

TESIS DOCTORAL

Circuitos Electrónicos
para el Procesamiento Paralelo
de Señales Estocásticas.

Juan García Ortega

Sevilla, 1995

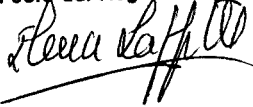
T. 131

Depositado en

de la
de esta Universidad desde el día
hasta el día

Que se deposita esta Tesis Doctoral
al folio 101 número 269 de mayo
correspondiente a
Sevilla, 27 JUN 1995

El Jefe del Negociado de Tesis,

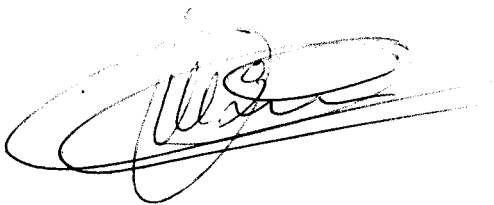


Circuitos Electrónicos para el Procesamiento Paralelo de Señales Estocásticas.

autor: Juan García Ortega

director: José Manuel Quero Reboul

Junio 1995



José M. Quero Reboul
Director de Tesis



JUAN GARCIA ORTEGA
Doctorando

Contenido

1	Agradecimientos.	6
2	Resumen de la tesis doctoral	8
3	Estudios y consideraciones previas	12
3.1	Introducción.	12
3.2	Generación de trenes de pulsos.	13
3.2.1	Generación de secuencias pseudoaleatorias.	16
3.2.2	Otras técnicas.	18
3.3	Multiplicación de trenes de pulsos estocásticos.	19
3.4	Suma de trenes de pulsos estocásticos.	22
3.4.1	Suma por medio de puertas OR.	22
3.4.2	Suma a base de contadores reversibles.	24

3.4.3	Suma empleando una transformación exponencial.	24
3.5	Realizaciones físicas estocásticas.	27
3.6	Arquitectura estocástica totalmente paralela: Aplicación a la reali- zación física de redes de neuronas.	31
3.6.1	Estructura de una red perceptrón	32
3.6.2	Realización física	34
3.7	Limitaciones del procesamiento estocástico paralelo.	49
4	Suma estocástica.	52
4.1	Introducción.	52
4.2	Suma de trenes estocásticos.	52
4.3	Extensión de la arquitectura de suma.	54
4.3.1	Deducción de la función de transferencia sigmoideal.	57
4.3.2	Arquitectura completa de la estructura de suma.	59
4.4	Aplicación a la realización de filtros de respuesta impulsional finita. . .	60
5	Generación de trenes de pulsos estocásticos.	67
5.1	Introducción	67
5.2	Generación de señales aleatorias.	69

5.3	Celdas generadoras de trenes de pulsos. Realización física	77
5.3.1	Bloques osciladores	78
5.3.2	Bloque muestreador.	84
5.4	Aplicación a la generación de números aleatorios	92
6	Conversión de señales basada en la lógica estocástica.	96
6.1	Introducción	96
6.2	Estudio estadístico de las señales estocásticas.	98
6.3	Conversión estocástica analógica.	100
6.4	Conversión analógica estocástica.	102
6.5	Conversión digital analógica y analógica digital basada en lógica estocástica.	108
6.5.1	Ventajas de la conversión a pulsos estocásticos.	108
6.5.2	Estructura completa de los convertidores digital analógico y analógico digital.	110
6.5.3	Aplicación en un circuito integrado industrial.	112
7	Procesadores programables estocásticos.	129
8	Conclusiones.	132

Lista de Figuras

3.1	Generación digital de trenes de pulsos.	14
3.2	Generador de números pseudoaleatorios.	16
3.3	Correlación entre la secuencia de números pseudoaleatorios y la secuencia de números generada p ciclos de reloj más tarde. Registro de 9 bits.	17
3.4	Multiplicación de trenes de pulsos.	20
3.5	Suma basada en puertas OR.	23
3.6	Suma basada en una compresión exponencial.	25
3.7	Aproximaciones de la transformación.	27
3.8	Red de Hopfield estocástica.	28
3.9	Esquema general de un Perceptrón.	32
3.10	Operaciones de una Neurona.	33
3.11	Esquema de bloques de una Neurona.	36

3.12	Error normalizado en la transformación exponencial, para primero, segundo y tercer orden de aproximación.	38
3.13	Suma comprimida de términos.	39
3.14	Estructura expansora de capa oculta.	40
3.15	Ciclo de números recorrido por un LFSR.	42
3.16	Generación de subsecuencias aleatorias.	43
3.17	Estructura de puertas xores.	44
3.18	Encadenamiento de números aleatorios en dos circuitos integrados. . .	45
3.19	Distribución del ciclo aleatorio en ambos integrados.	46
3.20	Realización ASIC de un Perceptrón con arquitectura estocástica . . .	47
3.21	Perceptrón realizado con dos circuitos integrados.	47
3.22	Comparacion entre la respuesta real y teórica de la neurona de salida.	48
3.23	Arquitectura estocástica de una neurona.	49
4.1	Arquitectura estocástica.	54
4.2	Circuito totalizador.	55
4.3	Función sigmoideal.	56
4.4	Proceso de markov.	58
4.5	Arquitectura de suma.	60

4.6	Filtro estocástico.	62
4.7	Respuesta del Filtro.	63
4.8	Respuesta del Filtro.	64
4.9	Respuesta del Filtro.	65
4.10	Filtro digital sobre una FPGA y generadores de números aleatorios. .	66
5.1	Generador aleatorio tipo A.	71
5.2	Generador aleatorio tipo B.	72
5.3	Generador aleatorio tipo C.	72
5.4	Representación del muestreo de una onda periódica.	73
5.5	Resultado del muestreo de una señal periódica.	74
5.6	Muestreo de una señal periódica con un ruido superpuesto.	75
5.7	Autocorrelación en función del periodo de muestreo y porcentaje de ruido.	76
5.8	Celda generadora de trenes de pulsos.	78
5.9	Mecanismo de oscilación a base de inversores CMOS.	79
5.10	Control del ancho de pulsos en todo el intervalo.	80
5.11	Curva de regulación de T_{off}/T en función de V_{N1}	81
5.12	Peso fijado por divisor de tensiones.	81

5.13	Oscilador con $T_{on}/T_{off} = 2.58$	82
5.14	Control del ancho de pulsos en la mitad del intervalo.	83
5.15	Oscilador con $T_{on}/T_{off} = 0.57$	83
5.16	Comparación de área entre ambos diseños.	86
5.17	Layout del oscilador tipo 1.	88
5.18	Layout del oscilador tipo 2.	88
5.19	Layout del oscilador tipo 3.	89
5.20	Layout del oscilador tipo 4.	89
5.21	Layout del circuito total.	90
5.22	Microfotografía del circuito integrado generador de trenes de pulsos estocásticos.	91
5.23	Prototipo de convertidor analógico digital.	91
5.24	Comparación entre las autocorrelaciones de diferentes trenes de pulsos.	93
6.1	Conexión entre sistemas analógicos y estocásticos.	97
6.2	Espectro de un tren de pulsos estocásticos.	100
6.3	Arquitectura propuesta para la conversión analógica estocástica. . . .	103
6.4	Esquema de una celda aleatoria.	104
6.5	Medida en osciloscopio de la incertidumbre acumulada en la oscilación.	105

6.6	Señal aleatoria de entrada al comparador.	106
6.7	Salidas correspondientes a dos niveles analógicos.	107
6.8	Respuesta de la red neuronal.	107
6.9	Desarrollo en serie de Fourier de una onda PWM.	109
6.10	Convertidor digital-analógico estocástico.	111
6.11	Convertidor analógico-digital estocástico.	111
6.12	Evolución del convertidor analógico-digital por aproximaciones sucesivas.	113
7.1	Dispositivo programable estocástico.	130

Capítulo 1

Agradecimientos.

Quiero expresar mi agradecimiento a todas aquellas personas que me han prestado su apoyo, y que han hecho posible la culminación de esta tesis. En primer lugar a mi director de tesis, D. José Manuel Quero Reboul, que ha mostrado durante todo este tiempo un gran interés en mi labor investigadora, así como una total entrega para ayudar a solucionar cualquiera de los problemas que fueron apareciendo a lo largo de estos años. Su gran capacidad para transmitir ideas y la gran cantidad de horas que me ha dedicado han sido fundamentales en el desarrollo de esta tesis, sobre todo en los momentos mas áridos como son los inicios de ésta.

A mi compañero, D. Carlos Janer Jimenez, persona con una gran experiencia investigadora en el campo del cálculo estocástico, quiero agradecele la gran cantidad de consejos que tanto me han ayudado. A mi esposa, M^a Carmen, cuyo apoyo moral y material ha hecho este trabajo mucho mas llevadero. Al director del Grupo de Tecnología Electrónica, D. Leopoldo García Franquelo, por su ayuda mostrada durante la realización de esta tesis.

A todos mis compañeros del Grupo de Tecnología Electrónica, que en la medida de sus posibilidades se han prestado de forma desinteresada para descargarme de otras labores en momentos críticos. El clima de compañerismo que ellos han creado ha hecho más cómoda mi labor investigadora.

Por último, a los alumnos Jerónimo, Manolo, Alfonso y Dani, que han mostrado un especial interés por el desarrollo de este trabajo.

Sevilla, 23 de Julio de 1.995 .

Capítulo 2

Resumen de la tesis doctoral

Varias de las líneas de trabajo que se ha desarrollado durante estos últimos años en el Grupo de Tecnología Electrónica guardan una estrecha relación con el procesamiento en paralelo de la información. Así, por ejemplo, pueden citarse las redes neuronales artificiales y la lógica borrosa. La potencia del procesamiento neuronal reside en el elevado número de unidades de procesamiento (neuronas) y su alta interconectividad. Otro tanto puede decirse de los controladores basados en lógica borrosa. De ahí el interés de desarrollar arquitecturas que por ser eficientes en términos de área de silicio consumida, permitan evaluar **simultáneamente** un elevado número de operaciones matemáticas. Las operaciones que hay que evaluar son, esencialmente, multiplicaciones y sumas.

En los sistemas estocásticos, la información se representa mediante un conjunto de trenes de pulsos aleatorios síncronos. El valor de una variable real perteneciente al intervalo $[0, 1]$ se codifica de una manera muy especial; es la **probabilidad** de que, en un ciclo de reloj dado, el tren de pulsos asociado a la señal tome el valor lógico alto. Los distintos trenes estocásticos deben generarse de forma que sean todos estadísticamente independientes unos de otros. Supóngase ahora que se dispone de n

trenes estocásticos que entran en una puerta AND de n entradas. Puesto que las variables aleatorias que representan el nivel lógico de las entradas son estadísticamente independientes entre sí, resulta que la variable aleatoria que representa el nivel lógico a la salida de la puerta AND tiene una probabilidad de estar a nivel alto igual al producto de las probabilidades de que cada una de las entradas esté a nivel lógico alto. En la operación producto estocástico, incluir un nuevo término en la multiplicación se reduce a añadir una entrada en la puerta AND, lo cual sólo supone añadir dos transistores más.

En varios trabajos hechos por miembros de este Departamento se propuso efectuar una compresión exponencial sobre los términos sumandos, de forma que se establece una aplicación entre el intervalo $[0, \infty)$ y el intervalo $(0, 1]$. De esta forma puede representarse la suma por medio de un par de trenes estocásticos que representan la suma comprimida de los términos positivos y negativos. Estos trabajos estaban encaminados a realización microelectrónica de redes neuronales con función de transferencia signo por lo que no se evaluaba la suma, sino que, para determinar el signo de ésta, se comprobaba cuál de los dos trenes era más denso.

Los inconvenientes que presentaba esta arquitectura y los sistemas de procesamiento estocástico en general son tres: En primer lugar no existe una arquitectura paralela capaz de evaluar la suma, en segundo lugar los circuitos encargados de la generación de trenes de pulsos estocásticos que aparecen en la literatura consumen mucha área. Por último las señales que existen en el mundo real son analógicas, por lo que es preciso concebir circuitos de conversión analógicos-estocásticos y viceversa.

En esta Tesis Doctoral se ha empleado la técnica de compresión anteriormente mencionada. Sin embargo resultaba evidente la necesidad de encontrar un forma de combinar los trenes de pulsos representativos de la suma de términos positivos y negativos de forma que pudiera obtenerse la suma total. Se ha demostrado que la **suma** puede obtenerse de un modo muy simple: un circuito combinacional de-

tecta cuándo el tren de pulsos representativo de la suma comprimida de los términos positivos está a nivel lógico alto y el tren de pulsos representativo de la suma comprimida de los términos negativos está a nivel lógico bajo. Cuando esto ocurre el circuito genera una señal de puesta a cero de un biestable. Si se produce la situación inversa se genera una señal de puesta a uno del biestable. El estado del biestable se describe mediante un proceso de Markov, cuyo estado promedio (interpretando el estado 1 como +1 y el estado 0 como -1) viene dado por la función sigmoideal, que en un entorno del origen puede aproximarse por medio de una línea recta. A modo de ejemplo, la arquitectura de suma descrita se ha empleado para la realización física de filtros digitales de respuesta impulsional finita. En esta aplicación todas las operaciones involucradas en el cálculo de la respuesta del filtro se efectúan en paralelo.

La manera de generar trenes de pulsos estocásticos que se describe en la bibliografía consiste en comparar digitalmente la codificación del número dado con una secuencia de números pseudoaleatorios generados por un registro de desplazamiento realimentado (LFSR). Esta solución presenta diversas desventajas. Por una parte el LFSR genera **secuencias cíclicas** de números **pseudoaleatorios**. Esto presenta inconvenientes cuando el número de pulsos de la señal estocástica se hace grande en comparación con el tamaño del ciclo. Por otro lado el área de silicio consumida en el comparador digital y en el generados de secuencias pseudoaleatorias es importante. Lo que se ha propuesto en esta Tesis es generar los trenes de pulsos de una manera distinta, teniendo en cuenta que es necesario garantizar la incorrelación de estos. Para ello se han diseñado **celdas generadoras de trenes de pulsos estocásticos**, con una probabilidad dada, basadas en osciladores de alta frecuencia, cuyas relaciones $\frac{T_{on}}{T_{off}}$ se puede ajustar por medio de una tensión de control. Debido a la presencia de ruidos (de origen térmico, por ejemplo) el periodo del oscilador varía aleatoriamente a lo largo del tiempo. Si este oscilador se muestrea con un intervalo de tiempo lo suficientemente grande de forma que la incertidumbre abar-

que a un ciclo completo, la señal muestreada será totalmente aleatoria, siendo $\frac{T_{on}}{T_{off}}$ la probabilidad de muestrear un 1 lógico. Se ha construido un circuito integrado de prueba en el que se han diseñado varios osciladores de este tipo. Este circuito se ha utilizado en una aplicación en la cual fué necesario generar una secuencia de números aleatorios.

Para que los sistemas de procesamiento estocástico puedan funcionar en un entorno de señales analógicas se han estudiado teóricamente y diseñado convertidores analógico-estocásticos y estocástico-analógicos. El convertidor analógico-estocástico es una sencilla red R-C que filtra la componente de continua del espectro. La señal filtrada tiene superpuesta una componente de ruido que proviene de la parte continua del espectro que ha sido integrada por el filtro. El filtro debe dimensionarse de forma que las fluctuaciones debidas a esta componente sean inferiores a la resolución requerida en la conversión. Este estudio ha servido como base para el diseño dos convertidores, uno digital-analógico y otro analógico-digital que emplean una codificación intermedia en forma de señales estocásticas. Estos convertidores se han implementado en un circuito integrado que se ha diseñado para una empresa española.

Dado que los resultados expuestos permiten la realización de sistemas electrónicos de cálculo estocásticos, la tesis finaliza con la presentación de una **arquitectura estocástica programable**, con un núcleo de celdas aleatorias combinables, funciones aritméticas estocásticas de sumas y productos, y un conjunto de convertidores que posibilitan las entradas y salidas de señales analógicas ó digitales. Con esta arquitectura se pueden sintetizar sistemas con una carga computacional elevada, como son los sistemas neuroborrosos, filtros y controladores digitales, ...

Capítulo 3

Estudios y consideraciones previas

3.1 Introducción.

En los sistemas estocásticos de procesamiento, [1], las variables se representan por medio de trenes de pulsos aleatorios. Estos trenes de pulsos se sincronizan por medio de un reloj base. En un determinado ciclo de reloj, cada uno de los trenes puede tomar sólo dos valores distintos: un nivel lógico alto o bien un nivel lógico bajo; además tomarán uno de estos dos valores de forma aleatoria, aunque, en general, la probabilidad de tomar el nivel lógico alto será distinta de la probabilidad de tomar el nivel lógico bajo. El valor de cada variable es, precisamente, la **probabilidad** de que, en un ciclo de reloj dado, el tren de pulsos asociado tome el valor lógico alto. Si se considera una sucesión infinita de ciclos de reloj, resulta que cada variable de la red tiene asociada una sucesión de niveles lógicos altos y bajos (tren de pulsos). Evidentemente, si en un horizonte temporal grande se estima la densidad de estos pulsos, el valor obtenido coincidirá con la probabilidad antes mencionada.

Codificar la información mediante pulsos estocásticos permite evaluar multiplicaciones y sumas de un número grande de operandos de una manera muy sencilla, [2]. Los circuitos encargados de evaluar estas operaciones son muy simples, por lo que el cálculo estocástico se puede utilizar en aplicaciones en las que hay que efectuar un gran número de multiplicaciones y sumas en paralelo. Algunos ejemplos de aplicación de los sistemas estocásticos son realizaciones físicas de redes neuronales artificiales, controladores basados en lógica difusa, síntesis de filtros digitales, etc.

En realidad la operación que se calcula fácilmente es el producto (una puerta AND de n entradas calcula el producto de n términos). En esto reside una de las mayores ventajas del cálculo estocástico: una operación que es difícil de evaluar digitalmente (los multiplicadores de punto fijo o punto flotante son bastante complejos) se calcula de una manera extremadamente simple en los sistemas estocásticos. La operación suma es más difícil de calcular. Téngase en cuenta que el producto de un conjunto de números que toman valores comprendidos entre cero y uno es también un número comprendido entre cero y uno, mientras que la suma no tiene por qué tomar valores en ese intervalo, por lo que no podría, en principio, representarse por medio de una probabilidad y, en consecuencia, por un tren de pulsos estocásticos. Diversos autores han propuesto estructuras distintas para evaluar resolver este problema.

3.2 Generación de trenes de pulsos.

Las realizaciones físicas estocásticas síncronas que se han descrito hasta ahora, utilizan una tecnología microelectrónica estrictamente digital, de ahí que los circuitos encargados de generar los trenes de pulsos estocásticos sean circuitos digitales. El procedimiento es el siguiente, [4]:

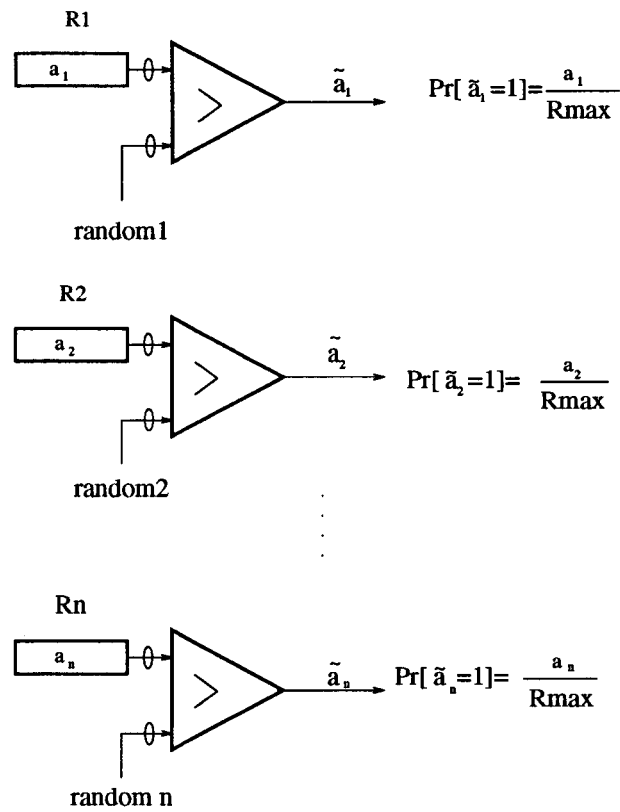


Figura 3.1: Generación digital de trenes de pulsos.

Se parte de una codificación digital del número que quiere convertirse en secuencia de pulsos estocásticos. Esta codificación digital puede estar guardada en un registro, o bien, ser un conjunto de niveles de tensión altos y bajos. (Esta última estrategia se seguiría si el término que fuese a codificarse fuera constante y no hubiese necesidad de reprogramarlo. La forma de realizarlo físicamente es evidente: no habría más que hacer conexiones a la fuente de alimentación del circuito integrado, nivel lógico alto, y a masa, nivel lógico bajo). Supóngase, además, que se dispone de un circuito capaz de generar números aleatorios que, codificados digitalmente, se distribuyen uniformemente entre cero y un número máximo. Se supondrá que las secuencias de números generados son estadísticamente independientes las unas respecto a las otras. Si, en un determinado ciclo de reloj, se compara la codificación digital del número que quiere convertirse en secuencia de pulsos estocásticos, con el número aleatorio generado en dicho ciclo de reloj, resulta que, a la salida del comparador digital, se pueden obtener sólo dos niveles de tensión: un nivel lógico bajo, en cuyo caso el número aleatorio generado es mayor o igual que el número a convertir, o un nivel lógico alto, en cuyo caso el número aleatorio que se ha generado ha resultado ser menor que el número que cuya secuencia aleatoria de pulsos se quiere obtener. Evidentemente, dado que la distribución de los números aleatorios es uniforme, la probabilidad de que a la salida del comparador digital haya un nivel lógico alto es proporcional al número cuya codificación por pulsos se está obteniendo. Si se considera una secuencia infinita de ciclos de reloj, se obtiene el tren de pulsos aleatorios. En la figura (3.1) se indica cómo se generarían los trenes de pulsos estocásticos asociados a los números guardados en los registros R_1 , R_2 , ..., R_n .

En la bibliografía se describen una serie de técnicas de generación de números aleatorios que se pasa ahora a enumerar:

3.2.1 Generación de secuencias pseudoaleatorias.

El circuito generador de números pseudoaleatorios consiste, básicamente, en un registro de desplazamiento cuyo bit más significativo se realimenta, a partir de algunos de los restantes bits, mediante puertas XOR, [5]. Este circuito se representa en la figura (3.2). Si la realimentación se hace de una manera adecuada, se consigue que, para cualquier semilla inicial distinta de 0, el sistema recorra un ciclo de números pseudoaleatorios de longitud máxima que, para un registro que conste de n bits, es de $2^n - 1$. A partir de entonces la secuencia de números aleatorios vuelve a repetirse.

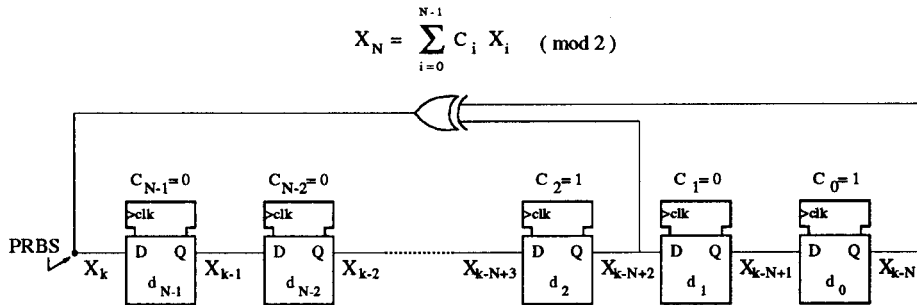


Figura 3.2: Generador de números pseudoaleatorios.

Si se tiene en cuenta cómo se calcula, a partir del número aleatorio generado en el ciclo de reloj t , el número aleatorio del ciclo de reloj $t+1$, se comprende bien que la correlación que existe entre ambos no es nula. Esto es así puesto que, al desplazar el registro hacia la derecha, se está dividiendo el número que había en el instante t por dos y, al obtener pseudoaleatoriamente el bit más significativo se está sumando, con una probabilidad 0.5, o bien 0 (bit 0 de realimentación) o bien 2^{n-1} (bit más significativo igual a 1). Por tanto parece claro que no resulta conveniente emplear una estructura de "pipelining" para hacer llegar los números aleatorios a los distintos generadores de secuencias de pulsos estocásticas, ya que entonces éstas estarían correladas entre sí, [6]. En la figura (3.3) se representa la función de autocorrelación de la secuencia de números pseudoaleatorio generada con un registro de 9 bits, donde se observa que para es preciso considerar para la

seuencia un desplazamiento al menos igual al tamaño del registro para tener una autocorrelación pequeña.

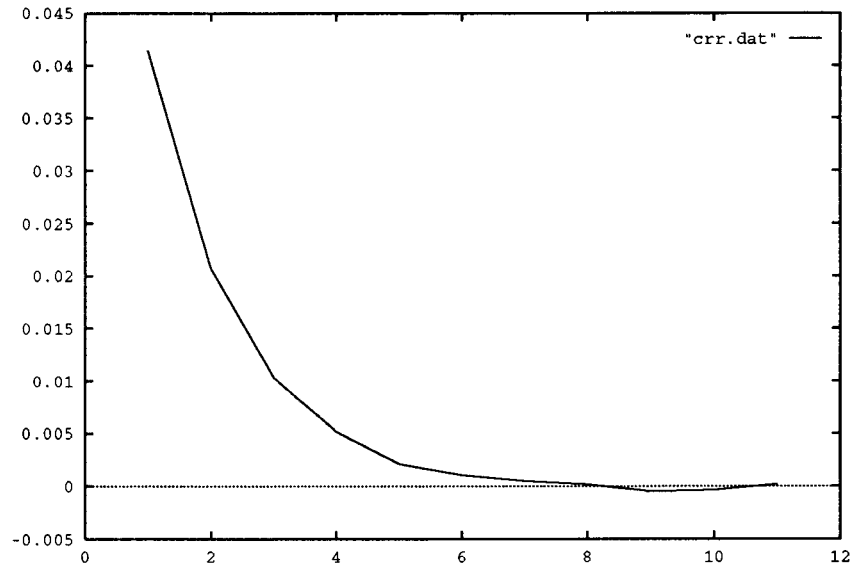


Figura 3.3: Correlación entre la secuencia de números pseudoaleatorios y la secuencia de números generada p ciclos de reloj más tarde. Registro de 9 bits.

Para resolver este problema lo que se puede hacer es, mediante un bloque lógico hecho a base de puertas XOR, calcular, a partir del número aleatorio generado, el que ocupa una posición en la cadena lo suficientemente alejada de forma que ambos estén poco correlados, [6].

Esta forma de generar números aleatorios está muy extendida en las realizaciones físicas de sistemas lógicos estocásticos debido a su sencillez. Los inconvenientes son, sin embargo, evidentes. No se generan verdaderos número aleatorios (hay una fuerte correlación temporal en la secuencia de números) y además se generan cíclicamente una serie de números pseudoaleatorios que empiezan a repetirse una vez que el ciclo se ha completado.

3.2.2 Otras técnicas.

La generación de bits aleatorios puede conseguirse forzando a que un latch entre en estado metastable, [7]. Si el circuito se ha diseñado convenientemente, la evolución desde este estado a cualquiera de los dos estados estables es equiprobable, es decir, la probabilidad de tener un uno o un cero lógico es 0.5. Las restricciones que hay que imponer al diseño del latch son bastante exigentes. El circuito debe ser totalmente simétrico, hay que forzar exactamente el estado metastable y por último las cargas que se coloquen en las dos salidas del latch deben ser idénticas. Si cualquiera de estas condiciones no se verifica, la evolución desde el estado metastable hasta uno de los estados estables tendrá una probabilidad de ocurrir mayor. El mayor inconveniente de este tipo de técnicas son las asimetrías introducidas por la dispersión inherentes al proceso de fabricación.

Pueden conseguirse bits aleatorios a partir de señales analógicas de tipo aleatorio. Esta señal puede ser, por ejemplo, el ruido asociado a la corriente inversa de saturación de una unión p-n, [8]. Normalmente estas señales tienen que ser amplificadas ya que son muy débiles y después compararlas con una referencia para obtener bits aleatorios. También se describen en la bibliografía generadores de señales aleatorias basadas en la evolución de sistemas caóticos, [9], en particular osciladores cuya fase evoluciona caóticamente. Los sistemas caóticos son muy sensibles a variaciones en las condiciones iniciales, de forma que pequeñas perturbaciones en el estado del sistema provocan, al cabo de un tiempo, estados muy distintos. Por consiguiente si el muestreo del sistema caótico se hace a intervalos de tiempo mayores que este tiempo crítico se tiene que el estado del sistema es impredecible.

3.3 Multiplicación de trenes de pulsos estocásticos.

Si en el proceso de creación de las secuencias de pulsos se emplean generadores de números aleatorios independientes entre sí, resulta evidente que las sucesiones de números creadas por éstos estarán mutuamente incorreladas. Este hecho garantiza que los trenes de pulsos serán estadísticamente independientes unos de otros. Supóngase ahora que se dispone de n trenes estocásticos que entran en una puerta AND de n entradas, tal como se muestra en la figura (3.4). ¿Cuál es el valor de la probabilidad de que, en un determinado ciclo de reloj, a la salida de la puerta lógica, haya un nivel lógico alto? Puesto que las variables aleatorias que representan el nivel lógico de las entradas, en un determinado ciclo de reloj, son estadísticamente independientes entre sí, resulta que la variable aleatoria que representa el nivel lógico de la salida de la puerta AND, en el ciclo de reloj considerado, tiene una probabilidad de estar a nivel lógico alto igual al producto de las probabilidades de que cada una de las entradas esté a nivel lógico alto.

Es importante destacar el hecho de que para que este resultado sea cierto, lo único que se exige es que, en cada ciclo de reloj, los números aleatorios sean todos independientes unos de otros. No se exige que los números aleatorios generados en dos ciclos de reloj distintos sean independientes unos de otros. Si esto se verifica, se puede deducir, además, que el tren de pulsos generado a la salida de la puerta AND está temporalmente incorrelado, es decir, que el hecho de que en un determinado ciclo de reloj haya un nivel lógico alto a la salida de la puerta AND no condiciona en lo más mínimo el estado de la salida de la puerta en el siguiente ciclo de reloj, o en cualquier otro. Es decir, el ruido que tiene superpuesto la densidad de pulsos que hay a la salida de la puerta es espectralmente blanco.

Si lo anterior no se verifica, piénsese por ejemplo en una arquitectura en la que haya un sólo generador de números aleatorios y éstos se pasen de unos registros a

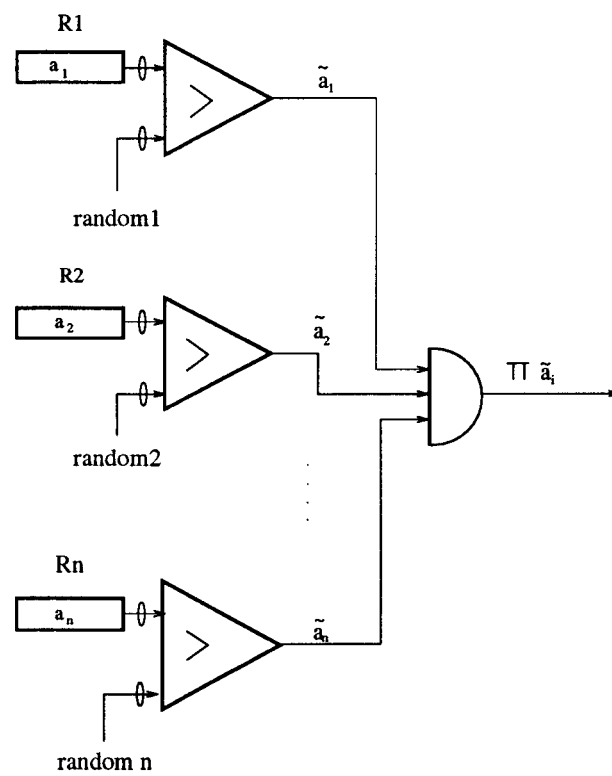


Figura 3.4: Multiplicación de trenes de pulsos.

otros por "pipelining", la salida de la puerta lógica AND tendrá superpuesto un ruido con un cierto comportamiento dinámico. Esto es así porque la secuencia de números aleatorios empleada para generar el i -ésimo tren de pulsos, es la misma que la empleada para generar el tren de pulsos $i-1$, salvo que se encuentran decaladas temporalmente un ciclo de reloj la una con respecto a la otra. Por tanto el hecho de que, en un determinado ciclo de reloj, a la salida de la puerta AND haya un pulso a nivel alto, influye sobre la probabilidad de que, en ciclos de reloj posteriores, haya pulsos a nivel alto. Por tanto no se tendrá un tren de pulsos temporalmente incorrelado, aún cuando la densidad del tren de pulsos coincide con el producto de las densidades de los trenes de entrada.

