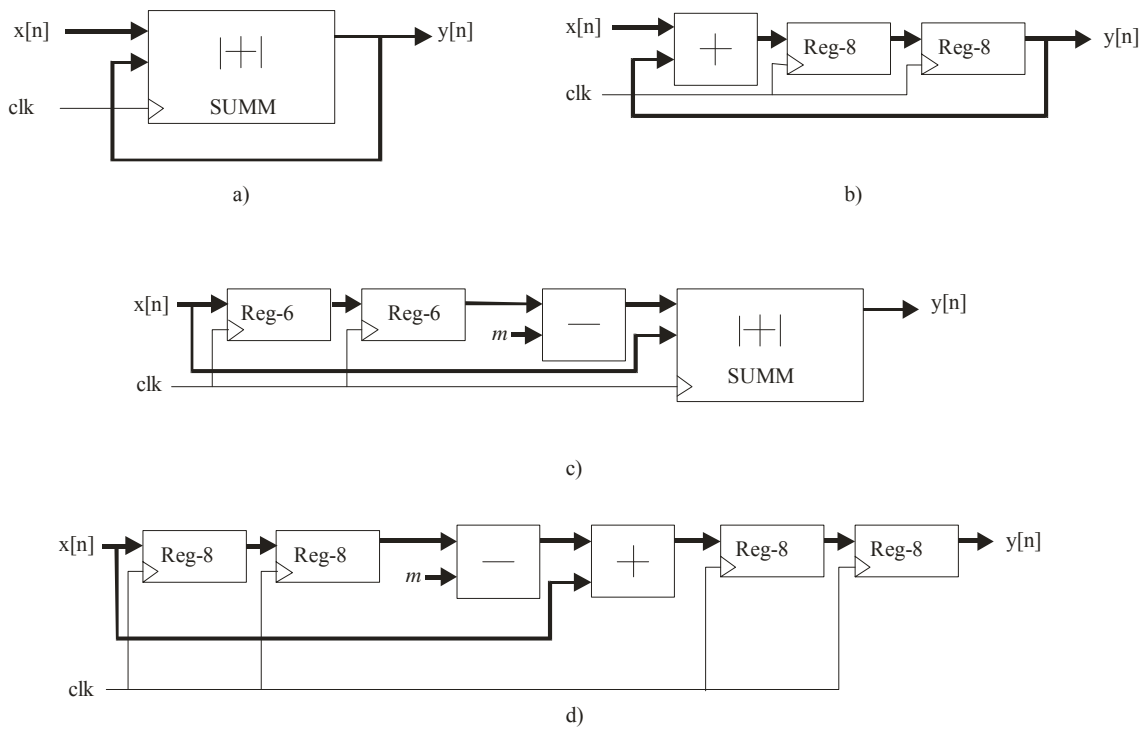


Figura 4.21. Implementación de integradores de 6 y 8 bits (a y b, respectivamente) y peines para los canales de 6 y 8 bits (c y d).

4.3.2.2.4. Peines

Los peines realizan la resta modular mostrada en (4.5). Para implementar la operación de resta mediante sumadores modulares se debe transformar el dato de salida de los registros a su complementario modular. El complementario modular de x es el número \bar{x} que verifica:

$$|x + \bar{x}|_m = 0 \quad (4.11)$$

de donde se puede deducir que:

$$\begin{aligned} x + \bar{x} &= m \\ \bar{x} &= m - x \end{aligned} \quad (4.12)$$

Así, la resta RNS se puede realizar sumando el complementario modular del sustraendo:

$$\begin{aligned}
 |x - y|_m &= \\
 |x + \bar{y}|_m &= \\
 |x + m - y|_m &=
 \end{aligned}
 \tag{4.13}$$

Teniendo en cuenta que en todo caso ($m > y$) la operación ($m - y$) se puede realizar directamente mediante un restador convencional. El circuito que implementa (4.13) es el mostrado en la Figura 4.21b y requiere un restador binario y un sumador modular, aparte de los registros que implementan la función propia del peine. Los peines del canal módulo 256 (Figura 4.21d) emplean dos sumadores convencionales con dos registros a su salida para mantener la latencia por los mismos motivos explicados en el apartado anterior.

4.3.2.2.5. Reductor de frecuencia

El filtro CIC simulado requiere una reducción de frecuencia en un factor de 32. El flujo de datos se ve reducido en ese factor mediante una columna de registros que almacenan uno de cada 32 datos procesados por los integradores (Figura 4.13a). Estos datos son posteriormente capturados y procesados por los peines, tal como ya se ha explicado en el apartado 4.3.2.2.1.

4.3.2.2.6. Conversión y escalado: módulo ε -CRT

Como etapa final del sistema se realiza la conversión de RNS a binario mediante un sistema ε -CRT (Capítulo 2) de conversión más escalado. Como muestra la Figura 4.22, los datos de cada canal se convierten mediante tablas de consulta a los sumandos de (2.20) y la suma modular se realiza mediante sumadores convencionales en los que se ignora el acarreo de salida ya que, como se explicó en el apartado 2.4.2, se ha elegido V de modo que $\lfloor M/V \rfloor$ sea una potencia de dos. Concretamente se ha elegido el factor de escala V para que la salida sea de 10 bits. Este módulo está sincronizado por la misma señal de reloj que controla la columna de registros del reductor de frecuencia, de tal modo que, cada vez que captura un dato, los cuatro canales de peines han producido el residuo correspondiente, tal como se ha explicado en el apartado 4.3.2.2.1 y se muestra en la Figura 4.15. El resultado de la conversión queda almacenado en un registro de salida, que mantiene el dato estable durante un ciclo del reloj de frecuencia reducida. Considerando que las memorias actualizan su salida en flanco de subida y el registro de salida carga el dato en flanco de bajada, la latencia de este módulo es de un ciclo de

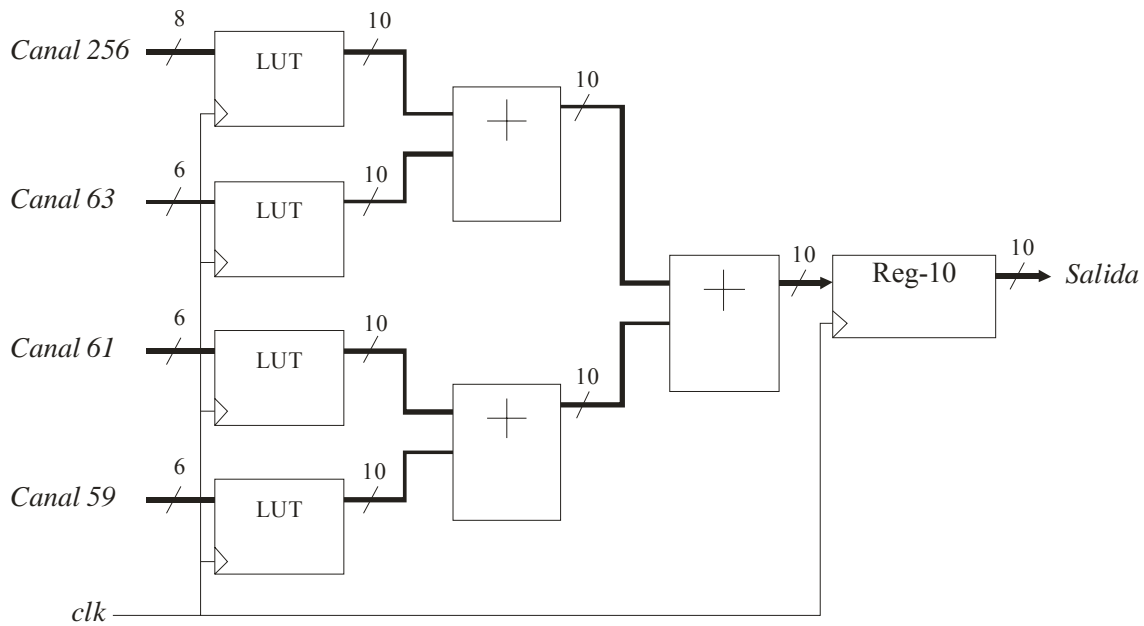


Figura 4.22. Módulo de conversión más escalado ε -CRT para el filtro CIC.

reloj de frecuencia reducida, es decir, la salida del sistema se actualiza cada 32 ciclos del reloj de entrada (Figura 4.17).

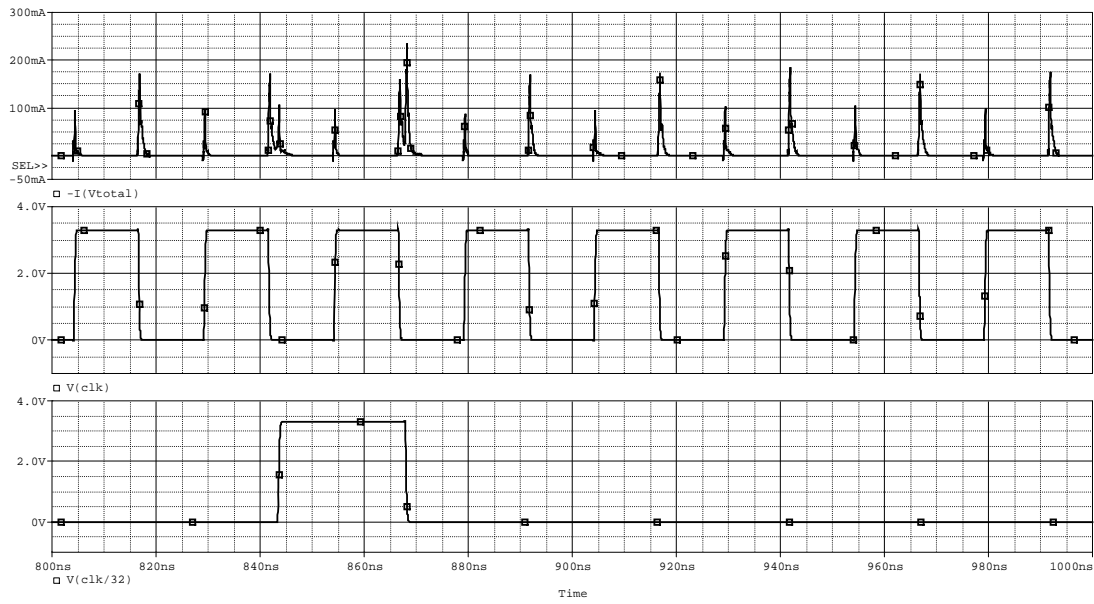
La sincronización del módulo final de conversión más escalado en ambas metodologías de sincronización es la mostrada en la Figura 4.22. En el caso de la estrategia de aplicación de varias señales de reloj existen dos posibilidades de sincronización: que todas las memorias y el registro de salida estuvieran sincronizadas por una señal de reloj común, o sincronizar cada memoria con la señal de reloj que controla el canal correspondiente, y que el registro final se active con la señal clk_{32} . Se ha optado por la primera alternativa, puesto que el objetivo de la generación de señales separadas en las etapas de peines y conversión más escalado es la distribución del funcionamiento de los dispositivos que componen el sistema, y cada uno de los canales supone una carga para la señal de reloj sensiblemente superior a la de la columna de registros más las módulo ε -CRT. Por ese motivo la sincronización mostrada en la Figura 4.21 resulta ser la más eficiente en cuanto a distribución de funcionamiento de la segunda etapa del sistema.

4.3.2.3. Resultados de simulación

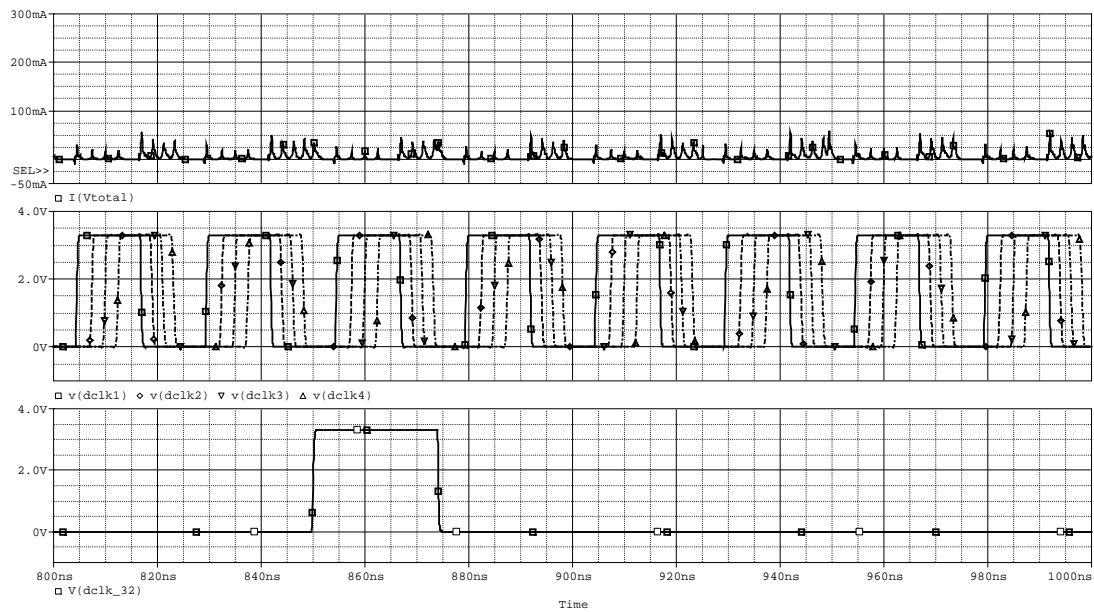
Una simplificación del sistema descrito ya fue exitosamente simulada [GON02a], [GON02b] y [GON02c], quedando de manifiesto las importantes ventajas que se obtienen al aplicar señales ligeramente desfasadas frente a la alternativa el reloj único. En este trabajo los sistemas simulados en [GON02a], [GON02b] y [GON02c] se han aplicado incluyendo las etapas de conversión de binario a RNS y la conversión más el escalado final de RNS a binario. Además, se han ampliado las frecuencias simuladas y se han realizado mediadas de consumo, de modo que los resultados obtenidos amplían y complementan los trabajos realizados.

El sistema descrito en el apartado anterior se ha simulado mediante PSpice bajo diferentes condiciones de funcionamiento. En primer lugar se ha simulado el sistema mediante el método convencional de una única señal de reloj; después se ha simulado mediante las celdas parametrizables `dCLK_cell`, con objeto de comprobar el efecto del funcionamiento distribuido. Posteriormente se ha repetido la simulación del sistema manteniendo una alimentación independiente en ambas redes de distribución del reloj respecto del sistema total, de modo que se pueda medir cómo afecta al sistema la aplicación de la estrategia en estudio. Por último, se ha simulado el sistema con entrada nula en ambas estrategias de sincronización para, mediante comparación con las anteriores simulaciones, comprobar el efecto de la red de distribución de la señal de reloj sin actividad en la entrada del sistema.

En la Figura 4.23 se muestran las demandas de corriente del sistema total junto con las señales de reloj que sincronizan todo el sistema en ambas metodologías de sincronización. Como se puede apreciar los picos de corriente asociados a ambos flancos de la señal de reloj se sustituyen por picos menores correspondientes a los distintos flancos de todas las señales de reloj desfasadas. La aplicación de la estrategia de distribución de la señal de reloj ha transformado un sistema con elevadas demandas de corriente en momentos muy concretos, en un sistema que funciona de modo distribuido a lo largo del tiempo. La Figura 4.23a muestra el caso de sincronización convencional, mientras que en la Figura 4.23b se muestra el resultado de la estrategia de sincronización mediante reloj distribuido. En las gráficas de corriente se ha mantenido la misma escala para facilitar la comparación entre ambos casos. En la Figura 4.23a se pueden apreciar la corriente instantánea demandada por el sistema junto con las dos



a)



b)

Figura 4.23. Señales de reloj y corriente instantánea con reloj único (a) y reloj distribuido (b).

señales de reloj que sincronizan al sistema: la primera, *clk*, que controla la etapa de conversión de binario a RNS y los integradores y la segunda, *clk/32*, que actúa sobre la columna de registros que toman uno de cada 32 datos, los canales de peines y el módulo de conversión y escalado ε -CRT. Los biestables del sistema están controlados por flanco de bajada mientras que las memorias actualizan su salida en flanco de subida, lo

cual queda reflejado en la coincidencia de los picos de corriente con ambos flancos de la señal de reloj. El efecto del consumo de corriente de la lógica combinacional está directamente relacionado con la procedencia de los datos de entrada a estos módulos, por lo que los picos de corriente incluyen el consumo de la lógica combinacional en función de que los datos de entrada procedan de un registro o de un módulo de memoria. Es también notorio el incremento, en torno al 38%, en la corriente instantánea que se produce cuando actúa la señal de reloj de frecuencia reducida, lo que activa toda la segunda etapa del filtro. En el sistema sincronizado mediante las celdas `dCLK_cell` se puede apreciar una drástica reducción de los picos de corriente, y una distribución de los mismos de acuerdo a los flancos de las señales de reloj que sincronizan los diferentes canales del sistema. La Figura 4.23b muestra las demandas de corriente instantáneas, las cuatro señales levemente desfasadas y una de las cinco señales que distribuyen el funcionamiento de la segunda etapa del filtro. Las demandas de corriente no experimentan un incremento apreciable cuando actúan las señales de reloj reducidas, ya que el funcionamiento de los canales de peines y el módulo ϵ -CRT se ha distribuido tal como se ha mostrado en la Figuras 4.14b y 4.16.

Un análisis más detallado del efecto en la corriente máxima demandada se muestra en la Tabla 4.5. En ella se puede comprobar cómo la reducción media de los picos de corriente, obtenida como la diferencia entre los valores de pico en ambas estrategias de sincronización, es del orden del 74%, es decir, los picos de corriente han

	$I_{\text{máx}}$ (mA)			
	40 MHz	64 MHz	100 MHz	125 MHz
Reloj único	234.22	249.70	248.75	241.93
Reloj distribuido	63.25	62.50	62.72	69.39
Disminución de las demandas de corriente instantánea	73%	75%	75%	71%

Tabla 4.5. Comparación de corrientes máximas con reloj único y distribuido en el filtro CIC.

disminuido en un factor similar al número de señales de reloj aplicadas. Estos resultados coinciden con los ya obtenidos para una versión simplificada de este sistema [GON02a], [GON02b] y [GON02c]. Esta importante reducción tiene un efecto directo en el diseño de la red de alimentación de todo el sistema: se puede reducir la sección de las líneas de alimentación, con la consecuente disminución de área, capacidades parásitas y acoplamiento entre líneas.

La aplicación de las señales de reloj distribuidas también tiene un importante efecto en la reducción de la tasa de demandas de corriente a lo largo del tiempo, tal como muestra la Tabla 4.6. En ella se muestran los valores máximos de la derivada temporal de la corriente instantánea, y se puede comprobar cómo la aplicación de las señales desfasadas suponen una reducción del orden del 80% en la corriente total y del sistema, y del 99% en las demandas de corriente de la RDR. Estas importantes reducciones, junto con el descenso de la corriente absoluta a lo largo de todo el sistema, suponen, además de las ventajas antes comentadas, una importante simplificación en el diseño y requerimientos de las fuentes de alimentación.

		$\left(\frac{dI}{dt}\right)_{\text{máx}}$ (A/ns)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	350.23	614.89	1063.03	1425.81
	dCLK	49.53	157.78	213.60	327.67
	Reducción	85.86%	74.34%	79.91%	77.02%
Sistema sin la RDR	CLK	308.78	586.80	999.60	1342.20
	dCLK	49.44	157.38	211.53	324.38
	Reducción	83.99%	73.18%	78.84%	75.83%
RDR	CLK	118.66	132.94	199.35	192.52
	dCLK	0.47	0.75	2.07	3.30
	Reducción	99.60%	99.44%	98.96%	98.29%

Tabla 4.6. Comparación de la máxima derivada de la corriente en el filtro CIC.

En cuanto al impacto en el consumo, la Tabla 4.7 muestra cómo la aplicación de la estrategia de sincronización basada en el reloj distribuido supone un incremento moderado en el consumo del sistema. Este incremento aumenta levemente con la frecuencia, aunque porcentualmente se va haciendo cada vez menor, tal como se puede apreciar en la Figura 4.24. La Tabla 4.7 también incluye, por separado, los consumos de la red de distribución del reloj y del resto del sistema. Como se puede apreciar en la tabla, el consumo de la red de distribución del reloj con el sistema distribuido es levemente inferior al consumo de la alternativa de reloj único, mientras que el consumo del resto del sistema es superior cuando se aplican varias señales de reloj desfasadas. Estas variaciones en el consumo entre ambas estrategias de sincronización se pueden atribuir a que las transiciones del *buffer* amplificador son más abruptas que en la celda `dCLK_cell`, debido a que, mientras el *buffer* ha sido diseñado para realizar transiciones rápidas, las celdas `dCLK_cell` tienen como principal cometido generar un retardo controlado. Así, el inversor de salida del *buffer* está controlado por otro inversor de dimensiones adecuadas para hacer conmutar al primero a velocidad óptima, mientras que los requerimientos de retardo son la prioridad en el diseño del inversor de entrada, que es el que hace conmutar al inversor restaurador de la celda parametrizable. A pesar que las señales de reloj generadas por la celda `dCLK_cell` son adecuadas para sincronizar un sistema, no son las óptimas para minimizar los tiempos de subida. La

		Consumo (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	18.84	29.75	47.45	59.52
	dCLK	22.42	34.65	52.32	64.44
Sistema sin la RDR	CLK	12.79	20.07	32.31	40.61
	dCLK	16.41	25.21	37.66	46.17
RDR	CLK	6.05	9.68	15.14	18.91
	dCLK	6.01	9.44	14.66	18.27

Tabla 4.7. Comparación de consumos con ambas estrategias de sincronización.

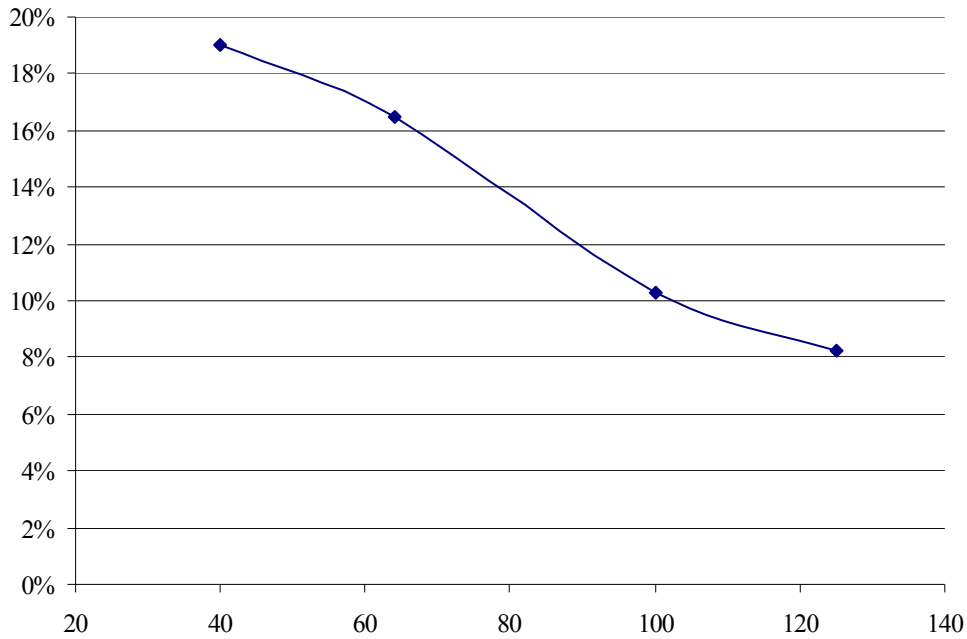


Figura 4.24. Incremento del consumo en sistemas sincronizados por celdas `dCLK_cell`.

penalización en consumo por emplear la cadena de celdas `dCLK_cell` disminuye a medida que aumenta la frecuencia. La Figura 4.24 muestra cómo, a medida que aumenta la frecuencia, la penalización en cuanto al consumo debida al uso de la cadena de celdas `dCLK_cell` es cada vez menos significativa. En cuanto a la red de distribución de la señal de reloj, la cadena de celdas `dCLK_cell` consume ligeramente menos que la alternativa de *buffer* único, debido al menor tamaño de los transistores que la componen en relación al *buffer* amplificador. Como conclusión se observa cómo la aplicación de la estrategia de sincronización basada en las celdas `dCLK_cell` aporta una mejora considerable en cuanto a la disminución de los picos de corriente y al ritmo en que dicha corriente es demandada, mientras que incrementa el consumo global del sistema. Este incremento del consumo se hace menos significativo a medida que aumenta la frecuencia por lo que, a altas frecuencias, la estrategia de sincronización analizada resulta más ventajosa.

El consumo estático del sistema es el que se observa cuando se introduce un patrón de entrada constantemente nulo. Este consumo se corresponde con el consumo estático de todos los dispositivos más la potencia requerida por la red de distribución de la señal de reloj cuando carga y descarga las puertas de los transistores sobre los que

		Consumo estático (mW)			
		40MHz	64MHz	100MHz	125MHz
Reloj único	Sistema total	10.85	17.05	26.92	33.12
	Sistema sin RDR	4.80	7.32	11.75	14.19
	RDR	6.05	9.73	15.17	18.93
Reloj distribuido	Sistema total	14.52	21.86	37.74	48.26
	Sistema sin RDR	8.6	12.61	23.49	30.50
	RDR	5.92	9.25	14.25	17.76

Tabla 4.8. Consumo del filtro CIC sin actividad a la entrada de datos.

actúa. El consumo estático del filtro CIC queda recogido en la Tabla 4.8, en la que se puede apreciar cómo el consumo de la red de distribución del reloj es prácticamente igual al caso de funcionamiento normal, mientras que, como cabía esperar, el consumo del resto del sistema es sensiblemente inferior. Si se compara el consumo entre las dos estrategias de sincronización se puede apreciar cómo el consumo estático es notablemente superior en el caso de la aplicación del reloj distribuido. Esto se debe a que el consumo estático es el que se produce fundamentalmente en los nodos afectados por la señal de reloj y, como ya se ha comentado, la calidad del perfil de las señales de reloj generadas por las celdas `dCLK_cell` no es tan buena como la señal generada por los *buffers* amplificadores. Esto tiene como resultado que el consumo estático del sistema aumente, ya que este consumo está ocasionado básicamente por la conmutación de las puertas de los dispositivos afectador por la señal de reloj y, como ya se ha comentado, la calidad del perfil de la señal de reloj es mejor en el caso de la señal de reloj única que en el caso distribuido.

Mediante la diferencia entre las Tablas 4.7 y 4.8 se puede calcular el consumo dinámico del sistema, es decir, el debido únicamente a la actividad de los circuitos cuando se está realizando el procesamiento de los datos. La Tabla 4.9 contiene los resultados del consumo dinámico del filtro CIC. Los resultados muestran cómo para las frecuencias inferiores (40 MHz y 64 MHz) el consumo dinámico de ambos sistemas es

		Consumo Dinámico (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	7.99	12.70	20.53	26.40
	dCLK	7.90	12.79	14.58	16.18
Sistema sin la RDR	CLK	7.99	12.75	20.56	26.42
	dCLK	7.81	12.60	14.17	15.67
RDR	CLK	0.00	-0.05	-0.03	-0.02
	dCLK	0.09	0.19	0.41	0.51

Tabla 4.9. Comparación del consumo dinámico en el filtro CIC.

muy similar mientras que a medida que aumenta la frecuencia el consumo dinámico del sistema sincronizado mediante la cadena de celdas `dCLK_cell` es inferior al del sistema sincronizado con una única señal de reloj. Además, se puede comprobar cómo el consumo dinámico se produce fundamentalmente en el sistema, mientras que la red de distribución de la señal de reloj permanece prácticamente constante o con una variación mínima (inferior en promedio al 2.5%). Este resultado concuerda con las conclusiones obtenidas para el consumo estático. La red de distribución de la señal de reloj tiene un consumo prácticamente independiente de la actividad del sistema, puesto que el procesamiento de los datos es independiente de la transmisión de la señal de reloj; sin embargo, la Tabla 4.9 muestra cómo el sistema sincronizado mediante una señal de reloj distribuida es más eficiente en términos energéticos que el sistema completamente síncrono desde la perspectiva del consumo exclusivamente debido al procesamiento, si bien este efecto es significativo para las frecuencias superiores.

4.3.3. Controlador PID

Un controlador PID [AST96] es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor de referencia que se pretende alcanzar, para aplicar una acción correctora que ajuste un proceso. El algoritmo de cálculo del controlador PID se da en tres parámetros distintos: el

proporcional, que determina la reacción ante el error actual; el integral, que genera una respuesta ante errores constantes; y el diferencial que anticipa una corrección ante la tendencia del error. La suma de estas tres acciones se usa para controlar un proceso mediante una selección adecuada de las constantes que multiplican a cada uno de los tres elementos de control. El controlador PID es adecuado para aplicaciones que requieren un control muy preciso de un sistema, como pueden ser sistemas de micro/nanoposicionamiento [XU11] empleados en microscopía de sonda o manipulación biológica.

4.3.3.1. Descripción del sistema

El controlador PID queda descrito mediante la ecuación [AST96]:

$$y(t) = K_p \left(x(t) + \frac{1}{T_i} \int_0^t x(\tau) d\tau + T_d \frac{dx(t)}{dt} \right) \quad (4.11)$$

donde K_p es la constante proporcional, T_i es la constante de tiempo integral y T_d es la constante de tiempo diferencial. La implementación discreta de (4.11) se obtiene mediante las siguientes aproximaciones:

$$\begin{aligned} \int_0^t x(\tau) d\tau &\approx \sum_{i=0}^{n-1} x[i]h \\ \frac{dx(t)}{dt} &\approx \frac{x[n] - x[n-1]}{h} \end{aligned} \quad (4.12)$$

donde $x[i]$ es la muestra i -ésima y h es el periodo de muestreo. Sustituyendo (4.12) en (4.11) se obtiene la expresión discreta de (4.11):

$$\begin{aligned} y[n] = y[n-1] + K_p (x[n] - x[n-1]) + \frac{K_p h}{T_i} x[n-1] + \\ + \frac{K_p T_d}{h} (x[n] - 2x[n-1] + x[n-2]) \end{aligned} \quad (4.13)$$

La ecuación (4.13) puede escribirse de la forma:

$$y[n] = y[n-1] + C_0 x[n] + C_1 x[n-1] + C_2 x[n-2] \quad (4.14)$$

donde se han definido:

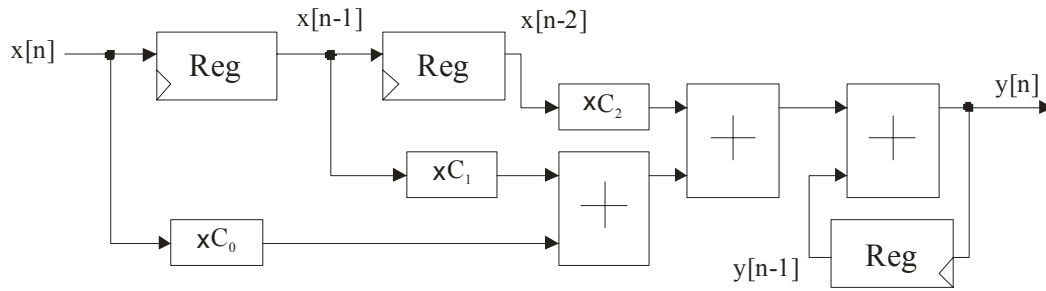


Figura 4.25. Implementación discreta de un controlador PID.

$$C_0 = K_p \left(1 + \frac{T_d}{h} \right) \quad C_1 = K_p \left(\frac{h}{T_i} - 1 - 2 \frac{T_d}{h} \right) \quad C_2 = \frac{K_p T_d}{h} \quad (4.15)$$

Conceptualmente (4.14) se puede implementar mediante un sistema como el mostrado en la Figura 4.25. El anterior sistema ya se ha implementado y probado [GON05] y en este trabajo se ha ampliado el estudio incluyendo las etapas de entrada y salida que convierten los datos de entrada de binario a RNS y los de salida realizan la conversión contraria más el escalado de RNS a binario. Además, el estudio se ha ampliado a más frecuencias de funcionamiento y se ha detallado el análisis del consumo separando el consumo de la RDR y el resto del sistema por separado.

El sistema de prueba [GON05] adquiere datos de entrada binarios en complemento a 2 de 12 bits que son multiplicados por coeficientes de 16 bits, con lo que, según (4.14), se obtiene un rango dinámico total de 32 bits. De este modo, el sistema RNS equivalente debe cubrir dicho rango dinámico. Para ello se considera un sistema con los módulos {256, 63, 61, 59, 55} que se implementa en cinco canales, uno de 8 bits y cuatro de 6 bits. La sincronización de este sistema de nuevo se lleva a cabo mediante una señal de reloj única o usando cinco celdas `dCLK_cell`, tal como muestra la Figura 4.26.

4.3.3.2. Implementación para la simulación

La implementación para simulación en PSpice del controlador PID en RNS es la que se muestra en la Figura 4.26. El sistema de prueba, formado por alrededor de 50.000 MOSFETs, adquiere datos de entrada binarios de 12 bits que convierte en datos RNS para los cinco canales considerados, de modo similar a como se hizo en el filtro CIC (apartado 4.3.1.2.2, Figura 4.19). Después cada residuo se procesa con un esquema

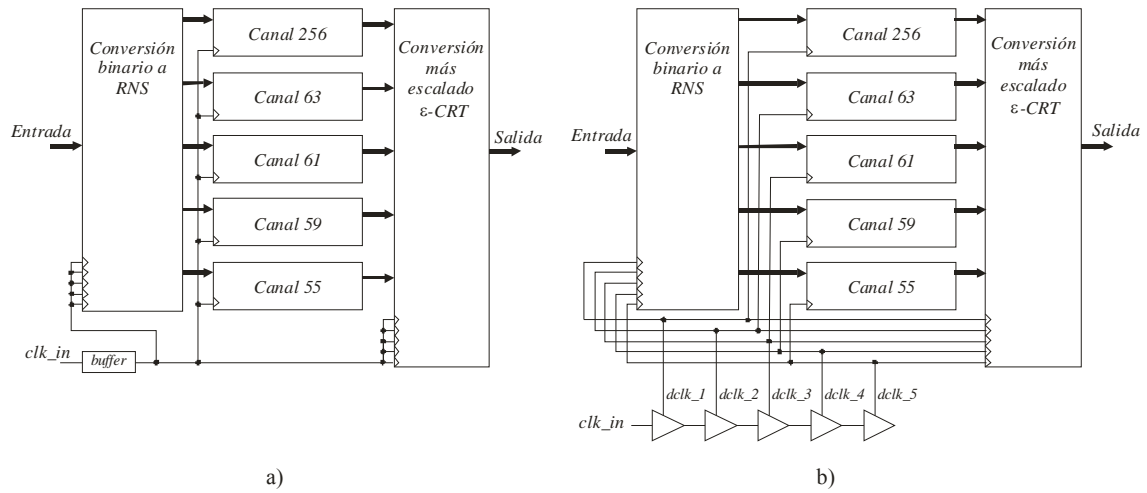


Figura 4.26. Implementación para simulación de un controlador PID basado en RNS de cinco canales mediante las dos metodologías de sincronización.

	Señal de reloj	Parámetro
Sistema con reloj único		W
	clk	196.5 μm
Sistema con reloj distribuido		W_b
	$dclk_1$	41.0 μm
	$dclk_2$	37.7 μm
	$dclk_3$	37.7 μm
	$dclk_4$	37.7 μm
	$dclk_5$	42.4 μm

Tabla 4.10. Dimensionado del *buffer* y de las celdas *dCLK_cell*.

como el de la Figura 4.25 y, finalmente, los residuos procesados se convierten de nuevo a dato binario mediante un módulo de conversión más escalado ϵ -CRT adaptado para los cinco canales en cuestión. El parámetro W del *buffer* amplificador, ya explicado en el apartado 4.3.1.2.1 para la señal de reloj única, y los parámetros W_b de las celdas

parametrizables `dCLK_cell`, ya visto en el apartado 4.2.2, han sido seleccionados para que las señales de reloj generadas sean adecuadas a la carga que soportan.

4.3.3.2.1. Sincronización del sistema

La señal de reloj que gobierna al controlador PID se ha implementado mediante las dos estrategias de sincronización sobre las que se está realizando el estudio comparativo. La metodología convencional de señal de reloj única requiere un *buffer* amplificador para todo el sistema, mientras que se necesitan cinco celdas parametrizables `dCLK_cell` para sincronizar los cinco canales que constituyen el controlador. Cada una de las señales generadas por la cadena de celdas `dCLK_cell` sincroniza la conversión de binario a RNS, el canal del controlador PID correspondiente y la primera etapa del módulo de conversión más escalado. La señal más retrasada, `dclk_5`, además controla el registro final del módulo ε -CRT, que mantiene el dato resultante a la salida durante un ciclo de reloj, de modo que se garantiza que los datos procesados por todos los canales

Frecuencia	Periodo	Desfase por señal	L_d	W_d
40MHz	25ns	1.56ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.8 \mu\text{m}$ $L_{d3} = 1.8 \mu\text{m}$ $L_{d4} = 1.8 \mu\text{m}$ $L_{d5} = 1.7 \mu\text{m}$	$W_{d1} = 4.4 \mu\text{m}$ $W_{d2} = 15.0 \mu\text{m}$ $W_{d3} = 13.0 \mu\text{m}$ $W_{d4} = 13.0 \mu\text{m}$ $W_{d5} = 13.0 \mu\text{m}$
64MHz	15.63ns	0.98ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.1 \mu\text{m}$ $L_{d3} = 1.1 \mu\text{m}$ $L_{d4} = 1.1 \mu\text{m}$ $L_{d5} = 1.0 \mu\text{m}$	$W_{d1} = 4.4 \mu\text{m}$ $W_{d2} = 14.0 \mu\text{m}$ $W_{d3} = 14.0 \mu\text{m}$ $W_{d4} = 13.8 \mu\text{m}$ $W_{d5} = 15.5 \mu\text{m}$
100MHz	10ns	0.63ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 0.8 \mu\text{m}$ $L_{d3} = 0.8 \mu\text{m}$ $L_{d4} = 0.8 \mu\text{m}$ $L_{d5} = 0.7 \mu\text{m}$	$W_{d1} = 4.4 \mu\text{m}$ $W_{d2} = 9.2 \mu\text{m}$ $W_{d3} = 10.0 \mu\text{m}$ $W_{d4} = 10.0 \mu\text{m}$ $W_{d5} = 11.5 \mu\text{m}$
125MHz	8ns	0.50ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 0.6 \mu\text{m}$ $L_{d3} = 0.6 \mu\text{m}$ $L_{d4} = 0.6 \mu\text{m}$ $L_{d5} = 0.6 \mu\text{m}$	$W_{d1} = 4.4 \mu\text{m}$ $W_{d2} = 8.5 \mu\text{m}$ $W_{d3} = 9.5 \mu\text{m}$ $W_{d4} = 9.5 \mu\text{m}$ $W_{d5} = 9.0 \mu\text{m}$

Tabla 4.11. Dimensionado de las celdas `dCLK_cell` para alcanzar los retardos requeridos.

estén disponibles para su captura simultánea. La Tabla 4.10 muestra los valores de los parámetros W y W_b del *buffer* y las celdas `dCLK_cell` que sincronizan adecuadamente al sistema.

El desfase aplicado a cada señal de reloj ha sido seleccionado de modo que la diferencia de fase entre la primera y la última señal sea de un cuarto de periodo, por lo que el desfase entre señales consecutivas resulta ser de un dieciseisavo del periodo. La Tabla 4.11 muestra los valores de los parámetros L_d y W_d que producen el desfase requerido. Al igual que en el sistema de prueba anterior, la primera celda no tiene que generar retardo alguno respecto de la señal previa, y se dimensiona de modo que produzca transiciones óptimas desde el punto de vista de la relación (4.2) analizada en el apartado 4.3.1.

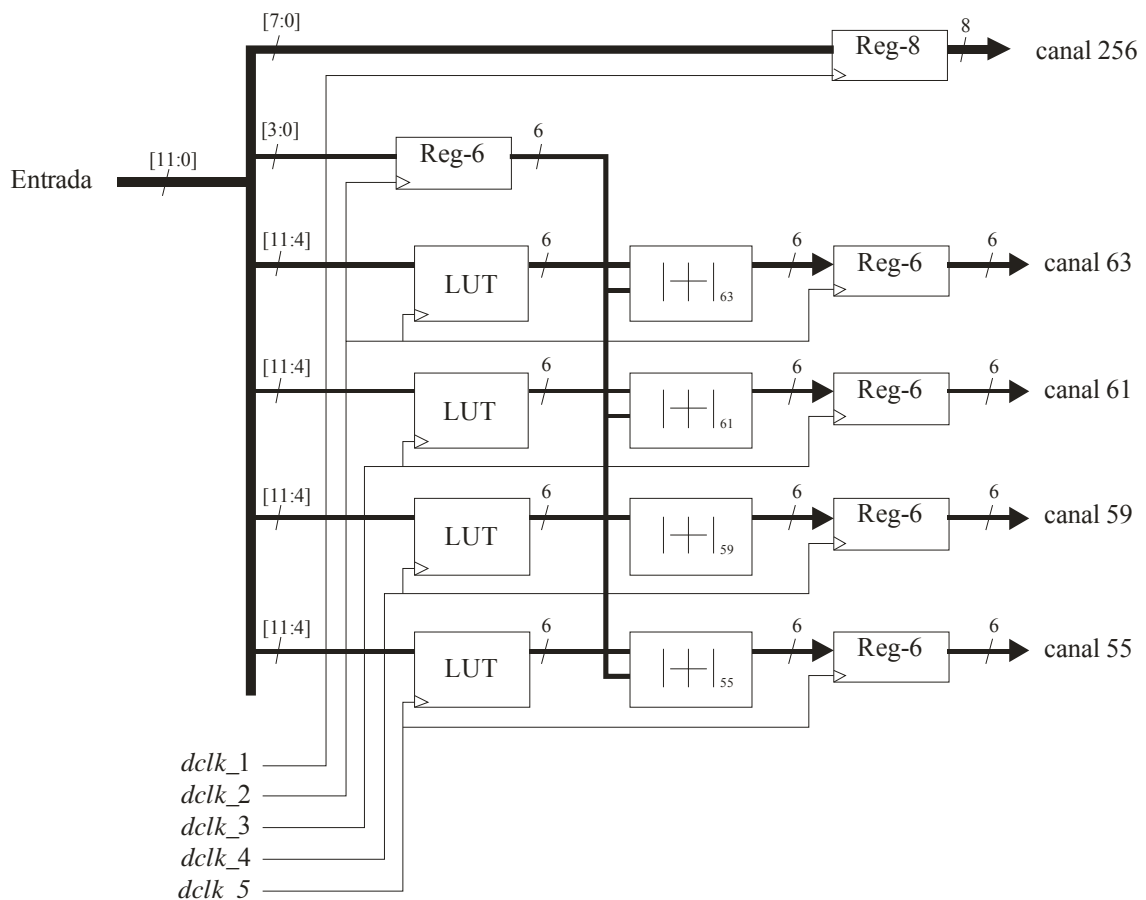


Figura 4.27. Módulo de conversión de binario a RNS del controlador PID.

