



**CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL**

**DEPARTAMENTO DE INGENIERÍA ELÉCTRICA  
SECCIÓN DE ELECTRÓNICA DEL ESTADO SÓLIDO**

**CIRCUITO INTEGRADO ANALÓGICO CMOS CON  
ARQUITECTURA DE RED NEURONAL CELULAR**

TESIS QUE PRESENTA:

**ING. JESÚS EZEQUIEL MOLINAR SOLIS \***

PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS EN LA  
ESPECIALIDAD DE :

**INGENIERÍA ELÉCTRICA**

ASESORES:

DR. FELIPE GÓMEZ CASTAÑEDA  
DR. JOSÉ ANTONIO MORENO CADENAS

# **Agradecimientos**

**A mis asesores, por su asesoría técnica y orientación en el desarrollo de este trabajo.**

**Al Dr. Oliverio Arellano C. por la revisión y sus consejos en la elaboración del manuscrito.**

**A todos mis profesores y compañeros de la SEES, del laboratorio de VLSI y en general del CINVESTAV.**

**A mis amigos: Alfredo, Jair, Martín... y a todos aquellos que de alguna manera u otra contribuyeron en la elaboración de este trabajo.**

**A la Sra. Graciela y todo el personal de la biblioteca de Ingeniería Eléctrica.**

**Y al COSNET, por el apoyo económico que me brindó durante el programa de maestría.**

# **Dedicatoria**

**A, en honor a su grandeza y por las continuas y tantas formas que tiene de manifestarse.**

**A mis Padres :**

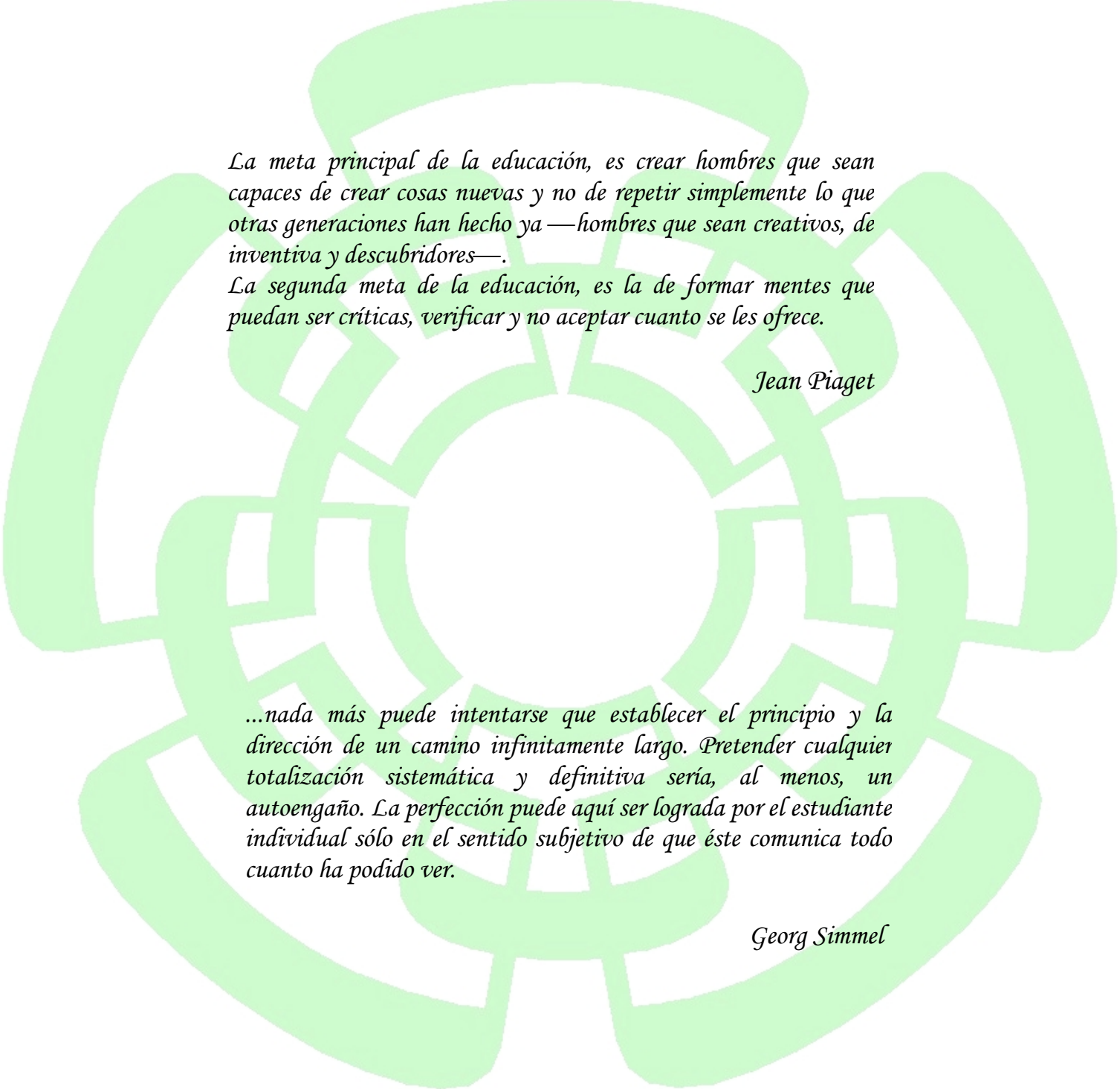
**Porque siempre me supieron guiar por el camino correcto, por entender mis ideas y mis proyectos, a ustedes les debo todo lo que soy.**