Existe otra estructura muy sencilla, [10], que permite calcular el producto de **dos** secuencias de trenes de pulsos estocásticos. El circuito que calcula el producto es una puerta XOR. Considérense dos términos v_1 y v_2 comprendidos en el intervalo $[-1,1]$. Puesto que no existen densidades de pulsos negativas, se codifican estocásticamente las cantidades p_1 y p_2 definidas como:

$$p_1 = \frac{v_1 + 1}{2}$$
$$p_2 = \frac{v_2 + 1}{2}$$

teniendo en cuenta que al efectuar la operación XOR de dos trenes de pulsos de densidades p_1 y p_2 se tiene a la salida otro tren de pulsos de densidad

$$p_0 = p_1(1 - p_2) + p_2(1 - p_1)$$

Poniendo esta expresión en función de p_1 y p_2 se tiene:

$$p_0 = \frac{1}{2} - \frac{1}{2}v_1v_2$$

Para interpretar este tren de pulsos se aplica a este término la transformación inversa, es decir, $v_0 = 2p_0 - 1$, con lo que queda:

$$v_0 = -v_1v_2$$

La ventaja que tiene esta estructura frente a la anterior es que el signo de los términos a multiplicar viene implícitamente codificado en el tren de pulsos. Sin embargo presentan el inconveniente de que sólo se pueden multiplicar dos términos.

3.4 Suma de trenes de pulsos estocásticos.

Para evaluar la suma de un conjunto de trenes estocásticos, se describen en la bibliografía tres estructuras distintas: la primera está basada en el empleo de puertas OR, [11], la segunda utiliza contadores reversibles, [12], mientras que en la tercera las operaciones de suma se transforman en productos haciendo uso de una transformación exponencial, [15].

3.4.1 Suma por medio de puertas OR.

Supóngase que se tienen dos secuencias de pulsos aleatorios que quieren sumarse. Estas dos secuencias son las entradas de una puerta OR, tal como se muestra en la figura (3.5). A la salida de dicha puerta tendremos otra secuencia de pulsos

aleatorios. En un determinado instante, es decir, en un determinado ciclo de reloj, la probabilidad de que a la salida de la puerta OR haya un nivel lógico alto es inferior a la suma de las probabilidades de que las entradas estén a nivel lógico alto. Téngase en cuenta que, al contrario de lo que ocurre con la operación producto, la suma de dos términos comprendidos entre cero y uno no está, generalmente, acotada entre cero y uno. De hecho cuando, en un instante determinado, varios pulsos a la entrada estén a nivel lógico alto, a la salida sólo tendremos un pulso a nivel lógico alto. Además, que la no linealidad **no** es función exclusivamente del resultado de la suma de las entradas, sino que depende, además, de cómo se estén sumando. Así, por ejemplo, si consideramos dos secuencias de pulsos que entran en una puerta OR, una de densidad cero y otra de densidad uno, a la salida tendremos un tren de pulsos de densidad uno. Sin embargo, si consideramos dos entradas de densidades 0.5, a la salida tendremos un tren de pulsos cuya densidad será inferior a 1, ya que, necesariamente, se produce solapamiento de los pulsos de entrada.

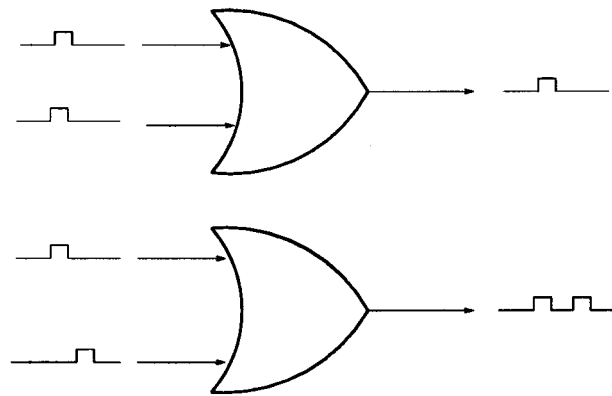


Figura 3.5: Suma basada en puertas OR.

3.4.2 Suma a base de contadores reversibles.

Una segunda estrategia para efectuar la suma estocástica es la de emplear contadores reversibles. Tiene el inconveniente de que la suma no se evalúa en paralelo, sino en serie. Se fija un horizonte temporal durante el cual se va a evaluar la influencia que tiene un sumando sobre el resultado de la suma. Si el sumando es positivo, cada vez que haya un pulso a nivel lógico alto, se incrementa el contenido de un contador reversible. Si el sumando es negativo, el contenido del contador se decrementa en una unidad. De esta forma, una vez se han tenido en cuenta todos los sumandos, en el contador hay un número representativo de la suma. Esta técnica se ha empleado en el campo de las realizaciones físicas de redes de neuronas, [12], [16].

3.4.3 Suma empleando una transformación exponencial.

El problema que presenta la operación suma estocástica frente a la operación producto estocástico es que, mientras que el producto de dos términos cuyos valores están comprendidos entre 0 y 1 es otro término también comprendido entre 0 y 1, en la suma esto no es necesariamente así. Por consiguiente, un tren de pulsos estocásticos puede representar el producto de dos trenes estocásticos, pero, en principio, no su suma.

Puesto que en los sistemas estocásticos los productos se calculan fácilmente, resulta interesante aplicar a los términos que se quieren sumar algún tipo de transformación que permita transformar las sumas en productos. Esta transformación es la exponencial. Considérese la siguiente relación:

$$\prod_{i=1}^{i=n} e^{-x_i} = e^{\sum_{i=1}^{i=n} -x_i} \quad (3.1)$$

donde los términos x_i se suponen positivos. Esto quiere decir que si se calcula estocásticamente la función exponencial de los términos que se van a sumar (cambiándoles antes el signo), y se multiplican, a la salida de la puerta producto se obtiene un tren de pulsos representativo de la suma. Las transformaciones exponenciales se pueden calcular fácilmente y esta estrategia se utiliza para calcular sumas. Los términos positivos y negativos se suman separadamente y después se combinan los dos trenes de pulsos resultantes para calcular la suma total. Hasta ahora esta estrategia se ha empleado en la realización física de redes neuronales artificiales con función de transferencia signo, [3]. En estas aplicaciones sólo se necesita conocer el signo de la suma de términos. Por ello, lo que se hace es comprobar cuál de los dos trenes de pulsos tiene una densidad mayor.

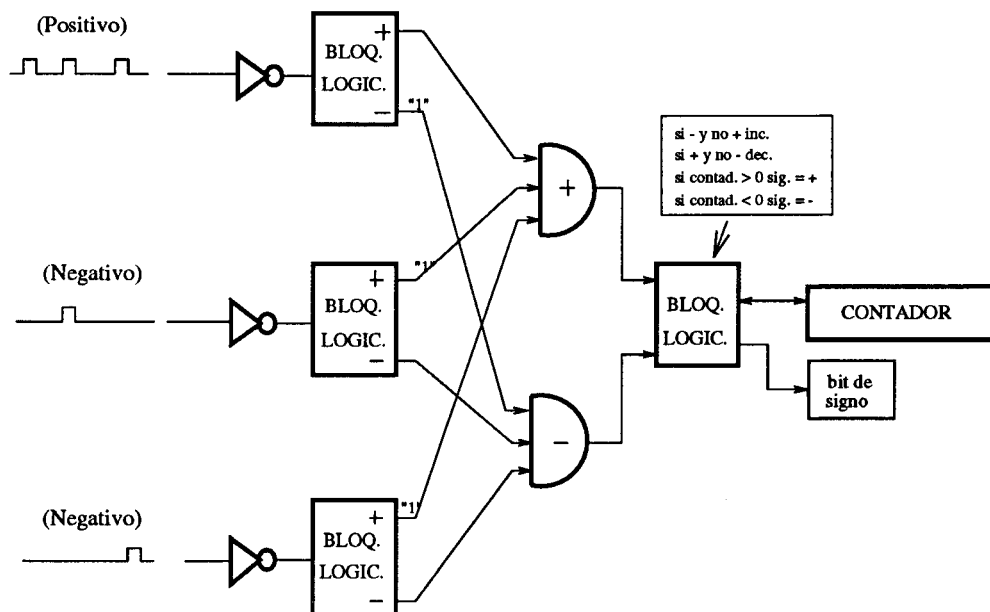


Figura 3.6: Suma basada en una compresión exponencial.

La arquitectura completa se representa en la figura (3.6). Se tienen como entradas un conjunto de trenes de pulsos estocásticos síncronos. Cada secuencia de pulsos

trae implícita un signo que se supone codificado por medio de un bit de signo. La primera operación que efectúa el circuito sumador es invertir (aproximación lineal de la suma $e^{-x} \simeq 1 - x$) y desdoblar los n trenes de pulsos de entrada que van a sumarse en $2n$ secuencias de pulsos. Estas $2n$ secuencias son de dos tipos distintos. Hay n trenes positivos y n trenes negativos. Los circuitos lógicos que hay a la entrada no son más que multiplexores de una entrada y dos salidas que hacen que las secuencias de entrada ataquen a una serie de puertas inversoras, para después entrar en una u otra de las dos puertas AND representadas en la figura, según el bit de signo asociado al tren. La salida del multiplexor por la que no sale el tren de entrada se encuentra en un nivel lógico alto, correspondiente a la transformación exponencial del valor cero.

La segunda estructura que aparece en la figura son dos puertas AND, una asociada a los pulsos positivos y otra a los negativos. En un determinado ciclo de reloj la salida de la puerta AND positiva estará a nivel lógico bajo si al menos hay una entrada a nivel lógico alto. Una afirmación análoga a ésta puede hacerse para la puerta AND negativa.

El tercer bloque que aparece en la figura tiene como objetivo evaluar el signo de la suma. Para ello se comprueba si el número que hay guardado en el contador reversible de la figura es positivo o negativo, con lo cual decide si hay que cambiar o no el bit de signo. Simultáneamente comprueba si se produce alguno de los siguientes sucesos: La presencia de un nivel alto a la salida de la puerta AND positiva y de un nivel bajo a la salida de la puerta AND negativa. El segundo suceso es la presencia de un nivel bajo a la salida de la puerta AND positiva y la presencia de un nivel alto a la salida de la puerta AND negativa. En caso de producirse la primera circunstancia, se genera una señal que decrementa en una unidad el contenido del contador reversible y si se produce la segunda circunstancia se genera otra señal que incrementa el contenido del contador reversible.

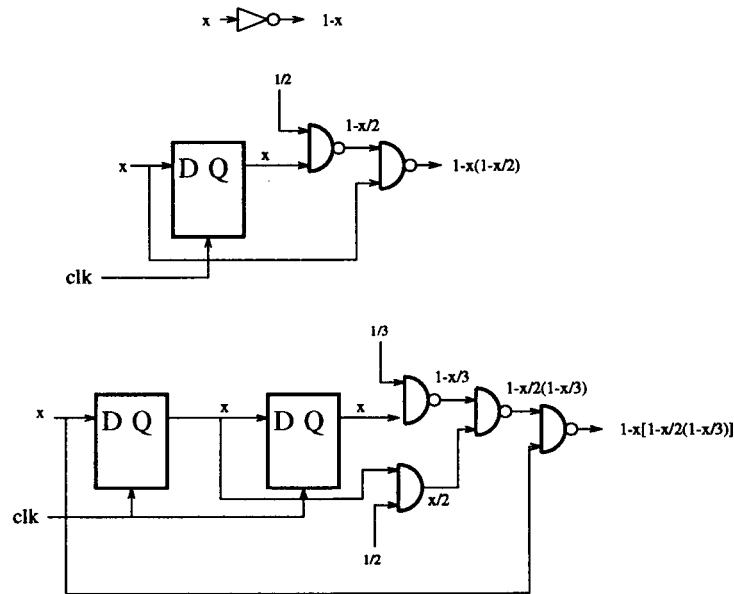


Figura 3.7: Aproximaciones de la transformación.

Se pueden considerar aproximaciones de orden superior de la transformación exponencial. Los circuitos que calculan estas operaciones se muestran en la figura (3.7).

3.5 Realizaciones físicas estocásticas.

Entre las realizaciones físicas aparecidas hasta ahora en la literatura, cabe destacar las siguientes:

Un grupo de investigadores de la North Carolina State University ha desarrollado diversas realizaciones físicas estocásticas. En sus primeros trabajos propusieron una arquitectura estocástica digital cuyo objetivo era simular el funcionamiento de las Redes de Hopfield, [12].

En este modelo se efectúan, de N^2 posibles operaciones de evaluación de conexiones, N simultáneamente, es decir, la arquitectura propuesta conlleva una serialización parcial de las operaciones a realizar. Esta serialización es el resultado de aplicar un algoritmo sistólico al proceso de evolución de una red de Hopfield, figura (3.8).

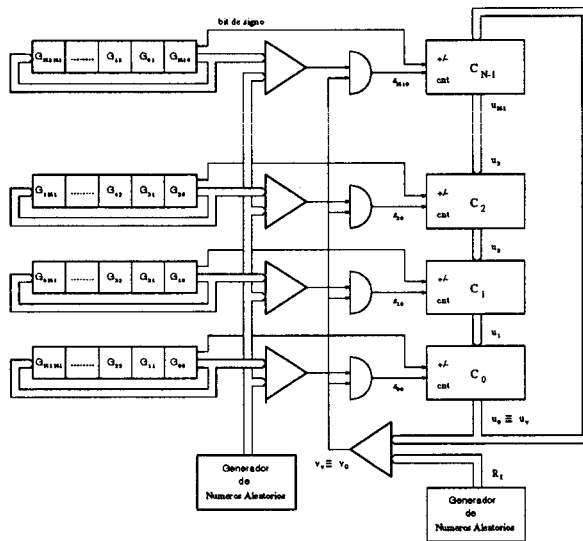


Figura 3.8: Red de Hopfield estocástica.

El número de multiplicadores estocásticos que contiene la red es igual al número de neuronas de dicha red. La estrategia que se sigue para efectuar la suma estocástica, es el empleo de contadores reversibles. Cada uno de estos contadores reversibles tiene asociado un multiplicador estocástico que consiste en una puerta AND, cuyas entradas son los trenes de pulsos procedentes del peso de conexión y de la neurona que está actuando sobre las demás. Téngase en cuenta que la serialización de la que se habló antes, consiste en calcular separada y sucesivamente la

influencia que tiene una determinada neurona sobre las demás que componen la red. La neurona asociada al contador inferior es la que, en un horizonte temporal dado, está actuando sobre las demás. Este contador es la entrada de uno de los comparadores digitales que actúan como generadores de secuencias aleatorias. Cada peso de conexión asociado al par de neuronas formado por la que está almacenada en este contador y una cualquiera de la red, está generando, por el mismo procedimiento, una secuencia estocástica de pulsos. Las puertas AND multiplican estos trenes y actúan como señales de incremento o decremento sobre los contadores reversibles. Una vez terminado el proceso, se hace un desplazamiento circular con los contenidos de los contadores reversibles y con los registros de almacenamiento de los pesos, de forma que el sistema queda preparado para evaluar la influencia de la siguiente neurona.

Poco tiempo después de proponer esta arquitectura, este grupo de investigadores la llevó a la práctica, [16]-[17]. Diseñaron un circuito integrado que contenía 20 neuronas y un registro circular para cien pesos. El sistema final se componía de 5 circuitos integrados, lo cual totalizaba 100 neuronas. En el apartado de resultados señalan los autores que a una velocidad de 10MHz un prototipo de 50 neuronas es capaz de resolver un problema de optimización combinatoria en 20ms, y que para resolver el mismo problema en el mismo tiempo utilizando una técnica de predicción y corrección se necesitaría un ordenador de 3.5 GFLOPS.

En los últimos años se ha hecho un esfuerzo importante para desarrollar realizaciones electrónicas de controladores borrosos que sean capaces de actuar en tiempo real. Togai y Watanabe desarrollaron el primer circuito integrado borroso en 1985, que era capaz de procesar 16 reglas en paralelo, [18]-[19]. Más tarde, Yamakawa diseñó un circuito basado en la lógica borrosa utilizando un tecnología analógica, [20]. Desde entonces se han concebido numerosos circuitos integrados que hacen uso de tecnologías tanto analógicas como digitales.

En los controladores borrosos se definen una serie de reglas de actuación deducidas heurísticamente a partir de una experiencia previa. Estas reglas se formulan mediante enunciados lógicos del tipo **si** la entrada 1 es A_1 y la entrada 2 es A_2 **entonces** la salida es S_{12} . Sin embargo estos enunciados pueden no ser, para un conjunto de entradas, ni totalmente ciertos ni totalmente falsos. El controlador evalúa **el grado de verosimilitud** de cada uno de los enunciados (que se representa mediante un número comprendido entre 0 y 1) y calcula la respuesta como una **ponderación** de las respuestas asociadas a cada una de las reglas si las premisas fuesen totalmente ciertas. Las reglas de actuación pueden hacerse adaptativas con objeto de dotar al controlador de un mecanismo de autoaprendizaje.

El cálculo estocástico se ha utilizado para evaluar procesos tan diversos como las operaciones matemáticas asociadas al cálculo de la respuesta y al proceso de autoadaptación, [21].

El cálculo estocástico se ha utilizado también en el campo de la electrónica de potencia. Así, por ejemplo, se ha utilizado para realizar físicamente controladores clásicos de motores, ya que las operaciones que hay que evaluar, a saber, multiplicaciones sumas e integraciones, se pueden evaluar estocásticamente de una manera sencilla.

En [22] se describe un controlador proporcional-integral que se utiliza para regular la velocidad de un motor de continua. El motor de continua está alimentado por un circuito troceador. Normalmente los circuitos troceadores son comandados por una señal PWM, de forma que la velocidad del motor depende de la relación $\frac{T_{on}}{T_{off}}$ de ésta. En la aplicación que se describe la propia señal estocástica generada por el controlador proporcional-integral se utiliza como comando del circuito troceador. En este caso, cuanto mayor sea la densidad del tren de pulsos mayor será la velocidad del motor.

En [23] se describe un sistema de control vectorial de la velocidad de un motor de inducción. Los interruptores del inversor trifásico de alimentación del motor de inducción se accionan por medio de señales PWM. Estas señales PWM generan armónicos indeseables en las corrientes que circulan por el motor, ya que provocan calentamientos en el núcleo. Este inconveniente se atenúa introduciendo en la fase del PWM una componente aleatoria que hace que la energía de los armónicos de la señal PWM se distribuya en un intervalo continuo de frecuencias.

3.6 Arquitectura estocástica totalmente paralela: Aplicación a la realización física de redes de neuronas.

Algunos trabajos publicados, así como una tesis recientemente desarrollada en el Grupo de Tecnología Electrónica de la Universidad de Sevilla [3], describen una arquitectura de procesamiento estocástico en la que todas las operaciones matemáticas se realizan simultáneamente. Esta es la única arquitectura totalmente paralela que se encuentra en la bibliografía y se ha empleado para la realización física de redes neuronales en las cuales se utiliza la función de activación signo.

Entre los campos en los cuales la lógica estocástica tiene aplicación está el de las Redes Neuronales. A diferencia de la estructura microprocesadora, basada en la existencia de un único y complejo módulo de cálculo, la potencia de las Redes Neuronales reside en la posibilidad de disponer de un número muy elevado de unidades simples de procesamiento y con gran cantidad de interconexiones entre ellas.

Las aplicaciones a las cuales van destinados cada uno de los dos métodos de computación anteriores están claramente diferenciadas por dos características, el

número de datos que se manejan y la cantidad de información contenida por cada uno de ellos. Las Redes Neuronales están pensadas para el manejo de multitud de datos con poca cantidad de información en ellos. El procesamiento de estos datos se realiza de forma paralela en unidades simples de cálculo, las Neuronas, mientras que en el microprocesador se realiza de forma seriada en un solo centro, la CPU. Otra diferencia reside en el hecho de que mientras este último almacena la información en una zona localizada como es la memoria, en las Redes Neuronales se distribuye homogéneamente en forma de interconexiones entre las neuronas.

3.6.1 Estructura de una red perceptrón

En este apartado se describe la realización de un Perceptrón, diseñado con tecnología digital, el cual es usado para llevar a cabo un problema de control en el que se implementa una superficie de decisión. El Perceptrón es una estructura compuesta por una cierta cantidad de neuronas dispuestas ordenadamente en varias capas. En general, estas capas se clasifican en dos, una de salida y varias ocultas.

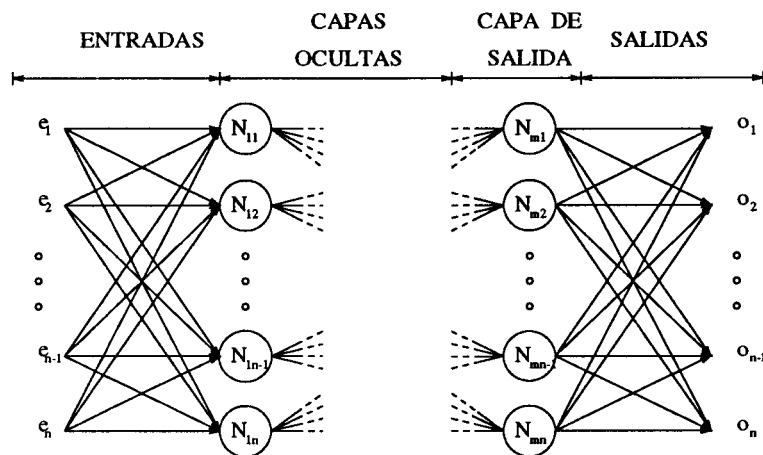


Figura 3.9: Esquema general de un Perceptrón.

Las entradas de la red van a todas y cada unas de las neuronas de la primera capa oculta, mientras que como salidas, se tomará las de la última capa. El hecho

de llamarse ocultas es debido a que, si bien son accesibles mediante las señales de entradas, sus salidas no son visibles desde el exterior. Las neuronas de una capa están conectadas a todas las de la capa siguiente.

Ante un determinado patrón de señales de entrada, el Perceptrón responde con unas determinadas señales de salida. La información necesaria para el proceso de correspondencia entre entradas y salidas es almacenada en cada neurona a través de los pesos. Dichos pesos son los que establecen la influencia que ejercen unas neuronas sobre otra. En general, a cada neurona le llega como estímulo la señal de salida de cada una de las de la capa anterior. Dichas señales se multiplican por su peso correspondiente teniendo así una determinada ponderación de éstas. De esta forma, se le concede más importancia a las señales procedentes de algunas neuronas que de otras, incluso dotándoles de un carácter activador o inhibitor, si se tiene en cuenta el signo de cada peso.

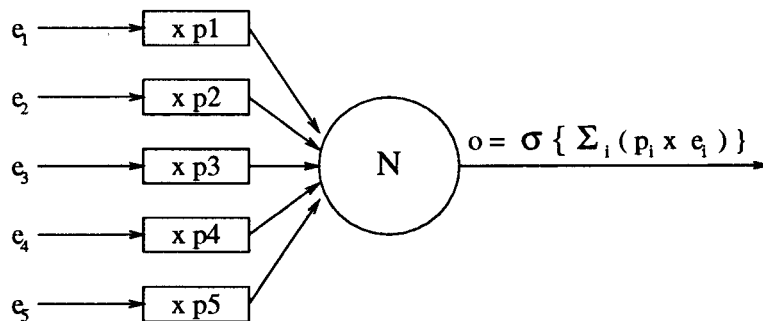


Figura 3.10: Operaciones de una Neurona.

Los pesos de una red son establecidos mediante un proceso de aprendizaje. El más usado es el llamado Back Propagation, [13]. La información de partida necesaria es un conjunto de pares de entradas-salidas que podrían ser una serie de puntos conocidos de una determinada superficie. Presentando patrones de entrada de forma aleatoria a la red, ésta responderá con una serie de señales de salida, de acuerdo con la información de los pesos contenidos en ese momento. La comparación entre las señales reales y las teóricas dará un error que servirá para adaptar cada uno de los

pesos de la red según unas leyes dadas. El proceso de aprendizaje acabará cuando los errores generados sean menores que un determinado valor, fijado con antelación.

El funcionamiento básico de una neurona puede resumirse según la siguiente expresión.

$$a = \sigma(\sum_i p_i \cdot e_i - b)$$

La señal de salida se calcula ponderando todas las entradas e_i por el conjunto de pesos p_i y sustrayendole un valor umbral b . Al resultado de esta ponderación se le aplica una función de activación no lineal σ , entre las cuales pueden estar la función signo, sigmoideal, etc...

3.6.2 Realización física

Con idea de comprobar la validez de la lógica estocástica en este campo, se diseñó y fabricó un circuito integrado, que implementa un Perceptrón, con tecnología puramente digital y utilizando esta técnica. Para ello, se decidió llevar a cabo un sencillo problema de control, [14], en el cual el conjunto de todos los pesos y umbrales aproximase una superficie descrita por 196 puntos, y que divide en dos un espacio de tres dimensiones. Mientras que las tres entradas forman un patrón determinado, la neurona de salida lleva asociada una función lógica que decide si dicho patrón pertenece a uno y otro subespacio. Las necesidades para esta aplicación eran de 10 neuronas ocultas y 1 de salida [6].

Con objeto de aumentar la versatilidad de circuito a realizar, se le dotó de cinco entradas en vez de tres. De este modo, las que no son utilizables se anulan poniéndolas a un nivel bajo. Por otro lado, el número de neuronas capaces de ser integradas en un colo chip fue de 5 ocultas y 1 de salida, lo que daba un total de 30 pesos y 6

valores umbrales. Este hecho hizo necesario dotar al circuito de la capacidad de ser conectado a otro con objeto de expandir el número de neuronas de la capa oculta. La expandibilidad de estos circuitos se puede realizar fácilmente aprovechando las características de la lógica estocástica. Si se considera la expresión anterior, aplicada a la neurona de salida, se ve como el sumatorio puede ser resuelto de forma parcial en diferentes sumas. De esta forma, cada una de estas sumas pueden ser realizadas en circuitos integrados distintos.

$$\sum_i p_i \cdot e_i - b = (\sum_l w_l \cdot e_l) + (\sum_m w_m \cdot e_m) + (\sum_n w_n \cdot e_n) - b$$

Tanto el conjunto de 30 pesos como los 6 umbrales de las neuronas debían ser fácilmente programables, por lo cual se utilizaron registros para almacenar cada uno de ellos. Los pesos fueron codificados con 7 bits más el signo, y debían ser introducidos de forma secuencial al principio de la aplicación.

La función de activación utilizada en cada neurona ha sido la función signo, ya que hasta entonces era la única que está resuelta de forma sencilla en lógica estocástica. De esta forma se hace una valoración de los sumandos, decidiendo si en esa neurona los de un signo tienen más peso que los del otro.

El esquema de bloques de cualquiera de las neuronas existentes en el circuito integrado se detalla en la figura.

Se observa como existe una zona de almacenamiento de pesos y otra de generación de números aleatorios, que junto con un bloque comparador son los encargados de producir una secuencia estocástica de bits que representan al valor del peso almacenado. Por otro lado, el resto de los componentes de la Neurona son los que realizan las operaciones específicas de ésta, como son la multiplicación de los pesos por las entradas, la suma de todos estos términos así como la resolución de la función signo.

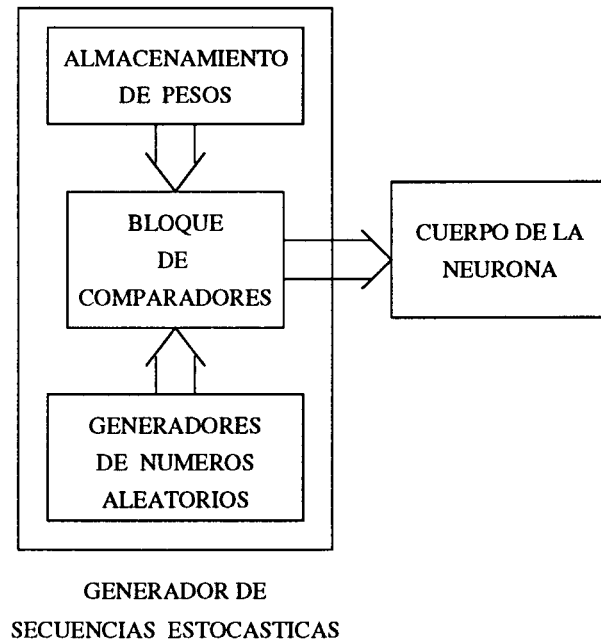


Figura 3.11: Esquema de bloques de una Neurona.

Cuerpo de la neurona

La neurona es la unidad básica de procesamiento del Perceptrón. En ella se realizan los cálculos mencionados anteriormente y que se resumen según la expresión

$$a = \sigma(\sum_i p_i \cdot e_i - b)$$

La primera operación que se realiza es la multiplicación de los pesos. En lógica estocástica, esto se realiza con una simple puerta AND. Los pesos están almacenados en registros de 8 bits, de los cuales, el más significativo es el signo. Para las neuronas de la capa oculta, las entradas son introducidas directamente desde el exterior, por lo que se supone que entran ya codificadas de forma estocástica. Sin embargo, los pesos deben ser convertidos mediante un mecanismo generador de trenes de pulsos. Por otra parte, el valor umbral es considerado como un peso más que es multiplicado por una entrada igual a 1.

Para la neurona de salida se tiene un esquema similar, pero con la diferencia de que las entradas a ésta son directamente los bits de signo de las neuronas anteriores, esto es, o bien un valor lógico alto o bien un valor lógico bajo.

Para evaluar el signo de la suma de términos se hace uso de la técnica de transformación exponencial descrita en la bibliografía, [3]. Esta técnica permite convertir la suma en productos considerando ciertas restricciones. La transformación exponencial de una secuencia estocástica se puede calcular haciendo uso del desarrollo en serie de Taylor. En este circuito se realiza la inversión de los términos $p_i \cdot e_i$ con lo que se consigue una aproximación lineal de dichas exponenciales.

$$\exp(-(p_i \cdot e_i)) = 1 - (p_i \cdot e_i) = NOT(p_i \cdot e_i)$$

Otras formas de aproximar la exponencial están descritas en [3], con las cuales se consiguen incluir términos cuadráticos y cúbicos. Los circuitos que realizan estos tres tipos de aproximaciones han sido mostrado en la figura (3.7). Se puede observar que aunque la complejidad de éstos aumenta con el grado de precisión, los circuitos implicados no dejan de ser bastante simples.