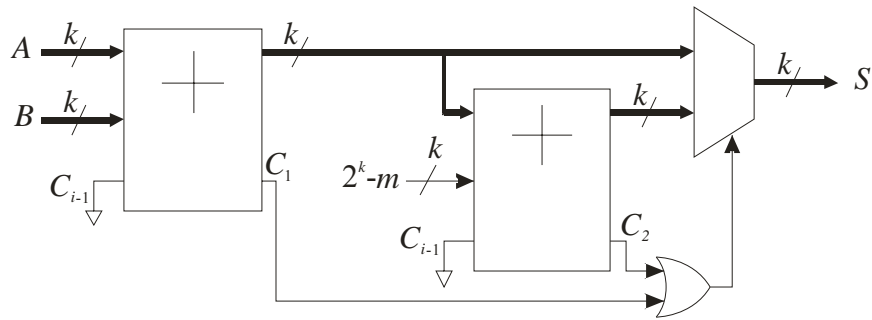


Figura 4.28. Sumador modular combinacional.

4.3.3.2.2. Conversión de binario a RNS

La conversión de binario a RNS (Figura 4.27) se realiza basándose en el mismo esquema de conversión que en sistema de prueba anterior. En los cuatro canales de 6 bits, de módulos 63, 61, 59 y 55, el dato de entrada de 12 bits se separa en dos buses; los 4 menos significativos no serán procesados puesto que coinciden con su residuo, mientras que los 8 más significativos son transformados a su residuo mediante tablas de consulta. Posteriormente se suman modularmente ambos términos para generar el residuo correspondiente. En el canal de 8 bits, de módulo 256, basta con truncar el dato de entrada y conservar los 8 bits menos significativos para realizar la conversión del dato de entrada a su residuo en dicho módulo. En este caso se han usado sumadores modulares combinacionales (Figura 4.28) de modo que en el canal de 8 bits sólo se requiere un registro para mantener la latencia con el resto de canales. Los módulos de memoria activan su salida en flanco de subida, por lo que el registro que captura los cuatro bits menos significativos también es de flanco de subida, mientras que los registros finales a la salida de los sumadores cargan el dato en flanco de bajada, por lo que, en conjunto, el módulo de conversión tiene una latencia de un ciclo de reloj. El registro que captura los cuatro bits menos significativos es de 4 bits de entrada y 6 bits de salida, por lo que añade dos ceros en las posiciones más significativas de su salida. Por último, en el caso de la sincronización mediante reloj distribuido, la señal más adelantada que actúa sobre los canales de 6 bits es la que captura los cuatro bits menos significativos, de modo que este dato, común a todos los canales, esté disponible cuando lleguen las señales de reloj desfasadas posteriores.

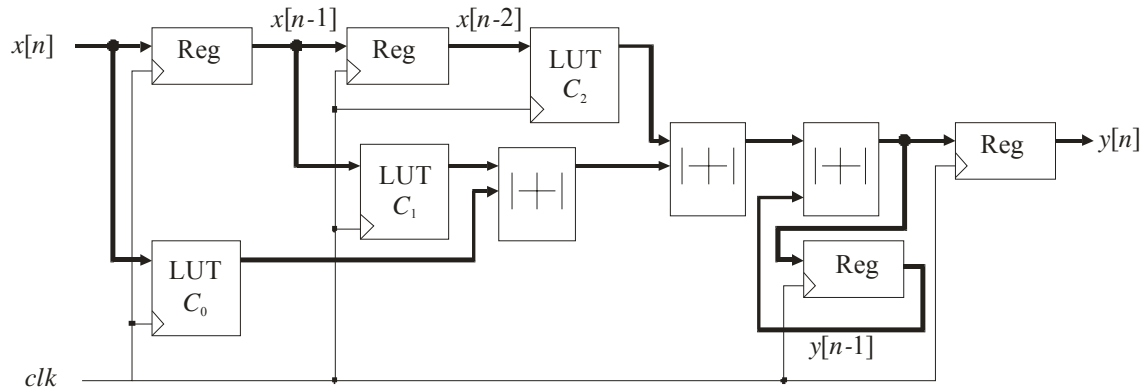


Figura 4.29. Canal PID implementado mediante tablas de consulta.

4.3.3.2.3. Canales del controlador PID

Cada canal está formado por un controlador PID [GON05], como el descrito en la Figura 4.25, adaptado para operar en el sistema RNS (Figura 4.29). Al controlador se le añade un registro a la salida para mantener estable el dato durante un ciclo completo de reloj. Las operaciones de multiplicación por los coeficientes C_0 , C_1 y C_2 se implementan mediante tablas de consulta que almacenan los productos del coeficiente por los valores de la entrada y los sumadores modulares son combinatoriales. Todos los registros cargan el dato a su entrada por flanco de bajada mientras que las memorias que constituyen las LUTs actualizan su salida en flanco de subida, por lo que la latencia de cada canal es de tres ciclos de reloj. La Figura 4.29 muestra la arquitectura de los canales del controlador PID para los módulos 63, 61, 59 y 55, que son de 6 bits. El canal correspondiente al módulo 256 opera con datos de 8 bits y utiliza sumadores binarios convencionales en los que se desprecia el acarreo de salida.

4.3.3.2.4. Conversión y escalado: módulo ε -CRT

El módulo final de conversión y escalado (Figura 4.30) consta de cinco tablas de consulta para acumular los coeficientes de (2.20) correspondientes a cada canal. La primera LUT corresponde al canal módulo 256, por lo que es una memoria de 256×10 bits, mientras que el resto son memorias de 64×10 bits. Tres sumadores binarios de 10 bits suman las salidas de las cinco tablas y un registro final de 10 bits mantiene estable el dato producido durante un ciclo de reloj. En el esquema de sincronización convencional una única señal de reloj actúa sobre todos los componentes mientras que en el caso de la sincronización mediante señales desfasadas, cada memoria está

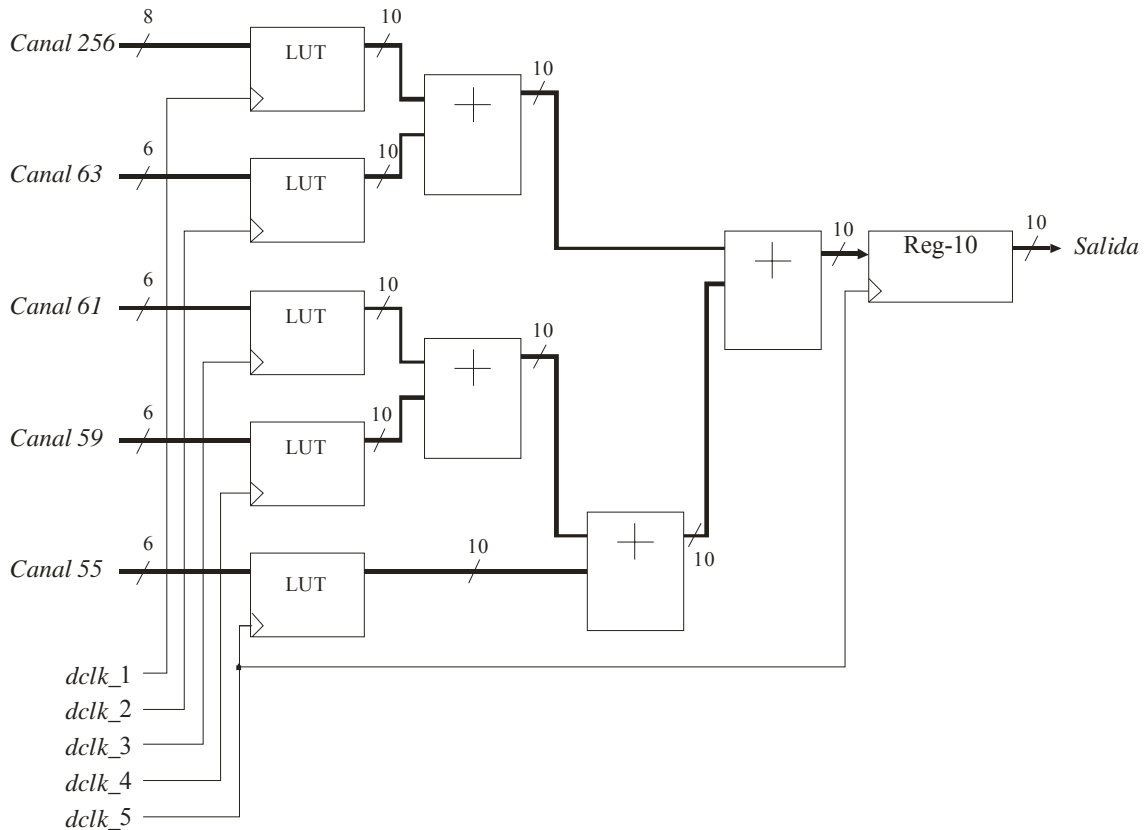


Figura 4.30. Módulo de conversión más escalado e-CRT para el controlador PID.

controlada por la señal que sincroniza el canal correspondiente, tal como se muestra en la Figura 4.26. La salida de las cinco tablas de consulta, correspondientes a cada uno de los cinco canales, se realiza en el registro final que está sincronizado por la señal de reloj más retrasada *dclk_5*. De este modo, cuando en dicho registro se produce la captura del dato, todas las memorias han generado su salida y los sumadores han realizado la suma de los cinco datos parciales correspondientes a cada canal.

4.3.3.3. Resultados de simulación

El controlador PID se ha simulado mediante las dos estrategias de sincronización a cuatro frecuencias de funcionamiento: 40, 64, 100 y 125 MHz. La Tabla 4.12 contiene los valores máximos de corriente en ambas metodologías de sincronización y la reducción porcentual que se obtiene para cada valor de frecuencia analizada.

	$I_{\text{máx}}$ (mA)			
	40 MHz	64 MHz	100 MHz	125 MHz
Reloj único	213.34	213.83	214.07	218.13
Reloj distribuido	61.60	68.20	85.92	94.48
Disminución de las demandas de corriente instantánea	71%	68%	60%	57%

Tabla 4.12. Corrientes máximas con reloj único y distribuido en el controlador PID.

Los resultados de simulación muestran cómo las demandas de corriente máximas se mantienen prácticamente constantes en todas las frecuencias en el caso del reloj único, mientras que en el caso del reloj distribuido aumentan a medida que aumenta la frecuencia. Esto supone que la mejora obtenida en cuanto a reducción de los picos de corriente sea más efectiva para bajas frecuencias. La Figura 4.31 muestra la variación de la corriente máxima en función de la frecuencia, y en ella se puede apreciar

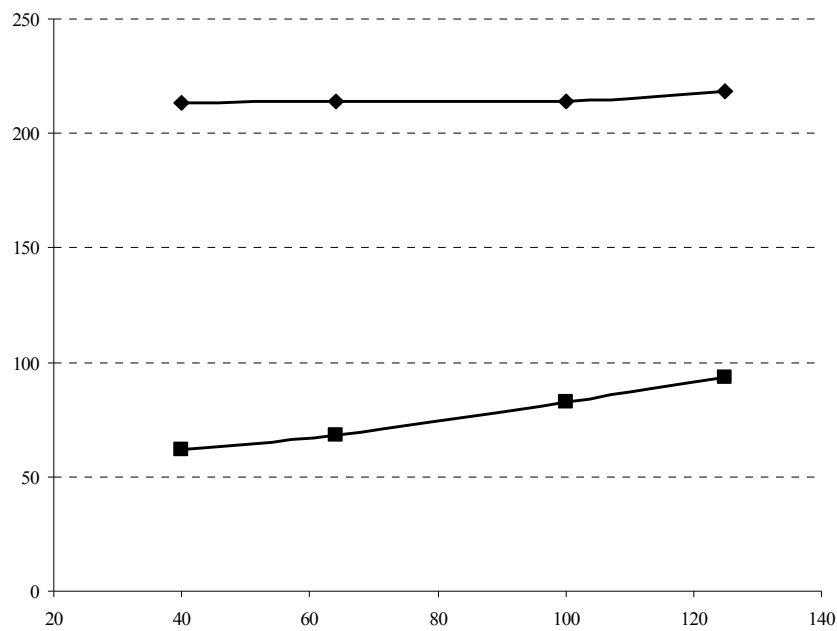
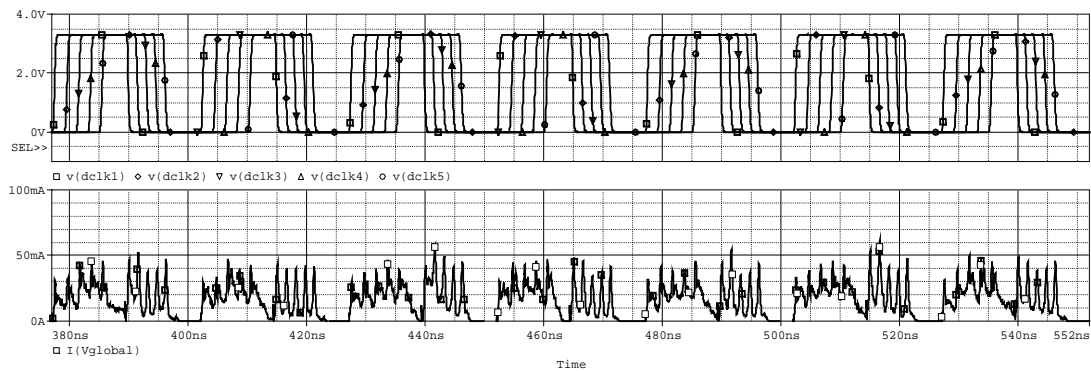


Figura 4.31. Comparación de los valores máximos de las demandas de corriente,

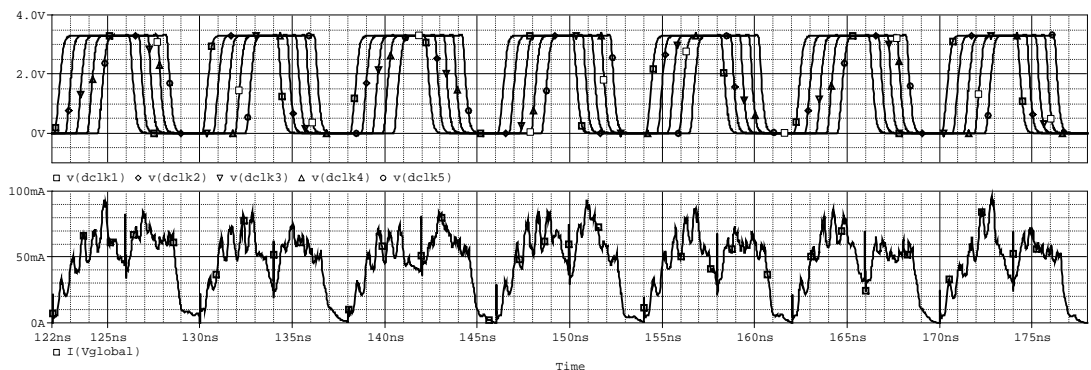
—◆— reloj único —■— reloj distribuido

cómo, para el caso del sistema controlado mediante una única señal de reloj, dichas demandas se mantienen prácticamente constantes a medida que varía la frecuencia mientras que en el caso de la aplicación de varias señales ligeramente desfasadas la tendencia es un incremento de la corriente máxima demandada. En todo caso, la reducción de los picos máximos de corriente es claramente ventajosa en todas las frecuencias analizadas.

La explicación de este comportamiento se encuentra en el solapamiento de los picos de corriente correspondientes a cada flanco de las señales de reloj desfasadas a medida que la frecuencia aumenta. La Figura 4.33 muestra las señales de reloj y las demandas instantáneas de corriente en el caso de la aplicación de las celdas `dCLK_cell` a las frecuencias inferior y superior. La Figura 4.32 muestra cómo los cinco picos de corriente claramente diferenciados a 40MHz se superponen de modo



a)



b)

Figura 4.32. Comparación de corrientes con reloj distribuido a 40MHz (a) y 125MHz (b).

		$\left(\frac{dI}{dt}\right)_{\text{máx}}$ (A/ns)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	321.93	395.62	731.83	2385.20
	dCLK	23.35	62.77	184.85	357.40
	Reducción	92.75%	83.94%	74.70%	84.83%
Sistema sin la RDR	CLK	227.85	359.53	671.93	2185.40
	dCLK	23.30	62.77	184.85	357.40
	Reducción	89.77%	82.54%	72.49%	83.65%
RDR	CLK	44.08	109.70	128.68	203.25
	dCLK	0.62	0.77	1.12	4.54
	Reducción	98.59%	99.30%	99.13%	97.77%

Tabla 4.13. Comparación de la máxima derivada de la corriente en el controlador PID.

notable en el caso de frecuencia 125MHz. El resultado es un incremento de los picos de corriente instantánea demandada a las fuentes. En el caso del sistema sincronizado con un reloj único este solapamiento no llega a producirse y la demanda máxima de corriente instantánea permanece prácticamente constante dentro del rango de frecuencias estudiadas, tal como muestran la gráfica de la Figura 4.31 y la Tabla 4.12.

En cuanto al ritmo de demanda de corriente los resultados muestran cómo, de nuevo, el sistema sincronizado mediante varias señales de reloj distribuidas presenta notables ventajas frente al sistema sincronizado mediante una única señal de reloj. El funcionamiento distribuido supone que la corriente instantánea generada en las fuentes varía de un modo mucho más moderado en contraposición a las fuertes demandas de corriente del sistema perfectamente síncrono. En la Tabla 4.13 se han recogido los valores experimentales de la derivada de la corriente instantánea y se puede comprobar cómo las reducciones medias son del orden del 84% en la corriente global, del 82% en

		Consumo (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	54.48	81.63	122.85	148.29
	dCLK	53.62	82.29	121.09	152.32
Sistema sin la RDR	CLK	44.35	64.44	97.46	116.61
	dCLK	44.41	67.47	97.90	121.53
RDR	CLK	10.13	16.19	25.39	31.68
	dCLK	9.41	14.82	23.19	30.79

Tabla 4.14. Comparación de consumos en el controlador PID.

la corriente demandada por el sistema sin incluir la RDR y del 99% en la corriente suministrada para la red de distribución de la señal de reloj.

El análisis de consumo del controlador en ambas metodologías de sincronización (Tabla 4.14) muestra que el consumo global del sistema es prácticamente el mismo en los dos casos. El análisis separado de los consumos de la red de distribución de la señal de reloj y del resto del sistema revela que, al igual que en el caso anteriormente analizado del filtro CIC, la red de distribución de la señal de reloj en el caso de las celdas `dCLK_cell` es ligeramente más eficiente en cuanto a consumo pero el resto del sistema consume más en este caso. El motivo de este resultado es el mismo que en el sistema anterior; el *buffer* amplificador genera señales de reloj de mejor calidad a costa de un mayor consumo propio, pero el resto del sistema se beneficia de perfiles de señal más abruptos, con lo que se reduce su consumo. En este caso sin embargo, el incremento del consumo por parte del sistema sin la red de distribución del reloj es sensiblemente menor que en el filtro CIC, donde dicho incremento oscilaba en promedio en torno al 37%, mientras que en este caso es ligeramente superior al 2%. La explicación de este resultado es que el filtro CIC hace un uso extensivo de registros y lógica combinatorial para implementar los integradores y peines, y la señal de reloj afecta a la mayoría de los componentes, por lo que la calidad de la señal de reloj supone un factor más influyente en el consumo. Sin embargo, en el controlador PID la mayor

		Consumo estático (mW)			
		40MHz	64MHz	100MHz	125MHz
Reloj único	Sistema total	17.53	20.06	43.93	54.97
	Sistema sin la RDR	7.37	11.82	18.50	23.16
	RDR	10.16	16.24	25.43	31.81
Reloj distribuido	Sistema total	26.01	42.67	60.83	67.66
	Sistema sin la RDR	16.54	27.70	37.54	38.54
	RDR	9.47	14.97	23.29	29.12

Tabla 4.15. Consumo del controlador PID sin actividad en la entrada de datos.

parte del sistema son memorias ROM que constituyen las tablas de consulta. Estas memorias se ven menos afectadas por la calidad de la señal de reloj, ya que la inmensa mayoría de sus componentes no están directamente controlados por esta señal. Además, casi toda la lógica combinatorial adquiere sus entradas a partir de estas memorias en contraste con el filtro CIC, en el que la práctica totalidad de los elementos aritméticos adquieren su entrada directamente de registros más sensibles al reloj. El resultado es que el filtro PID es menos sensible a la calidad de la señal de reloj que el filtro CIC.

La tabla 4.15, relativa al consumo estático, viene a corroborar este hecho, puesto que en ella se aprecia cómo el consumo, cuando la entrada al sistema no presenta ninguna actividad, decae notoriamente respecto al sistema con actividad normal. Además, se pueden apreciar claras diferencias entre el consumo en el sistema controlado por una única señal de reloj y cuando está sincronizado por la cadena de celdas `dCLK_cell`. En este caso, el consumo estático está relacionado con los elementos directamente controlados por la señal de reloj y aquí sí se pueden establecer diferencias más claras entre ambas metodologías de sincronización. La Tabla 4.16 se ha construido mediante la diferencia entre las Tablas 4.14 y 4.15 y recoge el consumo dinámico del sistema. En este caso se puede apreciar una mejora muy significativa en cuanto al consumo dinámico del sistema cuando el funcionamiento del mismo se distribuye a lo largo de un periodo de la señal reloj en lugar de concentrarse en las proximidades del flanco. En cuanto al consumo dinámico de la red de distribución de la

		Consumo Dinámico (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	36.95	61.57	78.92	93.32
	dCLK	27.61	39.62	60.26	84.66
Sistema sin la RDR	CLK	36.98	52.62	78.96	93.45
	dCLK	27.87	39.77	60.36	82.99
RDR	CLK	-0.03	-0.05	-0.04	-0.13
	dCLK	-0.06	-0.15	-0.10	1.67

Tabla 4.16. Comparación del consumo dinámico en el controlador PID.

señal de reloj se comprueba cómo es prácticamente nulo. Esto significa que la red de distribución de la señal de reloj presenta siempre el mismo consumo independientemente de los datos de entrada. Los conceptos de consumo estático y dinámico tienen más sentido desde el punto de vista del sistema, que es quien puede presentar o no actividad principalmente en función de la entrada de datos, la RDR siempre tenderá a mantener constante su consumo puesto que su funcionalidad siempre es la misma.

4.3.4. Transformada 1D-DWT por transformación al dominio de los índices

El tercer y último ejemplo de sistema para comprobar el efecto de aplicación de la estrategia de simulación basada en las celdas `dCLK_cell` consiste en un sistema para realizar la transformada 1D-DWT por transformación al dominio de los índices [RAM01]. Es un sistema extenso que cuenta con alrededor de 145.000 MOSFETs y con una red de distribución de la señal de reloj que divide a éste en dos fases, par e impar.

4.3.4.1. Descripción del sistema

Las nuevas aplicaciones multimedia requieren una considerable capacidad de almacenamiento y un elevado ancho de banda para la transmisión. El continuo

crecimiento de aplicaciones multimedia ha hecho necesario un importante esfuerzo en el desarrollo de metodologías para la compresión y la codificación de imágenes y vídeo. El estándar JPEG (*Join Photographic Experts Group*) hace uso de la transformada *wavelet* [VET95] como núcleo de su esquema de conversión desde el año 2000 debido a que la codificación basada en *wavelets* se ha mostrado como la mejor opción en cuanto a calidad de imagen y factor de compresión.

La transformada discreta *wavelet* unidimensional, 1D-DWT, se puede calcular mediante el algoritmo piramidal de Mallat [MAL89a, MAL89b]. En dicho algoritmo, se definen las secuencias de aproximación y detalle de orden N de una secuencia x_n mediante la siguiente descomposición multiresolución:

$$\begin{aligned} a_n^{(i)} &= \sum_{k=0}^{N-1} g_k a_{2n-k}^{(i-1)} & i = 1, 2, \dots, J \\ d_n^{(i)} &= \sum_{k=0}^{N-1} h_k d_{2n-k}^{(i-1)} & a_n^{(0)} \equiv x_n \end{aligned} \quad (4.16)$$

donde $a_n^{(i)}$ y $d_n^{(i)}$ son respectivamente las secuencias de aproximación y detalle de la octava i , mientras que g_k y h_k ($k=0, 1, \dots, N-1$) representan, respectivamente, los coeficientes de los filtros paso baja y paso alta de orden N . Las ecuaciones (4.16) describen el cálculo de la transformada *wavelet* unidimensional, donde los coeficientes del filtro se obtienen de la representación de la *wavelet* madre en tiempo continuo. La Figura 4.33 muestra la implementación de (4.16), donde se ha tenido en cuenta que una realización eficiente debe diezmar los datos a la entrada de los filtros, ya que sólo se necesitan una de cada dos muestras. El proceso de diezmado se implementa descomponiendo la secuencia de entrada en dos secuencias de datos alternativos que son almacenados en sendas cadenas de registros, desde donde se dirigen a los dos filtros para ser procesados.

Por otro lado, la conversión de un dato a su representación en el dominio de los índices consiste en la transformación:

$$\begin{aligned} x &= \left| a^y \right|_m \\ \Phi(x) &= y \end{aligned} \quad (4.17)$$

donde a es el elemento transformador o raíz. La conversión al dominio de los índices supone transformar los productos de datos, en la suma de los índices correspondientes con la consiguiente simplificación de los módulos aritméticos. En el Capítulo 2, en el apartado 2.5.2.3, se detallan más los fundamentos de la aritmética de índices.

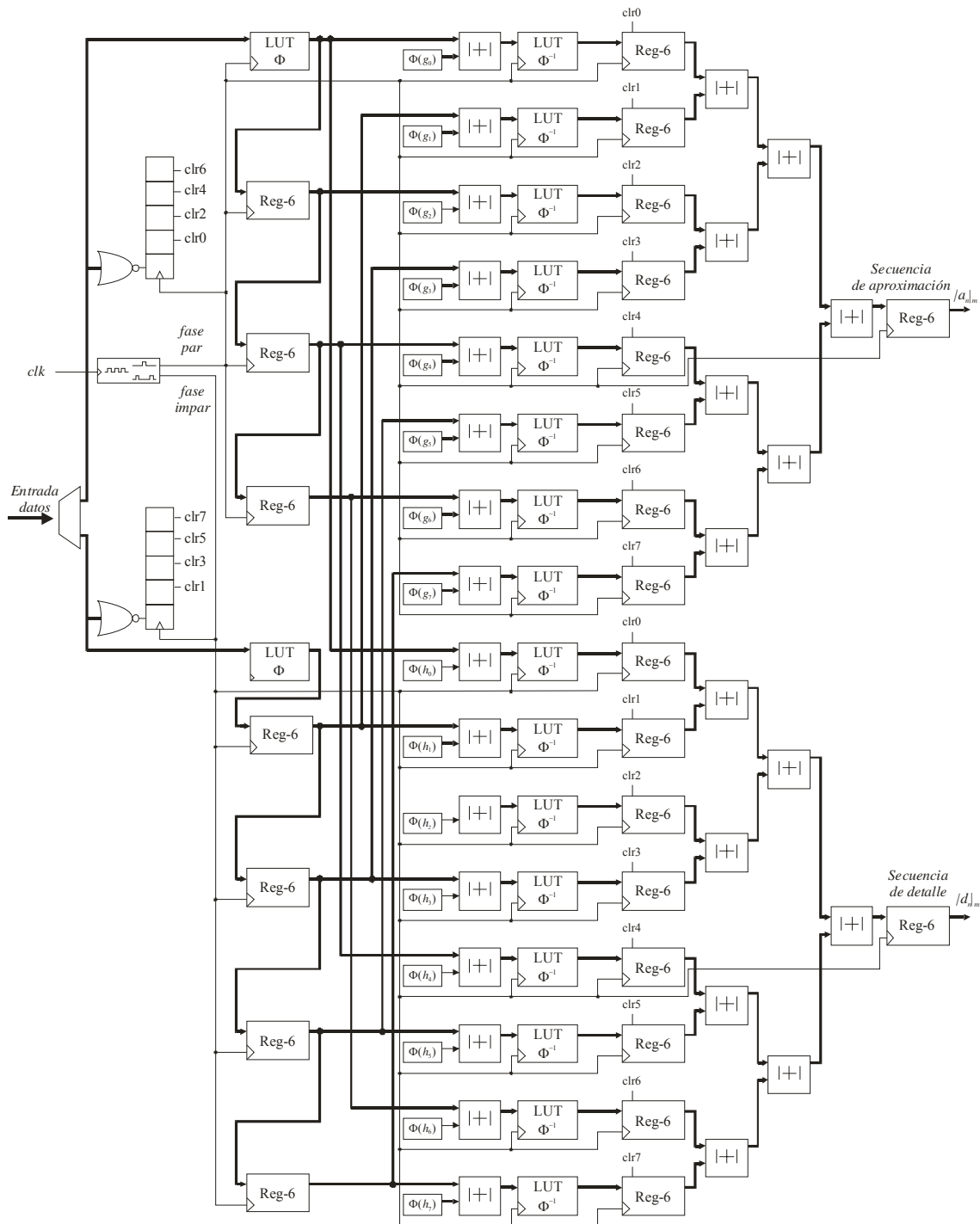


Figura 4.33. Arquitectura para la realización de la transformada 1D-DWT por transformación al dominio de los índices.

El sistema de prueba consiste en un circuito para realizar la transformada *wavelet* unidimensional (1D-DWT) por transformación al dominio de los índices [RAM01]. Esta transformación presenta la ventaja de convertir los productos de los números en sumas de sus índices, por lo que la operación producto se simplifica. Además, esta metodología permite la modificación de los coeficientes del filtro ya que, en esquemas de cómputo basados en la multiplicación mediante tablas de consulta, estos coeficientes no se pueden modificar.

La Figura 4.33 muestra la arquitectura para implementar la transformada 1D-DWT por transformación al dominio de los índices. El dato de entrada se separa en dos secuencias (par e impar) y es transformado mediante una tabla de consulta que almacena los resultados de la función Φ a su representación en el dominio de los índices. Posteriormente, los índices de cada dato se suman modularmente con los índices correspondientes de los coeficientes del filtro (g_i y h_i), lo que equivale a multiplicar el dato por dichos coeficientes. Después se realiza la transformación inversa de índice a coeficiente, Φ^{-1} , y se suman modularmente los productos resultantes. Finalmente, el filtro proporciona, a partir de una secuencia de entrada, las secuencias de aproximación y detalle, $|a_n|_m$ y $|d_n|_m$ respectivamente. El proceso de carga de los registros con los coeficientes deseados se realiza en un modo previo al procesamiento, a través de la entrada de datos al sistema y, por claridad, no se detalla en la figura. Teniendo en cuenta que no existe representación en (4.17) para $x=0$, el sistema incorpora un mecanismo de detección previa, con un registro de desplazamiento, de modo que en el momento adecuado los registros a la salida de las memorias que realizan la conversión inversa a (4.17), se ponen a cero mediante una señal de *reset*. La señal de reloj también se divide en dos secuencias de pulsos alternos de modo que la mitad del filtro actúa en la secuencia par y la otra mitad en la impar. Las dos salidas de cada canal se transforman mediante un módulo ε -CRT que aprovecha que cada salida se actualiza en pulsos de reloj alternos para multiplexar la conversión y escalado de RNS a binario de las secuencias de aproximación y el detalle.

4.3.4.2. Implementación para la simulación

El esquema global del sistema de prueba es el mismo que en los dos casos anteriores (Figura 4.34); primero se realiza la conversión de binario a RNS, después cuatro canales de módulos 61, 59, 53 y 47 realizan la transformada *wavelet* sobre el residuo

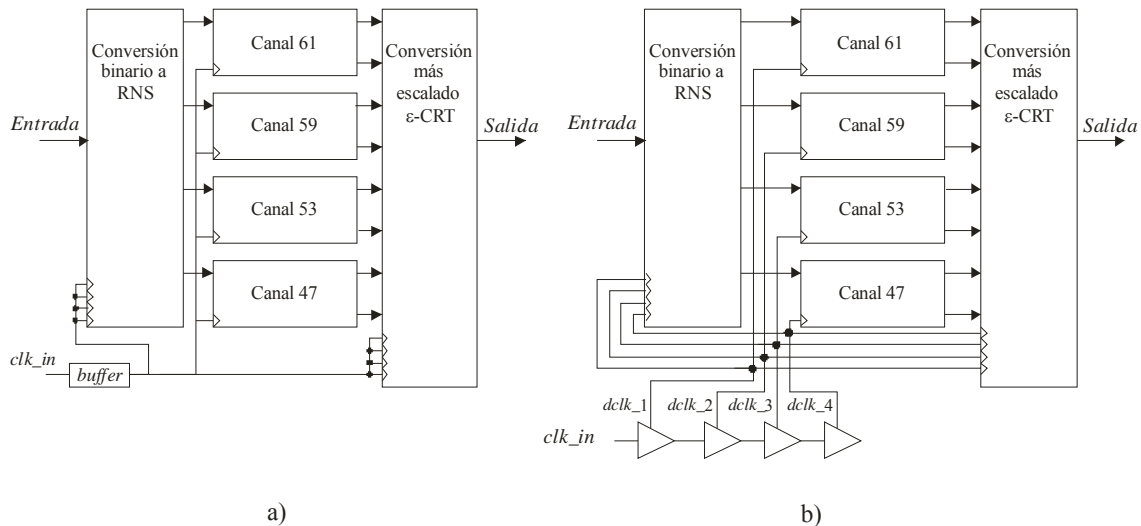


Figura 4.34. Implementación de un filtro basado en RNS para la transformada 1D-DWT *wavelets* mediante las dos estrategias de sincronización.

correspondiente, y por último la salida de los canales se transforma de nuevo a binario mediante un esquema de conversión y escalado basado en ε -CRT. Sin embargo, este sistema presenta algunas particularidades que lo diferencian de los demás. En primer lugar, la red de distribución del reloj es más compleja que en los casos anteriores; el funcionamiento del filtro requiere el procesamiento separado de datos pares e impares, lo que supone que la señal de reloj también debe ser dividida en pulsos pares e impares. Esta separación se realiza en cada canal por separado. Por otro lado, la salida del filtro consta de dos datos, aproximación y detalle, que aparecen, cada uno, cada dos pulsos de reloj, uno en el pulso par y otro en el impar, por lo que el sistema de conversión de RNS a binario debe funcionar, y consecuentemente ser sincronizado, de modo que en cada ciclo de reloj tome y convierta cuatro residuos en un dato binario de modo alternativo. El sistema posterior al filtro deberá separar los datos de aproximación y detalle que van siendo producidos sucesivamente. Este sistema de prueba acepta datos de entrada de 10 bits que transforma a RNS mediante cuatro canales de 6 bits de módulos 61, 59, 53 y 47. La elección de los módulos está condicionada porque se quiere implementar la multiplicación de los datos mediante el método de transformación al dominio de los índices en todos los canales, y dicha transformación solamente está garantizada si la base es un número primo (ver apartado 2.5.2.3). La etapa final convierte los resultados del procesamiento de cada canal en un dato de salida de 16 bits.

4.3.4.2.1. Sincronización del sistema

La Figura 4.34 muestra el sistema sincronizado con una única señal de reloj y mediante cuatro celdas dCLK_cell que controlan los cuatro canales de procesamiento. La

	Señal de reloj	Parámetro
Sistema con reloj único		W
	clk	63.4 μm
Sistema con reloj distribuido		W_b
	$dclk_1$	16.3 μm
	$dclk_2$	13.2 μm
	$dclk_3$	13.2 μm
	$dclk_4$	20.7 μm

Tabla 4.17. Dimensionado del *buffer* y de las celdas dCLK_cell.

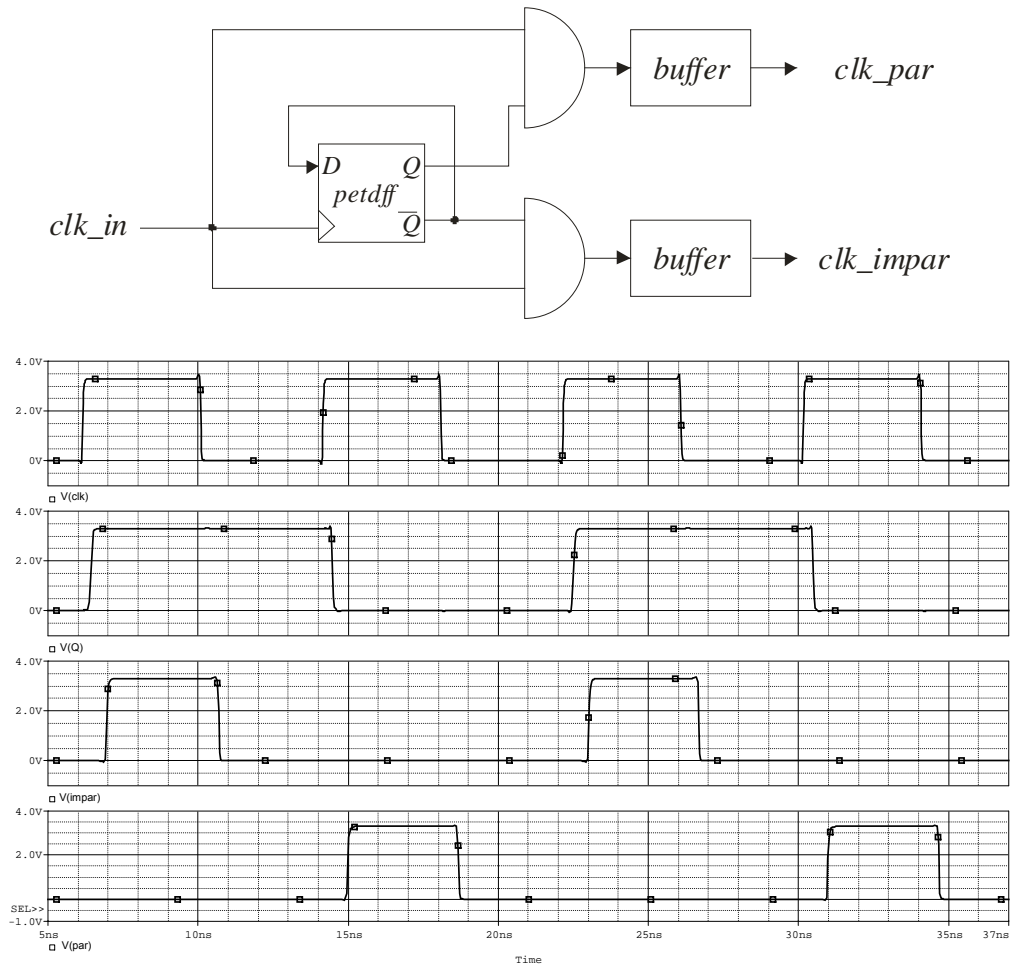


Figura 4.35. Divisor de la señal de reloj en sus fases y señales de reloj más relevantes.

Tabla 4.17 muestra los valores de los parámetros que dimensionan el *buffer* y las celdas `dCLK_cell`, dependiendo del estilo de sincronización, para controlar adecuadamente al sistema. La carga de las señales *dclk_1* y *dclk_4* es mayor puesto que la primera actúa sobre elementos comunes del módulo de conversión de binario a RNS y *dclk_4* debe actuar sobre los elementos comunes del módulo final de conversión más escalado como se verá en los siguientes apartados.

El diseño de la red de distribución de la señal de reloj en el caso que se aplican varias señales ligeramente desfasadas requiere que el retardo entre la señal más adelantada y la más retrasada sea de un cuarto de periodo, lo que implica que la cadena de celdas `dCLK_cell` debe generar un retardo de un doceavo de periodo entre señales sucesivas. La Tabla 4.18 muestra los retardos generados y los valores de los parámetros que permiten alcanzar dichos retardos.

Las señales de reloj de entrada a los módulos 1D-DWT se dividen, como muestra la Figura 4.35, en fase par e impar para actuar cada una sobre la mitad del filtro. El sistema que realiza esta división se muestra en la Figura 4.35, en la que se

Frecuencia	Periodo	Desfase por señal	L_d	W_d
40MHz	25ns	2.08 ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 3.3 \mu\text{m}$ $L_{d3} = 3.0 \mu\text{m}$ $L_{d4} = 2.8 \mu\text{m}$	$W_{d1} = 1.8 \mu\text{m}$ $W_{d2} = 13.5 \mu\text{m}$ $W_{d3} = 11.9 \mu\text{m}$ $W_{d4} = 12.0 \mu\text{m}$
64MHz	15.63ns	1.30 ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.9 \mu\text{m}$ $L_{d3} = 1.9 \mu\text{m}$ $L_{d4} = 1.8 \mu\text{m}$	$W_{d1} = 1.8 \mu\text{m}$ $W_{d2} = 14.3 \mu\text{m}$ $W_{d3} = 13.0 \mu\text{m}$ $W_{d4} = 12.0 \mu\text{m}$
100MHz	10ns	0.83 ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.2 \mu\text{m}$ $L_{d3} = 1.2 \mu\text{m}$ $L_{d4} = 1.1 \mu\text{m}$	$W_{d1} = 1.8 \mu\text{m}$ $W_{d2} = 13.0 \mu\text{m}$ $W_{d3} = 13.0 \mu\text{m}$ $W_{d4} = 13.0 \mu\text{m}$
125MHz	8ns	0.67 ns	$L_{d1} = 0.6 \mu\text{m}$ $L_{d2} = 1.0 \mu\text{m}$ $L_{d3} = 1.0 \mu\text{m}$ $L_{d4} = 1.0 \mu\text{m}$	$W_{d1} = 1.8 \mu\text{m}$ $W_{d2} = 11.5 \mu\text{m}$ $W_{d3} = 12.0 \mu\text{m}$ $W_{d4} = 10.0 \mu\text{m}$

Tabla 4.18. Dimensionado de las celdas `dCLK_cell` para alcanzar los retardos requeridos.

	Señal de reloj	Parámetro
Fases par e impar de los filtros 1D-DWT		W
	clk_par	76.1 μm
	clk_impar	78.9 μm

Tabla 4.19. Dimensiones de los buffers que generan las señales internas de los filtros 1D-DWT.

muestran también las señales de entrada, la señal Q que genera el biestable y las dos fases, par e impar, que controlan las dos partes del filtro. La señal de reloj de entrada controla al biestable activado por flanco de subida ($petdff$) realimentado con su propia salida invertida \bar{Q} . De este modo, en cada ciclo de reloj las salidas Q y \bar{Q} del biestable conmutan su valor y actúan sobre las entradas de dos puertas AND al mismo tiempo que la señal de reloj. Así, en cada ciclo de reloj, sólo una de las puertas permite el paso de la señal del reloj de entrada, mientras que la otra puerta mantiene en bajo su valor. Las salidas de las puertas AND actúan como señales de reloj, por lo que deben ser acondicionadas para ello mediante dos *buffers* amplificadores cuyas salidas clk_par y clk_impar actúan sobre el resto del sistema. Estos *buffers* están dimensionados adecuadamente según el criterio establecido en (4.2) para sincronizar correctamente los elementos internos del filtro que controlan. Dichas señales clk_par y clk_impar actúan exclusivamente dentro de cada filtro y, por lo tanto, son independientes del estilo de sincronización aplicado al sistema global, por lo que su valor es una característica constante del sistema. La Tabla 4.19 contiene los valores de los *buffers* que generan las señales clk_par y clk_impar .