**A mis Hermanos:**

**Por estar conmigo en todo momento.**

**Y para mi amigo**

**dondequiera que esté...**



*La meta principal de la educación, es crear hombres que sean capaces de crear cosas nuevas y no de repetir simplemente lo que otras generaciones han hecho ya —hombres que sean creativos, de inventiva y descubridores—.*

*La segunda meta de la educación, es la de formar mentes que puedan ser críticas, verificar y no aceptar cuanto se les ofrece.*

*Jean Piaget*

*...nada más puede intentarse que establecer el principio y la dirección de un camino infinitamente largo. Pretender cualquier totalización sistemática y definitiva sería, al menos, un autoengaño. La perfección puede aquí ser lograda por el estudiante individual sólo en el sentido subjetivo de que éste comunica todo cuanto ha podido ver.*

*Georg Simmel*

# Contenido

Prefacio	i
<b>Capítulo 1</b>	
<b>Teoría básica de las redes neuronales celulares.</b>	
1.0 Introducción	1
1.1 Modelo original de CNN	2
1.2 La conexión local	3
1.3 Las ecuaciones dinámicas	4
1.4 Modelo eléctrico	8
1.5 Variantes del modelo original de CNN	9
1.5.0 CNN con mascarillas no lineales y de tipo retardado	9
1.5.1 CNN con parámetros espacialmente variables	10
1.5.2 CNN con parámetros variables en el tiempo	10
1.5.3 CNN de rango completo	10
1.5.4 CNN de tiempo discreto	11
1.5.5 CNN con función de salida de rango positivo	12
1.5.6 Procesador celular analógico de imágenes	15
1.5.7 Máquina universal CNN	16
1.6 Aplicaciones de la CNN	17
Referencias	22
<b>Capítulo 2</b>	
<b>Diseño de mascarillas de la red neuronal celular.</b>	
2.0 Introducción	23
2.1 Diseño de mascarillas mediante el método simplex	23
2.1.0 Desigualdades para el estado estable	23
2.1.1 Desigualdades para el estado inicial	24
2.1.2 Análisis de la dinámica de la CNN	25
2.1.3 Función objetivo	27
2.1.4 Algoritmo simplex	28
2.1.5 Aplicación del método simplex a sistemas de desigualdades de dos variables	32
2.1.5.0 Ejemplo 1	33
2.1.5.1 Ejemplo 2	34
2.2 Aplicación del método simplex en el diseño de mascarillas de la CNN para diferentes tareas de procesamiento	35
2.2.0 Removedor de ruido	35
2.2.1 Detector de conectividad global	41
2.2.2 Extractor de bordes	44
2.2.3 Extractor de sombras	46
2.2.4 Detector de componentes conectados	48
Referencias	51

## Capítulo 3

### Modelo de CNN con función de salida escalón con desplazamiento.

3.0 Introducción	53
3.1 Red neuronal celular con función de salida escalón	53
3.2 Eliminación del elemento de retroalimentación $a_o$	54
3.3 Desplazamiento de la función escalón al semiplano positivo	56
3.4 Ecuaciones del modelo propuesto de CNN	59
Referencias	61

## Capítulo 4

### Implementación eléctrica de la red neuronal celular.

4.0 Introducción	63
4.1 Diagrama a bloques de una neurona CNN	65
4.2 CNN programable	67
4.3 Entrada, procesamiento y salida de datos de una CNN	67
4.4 Variables físicas utilizadas en la representación de señales en una CNN	68
4.5 Desarrollo electrónico de los bloques que constituyen una neurona CNN programable	68
4.5.0 Bloque sumatoria	69
4.5.1 Implementación de las mascarillas A o B	69
4.5.1.0 Espejos de corriente	71
4.5.1.1 Interruptor MOSFET	73
4.5.1.2 Inyección de carga	75
4.5.1.3 Efecto “ <i>feedthrough</i> ”	75
4.5.1.4 Bloque A o B	76
4.5.2 Bloque multiplicador	78
4.5.3 Bloque comparador	80
4.5.4 Bloques $S_1$ y $S_2$	84
4.5.5 Bloque de entrada	85
4.5.6 Etapa de salida	86
4.6 Arquitecturas de las neuronas propuestas	88
4.7 Neuronas de prueba	90
4.8 Simulaciones eléctricas de una CNN de 4x4 para la ejecución de las diferentes tareas de procesamiento	91
4.8.0 Diagrama esquemático del circuito	91
4.8.1 Removedor de ruido	94
4.8.2 Detector de conectividad global	95
4.8.3 Extractor de bordes	96
4.8.4 Extractor de sombras	97
4.9 Simulación eléctrica de una CNN de 1x8 para la ejecución del detector de componentes conectados	98
4.9.0 Diagrama esquemático del circuito	98
4.9.1 Resultados de la simulación	100
4.10 Desacoplamiento entre transistores	101
4.10.0 Modelo de Pelgrom	101
4.10.1 Simulación de análisis Monte Carlo	102
Referencias	104