Debido a que las entradas se consideran trenes de pulsos positivos, el signo perteneciente a los términos $p_i \cdot e_i$ es el correspondiente a cada uno de los pesos. La función de activación signo es la encargada de decidir si existe mayor densidad de pulsos de términos positivos o negativos. En el caso particular de aplicar esta función, no será necesario deshacer la operación de compresión. Sin embargo, en este momento se tendrá que conducir a ambos tipos de términos a dos puertas AND distintas, una

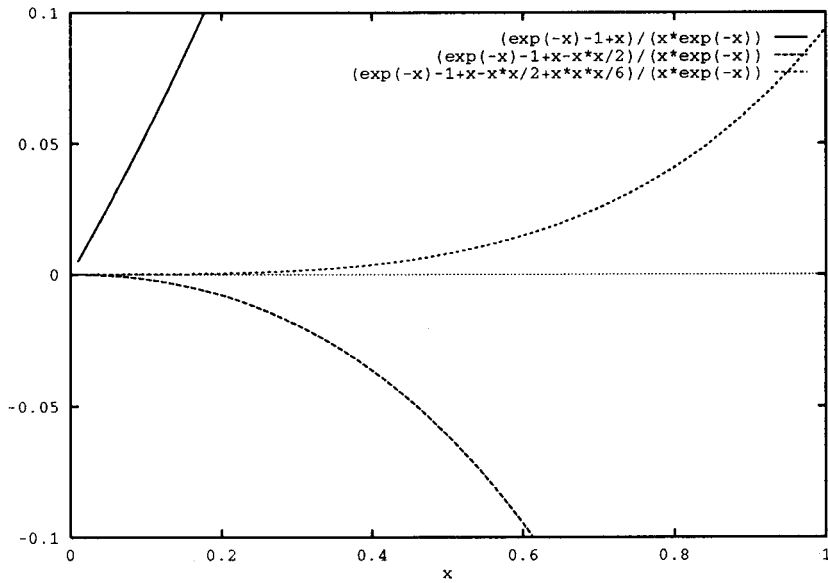


Figura 3.12: Error normalizado en la transformación exponencial, para primero, segundo y tercer orden de aproximación.

correspondiente a los términos positivos y otra a los negativos.

$$\text{signo}(\sum_i (p_i \cdot e_i)) = \text{signo}(\sum_i (p_i \cdot e_i)^+ - \sum_i (p_i \cdot e_i)^-) =$$

$$\text{signo}(\prod_i \exp(-(p_i \cdot e_i))^+ - \prod_i \exp(-(p_i \cdot e_i))^-)$$

Por cada sumando existe un bloque lógico encargado de conducir la secuencia estocástica a la puerta correspondiente en función del signo de éste, mientras que a la otra puerta le hace llegar un nivel alto constante de forma que no afecte a la operación AND. Una vez alcanzado este punto, se tienen dos trenes de pulsos **representativos** de la suma de ambos tipos de términos. Con objeto de expandir el número de sumandos de una neurona, a las dos puertas AND anteriores se les pueden hacer llegar sendos trenes de pulsos ψ_{in}^+ y ψ_{in}^- , realizados en otro circuito integrado.

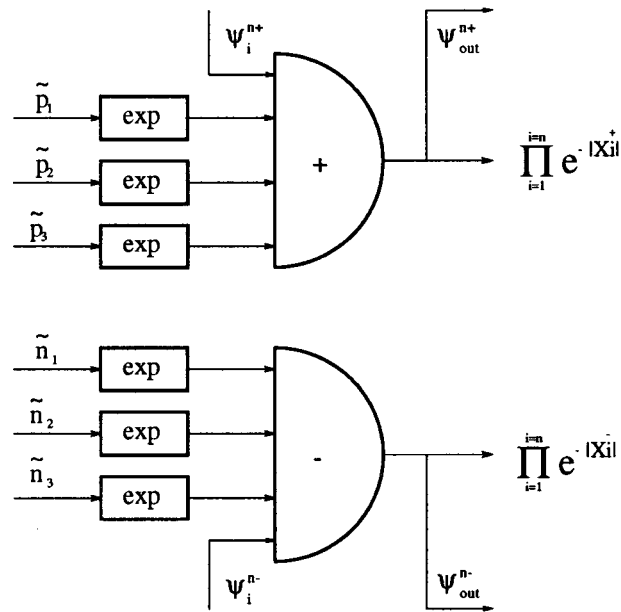


Figura 3.13: Suma comprimida de términos.

En el caso de no querer incrementar el número de sumandos, las señales ψ_{in}^+ y ψ_{in}^- deben ser conectadas a un nivel alto, para no afectar al resultado de la operación AND. A su vez, ψ_{out}^+ y ψ_{out}^- pueden servir como entradas a otra neurona.

En este circuito, la estructura AND expansora solo es incluida en la neurona de salida, de modo que el incremento del número de sumandos de ésta es equivalente a aumentar el número de neuronas de la capa oculta.

La última estructura que aparece en el cuerpo de la Neurona tiene como objetivo calcular la salida de ésta. En este conjunto se realiza una evaluación de las densidades de las dos secuencias de pulsos representantes de los términos positivos y negativos. Esta evaluación se realiza en cada ciclo de reloj, actuándose sobre un contador reversible de manera que, si llega un pulso exclusivamente de los términos negativos se decrementa en 1 unidad, si llega exclusivamente de los positivos se incrementa en 1, mientras que si se reciben ambos pulsos a la vez, o ninguno de ellos, el contador permanece inalterado. El signo de la neurona de salida vendrá dado por un bit que cambia de valor, dependiendo del valor del contador.

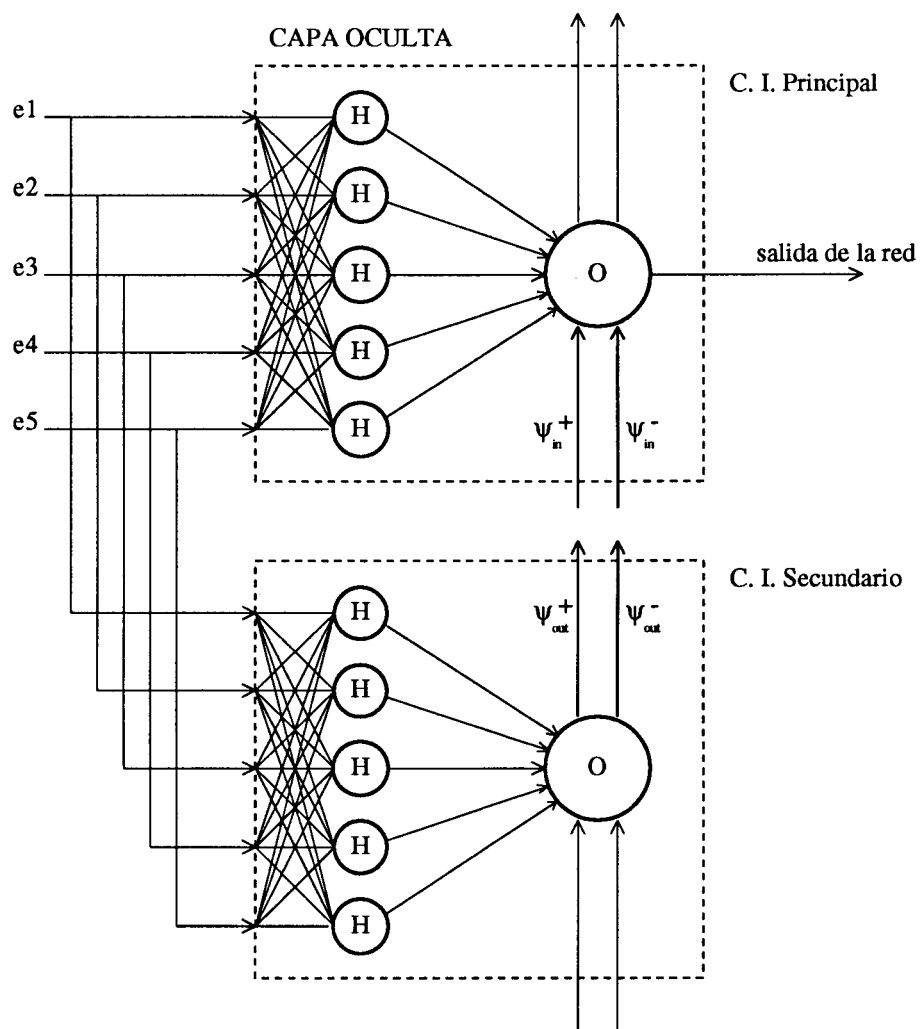


Figura 3.14: Estructura expansora de capa oculta.

Bloque generador de secuencias estocásticas.

Para generar los trenes de pulsos asociados a los pesos se dispone de registros de almacenamiento de ocho bits (incluido el signo), comparadores y generadores digitales de números aleatorios. Los registros están conectados en cascada, de forma que la programabilidad de la red se facilita. La introducción, tanto de los pesos como de los valores umbrales, se realiza mediante la señal auxiliar LOAD, de forma que entran desde el exterior al primer registro de almacenamiento y van siendo desplazados hasta su ubicación definitiva. El último registro tiene su salida visible desde el exterior, con una doble finalidad. La primera es la de comprobar la carga correcta de todos los valores. Esto se realiza introduciendo dos veces seguidas el conjunto de pesos, de manera que se puede ver una vez el paso de todos ellos por los 8 pines de salida correspondiente. La segunda finalidad es la de conectar dos o mas circuitos en cascada sin mas que unir la salida de pesos de uno con la entrada de pesos del otro. Así, la programación de pesos se puede hacer de forma mas eficiente sin necesidad de ir direccionando los pesos desde el exterior hacia uno u otro circuito.

Por cada peso, se realiza la comparación del valor de éste con el número generado por un registro de desplazamiento (LFSR) cuyo bit mas significativo se realimenta a través de un conjunto de puertas XOR, a partir de algunos de los restantes bits. Este método de generación de números aleatorios está ampliamente recogido en la bibliografía, [12], [6], [2]. De una manera sencilla se consigue generar un ciclo de números pseudoaleatorios de longitud $2^n - 1$, donde n es el número de bits del LFSR. La comparación de cada uno de los pesos con los valores pseudoaleatorios anteriormente descritos proporciona una secuencia de pulsos con características estocásticas aceptables, y cuya densidad representa el valor del peso. Sin embargo, el hecho de que para todos los pesos de una neurona, los trenes de pulsos estocásticos asociados deban ser estadísticamente independientes, impide utilizar el mismo generador LFSR para todos ellos.

Si se calcula la correlación entre una secuencia $x(t)$ generada por un LFSR y la obtenida desplazando ésta en el tiempo un ciclo de reloj, se obtiene un valor alto, lo cual indica el caracter pseudoaleatorio de ambas.

El número aleatorio generado en el ciclo $t + 1$ se puede calcular a partir del generado en t dividiendo éste por dos y sumándole 0 ó 2^{n-1} , que no es más que poner el bit más significativo a 0 ó a 1. Si se tiene en cuenta la secuencia $x(t + 8)$, la correlación baja a valores aceptables, con lo que para cada neurona se pueden utilizar las secuencias $x(t)$, $x(t + 8)$, $x(t + 16)$... para comparar cada uno de los pesos. Esto es así ya que al desplazar por 8 veces el registro LFSR, la nueva información contenida en él no guardará casi ninguna relación con la original. En la figura (3.15) se esquematiza el ciclo cerrado recorrido por un generador de números aleatorios de este tipo, y las subsecuencias descritas anteriormente.

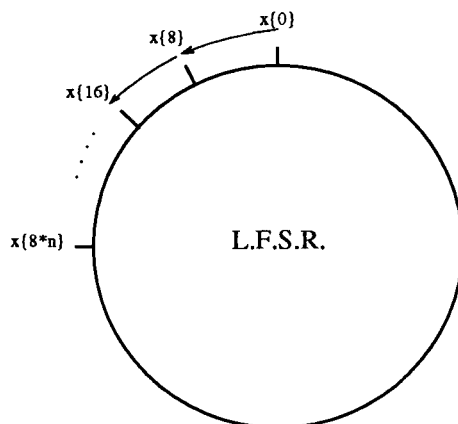


Figura 3.15: Ciclo de números recorrido por un LFSR.

Esta solución fue la adoptada en la realización que se está describiendo. El registro LFSR es único para todo el circuito, mientras que aparecen bloques compuestos por puertas XOR que son los encargados de evaluar, a partir de un número aleatorio dado, el que ocupa un lugar en la cadena distante 8 posiciones del primero. Ver figura (3.16).

La estructura de estos bloques, está basada en la lógica de desplazamiento del

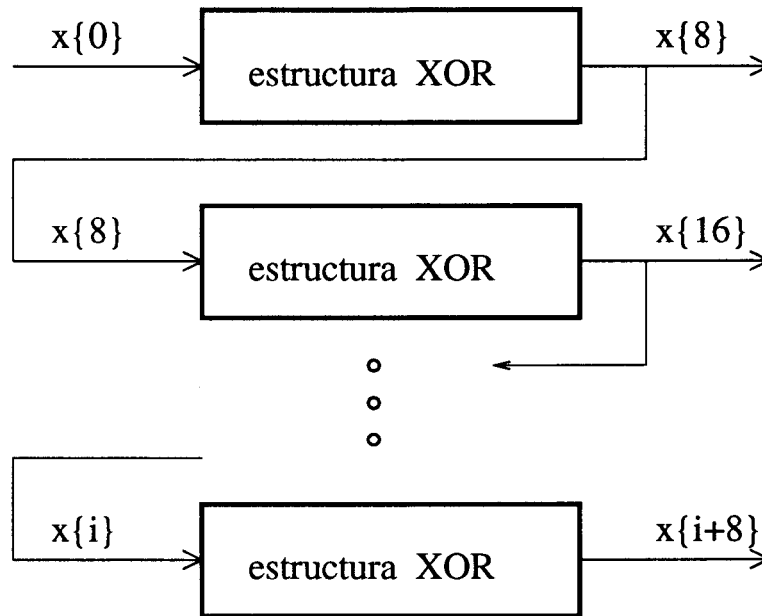


Figura 3.16: Generación de subsecuencias aleatorias.

LFSR. Si se quiere obtener el siguiente número en el ciclo barrido por un generador de números aleatorios de este tipo, basta con tomar como bits del nuevo valor los del anterior desplazados una posición hacia la derecha, mientras que como bit más significativo aparecerá el correspondiente al resultado de aplicar las puertas XORs a los bits del valor antiguo.

El generador LFSR de secuencias pseudoaleatorias consta de un registro de desplazamiento de nueve bits y una puerta XOR que realimenta el bit más significativo del registro, a partir de los bits 0 y 4. En este caso, si se quiere obtener el número del ciclo distante 8 posiciones basta con ir acumulando las operaciones XORs en uno de los bloques anteriores para así obtener los 7 primeros bits, mientras que al menos significativo se le hace corresponder el bit más significativo del valor original. El esquema de un bloque como el descrito se muestra en la figura (3.17).

Para una neurona es necesario suministrar un grupo de secuencias $x(t+i)$ de forma que cada peso reciba un número aleatorio diferente. Sin embargo, este grupo

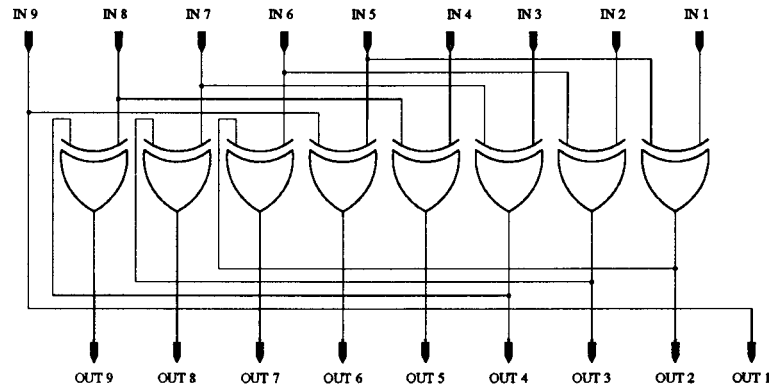


Figura 3.17: Estructura de puertas xores.

puede ser el mismo para todas las neuronas, ya que existe una total independencia entre las operaciones realizadas en ellas. En este caso, en una neurona de la capa oculta existen 5 pesos más el valor umbral, por lo que será necesario utilizar 6 estructuras XOR de las anteriores.

Para una neurona de la capa de salida existen dos posibilidades; que sea la neurona de salida del integrado principal, o que tenga un caracter auxiliar, formando parte de un segundo integrado, y suministrando sumandos extras a una neurona del primer tipo. En este último caso, las secuencias $x(t + i)$ utilizadas deben ser diferentes que las de la neurona principal. Para conseguir el mismo efecto que el descrito anteriormente, es necesario que el LFSR perteneciente al segundo integrado no tenga el mismo contenido que el del primero, sino que genere la secuencia siguiente a la correspondiente al último peso de la neurona de salida del integrado principal. Este hecho se esquematiza en la figura (3.18).

Para poder conseguir esto, el registro LFSR incorpora la posibilidad de alimentar el bit mas significativo con el correspondiente bit de la última estructura XOR del integrado principal. Esta alimentación debe ser controlada con una señal que indique que el integrado está funcionando en modo principal o en modo auxiliar.

Con una configuración de este tipo, el reparto de la secuencia aleatoria en los

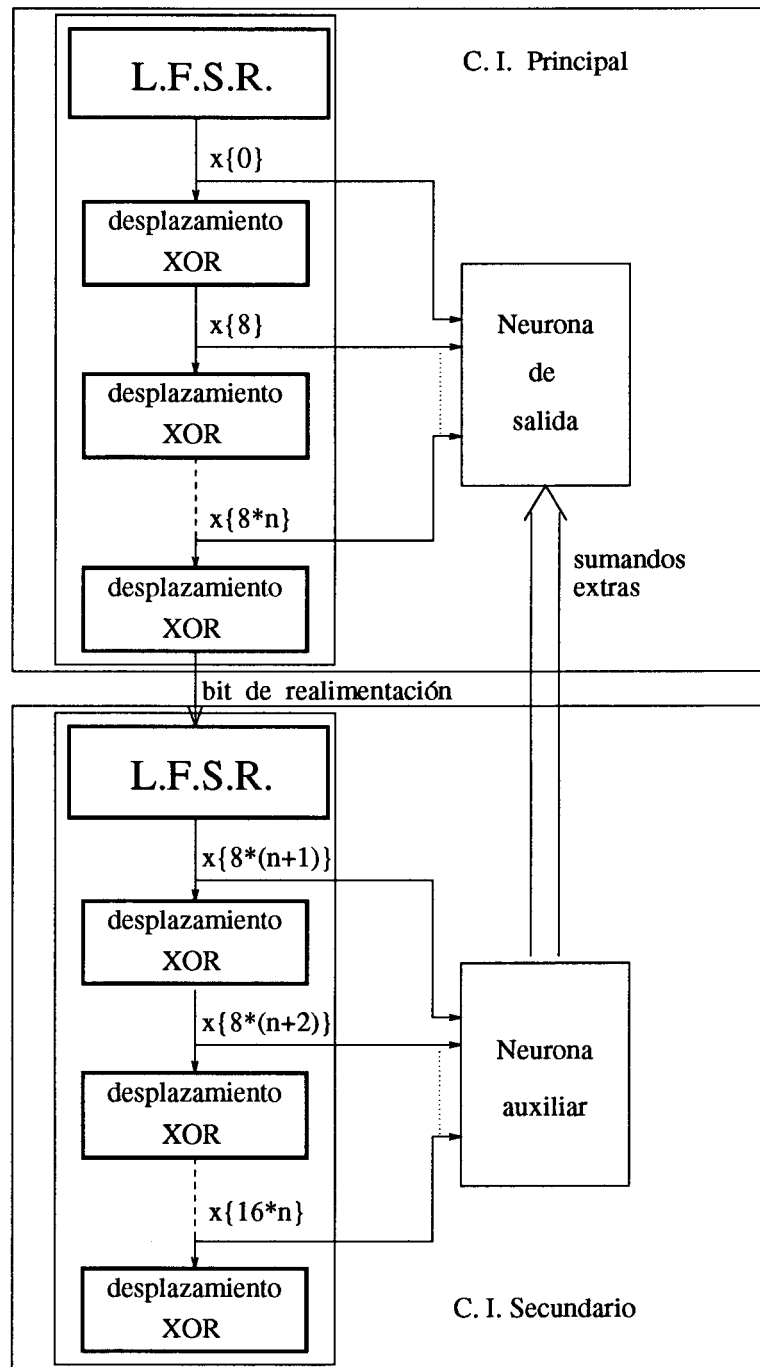


Figura 3.18: Encadenamiento de números aleatorios en dos circuitos integrados.

dos circuitos integrados se puede esquematizar el la figura (3.19).

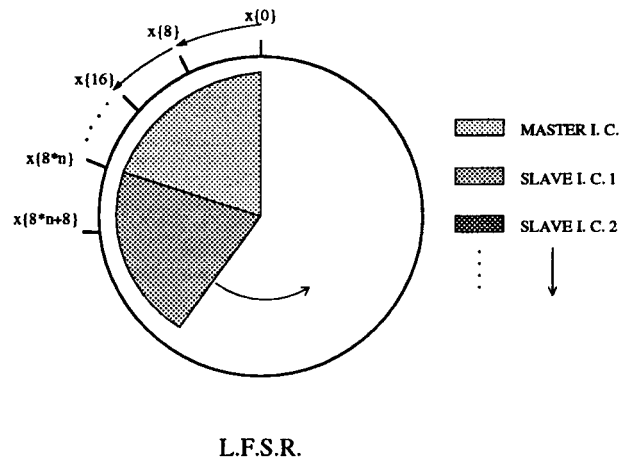


Figura 3.19: Distribución del ciclo aleatorio en ambos integrados.

En cuanto a las señales de entrada que van a ser multiplicadas por los pesos, se ha supuesto que para la primera capa de neuronas vendrán ya codificados de manera estocástica desde el exterior, mientras que para la neurona de salida éstas resultan ser las respuestas saturadas de la capa anterior, es decir, 0 ó 1.

La Red Neuronal descrita fue diseñada con tecnología ES2 1.5u, usando el paquete de programa EDGE, y fabricada via EUROCHIP. El conjunto comprende seis Neuronas ocultas y una de salida, con 5 pesos y 1 valor umbral cada una, que ocuparon $25mm^2$ de silicio, mientras que la frecuencia máxima de trabajo resultó ser de 7.5Mhz . La microfotografía (3.20) se corresponde con este circuito integrado, mientras que en la fotografía (3.21) se muestra el circuito de testeo de la red neuronal Perceptrón.

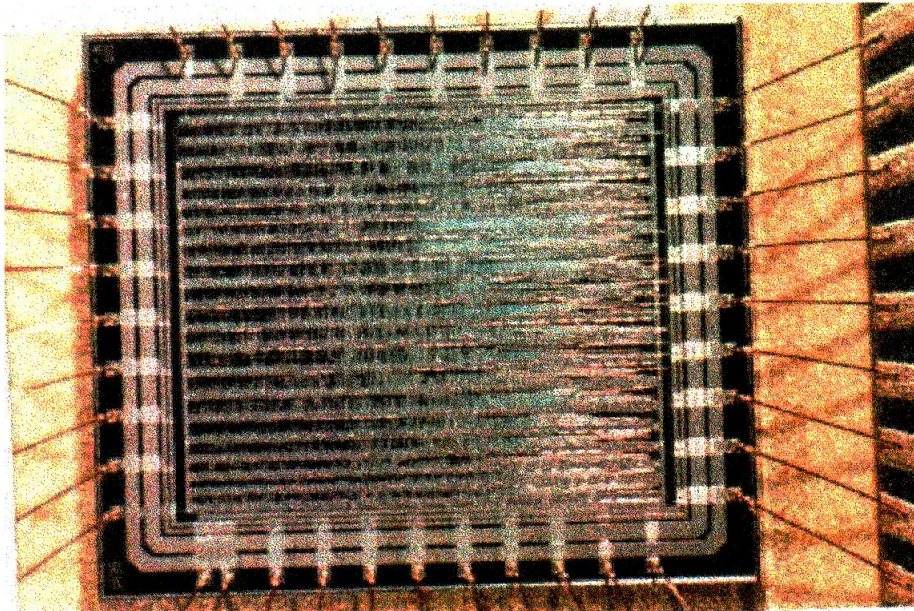


Figura 3.20: Realización ASIC de un Perceptrón con arquitectura estocástica

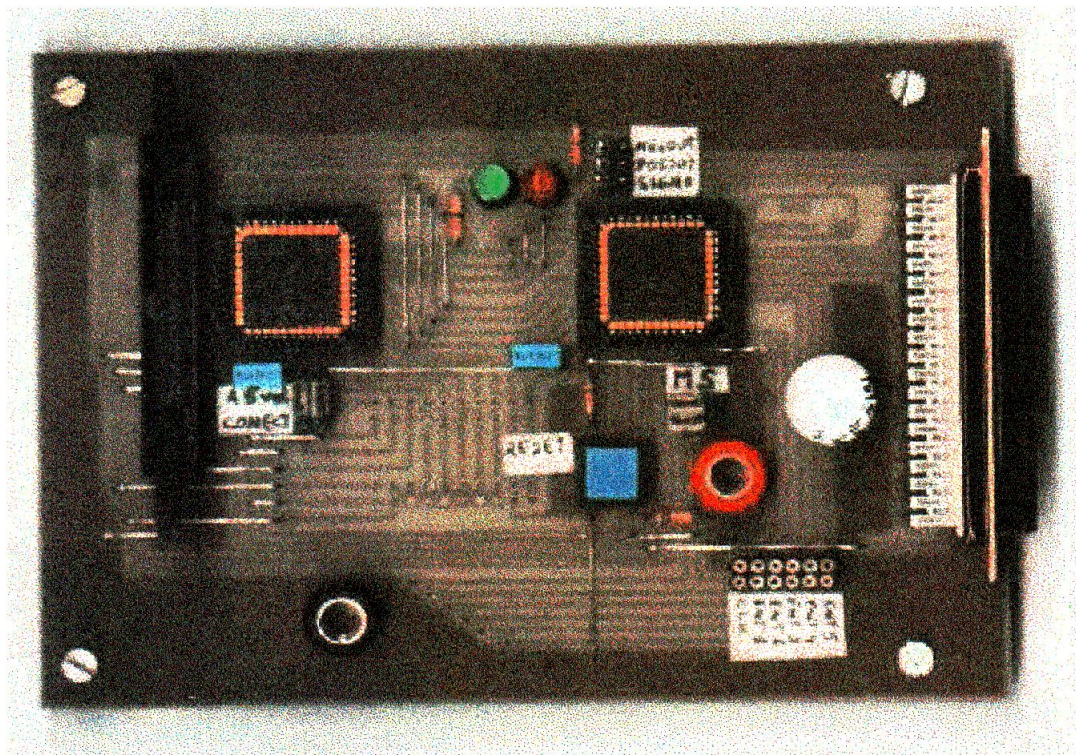


Figura 3.21: Perceptrón realizado con dos circuitos integrados.

Como ejemplo de aplicación, se reprodujo una superficie de control en un espacio de 3 dimensiones, a partir de una aplicación realizada previamente con componentes discretos, [14]. Dicha superficie venía aproximada por un conjunto de diez planos, que eran definidos por un total de 60 pesos. Por tanto, fue necesario la utilización de dos circuitos integrados, figura (3.21), para efectuar este ejemplo, [6].

Debido a la aproximación de la superficie por medio de planos y a la resolución de los pesos (128), la superficie teórica no es exactamente igual a la reproducida por la red. A esto hay que añadir los errores cometidos en la evaluación de las sumas estocásticas. Una comparación entre la respuesta real del Circuito Integrado y la ideal se muestra en la figura (3.22).

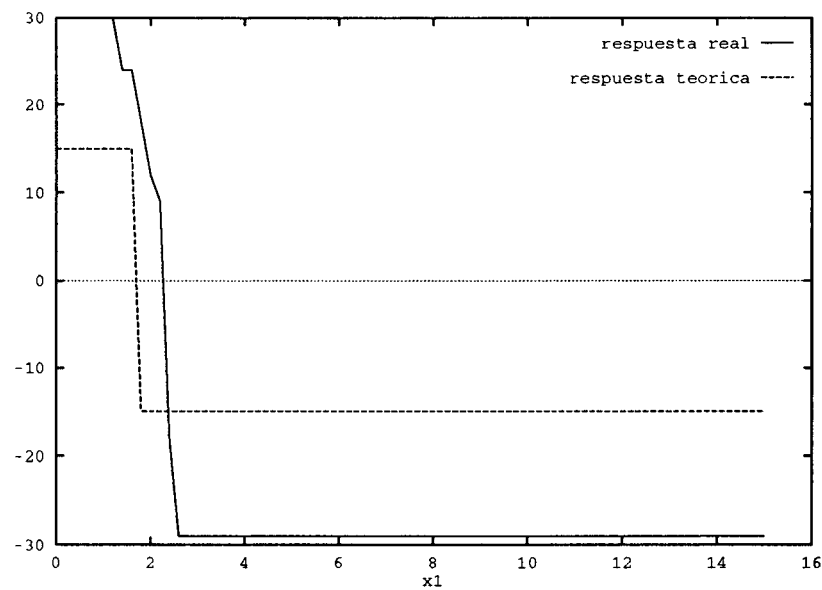


Figura 3.22: Comparación entre la respuesta real y teórica de la neurona de salida.

3.7 Limitaciones del procesamiento estocástico paralelo.

La exposición de las ventajas y limitaciones del cálculo estocástico se realizará analizando la estructura de una neurona correspondiente a la realización anterior. En la figura (3.23) se muestra detalladamente el esquema de una de estas neuronas.

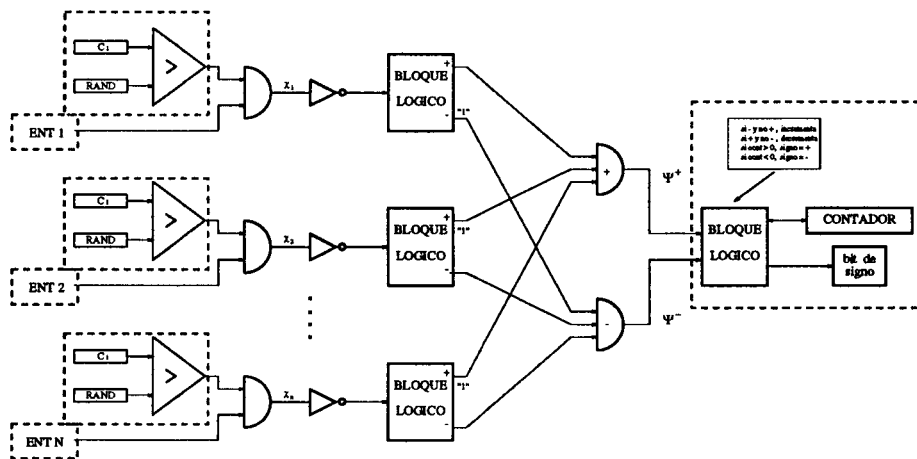


Figura 3.23: Arquitectura estocástica de una neurona.

Como se puede ver en el esquema, la característica que hace esta arquitectura más atractiva es la posibilidad de ejecutar de forma **masivamente paralela** un gran número de operaciones productos y sumas. Esto, unido al hecho de que la realización de estas operaciones es llevada a cabo por **circuitos electrónicos muy simples** hacen del cálculo estocástico una herramienta muy útil en la resolución de problemas con una gran cantidad de sumas y productos.

En cuanto a las limitaciones, la primera que se debe tener en cuenta es el hecho de que todavía no ha aparecido una estructura capaz de realizar la **suma estocástica paralela**. En el caso del clasificador realizado en el ejemplo anterior se ha utilizado la función de activación signo sobre una suma de términos. Esta es la única aplicación descrita en la bibliografía que aprovecha de una manera muy particular

la información contenida en los trenes de pulsos ψ^+ y ψ^- para hallar el signo de la suma. Es conveniente desarrollar una forma de realizar una verdadera suma sobre un conjunto de términos.

En la realización anterior también se pone de manifiesto que, la necesidad de generar trenes de pulsos que representen a los pesos de la red impone la existencia de circuitería adicional que **incrementa enormemente la superficie ocupada de silicio**. En este caso, el registro donde se almacena el peso, el comparador digital y el generador de números aleatorios (LFSR), junto con los bloques XOR, suponen el 90% del total de la Neurona.

Por otro lado, las entrada de un procesador estocástico deben estar **codificadas como trenes de pulsos aleatorios**, al igual que las señales resultantes de salida. Se hace necesario el desarrollo de circuitos conversores que permitan pasar de señales analógicas o digitales a señales estocásticas y viceversa. Este es un problema parecido al que nos encontramos en la tecnología digital. Su sencillez y fiabilidad han servido como base para un espectacular aumento en su utilización, pero como contrapartida han tenido que desarrollarse convertidores analógico-digital y digital-analógico.

A modo de resumen se enumeran las necesidades de mejoras de las arquitecturas estocásticas totalmente paralelas.

1. **Desarrollo de una estructura capaz de realizar la suma.**
2. **Nueva estructura generadora de trenes de pulsos estocásticos, de reducido costo en área de silicio.**
3. **Realización de dispositivos capaces de realizar la conversión entre señales analógicas y estocásticas.**

La solución de estos problemas con el diseño de circuitos electrónicos adecuados permitiría la integración de sistemas estocásticos complejos, con una elevada densidad de cálculo.

Capítulo 4

Suma estocástica.

4.1 Introducción.

Uno de los inconvenientes principales del cálculo estocástico es que no existe una estructura que permita sumar con facilidad trenes de pulsos en paralelo. En la sección anterior se ha resumido una arquitectura simple que permite decidir si la suma de un conjunto de términos codificados estocásticamente es una cantidad positiva o negativa. En este apartado se va a describir dicha arquitectura y se verá qué cambios hay que introducir para que pueda evaluarse el valor de la suma de los pulsos y no tan solo el signo de la suma. Tras este estudio teórico van a describirse las aplicaciones que se han desarrollado utilizando esta estructura.