4.3.4.2.2. Conversión de binario a RNS

En este sistema la entrada es de 10 bits, que se debe convertir en cuatro residuos para cada uno de los canales de seis bits. El esquema de conversión es similar al que se implementó en el controlador PID (Figura 4.27). La entrada se divide en los cuatro bits menos significativos, que no serán procesados, y los seis más significativos que se transformarán en el residuo correspondiente mediante cuatro tablas de consulta formadas por memorias de 64×6 bits. En el caso de sincronización mediante señales desfasadas, la señal de reloj más adelantada ($dclk_1$) carga el registro con los cuatro bits menos significativos, que son comunes a todos los canales, de modo que este dato

siempre esté disponible en todos los canales. Los sumadores modulares son combinacionales (Figura 4.28) y, a la salida de cada canal, un registro de 6 bits mantiene el dato estable para que sea adquirido por los filtros que realizan la transformada 1D-DWT. Tanto el registro que almacena la parte menos significativa común a todos los canales como los módulos de memoria modifican su salida en flanco de subida, mientras que el registro de salida es de flanco de bajada, por lo que la latencia del módulo de conversión es de un ciclo de reloj.

4.3.4.2.3. Canales del sistema 1D-DWT

Cada uno de los canales que constituyen el sistema se implementa tal y como muestra la Figura 4.33. Los datos de entrada de 6 bits se dividen alternativamente en dos secuencias (par e impar), al igual que la señal de reloj. Los datos de cada secuencia se convierten en su representación en el dominio de los índices mediante una LUT de 64 palabras de 6 bits. Previamente a esta transformación, se produce la detección de datos nulos que no tienen representación en el dominio de los índices, de modo que el

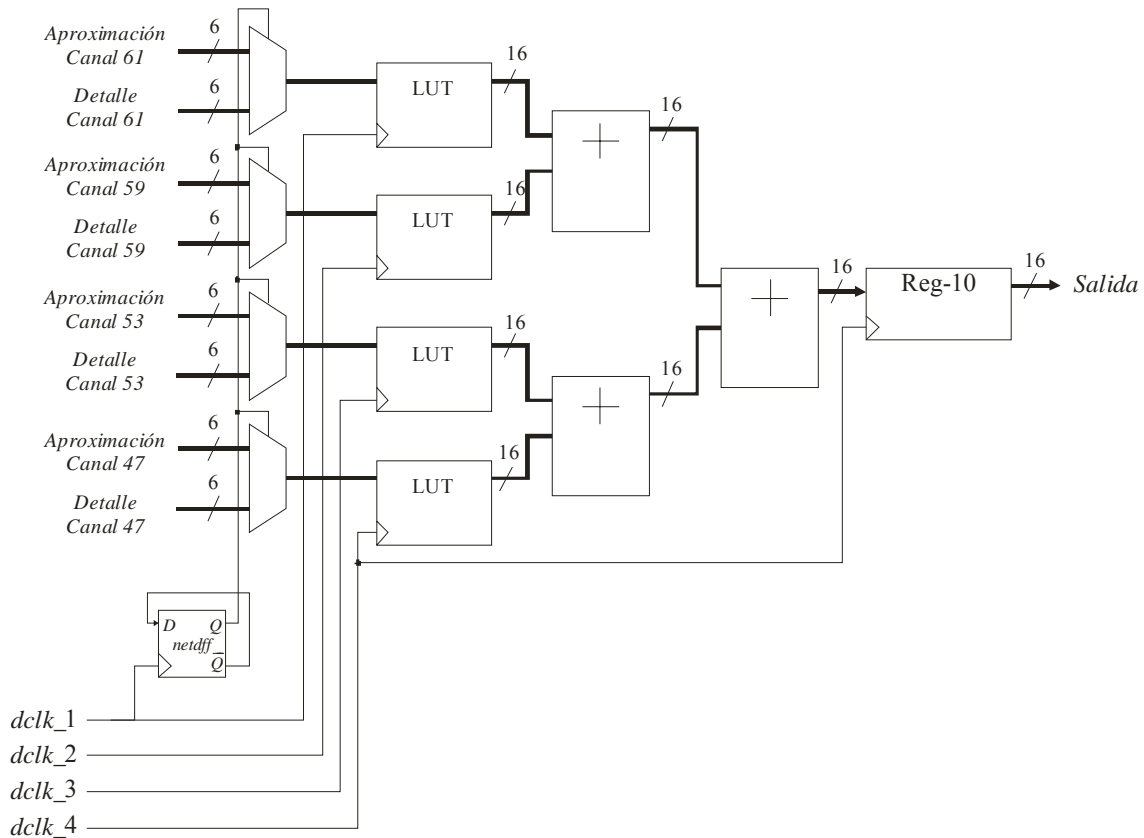


Figura 4.36. Módulo de conversión y escalado ϵ -CRT en el sistema 1D-DWT *wavelets*.

resultado del producto de un coeficiente por un dato de entrada nulo se implementa mediante una señal de *reset* que pone a cero el registro que contiene el resultado de dicho producto. La señal de *reset* se introduce en un registro de desplazamiento de modo que la latencia de la misma coincide con la propagación del dato nulo a través del canal. La representación en el dominio de los índices de los coeficientes del filtro están almacenados en registros y se suman con los índices de los datos de entrada mediante sumadores modulares combinatoriales. El resultado se transforma desde el dominio de los índices a su representación convencional mediante otra tabla de consulta cuya salida se almacena en un registro conectado a las señales de *reset* antes descritas. Las salidas de estos registros se suman modularmente en una sola etapa y su valor se almacena en dos registros de salida, uno para la secuencia de aproximación y otro para la de detalle. La latencia del módulo mostrado en la Figura 4.33 es de ocho ciclos, teniendo en cuenta que cada mitad funciona en ciclos de reloj alternos.

4.3.4.2.4. Conversión y escalado: módulo ε -CRT

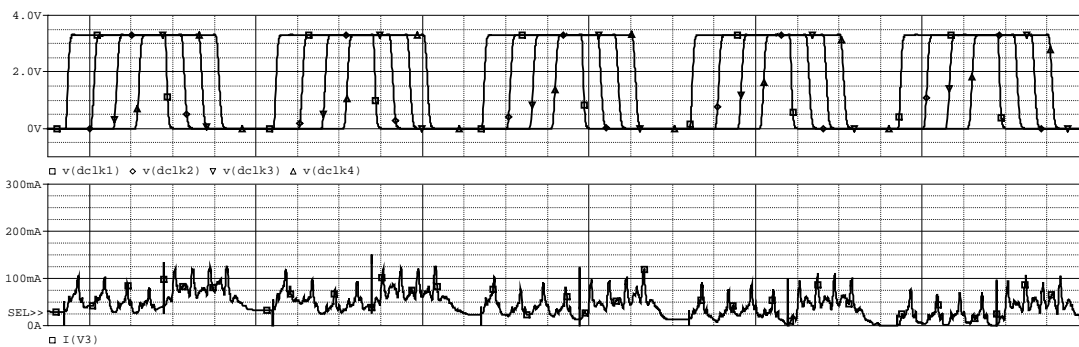
El módulo final de conversión y escalado toma las dos secuencias de salida de los canales de los filtros y, aprovechando que en cada canal se produce un dato de salida cada dos ciclos de reloj, procesa de modo alternativo un dato de la secuencia de aproximación y otro de la de detalle en cada ciclo de reloj. Esto se implementa mediante un biestable tipo *D* activado por flanco de bajada, controlado por la señal de reloj más adelantada, *dclk_1*, cuya entrada *D* está conectada a su salida \bar{Q} , de modo que, en cada flanco de bajada, la salida *Q* selecciona una de las dos entradas de cada uno de los cuatro multiplexores. Todos los multiplexores tienen sus dos entradas conectadas a las salidas de aproximación y detalle de cada uno de los canales de filtros, por lo que en cada flanco de bajada se seleccionan alternativamente una de las dos secuencias. La tabla de consulta actualiza su salida en flanco de subida, de modo que en ese instante los multiplexores mantienen estable el dato correspondiente de los filtros que realizan la transformada. La salida de cada tabla de consulta se suma mediante sumadores convencionales de 16 bits y el resultado es capturado por un registro final sincronizado por la señal más retrasada, *dclk_4*. La Figura 4.36 muestra el modo en el que esto se implementa.

4.3.4.3. Resultados de simulación

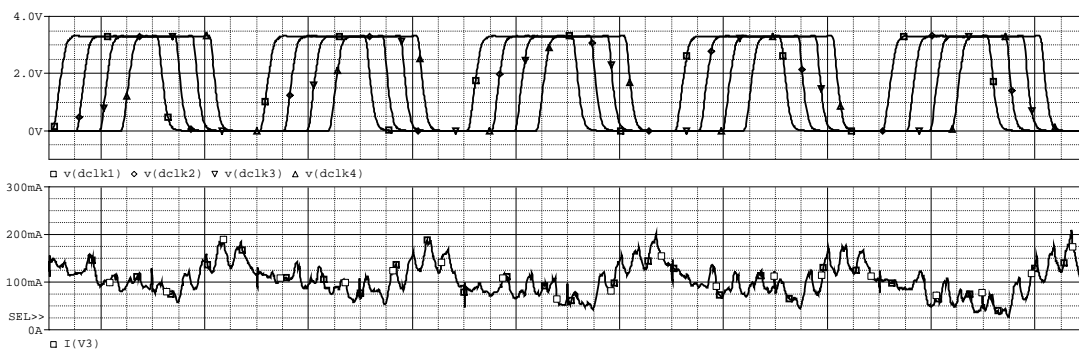
El análisis de los resultados de simulación del filtro 1D-DWT muestra cómo la reducción de la corriente máxima demandada es del 53% en promedio. En la Tabla 4.20 se muestran los valores de las corrientes máximas en los dos tipos de sincronización

	$I_{\text{máx}}$ (mA)			
	40 MHz	64 MHz	100 MHz	125 MHz
Reloj único	383.07	382.67	385.55	380.21
Reloj distribuido	150.88	155.63	209.13	208.83
Disminución de las demandas de corriente instantánea	61%	59%	46%	45%

Tabla 4.20. Corrientes máximas con reloj único y distribuido en el sistema 1D-DWT.



a)



b)

Figura 4.37. Comparación de las corrientes máximas en el sistema 1D-DWT a 40MHz y 125MHz.

analizados para cada una de las frecuencias estudiadas. Al igual que ocurrió en el caso del controlador PID, la reducción del valor máximo de los picos de corriente disminuye a medida que aumenta la frecuencia debido al solapamiento de los picos a medida que el periodo disminuye, tal como se puede apreciar en las gráficas de la Figura 4.37. La aplicación de las señales de reloj ligeramente desfasadas se ha mostrado en todos los casos de estudio como una alternativa altamente eficaz en la reducción de las demandas máximas de la corriente instantánea. La Figura 4.37 pone de manifiesto que la mejora obtenida es tanto más significativa cuanto menor es la frecuencia, ya que la superposición de picos de corriente reduce las mejoras obtenidas.

La derivada de la corriente experimenta una reducción similar a los casos anteriores aunque algo más leve. La Tabla 4.21 contiene los resultados experimentales obtenidos mediante PSpice. Se puede comprobar cómo la reducción de la tasa máxima de demanda de corriente ronda el 75% de media para la línea de alimentación principal, el 74% para el sistema sin contar con la RDR, y del 94% para la red de distribución del

		$\left(\frac{dI}{dt}\right)_{\text{máx}}$ (A/ns)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	190.80	240.37	890.62	1286.37
	dCLK	42.31	61.13	189.09	412.70
	Reducción	77.82%	74.57%	78.77%	67.92%
Sistema sin la RDR	CLK	181.08	231.03	855.38	1237.81
	dCLK	40.84	60.82	188.94	412.41
	Reducción	77.45%	73.67%	77.91%	66.68%
RDR	CLK	32.38	20.64	48.34	76.62
	dCLK	0.85	1.21	5.50	2.77
	Reducción	97.37%	94.14%	88.62%	96.38%

Tabla 4.21. Comparación de la máxima derivada de la corriente en el sistema 1D-DWT.

reloj. Estos resultados, aunque bastante ventajosos, son algo inferiores a los casos anteriores porque el sistema 1D-DWT es considerablemente más extenso y por lo tanto las demandas de corriente son superiores a los otros dos sistemas. En cualquier caso, quedan claramente demostradas las ventajas de emplear una red de distribución de la señal de reloj frente a la alternativa tradicional del reloj único, tanto en la reducción de los picos de corriente instantánea como en la reducción de la tasa de demanda de corriente de las fuentes.

En cuanto al consumo, la Tabla 4.22 contiene los resultados del consumo medio del filtro sincronizado mediante las dos estrategias de sincronización para las cuatro frecuencias que se han sido aplicadas. El incremento del consumo en este caso es muy significativo. Al igual que en los dos casos anteriores, se aprecia una leve disminución en el consumo de la red de distribución de la señal de reloj pero un incremento, en este caso más acusado, del consumo del resto del sistema. El aumento porcentual del consumo se hace menos significativo a medida que aumenta la frecuencia, tal como muestra la Figura 4.38. Este resultado muestra cómo la aplicación de la estrategia de sincronización basada en varias señales desfasadas resulta más interesante para valores más elevados de frecuencia. La explicación para este incremento tan llamativo del consumo se encuentra en la extensión del sistema. El sistema estudiado está constituido por alrededor de 145.000 MOSFETs, la mayoría de los cuales se encuentran en módulos

		Consumo (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	94.23	149.40	231.92	286.97
	dCLK	143.88	208.94	301.82	344.67
Sistema sin la RDR	CLK	75.09	118.77	184.06	227.15
	dCLK	125.82	178.78	254.87	286.97
RDR	CLK	19.14	30.63	47.86	59.82
	dCLK	18.06	30.16	46.95	58.60

Tabla 4.22. Comparación de consumos en el sistema 1D-DWT.

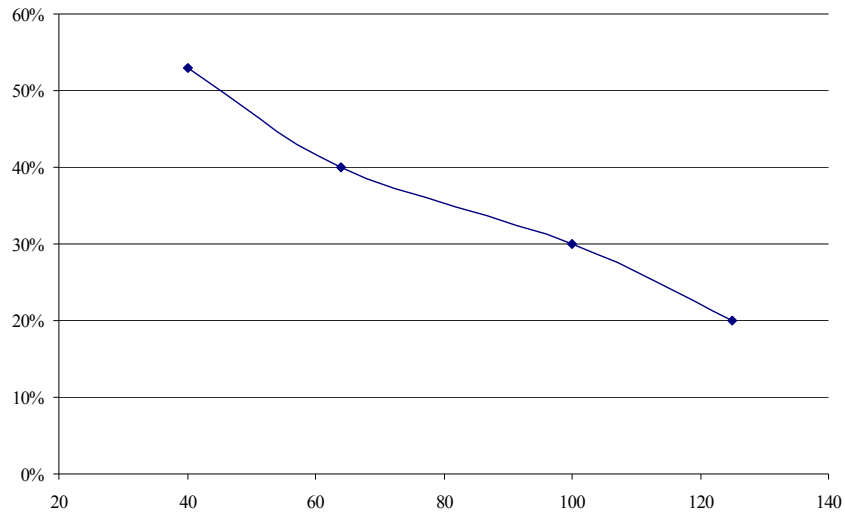


Figura 4.38. Variación porcentual del incremento de consumo al aplicar varias señales desfasadas.

aritméticos o registros que, como se ha comprobado en las simulaciones de los sistemas anteriores, son más sensibles a la calidad del perfil de la señal de reloj. La Tabla 4.23 muestra los resultados obtenidos del consumo estático del sistema. Al igual que en el caso del consumo dinámico se observa una leve reducción en el consumo de la red de distribución del reloj y un aumento significativo del consumo del sistema. Como cabía esperar todos los consumos aumentan con la frecuencia.

		Consumo estático (mW)			
		40MHz	64MHz	100MHz	125MHz
Reloj único	Sistema total	37.45	60.07	93.61	117.10
	Sistema sin la RDR	18.36	29.40	45.96	57.35
	RDR	19.09	30.67	47.65	59.75
Reloj distribuido	Sistema total	56.41	86.71	131.47	159.30
	Sistema sin la RDR	37.55	56.72	84.75	101.09
	RDR	18.86	29.99	46.72	58.21

Tabla 4.23. Consumo del filtro 1D-DWT sin actividad en la entrada de datos.

La Tabla 4.24 contiene los resultados experimentales del consumo dinámico en el sistema 1D-DWT obtenidos mediante la diferencia entre las Tablas 4.22 y 4.23. Como cabía esperar, el consumo dinámico se corresponde con el consumo del sistema sin contar la RDR. Como ya se ha comentado en apartados anteriores, y se puede comprobar en la Tabla 4.24, la RDR no presenta prácticamente variación en el consumo

		Consumo Dinámico (mW)			
		40 MHz	64 MHz	100 MHz	125 MHz
Sistema total	CLK	56.78	89.33	138.31	169.87
	dCLK	87.47	122.23	170.35	185.37
Sistema sin la RDR	CLK	56.73	89.37	138.10	169.80
	dCLK	88.27	122.06	170.12	185.88
RDR	CLK	0.05	-0.04	0.21	0.07
	dCLK	-0.80	0.17	0.23	0.39

Tabla 4.24. Comparación del consumo dinámico en el sistema 1D-DWT.

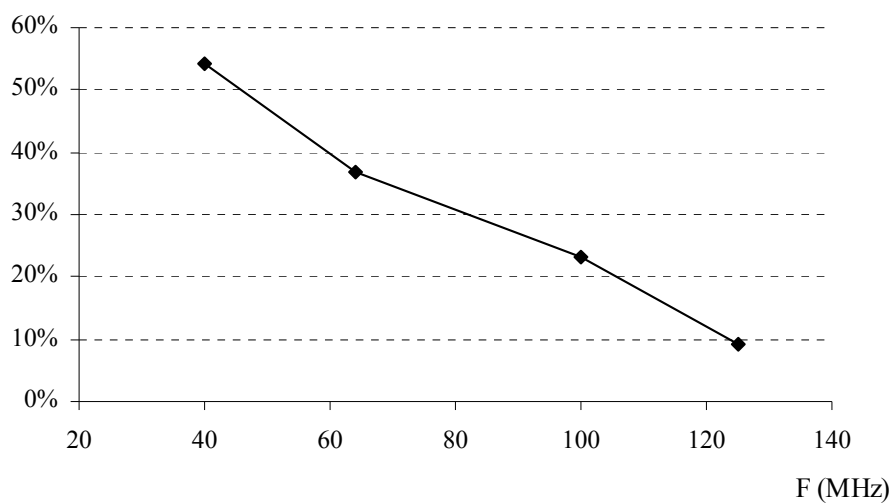


Figura 4.39. Variación porcentual del consumo dinámico en el sistema 1D-DWT.

entre el modo de funcionamiento normal y la secuencia de entrada nula. En este sistema se aprecia un incremento del consumo dinámico, debido a la aplicación de la nueva estrategia de sincronización, que se reduce a medida que aumenta la frecuencia. Desde el punto de vista del consumo dinámico, la aplicación de las señales levemente desfasadas supone una penalización de dicho consumo, que se va reduciendo a medida que aumenta la frecuencia. La Figura 4.39 muestra el incremento porcentual del consumo dinámico como consecuencia de la aplicación de la nueva estrategia de sincronización.

En el apartado siguiente se realiza un análisis comparativo entre los tres sistemas simulados.

4.3.5. Comparación de resultados

El análisis de los resultados obtenidos en las simulaciones realizadas sobre los tres sistemas en estudio muestra cómo la aplicación de diferentes estrategias de sincronización tiene efectos importantes sobre el rendimiento del sistema. En primer

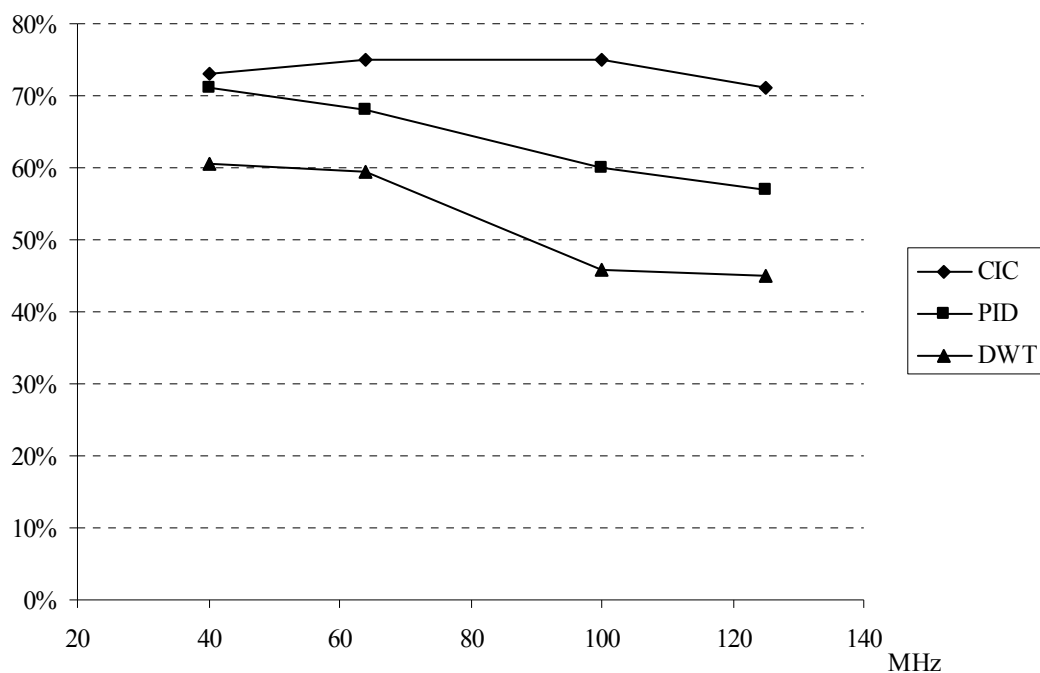


Figura 4.40. Reducción de los picos de corriente en los sistemas estudiados.

lugar, la aplicación de varias señales ligeramente desfasadas, en contraste con la estrategia convencional de una única señal de reloj, tiene como resultado una disminución muy significativa tanto de los valores máximos en las demandas de corriente del sistema como del ritmo de variación de la corriente instantánea. Este efecto es más acusado para los sistemas menos extensos y para las frecuencias bajas. Un sistema extenso requiere de más tiempo para que la lógica evalúe los datos y, por lo tanto, el funcionamiento se distribuye más, por lo que la reducción de los picos de corriente no es tan acusada como en sistemas menos extensos. Por otro lado, el incremento de frecuencia tiene como resultado el solapamiento de los diferentes picos de corriente, con lo que la reducción obtenida se ve atenuada. En todo caso, se obtienen reducciones de las demandas máximas de corriente muy significativas, tal como se puede apreciar en la Figura 4.40.

En cuanto a la variación de la corriente instantánea, se puede extraer como conclusión que la aplicación de la nueva estrategia de sincronización supone una

		Descenso porcentual de $\left(\frac{dI}{dt}\right)_{\text{máx}}$			
		40MHz	64MHz	100MHz	125MHz
CIC	Sistema total	85.86%	74.34%	79.91%	77.03%
	Sistema sin la RDR	83.99%	73.18%	78.84%	75.83%
	RDR	99.60%	99.44%	98.96%	98.29%
PID	Sistema total	92.75%	83.94%	74.70%	84.83%
	Sistema sin la RDR	89.77%	82.54%	72.49%	83.65%
	RDR	98.59%	99.30%	99.13%	97.77%
1D-DWT	Sistema total	77.82%	74.57%	78.77%	67.92%
	Sistema sin la RDR	77.45%	73.67%	77.91%	66.68%
	RDR	97.37%	94.14%	88.62%	96.38%

Tabla 4.25. Descenso porcentual de la derivada de la corriente al aplicar la nueva estrategia de sincronización

importante mejora en todos los casos, con reducciones importantes tanto en las líneas de alimentación del sistema como en las de la RDR (Tablas 4.6, 4.13 y 4.21). La Tabla 4.25 muestra el descenso porcentual medio de la derivada de la corriente en los tres sistemas para todas las frecuencias analizadas. Se puede comprobar que la derivada temporal de la corriente total se reduce, en promedio, un 79% en el filtro CIC, 84% en el controlador PID y en un 75% en el sistema que realiza la transformada 1D-DWT, y si se observa la reducción en la corriente de la RDR se obtienen reducciones del 99% en los sistemas CIC y PID y del 94% en el sistema 1D-DWT.

En cuanto al consumo del sistema, se ha comprobado que, en general, éste aumenta al emplear cadenas de celdas `dCLK_cell` con respecto a usar un único *buffer* amplificador. Esto se debe a que, mientras que el diseño del *buffer* está orientado a alcanzar rápidas transiciones mediante una cadena de dos inversores de dimensiones óptimas a ese efecto, las celdas parametrizables `dCLK_cell` están concebidas para generar un retardo configurable, de modo que se controle el instante en el que debe funcionar una parte del sistema. En todos los casos se ha apreciado un leve descenso en

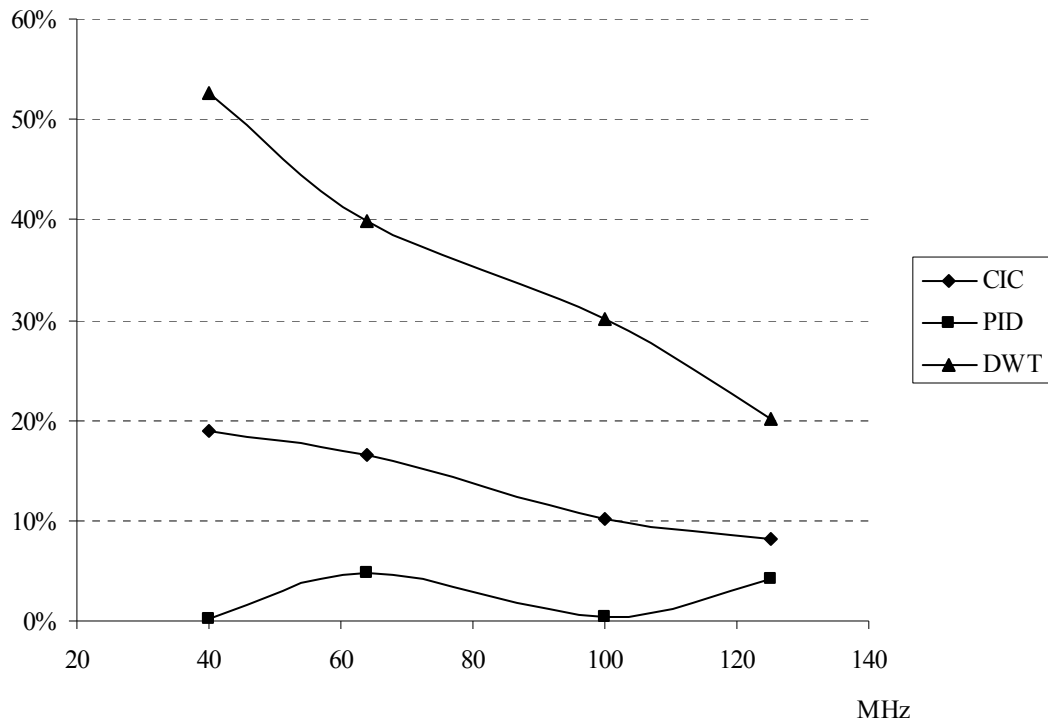


Figura 4.41. Incremento del consumo debido a la inclusión de la cadena de celdas `dCLK_cell`.

el consumo de la red de distribución cuando se emplean las cadenas de celdas parametrizables en lugar del *buffer*. Los resultados obtenidos muestran, además, cómo el impacto en el consumo depende de un modo importante del tipo de elementos de memoria del sistema. Se ha comprobado que los sistemas que hacen un uso extensivo de registros ven incrementado su consumo más que aquéllos en los que predominan las tablas de consulta, que están basadas en memorias ROM. La Figura 4.41 muestra el incremento porcentual del consumo en cada uno de los sistemas analizados cuando se reemplaza el *buffer* por la cadena de celdas parametrizables `dCLK_cell`. La Figura 4.41 muestra cómo porcentualmente el incremento del consumo se hace cada vez menos significativo a medida que aumenta la frecuencia, y cómo el controlador PID, que está fuertemente basado en tablas de consulta, apenas experimenta un incremento apreciable de consumo como consecuencia de la aplicación de la estrategia de sincronización basada en varias señales desfasadas.

El análisis del consumo separado del sistema sin incluir la RDR muestra que el incremento del consumo medio de los sistemas cuando se sincronizan mediante la cadena de celdas `dCLK_cell` es del 21% en el filtro CIC, 2% en el controlador PID y 46% en el sistema 1D-DWT. Como conclusión se extrae que la aplicación de la cadena de celdas `dCLK_cell` es muy interesante en sistemas que hagan uso extensivo de memorias ROM, puesto que:

- la penalización en cuanto al consumo es muy poco significativa en estos casos;
- la mejora en cuanto a la demanda máxima de corriente sigue siendo claramente ventajosa.

Por otro lado, el consumo exclusivamente debido a la red de distribución de la señal de reloj disminuye en un factor porcentual del 2% en el filtro CIC, 7% en el controlador PID y 3% en el sistema 1D-DWT controlado mediante la nueva estrategia de sincronización respecto a la metodología convencional de la señal de reloj única. Las variaciones en este aspecto no son muy significativas, observándose un ligero aumento en el sistema PID donde se han empleado cinco celdas `dCLK_cell` en lugar de las cuatro empleadas en los otros dos sistemas.

Las Tablas 4.8, 4.15 y 4.23 muestran el consumo estático de los tres sistemas con ambas metodologías de sincronización. A partir de ellas se puede comprobar cómo la inclusión de los recursos necesarios para generar las señales desfasadas supone un incremento significativo en el consumo tanto en el sistema global como en el sistema sin contar la RDR y, por el contrario, un leve descenso en el consumo exclusivo de la RDR. La conclusión es que, como ya se ha comentado, las señales desfasadas no presentan una calidad óptima en cuanto al consumo y en esta serie de medidas la contribución al consumo se debe fundamentalmente a los elementos síncronos, por lo que era de esperar un empeoramiento al aplicar la cadena de celdas parametrizables `dCLK_cell` con respecto al *buffer* único. Sin embargo, la cadena de celdas `dCLK_cell` por sí mismas resultan ser ligeramente más eficaces que el *buffer*, en cuanto a consumo, a pesar de emplear más recursos que éste.

A partir de los datos de las Tablas 4.9, 4.16 y 4.24, relativas al consumo dinámico, es decir, el debido exclusivamente al propio funcionamiento del sistema sin incluir el consumo estático de los dispositivos ni el consumo de la red de distribución de la señal de reloj, se construye la gráfica de la Figura 4.42. En ella se recogen los consumos dinámicos de los tres sistemas sincronizados mediante las dos metodologías de sincronización en estudio. Como se puede apreciar, el consumo dinámico del sistema se ve reducido en el caso del controlador PID y sin embargo aumenta en el caso del sistema 1D-DWT, como consecuencia del mayor porcentaje de dispositivos directamente afectados por la señal de reloj en este segundo caso, donde el uso de registros es mucho más significativo que en el caso del controlador PID. Las variaciones en el filtro CIC son menores debido a que la proporción entre elementos síncronos y no síncronos es más equilibrada. En los casos analizados se puede apreciar cómo la tendencia en los dos sistemas más extensos es que a medida que aumenta la frecuencia el incremento de consumo es menor con respecto a las frecuencias más bajas, por lo que la nueva estrategia de sincronización es más interesante para frecuencias mayores.

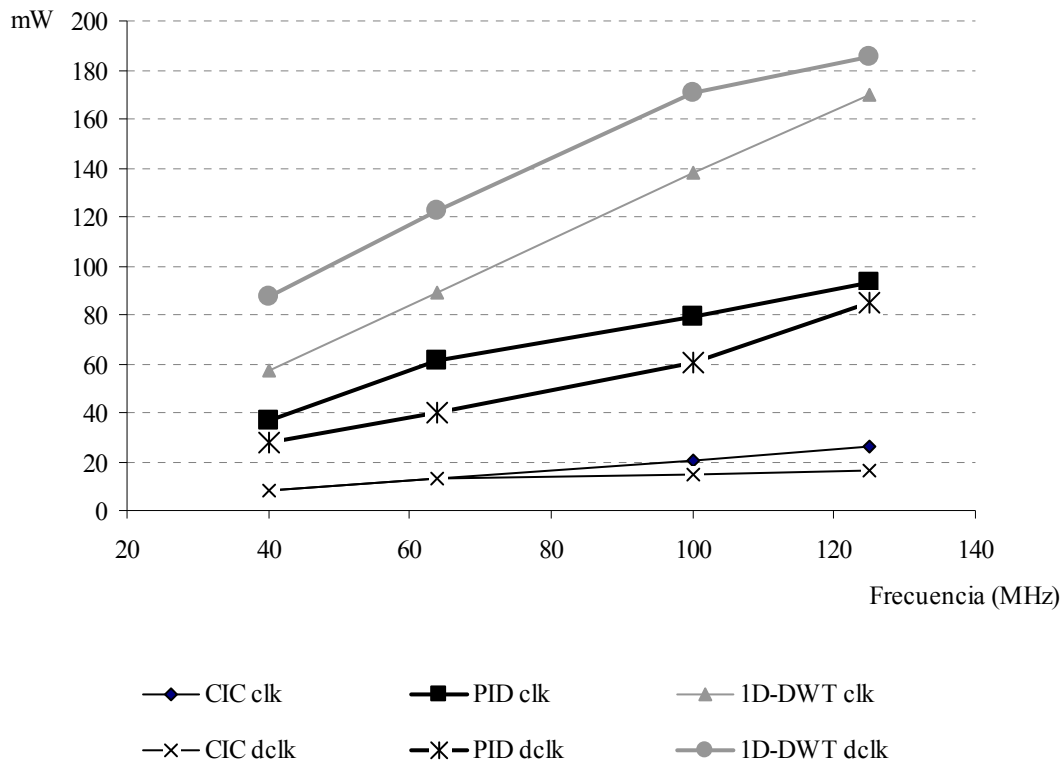


Figura 4.42. Consumo dinámico de los sistemas bajo las dos metodologías de sincronización.

4.4. Conclusión

En este capítulo se ha estudiado y caracterizado una estrategia de sincronización basada en la propagación de varias señales levemente desfasadas para sincronizar un sistema basado en RNS. También se ha definido un criterio objetivo de diseño de las redes de distribución de la señal de reloj basado en la eficacia energética del sistema sincronizado. Posteriormente, se han descrito tres sistemas de prueba sobre los que se van a aplicar las dos metodologías de sincronización en estudio, y se ha explicado cómo se implementan dichos sistemas en PSpice. Además, se ha justificado el diseño de las redes de distribución de la señal de reloj para sincronizar adecuadamente dichos sistemas. Por último, se han expuesto los resultados de las simulaciones analizando parámetros de funcionamiento tales como corriente instantánea y consumo dinámico y estático.

Los resultados obtenidos han permitido establecer que la estrategia de sincronización basada en la generación, mediante una cadena de celdas `dCLK_cell`, de varias señales desfasadas, resulta claramente ventajosa frente a la estrategia tradicional de sincronización, mediante una única señal de reloj generada mediante un *buffer*, en cuanto a la reducción de los picos de corriente dentro del circuito en todos los sistemas de prueba analizados. Las reducciones en las demandas de corriente máxima oscilan entre el 45% y el 78%, con un promedio de reducción del 63%. Estas importantes mejoras tienen una repercusión directa en la red de distribución de corriente a lo largo del chip, puesto que las líneas pueden ser escaladas a un menor tamaño con las consiguientes reducciones de área y capacidades parásitas o de acoplamiento entre líneas. Así mismo, el diseño de las fuentes de alimentación y los elementos dedicados al suministro y estabilización de la tensión a lo largo del chip pueden ver simplificadas sus especificaciones de operación.

Desde la perspectiva del consumo, la metodología de sincronización basada en las celdas `dCLK_cell` supone un incremento en el consumo global de los sistemas. Este incremento es consecuencia de una leve degradación del perfil de la señal de reloj inherente al proceso de generación del desfase entre las señales. Sin embargo, esta penalización es poco significativa en aquellos sistemas cuyo funcionamiento está basado en el uso extensivo de tablas de consulta (que están implementadas en memorias ROM), puesto que la señal de reloj actúa sobre un número relativamente menor de dispositivos que en el caso de sistemas cuyo funcionamiento se basa en registros como elementos de almacenamiento.

CAPÍTULO 5

APLICACIÓN EN DISPOSITIVOS LÓGICOS PROGRAMABLES

En este capítulo se va a implementar un sistema basado en el RNS sobre un dispositivo lógico programable FPGA (*Field Programmable Gate Array*) protegido mediante una firma digital. Se va a estudiar el efecto que tienen sobre el consumo la aplicación de varias señales de reloj ligeramente desfasadas, mediante los recursos que la propia FPGA ofrece para el control y reducción del *skew*. En este caso, dichos recursos serán empleados para generar un *skew* controlado de modo que se alcance un régimen de funcionamiento distribuido. Se realizará un análisis de los consumos global y dinámico de los sistemas bajo las diferentes estrategias de sincronización y se analizará por separado el consumo del núcleo del sistema y el sistema total. También se va a estudiar el efecto de la aplicación de varias señales desfasadas en las diferencias entre los modos de funcionamiento normal y de extracción de la firma con las dos metodologías de protección analizadas. Además, se comprobará cómo la aplicación de varias señales desfasadas fortalece al sistema frente a ataques laterales, basados en extraer información del funcionamiento del sistema mediante medidas de consumo.

5.1. Introducción

En el capítulo anterior se ha analizado una estrategia de sincronización consistente en la generación de varias señales de reloj levemente desfasadas que se aplican sobre los diferentes canales que forman parte de los sistemas basados en el RNS. De este modo, el funcionamiento global no se ve afectado por el funcionamiento no simultáneo de diferentes partes del mismo. Esto ha sido posible gracias a las características de los sistemas basados en el RNS, consistentes en canales de procesamiento paralelos e independientes. La red de distribución de las señales de reloj requiere un diseño *full-custom* ya que se especifican detalladamente las dimensiones de los transistores que la componen. Es por lo tanto una estrategia orientada a sistemas ASIC (*Application Specific Integrated Circuit*: circuito integrado de propósito específico).

En este capítulo se va a describir cómo se ha implementado uno de los sistemas analizados en el capítulo anterior en una plataforma FPGA, y cómo afecta a dicho sistema la aplicación de la estrategia de sincronización basada en la sincronización mediante varias señales de reloj levemente desfasadas. Concretamente se van a realizar medidas del consumo y se van a analizar las ventajas que se obtienen al conseguir un funcionamiento distribuido del sistema. La generación de las señales desfasadas se llevará a cabo mediante el uso de los recursos que la FPGA incorpora para compensar el *skew* generado en el sistema, es decir, se van a emplear los recursos inicialmente concebidos para minimizar el *skew*, para generarlo de un modo controlado. Con este propósito se va a implementar el sistema, ya analizado en el capítulo anterior, para la realización de la transformada 1D-DWT por transformación al dominio de los índices. En el sistema se ha incluido una firma digital para la protección de la propiedad intelectual mediante una técnica de *watermarking* [COX07]. Las medidas de consumo se realizarán sobre el mismo sistema al que se aplican diferentes metodologías de sincronización y con diferentes patrones de entrada, de modo que se pueda apreciar el efecto de la aplicación de la nueva estrategia de sincronización. Las medidas realizadas mostrarán el efecto sobre el consumo global y el consumo estático tanto de la inserción de los recursos necesarios para generar varias señales de reloj levemente desfasadas, como del efecto exclusivo del funcionamiento distribuido del sistema, en contraposición al funcionamiento concentrado en torno al flanco de una única señal de reloj. El sistema

se ha implementado en una FPGA Virtex-5 de Xilinx [XIL12b] montada sobre una placa de prototipado Virtex-5 LX [XIL11].

5.1.1. Dispositivos lógicos programables. FPGAs

Actualmente se dispone de una gran variedad de circuitos integrados para la implementación de sistemas digitales. Cuando el volumen de producción va a ser muy alto o los requerimientos de prestaciones elevados se prefieren los dispositivos ASIC. Son sistemas totalmente adaptados a una función concreta y tanto el coste como el tiempo de diseño son elevados. Las principales ventajas de estos sistemas son el bajo coste de producción por unidad y, como ya se ha dicho, el alto rendimiento con consumos de energía bajos [KUU07]. Una alternativa al diseño de sistemas ASIC es el uso de microprocesadores, que son sistemas programables de arquitectura fija, cuya flexibilidad es mayor gracias a que se puede cambiar su funcionalidad mediante cambios en el *software* que ejecutan. Sin embargo, en estos sistemas se reduce la eficiencia debido a las necesidades de lecturas continuas de memoria y las diferentes fases de decodificación y ejecución de instrucciones. Los dispositivos reconfigurables constituyen el compromiso entre flexibilidad y eficiencia, ya que se pueden configurar para que el *hardware* se adapte a la aplicación mejor que en los sistemas microprocesador o para que la funcionalidad se adapte a diferentes tareas para las que un ASIC no estaría preparado para realizar. El incremento de prestaciones y las cada vez mayores densidades de integración de los dispositivos programables los están convirtiendo en serias alternativas a los sistemas ASIC, llegando, en el caso concreto de sistemas de procesamiento de señal, a resultar una opción más eficiente [MOR11, ROS11].