## Capítulo 5

### Diseño geométrico de los circuitos.

5.0 Introducción	105
5.1 Celdas básicas	106
5.1.0 Neurona CNN	106
5.1.1 Copiadores de corriente	107
5.1.2 Flip-Flops tipo D	108
5.2 Circuito integrado completo	109
Referencias	111

## Capítulo 6

### Conclusiones.

6.0 Conclusiones	113
6.1 Trabajo futuro	115
Referencias	120

## Apéndices

Apéndice A	121
Apéndice B	123

## Abreviaciones

ABET	Accreditation Board of Engineering and Technology
BSIM	Berkeley Short Channel IGFET
CCD	Connected Component Detection
CCD*	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CNN	Cellular Neural Network
CNN UM	CNN Universal Machine
DSP	Digital Signal Processor
ESD	Electrostatic Discharge
FSR	Full Signal Range
GAPU	Global Analogic Programming Unit
ISR	Improved Signal Range
MOS	Metal Oxide Semiconductor
MOSIS	MOS Implementation Service
NMOS	Transistor MOS canal N
PMOS	Transistor MOS canal P
PSpice	Versión de SPICE para PC
TG	Transmission Gate
UG	Unity Gain
VLSI	Very Large Scale Integration
vMOS	<i>neuron</i> MOSFET



## PREFACE

Analog circuits have played an important role in the development of modern electronic technology. Even in our digital era, analog circuits still dominate some areas like communications, power, control, audio and video due their capabilities for real time processing.

The Artificial Neural Networks inspired on some aspects of neurobiology and defined as parallel networks that employ a large number of processing simple elements called “neurons”, have been a research issue in the past fifty years and many studies have demonstrated their great processing capabilities. While conventional serial computational techniques cannot process high information rates in real time and are reaching intrinsic physical limits of dimensions and speed, the neural networks with their inherent parallel processing are one of the best ways to achieve a superior computational performance.

Although the majority of advancements in neuro-computation results from theoretical analysis or by simulation of parallel networks on serial computers, many of the potential advantages of neural networks are obtainable from effective hardware implementations, that’s why the rapid advancement of VLSI technology in the last years has allowed to implement these systems successfully on many areas like optimization, linear and no linear programming, associative memories, pattern recognition and artificial vision.

From its inception in 1988, the Cellular Neural Network (CNN) structure has demonstrated a great capability for real time image processing. The CNN could be seen as a hybrid structure between cellular automata and neural networks because incorporates the best features of both concepts: On one hand, its dynamics and parallel structure allows real time processing, and for other, its local interconnection feature makes easy their physical realization on VLSI technology. Because all neurons and their interconnections are identical, only 19 parameters defined as the cloning templates makes possible many advantages from the software and hardware point of view. From the software perspective, only 19 numbers must be defined as a neural program to accomplish a specific task, in contrast with the conventional digital computer where a same algorithm must be written with thousands of instructions. This fact gives an idea of the compression capability to reduce the program instructions. And from the hardware perspective, only 19 wires to the outside system are needed to allow the external programming, regardless of the array size. This exclusive feature of the CNN makes possible the IC design with this program capability.

Unlike the serial digital computer, where the computing time increases exponentially with the array size, the parallel processing of a CNN makes possible to the processing time scale linearly (propagating templates) in some cases, while with others is independent of the array size. For this reason, today the CNN is one of the best ways to reach real time image processing.

The most important evolution of the CNN is the CNN Universal Machine (CNN-UM), where other electrical features are added to the neuronal array, such as analogue and digital memory, control units, global and local communications that allow the algorithmic processing of images. The most recent implementations of CNN-UM have reported a huge processing power in the order of  $10^{12}$  OPS (equivalent digital operations per second), a performance which can be matched only by supercomputers and in terms of speed, power and area, CNN-UM is superior to other similar DSP implementation at least in three orders of magnitude.

This thesis work consist of a CNN IC CMOS analog design with an array size of 4x4 neurons, which is employed to different binary image processing tasks such as Noise Remover, Edge Extraction, Global Connectivity Detection, Shadowing and Connected Component Detection. In chapter 1 is exposed the fundamental theory of CNN and some variations to the original model. Chapter 2 shows a design method to obtain the cloning templates, and from some modifications to this method arises the idea of a new model, which is detailed in chapter 3. Chapter 4 explains the electric design of the CNN new model and its simulations to the different image processing tasks. Chapter 5 shows the layout of the basic cells as well for the complete CNN 4x4 array, and finally in chapter 6 the conclusions and the idea of a new circuit for gray scale image processing are exposed.

## PREFACIO

Los circuitos analógicos han jugado siempre un papel importante en el desarrollo de la electrónica moderna. Aun en nuestra era digital, los circuitos analógicos dominan ciertas áreas tales como comunicaciones, potencia, control, audio y video debido a su gran capacidad de procesamiento en cuanto a tiempo real se refiere.

Las *redes neuronales*, inspiradas en aspectos de la neurobiología y definidas generalmente como redes paralelas que emplean un gran número de elementos procesadores simples denominados neuronas, han sido materia de estudio durante los últimos cincuenta años, lo cual ha permitido conocer su gran potencial de procesamiento. Mientras los métodos computacionales digitales convencionales dada su naturaleza serial, no poseen la capacidad de procesar grandes cantidades de información en tiempo real y se aproximan cada vez más a los límites físicos en dimensiones y velocidad, las redes neuronales, por su capacidad de procesamiento paralelo han demostrado ser uno de los mejores caminos para alcanzar desempeños computacionales superiores.