4.2 Suma de trenes estocásticos.

El problema que presenta la operación suma estocástica frente a la operación producto estocástico es que, mientras que el producto de dos términos cuyos valores

están comprendidos entre 0 y 1 es otro término también comprendido entre 0 y 1, en la suma esto no es necesariamente así. Por consiguiente un tren de pulsos estocásticos puede representar el producto de dos trenes estocásticos, pero no su suma. Para resolver este problema, en un trabajo anterior se propuso una nueva arquitectura estocástica.

En esta arquitectura se separan los sumandos positivos y los negativos. Los términos positivos y negativos se procesan separadamente. Primeramente se le aplica a cada término una transformación exponencial, cuyo argumento es el valor absoluto de los términos cambiado de signo. Los trenes de pulsos transformados se multiplican mediante dos puertas AND (los positivos por un lado y los negativos por otro). El motivo por el que se calculan estas transformaciones se comprende si se tiene en cuenta la siguiente identidad:

$$\prod_{i=1}^{i=n} e^{-x_i} = e^{\sum_{i=1}^{i=n} -x_i} \quad (4.1)$$

Se tiene que el resultado de multiplicar los términos positivos transformados por un lado y los términos negativos transformados por otro, es equivalente a haber evaluado separadamente la suma de los positivos y los negativos y, a los números resultantes, aplicarles una función (exponencial, cambiándole el signo al valor absoluto de los argumentos) de compresión que establece una aplicación entre el intervalo $[0, \infty)$ y el intervalo $(0, 1]$. De esta forma puede representarse la suma de trenes estocásticos por medio de un par de trenes estocásticos, ya que la función exponencial ha comprimido la suma de forma que tome valores entre 0 y 1.

La transformación exponencial de los trenes de pulsos no se calcula exactamente. Se aproxima por un número finito de términos del desarrollo en serie de Taylor de la función. En la figura (3.7) se muestran los circuitos que calculan estas aproximaciones.

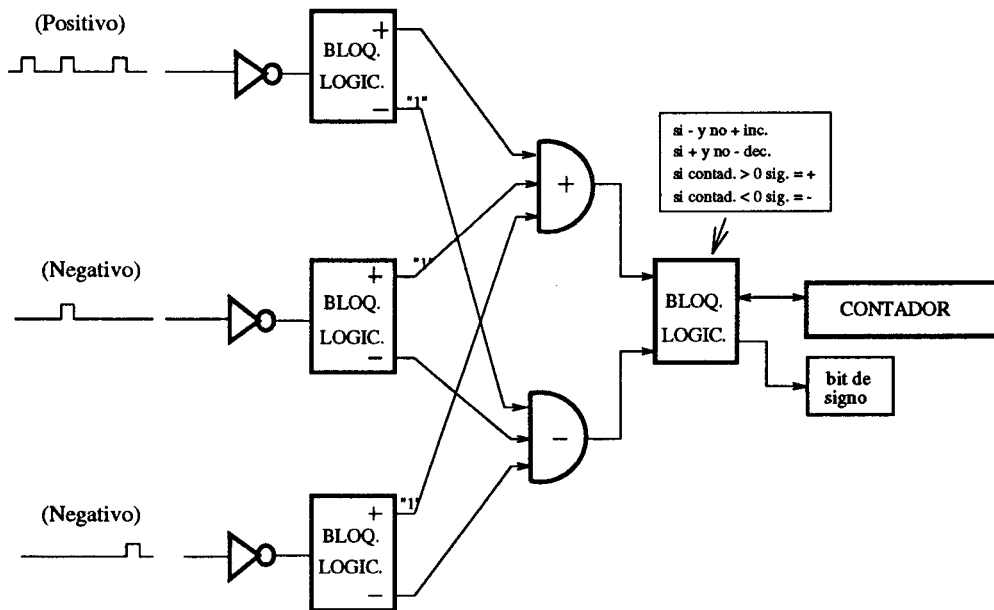


Figura 4.1: Arquitectura estocástica.

Como resultado de los procesos anteriores se obtienen dos trenes de pulsos representativos de la suma de los términos positivos y de los negativos. En la arquitectura propuesta anteriormente lo único que se pretendía era decidir cuál de estos dos trenes tenía una densidad mayor. De esta forma se determinaba cuál era el signo de la suma de todos los términos. La arquitectura resultante se muestra en la figura (4.1).

4.3 Extensión de la arquitectura de suma.

Los trenes de pulsos que aparecen a la salida de las dos puertas productos se pueden combinar entre sí de forma que se obtiene un resultado que, bajo determinadas hipótesis, es una función de la suma de **todos** los sumandos.

Antes de exponer la teoría en que se basa se va a describir la arquitectura que se propone. La estructura se muestra en la figura (4.2). Este circuito combinacional

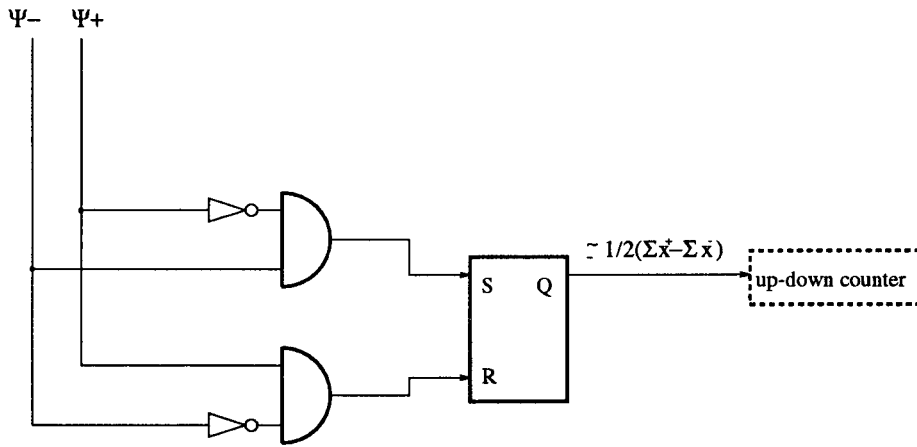


Figura 4.2: Circuito totalizador.

detecta cuándo la salida de la puerta producto de los sumandos positivos está a nivel lógico alto y la salida de la puerta producto de los sumandos negativos está a nivel lógico bajo. Cuando esto ocurre el circuito genera una señal de puesta a cero de un biestable. Si la puerta de los sumandos positivos está a nivel lógico bajo y la de los sumandos negativos está a nivel lógico alto se genera una señal de puesta a uno del biestable. Se va a demostrar que el estado del biestable se describe mediante un proceso de Markov, cuyo estado promedio (interpretando el estado 1 como +1 y el estado 0 como -1) viene dado por la expresión siguiente:

$$\frac{1 - e^{-(\sum_{i=1}^{i=n} x_i^+ - \sum_{i=1}^{i=n} x_i^-)}}{1 + e^{-(\sum_{i=1}^{i=n} x_i^+ - \sum_{i=1}^{i=n} x_i^-)}} \quad (4.2)$$

es decir, se obtiene una función de transferencia sigmoïdal, figura (4.3).

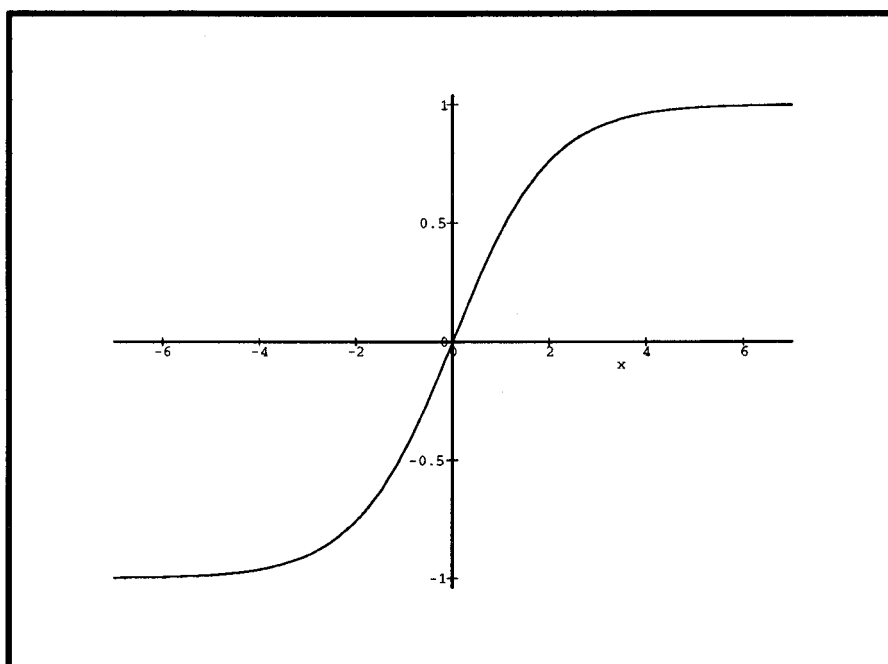


Figura 4.3: Función sigmoide.

4.3.1 Deducción de la función de transferencia sigmoidal.

Considérese la figura anterior. El estado del biestable es una variable aleatoria, ya que tanto la salida de la puerta positiva como la de la negativa lo son. Los dos estados del biestable se llamarán "positivo" y "negativo". Supóngase que en un determinado ciclo de reloj el estado es "positivo". En el próximo ciclo de reloj el estado conmutará a "negativo" sólo si la salida de la puerta producto positiva está a nivel lógico alto y la salida de la puerta producto negativa está a nivel bajo. De manera análoga, si en un determinado ciclo de reloj el estado es "negativo", en el próximo ciclo de reloj se conmutará al estado "positivo" sólo si la salida de la puerta producto negativa está a nivel alto y la salida de la puerta positiva está a nivel bajo.

Este es un proceso estocástico Markoviano y el diagrama de transición de estado se muestra en la siguiente figura. Este proceso depende de dos variables aleatorias que representan el estado de las salidas de las puertas producto positiva y negativa, que se denotan Ψ^+ y Ψ^- y que, evidentemente, sólo pueden tomar los valores 0 y 1. La probabilidad de que tomen los valores lógicos altos son, respectivamente, p_1 and p_2 y las probabilidades de que tomen los valores lógicos bajos son q_1 y q_2 , donde:

$$\begin{aligned}
 p_1 &= \exp\left(-\sum_{i=1}^{i=n} x_i^-\right) \\
 p_2 &= \exp\left(-\sum_{i=1}^{i=n} x_i^+\right) \\
 q_1 &= 1 - p_1 \\
 q_2 &= 1 - p_2
 \end{aligned}$$

La matriz de transición del proceso es:

$$\Pr[\Psi^+ = 1] = p_1$$

$$\Pr[\Psi^- = 1] = p_2$$

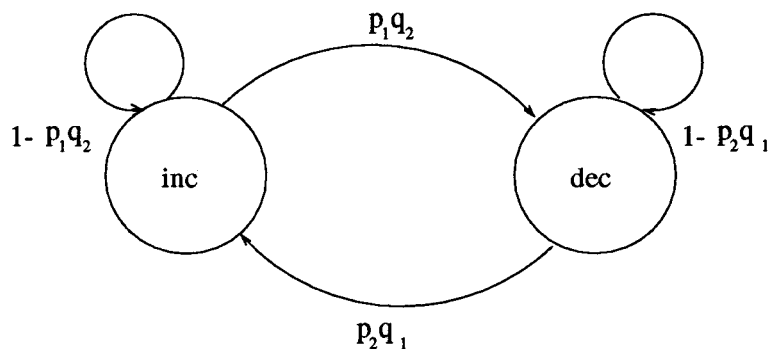


Figura 4.4: Proceso de markov.

	INC	DEC
INC	$1 - p_2 q_1$	$p_2 q_1$
DEC	$p_1 q_2$	$1 - p_1 q_2$

Los autovalores de esta matriz son 1 and $1 - p_1 q_2 - p_2 q_1 < 1$. Consiguientemente el proceso de Markov es ergódico, es decir, tiene sentido hablar de que existe un tiempo medio de permanencia en cada uno de los estados. El vector de estado promedio viene definido por las probabilidades:

$$\widehat{X} = \begin{bmatrix} \frac{p_1 q_2}{p_1 q_2 + p_2 q_1} \\ \frac{p_2 q_1}{p_1 q_2 + p_2 q_1} \end{bmatrix} \tag{4.3}$$

Esto quiere decir que si se considera un horizonte temporal suficientemente largo la probabilidad de que el estado sea "positivo" es la cantidad $\frac{p_1 q_2}{p_1 q_2 + p_2 q_1}$ y la probabi-

lidad de que sea "negativo" es la cantidad $\frac{p_2 q_1}{p_1 q_2 + p_2 q_1}$. Si al primer término le restamos el segundo, se obtiene:

$$\begin{aligned} \frac{p_1 q_2 - p_2 q_1}{p_1 q_2 + p_2 q_1} &= \frac{p_1 - p_2}{p_1 + p_2 - 2p_1 p_2} \approx \frac{p_1 - p_2}{p_1 + p_2} = \\ &= \frac{1 - e^{-(\sum_{i=1}^{i=n} x_i^+ - \sum_{i=1}^{i=n} x_i^-)}}{1 + e^{-(\sum_{i=1}^{i=n} x_i^+ - \sum_{i=1}^{i=n} x_i^-)}} \approx \frac{\sum_{i=1}^{i=n} x_i^+ - \sum_{i=1}^{i=n} x_i^-}{2} \end{aligned} \quad (4.4)$$

Para que la aproximación hecha sea admisible es preciso que se cumpla la desigualdad $2p_1 p_2 \ll p_1 + p_2$. Esto quiere decir que las cantidades positivas y negativas totales que se suman deben ser suficientemente grandes. En caso de que esto no sea así, se pueden sumar cantidades iguales positivas y negativas de forma que lo anterior se cumpla. Sin embargo esto haría que la dinámica del proceso fuese muy lenta. En realidad este tipo de arquitectura tienen verdadera aplicación cuando el número de sumandos que se tienen es grande, de forma que la desigualdad anterior se cumple.

4.3.2 Arquitectura completa de la estructura de suma.

La estructura de suma se muestra en la figura (4.5). Los trenes de pulsos de entrada son los términos a sumar. Estos trenes de pulsos se comprimen mediante una transformación exponencial. Los trenes de pulsos comprimidos son conducidos a la entrada de la correspondiente puerta producto. El multiplexor de la figura, en función del signo del sumando, lo hace llegar a la entrada de una de las puertas producto. La entrada de la puerta que no corresponda se deja a nivel lógico alto, para no inhibir la puerta. Se supone que los signos de las entradas pueden cambiar de ahí el sentido de los multiplexores. Las salidas de las dos puertas producto son las entradas de un circuito combinacional. Este circuito comprueba si hay un pulso positivo a nivel alto y no lo hay negativo. Cada vez que esto ocurre, el circuito

genera una señal de puesta a cero del biestable. También comprueba si la salida de la puerta negativa están a nivel lógico alto y la positiva no. Cuando esto ocurre se genera una señal de puesta a uno del biestable. Si se necesita una codificación digital del resultado de la suma, el estado del biestable determinaría las señales de incremento-decremento de un contador reversible.

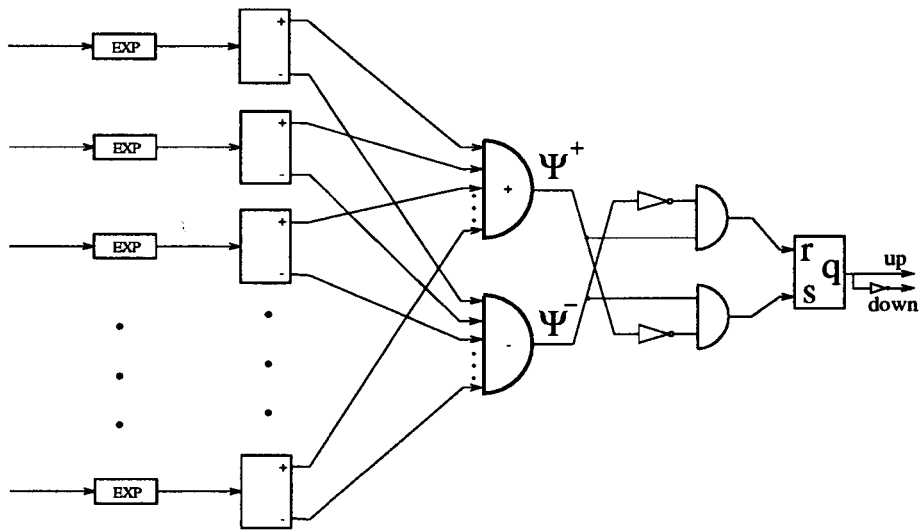


Figura 4.5: Arquitectura de suma.

4.4 Aplicación a la realización de filtros de respuesta impulsional finita.

La arquitectura que se ha descrito en los apartados anteriores puede utilizarse para realizar físicamente filtros digitales de respuesta impulsional finita. En la figura se muestra cómo se implementan estos filtros estocásticamente. Los coeficientes que definen el filtro se cargan en los registros c_1, c_2, \dots, c_n . La señal que se va a filtrar se muestrea con un periodo T que es grande en comparación con el periodo del reloj interno de funcionamiento del sistema estocástico. Cada vez que se muestrea la señal

se da una orden de desplazamiento de los contenidos de los registros i_1, i_2, \dots, i_n en los que se almacenan las entradas. El nuevo valor muestreado se almacena en el registro i_1 que ha quedado libre tras la orden de desplazamiento. Las codificaciones digitales de los coeficientes y las entradas se transforman en secuencias de pulsos estocásticos síncronos. Las entradas y los coeficientes correspondientes se multiplican por medio de puertas productos, con lo que se obtienen n trenes de pulsos estocásticos. A estos trenes de pulsos se les calcula la transformación exponencial. Las señales así generadas se separan en dos grupos distintos, dependiendo de que los signos de las entradas y los coeficientes sean iguales (sumandos positivos) o distintos (sumandos negativos). Es preciso usar los multiplexores puesto que el signo de las entradas puede cambiar. Los trenes de pulsos correspondientes a sumandos positivos se multiplican en la puerta positiva y los correspondientes a sumandos negativos se multiplican en la puerta negativa. A partir de los dos trenes de pulsos obtenidos a la salida de las dos puertas se calculan las señales de puesta a uno y puesta a cero del biestable de salida. En esta aplicación es preciso calcular una codificación digital del resultado, por lo que el biestable determina las señales de incremento y decremento de un contador reversible.

Para comprobar la arquitectura propuesta se ha hecho una simulación funcional de un filtro digital estocástico. El filtro en cuestión es un pasobanda tipo Yulewalk de orden ocho. Las frecuencias inferior y superior del filtro son, respectivamente, 0.1 y 0.25 veces la frecuencia de muestreo.

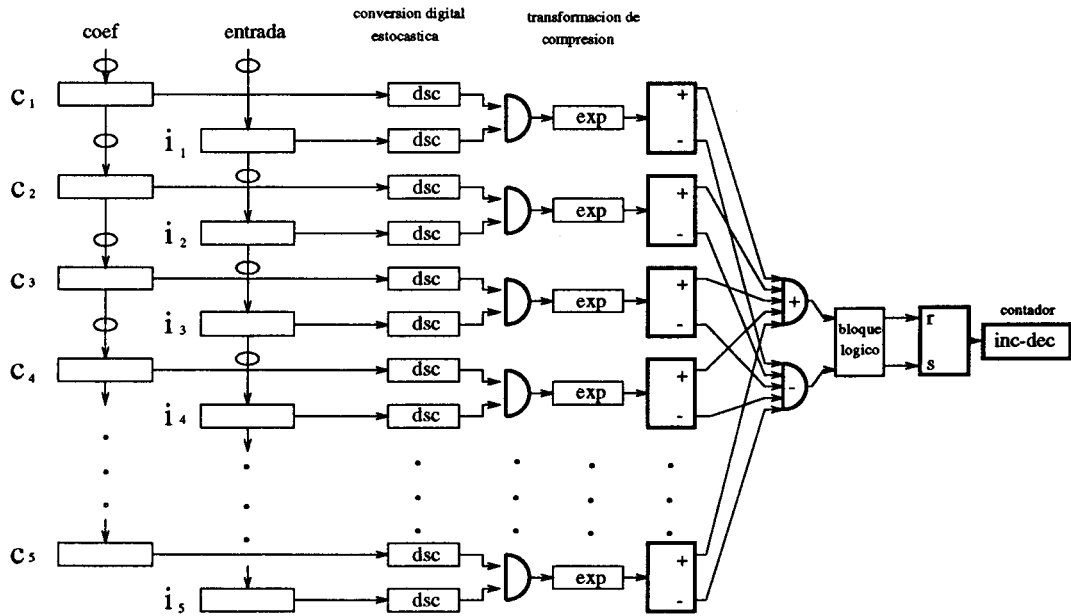


Figura 4.6: Filtro estocástico.

b_0	.00358	b_{14}	.14268	b_{28}	-.02452
b_1	.01365	b_{15}	.15043	b_{29}	-.03094
b_2	.02880	b_{16}	.10986	b_{30}	-.02838
b_3	.04439	b_{17}	.03985	b_{31}	-.01845
b_4	.05132	b_{18}	-.03055	b_{32}	-.00443
b_5	.03978	b_{19}	-.07493	b_{33}	.01057
b_6	.00497	b_{20}	-.08180	b_{34}	.02224
b_7	-.04771	b_{21}	-.05726	b_{35}	.02608
b_8	-.10185	b_{22}	-.01867	b_{36}	.01956
b_9	-.13508	b_{23}	.01436	b_{37}	.00401
b_{10}	-.12882	b_{24}	.02964	b_{38}	-.01470
b_{11}	-.07825	b_{25}	.02603	b_{39}	-.02807
b_{12}	.00299	b_{26}	.01116		
b_{13}	.08683	b_{27}	-.00931		

Tabla 1: Coeficientes del Filtro

El filtro era, inicialmente, de respuesta impulsional infinita. Se han sustituido sucesivamente las respuestas del filtro en los instantes $t-1$, $t-2$, ..., hasta que se ha observado que los coeficientes que afectan a las entradas retrasadas en el tiempo se hacen muy próximos a cero. Esto quiere decir que se aproxima el filtro de respuesta impulsional infinita por uno de respuesta impulsional finita. Los coeficientes así obtenidos se muestran en la tabla (4.4).

Para codificar los coeficientes del filtro se utilizan 8 bits, 7 para el módulo y uno para el signo. Para llevar a cabo la transformación exponencial se han tomado el término lineal y el cuadrático del desarrollo en serie de Taylor. De esta forma se llega a un compromiso entre las necesidades de resolución de los coeficientes y la sencillez de los circuitos.

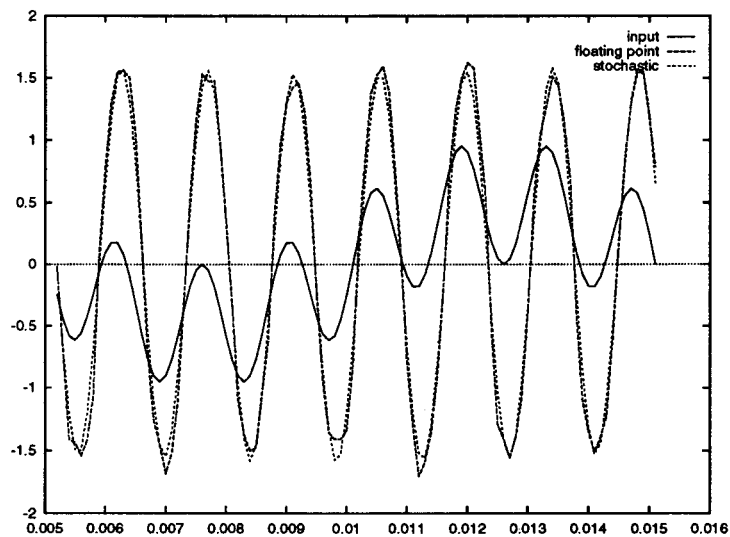


Figura 4.7: Respuesta del Filtro.

En las figuras (4.7) y (4.8) se observa la respuesta del filtro ante una combinación de señales sinusoidales y se compara con la respuesta del mismo filtro calculando las operaciones con un procesador de coma flotante. El horizonte de evolución del filtro estocástico es de 1000 periodos, aunque con 500 también se obtienen resultados

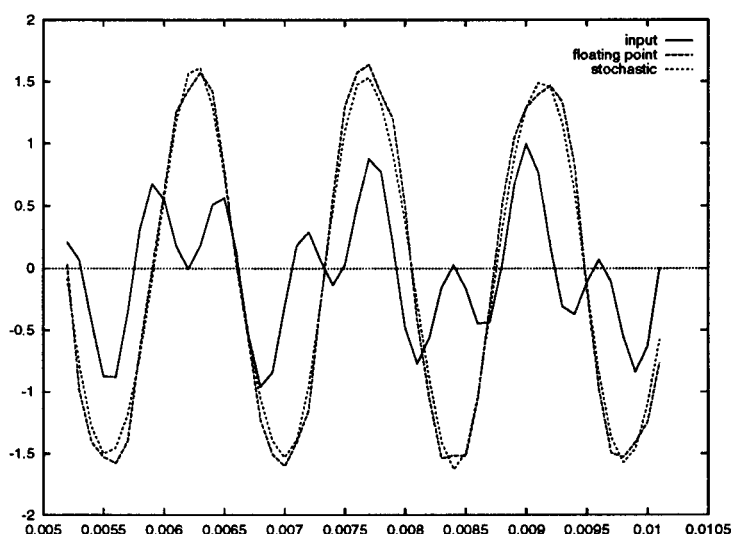


Figura 4.8: Respuesta del Filtro.

satisfactorios.

Se ha diseñado un dispositivo de lógica programable concebido para la realización de filtros digitales estocásticos. Se ha utilizado un dispositivo de capacidad media (TI-1020), siendo 9 el número de coeficientes que caben él, figura (4.10). El circuito se ha concebido de forma que se puedan poner en cascada varios dispositivos para implementar filtros que tengan un número de coeficientes mayor que nueve. Para ello se disponen de señales expansoras de entrada y salida en las puertas producto que evalúan la suma. Los estados del registro de almacenamiento correspondiente a la última entrada se sacan por ocho pines de salidas, de forma que se puedan poner en cascada con los registros de otro dispositivo de lógica programable.

Con uno de estos dispositivos se ha implementado un filtro paso alto Butterworth de cuarto orden. La frecuencia inferior del filtro es 0.1 veces la frecuencia de muestreo. En la figura (4.9) se observa la respuesta del filtro ante una combinación de señales sinusoidales y se compara con la respuesta del mismo filtro calculando las operaciones con un procesador de coma flotante. El horizonte de evolución del filtro

estocástico es de 1000 periodos.

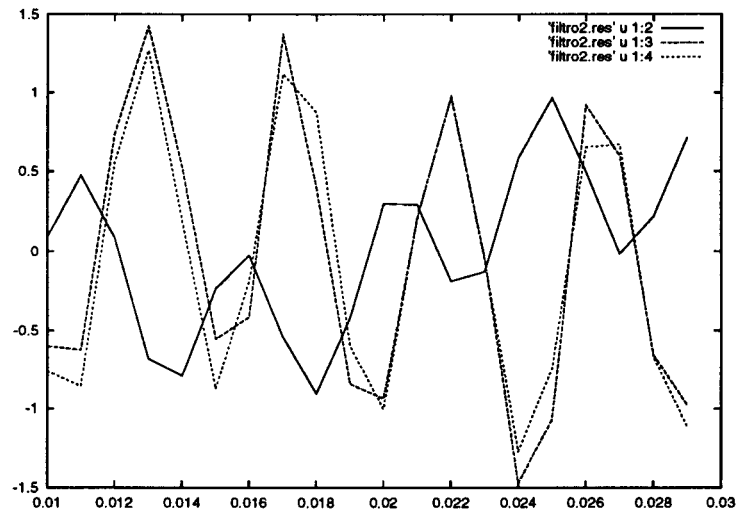


Figura 4.9: Respuesta del Filtro.

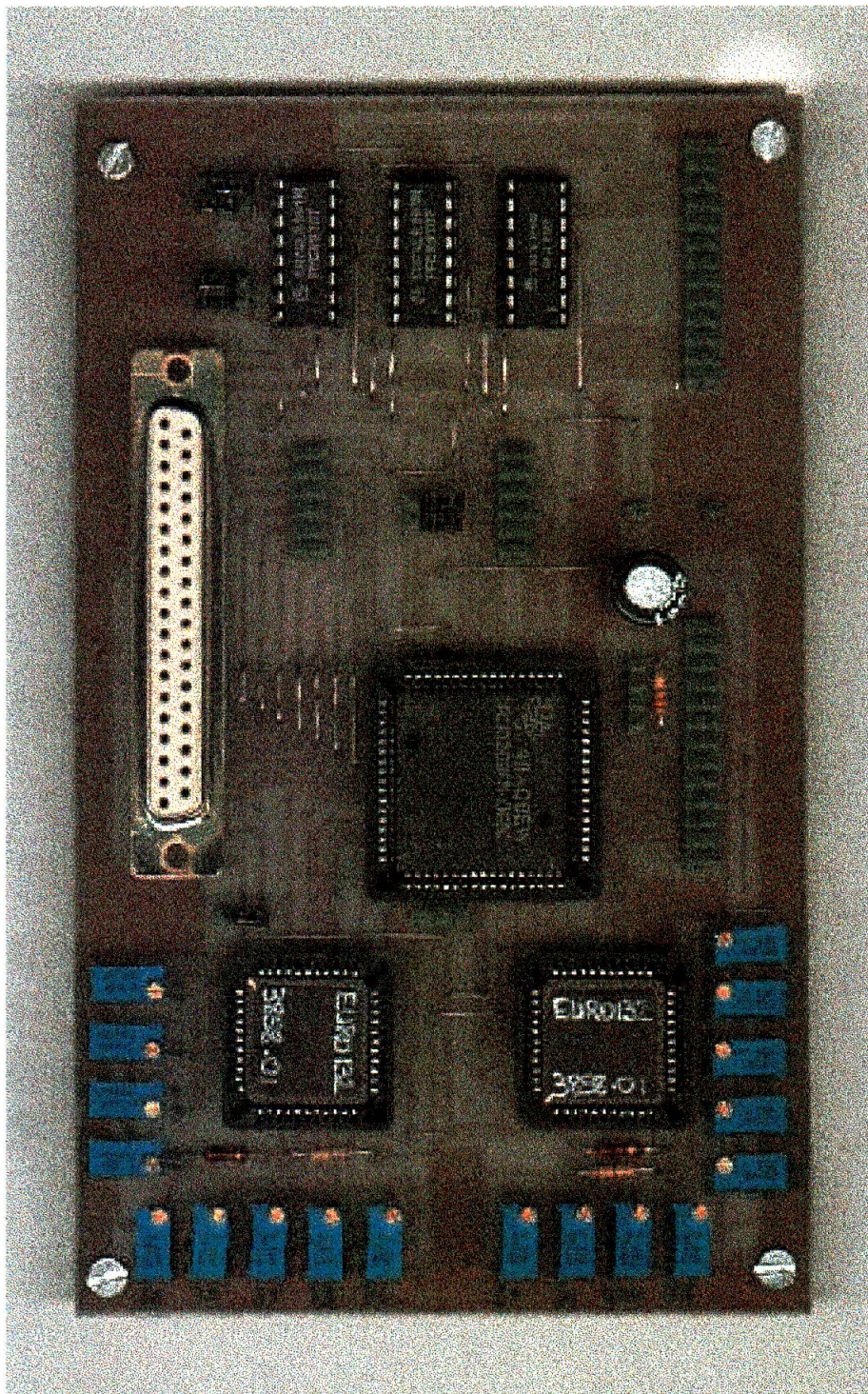


Figura 4.10: Filtro digital sobre una FPGA y generadores de números aleatorios.

Capítulo 5

Generación de trenes de pulsos estocásticos.

5.1 Introducción

La potencia del cálculo estocástico se pone de manifiesto en sistemas con gran cantidad de operaciones. Es en estos casos donde la ejecución en paralelo de todas ellas demuestra una clara ventaja frente a otras estructuras, como puede ser la microprocesadora.