Los dispositivos reconfigurables más versátiles y potentes son las FPGAs. Una FPGA está constituida por una matriz de bloques idénticos LBs (*Logic Blocks*, bloques lógicos) (Figura 5.1) que se comunican mediante una red de interconexiones programables, junto con los subsistemas de entrada/salida que permiten al usuario la comunicación con el dispositivo. El diseñador define la función que realiza cada LB así como las interconexiones entre los bloques y los sistemas de entrada y salida. El diseño del sistema se realiza mediante diversas herramientas de síntesis: bien mediante un sistema jerarquizado de bloques esquemáticos, mediante un lenguaje descriptivo del

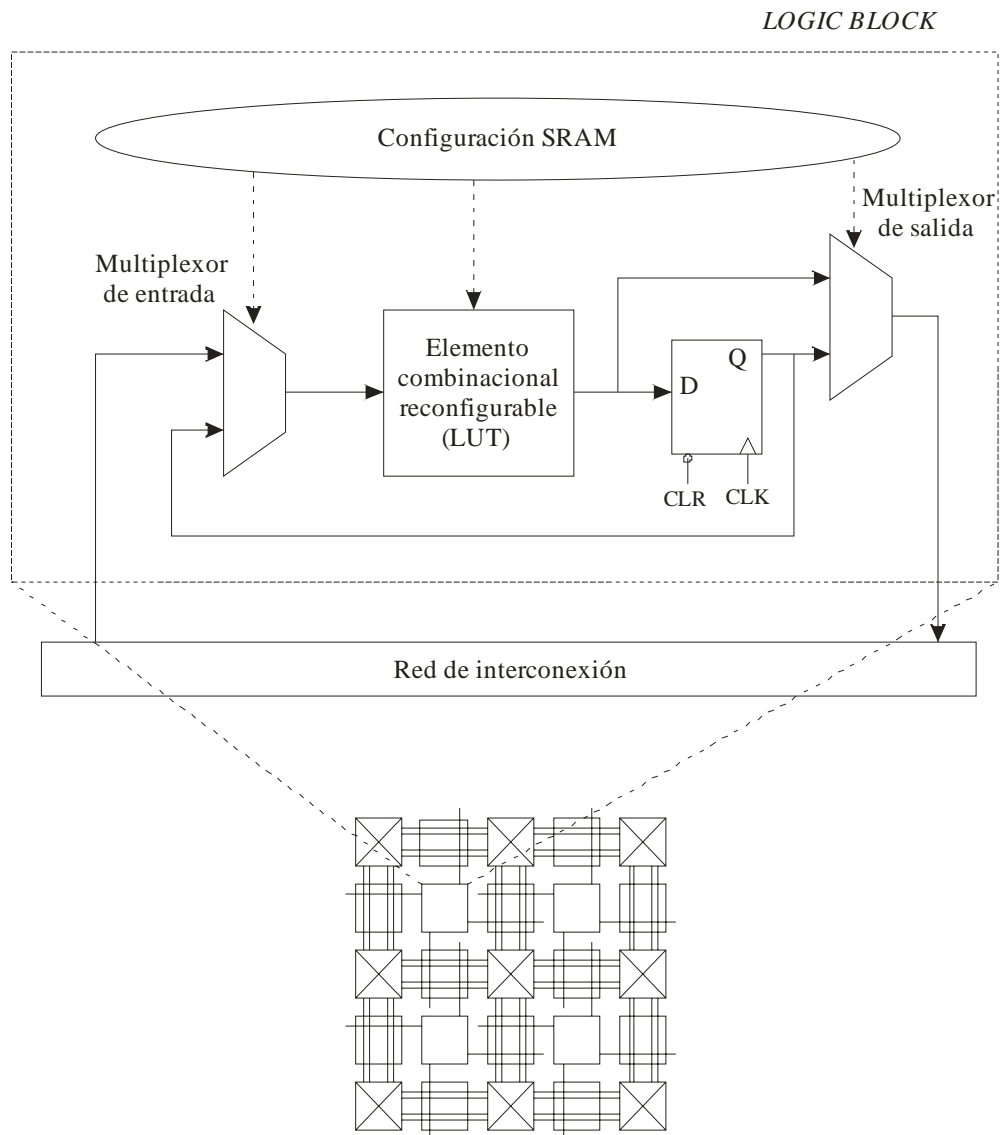


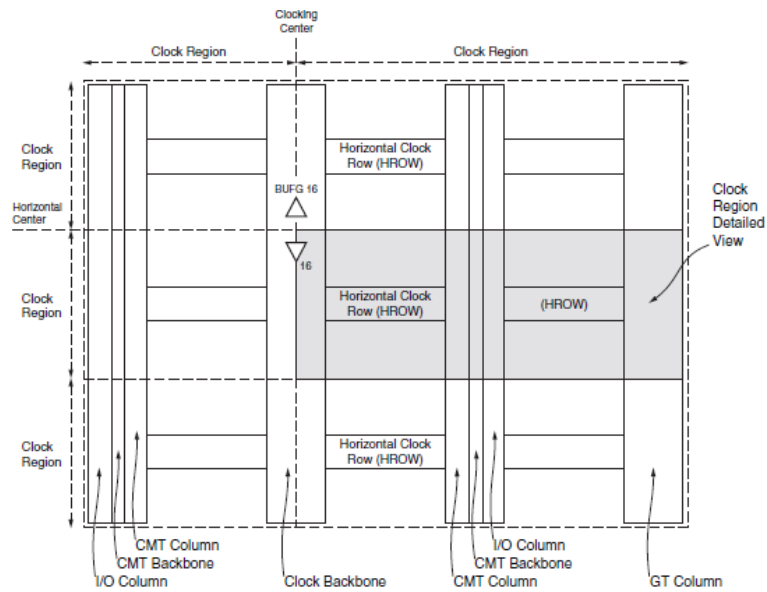
Figura 5.1. Arquitectura general de una FPGA y detalle de un Bloque Lógico.

hardware que se pretende implementar denominado HDL (*Hardware Description Language*; lenguaje de descripción de *hardware*) o una combinación de ambos. Los lenguajes HDL más frecuentes son VHDL (*Very High Speed Integrated Circuit HDL*), que es el que se ha utilizado en este trabajo y Verilog, además de nuevas alternativas de descripción *system-level* como SystemC. La programación del dispositivo se realiza mediante una secuencia de bits que definen la configuración del *hardware*. Dicha programación queda almacenada en la mayoría de los casos en una memoria SRAM (*Static RAM*) de configuración cuyo contenido es volátil y, por lo tanto, se pierde cuando se interrumpe la alimentación de la FPGA. Para solventar este inconveniente algunas FPGAs almacenan los datos de configuración del dispositivo en memorias

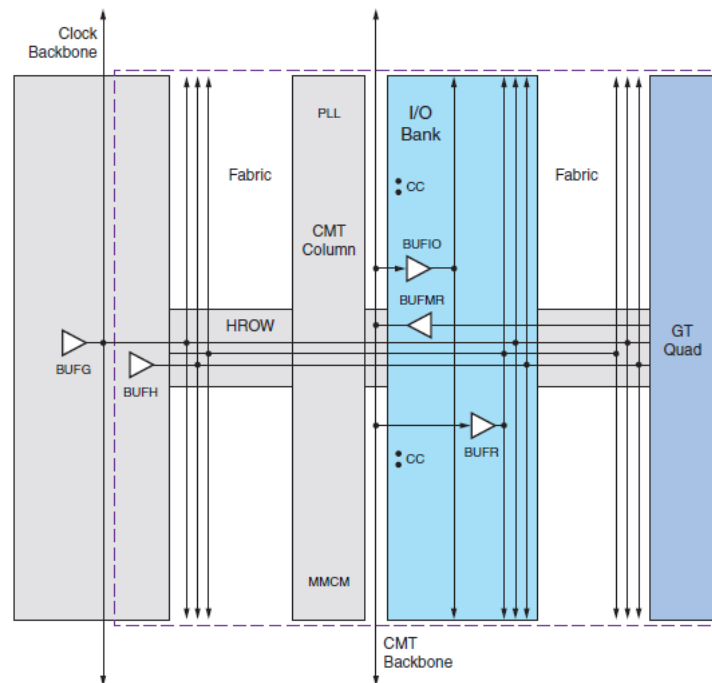
EEPROM (*Electrically Erasable Programmable ROM*), de modo que, al conectar el dispositivo, la información de configuración se extrae de estas memorias EEPROM y se carga de modo automático en las memorias SRAM. La Figura 5.1 muestra el esquema de un LB y la red de interconexiones entre diversos LBs. En la figura se puede comprobar como cada LB consta de una parte combinacional, que permite implementar funciones lógicas booleanas, y una parte secuencial formada por biestables y conectada a una señal de reloj externa, lo cual es útil para realizar circuitos secuenciales o para implementar registros. La estructura de un bloque lógico varía de un fabricante a otro, sin embargo, la parte combinacional se compone principalmente de una tabla de consulta (LUT) de n entradas que implementan cualquier función lógica.

5.1.1.1. Sincronización en FPGAs

El problema de la sincronización global de una FPGA sigue siendo una tarea complicada en cuanto al diseño y al control del *skew*. En este tipo de sistemas, al existir una red de distribución de la señal de reloj genérica y no adaptada a un tipo de sistema concreto, los problemas de sincronización son aún más significativos que en los casos en los que el diseño y posicionamiento, tanto de la red de distribución del reloj como de los bloques operativos, puede seguir criterios de minimización de los retardos. Es por este motivo que muchas FPGAs incorporan una importante cantidad de recursos para disponer de una red de distribución de la señal de reloj de alta velocidad y bajo *skew*, que sea capaz de distribuir la señal de reloj a todos los elementos síncronos del chip. En el caso de la familia Virtex-7 de Xilinx [XIL12a], la más moderna de este fabricante, los diferentes modelos de FPGA disponen de entre 8 y 24 CTMs (*Clock Management Tiles*, Celdas de Administración del Reloj) cada uno de los cuales dispone de un PLL y un MMCM (*Mixed-Mode Clock Manager*, Administrador de Reloj de Modo Mixto), ambos con capacidades de síntesis de frecuencia y filtrado del *jitter* de la señal de reloj de entrada. El MMCM además es capaz de generar desfases constantes o dinámicos de gran precisión en ambos sentidos sobre la señal de reloj de referencia y dispone de un contador fraccional que le permite realizar operaciones de síntesis avanzada de frecuencias. El dispositivo completo se divide en hasta 24 regiones de reloj cada una de ellas formada por 50 CLBs (*Configurable Logic Blocks*, Bloques Lógicos Configurables) y los recursos necesarios para distribuir la señal de reloj. Todos los chips cuentan con 32 líneas de distribución de la señal de reloj globales controladas por *buffers* que permiten el control síncrono de todos los dispositivos del circuito así como



a)



b)

Figura 5.2. Arquitectura de la distribución global de la señal de reloj (a) y detalle de la red de distribución de una región de reloj (b) en la serie 7 de Xilinx [XIL12a].

adaptarse a diversos requerimientos de elevado *fan-out*, bajo retardo de propagación o *skew* extremadamente bajo. La distribución del reloj en el nivel más alto de la jerarquía en la FPGA Virtex-7 (Figura 5.2) se basa en la arquitectura de columnas centrales (ver

Capítulo 3) horizontales y verticales que distribuyen la señal de reloj a lo largo del chip [XIL12a]. En estas columnas se incluyen *buffers*, controlados por los MMCMs o los PLLs, que permiten seleccionar entre diversas señales de reloj, minimizar los retardos de propagación y controlar el *skew*. La Figura 5.2a muestra la arquitectura global de la distribución de la señal de reloj y la Figura 5.2b el detalle de las señales de reloj distribuidas en columnas centrales de una región de reloj en la serie 7 de Xilinx [XIL12a].

El segundo fabricante de FPGAs del mercado es Altera. En sus FPGAs de alto rendimiento, la familia Stratix, incorpora los dispositivos fPLL (*fractional PLL*) con capacidades de síntesis fraccional de frecuencia avanzada y multiplicación de la frecuencia de entrada [ALT12]. Cada fPLL se puede configurar para funcionar como un PLL normal o en modo fraccional avanzado, que le permite conectar con hasta 18 contadores de salida, a partir de una única frecuencia de entrada, y minimizar el número de osciladores en la placa e interconexiones que se requieren en un diseño que haga uso de diferentes frecuencias de reloj. Adicionalmente se pueden emplear los fPLLs para reducción de retardos o compensación del *skew*. Dependiendo del nivel de integración del dispositivo se dispone de hasta 32 fPLL. Las señales de reloj se propagan a través de tres tipos de redes de distribución: 16 redes de distribución globales (GCLKs), 92 redes de distribución regionales (RCLKs) y hasta 309 redes de distribución periféricas (PCLKs) que se implementan en hasta 417 dominios de reloj. Las redes de distribución

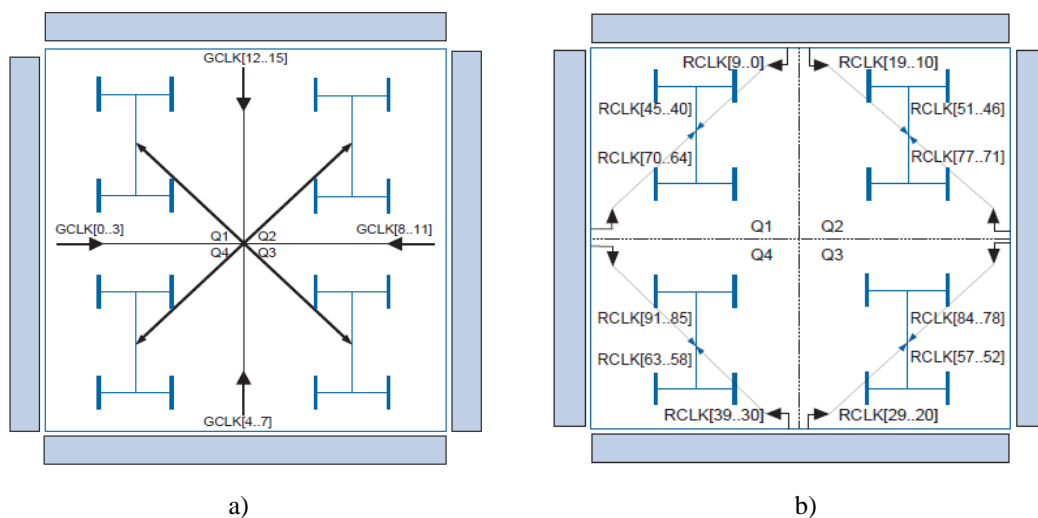


Figura 5.3. Distribuciones de reloj global (a) y regional (b) en la FPGA Stratix V [XIL12].

globales son capaces de distribuir la señal de reloj a lo largo de todo el dispositivo transmitiendo una señal de reloj de bajo *skew* a los diferentes bloques funcionales como los módulos lógicos adaptativos (ALM: *Adaptative Logic Modules*), bloques de procesamiento de señales (DSP), módulos de memoria embebidos y PLLs. La Figura 5.3a muestra la disposición de las redes GCLK. Mediante los elementos de entrada/salida del dispositivo y la lógica interna se pueden emplear las GCLKs para generar señales de reloj internas u otras señales de control de elevado *fan-out* como pueden ser señales de *clear* síncronas o asíncronas o señales del tipo *clock enable*. Las redes RCLK (Figura 5.3b) sólo actúan sobre cada uno de los cuatro cuadrantes en el que están implementadas y proporcionan las señales de reloj de retardo mínimo y bajo *skew* para la lógica contenida en el cuadrante en que están ubicadas. La Figura 5.3 muestra como la red de distribución está implementada en una topología mixta de *X-trees* y *H-trees*.

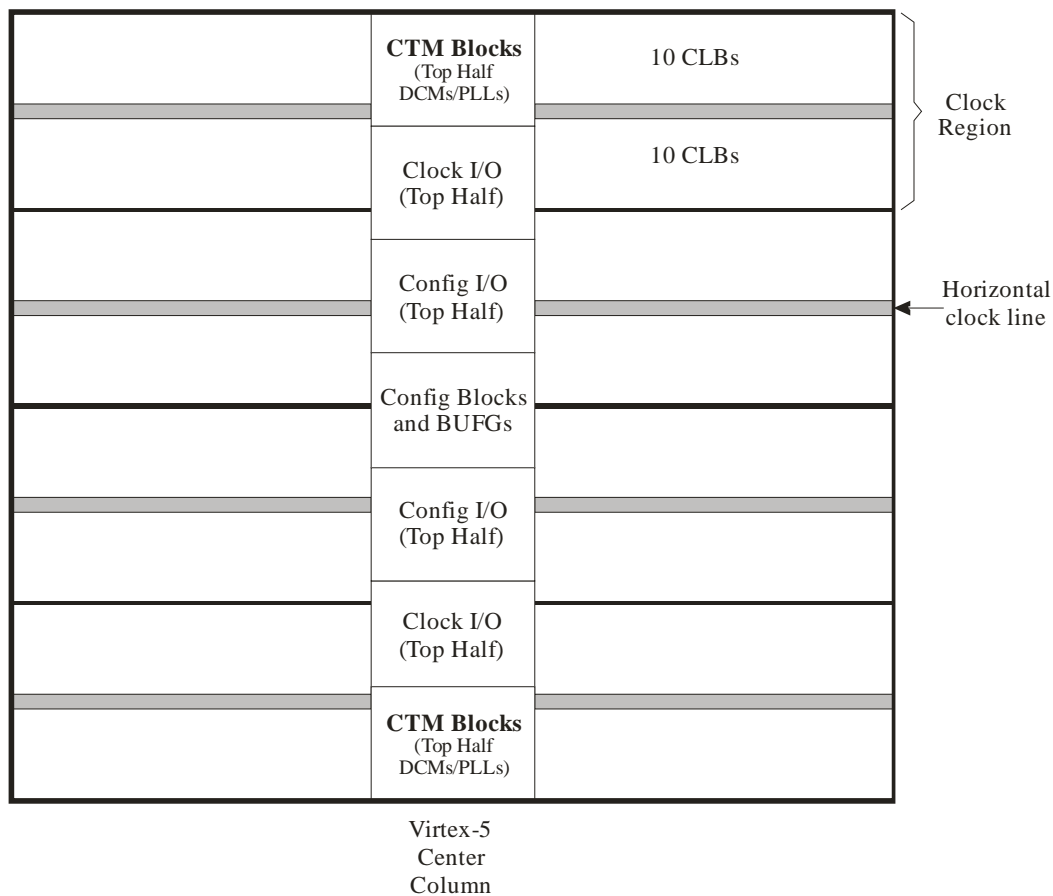


Figura 5.4. Estructura de alto nivel de la Virtex-5 XC5VLX30 [XIL12b].

Para realizar el estudio sobre el sistema de prueba seleccionado se va a utilizar una FPGA Virtex-5 de Xilinx [XIL12b], que es un sistema concebido como plataforma para diseños de altas prestaciones de baja y media densidad de integración para núcleos IP (*Intellectual Property*: propiedad intelectual) y para módulos diseñados a medida. La sincronización de la Virtex-5 se realiza desde una estructura de columna central (Figura 5.4) donde se encuentran los CTMs que contienen los DCMs (*Digital Clock Manager*, Administrador Digital de la Señal de Reloj) y los PLLs, los pines de entrada de la señal de reloj externa de referencia, la lógica de configuración y control de los *buffers* globales y los sistemas de entrada salida. La Figura 5.4 muestra la estructura general de esta FPGA, concretamente el modelo XC5VLX30, que es sobre el que se van a implementar los sistemas que se van a analizar, que consta de 8 regiones de reloj, cada una con 20 CLB (*Configurable Logic Blocks*: bloques lógicos configurables) y dividida por la mitad por una línea de distribución de la señal de reloj horizontal conectada a la columna central. Cada CLB está formado por dos *slices* interconectados entre sí mediante una matriz de interconexión externa al CLB, tal como se muestra en la Figura 5.5a. La interconexión entre CLB se realiza en forma de columna entre diferentes CLB tal como aparece en la Figura 5.5b. Cada *slice* está formado por cuatro LUTs, cuatro elementos de memoria, diversos multiplexores con funciones extendidas y lógica de propagación de acarreo. Estos elementos permiten a los *slices* realizar funciones lógicas, aritméticas y de almacenamiento ROM.

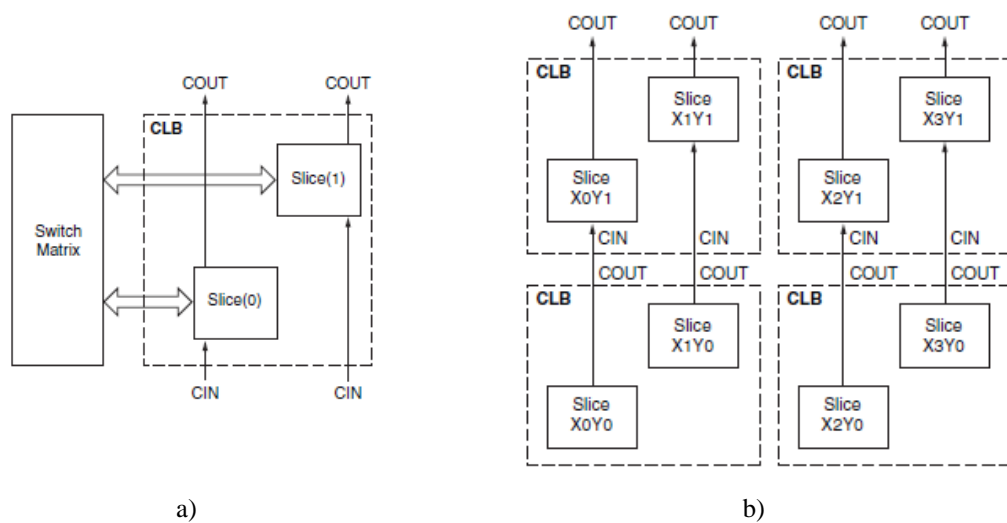


Figura 5.5. Estructura de un CLB (a) e interconexiones entre CLB y *slices* (b) [XIL12b].

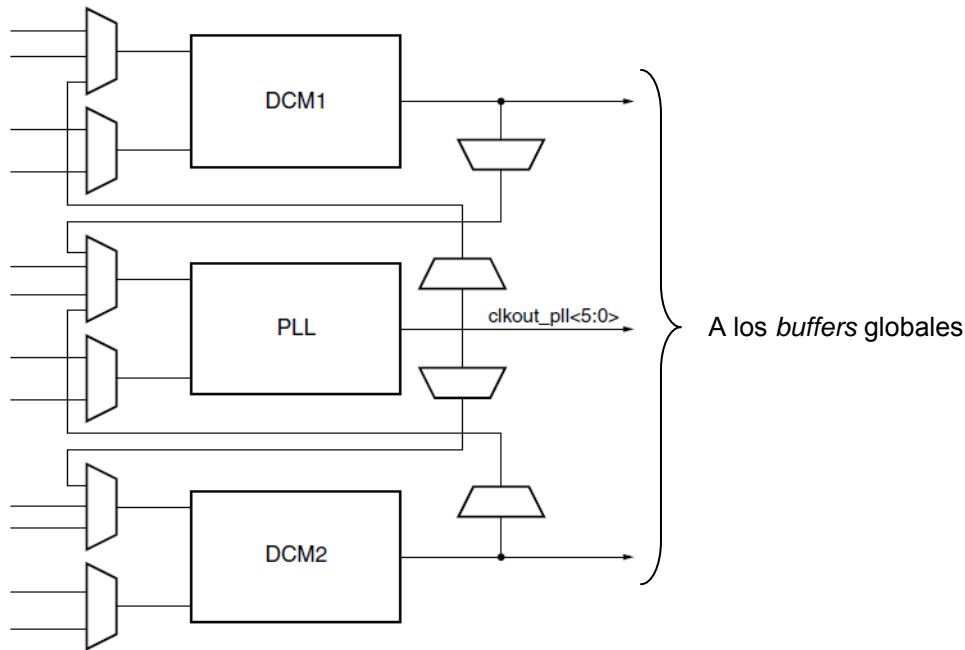


Figura 5.6. Esquema general de un CTM en la FPGA Virtex-5 [XIL12b].

La gestión de las señales de reloj en esta FPGA se realiza mediante los dos módulos CMT con que cuenta el chip, capaces de distribuir una señal de reloj a cualquier parte del sistema. Cada CTM dispone de un PLL, cuyas principales funciones son la síntesis de frecuencias y la eliminación del *jitter* tanto de señales externas como internas del chip. Complementando al PLL cada CTM cuenta con dos DCMs (Figura 5.6), cuyas capacidades son el control del *skew*, la síntesis de frecuencia, desplazamientos de la fase de la señal de reloj y su reconfiguración dinámica. Concretamente, cada DCM es capaz de generar un retardo de $n/256$ ($n=1, 2, \dots, 255$) del periodo de la señal de entrada mediante el atributo PHASE SHIFT de la primitiva CLKOUT PHASE SHIFT [XIL12b]. De este modo, la FPGA ofrece los recursos necesarios para la generación de las señales desfasadas posibilitando un ajuste muy detallado del desfase requerido. Los bancos de E/S ocupan el mismo tamaño que una región de reloj a excepción de algunos que están situados en la columna central. Cada banco dispone de cuatro entradas de reloj externas que pueden recibir entradas de reloj diferenciales o simples y distribuir a dos regiones adyacentes. La Virtex-5 dispone de 32 líneas de reloj globales capaces de controlar a todos los dispositivos síncronos del chip (CLBs, RAMs, CTMs y módulos de E/S) y además se puede usar para propagar

cualquier otro tipo de señal. Cada una de ellas está controlada por un *buffer* de reloj global que, además de emplearse para controlar la propagación de señales de reloj, puede actuar como un multiplexor libre de *glitches* con capacidad para conmutar entre dos señales de reloj diferentes o desconectar de fuentes de señales de reloj erróneas. Cada *buffer* de reloj global está controlado por un CTM, teniendo éstos capacidad de controlar más de un *buffer* simultáneamente. Adicionalmente a estos recursos globales cada región cuenta con dos *buffers* de reloj regionales y cuatro redes regionales de distribución de la señal de reloj. Cada uno de los *buffer* regionales puede controlar una red de distribución de la señal de reloj de la región adyacente. La existencia de cuatro DCMs posibilita la generación de cuatro señales de reloj levemente desfasadas para cada uno de los cuatro canales RNS del sistema que se ha implementado.

5.1.2. Protección de módulos IP

La creciente complejidad y potencia de los circuitos integrados hace cada vez más complicada la tarea de diseño de estos sistemas. Los equipos de trabajo, la productividad y las herramientas de diseño no evolucionan al mismo ritmo que las demandas de diseño de los circuitos digitales avanzados. Es por esto que un gran número de diseñadores de sistemas ASIC se están planteando la posibilidad de reutilizar módulos que ya han aplicado en otros sistemas. Algunos de los diferentes bloques operativos de un sistema se pueden utilizar en otros circuitos, ahorrando el tiempo y los recursos necesarios para volver a diseñar y verificar módulos ya utilizados [JAC01, SAH11]. Esta concepción del diseño basada en la reutilización de módulos operativos abre todo un campo de posibilidades en cuanto a la producción y comercialización de estos módulos IP (*Intellectual Property*: Propiedad Intelectual) ya terminados.

La reutilización de sistemas IP plantea el problema de la protección de la propiedad intelectual de dichos módulos [CHA00, VSI08]. La reutilización de los módulos operativos supone la colaboración entre entidades y la publicación de datos relativos a funcionamiento y aplicación. Esto supone un riesgo manifiesto frente a la piratería y al uso no autorizado de estos sistemas. Un método eficaz de protección es la técnica del *watermarking* [COX01] que consiste en ocultar información en el propio producto de modo que se pueda demostrar la propiedad y/o autoría del mismo. El interés del uso de las marcas de agua en sistemas digitales radica en tres factores: las

marcas de agua son imperceptibles, inseparables de los contenidos a los que están vinculadas y sufren las mismas transformaciones que los datos que protegen. Uno de los métodos de protección mediante *watermarking* consiste en que los módulos IP protegidos realizan la función para la que han sido diseñados mientras no se introduzca a la entrada del módulo la secuencia de extracción de la firma (SEF), en cuyo caso la salida del módulo consiste en la firma digital del mismo. La protección de módulos IP se puede realizar de un modo seguro y robusto mediante la inclusión de marcas de agua en la descripción HDL del módulo [CAS07]. El sistema así protegido incorpora la firma digital en los módulos de memoria o en los bloques combinacionales, a lo largo de todo el proceso que va desde su descripción hasta la implementación física final, independientemente de la tecnología sobre la que finalmente se va a realizar el circuito (ASIC, FPGA, etc.). A pesar de que la protección mediante marcas de agua en la descripción HDL resulta ser altamente eficaz [CAS07], existen métodos de ataque lateral [KOC99] consistentes en realizar medidas de voltaje, consumo o retardos en la propagación de las señales, de modo que se puedan medir variaciones entre el funcionamiento normal y el proceso de extracción de la firma. Este tipo de ataque es fácilmente automatizable y monitorizable y proporciona valiosa información sobre el funcionamiento interno del circuito. En este capítulo se va a analizar cómo la inclusión de las señales desfasadas fortalece al sistema protegido mediante la técnica del *watermarking* frente a los ataques laterales.

5.2. Aplicación de la nueva estrategia de sincronización a dispositivos lógicos programables

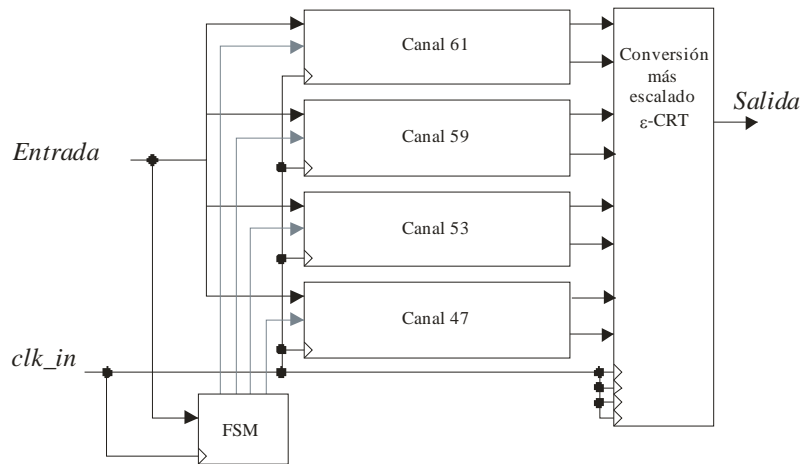
Como sistema de prueba y análisis para comprobar el efecto de la aplicación de las señales de reloj ligeramente desfasadas, se van a implementar varios sistemas que realizan la transformada 1D-DWT protegidos mediante *watermarking* desde su descripción HDL. Dichos sistemas constan de cuatro canales RNS, un módulo final de conversión más escalado ε -CRT y la lógica necesaria para la detección de la secuencia de extracción y la propagación hacia la salida del sistema de la firma digital [CAS07]. La elección de la SEF se ha realizado de dos modos diferentes e independientes: mediante un procedimiento manual y mediante un sistema basado en un LFSR (*Linear*

Feedback Shift Register: Registro de Desplazamiento con Retroalimentación Lineal). Estos sistemas se van a controlar mediante diferentes esquemas de sincronización y a diferentes frecuencias. Además, se van a introducir diferentes patrones de entrada: una secuencia de datos de entrada realistas generados con MATLAB, las secuencias de extracción correspondientes a las dos metodologías antes comentadas y una secuencia de datos nulos para medir el consumo estático del sistema. Los resultados de las medidas de consumo en estos sistemas se analizarán en dos ámbitos: desde el punto de vista del consumo del sistema en general y desde la perspectiva del contraste entre el modo de extracción de la firma y el modo de operación normal.

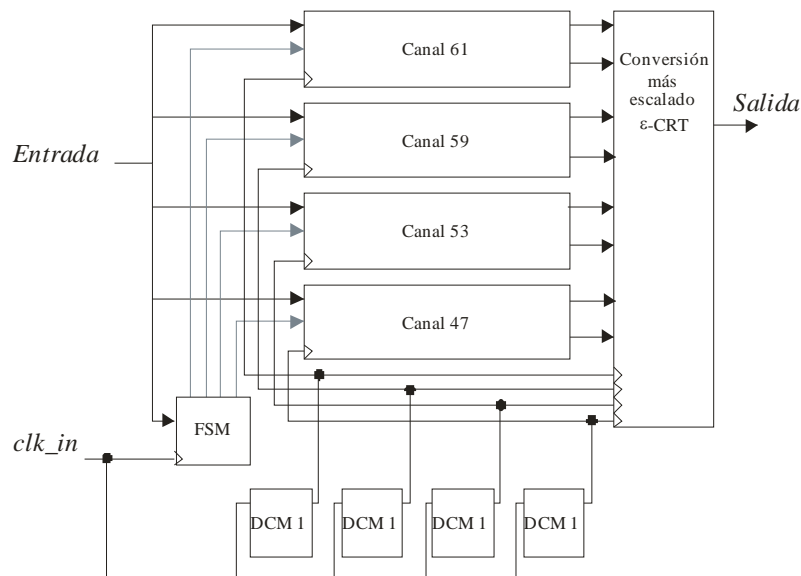
5.2.1. Descripción del sistema

El sistema de prueba consiste en un circuito para realizar la transformada discreta unidimensional basada en *wavelets* en el dominio de los índices mediante un sistema basado en el RNS. La Figura 5.7 muestra el esquema general del sistema formado por cuatro canales de módulos 61, 59, 53 y 47, sincronizados mediante una única señal de reloj y mediante cuatro señales generadas mediante los DCMs que generan el desfase requerido. Las Figuras 4.33 y 4.36 del capítulo anterior muestran el detalle de cada canal y la etapa final de conversión y escalado en la que los residuos RNS se convierten en datos finales binarios. En este caso la conversión de binario a RNS se realiza en cada canal en contraste con el sistema analizado en el capítulo anterior. La Figura 5.7 también muestra la máquina de estados, FSM (*Finite-State Machine*: máquina de estados finitos), que detecta la SEF y envía las señales de control y las posiciones de memoria en las que se oculta la firma a cada uno de los canales.

En todos los casos se aplicaron tres patrones de entrada: un patrón de datos generados con MATLAB para simular un funcionamiento real del sistema, las secuencias de extracción de la firma digital y un patrón de entrada constantemente nula. De este modo se pudieron medir los consumos del sistema en régimen normal de funcionamiento en el primer caso, en el modo de extracción de la firma en el segundo y el consumo estático en el tercer caso. A partir de las medidas de consumo estático y del consumo en los otros dos regímenes de funcionamiento se puede calcular el consumo dinámico, que es el asociado exclusivamente al funcionamiento del sistema.



a)



b)

Figura 5.7. Sistema 1D-DWT sincronizado mediante una señal de reloj única (a) y mediante DCMs (b).

5.2.1.1. Sincronización del sistema

La sincronización del sistema se ha realizado de tres modos distintos: un único reloj que sincroniza todo el sistema, mediante cuatro módulos DCM que generan señales ligeramente desfasadas respecto a la señal de entrada de referencia y, por último, se utilizan los DCMs sin que generen desfase alguno en la señal de reloj. Mediante este último estilo de sincronización se pueden comparar por un lado el impacto que tiene

sobre el consumo la inclusión de los DCMs respecto al mismo sistema sincronizado sin el uso de estos recursos y, por otra parte, el efecto exclusivo de la aplicación de las señales desfasadas sobre dos sistemas iguales por diferencia con el segundo modo de sincronización. En este último caso, cualquier variación en el consumo respecto del sistema con los DCMs generando retardos, será debida únicamente a la aplicación de las señales desfasadas frente a la alternativa de una única señal síncrona. La Figura 5.7 muestra el sistema que se ha implementado cuando está sincronizado por una única señal de reloj (a) y con la inclusión de los DCMs (b) en los dos modos de funcionamiento antes comentados.

5.2.1.2. Canales RNS de procesamiento y conversión más escalado

Cada uno de los canales RNS realiza la transformada 1D-DWT según el esquema mostrado en la Figura 4.33 del capítulo anterior. Los datos de entrada se dividen en dos secuencias formadas por datos alternos. Ambas secuencias se transforman en su representación en el dominio de los índices y se suman con la representación de los coeficientes del filtro. Una serie de registros generan la latencia adecuada para que se realice la suma de los datos en el momento adecuado en los dos bancos de sumadores, el de aproximación y detalle. Finalmente los resultados de las sumas se convierten desde el dominio de los índices a su representación normal y se suman para generar las secuencias de aproximación y detalle finales. Cada canal incorpora un sistema de detección de datos nulos, que no tienen representación en el dominio de los índices, de modo que se genera una señal de *reset* en los registros a los que le corresponde acumular el producto del dato nulo entrante por el coeficiente correspondiente. La latencia de dicha señal de *reset* correspondiente a la propagación de los productos con los datos nulos, se consigue con un registro de desplazamiento de un bit. En contraste con el sistema descrito en el capítulo anterior, cada canal incorpora la lógica necesaria para transformar el dato de entrada en el residuo correspondiente. La salida de los canales se selecciona entre la salida normal de los datos procesados o las posiciones de memoria enviadas desde la máquina de estados, mediante multiplexores de 2 a 1 controlados por la máquina de estados.

La transformación final de los residuos generados en cada canal a datos binarios se realiza mediante un módulo de conversión más escalado ε -CRT (Figura 4.36), que convierte los residuos de cada canal en los términos de la suma de (2.20).

Posteriormente se realiza la suma aritmética de dichos términos. El factor de escalado V se elige de modo que $\lfloor M/V \rfloor$ sea una potencia de dos, por lo que se puede ignorar el acarreo producido en los sumadores, tal como se explicó en el Capítulo 2. Un registro final mantiene el dato producido durante el ciclo de reloj. Puesto que las secuencias de aproximación y detalle generadas en los canales producen los datos alternativamente, cuatro multiplexores seleccionan cada una de las salidas correspondientes, de modo que a la salida del módulo de conversión y escalado van apareciendo alternativamente los datos correspondientes a ambas secuencias. Las tablas de consulta donde se almacenan los términos de (2.20) no se utilizan completamente ya que los módulos de los canales no son una potencia de 2, por lo que las últimas posiciones de las LUTs no se emplean. Es en esas posiciones donde se almacenan los datos correspondientes a la firma digital oculta, y son las posiciones que se envían desde la máquina de estados a través de los multiplexores a la salida de cada canal, una vez se ha validado la SEF.

5.2.1.3. Lógica de extracción de la firma

La protección del sistema mediante la ocultación de una firma digital requiere un subsistema capaz de detectar la SEF y generar las señales de control necesarias para acceder a las posiciones de memoria donde se encuentran los datos que constituyen la firma digital. En los sistemas implementados la firma se puede extraer mediante dos metodologías [CAS07]: un sistema basado en una máquina de estados en el que la secuencia de extracción se define manualmente, y otro en el que la secuencia se genera mediante un LFSR que además requiere un comparador y un contador junto con la máquina de estados (más sencilla que en el caso anterior). En los módulos empleados en este trabajo la firma digital se oculta en las tablas de conversión más escalado ε -CRT que se encuentran en el módulo final del sistema. A este módulo llegan las salidas de aproximación y detalle de cada canal que seleccionan mediante multiplexores entre la salida normal de datos procesados por el canal o las posiciones de memoria enviadas desde la máquina de estados. Como ya se ha explicado, las direcciones de memoria que envía la máquina de estados corresponden con las posiciones finales no utilizadas de las tablas de consulta, en las que se almacenan los sumandos de (2.20) para la conversión final más el escalado de los datos procesados en cada canal.

En el primer caso la máquina de estados tiene como entrada la del sistema $x(n)$ y como salidas 16 señales, cuatro por canal, que se aplicarán a las entradas y el control de los multiplexores finales de cada canal en las salidas de aproximación y detalle. Así se pueden seleccionar las posiciones de memoria de las LUTs en el módulo ε -CRT, donde se encuentra oculta la firma y ésta se propaga hacia la salida del sistema. Mientras los datos de entrada no constituyan la SEF, la máquina de estados se sitúa en modo de espera y mantiene todas sus salidas a cero. Cuando la máquina de estados detecta que se ha introducido la SEF completa, genera las 16 señales de control de los multiplexores y posiciones donde se encuentra la firma y ésta se propaga a través del módulo ε -CRT hasta la salida del sistema.

En la segunda metodología de extracción de la firma (Figura 5.8), la SEF se genera mediante un LFSR, que también está implementado dentro del sistema, junto con un comparador, un contador y una máquina de estados cuya finalidad es generar las señales de control y las direcciones de las posiciones de memoria donde se encuentra oculta la firma [CAS07]. En este caso la máquina de estados no es quien detecta la SEF y se activa sólo cuando la detección ha sido validada. El sistema de detección toma los datos de entrada y los va comparando con la salida del LFSR que contiene como valor inicial el primer dato de la SEF. Durante el modo de funcionamiento normal los datos de entrada son procesados con normalidad, el contador permanece inactivo y la máquina de estados tiene en valor bajo todas sus salidas. Cuando se va introduciendo la SEF el contador avanza y activa el dato siguiente del LFSR mediante la señal en_LFSR mientras se produzcan coincidencias. Cuando la secuencia de extracción se ha completado, se envía la señal en_FSM a la máquina de estados, que genera las señales

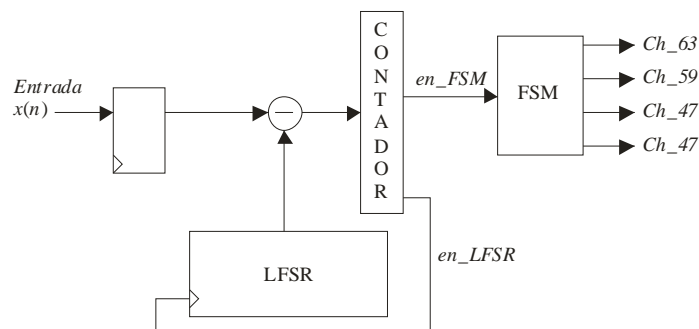


Figura 5.8. Lógica de extracción de la firma mediante un LFSR.

de control hacia los multiplexores, y genera las posiciones donde se encuentra la firma.

En ambos casos la lógica necesaria es muy reducida y las penalizaciones en cuanto a rendimiento del sistema son mínimas, debido a la simplicidad de los subsistemas.