Aun cuando gran parte de los avances de la neuro-computación han resultado del análisis teórico o mediante la simulación de estructuras paralelas en computadoras seriales, la mayoría de las ventajas potenciales de las redes neuronales son observables sobre implementaciones efectivas en hardware, por lo que el rápido avance de la tecnología VLSI a través de estos últimos años ha permitido que éstas sean actualmente implementadas exitosamente en diversos campos como optimización, programación lineal y no lineal, memorias asociativas, reconocimiento de patrones y visión artificial.

Desde su nacimiento en 1988, la estructura denominada Red Neuronal Celular (CNN) ha mostrado ser una estructura básica con enorme potencial en cuanto al procesamiento de imágenes en tiempo real. La estructura CNN puede considerarse como una estructura híbrida derivada de los autómatas celulares y de las redes neuronales, ya que incorpora las ventajas que brindan ambos conceptos: Por un lado, su dinámica y estructura paralela permiten el procesamiento en tiempo real y por otro, las interconexiones locales facilitan su implementación física en tecnología VLSI. Debido a que todas las neuronas y las interconexiones del arreglo son idénticas, sólo 19 parámetros que definen sus mascarillas (*Templates*) permiten muchas ventajas en cuanto a software y hardware se refiere. En cuanto a software, sólo 19 números deben ser declarados como un programa neuronal para llevar a cabo una tarea específica, en donde a comparación con lo que tendría que codificarse en una computadora digital convencional, un programa equivalente llevaría miles de instrucciones. Esto da una idea de la capacidad de compresión desde la perspectiva de instrucciones de programación. Y por otro lado, en cuanto a hardware, solamente un canal de 19 vías es necesario fuera del sistema para la programación externa de todo el arreglo neuronal, independientemente de su tamaño. Esta única característica de las CNN permite el diseño de circuitos integrados con esta capacidad de programación.

Debido a su característica de procesamiento paralelo, a diferencia de la computadora digital convencional donde el tiempo de cómputo en el procesamiento de imágenes crece exponencialmente con el tamaño del arreglo, en una CNN dicho procesamiento en algunos casos crece linealmente (casos con propagación) y en otros es independiente de su tamaño.

Por ello, la CNN representa ser hoy en día una de las mejores alternativas para el procesamiento de imágenes en tiempo real.

La generalización más importante de la CNN ha sido la invención de la máquina universal CNN (*CNN UM*) en donde al arreglo neuronal se le incorpora tanto memoria analógica como digital en forma local y distribuida, unidades de control y comunicaciones tanto locales como globales, que permiten el procesamiento de imágenes en forma algorítmica. Sus implementaciones en los últimos años han mostrado un gran potencial de procesamiento que se encuentra dentro del orden de  $10^{12}$  OPS (operaciones digitales equivalentes por segundo) lo cual sólo puede compararse con supercomputadoras, y en términos de velocidad, potencia y área, la CNN ha demostrado tener una capacidad computacional por encima de cualquier otra implementación equivalente en DSP en tres ordenes de magnitud.

Este trabajo de tesis trata sobre el diseño de un circuito integrado analógico CMOS con una arquitectura CNN de 4x4 neuronas, el cual es utilizado para la realización de diferentes tareas de procesamiento morfológico en imágenes binarias, entre las cuales se mencionan las siguientes: Removedor de ruido, extracción de bordes, detección de conectividad global, extracción de sombras y detector de componentes conectados. En el Capítulo 1 se expone la teoría fundamental de la CNN y algunas variantes de su modelo original. En el Capítulo 2 se analiza un método de diseño para la obtención de los parámetros de la CNN del cual, mediante algunas observaciones se deriva un modelo alternativo de CNN propuesto en este trabajo de tesis, el cual es detallado en el Capítulo 3. En el Capítulo 4 se hace el diseño eléctrico de dicho modelo, y es analizado y probado en base a simulaciones en PSpice para las diferentes tareas de procesamiento. El Capítulo 5 presenta el diseño geométrico total del circuito integrado, así como cada una de las celdas básicas que lo componen. Y, por último, en el Capítulo 6 se exponen las conclusiones, así como la propuesta de un circuito para la ejecución de operaciones morfológicas en imágenes en escala de grises para un trabajo futuro.

# CAPÍTULO 1

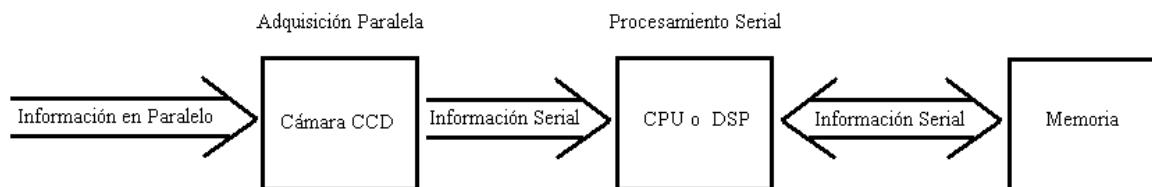


TEORÍA BÁSICA DE LAS REDES  
NEURONALES CELULARES

## 1.0 Introducción

En la actualidad, un gran número de procesadores digitales avanzados pueden resolver muchos problemas, pero en algunos campos sus capacidades están muy limitadas. Un caso específico de ellos es el campo de la visión artificial, en donde incluso seres biológicos simples superan a las supercomputadoras en la ejecución de tareas básicas de visión. Esta diferencia crucial entre lo “biológico” y lo “electrónico” se debe al tipo de procesamiento a que es sometida la información de las imágenes.