En este mismo sentido apunta la utilización de las Redes Neuronales. El rendimiento máximo de éstas se obtiene en aplicaciones en las cuales se necesitan un número elevado de neuronas. Igualmente ocurrirá en el caso de cualquier otra estructura en la cual el número de términos sumandos sea crítico, como es en la realización de filtros digitales.

Partiendo de la necesidad de los sistemas estocásticos de integrar el mayor

número posible de sumandos, pasaremos a analizar el Perceptrón descrito en capítulos anteriores [6]. El tamaño de la oblea de silicio utilizada para su realización microelectrónica ha sido de $25mm^2$. Si de aquí quitamos el espacio ocupado por los pads de entrada y salida del circuito, queda una superficie útil de $16mm^2$. Como se introdujeron 6 neuronas con un número de 5 pesos más el umbral, el total de sumandos, incluidos en este área es de **30**.

El hecho de que en una cantidad de área de silicio significativamente grande solo puedan incluirse un número muy limitado de términos hace que para cualquier aplicación, por muy sencilla que resulte, necesite varios de estos circuitos integrados. En este caso particular fue necesario utilizar 2 de estos circuitos para llevar a cabo un problema de control. El caso de los filtros digitales, resulta similar, ya que se ha conseguido simular el comportamiento de un **Paso Banda** con 40 términos, obteniéndose resultados satisfactorios. Sin embargo, cuando se introduce uno de estos filtros en un dispositivo real solo se podrá tener en cuenta un número muy limitado de términos. En particular, en el ejemplo que se comenta en el capítulo anterior el límite fue de 9 en un dispositivo de lógica programable de capacidad media ¹.

Analizando la Red Neuronal podemos darnos cuenta como el 90% del área ocupada corresponde a lo que se ha llamado el circuito generador de trenes de pulsos. Es decir, registro de almacenamiento de pesos, generador de números aleatorios, comparadores y bloques XOR, mientras que el 10% restante corresponde a lo que es el circuito neuronal en sí.

En este capítulo se hace un análisis previo de algunos métodos analíticos de generación de números aleatorios. Las conclusiones obtenidas en este apartado serán utilizadas en apartados posteriores. En segundo lugar, se describe el desarrollo de un nuevo circuito generador de trenes de pulsos a partir de un valor de tensión dado,

¹FPGA Texas Instrument 1020A con 547 macroceldas

atendiendo a consideraciones de tamaño y simplicidad con objeto de que pueda ser integrado fácilmente en silicio.

La nueva estructura propuesta basa su funcionamiento en la relación de frecuencias entre un circuito oscilador y otro muestreador para generar así un tren de pulsos con unas características aleatorias dadas. Esta nueva estructura incluye un mecanismo regulador de la densidad de pulsos que sirva para hacer corresponder a los diferentes pesos, trenes de pulsos de diferentes densidades. Por otra parte, el circuito propuesto utiliza exclusivamente transistores MOS, la mayoría de ellos con unas dimensiones de $l = 2\mu$ y $w = 2\mu$ lo que hace posible una total integración, reduciendo al mismo tiempo el diseño a un tamaño mínimo.

5.2 Generación de señales aleatorias.

Desde hace varias décadas, se ha estudiado la forma de generar secuencias aleatorias a partir de métodos computacionales. El más popular generador en uso en la actualidad fue introducido por D. H. Lehmer en 1949 [29].

La secuencia de números aleatorios deseada se obtiene a partir de la expresión general

$$X_{n+1} = (a \cdot X_n + c) \bmod m \quad n > 0$$

donde

$m > 0$ es el módulo

$0 < a < m$ es el multiplicador

$0 < c < m$ es el incremento

$0 < X_0 < m$ es el valor inicial

siendo todos los valores, números enteros positivos.

Esta es la llamada **secuencia congruencial lineal**. Por ejemplo, la secuencia obtenida cuando $m = 10, X_0 = a = c = 7$ es

$$7, 6, 9, 0, 7, 6, 9, 0, \dots$$

En este ejemplo, la secuencia no es realmente aleatoria debido a la elección de los parámetros. Sin embargo, sirve para ilustrar una propiedad de todas las secuencias congruentes del tipo $X_{n+1} = f(X_n)$, según la cual, siempre acaban entrando en un bucle de un determinado periodo. La elección de éstos resulta, pues, fundamental para conseguir una secuencia con buenas características aleatorias. En particular, es deseable elegir un valor de m lo mas elevado posible, ya que el periodo no tendrá mas elementos que m .

El método original de generación de Lehmer incluía el caso $c = 0$, mientras que la idea de tomar $c \neq 0$ para obtener periodos mas largos es debido a Thomson [30] e, independientemente, a Rotenberg [31]. Sin embargo, el hecho de tener un periodo largo no es condición suficiente. La elección correcta del multiplicador garantiza el barrido completo de la secuencia en todo el periodo.

Una discusión mas amplia sobre la elección de parámetros viene recogida ampliamente en la bibliografía [28]. En particular, para el caso $a = 1$ la secuencia resultante no tiene características realmente aleatorias.

$$X_{n+1} = (X_n + c) \bmod m \quad n > 0$$

o bien,

$$X_{n+1} = (X_0 + n \cdot c) \bmod m \quad n > 0$$

Algunos generadores citados en la literatura son:

- Generador A: $X_0 = 0$ $a = 3141592653$ $c = 2718281829$ $m = 2^{35}$.
- Generador B: $X_0 = 0$ $a = 2^7 + 1$ $c = 1$ $m = 2^{35}$.
- Generador C: $X_0 = 314159265$ $a = 2^{18} + 1$ $c = 1$ $m = 2^{35}$.

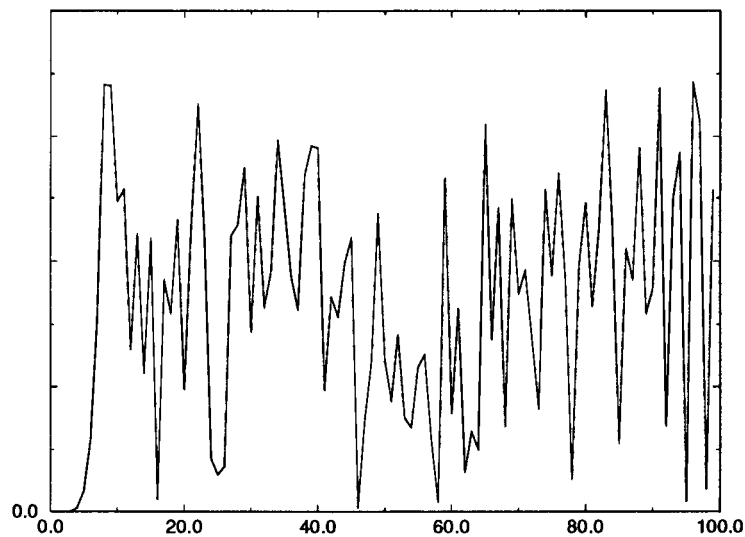


Figura 5.1: Generador aleatorio tipo A.

Mientras que los dos primeros parecen tener un buen comportamiento aleatorio, el último muestra una clara tendencia periódica, por lo que debe ser rechazado.

Todos estos métodos son fácilmente reproducibles por computador, sin embargo, su realización electrónica es bastante complicada a menos que se disponga de un microprocesador. Sería interesante disponer de un circuito sencillo capaz de producir unos resultados similares a los anteriores.

Si partimos de una onda cuadrada periódica a una frecuencia f_1 , que es muestreada a otra frecuencia f_2 menor, el resultado se puede obtener mediante la siguiente

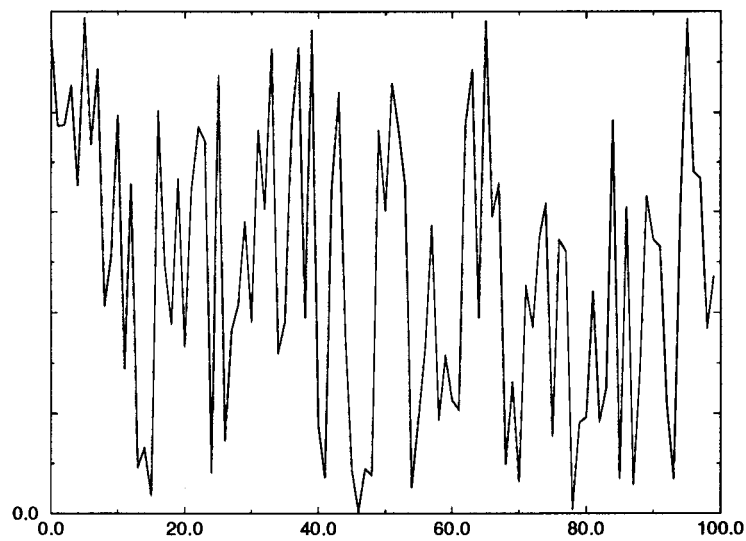


Figura 5.2: Generador aleatorio tipo B.

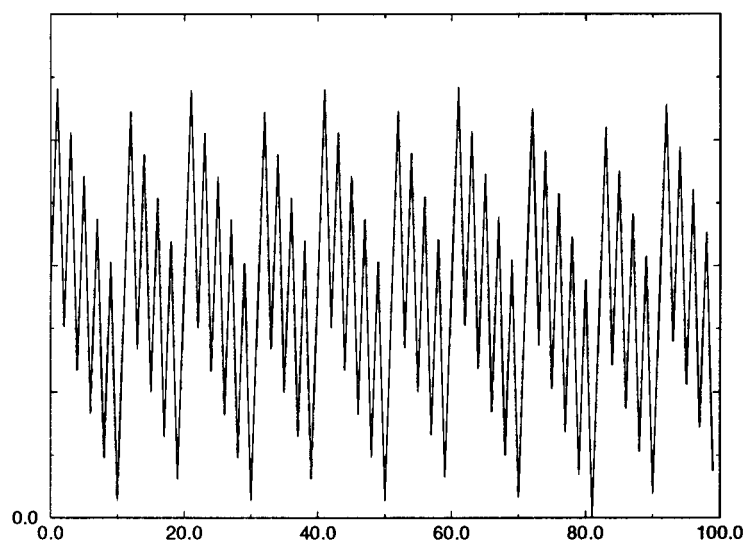


Figura 5.3: Generador aleatorio tipo C.

expresión.

$$X_{n+1} = (X_n + c) \bmod m \quad n > 0$$

donde los parámetros tienen el siguiente significado.

- m es el periodo correspondiente a f_1
- $c > m$ es el periodo correspondiente a f_2
- X_0 es el desfase inicial medido en unidades de tiempo

Como se puede comprobar, esta ecuación es similar a la de la **secuencia congruente lineal**, con la particularidad de ser $a = 1$ y de estar definida para valores de c mayores que m . Además, por regla general, m y c serán números reales, al ser así los valores de las frecuencias. En la figura (5.4) se observa el significado físico de los parámetros.

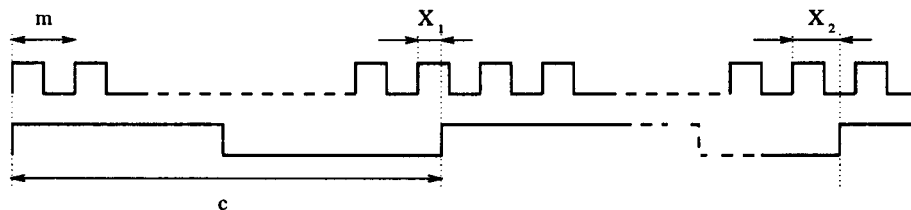


Figura 5.4: Representación del muestreo de una onda periódica.

Por comodidad, se puede tomar $m = 1$, de forma que la función módulo dará el parte decimal de $(X_n + c)$. Veamos un ejemplo.

Para el valor $c = 103.43921$ se obtiene el resultado de la figura (5.5).

Esto significa que la frecuencia de la onda de muestreo es algo menor de 103 veces la de la otra onda. Si en la expresión anterior se elige otro valor cualquiera de c se puede observar el mismo comportamiento periódico que en el generador tipo C

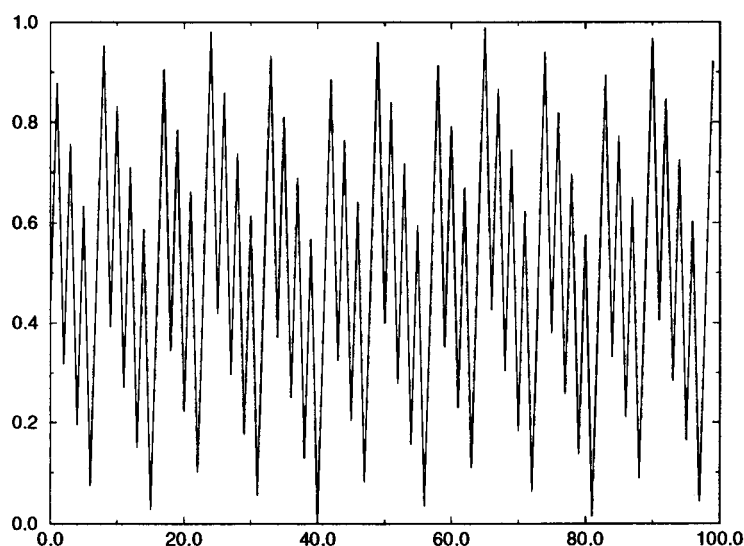


Figura 5.5: Resultado del muestreo de una señal periódica.

definido anteriormente. La conclusión sería, por tanto, que no se pueden usar como generadores de números aleatorios.

Sin embargo, si nos permitimos sumar a $(X_n + c)$ un valor aleatorio ψ , del orden del periodo de la onda menor, es decir, del orden de m , el resultado sería el mostrado en la figura (5.6). Se puede comprobar que se consigue el mismo efecto si se supone un ruido acoplado a la onda periodica de forma que se introduce una cierta aleatoriedad en el instante de conmutación del oscilador. Los efectos de esta aleatoriedad se irán acumulando de forma que, tras un determinado tiempo aparece una incertidumbre en la conmutación del orden del periodo de la onda.

$$X_{n+1} = (X_n + c + \psi) \bmod m \quad n > 0$$

Para medir el caracter aleatorio de una señal se definen los coeficientes de autocorrelación como:

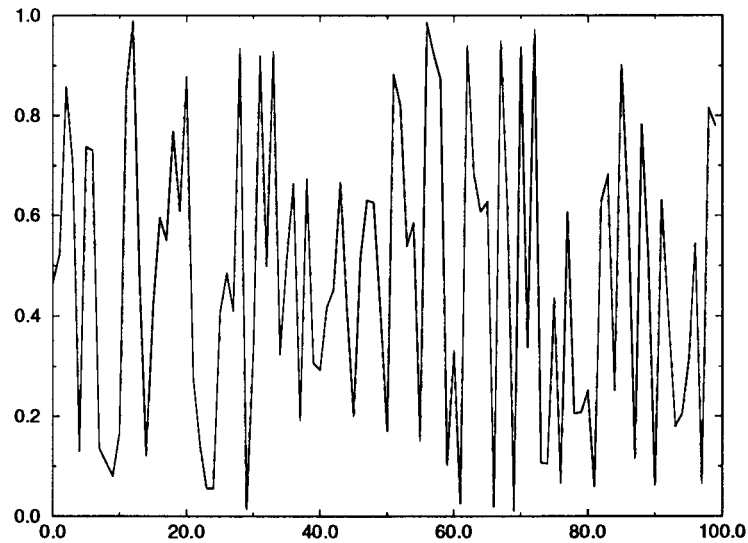


Figura 5.6: Muestreo de una señal periódica con un ruido superpuesto.

$$r_i = \frac{\sum u_k \cdot u_{k-i}}{n} - \frac{\sum u_k^2}{n} \quad (5.1)$$

Estos coeficientes se obtienen a partir del producto de una secuencia de números aleatorios y la resultante de desplazar i unidades de tiempo. Para tener un valor promedio de los 20 primeros coeficientes se define el **Factor de Autocorrelación**:

$$AF = \sum_i (r_i^2/20) \quad (5.2)$$

Un valor de AF alto indicará que existe una correlación muy grande entre el número generado en un instante de tiempo y los siguientes, por lo que la secuencia en cuestión tendrá un caracter poco aleatorio.

En la figura (5.7) se muestra AF frente a $\delta(T)/T$ y Ts/T , donde estos últimos coeficientes son el porcentaje de ruido en la oscilación y la relación de frecuencias

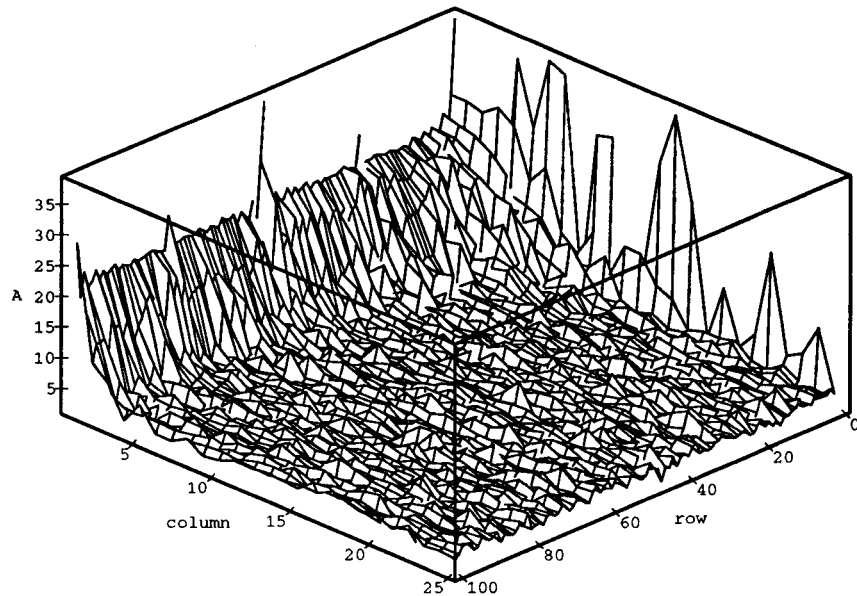


Figura 5.7: Autocorrelación en función del periodo de muestreo y porcentaje de ruido.

entre la del oscilador y el circuito de muestreo. Se puede observar que un incremento en el ruido o en la relación entre frecuencias se corresponde con una reducción en el Factor de Autocorrelación. Por ejemplo, si se tiene un porcentaje de ruido del 2% basta con coger una relación de frecuencias de 50 para asegurar un valor bajo de AF

Por otra parte, un generador con buenas características deberá repartir sus valores de forma homogénea por todo el intervalo de definición de éstos, además de tener un buen factor de autocorrelación. Una forma de ver la primera característica es calcular la media del tren de pulsos fabricado de esta manera. Si se compara el valor así generado con un valor fijo, L , perteneciente a dicho intervalo, el resultado tendrá una probabilidad L de ser un valor alto. El tren de pulsos generado tendrá una media representativa de tal valor, a partir de un horizonte de tiempo lo suficientemente grande. Si se tienen dos trenes de pulsos representativos de dos valores dados, la multiplicación de ambos trenes de pulsos daría una medida de la

incorrelación entre ellos. Es decir, el producto anterior deberá dar como resultado un nuevo tren de pulsos de media el productos de ambos valores. Estas dos pruebas serán incorporadas como forma de validación de cualquier generador de trenes de pulsos estocásticos.

5.3 Celdas generadoras de trenes de pulsos. Realización física

A continuación se describe el circuito integrado realizado, con objeto de probar nuevas estructuras generadoras de trenes de pulsos estocásticos [27].

En un área de $6mm^2$ se han repartido 12 celdas agrupadas en cuatro diseños independientes. Todas las celdas tienen como única salida el tren de pulsos resultante, mientras que las entradas varían según pertenezcan a un grupo u otro.

El primero de estos diseños fue pensado para tener un control de los pesos sobre todo el rango de variación de éstos, mediante dos tensiones analógicas, mientras que en el segundo se restringe el campo de variación de pesos a la mitad, teniendo solamente una tensión de control. Los otros dos grupos incluyen un valor ya prefijado de los pesos.

A grandes rasgos, la estructura de todas las celdas es similar, y se puede resumir en dos bloques bien diferenciados. El primero de estos es el bloque oscilador, encargado de generar una onda periódica cuadrada con una relación T_{on}/T_{off} determinada. El segundo está compuesto por un biestable tipo D, encargado de muestrear la señal anterior mediante un flanco de reloj.

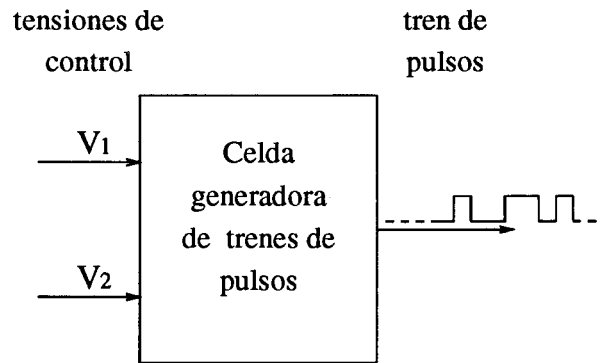


Figura 5.8: Celda generadora de trenes de pulsos.

5.3.1 Bloques osciladores

Cada bloque oscilador está compuesto por cinco inversores CMOS en cascada, capaces de oscilar a una frecuencia aproximada de 140 Mhz. Éste es el menor número de inversores que garantiza una onda cuadrada como resultado de la oscilación. Si por el contrario, se aumenta el número de inversores en la cadena, la frecuencia de oscilación bajará. La relación T_{on}/T_{off} viene dada por el tiempo de carga y descarga de las capacidades de entrada de cada uno de los inversores. Si el punto A de la cascada de inversores representados en la figura (5.9) está a nivel bajo, el transistor N del primer inversor estará cortado, mientras que el transistor P estará en conducción, cargando la capacidad del siguiente inversor. De forma análoga, si el punto A está a nivel alto, el transistor N del primer inversor estará en conducción, mientras que el transistor P estará cortado, descargando dicha capacidad. En la cadena de inversores el proceso de carga y descarga se produce de manera alternada, de forma que la tensión de salida B tendrá siempre el valor contrario a la tensión de entrada A .

Cuando los puntos A y B se unen, el circuito resultante no alcanzará un punto de trabajo estable, apareciendo una oscilación en todos los inversores. El tiempo que permanecerá a nivel bajo la señal de salida B vendrá dado por el máximo de los

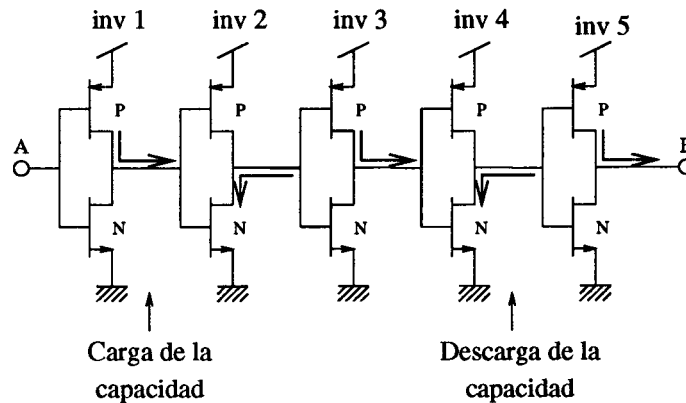


Figura 5.9: Mecanismo de oscilación a base de inversores CMOS.

tiempos de descarga de las capacidades de los inversores 1, 3 y 5 y de carga de los 2 y 4. Este tiempo define el ancho T_{off} de la señal de salida, teniéndose algo análogo para el tiempo T_{on} . Por tanto, la relación entre el tiempo en que el oscilador permanece a nivel alto y el que permanece a nivel bajo puede ser controlado estrechando los canales de uno de los transistores, lo que viene a suponer una alteración de los tiempos de carga o descarga.

En principio, sería deseable poder controlar ambos procesos intercalando dos transistores de regulación, uno P y otro N, entre los caminos de carga y descarga de uno de los inversores del oscilador. De esta forma, los valores T_{on} y T_{off} serían controlados independientemente, pudiendo alcanzar este cociente cualquier valor comprendido entre 0 e ∞ . Evidentemente, cualquier aumento en uno de los tiempos anteriores supone una disminución de la frecuencia de oscilación, y por tanto de la frecuencia de muestreo.

En la figura (5.10) se muestra la configuración descrita anteriormente. Una tensión V_p alta provocará un estrechamiento del canal del transistor P1, con lo cual se consigue un aumento del tiempo en cual el punto B permanece a nivel bajo. Igualmente, una tensión baja V_n provocará un aumento en el tiempo a nivel alto.

En el presente diseño se ha incluido un oscilador de este tipo, directamente

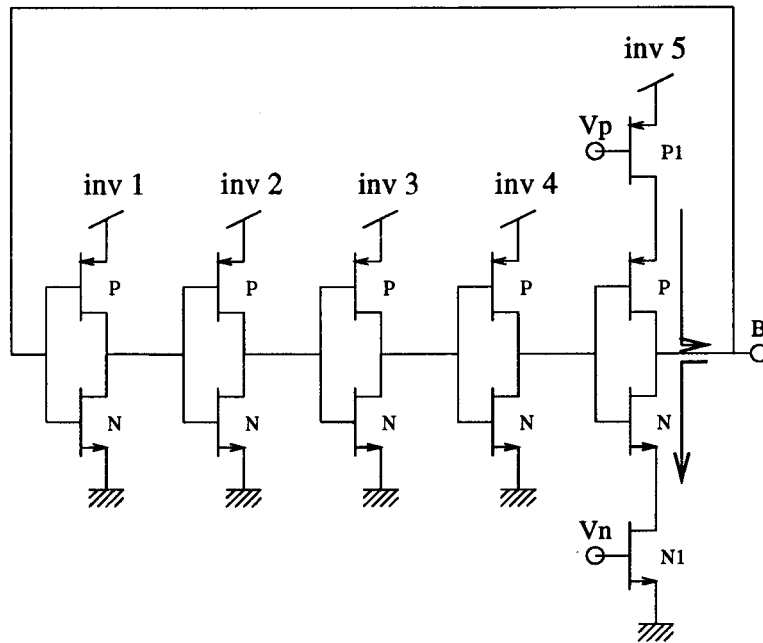


Figura 5.10: Control del ancho de pulsos en todo el intervalo.

controlado desde el exterior por dos tensiones, V_p y V_n , que llevan a ambos transistores de corte a saturación. En la figura (5.11) se muestra la curva de regulación de T_{off}/T , correspondiente a una configuración en la cual la tensión del transistor P1 permanece fijada a $0v$ mientras que la tensión del N1 se mueve en el rango $[0.7, 5v]$. Se observa que aparece una zona en la que la curva resulta bastante lineal.

Un segundo oscilador similar a éste ha sido incluido, con la diferencia de que las dos tensiones han sido fijadas ya por diseño. Para conseguir un valor $T_{on}/T_{off} = 2.58$ se ha anulado el efecto del transistor P1 conectando su puerta a $0v$, mientras que la tensión V_n se ha fijado a $1.6v$ mediante un divisor de tensiones constituido por 2 transistores Nmos saturados, figura (5.12).

En la figura (5.13) se puede observar las formas de onda obtenidas por simulación correspondientes a este último diseño.

Por otra parte, existe la posibilidad de eliminar uno de los transistores dejando al

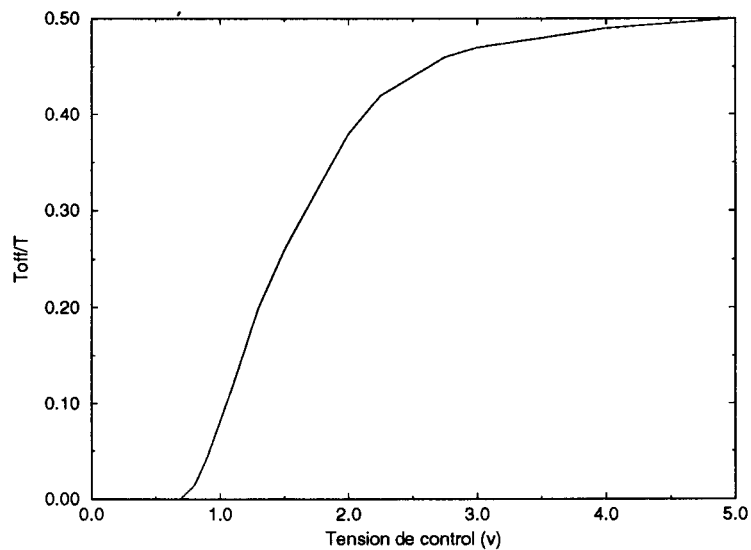


Figura 5.11: Curva de regulación de $Toff/T$ en función de V_{N1} .

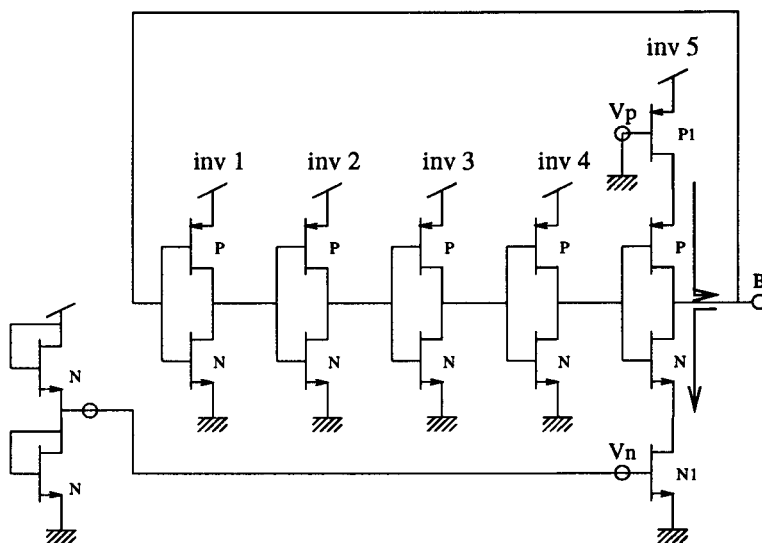


Figura 5.12: Peso fijado por divisor de tensiones.

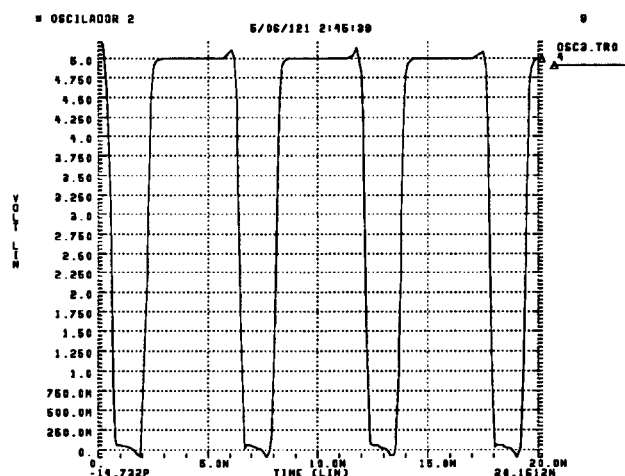


Figura 5.13: Oscilador con $T_{on}/T_{off} = 2.58$.

oscilador solo la opción de mover su relación T_{on}/T_{off} , bien en el intervalo $[0 - b]$ o en $[b - \infty]$ según se elimine el P o el N. El valor b vendría dado por las relaciones de aspecto del conjunto de los transistores. En el caso de una total simetría eléctrica se tiene $b = 1$, que se corresponde con la situación en la cual la velocidad de descarga de los mosfet es exactamente igual a la de carga.

En el integrado se han incluido 8 osciladores de este último tipo, regulados con un transistor tipo n, de forma que b se ha fijado a un valor de 0.57. El propósito de este conjunto de osciladores es conseguir 8 trenes de pulsos con probabilidades cercanas a $1/2$, de forma que el transistor n sirva para realizar un ajuste mas preciso de esta probabilidad. La agrupación de estos 8 bits constituirán un generador de bytes aleatorios. En la figura (5.14) se muestra el circuito correspondiente a estas 8 celdas mientras que en la figura (5.15) se puede ver la respuesta de este oscilador cuando la tensión aplicada a su puerta es de $5v$.