5.2.2. Implementación del sistema

Todos los sistemas se han implementado en un dispositivo Virtex-5 XC5VLX30 de Xilinx [XIL12b], montado en una plataforma de prototipado Virtex-5 LX [XIL11]. La plataforma dispone de tres líneas de alimentación independientes VCCINT, VCCO y VCCAUX que requieren las FPGAs de la familia Virtex. La línea VCCINT, de 1 voltio, alimenta el núcleo de la FPGA, la línea VCCO, de 2.5 voltios, suministra potencia a los *drivers* de salida y VCCAUX, de 2.5 voltios, proporciona el voltaje requerido por el resto de sistemas auxiliares. Así es posible realizar medidas independientes de cada parte del sistema. La plataforma dispone de 32 entradas de reloj con 4 pares de líneas diferenciales y un zócalo de 14 pines para alojar el chip que contiene el oscilador de

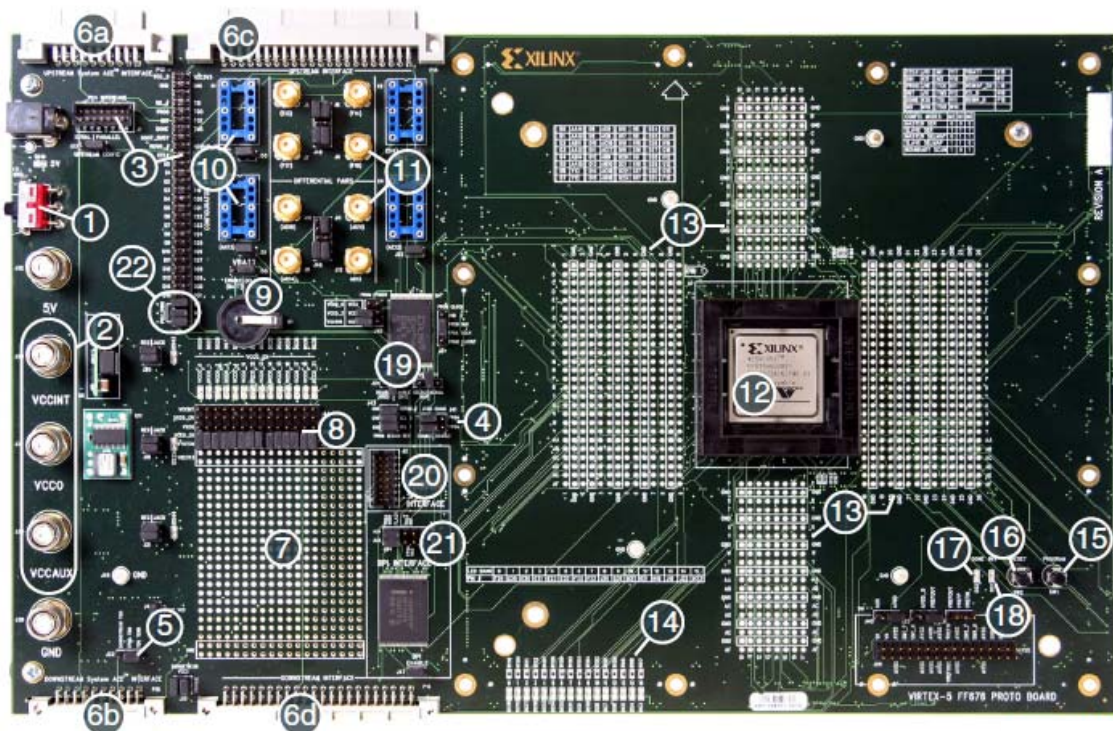


Figura 5.9. Placa de prototipado Virtex-5 LX [XIL11].

cristal que es el que se ha empleado para generar los relojes de referencia. La Figura 5.9 muestra una imagen de esta placa de prototipado.

Como ya se ha explicado, el dispositivo XC5VLX30 está formado por 8 regiones de reloj separadas por una columna central donde se encuentran los CTMs, las entradas de reloj externas y los *buffers* de reloj globales. Todas las regiones de reloj cuentan con 20 CLBs cada una, 10 CLBs por encima y 10 CLBs por debajo de la línea de distribución de la señal de reloj horizontal. El circuito dispone de 2 CMTs, cada uno de los cuales cuenta con un PLL y dos DCMs.

Para alimentar al sistema y medir la potencia consumida en cada una de las líneas de alimentación, se ha empleado un analizador de potencia N6705A DC de Agilent, que suministraba corriente a las tres líneas de alimentación; además se emplearon un generador de patrones de entrada y un analizador lógico para verificar el correcto funcionamiento del sistema. Mediante sondas de corriente y un osciloscopio se contrastaron y validaron los resultados proporcionados por el analizador lógico. Como reloj externo se emplearon chips comerciales con osciladores de 40MHz, 64MHz y 100MHz, de modo que se pudieran medir los resultados de aplicar las señales desfasadas a diferentes frecuencias.

La SEF en el caso de selección manual consta de 6 datos mientras que la SEF generada mediante un LFSR consta de 100 muestras. En ambos casos se ha aplicado una función criptográfica *hash* del tipo MD5 de 128 bits [SCH96] sobre la firma y se ha generado una secuencia de 128 bits que constituye la firma digital oculta en el diseño.

La Tabla 5.1 muestra los recursos necesarios para implementar los diferentes sistemas, con ambas metodologías para el sistema de extracción de la firma y con las dos estrategias de sincronización. En las cuatro implementaciones se requieren cuatro bloques de memoria RAM. La Tabla 5.1 también muestra la máxima frecuencia de funcionamiento del sistema en función tanto de la estrategia de sincronización como la metodología de extracción de la firma. Se puede apreciar como el sistema más eficaz en cuanto a velocidad de funcionamiento es la combinación de señales levemente desfasadas junto con el sistema de extracción basado en el LFSR.

		<i>Slices</i>		BUFGs	DCMs	$F_{\text{máx}}$ (MHz)
		Registros	LUTs			
SEF Manual	Reloj único	4768	4266	5	0	139.8
	DCMs	4818	4283	9	4	142.9
SEF con LFSR	Reloj único	4777	4257	5	0	125.4
	DCMs	4839	4297	9	4	160.0

Tabla 5.1. Recursos empleados y frecuencia máxima de funcionamiento.

5.2.2.1. Generación de las señales de reloj

Como ya se ha dicho antes, de cada sistema se han realizado tres implementaciones: reloj único, reloj único aplicado mediante DCMs (que no generan desfase) y cuatro señales de reloj levemente desfasadas mediante los DCMs. Como entradas al sistema se han empleado dos secuencias de datos: una generada con MATLAB que simula el funcionamiento real del sistema y otra consistente en la correspondiente secuencia de extracción. A continuación se describen los objetivos que se pretenden alcanzar mediante esta variedad de escenarios de sincronización y patrón de entrada.

- En el caso del patrón de entrada realista con las diferentes estrategias de sincronización, se va a medir el consumo del sistema en una situación de funcionamiento normal. Posteriormente se implementan los sistemas con las señales de reloj desfasadas y se le aplica el mismo patrón que en el caso anterior. Se pretende comparar el consumo del sistema con las dos metodologías de sincronización: una metodología de sincronización convencional de reloj único, frente a la estrategia propuesta de varias señales de reloj desfasadas.
- Sin embargo, la generación de las señales desfasadas requieren la introducción de los DCMs que no estarían operativos en el caso del sistema convencional, por lo que la comparación se realiza entre sistemas que presentan diferencias. Para apreciar tanto el efecto exclusivo de la introducción de los DCMs, como el efecto exclusivo de la sincronización mediante señales desfasadas, se realiza una implementación intermedia del sistema incluyendo los DCMs configurados de manera que no generen desfase alguno. De este modo se pueden estudiar

dos efectos: por un lado el impacto se la introducción del *hardware* adicional al comparar los dos sistemas en los que la señal de reloj no está desfasada, y por otro lado el efecto exclusivo del desfase de las señales de reloj por comparación de los resultados de los dos sistemas que incluyen los DCMs, en un caso en el que no generan desfase y en otro caso generándolo.

- Otro de los objetivos es comprobar cómo la aplicación de las señales levemente desfasadas, que suponen un funcionamiento del sistema más distribuido, afecta al consumo en los dos modos de funcionamiento: el modo normal y el modo de detección/extracción de la firma.
- Por otro lado, es interesante medir el efecto sobre el consumo de la aplicación de las señales de reloj levemente desfasadas sobre el funcionamiento exclusivo del núcleo del sistema, aislándolo de los consumos estáticos, es decir, aquéllos debidos a la propagación de la señal de reloj y al consumo estático de los dispositivos. Una vez se disponga de datos de consumo estático, por diferencia se puede calcular el consumo dinámico, es decir el debido exclusivamente al procesamiento de los datos del sistema. Para medir el consumo estático se introduce un patrón de datos de entrada constantemente nulos, de modo que se minimice la actividad del sistema.
- Por último, se realizaron medidas de consumo de todos los casos descritos anteriormente bajo tres frecuencias de funcionamiento de manera que se pueda apreciar la influencia de la frecuencia en los efectos de distribución de la señal de reloj que se están investigando.

Como se ha comentado antes, entre las capacidades de los DCMs está la generación de señales de reloj con un desfase configurable a partir de la señal de reloj de referencia. Los DCMs de la familia Virtex-5 disponen de cuatro modos de ajuste fino de desfase entre la señal de reloj de referencia y las señales de reloj de salida del DCM: el modo fijo establece un desfase constante, los modos variable-centrado y variable-positivo establecen un desfase dinámicamente reconfigurable todos ellos mediante el parámetro PHASE_SHIFT, mientras que el modo directo permite desfases en ambos sentidos que son múltiplos del parámetro DCM_TAP que representa el

	Desfase aplicado (ns)			
	DCM 1	DCM 2	DCM 3	DCM 4
40 MHz	0	2.31	4.62	6.93
64 MHz	0	1.46	2.87	4.33
100 MHz	0	0.94	1.84	2.77

Tabla 5.2. Desease generado por cada DCM en función de la frecuencia.

mínimo retardo que el DCM es capaz de generar. El valor del parámetro PHASE_SHIFT se puede elegir entre -255 y $+255$ en los modos de desfase fijo, que es el que se ha utilizado, y variable-centrado, mientras que en el modo variable-positivo este valor se selecciona entre 0 y 255 . El valor del parámetro DCM_TAP depende de la frecuencia de funcionamiento [XIL12b]. Para valores de la frecuencia de funcionamiento inferiores a 90MHz el mínimo desfase que el DCM es capaz de generar es:

$$\text{DCM_TAP} = \frac{1}{256} \times T_{CLKIN} \quad (5.1)$$

donde T_{CLKIN} es el periodo de la señal de reloj de referencia. Sin embargo, para valores superiores a 90MHz el parámetro DCM_TAP se debe obtener de la hoja de especificaciones de conmutación del dispositivo concreto, en este caso la FPGA Virtex-5 XC5VLX30 [XIL12c].

En los sistemas que se han implementado se ha tomado el mismo criterio de desfase que en el capítulo anterior, y el flanco de la señal de reloj más retrasada se encuentra a un cuarto de periodo del flanco de la señal más adelantada, tal como se puede apreciar en la Tabla 5.2. En todos los casos se aplicó una modalidad de desfase fijo con lo que se seleccionó el parámetro PHASE_SHIFT de modo adecuado para generar los desfases mostrados en la Tabla 5.2 según la expresión:

$$\Delta T = \frac{\text{PHASE_SHIFT}}{256} \times T_{CLKIN} \quad (5.2)$$

donde ΔT es el desfase obtenido, PHASE_SHIFT es el parámetro de configuración del desfase y T_{CLKIN} es el periodo de la señal de referencia.

5.3. Resultados experimentales

En este apartado se van a exponer los resultados experimentales de las medidas de consumo realizadas en los sistemas con las diferentes metodologías de sincronización, los diferentes patrones de entrada y a las diferentes frecuencias. Después se realizará el análisis del impacto en el consumo de la aplicación de la nueva estrategia de sincronización y posteriormente se estudiará como la nueva metodología de sincronización afecta a las variaciones del consumo entre el modo de funcionamiento normal y el modo de extracción de la firma para las dos metodologías implementadas.

En primer lugar se realizaron una serie de medidas del consumo de la FPGA sobre la línea VCCINT, correspondiente al núcleo de la FPGA, y sobre la línea VCCAUX que alimenta la lógica correspondiente a los sistemas auxiliares de la plataforma. Así se han obtenido medidas del consumo global de la FPGA, como la suma del consumo en las dos líneas, al tiempo que también se dispone de medidas del consumo del núcleo del sistema. En las siguientes tablas, en las que se recogen las

			Consumo del núcleo (mW)	Consumo global (mW)
40 MHz	Reloj único	Operación normal	314.3	871.1
		Extracción de la firma	287.6	845.9
	DCM sin desfase	Operación normal	351.2	1006.8
		Extracción de la firma	326.9	979.5
	DCM con desfase	Operación normal	323.9	985.9
		Extracción de la firma	300.0	957.3
64 MHz	Reloj único	Operación normal	344.2	922.0
		Extracción de la firma	302.1	873.9
	DCM sin desfase	Operación normal	388.8	1076.6
		Extracción de la firma	349.4	1034.3
	DCM con desfase	Operación normal	358.2	1050.2
		Extracción de la firma	319.7	1001.2
100 MHz	Reloj único	Operación normal	390.9	1006.7
		Extracción de la firma	325.0	941.3
	DCM sin desfase	Operación normal	455.8	1212.3
		Extracción de la firma	394.1	1141.9
	DCM con desfase	Operación normal	428.2	1185.8
		Extracción de la firma	366.8	1115.6

Tabla 5.3. Consumo del sistema con SEF manual.

medidas experimentales y en su posterior análisis, la columna de datos “Consumo del núcleo” hará referencia al consumo exclusivo del *core* de la FPGA y está asociada a la línea VCCINT, mientras que la columna “Consumo global” será relativa al consumo total del sistema, es decir, el debido al núcleo junto con el resto de sistemas auxiliares. Estas medidas se realizaron sobre los sistemas con las tres alternativas de sincronización (único reloj, señales desfasadas mediante DCMs y señales de reloj generadas en los DCMs sin desfase), a las tres frecuencias disponibles (40MHz, 64MHz y 100MHz) y con tres patrones de entrada (secuencia de datos reales, secuencias de extracción y entrada nula). Los dos primeros patrones de entrada consisten en una secuencia de datos reales generados mediante MATLAB y las secuencias de extracción correspondientes a los dos tipos de SEF implementados (manual y mediante el LFSR). El tercer patrón de entrada es una secuencia de datos constantemente nulos de modo que se puede medir el consumo estático del sistema, es decir, el consumo debido a los dispositivos bajo actividad mínima, junto con el consumo de la red de distribución de la red de reloj. De este modo se puede cuantificar el efecto en el consumo de la inclusión de los DCMs respecto al mismo sistema sin estos recursos y así calcular el consumo dinámico del

			Consumo del núcleo (mW)	Consumo global (mW)
40 MHz	Reloj único	Operación normal	305.2	923.4
		Extracción de la firma	306.1	954.8
	DCM sin desfase	Operación normal	336.6	998.0
		Extracción de la firma	337.7	1000.0
	DCM con desfase	Operación normal	335.8	996.1
		Extracción de la firma	336.8	998.4
64 MHz	Reloj único	Operación normal	336.1	958.0
		Extracción de la firma	337.6	998.4
	DCM sin desfase	Operación normal	356.3	1051.3
		Extracción de la firma	357.9	1055.2
	DCM con desfase	Operación normal	356.2	1048.6
		Extracción de la firma	357.6	1052.2
100 MHz	Reloj único	Operación normal	385.0	1045.1
		Extracción de la firma	387.3	1089.9
	DCM sin desfase	Operación normal	405.8	1158.4
		Extracción de la firma	408.0	1163.9
	DCM con desfase	Operación normal	403.9	1155.2
		Extracción de la firma	406.1	1160.5

Tabla 5.4. Consumo del sistema con SEF generada mediante LFSR.

			Consumo del núcleo (mW)	Consumo global (mW)
SEF generada de modo manual	40MHz	Reloj único	283.6	781.4
		DCMs sin skew	320.8	921.1
		DCMs con skew	296.0	902.1
	64MHz	Reloj único	295.4	795.6
		DCMs sin skew	340.7	951.2
		DCMs con skew	313.1	926.6
	100MHz	Reloj único	314.7	822.5
		DCMs sin skew	380.9	1031.0
		DCMs con skew	353.5	1005.4
SEF generada mediante LFSR	40MHz	Reloj único	277.1	814.1
		DCMs sin skew	308.9	899.9
		DCMs con skew	308.9	915.4
	64MHz	Reloj único	291.2	832.5
		DCMs sin skew	312.5	923.8
		DCMs con skew	312.6	927.4
	100MHz	Reloj único	312.6	860.9
		DCMs sin skew	338.0	988.3
		DCMs con skew	335.9	984.9

Tabla 5.5. Consumo estático del sistema.

sistema, que es el consumo debido exclusivamente a la actividad relativa al procesamiento de los datos. La Tabla 5.3 contiene los resultados del consumo en el caso del sistema con la SEF generada manualmente y la Tabla 5.4 con la SEF generada mediante el LFSR. La Tabla 5.5 muestra las medidas de consumo estático realizadas cuando la entrada al sistema era una secuencia de datos nulos. Mediante la diferencia entre las Tablas 5.3 y 5.4 con la Tabla 5.5 se obtienen resultados relativos al consumo dinámico del sistema, es decir, el debido exclusivamente al procesamiento de los datos, que se analizará en el siguiente apartado.

A continuación se van a analizar en primer lugar los resultados obtenidos en cuanto al consumo del sistema, como consecuencia de la inclusión de los DCMs y el desfase en la señal de reloj. Posteriormente se estudiara el impacto en cuanto a los modos de funcionamiento de extracción de la firma. En todos los casos los resultados obtenidos tienen su origen en los datos contenidos en las Tablas 5.3, 5.4 y 5.5.

	Incremento del consumo Reloj único vs. DCM (sin desfase)		Reducción del consumo DCM (sin desfase) vs DCM	
	Consumo del núcleo	Consumo global	Consumo del núcleo	Consumo global
40 MHz	11.7%	15.6%	7.8%	2.1%
64 MHz	13.0%	16.8%	7.9%	2.5%
100 MHz	16.6%	20.4%	6.1%	2.2%

a) SEF generada manualmente.

	Incremento del consumo Reloj único vs. DCM (sin desfase)		Reducción del consumo DCM (sin desfase) vs DCM	
	Consumo del núcleo	Consumo global	Consumo del núcleo	Consumo global
40 MHz	10.3%	8.1%	0.2%	0.2%
64 MHz	6.0%	9.7%	0.0%	0.3%
100 MHz	5.4%	10.8%	0.5%	0.3%

b) SEF mediante LFSR.

Tabla 5.6. Influencia sobre el consumo de la inclusión de los DCMs y de la aplicación de la nueva estrategia de sincronización.

5.3.1. Efecto de la nueva estrategia de sincronización sobre el consumo

Como primera conclusión de los resultados obtenidos en las Tablas 5.3 y 5.4 se obtiene un hecho esperado y coincidente con los resultados de simulación obtenidos en el capítulo anterior. La inclusión de los DCMs supone un incremento del consumo debido a la inclusión de *hardware* adicional que no está presente en el caso de la sincronización mediante reloj único. Este incremento del consumo (Tabla 5.6) adquiere un valor medio de 0.22% por MHz en el núcleo del sistema y del 0.29% por MHz para el sistema global en el caso de SEF seleccionada de modo manual y del 0.14% por MHz para el núcleo y del 0.18% por MHz para el sistema global con la SEF generada mediante el LFSR.

Otro aspecto destacable que muestra la Tabla 5.6, es que el consumo del núcleo de la FPGA disminuye en hasta un 7.9% cuando se aplican las señales de reloj desfasadas frente al mismo sistema, incluidos los DCMs, operando en modo globalmente síncrono. Esta reducción en el consumo neto es más importante en el

sistema con la SEF seleccionada de modo manual y menos significativa en el sistema basado en un LFSR. En cuanto al sistema global también se observa una reducción en el consumo, menos notoria que en el caso del núcleo, que vuelve a ser de menor magnitud en el sistema con la SEF generada mediante el LFSR. Los resultados obtenidos muestran cómo la máquina de estados que controla el sistema de protección basado en la SEF seleccionada manualmente es más sensible a las modificaciones, tanto de la señal de reloj como de su red de distribución, puesto que tanto la penalización por la inclusión de los DCMs como las ventajas de la aplicación de las señales de reloj levemente desfasadas son más significativas que en el caso del sistema de detección implementado mediante el LFSR, en el que las variaciones son de menor magnitud.

En cuanto a la influencia de la frecuencia sobre el consumo, la Tabla 5.6 muestra como para el sistema con la SEF manual se produce un incremento en el consumo del núcleo mayor a medida que se incrementa la frecuencia mientras que en el sistema con la SEF generada mediante el LFSR el incremento en el consumo del núcleo es menor. Este hecho viene a confirmar la mayor independencia en cuanto al consumo de la señal de reloj, y por lo tanto de la frecuencia, en este segundo sistema. Cuando se contempla el sistema global, en ambos casos se incrementa el consumo a medida que aumenta la frecuencia. Este hecho se explica teniendo en cuenta que en el consumo global se incluye el consumo de los sistemas auxiliares, que son independientes del modo en que se implementa el sistema en la FPGA. En cualquier caso, el incremento en el sistema basado en la SEF generada mediante el LFSR es menor al ser también menor la aportación a este consumo del núcleo del sistema. En cuanto a la reducción del consumo debida a la aplicación de las señales desfasadas se puede apreciar que su efecto es poco apreciable tanto en el consumo del núcleo como en el del sistema global.

Como principales conclusiones de este análisis preliminar sobre el impacto sobre el consumo neto del sistema se obtiene que:

- como era de esperar, la inclusión de *hardware* adicional para conseguir señales desfasadas supone un incremento del consumo del sistema;

- la aplicación de varias señales de reloj levemente desfasadas supone una reducción del consumo, tanto del núcleo como del sistema global, en contraposición a un sistema idéntico gobernado por un única señal de reloj;
- el sistema con SEF seleccionada manualmente es más sensible a la señal de reloj que el sistema con SEF generada mediante el LFSR;
- en el sistema con la SEF generada manualmente el incremento del consumo debido a la inclusión de los DCMs aumenta con la frecuencia tanto en el núcleo del sistema como en el sistema global, sin embargo al aplicar las señales levemente desfasadas el consumo se mantiene aproximadamente estable con una leve tendencia a la baja;
- en el sistema con la SEF generada mediante el LFSR el consumo de los sistemas auxiliares aumenta al incrementarse la frecuencia sin embargo se observa una reducción en el incremento del consumo del núcleo. Por otro lado, las reducciones obtenidas como consecuencia de la aplicación de las señales desfasadas se muestran prácticamente nulas e independientes de la frecuencia.

		Consumo del núcleo (mW)	Consumo global (mW)
40 Mhz	Reloj único	30.7	89.7
	DCM sin desfase	30.4	85.7
	DCM con desfase	27.9	83.8
64 MHz	Reloj único	48.8	126.4
	DCM sin desfase	48.1	125.4
	DCM con desfase	45.1	123.6
100 MHz	Reloj único	76.2	184.2
	DCM sin desfase	74.9	181.3
	DCM con desfase	74.7	180.4

Tabla 5.7. Consumo dinámico del sistema con SEF generada manualmente.

De nuevo se observa cómo la alternativa del sistema con la SEF generada mediante LFSR es menos sensible a la señal de reloj que el sistema con las SEF generada manualmente.

Mediante la diferencia entre las Tablas 5.3 y 5.4 con la Tabla 5.5 se obtiene el consumo dinámico del sistema, que es el debido al procesamiento de los datos, y queda recogido en las Tablas 5.7 y 5.8. El análisis de los datos contenidos en estas tablas muestra una reducción del consumo dinámico en todos los casos analizados respecto al caso de una única señal de reloj. Los resultados obtenidos demuestran cómo la aplicación de un leve desfase a las señales que controlan cada canal del sistema supone una reducción del consumo dinámico al respecto del sistema completamente síncrono. Los resultados obtenidos muestran cómo, la aplicación de varias señales de reloj ligeramente desfasadas sobre un sistema, suponen una mejora del consumo dinámico frente a la alternativa de una única señal de reloj síncrona. Dicho de otro modo, el consumo dinámico se reduce cuando, los diferentes canales que constituyen el sistema procesan sus datos de entrada de modo distribuido a lo largo del periodo de la señal de reloj de referencia, en lugar de hacerlo de modo simultáneo en el instante que ocurre el flanco de dicha señal de reloj.

		Consumo del núcleo (mW)	Consumo global (mW)
40 Mhz	Reloj único	28.1	84.7
	DCM sin desfase	27.7	72.6
	DCM con desfase	26.9	80.7
64 MHz	Reloj único	44.9	125.5
	DCM sin desfase	43.8	119.0
	DCM con desfase	43.6	121.2
100 MHz	Reloj único	72.4	184.2
	DCM sin desfase	67.8	170.1
	DCM con desfase	68.0	170.3

Tabla 5.8. Consumo dinámico del sistema con SEF generada mediante LFSR.

		Consumo del núcleo	Consumo global
40 MHz	DCM sin desfase	1.0%	4.5%
	DCM	9.1%	6.6%
64 MHz	DCM sin desfase	1.4%	0.8%
	DCM	7.6%	2.2%
100 MHz	DCM sin desfase	1.7%	1.6%
	DCM	2.0%	2.1%

b) SEF generada manualmente.

		Consumo del núcleo	Consumo global
40 Mhz	DCM sin desfase	1.4%	14.3%
	DCM	4.3%	4.7%
64 Mhz	DCM sin desfase	2.4%	5.2%
	DCM	2.9%	3.4%
100 Mhz	DCM sin desfase	6.4%	7.7%
	DCM	6.1%	7.5%

b) SEF generada mediante LFSR.

Tabla 5.9. Reducción porcentual del consumo debida a los DCMs.

La Tabla 5.9 muestra las reducciones porcentuales del consumo debidas a la introducción del *skew* controlado. Se puede comprobar cómo estas reducciones en el consumo dinámico del núcleo alcanzan hasta el 9.1% con respecto al sistema sincronizado con una única señal de reloj. Esta reducción es más significativa en el caso en el que los DCMs generan desfase que cuando no lo generan. La reducción del consumo debida a la introducción de un *skew* controlado es menos significativa a medida que aumenta la frecuencia. El motivo es que el *skew* inducido en el sistema es una fracción del periodo de la señal de reloj [GON05], por lo que el incremento en la frecuencia supone una reducción del periodo y, por lo tanto, las señales de reloj desfasadas tienen sus flancos más próximos, por lo que el efecto de la distribución del funcionamiento del sistema es menor. También se vuelve a comprobar cómo el sistema

		Consumo del núcleo (mW)	Consumo global (mW)
40 MHz	Reloj único	26.7	25.2
	DCM	23.9	28.6
64 MHz	Reloj único	42.1	48.1
	DCM	38.5	49.0
100 MHz	Reloj único	65.9	65.4
	DCM	61.4	70.2

a) SEF generada manualmente.

		Consumo del núcleo (mW)	Consumo global (mW)
40 MHz	Reloj único	- 0.9	- 31.4
	DCM	- 1.0	- 2.3
64 MHz	Reloj único	- 1.5	- 40.4
	DCM	- 1.4	- 3.6
100 MHz	Reloj único	- 2.3	- 44.8
	DCM	- 2.2	- 5.3

b) SEF generada mediante LFSR.

Tabla 5.10. Variaciones en el consumo entre los modos de funcionamiento normal y de extracción de la firma.

con la SEF generada de modo manual alcanza los valores más altos de reducción de consumo dinámico (a menores frecuencias) y al mismo tiempo la mayor variación en el porcentaje de reducción con la frecuencia. El sistema con el mecanismo de extracción basado en el LFSR mantiene una reducción media más estable a medida que varía la frecuencia tal como se puede apreciar en la Tabla 5.9.

5.3.2. Efecto de la nueva estrategia de sincronización sobre el proceso de extracción de la firma

Con respecto al proceso de extracción de la firma, la Tabla 5.10 muestra las diferencias de consumo entre los modos de funcionamiento normal, cuando la secuencia de datos

		Consumo del núcleo ($\mu\text{W}/\text{MHz}$)	Consumo global ($\mu\text{W}/\text{MHz}$)
40 MHz	Reloj único	668	630
	DCM	597	715
64 MHz	Reloj único	658	752
	DCM	602	766
100 MHz	Reloj único	659	654
	DCM	614	702

a) SEF generada manualmente.

		Consumo del núcleo ($\mu\text{W}/\text{MHz}$)	Consumo global ($\mu\text{W}/\text{MHz}$)
40 MHz	Reloj único	- 23	- 785
	DCM	- 25	- 58
64 MHz	Reloj único	- 23	- 631
	DCM	- 22	- 56
100 MHz	Reloj único	- 23	- 448
	DCM	- 22	- 53

b) SEF generada mediante LFSR.

Tabla 5.11. Variaciones en el consumo dinámico entre los modos de

de entrada es una serie de datos realistas generados mediante MATLAB, y el modo de funcionamiento de extracción de la firma, cuando se introducen las SEF en las dos modalidades en estudio: la SEF seleccionada manualmente y la SEF generada mediante el LFSR. Los resultados obtenidos muestran cómo en el caso de la SEF generada de modo manual las diferencias de consumo tanto del núcleo como el del sistema completo tienden a disminuir con la inserción de los DCMs, pero esta disminución es poco significativa y las diferencias en el consumo entre ambos modos de funcionamiento son apreciables. Este hecho se puede aprovechar para realizar ataques laterales [KOC99] en los que las diferencias en el consumo del sistema proporcionan información sobre el funcionamiento interno del circuito. En el sistema que utiliza el LFSR como generador de la secuencia de extracción, las diferencias entre el consumo del núcleo con las tres modalidades de sincronización son mínimas y la aplicación de la nueva estrategia de sincronización no supone una mejora en este aspecto. Sin embargo, cuando se mide el

consumo global del sistema la situación cambia radicalmente; la aplicación de las señales levemente desfasadas suponen una importante reducción de la diferencia de consumo entre los dos modos de funcionamiento estudiados. Queda probado que la aplicación de varias señales de reloj ligeramente desfasadas ayuda a proteger al sistema frente a ataques laterales en el sistema con la SEF generada mediante el LFSR.

Las diferencias en el efecto de la aplicación de la nueva estrategia de sincronización entre ambos métodos de extracción de la firma están motivadas por los diferentes mecanismos de comprobación de la SEF que, en caso manual, consta de unos pocos datos, mientras que en el caso del LFSR la secuencia de extracción está formada por muchas más muestras, que protegen al sistema de un modo mucho más robusto [CAS07]. La combinación del sistema de extracción de la firma basado en un LFSR junto con la aplicación de las señales de reloj levemente desfasadas, se han mostrado como dos métodos complementarios de protección del sistema muy eficaces.

En cuanto al consumo dinámico del sistema, la Tabla 5.11 muestra unos resultados análogos a los obtenidos en el análisis realizado sobre las variaciones del consumo neto. El sistema basado en el sistema de extracción de la firma mediante una SEF generada manualmente muestra diferencias apreciables entre ambos modos de funcionamiento tanto en el consumo del núcleo como en el consumo global. Por el contrario, en el sistema basado en el LFSR se aprecia un reducción de las diferencias de consumo dinámico importante en el consumo global, mientras que el consumo dinámico del núcleo permanece prácticamente constante. A medida que aumenta la frecuencia la reducción obtenida es menor ya que el efecto del *skew* controlado, que consiste en la distribución del funcionamiento del sistema, se atenúa al aproximarse los flancos de las señales de reloj.

En cualquier caso ha quedado demostrado que la aplicación de un *skew* controlado ayuda a la decorrelación en la actividad del sistema frente a patrones de entrada encaminados a detectar variaciones en el consumo. La introducción de los DCMs y la aplicación de un *skew* controlado tienen como resultado tanto una reducción del consumo dinámico del sistema como un fortalecimiento del sistema frente a ataques laterales contra los sistema de protección de la propiedad intelectual.

5.4. Conclusión

En este capítulo se han implementado varios sistemas basados en el RNS que realizan la transformada 1D-DWT por transformación al dominio de los índices protegidos mediante la técnica del *watermarking* digital desde la descripción HDL del módulo, sobre una FPGA Virtex-5. Se ha medido el consumo del sistema bajo dos alternativas de sincronización, una basada en una señal de reloj única y otra consistente en la aplicación de varias señales ligeramente desfasadas en virtud de los recursos de control de *skew*, DCMs, de los que dispone la FPGA. Adicionalmente, se ha medido el consumo del sistema incluyendo los DCMs pero sin que éstos generen desfase alguno, de modo que se pueda apreciar tanto la carga que supone la inclusión de dichos módulos cómo el efecto de la aplicación de varias señales desfasadas en contraposición con la sincronización mediante una única señal de reloj. Los sistemas implementados incluyen un subsistema de extracción de la firma digital no invasivo implementado en el propio sistema. Se han analizado dos métodos para implementar dicho subsistema basados en una máquina de estados en la que la SEF se genera de dos modos diferentes: manualmente o mediante un LFSR que está incluido en el sistema. En este segundo caso la secuencia de extracción es una serie de datos pseudoaleatorios de mayor extensión que en el caso de la SEF generada manualmente. Los datos de entrada han sido, por un lado, una secuencia de datos reales generados en MATLAB, las secuencias de extracción correspondientes, y una entrada constantemente nula con la cual se pretende medir el consumo estático del sistema, es decir, el consumo debido a los dispositivos cuando no realizan ningún procesamiento más el consumo de la red de distribución de la señal de reloj.

Los resultados obtenidos muestran como la aplicación de las señales de reloj levemente desfasadas tienen como resultado la reducción del consumo dinámico del sistema, así como una reducción en la diferencia entre el consumo del sistema en el modo de operación normal y el proceso de extracción de la firma digital. Esta reducción de la diferencia se ha mostrado más eficaz en el sistema protegido por una SEF generada mediante un LFSR. Esto último supone un fortalecimiento de la integridad del sistema frente a ataques laterales, que están basados en las diferencias de comportamiento en cuanto a consumo y retardos del sistema cuando se está realizando el proceso de extracción de la firma. Todas las medidas se han realizado bajo diferentes

frecuencias de funcionamiento y se ha comprobado como, a medida que la frecuencia aumenta, las ventajas obtenidas en la reducción del consumo dinámico y la decorrelación entre los modos de funcionamiento normal y de extracción de la firma se ven atenuados. Este hecho está originado por la reducción del efecto de funcionamiento distribuido al reducirse el periodo de la señal de reloj.

CAPÍTULO 6

CONCLUSIONES

En este capítulo final se concluye esta tesis y se exponen las principales aportaciones derivadas de esta investigación. Por último se plantean posibles líneas de investigación futuras.

6.1. Introducción

La sincronización de un sistema digital es una tarea complicada y crucial tanto para el correcto funcionamiento como para alcanzar el máximo rendimiento del sistema. La constante reducción del tamaño de los dispositivos tiene como resultados directos el aumento del nivel de integración de los sistemas VLSI y el incremento de la frecuencia de funcionamiento. El incremento en el nivel de integración de los dispositivos supone el aumento del tamaño relativo de los sistemas, por lo que las señales de control y datos deben recorrer rutas cada vez más largas a través de un número de dispositivos mayor. Al mismo tiempo, la reducción del tamaño de los dispositivos permite la conmutación más rápida de los mismos, por lo que la frecuencia de reloj aumenta de modo que se incrementan las prestaciones de los sistemas. El incremento de la frecuencia de funcionamiento supone que se deben reducir proporcionalmente al periodo de la señal de reloj las incertidumbres en dichas señales, de modo que el sistema disponga de márgenes de funcionamiento seguros. Las señales de reloj deben recorrer largas rutas a través de gran cantidad de dispositivos para mantener la sincronización a lo largo de todo el chip, y es fundamental para el correcto funcionamiento del sistema que dichas señales lleguen a todos los dispositivos en las mejores condiciones de calidad y bajo mínimos retardos. Las reducciones en el periodo de la señal de reloj no han ido acompañadas de modo similar al incremento de la precisión en la que ocurre el flanco de dicha señal, ni con la reducción del ruido de las fuentes de alimentación. Por todo ello han aumentado las incertidumbres tanto en el instante en el que ocurre el mismo flanco de la señal de reloj en diferentes partes del circuito (incertidumbres espaciales o *skew*) como entre flancos sucesivos de la señal de reloj en el mismo punto (incertidumbres temporales o *jitter*). Además, el incremento de la frecuencia de la señal de reloj ha puesto de manifiesto la aparición de una serie de efectos inductivos que requieren que la propagación de la señal de reloj a lo largo de una interconexión se deba tratar como una línea de transmisión. Estos efectos suponen atenuaciones diferentes en las diferentes frecuencias que componen a la señal de reloj y, por lo tanto, se produce una distorsión en la misma que empeora su calidad e incrementa las incertidumbres en el flanco.

Por otro lado, el funcionamiento completamente síncrono de un sistema formado por grandes cantidades de dispositivos supone que, en momentos muy concretos, asociados al flanco de la señal de reloj, las demandas de corriente del sistema sean muy elevadas mientras que en otros instantes sean sensiblemente inferiores. Estas demandas de corriente puntuales deben ser satisfechas por un lado con fuentes de alimentación más complejas capaces de suministrar grandes cantidades de carga en breves instantes de tiempo y, por otra parte, con líneas de alimentación de gran sección capaces de conducir grandes flujos de corriente al ritmo de la demanda. Estos factores complican el diseño de los sistemas y suponen un incremento del área y de los efectos capacitivos asociados a líneas de gran sección que deben cruzar todo el chip.

El objetivo de esta tesis ha consistido en analizar el efecto que sobre un sistema tiene la sustitución de una única señal de reloj completamente síncrona por una serie de señales levemente desfasadas, generadas en el propio sistema a partir de una señal de reloj de referencia externa, que sincronizan diferentes partes del sistema. Esta estrategia se ha particularizado para sistemas digitales basados en el RNS, ya que la naturaleza de dichos sistemas supone la existencia de canales de procesamiento paralelos e independientes. Sin embargo, los resultados son plenamente aplicables a cualquier sistema susceptible de ser descompuesto en subsistemas capaces de operar de modo independiente, aunque la entrada y salida de los datos u otras señales deban estar sincronizados globalmente. Para ello, se han realizado una serie de simulaciones eléctricas sobre varios sistemas realistas, en las que se han tomado detalladas medidas de corriente y consumo de los sistemas sincronizados mediante una única señal de reloj y mediante varias señales de reloj levemente desfasadas. Posteriormente se ha implementado uno de estos sistemas simulados en PSpice, al que se ha incorporado un sistema de protección mediante firma digital, en un dispositivo lógico programable FPGA. El objetivo era analizar el comportamiento de un sistema real bajo las dos metodologías de sincronización. En este caso se han realizado medidas del consumo del sistema y se ha analizado el efecto de la nueva estrategia de sincronización tanto en el consumo como en la decorrelación entre los modos de funcionamiento normal y de detección y extracción de la firma digital. En este segundo aspecto ha sido interesante comprobar cómo la aplicación de la nueva estrategia de sincronización ha supuesto un fortalecimiento del sistema frente a ataques laterales que se basan en el diferente comportamiento del sistema entre el modo de funcionamiento normal y el de extracción

de la firma. Los resultados obtenidos tanto en las simulaciones como en la implementación del sistema real han mostrado una serie de ventajas derivadas de la aplicación de la estrategia de sincronización analizada.

6.2. Principales aportaciones

La investigación realizada ha mostrado una serie de resultados que aportan interesantes ventajas de la aplicación de la estrategia de sincronización en estudio.

1. Se ha abordado el problema de la sincronización de sistemas basados en el RNS desde la perspectiva de la aplicación de varias señales desfasadas y se han definido las condiciones que deben cumplir, tanto las señales de reloj, como los cauces del flujo de datos dentro del sistema, para mantener la coherencia de entrada y salida de los datos globales.
2. Se ha analizado en detalle el funcionamiento y prestaciones de la celda parametrizable `dCLK_cell`, que es la generadora de las señales desfasadas, en función de los parámetros que la definen. De este modo se han obtenido los rangos de funcionamiento de la misma, para la tecnología utilizada, y las interrelaciones entre dichos parámetros. Este análisis permite establecer los valores de los parámetros según las necesidades de sincronización y retardos del sistema que se pretende sincronizar.
3. Se ha establecido una métrica, para la tecnología empleada, que permite el diseño de la red de distribución del reloj en función de la conmutación óptima, desde el punto de vista del consumo, de los dispositivos empleados. Esta métrica es crucial para obtener resultados comparativos objetivos entre diferentes metodologías de sincronización y permite establecer la relación entre las dimensiones de los *buffers* amplificadores y el área de puerta de los dispositivos controlados por ella.
4. Se ha comprobado cómo la calidad del perfil de la señal de reloj influye en el consumo de un circuito VLSI, y cómo a partir de una determinada calidad de la

señal de reloj, cualquier mejora no afecta al consumo del sistema. Esto ha permitido diseñar redes de distribución de la señal de reloj óptimas, de modo que se evita el sobredimensionamiento de las mismas, que tendría como consecuencia un incremento del consumo de la RDR.

5. Se han adaptado tres sistemas convencionales: un filtro CIC, un controlador PID y un sistema para realizar la transformada *wavelet* por transformación al dominio de los índices para operar en el sistema RNS. Se han diseñado las redes de distribución de la señal de reloj y las estructuras hardware adicionales necesarias para distribuir el funcionamiento de estos sistemas en varios canales RNS, de modo que la entrada y salida de los datos globales sean síncronas mientras que el funcionamiento interno de cada canal se realice mediante señales de reloj no sincronizadas entre sí.
6. Se ha demostrado cómo la aplicación de una serie de señales ligeramente desfasadas reduce los picos máximos de corriente de un sistema en hasta un 75%, y la tasa de incremento de la corriente en hasta un 86%, con incrementos del consumo medios del 20%. También se ha demostrado como las ventajas derivadas de la aplicación de la estrategia de sincronización analizada, tanto en reducción de picos de corriente como en la tasas de demanda de corriente, disminuyen a medida que aumenta la frecuencia como consecuencia del acercamiento de los flancos de la señal de reloj y, por lo tanto, la menor distribución del funcionamiento del sistema a lo largo del tiempo. Adicionalmente, la penalización en cuanto al consumo también se hace menor a medida que aumenta la frecuencia de funcionamiento.
7. Se ha comprobado como la estrategia de sincronización propuesta es especialmente interesante en sistemas que hacen uso extensivo de memorias ROM, en los que el incremento del consumo como consecuencia de la aplicación de las señales desfasadas es, en todas las frecuencias analizadas inferior al 5% y las reducciones de los picos de corriente oscilan entre el 71% y 57%.

8. Se han realizado simulaciones del funcionamiento de los sistemas frente a una entrada de datos nula, lo que ha permitido estimar el consumo estático del sistema, es decir, el debido al consumo de la red de distribución de la señal del reloj junto con el consumo estático de los dispositivos. A partir de estas medidas ha sido posible obtener resultados relativos al consumo dinámico del sistema, que es el debido fundamentalmente al procesamiento de los datos.
9. Se ha implementado un sistema en un dispositivo lógico programable FPGA y se ha sincronizado mediante una única señal de reloj y con varias señales levemente desfasadas. La generación de dichas señales se ha realizado mediante los recursos de los que la FPGA dispone y se han realizado medidas de consumo que muestran cómo, aunque la inclusión de *hardware* adicional supone un obvio incremento del consumo, la simple aplicación de las señales desfasadas implican una reducción del consumo del núcleo del sistema. Más aún, el consumo dinámico del sistema se ve reducido como consecuencia de la aplicación de las señales de reloj desfasadas en todos los casos analizados.
10. Se ha analizado el efecto de la nueva estrategia de sincronización sobre las diferencias de consumo entre el modo normal de funcionamiento y el proceso de extracción de la firma digital. Los resultados obtenidos muestran cómo la aplicación de las señales de reloj levemente desfasadas supone una mejora muy representativa en cuanto a la decorrelación entre los modos de funcionamiento normal y de extracción, por lo que el sistema queda más protegido frente a ataques laterales.