La mayoría de los métodos convencionales de visión artificial (figura 1.0) utilizan una cámara con dispositivos CCD\* para la adquisición paralela de la imagen de entrada, éstos descomponen dicha imagen en una versión digitalizada la cual es transferida posteriormente de manera serial a un microprocesador por separado para la ejecución de operaciones espacio-temporales; y finalmente toda la información es acumulada en un dispositivo de almacenamiento. Si consideramos que una cámara de 512x512 píxeles proporciona 20MB/s para una razón típica de 25 cuadros por segundo, esta metodología no es capaz de analizar tales cantidades de información en tiempo real. A pesar de que tienen la capacidad para procesar gran cantidad de información en la realización de autoenfoco, estabilización de la imagen y control de luminiscencia, la ejecución de operaciones espacio-temporales de imágenes en tiempo real, requiere de procesadores mucho más sofisticados que son voluminosos, caros y consumen mucha energía.



**Figura 1.0** Diagrama a bloques de un sistema convencional de visión artificial.

Las retinas biológicas por otro lado, han permitido a los seres vivos ser capaces de analizar en tiempo real escenas complejas cambiantes en el tiempo, dicha característica se debe al inherente procesamiento paralelo que estas últimas estructuras realizan sobre las imágenes.

La retina de los seres humanos es uno de los mejores ejemplos, contiene dos tipos diferentes de células, por un lado las denominadas *células foto-receptoras* como son: los conos y los bastones, que se encargan de convertir la imagen de entrada en señales eléctricas, y las *células procesadoras* cuyos nombres son: horizontal, bipolar, amacrina y ganglionar, que desempeñan operaciones espacio-temporales y reducen la cantidad de información que será transmitida al cerebro (Fig. 1.1).

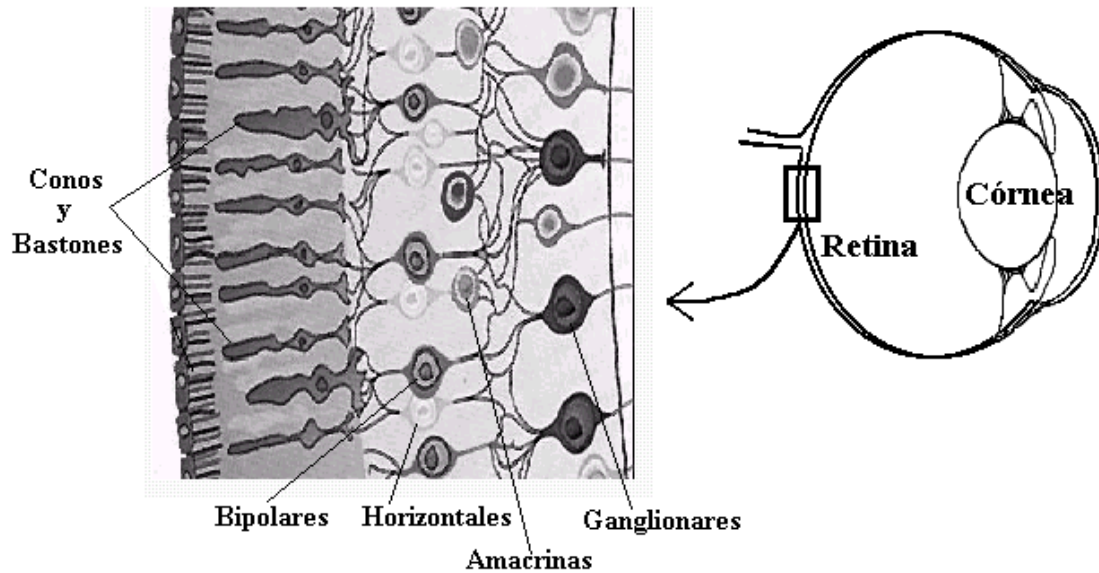


Figura 1.1 Representación parcial del sistema humano de visión.

Así, con la inspiración de los sistemas naturales de visión, universidades y compañías han enfocado sus esfuerzos hacia el desarrollo de nuevas propuestas de sistemas para visión artificial capaces de superar las desventajas de los sistemas convencionales. A través de estos últimos años han existido varias propuestas al respecto tales como: circuitos integrados CNN, autómatas celulares, retinas de silicio, entre otros.

### 1.1 Modelo Original de CNN

El concepto de las Redes Neuronales Celulares (CNN) fue introducido por L. O. Chua y L. Yang [1]. La CNN es una estructura híbrida derivada de los autómatas celulares y de las redes neuronales que reúne las mejores características de ambos conceptos. El término *neuronal* hace referencia a las unidades analógicas elementales que permiten el procesamiento asíncrono paralelo con dinámica en tiempo continuo. Y el término *celular* se refiere al arreglo regular de las neuronas, en donde, cada una se comunica solamente con aquellas definidas dentro de un “vecindario”.