Por último, con objeto de medir la dispersión de resultados obtenidos correspondientes a celdas exactamente iguales, se han incluido dos osciladores sin transistores

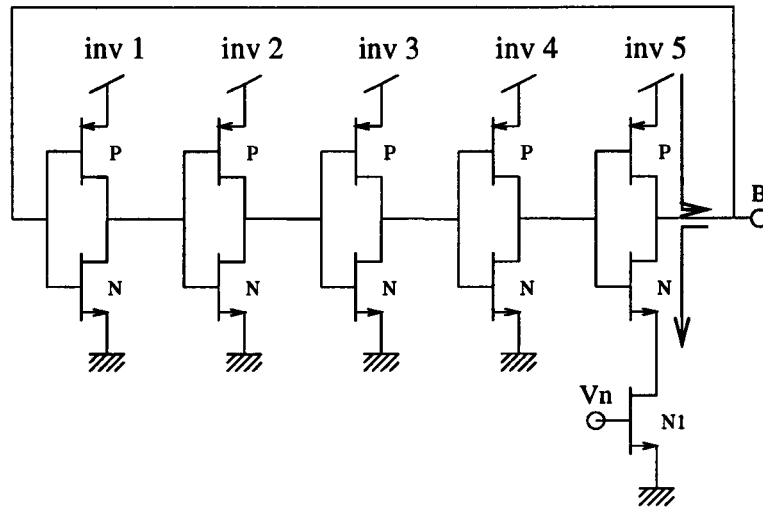


Figura 5.14: Control del ancho de pulsos en la mitad del intervalo.

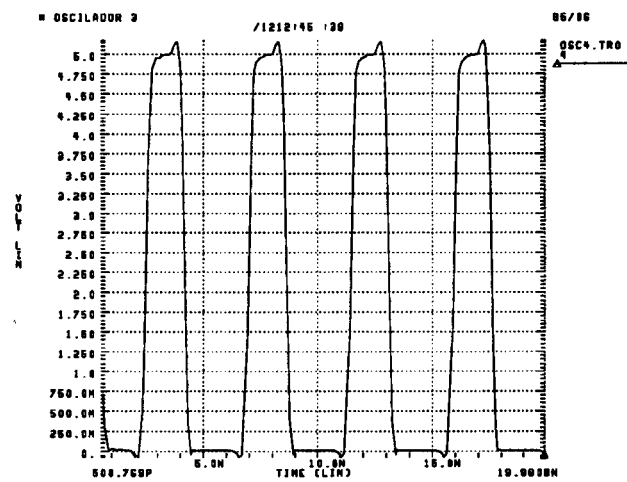


Figura 5.15: Oscilador con $T_{on}/T_{off} = 0.57$.

de regulación de ningún tipo, y cuya relación T_{on}/T_{off} se ha fijado lo mas cercano posible al valor de 1, manipulando exclusivamente las relaciones de aspecto de los transistores correspondientes a los inversores básicos.

5.3.2 Bloque muestreador.

Está compuesto por un biestable activo por flanco de subida. Es el encargado de muestrear la forma de onda resultante de cada uno de los osciladores, a una frecuencia inferior. La forma de onda resultante tendrá asociada una media en el tiempo igual a las relación T_{on}/T de la onda muestreada.

La utilización de todo dispositivo real introduce una componente de ruidos que va a hacer conmutar a los inversores dentro de un δT alrededor del instante teórico de conmutación. Este hecho va a servir para acumular en el tiempo una incertidumbre en la fase de la onda, cumpliendo el mismo papel que la variable aleatoria ψ introducida anteriormente. Dado que el ruido presente está fijado por diseño, la incognita será saber que relación de frecuencias es la adecuada para conseguir una incertidumbre acumulada en la fase igual al periodo de la onda.

El circuito descrito anteriormente ha sido diseñado utilizando el programa MAGIC, mientras que los pads han sido añadidos por medio del programa SOLO 1400. Su realización física fue encargada al fabricante de circuitos integrados ES2, via EUROCHIP.

El tamaño total de la oblea de silicio es de $6mm^2$, incluyendo pads. Sin embargo, el circuito en sí ocupa un área neta de $5.5 \cdot 10^5 \mu^2$, lo que quiere decir que la mayoría de la superficie de la oblea se encuentra vacía. En estos casos se habla de diseño "pad limited", es decir, que la longitud de la periferia se encuentra limitada inferiormente

por el número de pads que se quieren incluir.

El layout correspondiente a cada uno de los osciladores, así como al total del circuito, conteniendo las 12 celdas y biestables se muestra en las figuras (5.17 a 5.21). En este último se puede observar como las tensiones de control se introducen desde la parte superior e inferior del diseño. Una microfotografía del circuito integrado se observa en la figura (5.22).

La superficie de cada una de las celdas generadoras de trenes de pulsos es de $4.5 \cdot 10^4 \mu^2$, correspondiendo $1 \mu^2$ al oscilador y el resto al biestable. Para poderla comparar con la superficie ocupada por el circuito equivalente en la realización totalmente digital descrita, es necesario hacerlo en igualdad de condiciones. En primer lugar, hay que tener en cuenta que no se incluye ningún tipo de almacenamiento programable de pesos. Los dispositivos que realizan la misma función que estas celdas serán el comparador digital, los bloques XOR y la parte proporcional del registro LFSR. En segundo lugar está el hecho de que los dispositivos usados en el diseño digital corresponden a la librería del fabricante y están muy optimizados en área. Teniendo en cuenta que el oscilador ocupa un espacio inferior al biestable, se puede suponer que en el caso de usar componentes de librería estandar, el tamaño correspondiente a una celda sería menor que el de dos biestables.

Según lo anterior, el número de puertas de ambos diseños, y su equivalente en área será el siguiente:

Como se puede ver, la relación de áreas es de $29/3 \simeq 10$. Si añadimos el área consumida por el rutado en el diseño digital, la relación será mucho mayor. La conclusión es que el número de sumandos puede ser aumentado entre uno y dos ordenes de magnitud, es decir, que en el mismo área de silicio ocupada por la red neuronal digital de 30 pesos, $25mm^2$, se podrían introducir una cantidad aproximada al millar en el caso de usar las celdas estocásticas.

DISEÑO DIGITAL			CELDA ESTOCÁSTICA		
componente	uds.	(μ^2)	componente	uds.	(μ^2)
Comparador Digital	20	25400	Oscilador	1	1270
Bloque XOR	8	10160	Muestreador	2	2540
Parte prop. LFSR	1	1270	—		
TOTAL	29	36830	TOTAL	3	3810

Figura 5.16: Comparación de área entre ambos diseños.

Uno de los problemas que plantea la solución elegida es el del almacenamiento de los pesos. En el caso de pesos programados, basta con realizar el divisor de tensiones correspondiente para conseguir el tren de pulsos deseado. Como hemos visto, esto supone poco gasto de silicio, ya que involucra el uso de solo dos transistores. Sin embargo, para establecer una buena equivalencia con el diseño del Perceptrón descrito se necesita una estructura programable capaz de almacenar pesos. Ciertos estudios han tratado esta problemática, [32] llegando a diversas soluciones entre las cuales se pueden destacar:

1. Almacenamiento digital Tiene como ventaja la precisión en el valor se quiere almacenar, pero como inconveniente que debe adaptarse de alguna forma la codificación digital a un valor de tensiones analógica. Esto último representa un aumento considerable del área consumida que hace poco atrayente esta solución, [33].
2. Almacenamiento dinámico de pesos El valor analógico a almacenara es colocado en forma de carga Q en una capacidad C . Para evitar la perdida de carga debe haber un mecanismo de refresco de ésta que actue de forma periódica. Este tiempo vendrá dado según el porcentaje de fluctuación que se quiera para esta señal, [34], [35].
3. Cargas electricas en tecnología MNOS (metal nitride oxide silicon), [36], [37].

4. Tecnología de puertas flotantes, [38], [39], etc...

El problema del almacenamiento de pesos escapa al objetivo de esta tesis, remitiendo a la bibliografía anterior la discusión que existe al respecto.

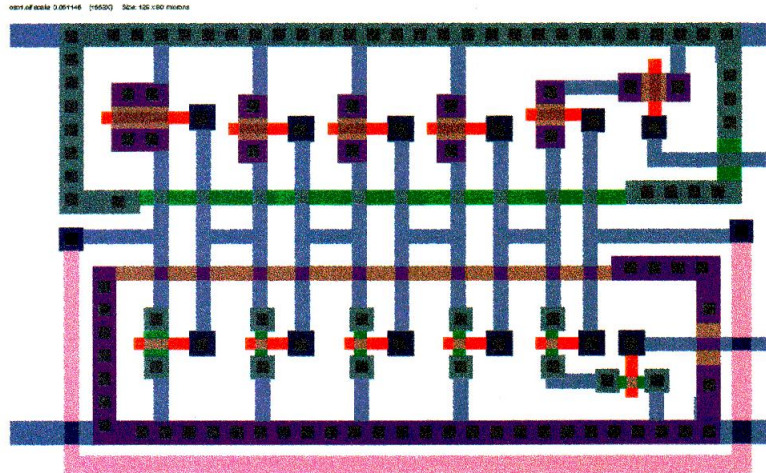


Figura 5.17: Layout del oscilador tipo 1.

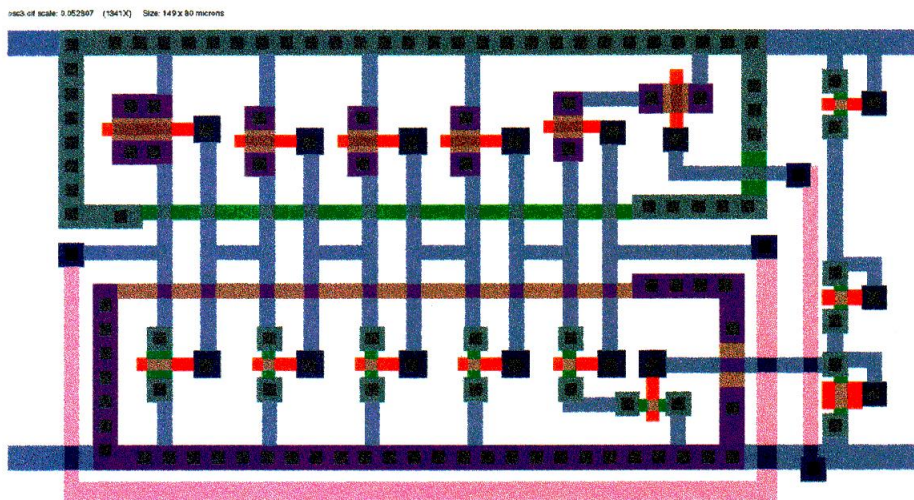


Figura 5.18: Layout del oscilador tipo 2.

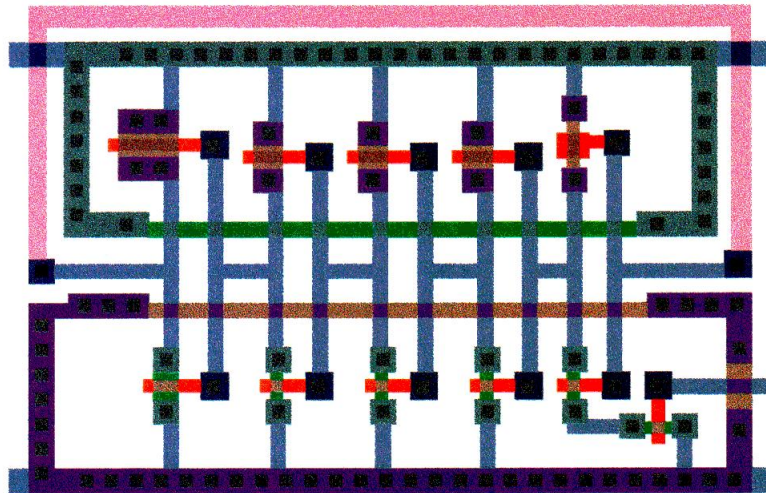


Figura 5.19: Layout del oscilador tipo 3.

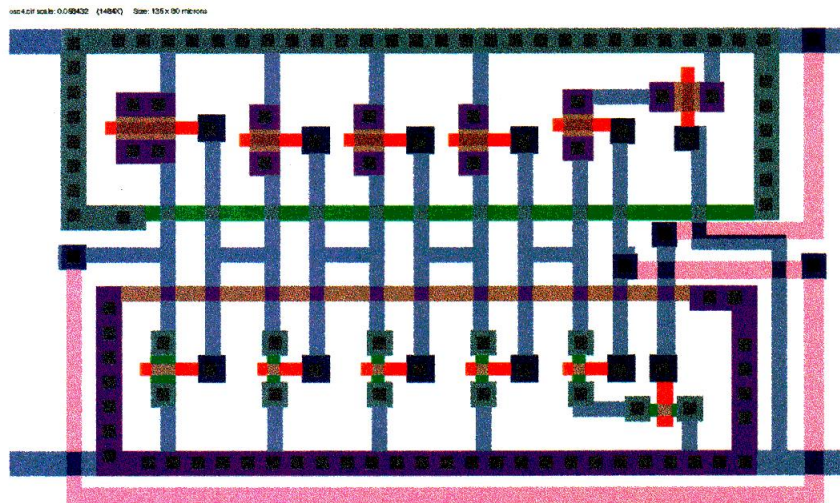


Figura 5.20: Layout del oscilador tipo 4.

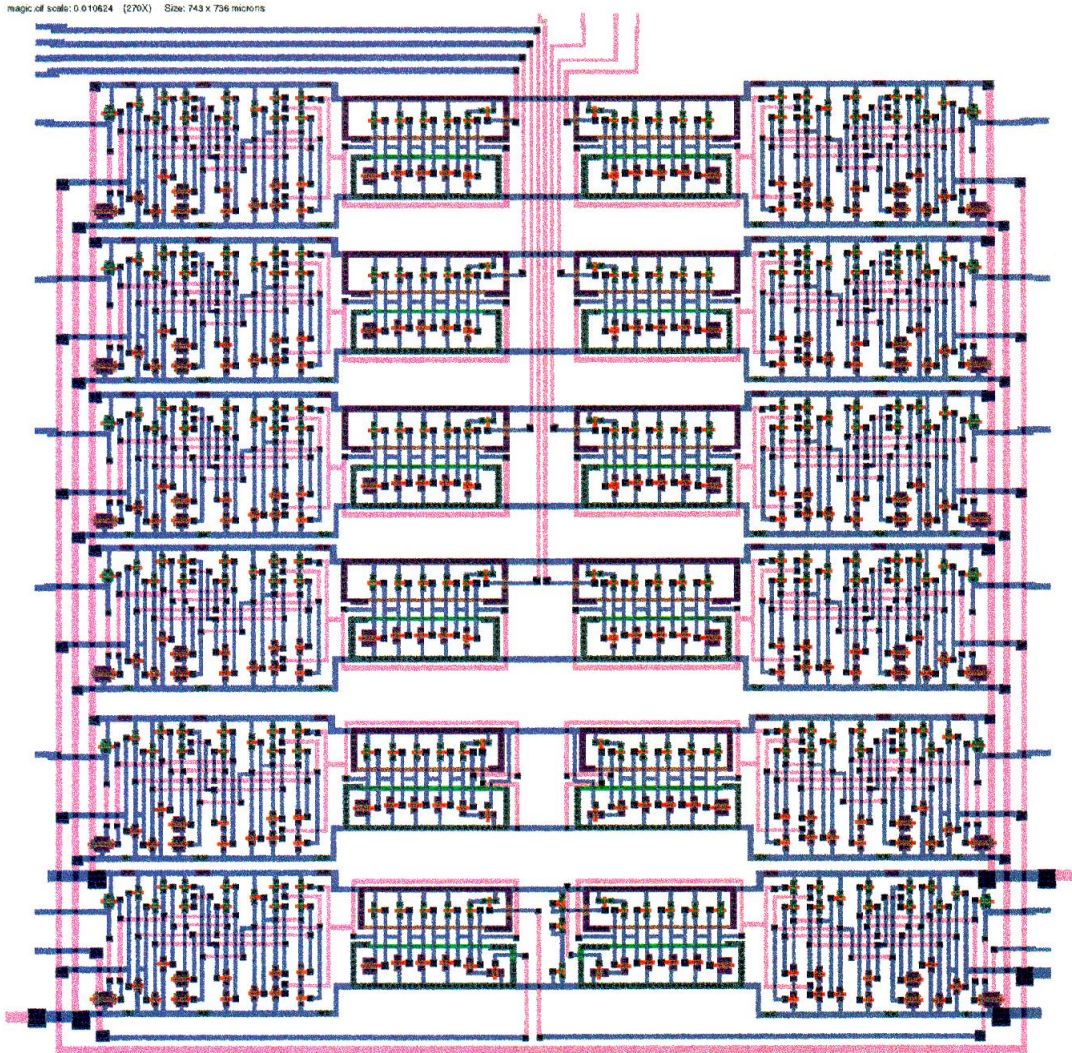


Figura 5.21: Layout del circuito total.

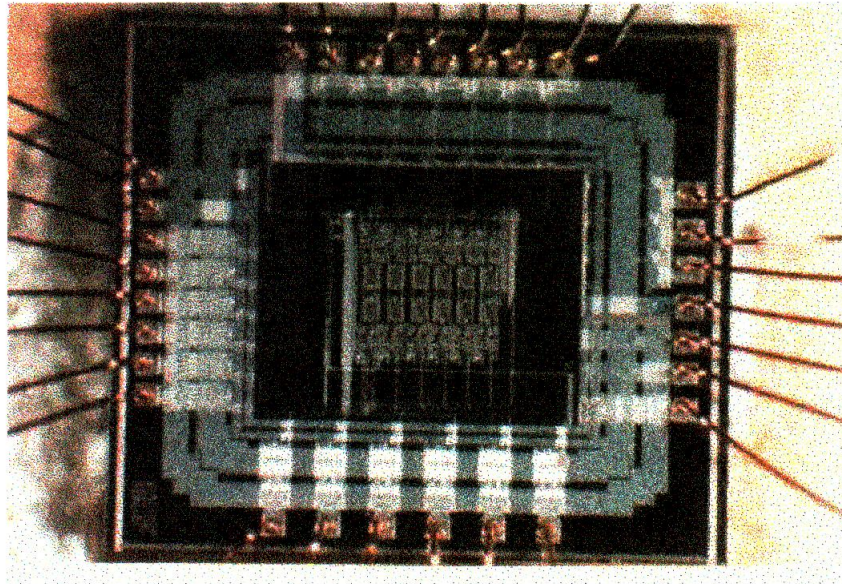


Figura 5.22: Microfotografía del circuito integrado generador de trenes de pulsos estocásticos.

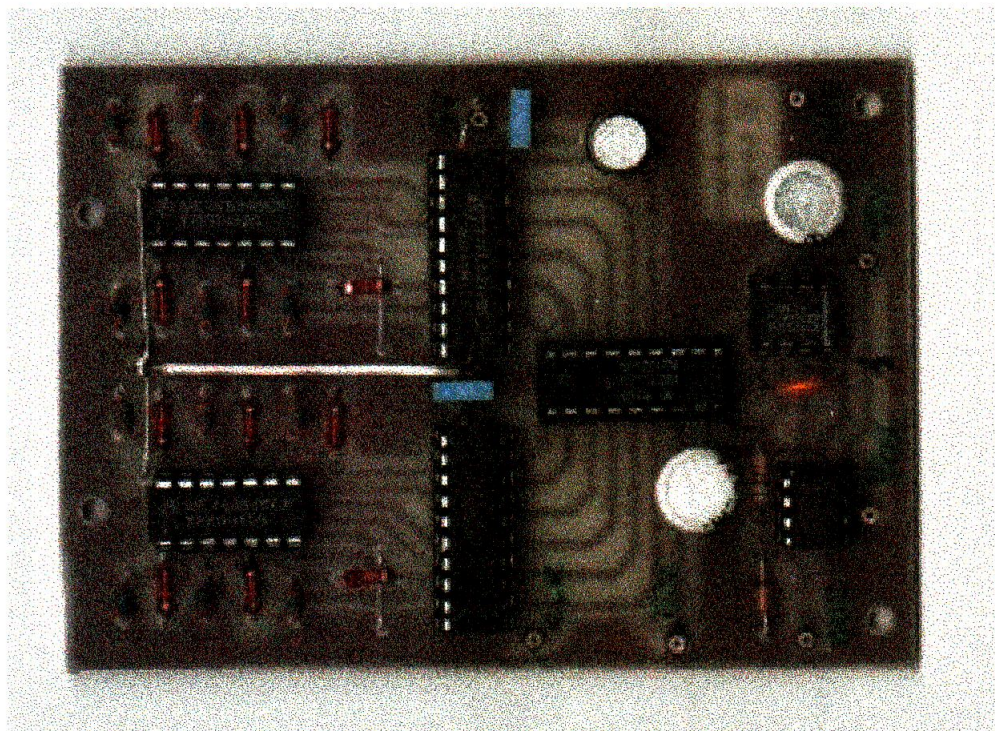


Figura 5.23: Prototipo de convertidor analógico digital.

5.4 Aplicación a la generación de números aleatorios

El circuito integrado anterior ha sido utilizado para generar números aleatorios. Ajustando las tensiones de control de las 8 celdas iguales que contiene, de forma que la forma de onda de salida tenga una relación $T_{on}/T_{off} = 1$, se consiguen otros tantos trenes de pulsos con una probabilidad asociada de $1/2$. El conjunto de estos bits forman un byte con características aleatorias en el cual todos los estados son equiprobables.

Este generador de bytes aleatorios se aplicó al filtro digital realizado en el dispositivo de lógica programable que ha sido expuesto en el capítulo anterior. De esta forma, los resultados obtenidos en el filtrado han sido comparados con aquellos otros obtenidos utilizando un generador de números aleatorios LFSR. Las curvas resultantes de la utilización de este nuevo generador consiguen una aproximación a los resultados teóricos similar a las del segundo.

El circuito integrado ha sido probado a diferentes frecuencias y con diferentes configuraciones, con objeto de obtener un conjunto de autocorrelaciones y correlaciones cruzadas que sirviesen para definir el rango de frecuencias útiles y las condiciones de funcionamiento. En principio, la frecuencia máxima de trabajo vendrá dada por el valor de la autocorrelación de los trenes de pulsos. Como se desarrolló anteriormente, a frecuencias bajas, la aleatoriedad de los pulsos estaba garantizada.

En la figura (5.24) se muestra los coeficientes de autocorrelación, de ordenes 0 a 20, de una serie de secuencias de números aleatorios generados utilizando ocho celdas oscilantes ajustadas a 0.5, así como de otra generada por un registro LFSR.

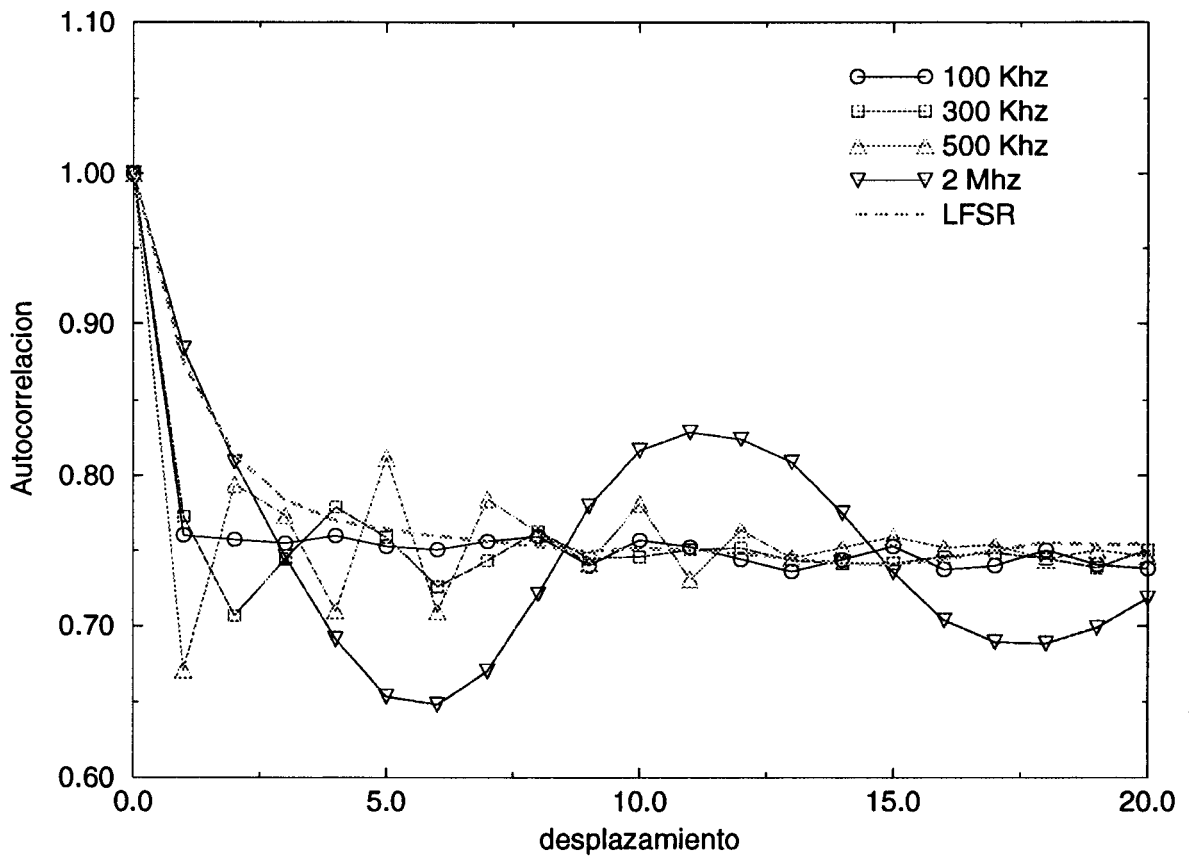


Figura 5.24: Comparación entre las autocorrelaciones de diferentes trenes de pulsos.

El coeficiente de autocorrelación se ha definido en este caso como

$$r_i = E[x_k \cdot x_{k-i}] = \frac{\sum u_k \cdot u_{k-i}}{n} \quad (5.3)$$

El valor máximo que alcanzará será con desplazamiento i igual a cero, siendo éste

$$E[x_k^2] = \frac{\sum u_k^2}{n} = \frac{1}{3} \quad (5.4)$$

Para deducir este valor se ha supuesto que se tiene una función de densidad constante en el intervalo $[0, 1]$. Por otro lado, el valor mínimo se alcanza cuando ambas secuencias son incorreladas, cumpliéndose que

$$E[x_k \cdot x_{k-i}] = E[x_k] \cdot E[x_{k-i}] = 0.5 \cdot 0.5 = \frac{1}{4} \quad (5.5)$$

Si se multiplican ambos valores por 3 se consigue un intervalo de $[1, 0.75]$. Para un desplazamiento cero se tendrá una autocorrelación de 1, mientras que conforme se aumenta el desplazamiento se tiende al valor 0.75 .

El comportamiento de una secuencia de valores generados por un LFSR se interpreta de la siguiente forma. Para desplazamientos pequeños los valores de autocorrelación son muy elevados, lo cual es de esperar dado el carácter determinista de este tipo de generador. Cuando este desplazamiento se hace del orden del tamaño del registro ya no quedará ninguno de los bits originales, sino que todos han sido recalculados, por lo que la autocorrelación bajará a valores aceptables.

Para una frecuencia de 100 KHz. la secuencia de números generados muestra un comportamiento altamente aleatorio, ya que desde el primer coeficiente se tienen valores en torno a 0.75 . Si se va aumentando la frecuencia de muestreo, se observa una oscilación que se corresponde a la aparición de un comportamiento periódico en la secuencia de números. Esta característica empieza a ser importante a partir de los 500 KHz. pudiendose decir que éste es un límite en el cual se obtienen resultados comparable a los del LFSR. Por último, se muestra el resultado obtenido a una frecuencia de muestreo muy cercana a la de oscilación. El carácter periódico es claramente predominante.

Si se toma como aceptable el punto de trabajo de 500 khz., la relación de frecuencia entre la onda del oscilador y la de muestreo resulta ser de 300, mayor que en el caso de la tarjeta convertidora con componentes discretos. Por otro lado, las simulaciones que permiten suponer la frecuencia de oscilación en, aproximadamente, 150 Mhz. utilizan unos parámetros SPICE suministrados por el fabricante de circuitos integrados digitales ES2 para elementos de su librería. La realización de un diseño que incluye estos elementos permitirá suponer un aumento en el rendimiento real de los componentes definidos, y por tanto de la frecuencia de oscilación.

En el siguiente capítulo se hará uso de este circuito para la realización de un circuito convertidor analógico estocástico (CAE).

Capítulo 6

Conversión de señales basada en la lógica estocástica.

6.1 Introducción

En los procesadores estocásticos, todas las señales son trenes de pulsos estocásticos, por tanto, las entradas deben ser codificadas de esta manera. En los sistemas reales, las señales son analógicas o digitales y en consecuencia deben ser convertidas a pulsos estocásticos. De igual manera, los resultados de dichos procesadores estocásticos deben sufrir la conversión contraria.

En la bibliografía aparecen diversas realizaciones que incluyen tanto convertidores Digitales–Estocásticos como Estocásticos–Digitales [3], [6]. En particular, un contador digital es usado para promediar el número de pulsos de un tren estocástico que existen en un periodo de tiempo dado. De igual manera, en tales realizaciones se consiguen trenes de pulsos representativos de un valor digital sin más que comparar éste con el valor resultante de un registro LFSR.

Por el contrario, en la bibliografía recopilada no aparece ningun método de conversión entre señales analógicas y estocásticas, debiéndose pasar previamente por una codificación digital. Este aspecto limita en gran medida el uso de sistemas estocásticos al campo de los sistemas digitales. Como ejemplo, basta recordar que el Perceptrón comentado anteriormente fue desarrollado para ser aplicado al control de un convertidor resonante en el que tanto las variables de estado $i_e v_c$ como la de control v_o son señales analógicas, [14].

En la figura (6.1) se esquematiza el planteamiento del problema de las conversiones con señales estocásticas.

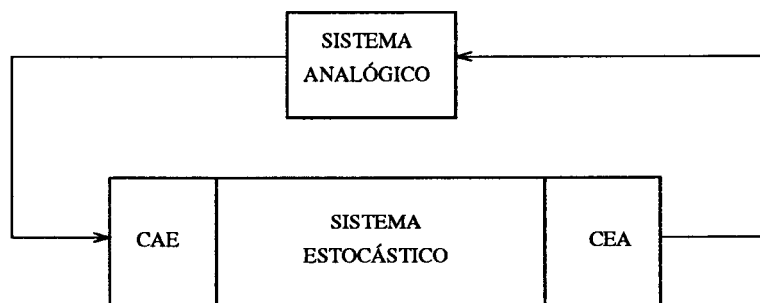


Figura 6.1: Conexión entre sistemas analógicos y estocásticos.

A continuación se realizará un estudio estadístico de las señales estocásticas, del cual se extraerán algunas conclusiones útiles, que servirán para establecer criterios de diseño de un convertidor estocástico-analógico muy simple.

Posteriormente se desarrollará el prototipo de un convertidor analógico-estocástico utilizado en la puesta en funcionamiento del Perceptrón anterior. La estructura propuesta se basa en la comparación entre una secuencia síncrona de números estocásticos y el valor en cuestión. Sin embargo, en la generación de estos números aleatorios no se utiliza la técnica del Registro de Desplazamiento Realimentado (LFSR), sino aquella que sirvió de base para la estructura expuesta en la generación de trenes de pulsos estocásticos, es decir, la relación de frecuencias entre un circuito oscilador y otro muestreador; método desarrollado en esta tesis.

Por último, partiendo de resultados obtenidos en las estructuras anteriores se desarrollará un convertidor analógico–digital y otro digital–analógico basados en técnicas estocásticas.

6.2 Estudio estadístico de las señales estocásticas.

Para calcular la densidad espectral de potencia de un tren de pulsos estocásticos es preciso, primero, dar una expresión anítica de esta señal. Para ello se van a emplear dos funciones distintas, la primera de las cuales es determinista y dependiente de una variable continua, mientras que la segunda es aleatoria y discreta.

La primera de estas funciones (rectángulo de anchura T) se define como:

$$u_k(t) = \begin{cases} 1 & kT < t < (k+1)T \\ 0 & (k+1)T < t < \infty \\ 0 & -\infty < t < kT \end{cases}$$

donde T es el periodo del reloj.