Estas aportaciones se recogen parcialmente en las publicaciones siguientes, derivadas en parte de este trabajo de investigación:

- Parrilla, L.; Castillo, E.; Meyer-Bäse, U.; García, A.; González, D.; Todorovich, E.; Boemo, E.; A. Lloris, A.: *Watermarking strategies for IP protection of micro-processor cores*, PROC. OF SPIE INDEPENDENT COMPONENT ANALYSES, WAVELETS, NEURAL NETWORKS, BIOSYSTEMS, AND NANOENGINEERING, vol. 7703, págs. 77030L-77030L-11, Orlando (Florida), 23 abril 2010.

- Parrilla, L.; Castillo, E.; García, A.; Todorovich, E.; González, D.; Lloris, A.: *Intellectual Property Protection of uP Cores*, PROC. OF XXIV CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS'2009), págs. 315-320, Zaragoza, 17-20 noviembre, 2009.
- González, D.; Parrilla, L.; García, A.; Castillo, E.; Lloris, A.: *Efficient Clock Distribution Scheme for VLSI RNS Enabled Controllers*, LECTURE NOTES IN COMPUTER SCIENCE, vol. 3728, págs. 657-665, 2005.
- González, D.; García, A.; Jullien, G. A.; Parrilla, L.; Castillo, E.; Lloris, A.: *Improved Clock Distribution Strategy for RNS-based DSP VLSI Systems*, PROC. OF XVIII CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS'2003), págs. 256-260, 2003.
- Parrilla, L.; García, A.; González, D.; Castillo, E.; Lloris, A.: *Efficient Polyphase Architectures for DWT Computation using RNS over FPL Devices*, PROC. OF XVIII CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS'2003), págs. 562-565, Ciudad Real, 18-21 noviembre 2003.
- González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *Clock Distribution in RNS-based VLSI Systems*. ADVANCES IN SYSTEMS ENGINEERING: SIGNAL PROCESSING AND COMMUNICATIONS (N. Mastorakis, Ed.), págs. 323-328. ELECTRICAL AND COMPUTER ENGINEERING SERIES, WORLD SCIENTIFIC AND ENGINEERING SOCIETY PRESS, (ISBN 960-8052-69-6), 2002.
- González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *A New Methodology for Efficient Synchronization of RNS-based VLSI Systems*, LECTURE NOTES IN COMPUTER SCIENCE, vol. 2451, págs. 188-197, Sep. 2002.

- González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *A New Clock Distribution Strategy in RNS-based VLSI Systems*, PROC. OF XVII CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS'2002), págs. 175-180, 2002.
- González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *Clock Distribution in RNS-based VLSI Systems*. PROC. OF 2002 WSEAS INTERNATIONAL CONFERENCE ON ELECTRONICS AND HARDWARE SYSTEMS (IEHS'2002), págs. 1511-1516, 2002.

6.3. Líneas de investigación futuras

El trabajo desarrollado en esta tesis deja abiertas varias líneas de investigación que podrían representar la continuación del trabajo recogido en esta memoria:

1. Toda esta investigación ha estado centrada en la sincronización de sistemas basados en el RNS. Sin embargo, la estrategia de sincronización estudiada es aplicable en cualquier sistema susceptible de descomponerse en módulos independientes, a condición que la entrada y/o la salida de datos globales fueran síncronas. La búsqueda de nuevos sistemas a los que aplicar la estrategia de sincronización analizada abre un amplio e interesante campo de estudio.
2. Las simulaciones han mostrado cómo la aplicación de varias señales desfasadas, en contraposición de una única señal de reloj, reduce significativamente las demandas instantáneas de corriente, aunque supone un incremento global del consumo. Esta estrategia de sincronización podría ser de interesante aplicación en sistemas en los que se dispusiera de un flujo de energía constante pero de baja potencia, como por ejemplo sistemas alimentados por energía solar, sistemas implantados en el cuerpo humano que aprovechasen gradientes de temperatura u otras opciones de *harvesting* o *skavenging*.
3. El análisis realizado en las simulaciones está basado en una tecnología concreta y se ha comprobado como la respuesta de los sistemas está influenciada por el

tipo de elementos que los constituyen. Resultaría interesante extender a otras tecnologías o analizar estrategias para alterar la implementación hardware de un sistema, por ejemplo incrementando la cantidad de LUTs, para hacer que la aplicación de las señales desfasadas sea más ventajosa.

6.4. Conclusión

Con la presentación de las principales aportaciones y la exposición de las posibles líneas de investigación futuras concluye esta memoria. Se ha analizado el problema de la sincronización de sistemas digitales VLSI y se ha estudiado la viabilidad de una estrategia de sincronización alternativa para sistemas basados en el RNS extendible a otros tipos de sistemas en los que puedan existir módulos capaces de funcionar independientemente. Se han analizado las principales ventajas e inconvenientes de la estrategia de sincronización y se ha aplicado a sistemas realistas desde dos perspectivas: la simulación eléctrica y la implementación en dispositivos programables. En los resultados obtenidos se han comprobado las ventajas e inconvenientes de la aplicación de la citada estrategia así como bajo que condiciones es más interesante su aplicación.

APÉNDICE A

GENERACIÓN DE LA SEÑAL DE RELOJ

Es este apéndice se completa el Capítulo 3 dedicado a la sincronización de circuitos VLSI. La sincronización de un sistema digital comienza con la generación de la señal de reloj, posteriormente la señal se introduce en el chip donde se acondicionada y se aumenta su frecuencia para actuar como señal de reloj. Es este apéndice se analizan las técnicas más frecuentes de acondicionamiento de la señal de reloj dentro del chip.

A.1. Generación de la señal de reloj

La señal de reloj de un sistema se genera en virtud a las propiedades piezoeléctricas de determinados cristales como el cuarzo o el tantalato de litio. El fenómeno piezoeléctrico consiste en la acumulación de carga como consecuencia de la presión mecánica aplicada, y el fenómeno contrario, la deformación del cristal cuando se le aplica una diferencia de potencial. La aplicación inicial de una diferencia de potencial inicia una vibración del cristal que genera un voltaje que se utiliza para mantener la vibración. De esta manera, el cristal se puede considerar como una representación mecánica de un circuito resonante RLC serie. La frecuencia de oscilación se regula mediante el espesor del cristal, siendo valores típicos 30 a 50MHz; a partir de aquí se requieren láminas muy delgadas pudiéndose alcanzar valores de unos pocos cientos de MHz como máximo. La principal ventaja de estos osciladores es el elevado factor de calidad (Q), que oscila entre 1.000 y 10.000 y que representa una elevada estabilidad de oscilación; la desviación de frecuencia es de tan solo unas pocas partes por millón. En cuanto al consumo del cristal, que depende de la frecuencia y de factores externos como la temperatura, se tienen valores típicos de centenares de microvatios. La Figura A.1 representa un circuito oscilador típico donde el elemento activo compensa las pérdidas resistivas [OKL03].

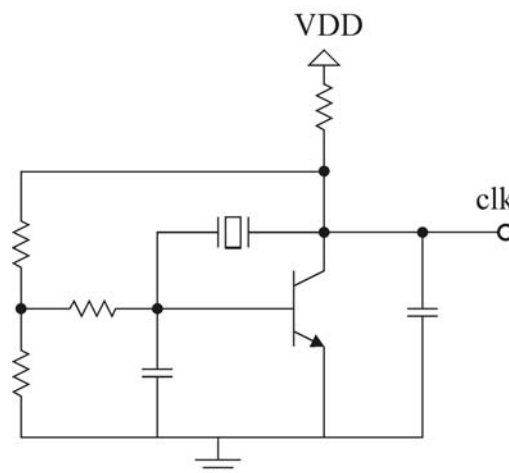


Figura A.1. Circuito oscilador de cristal.

La frecuencia obtenida directamente del oscilador de cristal es adecuada para la sincronización de los buses de la placa, sin embargo el interior del chip opera a frecuencias muy superiores y con corrientes y capacidades de carga muy inferiores. Por tanto, es necesario acondicionar la señal de reloj para adaptarla a las necesidades internas del chip. Debido a los importantes efectos capacitivos e inductivos de las líneas exteriores con respecto al interior del circuito integrado, en primer lugar la señal de reloj se introduce dentro del chip a través de una red de distribución de entrada formada por *buffers* que adaptan la señal, lo que impone un retardo apreciable denominado *insertion delay* (retardo de inserción). El alineamiento de los relojes interno y externo y la posterior multiplicación de frecuencia son llevadas a cabo por circuitos PLL. Existen dos tipos de circuitos para conseguir la sincronización de los relojes externo e interno: el PLL propiamente dicho y el DLL (*Delay-Locked Loop*: lazo de fijación por retardo), aunque, por motivos históricos, todos reciben la denominación genérica PLL.

La Figura A.2 muestra un esquema general de bloques de un PLL típico [GAR77]. El PLL está formado por un detector de fase, un filtro paso-baja y un oscilador controlado por tensión que genera la señal de reloj interna, conectados tal como se muestra en la Figura A.2. El detector de fase compara la señal de reloj de referencia (externa) con la señal interna generando como salida la operación EXOR de ambas entradas. De esta manera, el detector de fase produce pulsos tanto más anchos cuanto mayor sea el desfase entre sus entradas, y no produce salida alguna si las señales están alineadas. El tren de pulsos generado por el detector no es apto como señal de control, por lo que debe ser filtrado paso-baja para producir una señal estable antes de

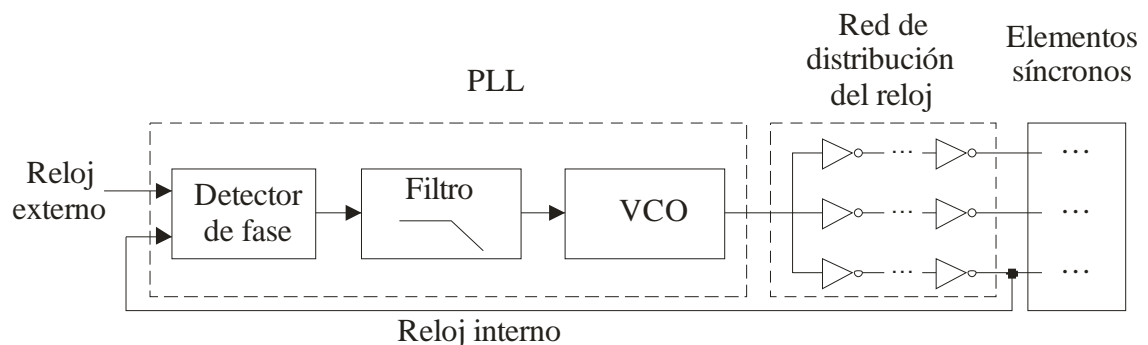


Figura A.2. Esquema general de un PLL.

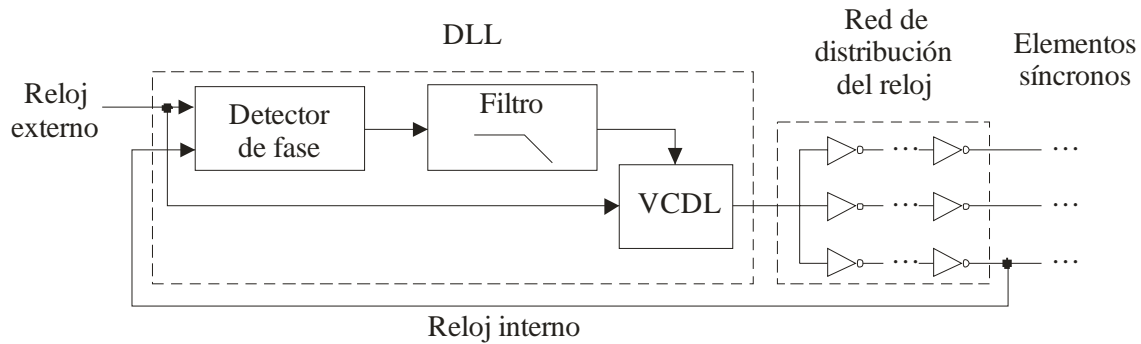


Figura A.3. Esquema general de un DLL.

acceder al VCO. El VCO genera una señal de reloj controlada por la señal de salida del filtro, que la red de distribución de la señal de reloj lleva a los elementos síncronos del sistema, desde donde es realimentada al comparador de fase. La realimentación negativa del lazo, el hecho de que las señales de reloj interna y externa sean periódicas y la condición inicial de la salida del VCO permiten la alineación de ambas señales.

El sistema basado en DLL (Figura A.3) [OKL03] tiene un funcionamiento que consiste en la alineación de las señales mediante el retardo de la señal de entrada. De manera similar al PLL, la señal externa es comparada con la señal aplicada al sistema, el resultado de la comparación es filtrado paso baja y la señal de control se aplica sobre el módulo VCDL (*Voltage Control Delay Line*: línea de retardo controlada por tensión), cuya finalidad es retardar la señal de entrada, que es directamente la señal de referencia externa. La periodicidad de las señales de reloj interna y externa asegura la alineación de las fases. La principal diferencia entre el PLL y el DLL es que mientras que en el primero se produce una modificación de la frecuencia, en el segundo se altera la fase.

El sistema basado en PLL permite obtener altas frecuencias de reloj, generándose éste en el interior del propio chip. Sin embargo, la existencia de circuitos analógicos integrados dentro de un sistema digital genera altos niveles de ruido. Como alternativa a los sistemas PLL convencionales se han desarrollado sistemas ADPLL (*All Digital Phase-Locked Loop*: ciclo de fijación de fase completamente digital) [DUN95, KIM97, CHI99]. Los sistemas ADPLL presentan varias ventajas, ya que eliminan el ruido asociado a los dispositivos analógicos, se obtiene un rápido alcance de la frecuencia de referencia (en unos 50 ciclos de reloj), tienen buena estabilidad y, como

ya se ha mencionado, llevan a una completa digitalización del circuito. El núcleo del sistema ADPLL es el oscilador controlado digitalmente (DCO: *Digital Controlled Oscillator*). Generalmente se usan tres tipos de DCO [CHI99]:

1. *path delay oscillator* (oscilador con ruta de retardo), formado por un anillo oscilador compuesto por una cascada de puertas lógicas. Su frecuencia de funcionamiento es limitada y no es adecuado para sistemas de alta velocidad.
2. *Schmitt-trigger based current-driven oscillator* (oscilador controlado por corriente basado en el disparador de Schmitt), que necesita un inversor *Schmitt-trigger*, una gran capacidad (del orden de cientos de pF) y varios transistores MOS de control para completar la función de oscilación. Debido a la gran capacidad que se requiere, ésta se suele conectar externamente, lo que provoca la disminución del rendimiento.
3. *current-starved ring oscillator* (oscilador en anillo con bajo nivel de corriente, Figura A.4), que debido a su buena linealidad se suele usar en la mayoría de los sistemas con microprocesador. Las diferentes frecuencias se controlan mediante conmutadores MOS. Sin embargo, este sistema es muy costoso y requiere un área apreciable.

En el modo de funcionamiento de control mediante tensión, puede producirse una pérdida de prestaciones: limitación en el rango de selección de frecuencia del VCO, disminución del rango dinámico lineal del detector de fase o una disminución de la máxima amplitud de oscilación. Estas limitaciones se acentúan en el caso de aplicaciones de baja potencia. Una alternativa es el funcionamiento controlado por corriente [TOU66], con el que se puede, en determinadas circunstancias, conseguir un aumento del rango dinámico o del ancho de banda para baja potencia, todo ello gracias a la disminución de las oscilaciones de tensión.

Chiang *et al.* [CHI99] diseñaron un ADPLL con sistema *current-starved ring oscillator* que reduce considerablemente el *hardware* necesario con un proceso CMOS SPDM (*Single Poli Double Metal*: una capa de polisilicio y dos capas de metal) de

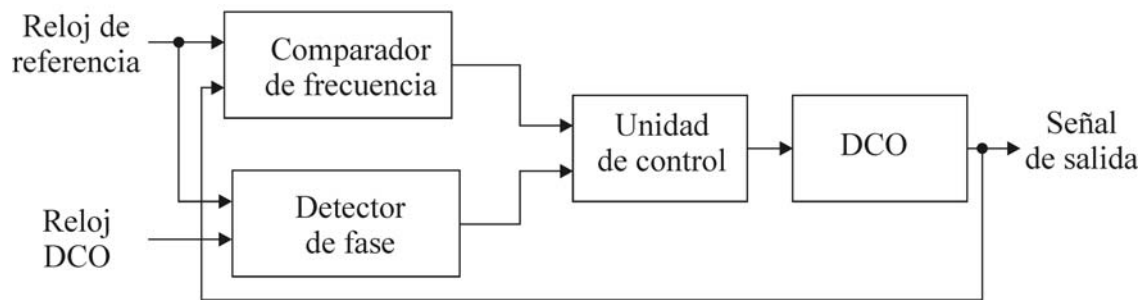


Figura A.4. Diagrama de bloques de un sistema ADPLL *current-starved ring oscillator*.

0.6 μ m (Figura A.4). Este sistema usa un DCO que ocupa aproximadamente la mitad de área que uno convencional y es capaz de operar entre 60 y 400MHz a cuatro veces la frecuencia del reloj de referencia; el tiempo de estabilización de la señal es de 47 ciclos y el error de fase es inferior a 0.1ns. El proceso seguido para la generación de la señal de reloj comienza con la adquisición de frecuencia; se adquiere posteriormente la fase, y por último el sistema evoluciona de manera que se mantienen ambos parámetros, frecuencia y fase.

La segunda aplicación principal de los PLL es como multiplicadores de frecuencia [OKL03]. El esquema básico de un multiplicador se tiene en la Figura A.5, donde la frecuencia de oscilación interna es:

$$f_{\text{int}} = \frac{1}{B} f_{\text{VCO}} = \frac{C}{A} f_{\text{ext}} \quad (\text{A.1})$$

Típicamente se emplean los valores $A=1$ y $B=2$, siendo C el factor por el que se multiplica la frecuencia externa y que, típicamente, adquiere valores enteros potencia de dos [YOU92]. Otros valores para A , B y C se emplean para conseguir reducir el ruido durante la fase de encendido en chips con gran número de dispositivos [VON98].

La decisión de emplear un sistema basado en PLL o en DLL depende en parte del origen del ruido dominante en el sistema [OKL03]. El PLL filtra paso-baja el ruido de la señal de referencia, no siendo así en el caso del DLL, mientras que el ruido generado en el VCO del PLL es realimentado por el lazo, acumulándose por encima del ancho de banda de rechazo del lazo de retroalimentación, por lo que dicho ruido es de alta frecuencia. Así pues, si la fuente de ruido dominante proviene de la señal de

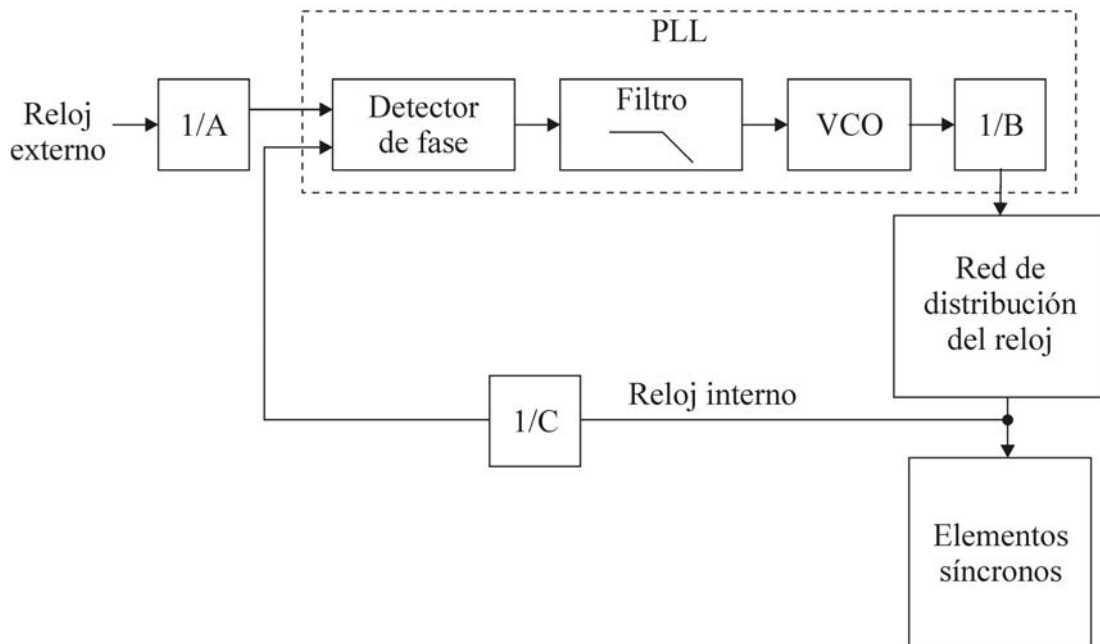


Figura A.5. PLL como multiplicador de frecuencia.

referencia interesa emplear el PLL, mientras que si el ruido se genera en el VCO o VCLD la alternativa DLL es la más interesante. Por otro lado, los PLL son más complicados de diseñar por cuestiones de estabilidad (sistema de segundo orden); sin embargo ofrecen un mayor rango de captura de frecuencia y posibilitan la multiplicación de frecuencia, como se ha visto. Los DLL son más estables (sistema de primer orden) y por lo tanto más sencillos de diseñar, pero tienen menor rango de fijación de frecuencia. Es posible diseñar DLLs con prestaciones similares a los PLLs pero a costa de aumentar la complejidad del sistema. Los PLLs se suelen emplear como multiplicadores de frecuencia y para eliminar el ruido de alta frecuencia en el reloj de referencia, mientras que los DLLs se emplean para la alineación de fase entre señales, tanto externa respecto interna como dentro del chip.

Como se ya se ha visto, el VCO es el generador de la señal de reloj dentro del chip. Fundamentalmente existen dos alternativas para el diseño del VCO (Figura A.6): el anillo oscilador y el sistema basado en oscilador LC (LC VCO). La primera alternativa ha sido la más empleada tradicionalmente debido a que las inductancias integradas en tecnología bipolar presentaban un bajo factor de calidad y además eran excesivamente grandes, ya que el tamaño de una inductancia es inversamente proporcional al cuadrado de la frecuencia. La tecnología CMOS supuso la aparición de

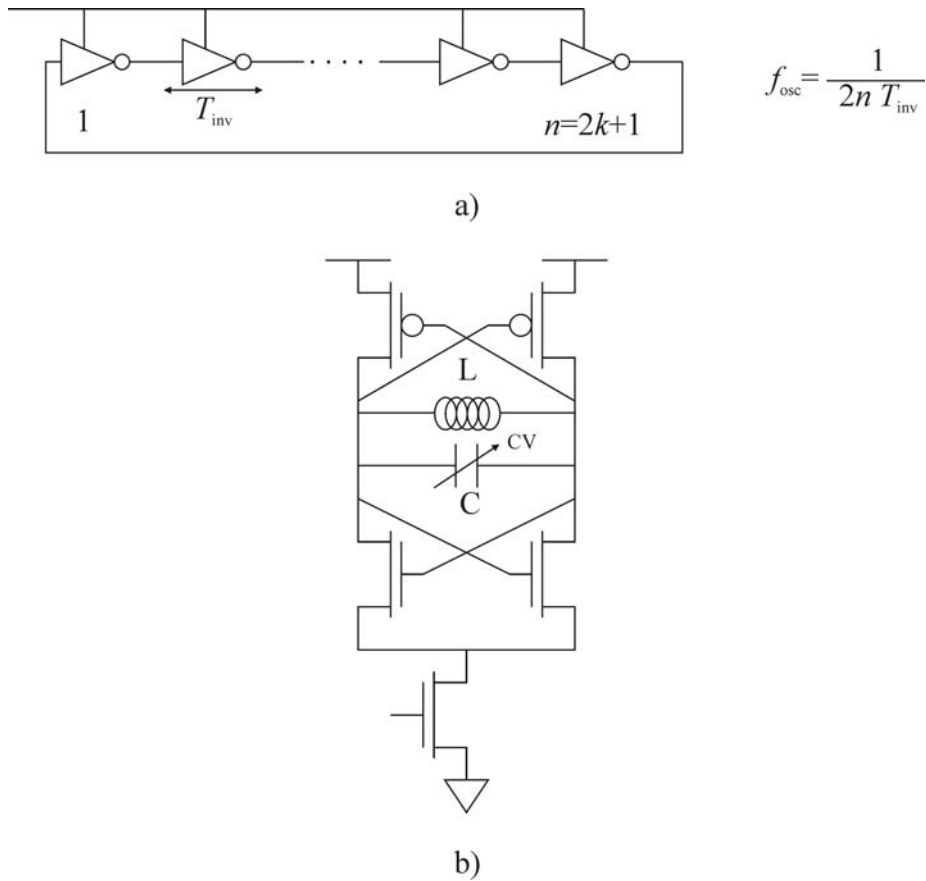


Figura A.6. Oscilador basado en anillo (a) y en oscilador LC (b).

dispositivos con alta frecuencia de conmutación, muy buena eficacia energética y una alta capacidad de escalado. Estos factores posibilitan la implementación de circuitos de alta frecuencia de funcionamiento, con lo que el tamaño de las inductancias requeridas era cada vez menor. Además, la tecnología CMOS en altos niveles de escalado se vuelve muy sensible a las variaciones de proceso, voltaje y temperatura (PVT), por lo que la alternativa del oscilador LC es más interesante en tanto que presenta una mejor estabilidad de fase y tiene un alto nivel de rechazo al ruido de alimentación, por lo que genera señales menos ruidosas.

Las inductancias implementadas en el chip dependen básicamente de la finalidad del oscilador y de la tecnología disponible. En los LC VCOs se emplean inductancias de alto factor de calidad en torno a la frecuencia de oscilación del mismo para minimizar el ruido de fase en la señal generada. La determinación del factor de calidad se puede realizar mediante modelos analíticos [YUE98], aunque lo más frecuente es realizar mediciones directas del *hardware*. Otro factor importante es el valor de L ya que debe

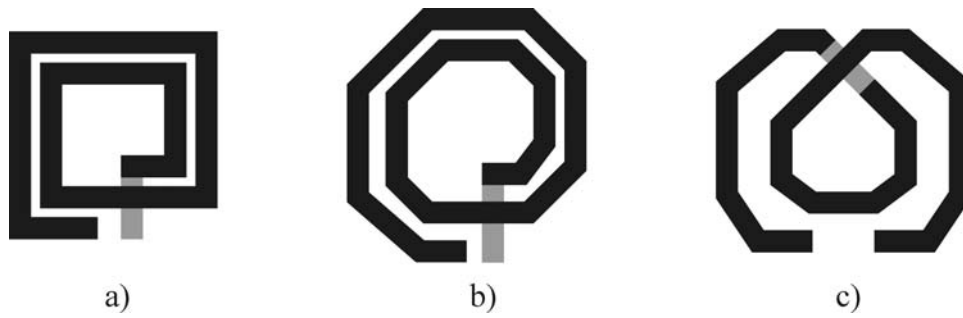


Figura A.7. Inductores espirales de dos vueltas: a) cuadrado, b) octogonal y c) octogonal simétrico.

estar dentro del rango de las frecuencias a las que oscila el VCO. Para su implementación se suelen usar capas de metal de baja resistencia y es frecuente interconectar capas de varios niveles. El cálculo de L se puede realizar, en primera aproximación, mediante el método de Greenhouse [GRE74], aunque existen modelos analíticos [MOH99, JEN02] basados en la geometría de la bobina: forma, vueltas, radio, anchura y grosor del metal y espaciado entre líneas. Otras alternativas para el cálculo con mayor exactitud involucran el uso de herramientas de análisis electromagnético o la creación de modelos basados en medidas empíricas previas. En todo caso se debe mantener un equilibrio rentable entre exactitud y tiempo de análisis de los diferentes inductores en un sistema. La Figura A.7 muestra algunas de las bobinas integradas más frecuentes.

APÉNDICE B

ANÁLISIS DE LOS DISPOSITIVOS Y MÓDULOS USADOS EN LAS SIMULACIONES

B.1. Transistores

Como elementos básicos de implementación de todos los demás módulos se han empleado transistores MOSFET (*Metal-Oxide-Semiconductor Field Effect Transistor*, Transistor de Efecto Campo Metal Óxido Semiconductor), nMOS y pMOS descritos mediante un modelo de librería *level 3* de PSpice. Las dimensiones del canal de dichos transistores, es decir, su anchura y longitud, se pueden controlar mediante los parámetros W y L , habiéndose considerado una dimensión típica de $0.6\mu\text{m}$ correspondiente al proceso CMOS14TB, que es una adaptación del proceso HP-AMOS14TB de Hewlet-Packard para CMC (*Canadian Microelectronics Corporation*) a través del servicio MOSIS.

Los modelos de PSpice de los transistores utilizados para las diferentes simulaciones realizadas son los siguientes:

```
.MODEL CMOSN NMOS
+ PHI=0.70000 TOX=9.6000E-09 XJ=0.200000U TPG=1
+ VTO=0.6566 DELTA=6.9100E-01 LD=4.7290E-08 KP=1.9647E-04
+ UO=546.2 THETA=2.6840E-01 RSH=3.5120E+01 GAMMA=0.5976
+ NSUB=1.3920E+17 NFS=5.9090E+11 VMAX=2.0080E+05
  ETA=3.7180E-02
+ KAPPA=2.8980E-02 CGDO=3.0515E-10 CGSO=3.0515E-10
+ CGBO=4.0239E-10 CJ=5.62E-04 MJ=0.559 CJSW=5.00E-11
+ MJSW=0.521 PB=0.99
+ XCQ=0.5 XPART=0.5

.MODEL CMOSP PMOS
+ PHI=0.70000 TOX=9.6000E-09 XJ=0.200000U TPG=-1
+ VTO=-0.9213 DELTA=2.8750E-01 LD=3.5070E-08 KP=4.8740E-05
+ UO=135.5 THETA=1.8070E-01 RSH=1.1000E-01 GAMMA=0.4673
+ NSUB=8.5120E+16 NFS=6.5000E+11 VMAX=2.5420E+05
  ETA=2.4500E-02
+ KAPPA=7.9580E-02 CGDO=2.3922E-10 CGSO=2.3922E-10
+ CGBO=3.7579E-10 CJ=9.35E-04 MJ=0.468 CJSW=2.89E-10
+ MJSW=0.505 PB=0.99
+ XCQ=0.5 XPART=0.5
```

Todos los sistemas que se han diseñado utilizan tecnología CMOS, caracterizada por su alta eficacia energética, ya que estos sistemas tan sólo consumen energía en los cambios de estado o por corrientes de fugas en los estados estables. El consumo dinámico ocurre durante los breves lapsos de tiempo de las transiciones de estado, cuando ambos transistores se encuentran en conducción. Debido a que la movilidad de los electrones es superior a la de los huecos, la cantidad de corriente que circula por los transistores NMOS es superior a la que circula por los PMOS en igualdad de condiciones. Es necesario establecer diferentes dimensiones de canal en ambos tipos de transistor, de modo que se igualen las conductividades. Según el modelo de transistor que se está empleando, las movilidades de electrones y huecos están descritas por el parámetro U_0 y son:

$$\mu_{0e} = 546.2 \text{ cm}^2/\text{V}\cdot\text{s} \quad \mu_{0h} = 135.5 \text{ cm}^2/\text{V}\cdot\text{s}$$

Estas movilidades son sensiblemente inferiores a los valores típicos de movilidad de portadores de carga en semiconductores. Esto se debe a que estos portadores de carga están moviéndose junto a una interfaz de óxido y, además, están confinados en un pozo de potencial, de modo que su movilidad se ve reducida respecto al portador de carga que se mueve libremente en un semiconductor ilimitado en un espacio tridimensional. La relación entre movilidades permite establecer la proporción entre las anchuras del canal para conseguir la igualdad de corrientes. En este caso el cociente resulta ser:

$$\frac{\mu_{0e}}{\mu_{0p}} = 4.03 \quad (\text{B.1})$$

Por lo tanto se toma la razón entre dimensiones igual a cuatro con lo que los transistores empleados por defecto tendrán las dimensiones según indica la Tabla B.1.

	<i>L</i>	<i>W</i>
<i>NMOS</i>	0.6 μm	1.6 μm
<i>PMOS</i>	0.6 μm	6.4 μm

Tabla B.1. Dimensiones de los transistores NMOS y PMOS.

En determinadas circunstancias estas dimensiones han sido modificadas para adaptarse mejor a los requerimientos de retardo, capacidad de entrada o capacidad de conducción de corriente.

B.2. Elementos de memoria

B.2.1 Biestables

El elemento de memoria básico utilizado es un biestable tipo D, basado en TSPC [KAR88, AFG90], activado por flanco de bajada, que recibe la denominación nETDFF (*Negative Edge-Triggered D flip-flop*), y cuyo esquema muestra la Figura B.1. La mayoría de los transistores que constituyen en biestable son de dimensiones expresadas en la Tabla B.1, sin embargo, las dimensiones de algunos de ellos se han modificadas como consecuencia de las necesidades de conmutación del biestable. Una simulación detallada del funcionamiento del biestable muestra que el retardo asociado al mismo es del orden de 0.35ns, que es un valor adecuado para las frecuencias de funcionamiento que se han aplicado.

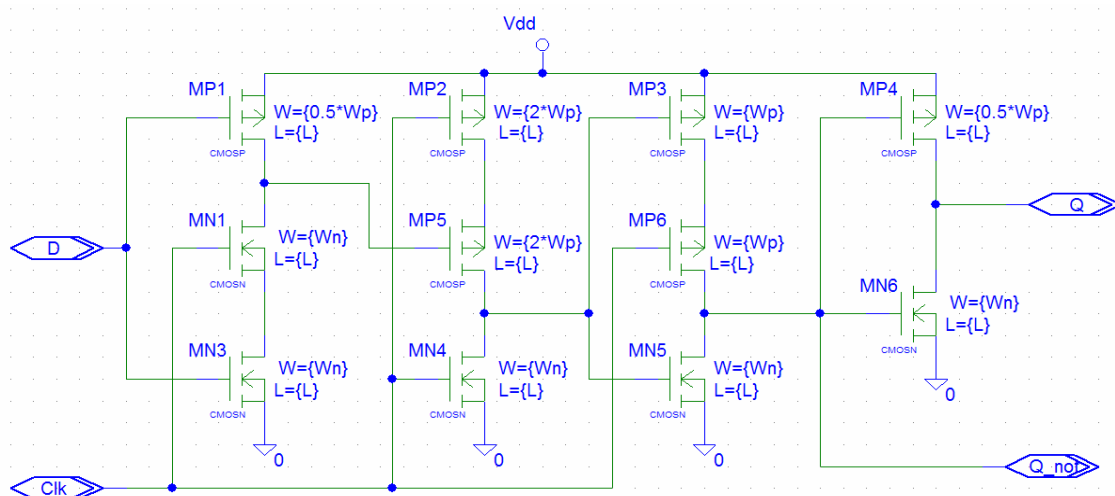


Figura B.1. Esquemático de un biestable tipo D activado por flanco de bajada.

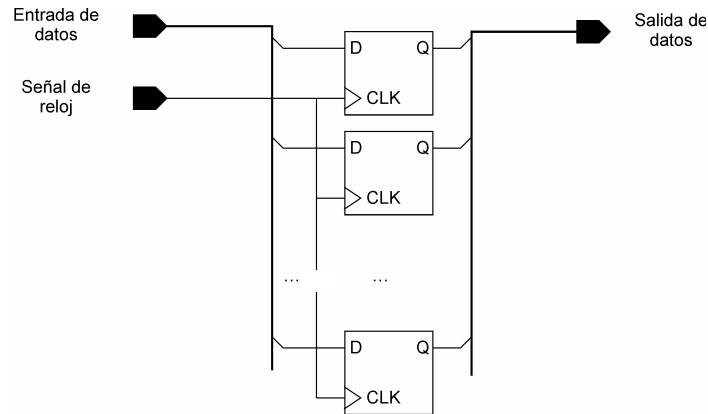


Figura B.2. Implementación de un registro de n bits.

B.2.2 Registros

En las simulaciones se han empleado registros de diferente número de bits y cada registro está formado por la asociación en paralelo del número adecuado de biestables con señal de reloj común tal como muestra la Figura B.2.

B.2.3 Memorias ROM

Como se ha visto a lo largo del Capítulo 4, un componente importante de los sistemas son las tablas de consulta (LUTs). Dichas tablas se usan como multiplicadores, como parte de los módulos de conversión de binario a RNS y viceversa, o para realizar la transformación de un coeficiente al índice correspondiente y su inversa, en el sistema para la realización de la transformada 1D-DWT. El fundamento de las LUTs consiste en asociar a cada posición de memoria el valor correspondiente a la transformación del dato correspondiente a dicha dirección. De ese modo, cuando en el bus de direcciones de la entrada se introduce un dato, la salida de la tabla de consulta es en valor modificado de dicho dato. Entre las diversas alternativas [UYE92] existentes para la implementación de memorias ROM se ha optado por las memoria desarrollada por el VLSI *Research Group* de la Universidad de Windsor, ya que son memorias de alta velocidad y reducido consumo. La Figura B.3 muestra, a modo de ejemplo, el esquema de una memoria de 256 palabras de 4 bits y en las Figuras B.4 hasta B.7 se detallan los subcircuitos que forman el bloque de memoria. La Figura B.4 representa un módulo de selección de filas (`dec_64`), la Figura B.5 una celda de memoria donde se almacena el

contenido de la memoria (*cell_2x8*), la Figura B.6 muestra el selector de columnas (*sel_2_1*) y por último en la Figura B.7 se puede apreciar la celda final amplificadora que acondiciona los datos de salida del bloque de memoria (*amp_inv*).

Tal como se puede apreciar en la Figura B.3, el bus de direcciones se divide en dos partes: los seis bits más significativos junto con sus complementarios son dirigidos a los selectores de fila, mientras que los bits restantes seleccionan las columnas para obtener el dato de salida. El menos significativo de los bits de selección de filas y su complemento acceden a todas las celdas *dec_64* mientras que, a cada celda, sólo accede una de las 32 posibles combinaciones de los cinco bits restantes y sus complementos tal y como se puede apreciar en la Figura B.4. De este modo, cualquier combinación de estos bits activará solamente una de las 64 filas que se pueden codificar con los seis bits. La celda *dec_64*. Está formada por 8 transistores NMOS de

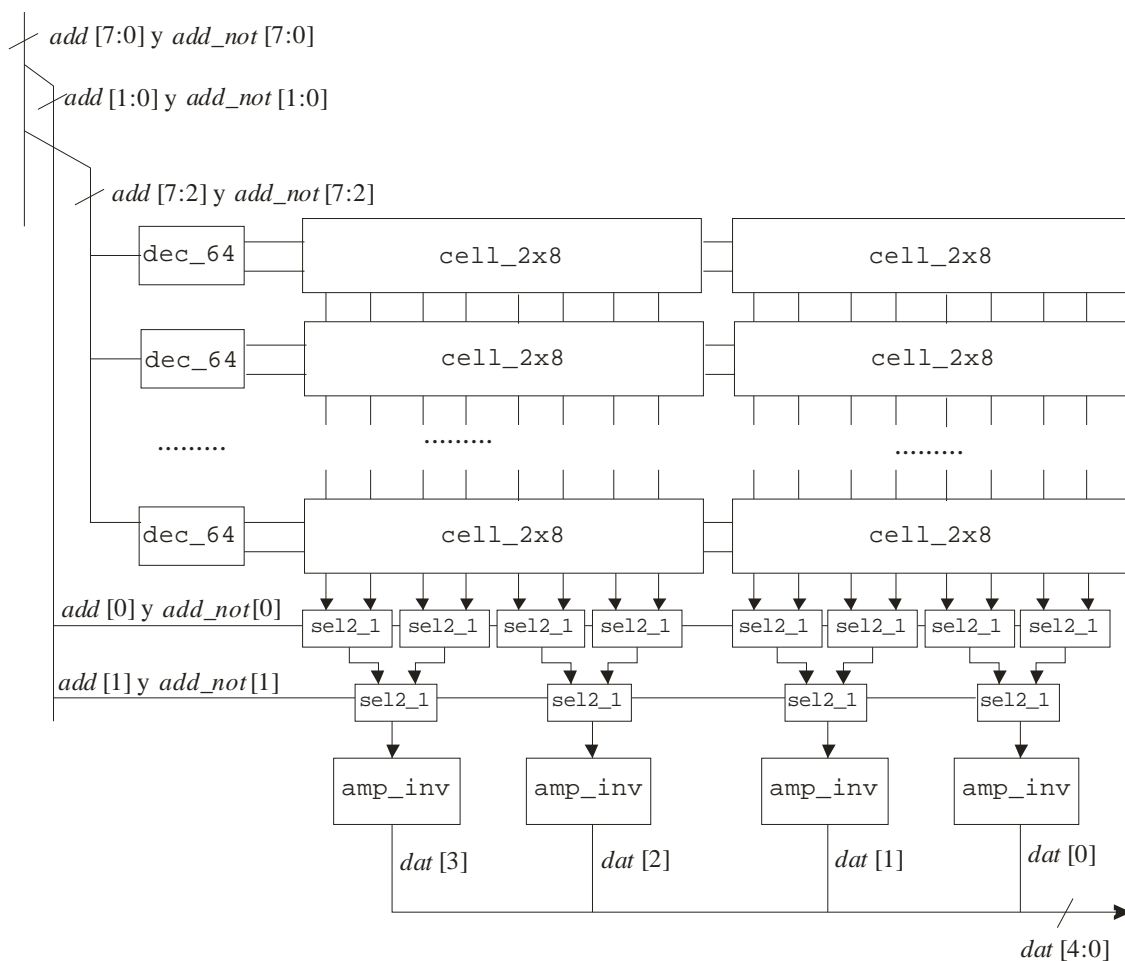


Figura B.3. Visión de global de un módulo de memoria de 256x4.