La CNN puede definirse [2] como un arreglo bidimensional de neuronas localmente acopladas, donde cada una de ellas es un sistema dinámico que tiene una entrada, una salida y una variable de estado regida por un grupo de ecuaciones dinámicas. Aunque dicho arreglo puede extenderse a estructuras multicapa [1],[3], a lo largo de este trabajo sólo se considerarán arreglos de tipo bidimensional, es decir, aquellos que son utilizados para el tratamiento de imágenes en donde cada neurona procesa un píxel de la imagen de entrada.

## 1.2 La Conexión Local

La conexión local es una de las características más importantes de la CNN, ya que facilita el enrutamiento entre neuronas y favorece su implementación en tecnología VLSI.

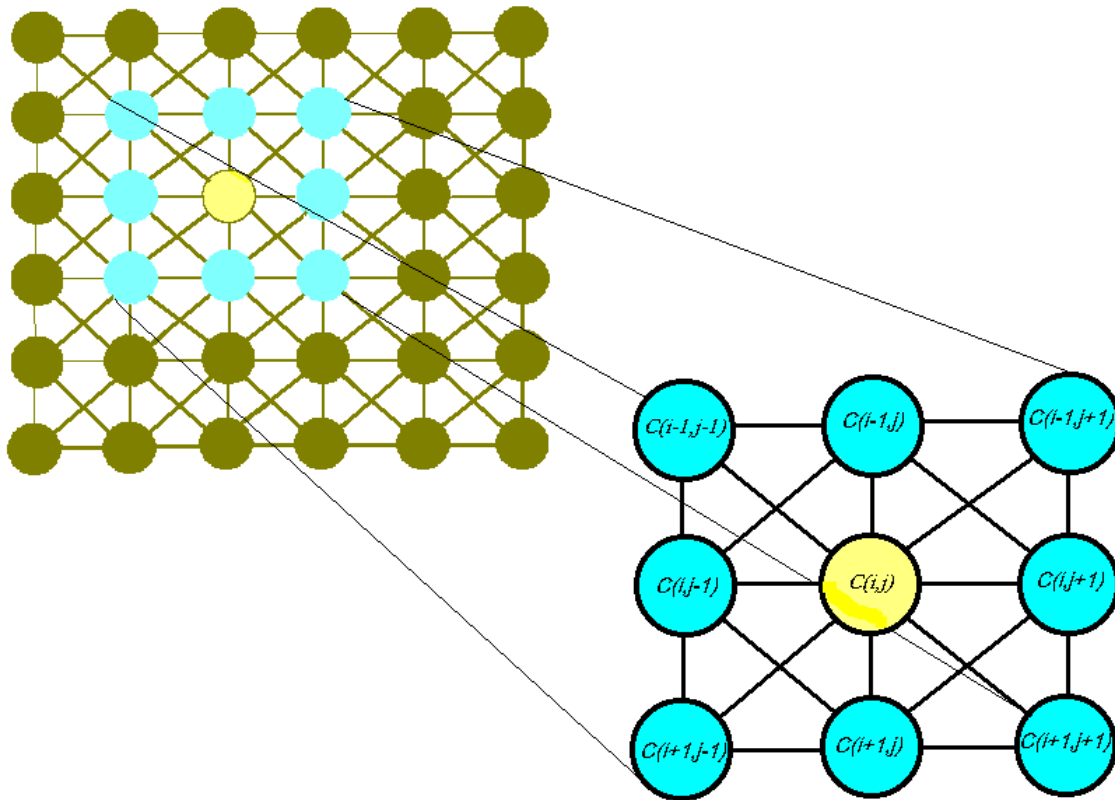
Dentro de un arreglo bidimensional de tamaño  $M \times N$ , cada neurona es identificada por dos variables discretas  $(i,j)$  que hacen la localización de ésta en el espacio discretizado. Cada neurona  $C(i,j)$  interactúa solamente con otras neuronas que han sido definidas dentro de su “vecindario”  $N_r(i,j)$ . Es decir, aquellas otras neuronas que se encuentran dentro de un determinado radio de influencia.

El “vecindario”  $N_r(i,j)$  de una neurona  $C(i,j)$  en una CNN puede definirse por :

$$N_r(i,j) = \{C(k,l) \mid \max\{|k-i|, |l-j|\} < r, 1 \leq k \leq M; 1 \leq l \leq N\}$$

donde  $r$  es siempre un entero positivo.

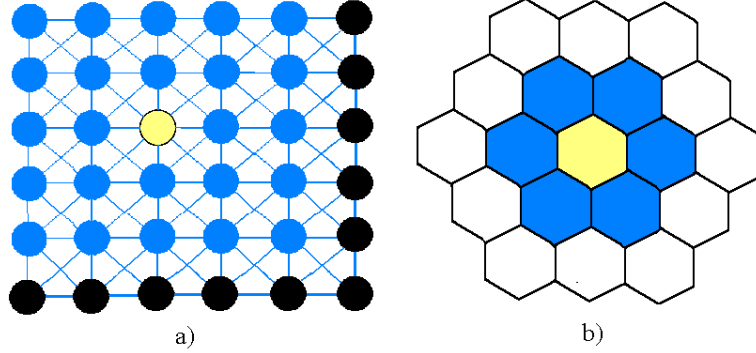
En la práctica existen varias topologías posibles para el arreglo neuronal. En la figura 1.2 se muestra una topología rectangular de orden 1, esto es, para un valor de  $r = 1$  que corresponde a un vecindario de  $3 \times 3$ . Cada círculo representa una neurona y las líneas diagonales, verticales y horizontales indican las conexiones bidireccionales entre éstas.