La función aleatoria y discreta (χ) se define:

$$\chi_k: \begin{cases} Pr[\chi_k = 1] = p \\ Pr[\chi_k = 0] = q = 1 - p \end{cases}$$

Una secuencia de trenes de pulsos estocásticos síncronos se puede definir como la suma de una serie de términos que se expresan como el producto de un rectángulo y una variable aleatoria discreta dicotómica (χ). La expresión analítica es:

$$x(t) = \sum_{k=-\infty}^{k=\infty} u_k(t)\chi_k \quad (6.1)$$

La función (6.1) representa una señal aleatoria, por lo que para calcular su densidad espectral de potencia hay que calcular su función de autocorrelación y hallarle a ésta su función transformada de Fourier. La función de autocorrelación asociada al tren de pulsos estocástico se puede calcular como:

$$R_{xx}(\tau) = E[x(t)x(t - \tau)] \quad (6.2)$$

es decir, como la esperanza matemática del producto de la propia señal multiplicada por ella misma desplazada τ segundos en el tiempo. La función de autocorrelación depende del desplazamiento temporal τ que es la variable de integración a la hora de calcular la transformada de Fourier. Desarrollando la expresión (6.2) queda:

$$R_{xx}(\tau) = E[x(t)x(t - \tau)] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=-N}^{k=N} \sum_{l=-N}^{l=N} \frac{1}{T} \int_{-\infty}^{\infty} u_k(t)u_l(t - \tau)dt E[\chi_k\chi_l]$$

Esta expresión, en apariencia complicada, se simplifica mucho si tenemos en cuenta que la integral de convolución de las funciones $u_k(t)$ y $u_l(t)$ no es más que una función triangular centrada en $\tau = (k - l)T$ (es decir, para este valor de τ la función vale 1) y de anchura $2T$ (desplazando τ T segundos a derecha o izquierda la función ya vale 0). Por otra parte, la esperanza matemática $E[\chi_k\chi_l]$ toma valores distintos, según sea $k = l$, en cuyo caso vale p , o $k \neq l$, en cuyo caso vale p^2 . Por tanto el cálculo de la función de autocorrelación queda reducido a sumar los triángulos representados en la figura, ponderando el triángulo central con una cantidad p y el resto con la cantidad p^2 . En la figura se ha representado también la función resultante de esta

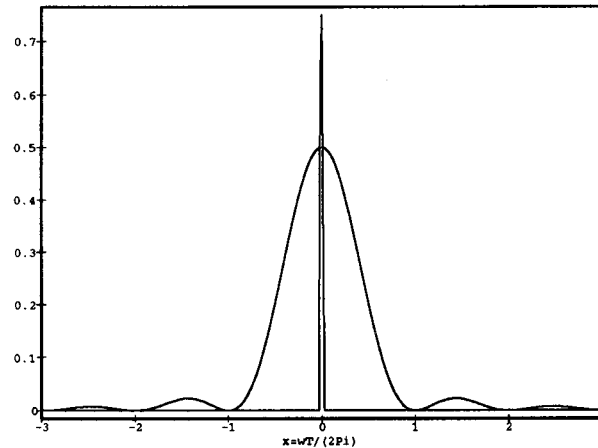


Figura 6.2: Espectro de un tren de pulsos estocásticos.

superposición. Teniendo en cuenta que esta función es la superposición de una función constante resulta que la densidad espectral de potencia del tren de pulsos estocásticos es:

$$|S(\omega)| = 2\pi p^2 \delta(\omega) + 2pqT \frac{1 - \cos(\omega T)}{(\omega T)^2} \quad (6.3)$$

que se representa en la figura (6.2).

6.3 Conversión estocástica analógica.

El espectro de una señal estocástica, que se ha deducido, consta de una parte discreta y una parte continua. La parte discreta del espectro representa la componente de continua de la señal y es la parte que hay que conservar filtrándola. La parte continua del espectro representa un ruido superpuesto que hay que eliminar que, a diferencia

del caso anterior, tiene una potencia que no está concentrada en unas frecuencias determinadas, sino que está distribuida en un continuo de frecuencias. Por este motivo no es necesario exigir que el filtro tenga una pendiente de caída importante. Lo que es preciso imponer es que la potencia total de ruido integrada por el filtro sea inferior a la resolución requerida.

Para determinar los criterios de dimensionamiento del filtro se va suponer que la dinámica de la conversión es lenta comparada con el periodo del reloj de la conversión a tren de pulsos (para poder estimar la componente de continua de un tren de pulsos es preciso considerar un número grande de ciclos de reloj). Esto supone que la parte continua del espectro $|S(\omega)|$ permanece prácticamente constante en el intervalo de frecuencias de integración del filtro, por lo que se puede hacer la siguiente aproximación:

$$|S(\omega)| \simeq 2\pi p^2 \delta(\omega) + pqT \quad (6.4)$$

La potencia integrada por el filtro paso bajo es:

$$P \simeq 2\pi p^2 + 2\omega_{max} pqT \quad (6.5)$$

donde ω_{max} representa la pulsación máxima de un filtro ideal, es decir, de pendiente de caída infinita, equivalente al filtro real. La equivalencia consiste en que ambos filtros integren la misma potencia. El caso más desfavorable es aquél en el que la potencia del ruido es máxima, lo cual ocurre para $p = q = 0.5$. La potencia filtrada debe ser inferior a la asociada al bit menos significativo, ya que en caso contrario no tendría sentido usar un convertidor de n bits. Imponiendo esta condición, se tiene:

$$2\pi 2^{-2n} > \frac{\omega_{max} T}{2} \quad (6.6)$$

es decir:

$$\omega_{max} < 2^{-2(n-1)} \omega_{clk} \quad (6.7)$$

donde n representa el número de bits del convertidor.

Si se toma $n = 8$ y una frecuencia de reloj $f_{clk} = 10MHz$ se llega a $f_{max} = 610Hz$. En una señal PWM el primer armónico estaría situado en la frecuencia $f_1 = 2^{-8} 10MHz = 39KHz$. La potencia del primer armónico es prácticamente la misma que la de la componente de continua. Por tanto, para tener una resolución de 8 bits, habría que atenuar la potencia del primer armónico al menos 2^{-8} veces su valor original. Si se emplea un filtro de primer orden (una red RC), la frecuencia de corte del filtro habría que colocarla más dos décadas a la izquierda del primer armónico, con lo que la frecuencia de corte sería inferior a 390Hz.

6.4 Conversión analógica estocástica.

El arquitectura propuesta para un circuito convertidor analógico estocástico se muestra en la figura (6.3). Consta de un generador de bytes aleatorios, cuyos valores son introducidos en un convertidor Digital-Analógico (CDA), de forma que a la salida son comparados con el valor analógico a convertir. El resultado será una secuencia de bits cuya probabilidad asociada es $Vin/Vmax$, siendo Vin la entrada analógica y $Vmax$ el valor máximo alcanzable por ésta.

Las propiedades de esta secuencia de bits estocásticos serán las de la secuencia

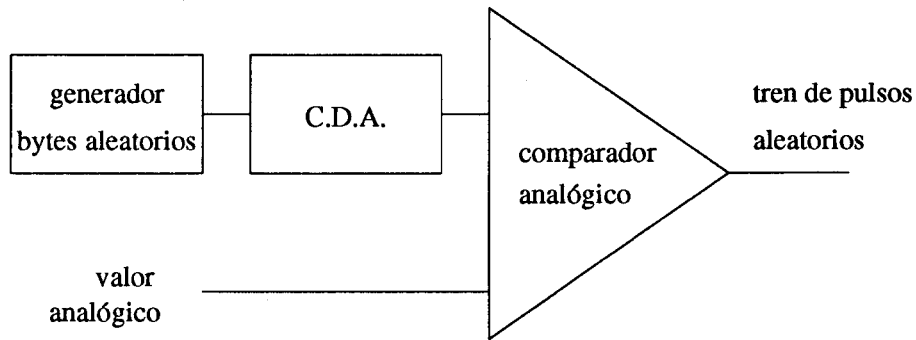


Figura 6.3: Arquitectura propuesta para la conversión analógica estocástica.

de bytes aleatorios. Para garantizar un buen resultado se requiere que la segunda tenga una autocorrelación baja y que esté uniformemente distribuida por todo el intervalo.

A continuación se pasa a describir el circuito electrónico correspondiente a un prototipo realizado con elementos discretos. El generador de bytes aleatorios consta de un grupo de celdas aleatorias básicas que son muestreadas, produciendo otros tantos trenes de pulsos síncronos. La arquitectura interna de una celda aleatoria básica esta compuesta por un oscilador a alta frecuencia y un circuito muestreador a una frecuencia inferior. El principio de funcionamiento de este circuito ha sido estudiado en un capítulo anterior con profundidad. El oscilador consta de un inversor Schmitt-Trigger con una realimentación positiva formada por una red RC y que se muestra en la figura (6.4). El resultado es una señal cuadrada periódica con una relación T_{on}/T_{off} determinada por los valores de la resistencia y la capacidad, así como por los parámetros del ciclo de histeresis (V_{ih} y V_{il}). Cuando la tensión del condensador es baja, la salida del inversor es alta, produciéndose el proceso de carga. La situación inversa provocará la descarga del condensador.

Los osciladores fueron contruidos mediante inversores Schmitt-Trigger MC14584, resistencias de 2k2 y capacidades de 300pF. El circuito muestreador lo compuso un conjunto de biestables integrados en el dispositivo SN74LS374. La tarjeta de prue-

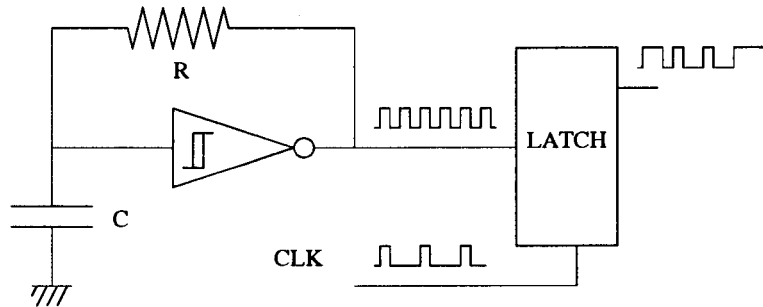


Figura 6.4: Esquema de una celda aleatoria.

bas consta de 12 módulos generadores de bits aleatorios con probabilidad de 0.5. El conjunto de estos bits forman un registro cuyo contenido representa un valor aleatorio de 12 bits. Este valor es introducido en un convertidor digital analógico (DAC1222LCN) en cuya salida aparecerá el correspondiente valor analógico, ver figura (5.23).

Dadas las características de los trenes de pulsos que la definen, es decir, con probabilidad 0.5 de que en un momento dado se encuentren a nivel alto o bajo y siendo incorrelados entre ellas, el valor aleatorio tendrá probabilidad uniforme de alcanzar cualquier valor en dicho intervalo. La comparación entre esta secuencia aleatoria y el valor a convertir (CA311) dará como resultado un tren de pulsos de características estocásticas y valor promedio representativo de éste.

Con los valores anteriores de R y C se alcanzó una frecuencia de oscilación de 1 Mhz., y una frecuencia máxima de muestreo de 20 Khz., es decir, 2 ordenes de magnitud inferior para asegurar la aleatoriedad. En caso de haber utilizado el circuito integrado desarrollado en la sección anterior, capaz de dar 8 trenes de pulsos aleatorios ajustados a una densidad 1/2 se hubiese conseguido una frecuencia de muestreo de 500 Khz.

En nuestra aplicación se ha medido un nivel de ruido del 2%, con lo cual, eligiendo una relación de frecuencias de 50 se asegura un nivel bajo de AF. La figura

(6.5) representa la superposición de 50 ondas muestreadas sucesivamente por un osciloscopio digital. Se puede observar como al cabo de 20 ondas la incertidumbre en la fase llega a ser bastante importante.

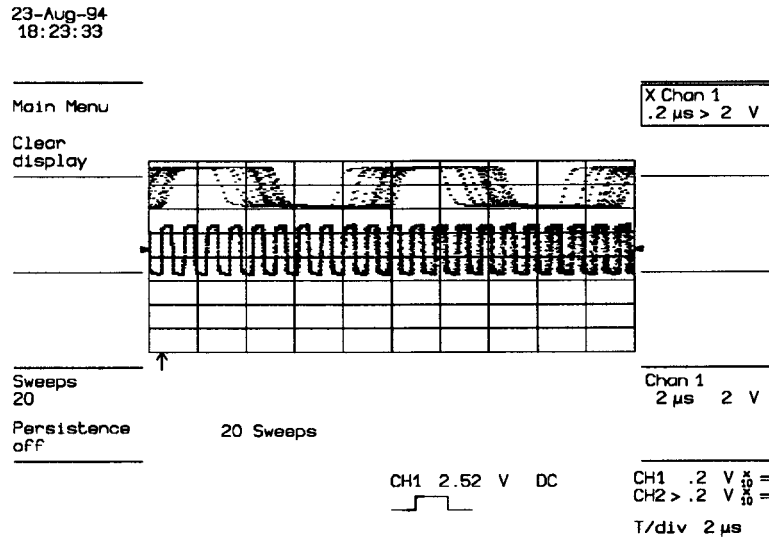


Figura 6.5: Medida en osciloscopio de la incertidumbre acumulada en la oscilación.

En la figura (6.6) se muestra el valor resultante de la conversión analógica del generador de bytes aleatorios. Se puede ver que si se compara dicha señal con un valor de referencia dado se obtiene el tren de pulsos requerido.

En (5.23) se expone una fotografía del prototipo anteriormente descrito.

Las ideas aquí expuestas fueron aplicadas al problema de la superficie de control usando el Perceptrón descrito en el capítulo introductorio, en el cual se requería un dispositivo previo de conversión de valores analógicos a estocásticos.

La red representa un perceptrón con 3 entradas, 5 neuronas ocultas y 1 salida. Todas las neuronas son puramente estocásticas y utilizan como función no lineal la función signo. Los pesos de la red han sido adaptados para implementar una superficie de decisión.

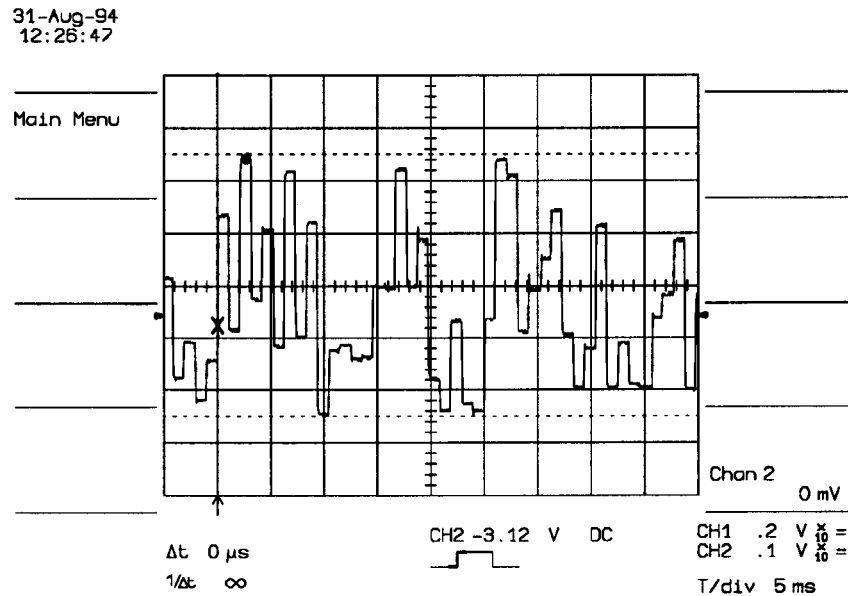


Figura 6.6: Señal aleatoria de entrada al comparador.

Para poner la red en funcionamiento han sido necesarias tres tarjetas convertidoras Analógico-Estocásticas, generando trenes de pulsos cuyas densidades son proporcionales a otros tantos valores analógicos.

En la figura (6.7) se muestran dos trenes de pulsos estocásticos correspondientes a dos entradas analógicas cuyos valores son 3v y 1v.

La figura (6.8) corresponde a la salida de la red neuronal cuando se mantiene constante a 1v una de las entradas. Si el vector de entrada pertenece a la región interior de la superficie de decisión, la salida de la red será igual a cero, mientras que si pertenece al exterior, la salida será 1. Las otras dos entradas fueron variadas desde el valor cero hasta una densidad igual a 0.125.

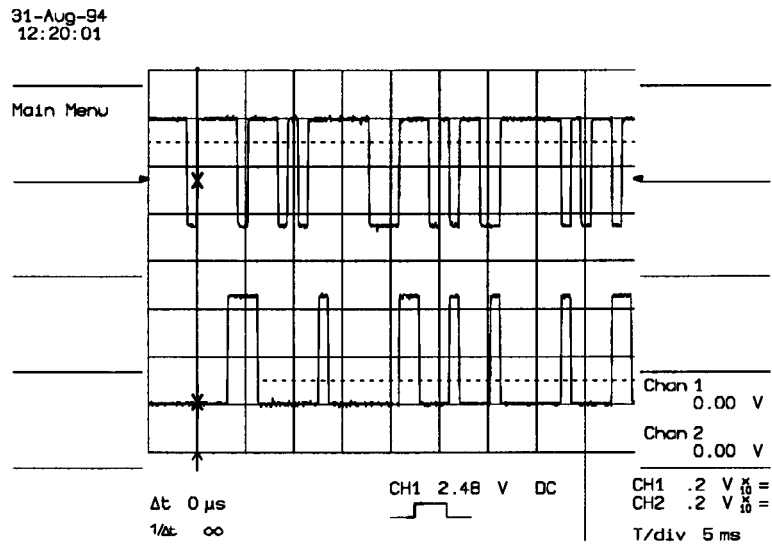


Figura 6.7: Salidas correspondientes a dos niveles analógicos.

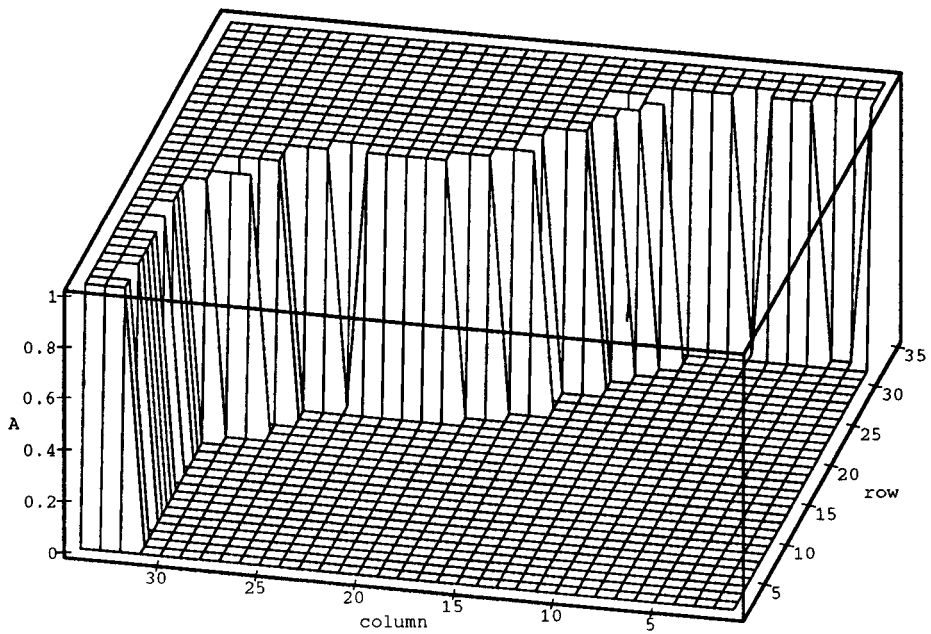


Figura 6.8: Respuesta de la red neuronal.

6.5 Conversión digital analógica y analógica digital basada en lógica estocástica.

La conversión de valores analógicos a digitales puede llevarse a cabo con tecnologías VLSI estrictamente digitales si se añaden algunos circuitos analógicos simples en el exterior del circuito integrado. A partir de la codificación digital de un número se genera una secuencia de pulsos representativos de ese valor digital. Esta codificación se filtra externamente, obteniéndose su valor medio. Este valor medio se compara con la referencia analógica (valor analógico que se quiere convertir a digital), y el resultado de la comparación entra dentro del circuito integrado. Esta señal gobierna el funcionamiento de un circuito combinacional de aproximaciones sucesivas que actualiza el valor digital cuya secuencia de pulsos se está calculando. Cuando el conjunto deja de evolucionar, en el registro se obtiene la codificación digital del valor analógico.

Si lo que se desea es realizar una conversión digital-analógica basta con utilizar la señal de salida del filtro paso-bajo.

La codificación en forma de pulsos podría conseguirse por medio de un PWM. Utilizar una codificación en forma de trenes de pulsos estocásticos presenta ventajas que van a detallarse seguidamente.

6.5.1 Ventajas de la conversión a pulsos estocásticos.

La señal que genera un PWM es periódica y, por consiguiente, tiene un espectro de potencia discreto, estando la potencia concentrada en la frecuencia 0 (componente de continua que es la que realmente se quiere obtener), en la frecuencia base de funcionamiento del PWM (la del reloj dividida por 2^n , donde n es el número de bits

del convertidor) y en sus múltiplos enteros. En la figura (6.9) se muestra el desarrollo en serie de Fourier de una señal PWM de T/T_{on} dado. El espectro de potencia será el cuadrado de esta señal. Para que el filtro paso bajo cumpla su función, la atenuación del filtro a la frecuencia del reloj debe ser lo suficientemente grande como para que la potencia resultante sea inferior a la potencia correspondiente a la componente de continua asociada al bit menos significativo. Sólo de esta forma se puede conseguir convertir el valor analógico con un grado de resolución igual al número de bits digitales. Esto quiere decir que la pendiente de caída del filtro paso bajo debe ser lo suficientemente grande como para que a la frecuencia del primer armónico de la señal éste se atenúe suficientemente.

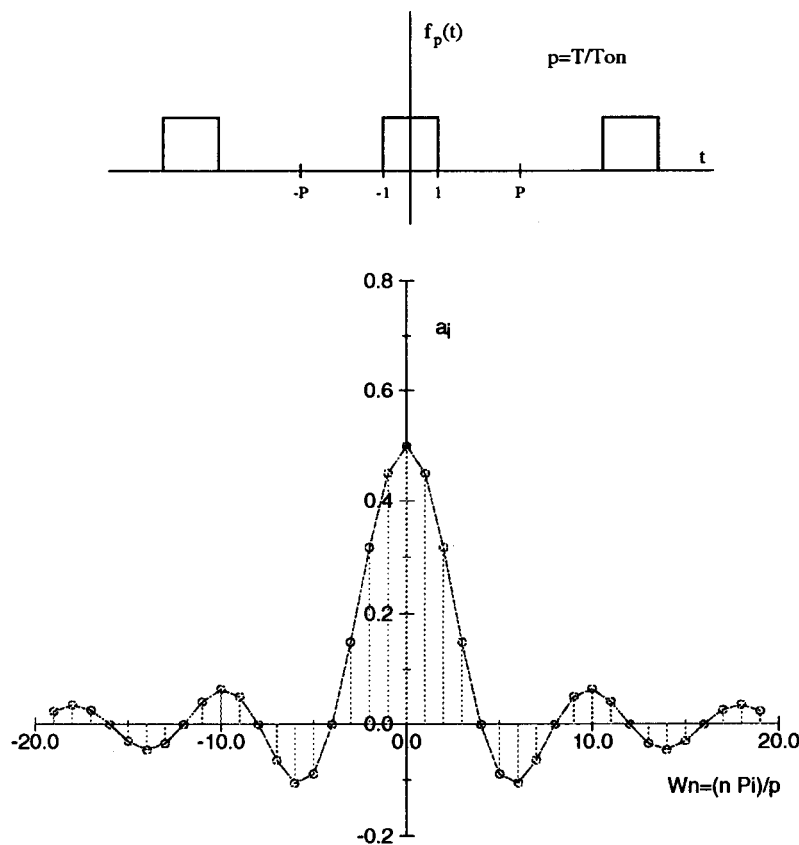


Figura 6.9: Desarrollo en serie de Fourier de una onda PWM.

El espectro de una señal estocástica, que se ha deducido en la sección (5.2),

consta de una parte discreta y una parte continua. La parte discreta del espectro representa la componente de continua de la señal y es la parte que hay que conservar en el filtrado. La parte continua del espectro representa un ruido superpuesto que hay que eliminar que, a diferencia del caso anterior, tiene una potencia que no está concentrada en unas frecuencias determinadas, sino que está distribuido en un continuo de frecuencias. Por este motivo, a diferencia del caso anterior, no es necesario exigir que el filtro tenga una pendiente de caída importante. En este caso lo que es preciso imponer es que la potencia total de ruido integrada por el filtro sea inferior a la potencia de la componente de continua del bit menos significativo.

Esta arquitectura puede utilizarse también para construir convertidores digitales analógicos. Codificar la señal pseudoanalógica por medio de un tren de pulsos estocásticos, en vez de utilizar una señal PWM, tiene una ventaja adicional a la antes indicada. Se pueden realizar operaciones matemáticas (productos y sumas) de una manera muy sencilla con las salidas de los convertidores, no siendo ése el caso si la codificación que se utiliza es la de un PWM.

6.5.2 Estructura completa de los convertidores digital analógico y analógico digital.

En la figura (6.10) se muestra el circuito que efectúa la conversión digital-analógica. El número digital a convertir se compara con una secuencia de números pseudoaleatorios generados por medio de un registro de desplazamiento de realimentación lineal. El bit más significativo se calcula en cada ciclo de reloj evaluando la función OR-exclusiva de un conjunto de bits del registro. Si los bits de realimentación se eligen convenientemente, se consigue que el ciclo de números pseudoaleatorios generados sea $2^n - 1$, donde n es el número de bits del registro. El tren de pulsos que se genera de esta forma se filtra por medio de una red R-C, tal y como se muestra.

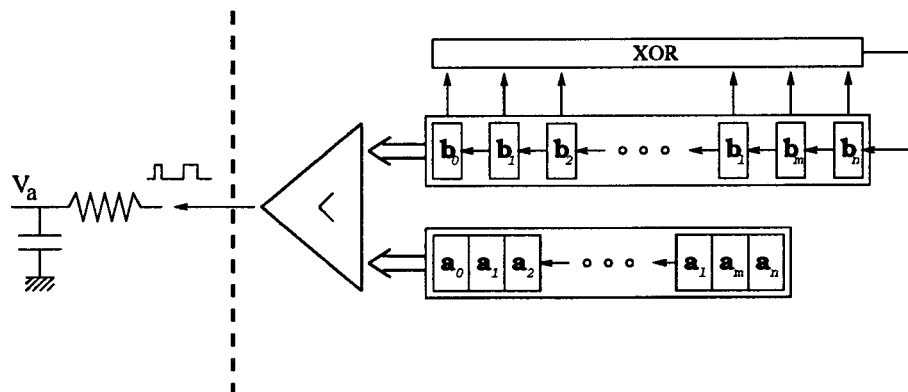


Figura 6.10: Convertidor digital-analógico estocástico.

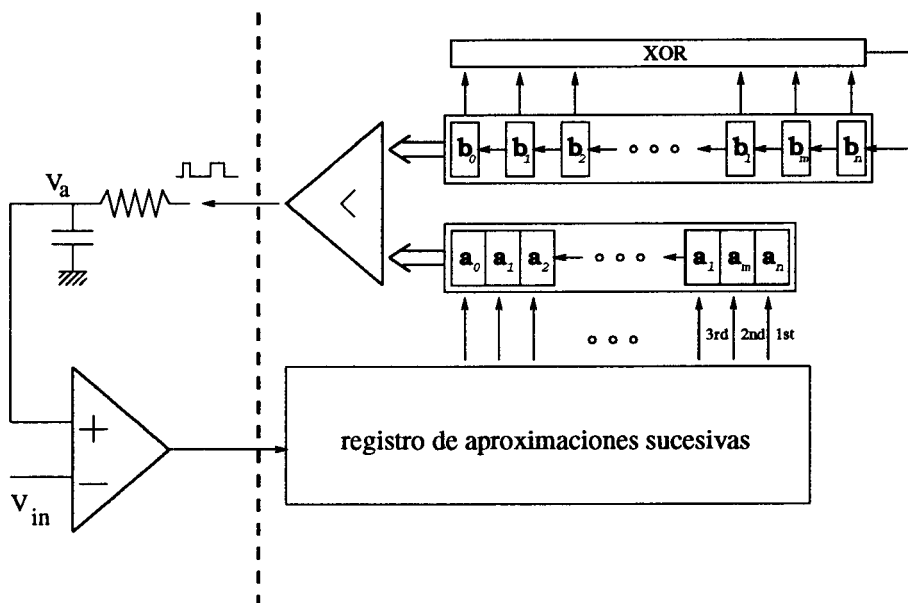


Figura 6.11: Convertidor analógico-digital estocástico.

La conversión analógico-digital se hace empleando un convertidor digital-analógico y un registro de aproximaciones sucesivas, figura (6.11). Debido a la naturaleza aleatoria de los pulsos siempre hay un cierto ruido superpuesto al valor analógico V_a presente a la salida del filtro. Recuérdese que el filtro integra una pequeña parte del espectro continuo de la señal. Este ruido se puede filtrar también digitalmente. Para ello el registro de aproximaciones sucesivas incluye un bloque adicional. Este circuito determina, una vez que la tensión V_a se ha estabilizado, qué salida da, en promedio, el comparador analógico. Si la salida indica que la salida del filtro es inferior a la referencia analógica el registro del convertidor digital analógico es incrementado. Si ocurre lo contrario es decrementado.

6.5.3 Aplicación en un circuito integrado industrial.

Esta estructura se ha utilizado para realizar conversiones analógico-digitales y digital-analógicas en un circuito integrado que se ha fabricado para la empresa SAINCO dentro de un proyecto industrial incluido en el marco GAME. Por motivos económicos era preciso utilizar una tecnología estrictamente digital. Tras efectuarse el estudio teórico antes expuesto, se decidió construir estos convertidores con circuitos basados en lógica estocástica. Las ventajas que éstos presentaban frente a la modulación por anchura de pulso (PWM) eran, principalmente, que los requerimientos de filtrado eran menos estrictos y que las salidas de los convertidores digitales analógicos se podían procesar matemáticamente de una forma muy simple.

El diseño de estos circuitos fue realizado mediante el paquete de programas SYNOPSIS, utilizando el lenguaje VHDL. Para testear el diseño se hizo uso del emulador hardware METASYSTEM. De esta forma el comportamiento real del circuito integrado ha sido probado sin necesidad de ser construido. La figura (6.12) representa el resultado de la conversión analógica-digital cuando se le da un cierto valor

de referencia.

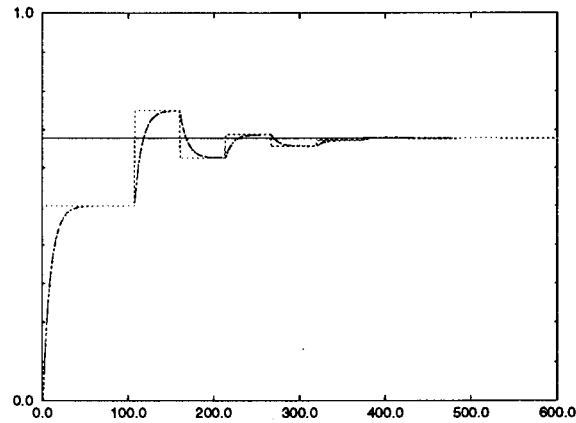


Figura 6.12: Evolución del convertidor analógico-digital por aproximaciones sucesivas.

A continuación se incluyen los listados VHDL de ambos convertidores.

El bloque CDEST.VHD se corresponde con el convertidor digital estocástico, definiéndose las siguientes señales.

- **DATOIN.** Dato digital de 9 bits a convertir.
- **LOAD.** Pulso de carga de DATOIN en registro.
- **PULSE.** Tren de pulsos estocásticos de salida.

La funcionalidad del circuito se define según los siguientes procesos.

- **CARGA.** Carga ó resetea el registro de almacenamiento de DATOIN.
- **LFSR.** Calcula la secuencia digital de números aleatorios a partir de un registro LFSR.

- **COMPARADOR.** Comparación entre los números aleatorios anteriores y el valor digital a convertir.
- **SINCRONISMO.** Sincronización de los registros con el flanco de subida del reloj.