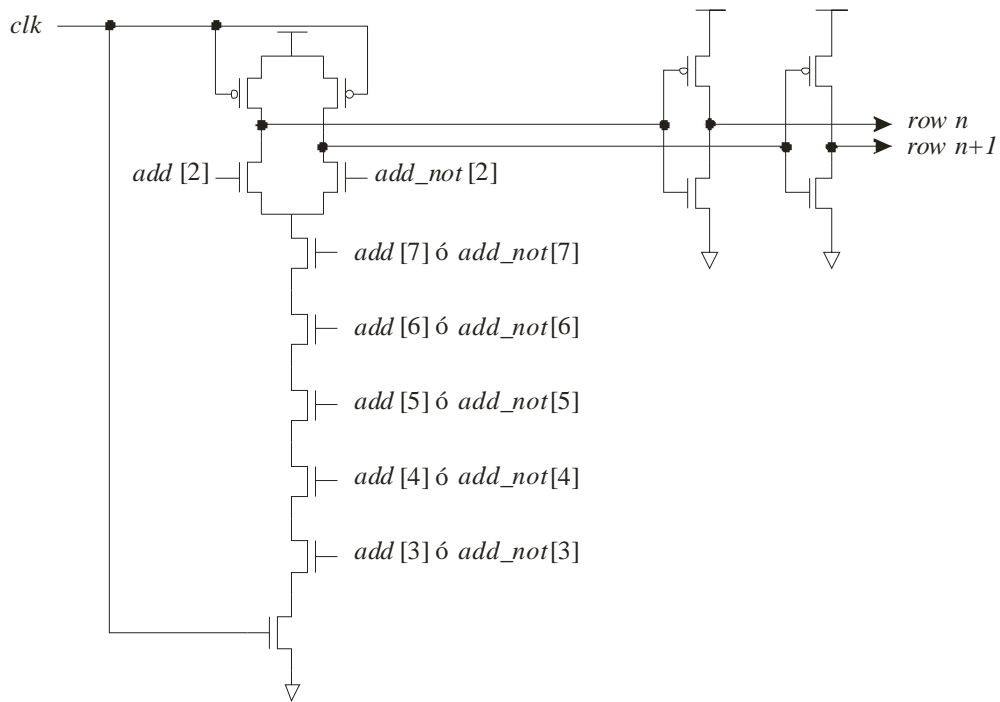


Figura B.4. Celda de selección de filas `dec_64`.

dimensiones $0.6\mu\text{m}\times 1.6\mu\text{m}$ y dos PMOS de $0.6\mu\text{m}\times 2.0\mu\text{m}$ para la precarga de los nodos internos y dos inversores que generan las correspondientes señales de selección de filas, formados por transistores NMOS de $0.6\mu\text{m}\times 1.6\mu\text{m}$ y PMOS de $0.6\mu\text{m}\times 6.4\mu\text{m}$. Cuando la señal de reloj permanece en baja los nodos de entrada a los inversores se precargan a 1 hasta que el reloj pasa a valor alto. Entonces se producirá la descarga a GND del nodo en aquella celda en la que todos los transistores NMOS conduzcan, con lo que la correspondiente línea de selección de filas inyectará un 1, permaneciendo el resto de las líneas a cero. De este modo cada combinación de los seis bits más significativos activa una única fila de las celdas `cell_2x8`.

La Figura B.5 muestra la estructura de la celda de memoria `cell_2x8`. Está formada por 16 transistores NMOS de $0.6\mu\text{m}\times 1.6\mu\text{m}$ con fuente común a GND y drenadores conectados o no, a la línea de datos vertical. Durante la fase inactiva del reloj las líneas de datos seleccionadas por los selectores de columnas (`sel_2_1`) se precargan a 1 desde el amplificador inversor de la etapa final. Cuando la señal de reloj pasa a nivel alto, la línea de datos vertical se descargará a través del transistor activado a través de la línea de selección de filas si el drenador de este está conectado a dicha línea de datos. En este caso, la salida del inversor final se establece en alta generándose un 1

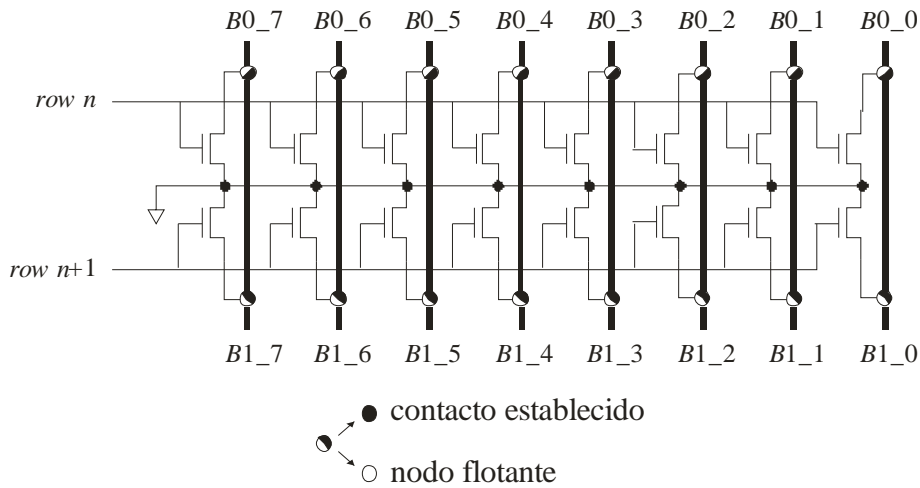


Figura B.5. Celda de almacenamiento de datos *cell_2x8*.

lógico. En el caso que el drenador no esté conectado a la línea de datos, ésta no se descarga y la salida se convertirá en un 0 lógico. Este es el mecanismo de la memoria para el almacenamiento de datos; la conexión del drenador a la línea de datos para el valor '1' y la no conexión, dejando el terminal flotante, para el valor '0'. Las celdas *cell_2x8* se pueden conectar fácilmente en paralelo a las líneas de selección de filas si se requieren memorias de mayor tamaño.

En la Figura B.6 se encuentra el selector de columnas, formado por transistores NMOS de dimensiones $0.6\mu\text{m}\times 3.6\mu\text{m}$. Es una celda de tamaño reducido sobre la que actúan a los bits menos significativos (y sus complementarios) del dato de entrada. Sobre cada pareja de líneas seleccionan una de las dos en los niveles que sean

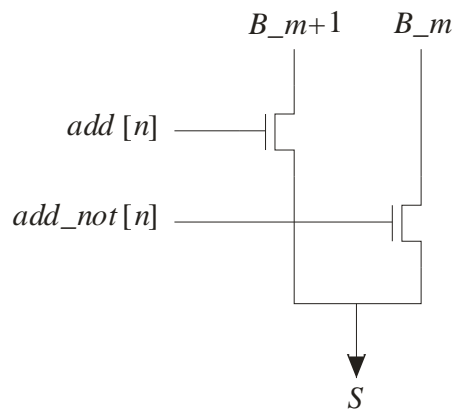


Figura B.6. Esquemático de la celda *sel2_1*.

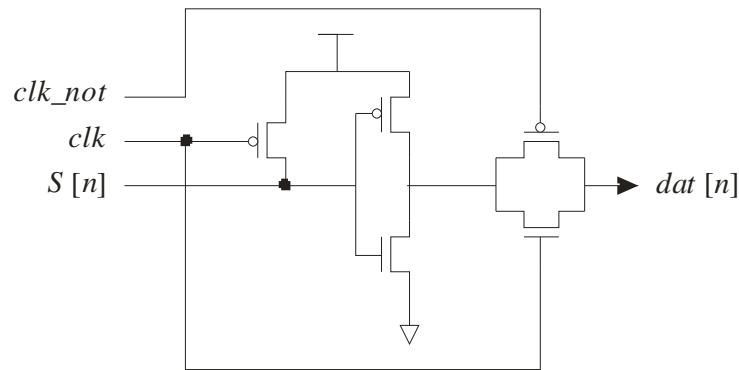


Figura B.7. Esquemático de la celda `amp_inv`.

necesarios para reducir las líneas de datos a la anchura final del dato de salida. En muchos de los módulos de memoria usados en esta tesis el dato de entrada es de 6 bits, por lo que no son necesarias estas celdas al quedar el dato de salida perfectamente definido por la selección de filas.

La salida final de datos se realiza a través de la celda `amp_inv` (Figura B.7) que como ya se ha explicado precarga a '1' el nodo $S[n]$ durante la fase negativa del reloj al tiempo que la puerta de paso mantiene el '0' generado en el inversor aislado de la salida. Durante la fase activa del reloj la línea de datos se descargará o no dependiendo de si el transistor seleccionado posee o no contacto a tierra en la máscara de programación de la memoria, y generará un '1' o un '0' respectivamente que será transmitido por la puerta de paso durante esta fase del reloj.

Resumiendo todo lo anterior, el funcionamiento de la memoria consiste en la precarga de los nodos y el mantenimiento de la salida anterior durante la fase negativa de la señal de reloj. En el flanco de subida se evalúa la fila seleccionada mediante los seis bits más significativos de la dirección de memoria a través de los módulos `dec_64`. La fila seleccionada del módulo `cell_2x8` puede establecer una conexión a tierra o mantener flotante una línea precargada por el módulo `amp_inv` en función de la programación de la celda `cell_2x8`. Si las direcciones de memoria son de más de seis bits se seleccionan, con los bits menos significativos mediante las celdas `sel2_1`, las columnas correspondientes al dato de salida y se transfiere el dato correspondiente hacia la salida, constituida por las celdas `amp_inv`, donde se mantiene hasta el siguiente flanco de subida de la señal de reloj.

La implementación de las memorias se ha realizado mediante librerías que describen la estructura y programación de las memorias a través de líneas de texto, como alternativa a de su realización mediante esquemáticos. Los motivos principales son el tamaño de las memorias así como la compleja programación de las celdas `cell_2x8`, especialmente en los casos de datos de entrada de más de seis bits. Una vez generada la librería correspondiente al módulo de memoria, se instancia desde un módulo esquemático mediante un parámetro que es el nombre de la memoria en cuestión. La generación de las librerías se realiza de modo automático mediante un programa desarrollado para este trabajo, a partir de un archivo de texto con el contenido de la memoria, y las longitudes del bus de direcciones y el de datos. El Apéndice C contiene una descripción de dicho programa.

B.3. Sumadores

B.3.1 Sumador binario de n bits

La suma modular, como operación fundamental en todos los sistemas que se han simulado, se implementa a partir de sumadores completos binarios. La Figura B.8

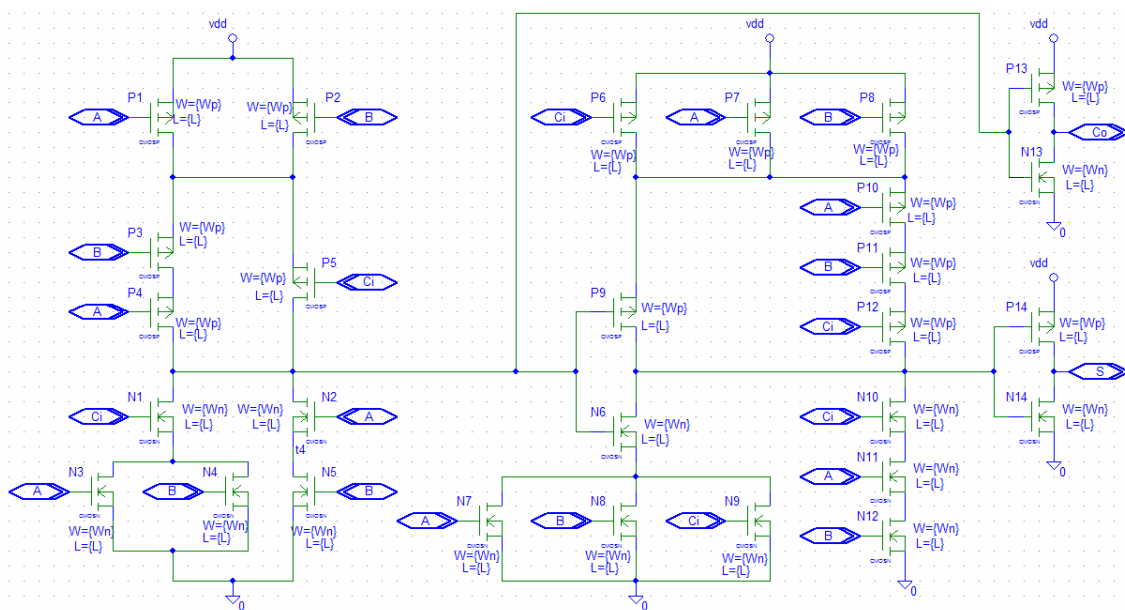


Figura B.8. Sumador completo de un bit.

muestra el esquemático que se ha empleado como sumador completo de un bit. La asociación en paralelo de n celdas como la mostrada en la Figura B.8, con las salidas de acarreo de salida de cada bit conectada a la entrada de acarreo del siguiente permite implementar sumadores binarios de n bits. Mediante simulación se ha comprobado cómo el retardo máximo de un sumador de un bit es 0.37ns, por lo que en el sumador de 10 bits, que es el mayor que se ha utilizado, tendría un retardo máximo de 3.7ns. Este valor, aunque significativo, es aceptable para todas las frecuencias simuladas.

B.3.2 Sumador módulo m

El sumador módulo m se puede obtener a partir del sumador binario tal como se comentó en el Capítulo 4. La Figura B.9 contiene el sumador módulo m utilizado en las simulaciones. Inicialmente se realiza la suma de los datos de entrada y posteriormente al resultado se le resta el módulo m mediante la suma del complemento a 2^k ($2^k - m$). Si en alguna de las dos operaciones se produce un acarreo se selecciona la segunda suma

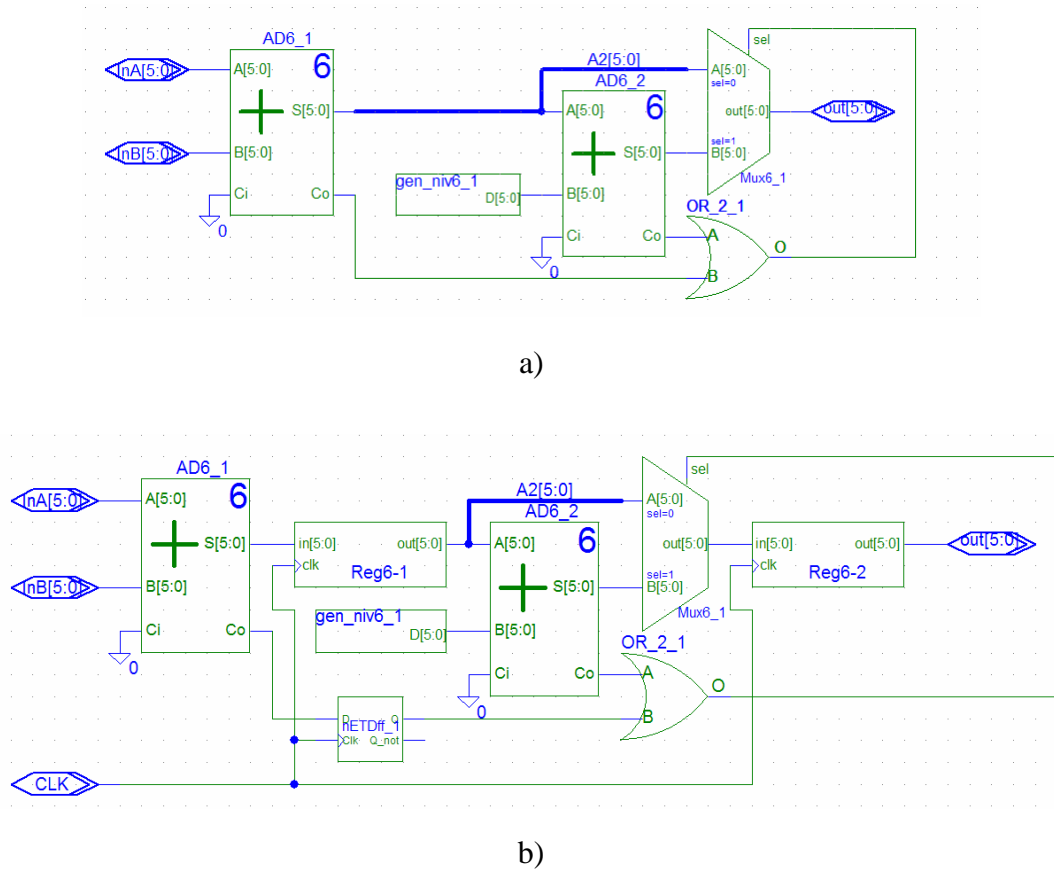


Figura B.9. Sumadores modulares simple (a) y con *pipelinig* (b).

realizada [BAY87] según la expresión para la suma modular:

$$(x + y)_{\text{mod } m} = \begin{cases} x + y & \text{si } x + y < m \\ (x + y) - m & \text{si } x + y \geq m \end{cases} \quad (\text{B.2})$$

Dentro del módulo `gen_nivk` se genera el valor de $(2^k - m)$ mediante un parámetro pasado desde la instancia del sumador. Existen dos versiones del sumador: la versión sencilla Figura B.9(a) y otra que implementa la técnica del *pipelining* que cuenta con un registro a la salida para el mantenimiento del dato generado durante un ciclo de reloj Figura B.9(b). Ambas versiones responden a las necesidades de latencia o velocidad de los diseños. El modelo sencillo se ha empleado en las simulaciones del controlador PID y del sistema para realizar la transformada 1D-DWT mientras que en el filtro CIC de Hoguenauer se ha empleado la versión con *pipelining*.

B.4. Otros módulos

Bajo el presente epígrafe se describen los restantes elementos que se han empleado en las simulaciones. Todos los elementos relativos a la distribución y acondicionamiento

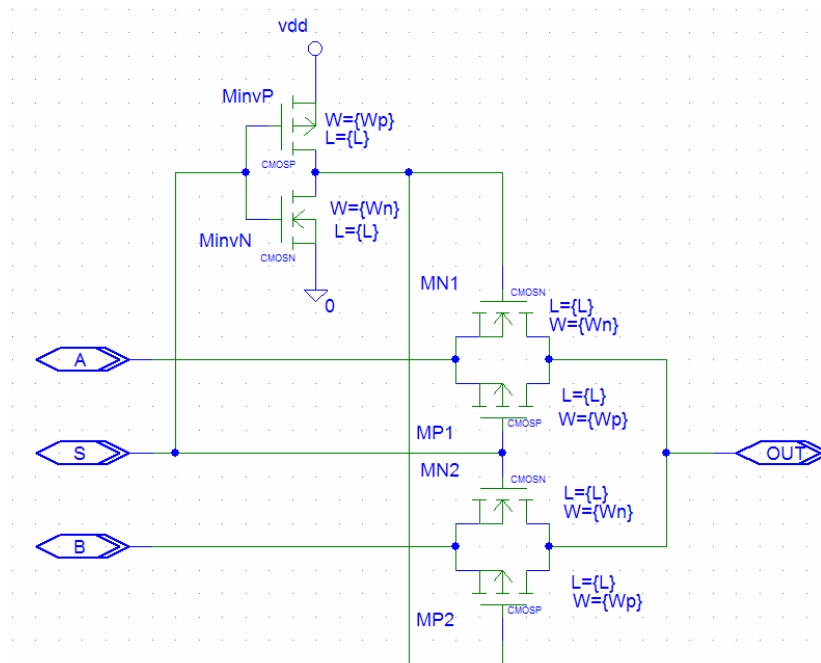


Figura B.10. Celda básica del multiplexor.

de la señal de reloj se han descrito suficientemente en el capítulo 4 y no se van a repetir aquí.

B.4.1 Multiplexores

Los multiplexores se han implementado mediante puertas de paso tal como se muestra en la Figura B.10. Los transistores empleados son de dimensiones típicas: $0.6\mu\text{m}\times 1.6\mu\text{m}$ para los transistores nMOS y $0.6\mu\text{m}\times 6.4\mu\text{m}$ para los transistores pMOS. La asociación de n celdas con $2n$ entradas, n salidas y señal de selección (S) común permite implementar multiplexores de $2n$ a n .

B.4.2 Puertas lógicas

Las simulaciones realizadas han requerido la utilización de puertas lógicas, concretamente puertas AND, OR y NAND (de 2, 3, 4 y 5 entradas) e inversores. En todos los casos se han empleado las dimensiones típicas contenidas en la Tabla B.1 puesto que el acondicionamiento de las señales, cuando ha sido necesario, se ha realizado mediante los *buffers* ampliamente explicados en el Capítulo 4.

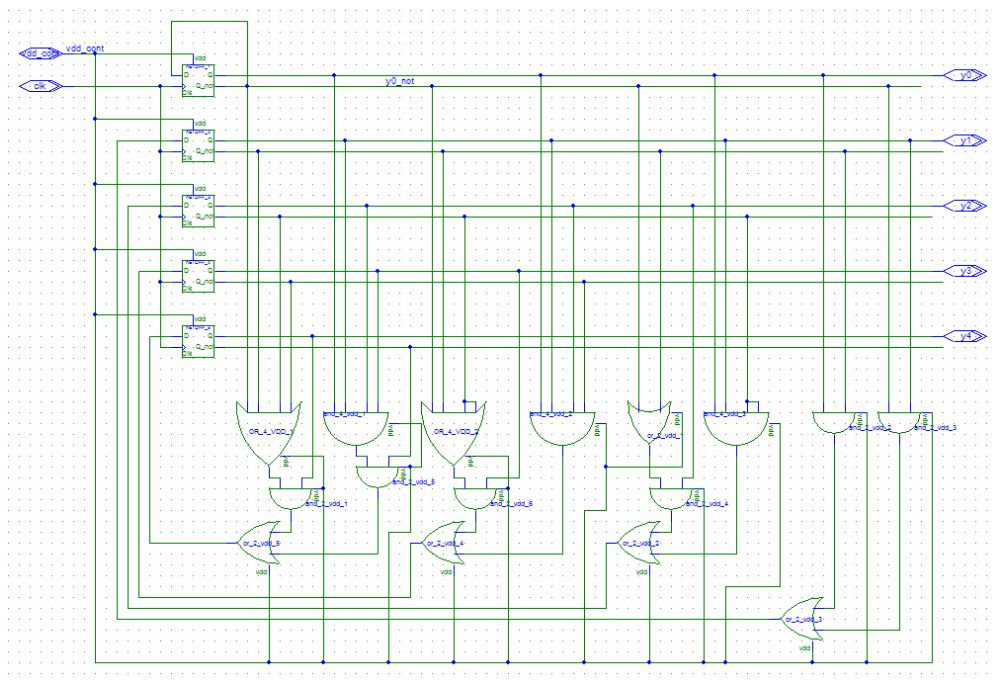


Figura B.11. Contador de 32 bits.

B.4.3 Contador de 32 bits

En los sistemas simulados ha sido necesaria la introducción de un contador de 32 bits que se ha empleado en reductor de frecuencia empleado en el filtro CIC, y en todos los módulos generadores de patrones de entrada que se han empleado en todas las simulaciones. La Figura B.11 muestra el contador empleado que está constituido por cinco biestables tipo nETDFF ya explicados y la lógica que realimenta cada uno de los biestables y genera la cuenta.

B.5. Generación de la señal de reloj externa

La señal de reloj de entrada se simula mediante un fuente de tensión ideal que genera una secuencia de pulsos como el representado en la figura B.12. En las simulaciones se aplicaron señales de reloj de frecuencias 40MHz, 64MHz, 100MHz y 125MHz cuyas características se indican en la tabla B.2.

	Tiempo en alta (ns)	t_r y t_f (ns)	T (ns)
40 MHz	12.3	0.100	25.0
64 MHz	7.69	0.063	15.6
100 MHz	4.92	0.040	10.0
125 MHz	3.94	0.032	8.0

Tabla B.2. Especificaciones para las dos señales de reloj ideales aplicadas.

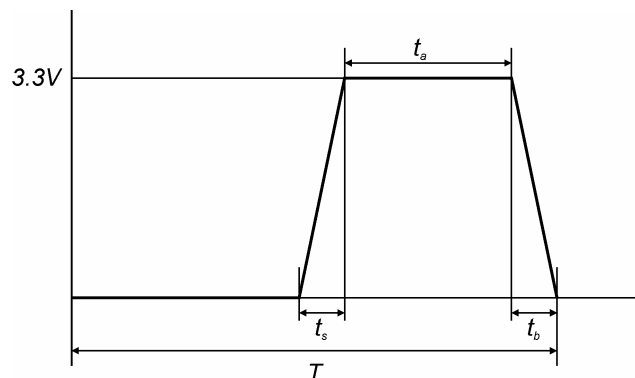


Figura B.12. Señal de reloj de entrada.

En el diseño de las señales de reloj de entrada se ha tenido en cuenta el escalado del periodo de dichas señales a efectos de los parámetros más significativos que son los tiempos de subida y bajada. El periodo de la señal de reloj afecta a las simulaciones en cuanto a los tiempos en los que se producen los flancos y, como ya se explicó en el Capítulo 4, al emplearse señales de reloj levemente desfasadas se produce un solapamiento en la conmutación de los dispositivos al aumentar la frecuencia. Sin embargo otro efecto que se ha querido modelar es la variación en los tiempos de subida y bajada, que tiene un efecto directo sobre lo abrupta que es la transición en señales de reloj cada de cada vez mayor frecuencia. En la Tabla B.2 se han ajustado todos los parámetros de la señal de reloj de referencia escalándolos con el periodo de la señal de reloj.

APÉNDICE C

PROGRAMA GENERADOR DE LIBRERÍAS DE MEMORIAS ROM

C.1. Introducción

Los diseños empleados en las simulaciones han requerido en todos los casos el uso de tablas de consulta LUTs. Las tablas de consulta son necesarias para realizar la conversión de un dato binario en el residuo correspondiente para cada canal, según se expresa en (2.2). También se necesitan LUTs para la transformación inversa en la aplicación de la ε -CRT en la que un conjunto de residuos se convierten en la representación binaria escalada del dato de salida correspondiente, tal como quedó explicado en (2.20). Además, en los canales del controlador PID la implementación de la Figura 4.24 con los coeficientes dados por (4.15) se requieren multiplicadores, que se implementan de modo eficaz mediante LUTs. En el sistema para la realización de la transformada 1D-DWT, cada uno de los canales del filtro requería la transformación de un residuo en el correspondiente índice y posteriormente la transformación inversa, según queda representado en la Figura 4.32 y en la expresión (2.25).

En definitiva, ha quedado claro que el uso de las tablas de consulta, implementadas mediante memorias ROM, es fundamental para todos los sistemas que se han simulado. Esta necesidad se ha materializado en el desarrollo de un programa capaz de generar una librería para PSpice que modela una memoria ROM según se ha descrito en el apéndice B. El archivo de entrada es de tipo *mif* (*memory initialization*

```
-- Comments

WIDTH = 10;
DEPTH = 8;

CONTENT BEGIN
    0      :      0;
    1      :     302;
    2      :     395;
    3      :     935;
    4      :     66;
    5      :     38;
    6      :     965;
    7      :     41;
END;
```

Figura C.1. Ejemplo de archivo de descripción de la memoria.

file, archivo de inicialización de memoria) que es un formato típico de Altera, contiene en formato ASCII, la anchura en bits del dato de salida, la profundidad de la memoria, es decir, el número de datos que la constituyen y el contenido de la propia memoria. La Figura C.1 muestra un ejemplo de descripción de memoria.

C.2. Programa generador de librerías

El organigrama que el programa sigue en la generación del archivo de librería se muestra en la Figura C.2. En primer lugar se capturan del archivo de texto que describe la memoria los parámetros básicos como son las longitudes del bus de direcciones y de la palabra de salida. Si el bus de direcciones tiene una longitud mayor que 6, será necesaria la inclusión de las celdas `sel2_1` para completar la selección del dato de salida con los bits menos significativos de la dirección. En este caso, en primer lugar se determina la profundidad de las celdas `cell_2x8` según la expresión (C.1).

$$\left\lceil \frac{2^{D-6} \times L}{8} \right\rceil \quad (\text{C.1})$$

Posteriormente se generan 32 celdas `dec_64` para generar las 64 líneas de selección de fila. Cada celda tiene dos salidas cada una de las cuales se activan con dos posibles combinaciones de los 6 bits más significativos del bus de direcciones, tal como se explicó en el apéndice B. Después se generan los $D-6$ niveles de celdas `sel2_1` con $2^{D-6-i} \times L$ ($i = 1, \dots, D-6$) celdas en cada nivel. En este caso los datos aparecerán entremezclados en las celdas `cell_2x8` de modo que los bits correspondientes a cada dato se encuentran separados cada 2^{D-6} bits de las celdas `cell_2x8`.

En el caso que el bus de direcciones fuera menor o igual que 6 bastaría con la decodificación de filas para obtener el dato completo. En este caso resulta claro que la profundidad de las celdas `cell_2x8` solo depende del dato de salida según (C.2).

$$\left\lceil \frac{L}{8} \right\rceil \quad (\text{C.2})$$

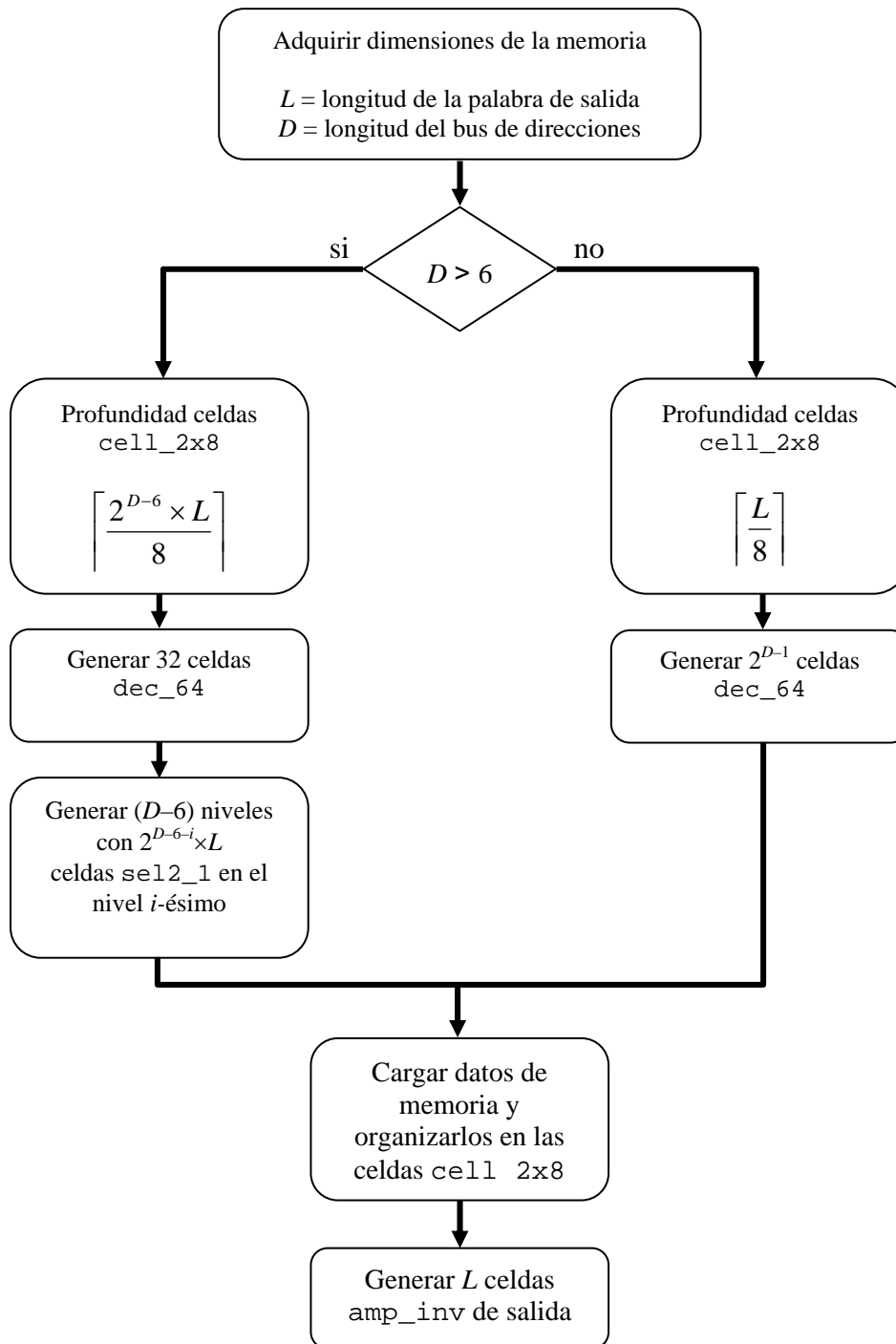


Figura C.2. Organigrama para la generación de la librería.

En este caso solamente son necesarias 2^{D-1} celdas dec_64 para generar las 2^D palabras que contiene la memoria.

En este momento se accede al archivo de texto para extraer el contenido de la memoria. En primer lugar se realiza la transformación de decimal a binario de cada dato y se traducen los bits a conexión en el caso de los '1' o no conexión en el caso de los '0' entre los drenadores de los transistores que constituyen la celda `cell_2x8` con la línea vertical de datos, tal como se explicó en el apéndice B. En el establecimiento de las conexiones se tienen en cuenta los niveles de celdas `sel2_1` para intercalar los bits que corresponden a cada dato. Finalmente, se generan las L celdas `amp_inv` que corresponden a cada bit de la palabra de salida que tendrán como entrada bien las salidas del último nivel de celdas `sel2_1` en el caso $D > 6$ o directamente las líneas verticales provenientes de las celdas `cell_2x8` en el caso $D \leq 6$.

Bibliografía

- [ACK05] Ackland, B.; Razavi, B.; West, L.: *A Comparison of Electrical and Optical Clock Networks in Nanometer Technologies*. IEEE 2005 CUSTOM INTEGRATED CIRCUITS CONFERENCE, págs. 779-782, 2005.
- [AFG89] Afghahim, M.; Svensson, C.: *Calculation of clock path delay and skew in VLSI synchronous systems*. EUROPEAN CONFERENCE ON CIRCUIT THEORY AND DESIGN IEE CONFERENCE PUBLICATION. nº 308, págs. 265-269, 1989.
- [AFG90] Afghahim, M.; Svensson, C.: *A unified single-phase clocking scheme for VLSI systems*. IEEE JOURNAL OF SOLID STATE CIRCUITS 1990, vol. 25, nº 1, págs. 225-233, 1990.
- [ALI91] Alia, G.; Martinelli, E.: *A VLSI modulo m multiplier*. IEEE TRANSACTIONS ON COMPUTERS, vol. 40, nº 7, págs 873-878, 1991.
- [ALT12] Altera Corp. *Stratix V Device Handbook*. Disponible online en: <http://www.altera.com/literature/lit-stratix-v.jsp>
- [ASG04] Asgari, F.; Sachdev, M.: *A Low-Power Reduced Swing Global Clocking Methodology*. IEEE TRANSACTIONS ON VERY LARGE SCALE

INTEGRATION (VLSI) SYSTEMS, vol. 12, n° 5, págs. 534-545, mayo 2004.

- [AST96] Astrom K. J.; Hagglund T., *PID control*. en *The Control Handbook*, CRC Press, 1996.
- [ASW08] Aswatha, A. R.; Basavaraju, T.: *Faster delay modeling and power optimization for on-chip global interconnects*. IEEE INTERNATIONAL CONFERENCE ON SEMICONDUCTOR ELECTRONICS (ICSE), págs. 82-86, 2008.
- [ATM03] Atmaca, E.; Lei, V.; Teo, M.; Drego, N.; Boning, D.; Fonstad, C. G.; Khai, L. W.; Fatt, Y. S.: *RM³ integration on InP based 1.55mm P-i-N photodetectors with silicon CMOS optical clock distribution circuits*. INTERNATIONAL SYMPOSIUM ON COMPOUND SEMICONDUCTORS: POST-CONFERENCE PROCEEDINGS, págs. 204-209, 2003.
- [BAN74] Banerji, D. K.: *A novel implementation method for addition and subtraction in residue number system*. IEEE TRANSACTIONS ON COMPUTERS 1974, vol. 23, n° 1, págs. 106-109, 1974.
- [BAN81] Banerji, D. K.: *A high speed division method in residue arithmetic*. PROCEEDINGS OF 1981 IEEE FIFTH SYMPOSIUM ON COMPUTER ARITHMETIC, págs. 158-164, mayo, 1981.
- [BAI04] Bai, X.; Chandra, R.; Dey, S.; Srinivas, P.V.: *Interconnect coupling-aware driver modeling in static noise analysis for nanometer circuits*. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, vol. 23, n° 8, págs. 1256-1263, 2004.
- [BAK86] Bakoglu, H. B.; Walker, J. T.; Meindl, J. D. *A symmetric clock-distribution tree and optimized high-speed interconnections for reduced clock skew in WSI circuits*. IEEE INTERNATIONAL CONFERENCE ON COMPUTER

- DESIGN: VLSI IN COMPUTERS AND PROCESSORS (ICCD-86), págs. 118-122, octubre 1986.
- [BAY87] Bayoumi, M.A.; Jullien, G. A.; Miller, W. C.: *A VLSI implementation of residue adders*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS 1987, vol. 34, págs. 284-288, 1987.
- [BEN94] Benini, L.; Siegel, P.; De Micheli, G.: *Saving power by synthesizing gated clocks for sequential circuits*. IEEE DESIGN & TEST OF COMPUTERS, vol. 11, nº 4, págs. 32-41, 1994.
- [BEN96] Benini, L.; Vuillod, P.; Bogliolo, A.; De Micheli, G.: *Clock skew optimization for peak current reduction*, JOURNAL OF VLSI SIGNAL PROCESSING 1997 vol. 16, págs. 117-130, 1996.
- [BHA98] Bhardwaj, M. A.; Balaram, A.: *Low power signal processing architectures using residue arithmetic*. Proc. of 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (Seattle, USA, 11–15 mayo), 1998.
- [BIN03] Bindal, N.; Kelly, T.; Velastegui, N.; Wong, K.: *Scalable sub-10ps skew global clock distribution for a 90nm multi-GHz IA microprocessor*, VISUAL SUPPLEMENT IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'03), págs. 281-284, 2003.
- [BLA92] Blair, G.M.: *Skew-free clock distribution for standard-cell VLSI designs*. IEE Proceedings on Circuits, Devices and Systems, vol.139 nº 2, págs. 265-267, abril 1992.
- [BOE98] Boerstler, D.; Jenkins, K.: *A phase-locked loop clock generator for a 1 GHz microprocessor*. PROCEEDINGS OF GIGEST TECHNICAL PAPERS VLSI CIRCUITS 1998 SYMPOSIUM, págs. 212-213, junio 1998.

- [BOE99] Boerstler, D.: *A low jitter PLL clock generator for microprocessors with lock range of 340-612 MHz*. IEEE JOURNAL OF SOLID STATE CIRCUITS, vol. 34, n° 4, págs. 513-519, abril 1999.
- [BUT07] Buthada, R.; Manoli, Y.: *Complex Clock Gating with Integrated Clock Gating Logic Cell*. INTERNATIONAL CONFERENCE ON DESIGN & TECHNOLOGY OF INTEGRATED SYSTEMS IN NANOSCALE ERA (DTIS), págs. 164-169, 2007.
- [CAS07] Castillo, E.; Meyer-Baese, U.; García, A.; Parrilla, L.; Lloris, A.: *IPP@HDL: Efficient Intellectual Property Protection Scheme for IP Cores*. IEEE TRANSACTIONS ON VLSI SYSTEMS, vol. 15, págs. 578-591, 2007.
- [CEL99] Celik, M.; Pileggi, Lawrence T.: *Metrics and bounds for phase delay and signal attenuation in RC(L) clock trees*. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS 1999, vol. 18, n° 3, págs. 293-300, 1999.
- [CHA00] Charbon, E.; Torunoglu, I. H.: *On intellectual property protection*. PROCEEDINGS OF 2000 IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE, págs 517-523, 2000.
- [CHA05] Chang, J.; Rusu, S.; Shoemaker, J.; Tam, S.; Ming Huang; Haque, M.; Siufu Chiu; Kevin Truong; Karim, M.; Leong, G.; Desai, K.; Goe, R.; Kulkarni, S.: *A 130-nm triple-Vt 9-MB third-level on-die cache for the 1.7-GHz Itanium® 2 processor*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 40 , n°: 1, pás: 195-203, 2005.
- [CHI99] Chiang, J.; Chen, K.: *The desing of an All-Digital Phase-Locked Loop with small DCO hardware and fast phase lock*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING. vol. 45, n° 9, págs. 1220–1231, septiembre 1998.

- [CHI08] Chang, C.; Huang, S.; Ho, Y.; Lin, J.; Wang, H.; Lu, Y.: *Type-Matching Clock Tree for Zero Skew Clock Gating*. 45TH ACM/IEEE DESIGN AUTOMATION CONFERENCE (DAC 2008), págs. 714-719, 2008.
- [CHO03] Cho, S.; Seo, S.; Jokerst, N. M.: *Planar Lightwave Circuit Balanced Optical Fanout Using a Thin Film Photodetector Array Embedded in a Polymer MMI Coupler*. THE 16TH ANNUAL MEETING OF THE IEEE LASERS AND ELECTRO-OPTICS SOCIETY, 2003 (LEOS 2003), vol. 1, págs. 65-66, 2003.
- [CLA04] Claves, J.; Friedrich, J.; Sweet, M.; Dilullo, J.; Chu, S.; Plass, D.; Dawson, J.; Muench, P.; Powell, L.; Floyd, M.; Sinharoy, B.; Lee, M. Goulet, M.; Wagoner, J.; Schwartz, N.; Runyon, S.; Gorman, G.; Restle, P.; Kalla, R.; McGill, J; Dodson, S.: *Design and implementation of the POWER5TM microprocessor*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'04), págs. 56-57, 2004.
- [CON93] Cong, J.; Kahng, A. B.; Robins, G.: *Matching-based methods for high-performance clock routing*. IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN, vol. 12, n° 8, págs. 1157-1169, 1993.
- [COX07] Cox, I., Miller, M., Bloom, J., Fridrich, J.: *Digital watermarking and steganography*. Segunda edición. The Morgan Kauffmann Series in Multimedia Information and Systems, 2007.
- [DEB03] Debaes, C.; Bhatnagar, A.; Agarwal, D.; Chen, R.; Keeler, G. A.; Helman, N. C.; Thienpont, H.; Miller, D. A. B.: *Receiver-Less Optical Clock Injection for Clock Distribution Networks*. IEEE JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS, vol. 9, n° 2, págs. 400-409, marzo/abril 2003.
- [DEU95] Deutsch, A.; Kopcsay, G. V.; Surovic, C. W.; Rubin, B. J.; Terman, L. M.; Dunne, R. P.; Gallo, T. A.; Dennard, R. H.: *Modeling and characterization*

of long on-chip interconnections for high-performance microprocessors. IBM JOURNAL OF RESEARCH AND DEVELOPMENT, vol. 39, págs. 547-567, septiembre 1995.