**Figura 1.2** CNN con topología rectangular de orden 1.



En la figura 1.3 se muestra una topología rectangular de orden 2 ( $r = 2$ ) con un vecindario de 5x5 y un arreglo hexagonal de orden 1. Aun cuando existen otras posibles topologías, las antes mencionadas son las más utilizadas en la realización de circuitos integrados.



**Figura 1.3** a) CNN con topología rectangular de orden 2, b) Topología hexagonal de orden 1.

### 1.3 Las Ecuaciones Dinámicas

Según la definición de L. O. Chua y L. Yang en [1], para una CNN de  $M \times N$  neuronas, cada neurona interna  $C(i,j)$  estará definida por una **ecuación de estado**:

$$C_x \frac{dv_{xij}(t)}{dt} = -\frac{v_{xij}(t)}{R_x} + \sum_{C(k,l) \in N_r(i,j)} A(i,j:k,l)v_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j:k,l)v_{ukl} + I_{ij} \quad (1.0)$$

donde haciendo la capacitancia  $C_x=1$  y la resistencia  $R_x=1$ , se obtiene la ecuación de estado normalizada en la forma:

$$\frac{dv_{xij}(t)}{dt} = -v_{xij}(t) + \sum_{C(k,l) \in N_r(i,j)} A(i,j:k,l)v_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j:k,l)v_{ukl} + I_{ij} \quad (1.1)$$

Donde  $v_{xij}$  representa la *variable de estado* de la neurona  $C(i,j)$ ,  $v_{ukl}$  son las *entradas* de todas las neuronas definidas dentro del vecindario  $N_r(i,j)$ .  $v_{ykl}$  son las *salidas* de las neuronas del vecindario, el término  $I_{ij}$  es definido como el *umbral*. La matriz  $A(i,j:k,l)$  representa la *maska de retroalimentación* y la matriz  $B(i,j:k,l)$  la *maska de control*. Los valores de los elementos en las matrices de ambas maskas definen la interacción entre neuronas y dependiendo del estado inicial de las neuronas  $v_{xij}(0)$  la CNN se puede configurar o programar para que realice diferentes tareas de procesamiento de las imágenes de entrada.

Para una CNN que contempla una topología rectangular de orden 1 y considerando parámetros invariantes, las matrices de retroalimentación, de control y el umbral pueden ser representados de la siguiente manera:

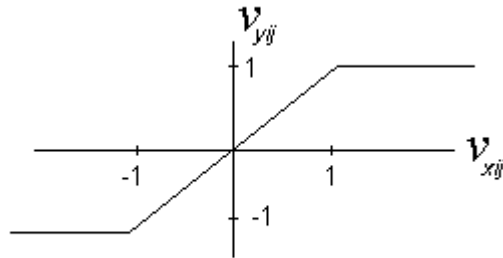
$$A = \begin{bmatrix} A_{i-1,j-1} & A_{i-1,j} & A_{i-1,j+1} \\ A_{i,j-1} & A_{i,j} & A_{i,j+1} \\ A_{i+1,j-1} & A_{i+1,j} & A_{i+1,j+1} \end{bmatrix}, \quad B = \begin{bmatrix} B_{i-1,j-1} & B_{i-1,j} & B_{i-1,j+1} \\ B_{i,j-1} & B_{i,j} & B_{i,j+1} \\ B_{i+1,j-1} & B_{i+1,j} & B_{i+1,j+1} \end{bmatrix}, \quad I_{ij}$$

Donde, como se muestra en la ecuación (1.1), cada elemento de la matriz  $A$  con subíndice  $(k,l)$  es multiplicado por la salida de la neurona  $C(k,l)$  correspondiente. De igual manera los elementos de la matriz  $B$  se multiplican por las entradas de las neuronas vecinas.

También, se establece la **ecuación de salida**:

$$v_{yij} = f(v_{xij}) = \frac{1}{2} (|v_{xij}(t) + 1| + |v_{xij}(t) - 1|) \quad (1.2)$$

Esta ecuación de salida está representada por el elemento no lineal característico del modelo. En la figura 1.4 se ilustra la gráfica de dicha función, en donde siguiendo la convención, un valor de salida  $v_{yij} = -1$  es el correspondiente a un píxel blanco y  $v_{yij} = 1$  a un píxel negro. En lo posterior haremos referencia a esta función como “función de salida UG”.



**Figura 1.4** Función de salida UG.

También se define una *ecuación de entrada*:

$$v_{uij} = E_{ij}$$

La entrada simplemente está representada por  $E_{ij}$  que es la señal proveniente de algún fotodetector o dispositivo de almacenamiento. Esta señal en lo posterior será considerada como invariante en el tiempo y siempre estará normalizada en el rango  $-1 \leq v_{uij} \leq 1$ .

Para todo sistema se tiene una *condición inicial*:

$$v_{ij}(0)$$

En una CNN las condiciones iniciales son muy importantes, ya que como se observa en la ecuación de estado (1.1), la evolución en el tiempo de la variable de estado depende de dicha condición.