-- VHDL Model Created from SGE Symbol CDEST.sym -- Dec 16 18:35:48 1994

```
library IEEE;
    use IEEE.std_logic_1164.all;
    use IEEE.std_logic_misc.all;
    use IEEE.std_logic_arith.all;
    use IEEE.std_logic_components.all;

entity CDEST is
    Port ( DATOIN : In    STD_LOGIC_VECTOR(9 DOWNTO 0);
          CLK    : In    STD_LOGIC;
          LOAD   : In    STD_LOGIC;
          RESETA : In    STD_LOGIC;
          RESETS : In    STD_LOGIC;
          PULSE  : Out   STD_LOGIC );
end CDEST;

architecture BEHAVIORAL of CDEST is
    signal num:STD_LOGIC_VECTOR(9 DOWNTO 0);
    signal dat:STD_LOGIC_VECTOR(9 DOWNTO 0);
    signal reg_gen:STD_LOGIC_VECTOR(9 DOWNTO 0):="0000000000";
```

```
signal reg_dat:STD_LOGIC_VECTOR(9 DOWNT0 0):="0000000000";
```

```
begin
```

```
--PROCESO DE CARGA Y RESETEO DEL REGISTRO
```

```
CARGA: process(resets,load,reg_dat,busin)
```

```
begin
```

```
if ( resets='1' ) then dat<="0000000000";
```

```
elseif ( load='1' ) then dat<=busin;
```

```
else dat<=reg_dat;
```

```
end if;
```

```
end process CARGA;
```

```
--GENERADOR DE NUMEROS ALEATORIOS (LFSR)
```

```
LFSR: process(resets,reg_gen)
```

```
begin
```

```
if ( resets='1' ) then num<="1111111111";
```

```
else
```

```
num(8 downto 0)<=reg_gen(9 downto 1);
```

```
num(9)<=reg_gen(0) xor reg_gen(3);
```

```
end if;
```

```
end process LFSR;
```



```
--COMPARADOR ENTRE EL VALOR A CONVERTIR Y EL NUMERO ALEATORIO
```

```
COMPARADOR: process(reg_gen,reg_dat)
```

```
begin
```

```
    if ( reg_dat > reg_gen ) then pulse<='1';
```

```
    else pulse<='0';
```

```
    end if;
```

```
end process COMPARADOR;
```

```
--PROCESO DE SINCRONISMO
```

```
SINCRONISMO: process(clk,reseta)
```

```
begin
```

```
    if (reseta='1') then
```

```
        reg_dat<="0000000000";
```

```
        reg_gen<="1111111111";
```

```
    elsif ((clk'event) and (clk='1')) then
```

```
        reg_dat<=dat;
```

```
        reg_gen<=num;
```

```
    end if;
```

```
end process SINCRONISMO;
```

```
end BEHAVIORAL;
```

```
configuration CFG_CDEST_BEHAVIORAL of CDEST is
```

```
for BEHAVIORAL

end for;

end CFG_CDEST_BEHAVIORAL;
```

El bloque LOGICSAR.VHD es el encargado de ejecutar la lógica de aproximaciones sucesivas. Junto con el bloque anterior y la circuitería adicional forma el convertidor analógico digital. Este bloque está preparado para que, según la señal MODO el conjunto funcione como convertidor analógico digital ó digital analógico. Las señales que intervienen en este bloque son las siguientes

- **BITENT.** Tren de pulsos estocásticos de entrada.
- **DATOIN.** Dato digital de 9 bits a convertir, si el modo de conversión es digital analógico.
- **LOAD.** Señal de carga del valor digital a convertir en el caso de tener estar en modo de conversión digital analógica.
- **MODO.** Flag que indica el modo en el que se está usando el conjunto.
- **T1 y T2.** Tiempos característicos de la conversión.
- **DATO_TO_CDEST.** Valor que debe ser cargado en el registro del convertidor CDEST.
- **DATOLISTO.** Señal que indica el final de la conversión analógica digital.
- **LOAD_TO_CDEST.** Pulso de carga del dato correspondiente en registro del convertidor CDEST.

Los procesos más importantes definidos en este bloque son:

- **MUX.** Multiplexor de las señales de carga LOAD y los valores a cargar en el registro del CDEST según el flag MODO.
- **SAR.** Algoritmo de aproximaciones sucesivas.
- **TEMP1 y TEMP2.** Temporizadores correspondientes al proceso de espera y promediado de la conversión.
- **LISTO.** Es el encargado de avisar cuando la conversión ha acabado.

```
-----  
-- VHDL Model Created from SGE Symbol logicsar.sym -- Jan 20 17:16:07 1995  
-----
```

```
library IEEE;  
use IEEE.std_logic_1164.all;  
    use IEEE.std_logic_arith.all ;  
use IEEE.std_logic_unsigned.all ;  
  
entity LOGICSAR is  
    Port ( BITENT : In    STD_LOGIC;  
          CLK    : In    STD_LOGIC;  
          DATOIN : In    STD_LOGIC_VECTOR(9 DOWNT0 0);  
          LOAD   : In    STD_LOGIC;  
          MODO   : In    STD_LOGIC;  
          RESETA : In    STD_LOGIC;  
          RESETS : In    STD_LOGIC;
```

```
        T1 : In    STD_LOGIC_VECTOR(7 DOWNT0 0);
        T2 : In    STD_LOGIC_VECTOR(15 DOWNT0 0);
    DATO_TO_CDEST : Out  STD_LOGIC_VECTOR(9 DOWNT0 0);
    DATOLISTO : Out  STD_LOGIC;
    LOAD_TO_CDEST : Out  STD_LOGIC );
end LOGICSAR;
```

```
architecture BEHAVIORAL of LOGICSAR is
```

```
    signal regsarp,regsard: std_logic_vector(9 downto 0);
    signal aux,indicador: std_logic_vector(10 downto 0);
    signal contp,contd: integer range 0 to 255;
    signal regt2p,regt2d: integer range 0 to 65535;
    signal regt1p,regt1d: integer range 0 to 255;
    signal pulso1,pulso2: std_logic;
    signal despp,despd: std_logic;
```

```
    begin
```

```
--MULTIPLEXOR
```

```
--SI MODO ES 1 AL CDEST LE LLEGAN LOAD Y DATOIN EXTERIOR
```

```
--SI MODO ES 0 EL DATO LE LLEGA DESDE EL LOGICSAR Y LOAD ES 1
```

```
MUX: process(modo,load,datoin,regsard)
```

```
begin
```

```
if(modo='1') then
```

```
load_to_cdest <= load;
```

```
dato_to_cdest <= datoin;
```

```
else
```

```
load_to_cdest <= '1';
```

```
dato_to_cdest <= regsard;
end if;
end process MUX;
```

```
--CUANDO EL TIEMPO DE APROXIMACION SE HA CUMPLIDO
--SE DA LA ORDEN DE DESPLAZAR EL INDICADOR DE BITS
DESPLAZA: process(regt1d, resets)
```

```
begin
if(resets='1') then despp <= '0';
elsif(regt1d=1) then despp <= '1';
else    despp <= '0';
end if;
end process DESPLAZA;
```

```
--MODULO INDICADOR DEL BIT SOBRE EL CUAL ESTA ACTUANDO
--EL ALGORITMO DE APROXIMACIONES SUCEASIVAS
INDICA: process(resets, indicador, regt1d, desp1)
```

```
begin
if(resets='1') then
aux <= "10000000000";
elsif (desp1='1' or indicador="00000000001") then
aux(10) <= '0';
aux(9 downto 0) <= indicador(10 downto 1);
else    aux <= indicador;
end if;
end process INDICA;
```

```
--CUANDO EL INDICADOR LLEGA AL FINAL SE TIENE EL DATO LISTO
```

```
LISTO: process(indicador,modo)
```

```
begin
```

```
if(indicador = "00000000000" and modo = '0') then
```

```
datolisto <= '1';
```

```
else
```

```
datolisto <= '0';
```

```
end if;
```

```
end process LISTO;
```

```
--ALGORITMO DE APROXIMACIONES SUCESIVAS
```

```
SAR: process(resets,regsard,indicador,contd,t1,despd)
```

```
variable prov: integer range 0 to 255 := 0;
```

```
variable indice: integer range 0 to 10:= 0;
```

```
begin
```

```
--SE INICIALIZA EL NUMERO CON EL PRIMER BIT A UNO
```

```
if(resets='1') then
```

```
regsarp <= "10000000000";
```

```
--CUANDO EL TIEMPO DE APROXIMACION SE HA CUMPLIDO
```

```
elsif (despd='1') then
```

```
--SE CODIFICA DE FORMA ENTERA LA POSICION DEL BIT A TRATAR
```

```
for i in 10 downto 1 loop
```

```
if(indicador(i)='1') then indice := i;
```

```
else indice := indice;
```

```
end if;
end loop;
--SIEMPRE QUE NO SE ESTE EN EL ULTIMO BIT
--SE PONE EL SIGUIENTE A UNO PARA CONTINUAR EL ALGORITMO
if(indice>1) then regsarp(indice-2) <= '1';
else regsarp <= regsard;
end if;
--PROCESO DE DECISION SOBRE EL BIT A TRATAR
prov := conv_integer(t1);
if(contd<prov/2) then
regsarp(indice-1) <= '0';
else
regsarp(indice-1) <= '1';
end if;
else
regsarp <= regsard;
end if;
end process SAR;

--TEMPORIZADOR T2 DE TIEMPO DE CONVERSION DE UN BIT
TEMP2: process(resets,t2,regt2d,regt1d,indicador,modo)
variable prov: integer range 0 to 255 := 0;
begin
--SI FUNCIONA COMO CODE O INICIO EN MODO CAD, SE CARGA T2
if(resets='1' or modo='1') then
regt2p <= conv_integer(t2);
--SI LA TEMPORIZACION T1 HA ACABADO Y NO ESTA
```

```
--EN EL ULTIMO BIT, SE CARGA T2/2
elsif(regt1d = 1 and indicador(1)='0') then
prov := conv_integer(t2);
regt2p <= prov/2;
else
--SINO DECREMENTA T2 SIEMPRE QUE SEA MAYOR QUE CERO
if(regt2d>0) then
regt2p <= regt2d - 1;
else
regt2p <= regt2d;
end if;
end if;
end process TEMP2;

--TEMPORIZADOR T1 DE TIEMPO DE PROMEDIADO DE PULSOS
TEMP1: process(resets,regt2d,regt1d,t1,modo)
begin
--SI FUNCIONA COMO CODE O INICIO EN MODO CAD SE RESETEA
if(resets='1' or modo='1') then
regt1p <= 0;
else
--SI LA TEMPORIZACION T2 HA ACABADO SE CARGA T1
if(regt2d = 1) then
regt1p <= conv_integer(t1);
--SINO DECREMENTA T1 SIEMPRE QUE SEA MAYOR QUE CERO
elsif(regt1d >0) then
regt1p <= regt1d - 1;
else
```



```
regt1p <= regt1d;  
end if;  
end if;  
end process TEMP1;
```

```
--SENALES AUXILIARES QUE INDICAN SI SE ESTA TEMPORIZANDO T1 O T2
```

```
PULSOS: process(regt1d,regt2d)  
begin  
if(regt1d > 0) then pulso1 <= '1';  
else    pulso1 <= '0';  
end if;  
if(regt2d > 0) then pulso2 <= '1';  
else    pulso2 <='0';  
end if;  
end process PULSOS;
```

```
--PROCESO DE CUENTA DE PULSOS DE ENTRADA DURANTE T2
```

```
CONTAR: process(resets,pulso1,contd,bitent)  
begin  
if(resets='1') then contp <= 0;  
else  
if(pulso1='0') then contp <= 0;  
elsif(bitent='1') then contp <= contd +1;  
else contp <= contd;  
end if;  
end if;  
end process CONTAR;
```

```
--PROCESO DE SINCRONISMO
SINCRONO: process(reseta,clk,t1,t2)
begin
if (reseta='1') then
indicador <= "10000000000";
regsard <= "10000000000";
regt2d <= 30000;
regt1d <= 0;
contd <= 0;
despd <= '0';
elsif ((clk'event) and (clk='1')) then
indicador <= aux;
regsard <= regsarp;
regt2d <= regt2p;
regt1d <= regt1p;
contd <= contp;
despd <= despp;
end if;
end process SINCRONO;

end BEHAVIORAL;

configuration CFG_LOGICSAR_BEHAVIORAL of LOGICSAR is
    for BEHAVIORAL

        end for;

end CFG_LOGICSAR_BEHAVIORAL;
```

La conexión entre ambos bloques se realiza a través de CAD.VHD .

```
-----  
-- VHDL Model Created from SGE Schematic cad.sch -- Jan 20 17:16:08 1995  
-----
```

```
library IEEE;  
use IEEE.std_logic_1164.all;  
    use IEEE.std_logic_arith.all ;  
use IEEE.std_logic_unsigned.all ;  
  
entity CAD is  
    Port ( BITENT : In    STD_LOGIC;  
          CLK : In    STD_LOGIC;  
          DATOIN : In    STD_LOGIC_VECTOR(9 DOWNT0 0);  
          LOAD : In    STD_LOGIC;  
          MODO : In    STD_LOGIC;  
          RESETA : In    STD_LOGIC;  
          RESETS : In    STD_LOGIC;  
          T1 : In    STD_LOGIC_VECTOR(7 DOWNT0 0);  
          T2 : In    STD_LOGIC_VECTOR(15 DOWNT0 0);  
          BITSAL : Out   STD_LOGIC;  
          DATOLISTO : Out   STD_LOGIC;  
          DATOOUT : Out   STD_LOGIC_VECTOR(9 DOWNT0 0) );  
end CAD;  
  
architecture SCHEMATIC of CAD is
```

```
signal      N_2 : STD_LOGIC;
signal DATOOUT_DUMMY : STD_LOGIC_VECTOR(9 DOWNT0 0);

component CDEST
  Port (  BUSIN : In      STD_LOGIC_VECTOR(9 DOWNT0 0);
         CLK   : In      STD_LOGIC;
         LOAD  : In      STD_LOGIC;
         RESETA : In     STD_LOGIC;
         RESETS : In     STD_LOGIC;
         PULSE : Out     STD_LOGIC );
end component;

component LOGICSAR
  Port (  BITENT : In     STD_LOGIC;
         CLK   : In     STD_LOGIC;
         DATOIN : In     STD_LOGIC_VECTOR(9 DOWNT0 0);
         LOAD  : In     STD_LOGIC;
         MODO  : In     STD_LOGIC;
         RESETA : In     STD_LOGIC;
         RESETS : In     STD_LOGIC;
         T1   : In     STD_LOGIC_VECTOR(7 DOWNT0 0);
         T2   : In     STD_LOGIC_VECTOR(15 DOWNT0 0);
         DATO_TO_CDEST : Out  STD_LOGIC_VECTOR(9 DOWNT0 0);
         DATOLISTO : Out  STD_LOGIC;
         LOAD_TO_CDEST : Out  STD_LOGIC );
end component;

begin
```

```
DATOOOUT <= DATOOOUT_DUMMY;

I_2 : CDEST
  Port Map ( BUSIN=>DATOOOUT_DUMMY, CLK=>CLK, LOAD=>N_2,
            RESETA=>RESETA, RESETS=>RESETS, PULSE=>BITSAL );

I_3 : LOGICSAR
  Port Map ( BITENT=>BITENT, CLK=>CLK, DATOIN=>DATOIN, LOAD=>LOAD,
            MODO=>MODO, RESETA=>RESETA, RESETS=>RESETS, T1=>T1,
            T2=>T2, DATO_TO_CDEST=>DATOOOUT_DUMMY,
            DATOLISTO=>DATOLISTO, LOAD_TO_CDEST=>N_2 );

end SCHEMATIC;

configuration CFG_CAD_SCHEMATIC of CAD is

  for SCHEMATIC
    for I_2: CDEST
      use configuration WORK.CFG_CDEST_BEHAVIORAL;
    end for;
    for I_3: LOGICSAR
      use configuration WORK.CFG_LOGICSAR_BEHAVIORAL;
    end for;
  end for;

end CFG_CAD_SCHEMATIC;
```

Capítulo 7

Procesadores programables estocásticos.

En esta sección se propone la realización de un dispositivo programable que incorpore elementos de cálculo independientes, pero que interconectados entre sí formen estructuras estocásticas como puedan ser redes neuronales, filtros digitales, etc... . Este dispositivo está constituido por módulos funcionales, mostrados en la figura (7.1), entre los que se encuentran:

- Pesos programables. Para ello se difundirán celdas generadoras de trenes de pulsos con la tensión de programación accesible.
- Valores distribuidos de tensión. Servirían para establecer la programación de los pesos anteriores. La unión de estos con las celdas correspondientes se realizaría a través de una matriz de conexiones.
- Funciones productos y sumas. Se utilizarán puertas AND y bloques sumadores.
- Funciones matemáticas. Las operaciones *signo*, *sigmoide*, *lineal*, etc.. son realizadas por bloques lógicos que podrían estar disponibles para su conexión

a los trenes de pulsos resultantes de las diversas sumas.

- Convertidores Analógicos-Estocásticos. Su misión sería dotar de una total autonomía al dispositivo resultante, pudiéndose conectar a cualquier sistema. La conversión estocástica analógica no se incluye ya que es fácilmente realizable con la ayuda de un filtro RC externo.

La inclusión de los circuitos desarrollados en esta tesis hace posible realizar funciones complejas dentro de la lógica estocástica en un espacio reducido y con un grado de fiabilidad ya demostrado. Mediante la estructura expuesta en la figura (7.1) se podrán realizar en paralelo cientos de operaciones matemáticas, y una gran variedad de funciones. Al tener la posibilidad de programar sus conexiones internas, las combinaciones que se pueden realizar con operadores y operandos dan como resultado una enorme variedad de procesadores. Del mismo modo, la conexión con un sistema analógico es posible mediante el uso de convertidores estocásticos.

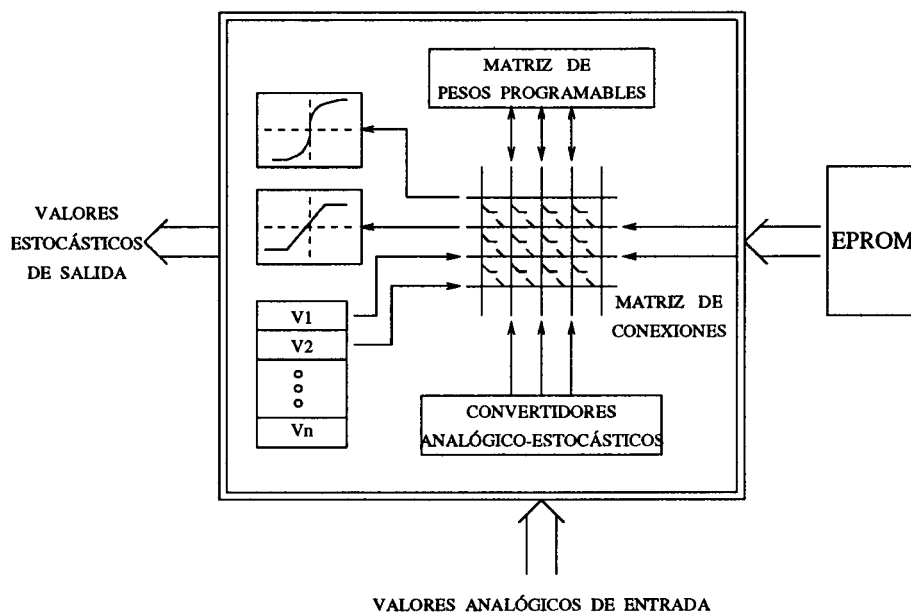


Figura 7.1: Dispositivo programable estocástico.

Capítulo 8

Conclusiones.

Actualmente, los procesadores estocásticos encuentran su principal aplicabilidad en sistemas con elevada carga computacional aunque existen una serie de factores que limitan su eficacia. Con un estudio previo, desarrollado sobre un procesador estocástico concreto, se ha expuesto de forma clara un conjunto de limitaciones que hay que eliminar para hacer del procesamiento estocástico una herramienta electrónica útil. Este processador corresponde a una red neuronal en la cual se ejecutan de forma paralela gran cantidad de productos y sumas.

La primera de las limitaciones en ponerse de manifiesto en el diseño del procesador anterior es la impuesta en la suma de términos. Los métodos encontrados en la bibliografía para realizar la suma estocástica están lejos de resultar eficientes. Algunos de ellos introducen una serialización en las operaciones, eliminando gran parte del atractivo de este tipo de procesamiento: la paralelización de operaciones. En el diseño del procesador neuronal se ha elegido una estructura, descrita ya en la bibliografía, que consigue aprovechar la información contenida en los trenes de pulsos para evaluar de forma totalmente paralela una función concreta; el signo de la suma de términos. Es decir, se decide si el tren de pulsos representativos de

la suma de términos positivos es más denso que el de los términos negativos, ó al contrario. Partiendo de la idea que sustenta esta última estructura de cálculo se ha desarrollado otra capaz de realizar la **suma estocástica totalmente paralela** utilizando un circuito bastante simple. Debido a que las operaciones se realizan de forma paralela, se pueden realizar un número muy elevado de ellas en un tiempo muy pequeño.

Esta forma de resolver la suma se ha empleado para realizar un filtrado digital de señales, en el cual se hace uso de gran cantidad de operaciones productos y sumas. Mediante el uso de un dispositivo de lógica programable (FPGA) se ha realizado físicamente un filtro digital paso alto Butterworth de cuarto orden.

El segundo punto tratado es el relativo al método de generación de trenes de pulsos estocásticos. El circuito digital está compuesto por una gran cantidad de puertas. Como consecuencia aparece una limitación en el número de términos que caben en el circuito integrado. Para resolver este problema se propone un **nuevo circuito generador de trenes de pulsos** basado en el muestreo de una señal periódica. Mediante una tensión de control se pueden obtener distintos trenes de pulsos correspondientes a diferentes valores. La reducción de área ocupada es de dos ordenes de magnitud, de forma que el número de términos que se pueden introducir en una superficie de silicio dada se incrementa en el mismo orden. Estas ideas han llevadas a la práctica en una realización física correspondiente a un circuito integrado en el cual se incluyen un conjunto de celdas generadoras de trenes de pulsos. Estas celdas son estudiadas estadísticamente de forma que se extraen las características principales. La comparación con la técnica tradicional de generación de trenes de pulsos (LFSR) es favorable para el nuevo método hasta una determinada frecuencia de trabajo.

El tercero de los puntos trata el problema de la conexión entre los procesadores estocásticos y los sistemas analógicos mediante **convertidores basados en lógica**

estocástica. En el procesador neuronal que sirve de punto de partida, las entradas son supuestas en forma de trenes de pulsos estocásticos. La aplicación de este procesador a un sistema analógico se realiza mediante el uso del correspondiente convertidor analógico estocástico. Para ello se ha construido un prototipo compuesto por un generador de bytes aleatorios, un CDA y un comparador analógico. Tanto el convertidor analógico estocástico como el procesador neuronal se han usado para resolver un problema de control en el que se implementaba una superficie en un espacio de tres variables. En segundo lugar se aborda el problema de la conversión estocástica analógica. Mediante el análisis estadístico de las señales estocástica se llega a la conclusión de que un simple filtro RC de primer orden es capaz de dar el valor analógico asociado a un determinado tren de pulsos. Igualmente se obtienen los criterios de diseño de dicho filtro.

Enlazando con este tercer punto, se han desarrollado sendos convertidores analógico digital y digital analógico usando tecnología puramente estocástica. La comparación con otras estructuras similares que usan señales PWM resulta satisfactoria. Como conclusión se obtiene que en los casos en los cuales los tiempos característicos de la conversión no son muy exigentes se puede realizar estos procesos en circuitos integrados puramente digitales con la ayuda de una circuitería adicional muy simple. Adicionalmente, se pueden aprovechar las propiedades estocásticas de los trenes de pulsos de salida para realizar de forma muy simple tanto la suma como el producto de valores analógicos. Estos dos circuitos han sido desarrollados para un proyecto industrial financiado dentro del programa europeo GAME, probados mediante el emulador de circuitos integrados METASYSTEM.

Se finaliza la tesis proponiendo una estructura programable que incorpore los avances aquí expuestos. De esta forma se conseguirá un dispositivo en el que se pueda realizar cualquier tipo de procesador estocástico, con gran densidad de cálculo, capaz de conectarse a sistemas analógicos externos y fácilmente configurable.

Bibliografía

- [1] B.R. Gaines. "Stochastic Computing Systems". *Advances in Information Systems*, vol. 2, pp. 37-172. 1969.
- [2] Yoshikazu Kondo y Yasuji Sawada. "Functional Abilities of a Stochastic Neural Network". *IEEE Transactions on Neural Networks*, vol. 3, pp. 375-384 n° 3, Mayo de 1992.
- [3] Carlos Janer Jiménez. "Arquitecturas Estocásticas Paralelas para la Realización Microelectrónica de Redes Neuronales". *Tesis Doctoral*, Mayo de 1994.
- [4] J.M. Quero, C.L. Janer y L.G. Franquelo. "Constrained Hopfield Neural Network for Real-Time Predictive Control". *Proceedings of the 1994 IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'94*. Bologna (Italia), Sept. de 1994.
- [5] W. Peterson. "Error Correcting Codes". *MIT Press*. 1992.
- [6] J.M. Quero, J.G. Ortega, C.L. Janer y L.G. Franquelo. "VLSI Implementation of a fully parallel stochastic Neural Network". *Proceedings of the 1994 IEEE Int. Conf. on Neural Networks, ICNN'94 Orlando (USA)*, Julio de 1994.
- [7] M.J. Bellido, A.J. Acosta, M. Valencia, A. Barriga y J.L. Huertas. "Simple Binary Random Number Generator". *Electronic Letters*, vol. 28, pp. 617-618, n° 7, Marzo de 1992.

- [8] J. Alspector, B. Gupta, R.B. Allen. "Performance of a Stochastic Learning Microchip". *Advances in Neural Inform. Processing Syst.*, pp. 748-760, Denver (USA), Noviembre de 1988.
- [9] G.M. Bernstein y M.A. Lieberman. "Secure Random Number Generation Using Chaotic Circuits". *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 1157-1164, Septiembre de 1990.
- [10] M. van Daalen, P. Jeavons y J. Shawe-Taylor. "A Stochastic Neural Architecture that Exploits Dynamically Reconfigurable FPGAs". *IEEE Proceedings of NAPA conference on FPGAs*, pp. 202-211, 1993.
- [11] Alan F. Murray, Dante Del Corso and Lionel Tarassenko. "Pulse-Stream VLSI Neural Networks Mixing Analog and Digital Techniques". *IEEE Transaction on Neural Networks*, vol. 2, n° 2, pp. 193-204, 1991.
- [12] David E. Van den Bout y Thomas K. Miller III. "A Digital Architecture Employing Stochasticism for the Simulation of Hopfield". *IEEE Transactions on Circuits and Systems*, vol. 36, n° 5, pp. 732-738, Mayo de 1989.
- [13] D. E. Rumelhart and J. L. McClelland. "Parallel Distributed Processing". *MIT Press*. 1986.
- [14] J.M. Quero, J.M. Carrasco, L.G. Franquelo "Adaptive Energy Feed-Back Control for Resonant Converters Using Neural Networks". *Proceedings 2nd Symp. on Large-Scale Digital Calculating Machinery*, pp. 141-146, Harvard University Press, 1951.
- [15] C.L. Janer, J.M. Quero and L.G. Franquelo. "Fully Parallel Summation in a New Stochastic Neural Network Architecture". *IEEE Int. Conf. on Neural Networks*, pp. 1498-1503, San Francisco, 1993.
- [16] William Wike, David E. Van den Bout y Thomas K. Miller III. "The VLSI Implementation of STONN".

- [17] Matthew S. Melton, Tan Phan, Douglas S. Reeves y David E. Van den Bout. "The TInMANN VLSI Chip". *IEEE Transactions on Neural Networks*, vol. 3, n° 3, pp. 434-443, Mayo de 1992.
- [18] M. Togai y H.Watanabe. "Expert System on a chip: an engine for real-time approximate reasoning". *IEEE Expert Syst. Mag.*, pp. 55-62, 1986.
- [19] M. Togai y S.Chiu. "A fuzzy accelerator and a programming environment for real-time fuzzy control". *Proceedings 2nd Symp. IFSA*, pp. 147-151, Tokio (Japón), Julio de 1987.
- [20] T. Yamakawa y T. Miki. "The current mode fuzzy logic integrated circuits fabricated by the standard CMOS process". *IEEE Transactions on Computers*, vol. C-35, n° 2, pp. 161-167. 1986.
- [21] F. Colodro, A. Torralba y L.G. Franquelo. "A Fuzzy-logic chip using stochastic-logic: The defuzzifier". *European Workshop on Industrial Control and Applications*, Abril de 1993.
- [22] A. Torralba, E. Galván y L.G. Franquelo. "A Simple Controller Based on Stochastic Logic. Application to the Control of a DC Motor". *Proceedings of IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, IECON'94* Bologna (Italia), Septiembre de 1994.
- [23] J.M. Retif y B. Allard. "A PWM Asic using Stochastic Coding". *Proceedings of European Power Electronic Conference*, 1992.
- [24] J.G. Ortega, J.M. Quero, C.L. Janer, L.G. Franquelo. "Interfaces to Stochastic Logic: Application to Stochastic Neural Network". *Proceedings of Conference on Electronic, Circuits and Systems, ICECS'94* El Cairo (Egipto), Diciembre de 1994.
- [25] J.G. Ortega, J.M. Quero, C.L. Janer, L.G. Franquelo. "Design Criteria for Fully Parallel Stochastic Neural Network Implementation". *Proceedings of Conference*

- on Electronic, Circuits and Systems*. ICECS'94 El Cairo (Egipto), Diciembre de 1994.
- [26] J.G. Ortega, C.L. Janer, J.M. Quero, L.G. Franquelo. "Analog to Digital and Digital to Analog Conversion Based on Stochastic Logic". *IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation*, IECON'95 Orlando (USA), Noviembre de 1995.
- [27] J.G. Ortega, J.M. Quero, C.L. Janer, L.G. Franquelo "Synaptic Weight Generation in VLSI Stochastic Neural Networks". *IEEE Int. Conf. on Neural Networks*, ICNN'95 Crawford (Australia), Diciembre de 1995.
- [28] Knuth, D.E. "The Art of Computer Programming". *Seminumerical Algorithms*, 2ª Ed., Vol. 2, Addison Wesley Pubs. Company 1981.
- [29] D. H. Lehmer. "Adaptive Energy Feed-Back Control for Resonant Converters Using Neural Networks". . .
- [30] Thomson. *Comp. J.*, vol. 1, pp. 83-86, 1958.
- [31] Rotenberg. *JACM* vol. 7, pp. 75-77, 1960.
- [32] Alan F. Murray. "Silicon Implementations of neural networks". *IEEE Proceedings-F*, vol. 138, nº 1, pp. 3-12, Febrero de 1991.
- [33] Alan F. Murray, A.V.W. Smith. "Asynchronous VLSI neural networks using pulse stream arithmetic". *IEEE J. Solid-State Circuits*, vol. 3, pp. 688-697, 1988.
- [34] W.S. Mackie, H.P. Graf, J.S. Denker. "Microelectronic implementation of connectionist neural network models". *Proceedings of Neural information processing systems Conf.*, (NIPS) ,pp. 515-523, 1987.

- [35] Alan F. Murray, M. Brownlow, A. Hamilton, IL. Songhan, H.M. Reekie. "Pulse-firing neural chips for hundreds of neurons". *Proceedings of Neural information processing systems Conf.*, (NIPS), pp. 785-792, 1990.
- [36] A.P. Thakoor, J.L. Lamb, A. Moonpenn, J. Lambe. "Binary synaptic connections based on memory switching in α -Si:H". *AIP Conf. Proc. 151, Neural networks for computing*, pp. 426-431, 1986.
- [37] M.J. Rose, J. Hajto, P.G. Lecomber. "Amorphous silicon analogue memory devices". *J. Non-Cryst. Solids*, **115**, pp. 168-170, 1989.
- [38] E. Vittoz, H. Oguey, M.A. Maher. "Analog storage of adjustable synaptic weights". *Proceedings of ITG/IEEE Workshop on Microelectronis for neural networks*, pp. 69-79, Dortmund (Alemania), Junio de 1990.
- [39] M. Holler, S. Tam, H. Castro, R. Benson. "An electrically trainable artificial neural network (ETANN) with 10240 floating gate synapses". *Int. Joint Conf. on Neural Networks, IJCNN89*, pp. 191-196, Junio de 1989.