- [DHA91] Dhar, S.; Franklin, M.A: *Optimum buffer circuits for driving long uniform lines.* IEEE JOURNAL OF SOLID-STATE CIRCUITS. vol. 26 n° 1, págs. 32-40, enero 1991.
- [DIC95] Di Claudio, E.; Piazza, F.; Orlandi, G.: *Fast combinational RNS processors for DSP applications.* IEEE TRANSACTIONS ON COMPUTERS, págs. 325-329, 1992.
- [DOR07] Dorsey, J., Searles, S.; Ciraula, M.; Johnson, S.; Bujanos, N.; Wu, D.; Braganza, M.; Meyers, S. Fang, E.; Kumar, R.: *An integrated quad-core Opteron Processor.* DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'07), págs. 102-103, 2007.
- [DUG92] Dugdale, M.: *VLSY Implementation of residue adders based on binary adders.* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM II, vol. 39, n° 5, págs. 325-329, 1992.
- [DUG94] Dugdale, M.: *Residue multipliers usign factored decopmposition.* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM II: ANALOG AND DIGITAL SIGNAL PROCESSIGN, vol. 41, págs. 623-627, 1994.
- [DUN95] Dunning, J.; García, G.; Lundberg, J.; Nuckolls, E.: *An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors.* IEEE JOURNAL OF SOLID STATE CIRCUITS, vol. 30, págs. 412-422, Abril 1995.
- [ELL95] Elleithy, K. M.; Bayoumi, M. A.: *A systolic architecture for modulo multiplication.* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEM II:

- ANALOG AND DIGITAL SIGNAL PROCESSING, vol. 42, nº 11, págs. 725-729, 1995.
- [ESM10] Esmaeili, S.E.; Farhangi, A.M.; Al-Khalili, A.J.; Cowan, G.E.R.: *Skew compensation in energy recovery clock distribution networks*. COMPUTERS & DIGITAL TECHNIQUES (IET), vol. 4, nº 1, págs. 56-72, 2010.
- [ETZ80] Etzel, M.; Jenkins, W. K.: *Redundant residue number systems for error detection and correction in digital filters*. IEEE TRANSACTIONS ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING 1980, vol. 29, nº 5, 1980.
- [FIS90] Fishburn, J. P.: *Clock skew optimization* IEEE TRANSACTIONS ON COMPUTERS, vol. 39, nº 7, págs. 945-951, Julio 1990.
- [FRI93] Friedman, E. G.: *Clock Distribution design in VLSI circuits – an overview*. IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS 1993 (ISCAS'93), vol. 3, págs. 1475-1478, mayo 1993.
- [FRI01] Friedman, E., G.: *Clock distribution networks in synchronous digital integrated circuits*. PROCEEDINGS OF THE IEEE 2001, vol. 89, nº 5, págs. 665-692, 2001.
- [FRI07] Friedrich, J.; McCredie, B.; James, N.; Huott, B.; Curran, B.; Fluhr, E.; Mittal, G.; Chan, E.; Chan, Y.; Plass, D.; Sam Chu; Hung Le; Clark, L.; Ripley, J.; Taylor, S.; Dilullo, J.; Lanzerotti, M.: *Design of the Power6 Microprocessor*. IEEE INTERNATIONAL DIGEST OF TECHNICAL PAPERS IN SOLID-STATE CIRCUITS CONFERENCE, 2007 (ISSCC'07), págs 96-97, 2007.
- [GAR77] Gardner, F. M.: *Hangup in phase-lock loops*. IEEE TRANSACTIONS ON COMMUNICATIONS, vol. COM-25, nº 10, págs. 1210-1214, octubre 1977.

- [GAR98] García, A.; Meyer-Baese, U.; Taylor, F. J.: *Pipelined Hogenauer CIC Filters Using Field-Programmable Logic and Residue Number System*. PROC. OF 1998 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING ICASSP'98 (Seattle WA), vol. 5, págs. 3085-3088, 11-15 de mayo, 1998.
- [GAR99] García, A.; Lloris Ruíz, A.: *Procesamiento digital de señales de altas prestaciones utilizando el sistema numérico de residuos*. Universidad de Granada, 1999. ISBN 84-699-1801-X.
- [GEA98] Geannopoulos, G.; Dai, X.: *An adaptative digital deskewing circuit for clock distribution networks*, VISUAL SUPPLEMENT IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'98), 1998, págs. 400-4001, 1998.
- [GER08] Gerosa, G.; Curtis, S.; D'Addeo, M.; Bo Jiang; Kuttanna, B.; Merchant, F.; Patel, B.; Taufique, M.; Samarchi, H.; *A Sub-1W to 2W Low-Power IA Processor for Mobile Internet Devices and Ultra-Mobile PCs in 45nm Hi-K Metal Gate CMOS*. IEEE INTERNATIONAL DIGEST OF TECHNICAL PAPERS SOLID-STATE CIRCUITS CONFERENCE, 2008 (ISSCC'08), págs. 256-611, 2008.
- [GIA07] Giacomotto, C.; Nedovic, N.; Oklobdzija, V.G.: *The Effect of the System Specification on the Optimal Selection of Clocked Storage Elements*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 42, nº 6, págs. 1392-1404, Junio 2007.
- [GIE97] Gieseke, B.A.; Allmon, R.L.; Bailey, D.W.; Benschneider, B.J.; Britton, S.M.; Clouser, J.D.; Fair, H.R., III; Farrell, J.A.; Gowan, M.K.; Houghton, C.L.; Keller, J.B.; Lee, T.H.; Leibholz, D.L.; Lowell, S.C.; Matson, M.D.; Matthew, R.J.; Peng, V.; Quinn, M.D.; Priore, D.A.; Smith, M.J.; Wilcox, K.E.: *A 600 MHz superscalar RISC microprocessor with out-of-order*

- execution*. DIGEST OF TECHNICAL PAPERS IN SOLID-STATE CIRCUITS CONFERENCE 1997, págs. 176-177, 1997.
- [GOL94] Golshan, R.; Haroun, B.: *A novel reduced swing CMOS BUS interface circuit for high speed low power VLSI systems*. PROCEEDINGS IEEE INTERNATIONAL SYMPOSIUM CIRCUITS AND SYSTEMS, vol. 4, págs. 351–354, mayo, 1994.
- [GOL06] Golden, M.; Arekapudi, S.; Dabney, G.; Haertel, M.; Hale, S.; Herlinger, L.; Kim, Y.; McGrath, K.; Palisetti, V.; Singh, M.: *A 2.6GHz dual-core 64bx86 microprocessor with DDR2 memory support*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'06), 2006, págs. 325-332, 2006.
- [GON83] Goncalves, N.; De Man H. J.: *NORA: A racefree dynamic CMOS technique for pipelined logic structures*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol SC-18, nº 5, págs. 261-266, 1983.
- [GON02a] González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *A new methodology for efficient synchronizaton of RNS-based VLSI systems*, LECTURE NOTES IN COMPUTER SCIENCE, vol. 2451, págs 188-197, 2002.
- [GON02b] González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *Clock Distribution in RNS-based VLSI Systems*, ADVANCES IN SYSTEMS ENGINEERING: SIGNAL PROCESSING AND COMMUNICATIONS (N. Mastorakis, Ed.). Electrical and Computer Engineering Series, World Scientific and Engineering Society Press, págs. 323-328, 2002 (ISBN 960-8052-69-6).
- [GON02c] González, D.; García, A.; Jullien, G. A.; Ramírez, J.; Parrilla, L.; Lloris, A.: *A New Clock Distribution Strategy in RNS-based VLSI Systems*, PROC. OF XVII CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS DCIS'2002, págs 175-180, Santander, Noviembre, 2002.

- [GON05] González, D.; Parrilla, L.; García, A.; Castillo, E.; Lloris, A.: *Efficient Clock Distribution Scheme for VLSI RNS Enabled Controllers*. LECTURE NOTES IN COMPUTER SCIENCE, vol. 3728, págs. 657-665, 2005.
- [GRE74] Greenhouse, H.: *Design of planar rectangular microelectronic inductors*. IEEE TRANS PARTS, HYBRID AND PACKAGING, vol. 10, n° 2, págs 101-109, junio 1974.
- [GRI88] Griffin, M; Taylor, F. J.; Sousa, M.: *New scaling algorithms for the chinese remainder theorem*. PROCEEDINGS OF 22ND ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS AND COMPUTERS, 1988.
- [GRI89] Griffin, M; Taylor, F. J.; Sousa, M.: *Efficient scaling in the residue number system*. PROCEEDINGS OF 1989 INTERNATIONAL CONFERENCE ON ACUOSTICS, SPEECH AND SIGNAL PROCESSING, págs 1075-1078, 1989.
- [GRO94] Grover, W. D., *A new Method for clock distribution*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: FUNDAMENTAL THEORY AND APLICATIONS, vol. 41, n° 2, págs. 149-160, febrero 1994.
- [GRO98] Gronowski, P.E.; Bowhill, W.J.; Preston, R.P.; Gowan, M.K.; Allmon, R.L.: *High-performance microprocessor design*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 33, n° 5, págs. 676-686, 1998.
- [HAJ98] Hajimiri, A. Lee, T. H.: *A general theory of phase noise in electrical oscillators*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 33, n° 2, págs. 179-194, febrero 1998.
- [HAR97] Harris, D.; Horowitz, M.A.: *Skew-tolerant domino circuits*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 32, n° 11, págs. 1702-1711, noviembre 1997.

- [HAT88] Hatamian, M.; Tewksbury, S. K.; Dickinson, B. W.; Schwartz, S. C.: *Understanding clock skew in synchronous systems*. CONCURRENT COMPUTATIONS (ALGORITHMS, ARCHITECTURE AND TECHNOLOGY), Ed. New York: Plenum, págs. 87-96, 1988.
- [HER99] Herzel, F., Razavi, B.: *Oscillator jitter due to supply and substrate noise*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, vol. 46, n° 1, págs. 56-62, 1999.
- [HEY00] Heydari, P.; Pedram, M.: *Analysis of jitter due to power-supply noise in phase-locked loops*. PROCEEDINGS OF THE IEEE CUSTOM INTEGRATED CIRCUITS CONFERENCE, págs. 443-446, mayo 2000.
- [HEY01] Heydari, P.; Pedram, M.: *Jitter-induced power/ground noise in CMOS PLLs: a design perspective*. INTERNATIONAL CONFERENCE ON COMPUTER DESIGN, 2001. pág. 209-213, 2001.
- [HIA96] Hiasat, A.: *Semi-Custom VLSI Design for RNS Multipliers Using Combinational Logic Approach*. PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER SYSTEMS, vol. 2, págs. 935-938, 1996.
- [HIA97] Hiasat, A.; Abdel-Aty-Zohdy, H. S.: *Design and implementation of an RNS division algorithm*. PROCEEDINGS OF 13th IEEE SYMPOSIUM ON COMPUTER ARITHMETIC (Asilomar, USA), págs. 240-249, julio 1997.
- [HIA00] Hiasat, A. A.: *New Efficient Structure for a Modular Multiplier for RNS*. IEEE TRANSACTIONS ON COMPUTERS, vol. 49, n° 2, 170-174, 2000.
- [HIG11] Higami, Y.; Takahashi, H.; Kobayashi, S.; Saluja, K.K.: *On Detecting Transition Faults in the Presence of Clock Delay Faults*. 20th Asian Test Symposium (ATS), págs. 1-6, 2011.

- [HIT95]** Hitz, M. A.; Kaltofen, E.: *Integer division in residue number system*. IEEE TRANSACTIONS ON COMPUTERS 1995, vol. 44, n° 8, págs. 983-989, 1995.
- [HOF00]** Hofstee, P.; Akou, N.; Boerstler, D.; Coulman, P.; Dhong, S.; Flachs, B.; Kojima, N.; Kwon, O, Lee, K.; Meltzer, D.; Nowka, K.; Park, J.; Peter, J.; Posluszny, M.; Shapiro, N.; Silberman, J.; Takahashi, O; Weinberger, B.: *A 1 GHz single-issue 64b PowerPC processor*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'00), págs. 92-93, 2000.
- [HOG81]** Hogenauer, E. B.: *An economical class of digital filters for decimation and interpolation*. IEEE TRANSACTIONS ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING 1981, vol. 29, n° 2, págs. 155-162, 1981.
- [HON09]** Honkote, V.; Taskin, B.: *Zero clock skew synchronization with rotary clocking technology*. 10th INTERNATIONAL SYMPOSIUM ON QUALITY ELECTRONIC DESIGN, págs. 588-593, 2009.
- [HUD09]** Huda, S.; Mallick, M.; Anderson, J. H.: *Clock gating architectures for FPGA power reduction*. INTERNATIONAL CONFERENCE ON FPL 2009, págs. 112-118, 2009.
- [ISM00]** Ismail, Y.I.; Friedman, E.G.: *Effects of inductance on the propagation delay and repeater insertion in VLSI circuits*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 8 n° 2, págs. 195-206, abril 2000.
- [JAC90]** Jackson, M. A. B.; Srinivasan, A.; Kuh, E. S.: *Clock routing for high performance IC's*. 27th ACM/IEEE DESIGN AUTOMATION CONFERENCE, págs 573-579, 1990.
- [JAC01]** Jacomem, M. F.; Peixoto, H. P.: *A survey of digital design reuse*. IEEE DESIGN & TEST OF COMPUTERS, vol. 18, n° 3, págs. 98-107, 2001.

- [JAN12] Jang, J.; Franza, O.; Burleson, W.: *Compact Expressions for Supply Noise Induced Period Jitter of Global Binary Clock Trees*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 20, nº 1, págs. 66-79, 2012.
- [JEN83] Jenkins, W. K.; Krozmeier, J. J.: *Error detection and correction in quadratic residue number systems*. PROC. OF 26th MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS (Pueblo Mexico, USA), agosto 1983.
- [JEN02] Jenei, S.; Nauwelaers, B.; Decoutere, S.: *Physics-based closed-form inductance expression for compact modeling of integrated spiral inductors*. IEEE JOURNAL OF SOLID STATE CIRCUITS, vol. 37, nº 1, págs. 77-80, enero 2002.
- [JIA10] Jiang, Y.; Wong, K.-F.; Cai, C.-Y.; Sin, S.-W.; Seng-Pan U.; Martins, R.P.: *A reduced jitter-sensitivity clock generation technique for continuous-time Sigma-Delta modulators*. 2010 IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS (APCCAS), págs. 1011-1014, 2010.
- [JIR87] Ji-Ren, Y.; Karlsson, I.; Svensson, C.: *A true single-phase-clock dynamic CMOS circuit technique*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 22, nº 5, págs. 899-901, 1987.
- [JIA01] Jiang, X.; Horiguchi, S.: *Statistical skew modeling for general clock distribution networks in presence of process variations*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 9, nº 5, octubre 2001.
- [JOH80] Johnson, E.: *A digital quarter square multiplier*. IEEE TRANSACTIONS ON COMPUTER, vol. C-19, nº 3, págs. 258-261, marzo 1980.

- [JUL78] Jullien, G. A.: *Residue number scaling and other operations using ROM arrays*. IEEE JOURNAL OF SOLID STATE CIRCUITS 1987, vol.22, nº 5, págs. 899-901, 1978.
- [JUL80] Jullien, G.: *Implementation of multiplication, modulo a prime number, with applications to number theoretic transforms*. IEEE TRANSACTIONS ON COMPUTER, vol. C-29, nº 10, págs. 899-905, 1980.
- [KAH91] Kahng, A.; Cong, J.; Robins, G.; *High-performance clock routing based on recursive geometric matching*. PROCEEDINGS OF DESIGN AUTOMATION CONFERENCE, págs. 322-327, 1991.
- [KAR88] Karlsson, I.; *True single phase clock dynamic CMOS circuit technique*. IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, 1988, vol. 1 págs. 475-478, 1988.
- [KAR10] Kar, R.; Maheshwari, V.; Choudhary, A.; Singh, A.; Mal, A.K.; Bhattacharjee, A.K.: *Inductive coupling aware explicit cross-talk and delay formula for on-chip VLSI RLCG interconnects using difference model approach*. INTERNATIONAL CONFERENCE ON COMPUTING COMMUNICATION AND NETWORKING TECHNOLOGIES (ICCCNT), págs. 1-6, 2010
- [KEE91] Keezer, D. C.; Jain, V.K.: *Clock distribution strategies for WSI: a critical survey*. PROC. [3RD] INTERNATIONAL CONFERENCE ON WAFER SCALE INTEGRATION, págs. 277-283, enero 1991.
- [KHL04] Khlifi, H.; Gregoire, J.-C.: *Estimation and removal of clock skew from delay measures*. 29TH ANNUAL IEEE INTERNATIONAL CONFERENCE ON LOCAL COMPUTER NETWORKS, págs. 144-151, 2004.
- [KIM97] Kim, B.; KIM, L.: *A low-power 100MHz all digital delay-locked loop*. IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, págs. 1820-1823, junio 1997.

- [KIN74] Kinoshita, E.; Kosako, H.; Kojima, Y.: *Floating-point arithmetic algorithms in the symmetric residue number system*. IEEE TRANSACTIONS ON COMPUTERS, vol. 23, n° 1, págs. 9-23, 1974.
- [KIM10] Kim, T.-Y.; Kim, T.: *Bounded skew clock routing for 3D stacked IC designs: Enabling trade-offs between power and clock skew*. 2010 INTERNATIONAL GREEN COMPUTING CONFERENCE, págs. 525-532, 2010.
- [KLA99] Klass, F.; Amir, C.; Das, A.; Aingaran, K.; Truong, C.; Wang, R.; Mehta, A.; Heald, R.; Yee, G.: *A new family of semidynamic and dynamic flip-flops with embedded logic for high-performance processors*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 34, n° 5, págs. 712-716, mayo, 1999.
- [KOC99] Kocher, P.; Jaffe, J.; Jun, D.: *Differential Power Analysis*, LECTURE NOTES IN COMPUTER SCIENCE, vol. 1666, págs. 388-397, 1999.
- [KRI94] Krishna, H; Krishna, B.; Lin K. Y.; Sun, J. D.: *Computational number theory and digital signal processing. Fast algorithms and error control techniques*. CRC Press, 1994.
- [KRI98] Krishna, H: *Digital signal processing algorithms, number theory, convolution, fast Fourier transforms, and applications*. CRC Press, 1998.
- [KRU08] Krueger, D.; Francom, E.; Langsdorf, J.: *Circuit design for voltage scaling and SER immunity on a quad-core itanium® Processor*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC 2008), págs. 94-95, 2008.
- [KUG88] Kugelmass, S. D.; Steiglitz, K.: *A probabilistic Model for clock Skew*. PROCEEDINGS OF INTERNATIONAL CONFERENCE ON SYSTOLIC ARRAYS PUBLISHED BY IEEE, págs. 545-554, 1988.

- [**KUO07**] Kuon, I; Rose, J: *Measuring the Gap Between FPGAs and ASICs*, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, vol. 26, nº 2, págs 203-215, 2007.
- [**KUP01**] Kuppuswamy, R.; Callahan, K.; Keng Wong; Ratchen, D.; Taylor, G.: *On-die clock jitter detector for high speed microprocessors*. 2001 SYMPOSIUM ON VLSI CIRCUITS DIGEST OF TECHNICAL PAPERS, págs: 187-190, 2001.
- [**KUR01**] Kurd, N.; Barkatullah, J.; Dizon, R.; Fletcher, T.; Madlan, P.: *A multigigahertz clocking scheme for the Pentium® 4 microprocessor*. 2001 IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 36, nº 11, págs. 1647-1653, noviembre 2001.
- [**LAM96**] Lam, K. N.; Devadas, S.: *PECS: A peak current and power simulator for CMOS combinational circuits*. 1996 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS'96), vol. 4 págs. 488 –491, 1996.
- [**LEE98**] Lee, M.; Darley, M.H.: *An interconnect transient coupling induced noise susceptibility for dynamic circuits in deep submicron CMOS technology*. PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS '98), vol. 2, págs. 256-259, 1998.
- [**LEU81**] Leung, S. H.: *Application of residue Number Systems to Complex Digital Filters*. PROCEEDINGS OF THE 15th ASILOMAR CONFERENCE ON SIGNALS, SYSTEM AND COMPUTERS, 1981.
- [**LIU01**] Liu, Y.; Zhao, M.; Hong, X.; Cai, Y.; Wu, W.: *A clustering-based algorithm for zero-skew routing with buffer insertion*. PROCEEDINGS ON 4th INTERNATIONAL CONFERENCE ON ASIC, págs. 183-186, 2001.

- [LIN92] Lin, I.; Ludwig, J.A.; Eng, K.: *Analyzing cycle stealing on synchronous circuits with level-sensitive latches*. PROCEEDINGS 29th ACM/IEEE DESIGN AUTOMATION CONFERENCE, págs. 393-398, 1992.
- [LLO92] Lloris, A.; *Teoría de la Conmutación*. Publicaciones de la Facultad de Ciencias de Granada, 1992.
- [MAH99] Maheshwari, N.; Sapatnekar, S.: *Optimizing large multiphase level-clocked circuits*. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, vol. 18, n° 9, págs. 1249-1264, septiembre 1999.
- [MAH06] Mahawar, H.; Sarin, V.: *Parallel algorithms for inductance extraction of VLSI circuits*. 20TH INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS), 2006.
- [MAL89a] Mallat, S. G.: *A theory for multiresolution signal decomposition: the wavelet representation*. IEEE TRANSACTIONS ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE, vol. 11, n° 7, págs. 674-693, 1989.
- [MAL89b] Mallat, S. G.: *Multifrequency channel decompositions of images and wavelets models*. IEEE TRANSACTIONS ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, vol. 37, n° 12, págs. 2091-2110, 1989.
- [MAN72] Mandelbaun, D.: *Error correction in residue arithmetic*. IEEE TRANSACTIONS ON COMPUTERS, vol. 21, n° 6, págs. 538-545, 1972.
- [MAR01] Markovic, D.; Nikolic, B.; Brodersen, R.W.: *Analysis and design of low-energy flip-flops*. INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN 2001, págs. 52-55, 2001.
- [MAT94] Matsui, M.; Hara, H.; Uetani, Y.; Kim, L.-S.; Nagamatsu, T.; Watanabe, Y.; Chiba, A.; Matsuda, K.; Sakurai, T.: *A 200MHz 13mm²-2D DCT macrocell*

using sense-amplifying pipeline flip-flop scheme. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 29, n° 12, págs. 1482-1490, diciembre 1994.

[MEY01] Meyer-Baese, U.; García, A.; Taylor, F.: *Implementation of a Communications Channelizer using FPGAs and RNS Arithmetic.* JOURNAL OF VLSI SIGNAL PROCESSING, vol. 28, n°. 1/2, págs. 115-128, mayo 2001.

[MIL98] Miller, D. F.; McCormick, W. S.: *An arithmetic free parallel mixed-radix conversion algorithm.* IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-ANALOG AND DIGITAL SIGNAL PROCESSING 1998, vol. 45, n° 1, págs. 158-162, 1998.

[MOH99] Mohan, S; Hershenson, M. d M.; Boyd, S.; Lee, T.: *Simple accurate expression for planar spiral inductances.* IEEE JOURNAL OF SOLID STATE CIRCUITS, vol. 34, n° 10, págs. 1419-1424, octubre 1999.

[MOH11] Mohammadi-Arfa, Z.; Jahanian, A.: *A Hybrid RF/Metal Clock Routing Algorithm to Improve Clock Delay and Routing Congestion.* IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI (ISVLSI), págs. 138-143, 2011.

[MOI00] Moisiadis, Y.; Bouras, I.; Arapoyanni, A.: *High performance level restoration circuits for low-power reduced-swing interconnect schemes.* PROCEEDINGS IEEE 7TH INTERNATIONAL CIRCUITS AND SYSTEMS (ICECS), vol. 1, págs. 619-622, 2000.

[MOO65] Moore, G. E.: *Cramming more components onto Integrated circuits.* ELECTRONICS 1965, págs 114-117, 1965.

[MOR11] Moreno, P.; Carrio, F.; Castillo V.; Ferrer A.; Fiorini, L.; González, V.; Hernandez, Y.; Higon, E.; Sanchís, E.; Solans, C.; Valero, A.; Valls, J.: *Optimal Filtering Algorithm implementation in FPGAs for the ATLAS*

- TileCal Read-Out Drivers*. IEEE NUCLEAR SCIENCE SYMPOSIUM CONFERENCE RECORD, 2011.
- [MUE08] Mueller, J.; Saleh, R.: *A Tunable Clock Buffer for Intra-die PVT Compensation in Single-Edge Clock (SEC) Distribution Networks*. 9th INTERNATIONAL SYMPOSIUM ON QUALITY ELECTRONIC DESIGN, págs. 572-577, 2008.
- [NAF02] Naffziger, S.D.; Colon-Bonet, G.; Fischer, T.; Riedlinger, R.; Sullivan, T.J.; Grutkowski, T.: *The implementation of the Itanium 2 microprocessor*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 37, n° 11, págs. 1448-1460, 2002.
- [NAK93] Nakagome, Y.; Itoh, K.; Isoda, M.; Takeuchi, K.; Aoki, M.: *Sub-1-V swing internal bus architecture for future low power ULSIs*. IEEE J. SOLID-STATE CIRCUITS, vol. 28, págs. 414–419, abril, 1993.
- [NEK97] Nekili, M.; Savaria, Y.; Bois, G.: *Pipelined H-trees for high-speed clocking of large integrated systems in presence of process variations*, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 5 n° 2, págs. 161-174, junio 1997.
- [NEK98] Nekili, M.; Savaria, Y.; Bois, G.: *Design of clock distribution networks in presence of process variations*, PROCEEDINGS OF THE 8th GREAT LAKES SYMPOSIUM ON VLSI 1998, págs. 95 -102, 1998.
- [NEK01] Nekili, M.; Savaria, Y.; Bois, G.: *Minimizing process-induced skew using delay tuning*, THE 2001 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS'01), vol. 4, págs. 426-429, 2001.
- [NIC12] Nicolaidis, M.; Anghel, L.; Zergainoh, N.-E.; Zorian, Y.; Karnik, T.; Bowman, K.; Tschanz, J.; Shih-Lien Lu; Tokunaga, C.; Raychowdhury, A.; Khellah, M.; Kulkarni, J.; De, V.; Avresky, D.: *Design for test and*

reliability in ultimate CMOS. DESIGN, AUTOMATION & TEST IN EUROPE CONFERENCE & EXHIBITION (DATE), págs. 677-682, 2012.

- [NUS76] Nussbaumer, H.: *Digital filters using Read-Only Memories*. ELECTRONIC LETTERS, vol. 12, 1976.
- [OKL03] Oklobdzija, V.; Stojanovic, V.; Markovic, D.; Nedovic, N.: *Digital system clocking. High-performance and Low-power aspects*, Wiley-Interscience, 2003.
- [PAR03] Parrilla, L.; García, A.; González, D.; Castillo E.; Lloris, A.: *Efficient Polyphase Architectures for DWT Computation using RNS over FPL Devices*. PROCEEDINGS OF XVIII CONFERENCE ON DESIGN OF CIRCUITS AND INTEGRATED SYSTEMS (DCIS'2003 Ciudad Real) págs. 562-565, noviembre 2003.
- [PAR96] Partovi, H.; Burd, R.; Salim, U.; Weber, F.; DiGregorio, L.; Draper, D.: *Flow-through latch and edge-triggered flip-flop`hybrid elements*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC 1996), págs, 138-139, 1996.
- [PAY98] Payne, A.; Thanachayanont, A.; Papavassiliou, C.: *A 150-MHz translinear Phase-Locked Loop*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, vol 45, nº9, págs. 1220-1231.
- [PET95] Petrescu, V.; Mouthaan, T.; Schoenmaker, W.; Angelescu, S.; Vissarion, R.; Dima, G.; Wallinga, H.; Profirescu, M.D.: *Numerical analysis of electromigration in thin film VLSI interconnections*. INTERNATIONAL PROC. ON SEMICONDUCTOR CONFERENCE (CAS'95), págs. 327-330, 1995.

- [POS96] Posch, K. C.; Posch, R.: *División in residue number systems involving length indicators*. JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS 1996, vol. 66, págs. 456-462, 1996.
- [POL76] Pollard, J. M.: *Implementation of number-theoretic transforms*. ELECTRONIC LETTERS, vol. 12, nº 22, págs. 378-379, 1976.
- [RAB03] Rabaey, J. M.; Chandrakasan, A.; Nicolic B.: *Digital integrated circuits: a design perspective*. PRENTICE HALL ELECTRONICS AND VLSI SERIES, segunda edición 2003.
- [RAD91] Radhakrishnan, D.; Yuan, Y.: *Fast and highly compact RNS multipliers*. INTERNATIONAL JOURNAL OF ELECTRONICS, vol. 70, nº 2, págs. 281-293, 1991.
- [RAD92] Radhakrishnan, D.; Yuan, Y.: *Novel approaches of the design of VLSI RNS multipliers*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: ANALOG AND DIGITAL SIGNAL PROCESSING, vol. 39, págs. 52-57, 1992.
- [RAD98] Radhakrishnan, D.; Preethy, A.: *A new approach to data conversion: direct analog-to-residue converter*. PROCEEDINGS OF THE 1998 IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (Seattle, USA), vol. 5, págs. 3013-3016, mayo 1998.
- [RAM89] Ramanathan, P.; Shin, K. G.: *A clock distribution scheme for non-symmetric VLSI circuits*. PROCEEDINGS IEEE INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, págs. 398-401, 1989.
- [RAM94] Ramanathan, P.; Dupont, A. J.; Shin, K. G.: *Clock distribution in general VLSI circuits*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I: FUNDAMENTAL THEORY AND APPLICATIONS, vol. 41, nº 5, págs. 395-404, mayo 1994.

- [RAM01] Ramírez, J.; Fernández, P. G.; Meyer-Bäse, U.; Taylor, F.; García, A.; Lloris, A.: *Index-based RNS DWT architectures for Custom IC Designs*. PROC. OF THE 2001 IEEE WORKSHOP ON SIGNALS PROCESSING SYSTEMS (SiPS), págs. 70-79, septiembre 2001.
- [RAM05] Ramírez, J.; Meyer-Bäse, U.; García, A.: *Efficient RNS-based Design of Programmable FIR Filters Targeting FPL Technology*. JOURNAL OF CIRCUITS, SYSTEMS AND COMPUTERS, vol. 14, nº 1, págs. 165-177, febrero 2005.
- [RES98] Restle, P.J.; Jenkins, K.A.; Deutsch, A.; Cook, P.W.: *Measurement and modeling of on-chip transmission line effects in a 400 MHz microprocessor*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 33, nº 4, págs. 662-665, 1998.
- [RES02] Restle, P.; Carter, C.; Eckhardt, B.; Krauter, B.; McCredie, K.; Jenkins, K.; Weger, A.; Mule, A.: *The clock distribution of the Power4 microprocessor*. DIGEST OF TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'02), vol. 1, págs. 144-145, febrero 2002.
- [RJO99] Rjoub, A.; Koufopavlou, O.: *Efficient drivers, receivers and repeaters for low power CMOS bus architectures*. PROC. 6TH IEEE INT. SYMP. CIRCUITS AND SYSTEMS (ICECS), págs. 789-794, 1999.
- [ROS11] Rosiere, M.; Desbarbieux, J.-L.; Drach, N.; Wajsburt, F.: *Morpheo: A high-performance processor generator for a FPGA implementation*. CONFERENCE ON DESIGN AND ARCHITECTURES FOR SIGNAL AND IMAGE PROCESSING (DASIP), págs 1-8, 2011.
- [SAK07] Sakran, N.; Yuffe, M.; Mehalel, M.; Doweck, J.; Knoll, E.; Kovacs, A. *The implementation of the 65nm dual-core 64b Merom processor*. DIGEST OF

- TECHNICAL PAPERS IEEE INTERNATIONAL SOLID-STATE CIRCUITS CONFERENCE (ISSCC'07), 2007, págs. 106-107, 590.
- [SAH11] Saha D.; Sur-Kolay, S.: *SoC: A Real Platform for IP Reuse, IP Infringement, and IP Protection*. VLSI DESIGN HINDAWI PUBLISHING CORPORATION, vol. 2011, 2011.
- [SCH96] Schneider, B.: *Applied Cryptography*. Segunda edición, Wiley&Sons, 1996.
- [SHI10] Shin, M.; Shim, J.; Shim, Y.; Kim, J.: *Investigation of experimental verification for various Power Distribution Network cases through DLL Clock jitter affected by SSN*. 2010 ASIA-PACIFIC SYMPOSIUM ON ELECTROMAGNETIC COMPATIBILITY (APEMC), págs. 325-329, 2010.
- [SOD80] Soderstrand, M.; Vernia, C.: *A high-speed low-cost modulo p_i multiplier with RNS arithmetic application*. PROCEEDINGS OF THE IEEE, vol. 68, págs 529-532, 1980.
- [SOD86] Soderstrand, M.; Jenkins, W. K.; Jullien, G. A.; Taylor, F. J.: *Residue number system arithmetic: modern applications in digital signal processing*. IEEE Press, 1986.
- [SOM11] Soman, S.; Brahme, A.; Venkatraman, R.; Shaikh, R.; Thiyagaraja, S.; Patil, M.: *Ensuring On-Die Power Supply Robustness in High-Performance Designs*. 2011 24TH INTERNATIONAL CONFERENCE ON VLSI DESIGN (VLSI DESIGN), págs. 220-225, 2011.
- [STO99] Stojanovic, V.; Oklobdzija, V.G.: *Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 34, n° 4, págs. 536-548, 1999.

- [SUT99] Sutherland, I.; Sproull, R. F.; Harris, D.: *Logical Effort: Designing Fast CMOS Circuits*. THE MORGAN KAUFMANN SERIES IN COMPUTER ARCHITECTURE AND DESIGN, 1999.
- [SUZ73] Suzuki, Y.; Odagawa, K.; Abe, T.: *Clocked CMOS calculator circuitry*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 8 , n° 6, págs. 462-469, 1973
- [SZA67] Szabo, N. S.; Tanaka, R. I.: *Residue arithmetic and its applications to computer technology*. McGraw-Hill, New York, 1967.
- [TAK08] Takeuchi, K.; Yoshikawa, A.; Komoda, M.; Kotani, K.; Matsushita, H.; Katsuki, Y.; Yamamoto, Y.; Sato, T.: *Clock-Skew Test Module for Exploring Reliable Clock-Distribution Under Process and Global Voltage-Temperature Variations*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 16, n° 11, págs. 1559-1566, noviembre 2008.
- [TAM06] Tam, S.; Leung, J.; Limaye, R.; Choy, S.; Vora, S.; Adachi, M.: *Clock Generation and Distribution of a Dual-Core Xeon® Processor with 16MB L3 Cache*. DIGEST OF TECHNICAL PAPERS SOLID-STATE CIRCUITS CONFERENCE (ISSCC'06), págs. 1512-1521, 2006.
- [TAY81] Taylor, F.: *Large moduli multipliers for signal processing*. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, vol. CAS-28, n° 7, págs 731-736, 1981.
- [TAY84] Taylor, F. J.: *Residue arithmetic: a tutorial with examples*. IEEE COMPUTER 1984, vol. 17, n° 5, págs50-62, 1984.
- [TAY85] Taylor, F.; Papadourakis, A.; Skavantzios, A.; Stouraitis, A.; *A radix-4 FFT using complex RNS arithmetic*. IEEE TRANSACTIONS ON COMPUTERS, vol. C-34, págs. 573-576, 1985.

- [TOU66] Toumazou, C.; Lidgey, F. J.; Haigh, D. G.: *Analog IC design: the current mode approach*. IEE CIRCUITS AND SYSTEMS SERIES 2. 1990
- [TOS04] Tosik, G.; Gaffiot, F.; Lisik, Z.; O'Connor, I.; Tissafi-Drissi F.: *Power dissipation in optical and metallic clock distribution networks in new VLSI technologies*. ELECTRONIC LETTERS, vol. 40, n° 3, febrero 2004.
- [TOS07] Tosik, G.; Abramowicz, F.; Lisik, Z.; Gaffiot, F.; *Clock Skew Analysis in Optical Clock Distribution Network*. 9TH INTERNATIONAL CONFERENCE - THE EXPERIENCE OF DESIGNING AND APPLICATIONS OF CAD SYSTEMS IN MICROELECTRONICS (CADSM '07). págs. 422-425, febrero, 2007.
- [TSA91] Tsay, R. S.; Watson, T. J.; *Exact zero skew*. PROCEEDINGS OF THE 1991 INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, pág. 336-339, 1991.
- [UYE92] Uyemura, J. D.: *Circuit design for CMOS VLSI*. Kluwer Academic Publishers, 1992.
- [VAN78] Vanwormhoudt, M. C.: *Structural properties of complex residue rings applied to number Theoretic Fourier Transforms*. IEEE TRANSACTIONS ON ACOUSTIC, SPEECH AND SIGNAL PROCESSING, vol. 26, págs. 447-455, 1978.
- [VAS96] Vasseghi, N.; Yeager, K.; Sarto, E.; Seddighnewzhad, M: *200MHz superscaler Risc Microprocessor*. IEEE SOLID-STATE CIRCUITS, vol. 31, págs. 1675-1686, noviembre 1996.
- [VEL04] Velenis, D.; Papaefthymiou, M. C.; Friedman, E. G.; *Clock tree layout design for reduced delay uncertainty*. PROCEEDINGS IEEE INTERNATIONAL SOC CONFERENCE, págs 179-180, 2004.

- [VEN07] Venkataraman, G.; Hu, J.; Liu, F. Y.: *Integrated placement and skew optimization for rotary clocking*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 15, n° 2, págs. 149-158, febrero 2007.
- [VET95] Vetterli, M.; Kovacevic, J.: *Wavelets and subband coding*. Prentice Hall, 1995.
- [VON98] Von Kaenel, V.; Aebischer, d.; Van Dongen, R.; Piguet, C.: *A MHz 600 MHz CMOS PLL microrprocessor clock Generator with a 1.2GHz VCO*. PROCEEDINGS OF THE IEEE INTERNATIONAL SOLID-STATE CIRCUIT CONFERENCE, vol. XLI, págs. 396,-397, febrero 1998.
- [VSI08] Virtual Socket Interface Alliance (VSIA): VSIA Homepage 2008. <http://www.vsi.org>.
- [WAL93] Walker, C. D.: *Systolic Modular Multiplier*. IEEE TRANSACTIONS ON COMPUTERS, vol. 42, n° 3, págs. 376-378, 1993.
- [WAR06] Warnock, J.; Wendel, D.; Aipperspach, T.; Behnen, E.; Cordes, R.A.; Dhong, S.H.; Hirairi, K.; Murakami, H.; Onishi, S.; Pham, D.C.; Pille, J.; Posluszny, S.D.; Takahashi, O.; Wen, H.: *Circuit design techniques for a first-generation cell broadband engine processor*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 41, n° 8, págs. 1692-1706, 2006.
- [WRZ96] Wrzyszc, A.; Milford, D.; Dagless, E.: *A New Approach to Fixed Coefficient Inner Product Computation over Finite Rings*. IEEE TRANSACTIONS ON COMPUTERS, vol. 45, n° 12, págs. 1345-1355, 1996.
- [WOO01] Wood, J.; Edwards, T. C.; Lipa, S.: *Rotary Traveling-Wave Oscillator Arrays: A New Clock Technology*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 36, n° 11, págs. 1654-1665, noviembre, 2001.
- [XAN09] Xanthopoulos, T.: *Clocking in modern VLSI Systems*. SPRINGER, 2009.

- [XIL11] Xilinx, Inc.; *Virtex-5 LX FPGA Prototype Platform*. Disponible online en: http://www.xilinx.com/support/documentation/boards_and_kits/ug222.pdf
- [XIL12a] Xilinx, Inc.; *Virtex-7 User Guide*. Disponible online en: http://www.xilinx.com/support/documentation/7_series_user_guides.htm
- [XIL12b] Xilinx, Inc.; *Virtex-5 User Guide*. Disponible online en: <http://www.xilinx.com/support/documentation/virtex-5.htm>
- [XIL12c] Xilinx, Inc.; *Virtex-5 FPGA Data Sheet: DC and Switching Characteristics*. Disponible online en: http://www.xilinx.com/support/documentation/data_sheets/ds202.pdf
- [XU09] Xu, Z.; Shepard, K. L.: *Design and analysis of actively-deskewed resonant clock networks*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 44, nº 2, págs. 558-568, febrero 2009.
- [XU11] Xu, Q.; Li, Y.: *Micro-/Nanopositioning Using Model Predictive Output Integral Discrete Sliding Mode Control*. IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, vol. 59, nº. 2, mayo 2011.
- [YAM01] Yamaguchi, T. J.; Soma, M.; Halter, D.; Raina, R.; Nissen, J.; Ishida, M.: *A method for measuring the cycle-to-cycle period jitter of high-frequency clock signals*. 19th IEEE PROCEEDINGS ON VLSI TEST SYMPOSIUM, págs: 102 –110, 2001.
- [YAM04] Yamaguchi, T. J.; Soma, M.; Nilsen, J.; Halter, D.; Raina, R.; Ishida, M.: *Skew measurement in clock distribution circuits using an analytic signal method*. IEEE TRANSACTIONS ON COMPUTER AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, vol. 23, nº 7, págs. 997-1009 julio 2004.

- [YI01] Yi L.; Meng, Z.; Xianlong, H.; Yic C., Weimin W.: *A clustering-based algorithm for zero-skew clock routing with buffer insertion*. PROCEEDINGS OF 2001 4th INTERNATIONAL CONFERENCE ON ASIC, págs. 183-186, 2001.
- [YOO99] Yoo, J.; Gopalakrishnan, G.; Smith, K. F.: *Timing constraints for high-speed counterflow-clocked pipelining*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol. 7, n° 2, págs. 167-173, junio 1999
- [YOU92] Young, I. A.; Greason, J. K.; Wong, K. L.: *A PLL clock generator with 5 to 110 MHz of lock range for microprocessors*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, vol. 27, n° 11, noviembre 1992.
- [YU05] Yu, Z.; Liu, X.: *Power analysis of rotary clock*. PROCEEDINGS OF THE IEEE COMPUTER SOCIETY ANNUAL SYMPOSIUM ON VLSI, págs. 150-155, mayo 2005.
- [YU07] Yu, Z.; Liu, X.: *Design of Rotary Clock Based Circuits*. 44th ACM/IEEE DESIGN AUTOMATION CONFERENCE 2007 (DAC'07) págs. 43-48, 2007.
- [YUA87] Yuan, J.; Karlsson, I.; Svensson, C.: *A true single-phase-clock dynamic CMOS circuit technique*. IEEE JOURNAL OF SOLID STATE CIRCUITS 1987, vol. SC-22, n°5, págs. 899-901, 1987.
- [YUA89] Yuan J.; Svensson, C.: *High-speed CMOS circuit technique*. IEEE JOURNAL OF SOLID STATE CIRCUITS 1989, vol. 24, n° 1, págs. 62-70, 1989.
- [YUE98] Yue, C.; Wong, S.: *On chip spiral inductors with patterned ground shields for Si-based RF ICs*, IEEE JOURNAL OF SOLID STATE CIRCUITS, vol. 33, n° 5, págs. 743-752, mayo 1998.

- [ZHO91] Zhou, D., Preparata, F. P., Kang, S. M.: *Interconnection delay in very high-speed VLSI*. IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS 1991, vol. 38, n° 7, págs. 779-790, 1991.
- [ZHA99] Zhang, H., George, V.; Rabaey, J. M.: *Low-swing on-chip signaling techniques: Effectiveness and robustness*. PROCEEDINGS INTERNATIONAL SYMPOSIUM LOW-POWER ELECTRONICS AND DESIGN , págs. 145–150, agosto, 1999.
- [ZHA11] Zhao, P.; McNeely, J.; Kuang, W.; Wang, N.; Wang, Z.: *Design of Sequential Elements for Low Power Clocking System*. IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, vol 19, n° 5, págs. 914-918, 2011.
- [ZYU04] Zyuban, V.; Brooks, D.; Viji Srinivasan; Gschwind, M.; Pradip Bose; Strenski, P.N.; Emma, P.G.: *Integrated analysis of power and performance for pipelined microprocessors*. IEEE TRANSACTIONS ON COMPUTERS, vol. 53, n° 8, págs. 1004-1016, agosto 2004.