### *Neuronas regulares y de frontera*

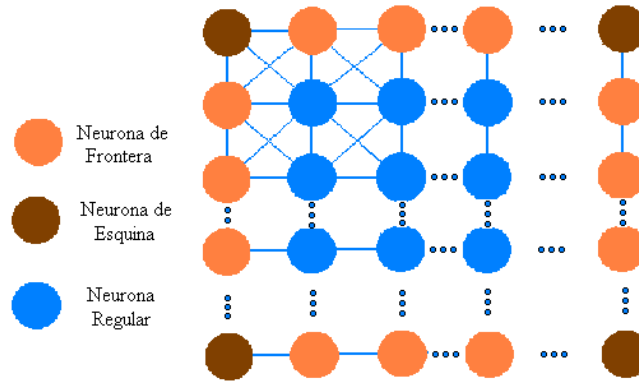
Dentro de una CNN existirán “neuronas regulares” y “neuronas de frontera” tanto en los bordes como en las esquinas del arreglo bidimensional (figura 1.5). Estas neuronas deben satisfacer las condiciones de frontera para el correcto desempeño de la red. Existen tres condiciones de frontera más comúnmente utilizadas :

1).- *Condición de frontera fija (Dirichlet)*. En este tipo de frontera, los valores de la variable de estado  $v_{xkl}$  y la salida  $v_{ykl}$  de cada neurona  $C(k,l)$  que se encuentra en la frontera, están asignados a un valor constante.

2).- *Condición de frontera de Neumann*. Aquí, las variables de estado  $v_{xij}$  de las neuronas perpendiculares a las fronteras están restringidas a ser iguales las de unas con otras.

3).- *Condición de frontera periódica (Toroidal)*. En este caso, las neuronas de la frontera de un extremo (horizontal o vertical) son conectadas con aquellas de su extremo opuesto.

El trabajo [4] es un ejemplo de una CNN con capacidad de programar las condiciones de frontera tanto en la forma (1) como en la forma (3). En este trabajo, como se verá mas adelante, se utiliza sólo la condición de frontera de tipo (1).



**Figura 1.5** Neuronas regulares, neuronas de frontera y neuronas de esquina.

### *Convergencia y estabilidad*

En el trabajo original publicado por L. O. Chua y L. Yang [1], se hace la demostración de la convergencia y la estabilidad para el modelo siempre y cuando se verifique que:

1.- La matriz de retroalimentación  $A$  tenga la propiedad de simetría. Esto es:

$$A(i,j;k,l) = A(k,l;i,j), \text{ y}$$

2.- Para garantizar una salida binaria en estado estable, es necesario que se cumpla:

$$A(i,j;i,j) > 1 \quad (1.3)$$

En las implementaciones físicas de las CNNs dicha condición de simetría no se satisface siempre, ya que la variación de parámetros físicos atribuibles a la tecnología y las tolerancias de los elementos del circuito, así como cambios deliberados para la realización de tareas particulares, hacen que tal condición no se cumpla en la mayoría de los casos. En el trabajo hecho en [5], se hace un análisis de tales casos, así como la demostración de convergencia y estabilidad para mascarillas no simétricas.

### *Rango dinámico*

Para llevar a cabo la implementación física de una CNN de manera exitosa, es necesario hacer un estudio de la dinámica de las variables involucradas a fin de determinar las magnitudes de los voltajes y las corrientes del diseño analógico que se pretende realizar. A partir de la ecuación de salida (1.2) se tiene que la salida de una neurona estará acotada por:

$$v_{yij} \in [-1,1]$$

Además, la variable de estado según lo establecido en [1], estará limitada por:

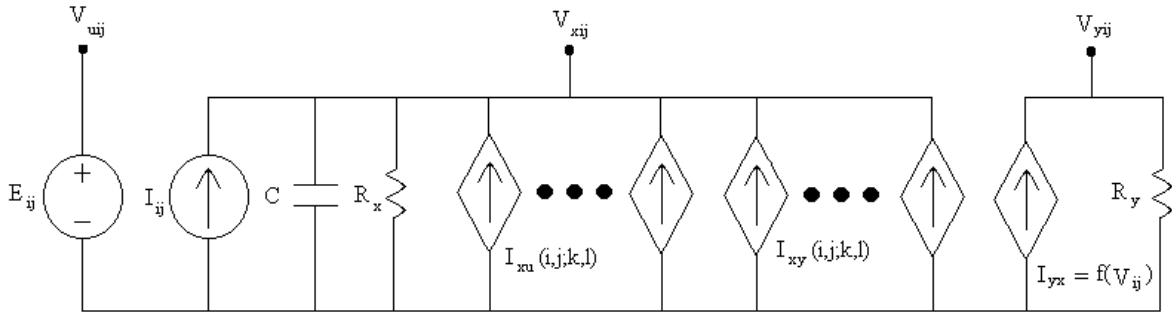
$$v_{\max} = 1 + |I| + \text{Max}_{1 \leq i \leq M, 1 \leq j \leq N} \left[ \sum_{C(k,l) \in N_r(i,j)} (|A(i,j:k,l)| + |B(i,j:k,l)|) \right]$$

siempre y cuando se satisfaga:

$$\begin{aligned} v_{xij}(0) &\leq 1, & 1 \leq i \leq M; 1 \leq j \leq N \\ v_{uij} &\leq 1, & 1 \leq i \leq M; 1 \leq j \leq N \end{aligned}$$

### 1.4 Modelo Eléctrico

El modelo eléctrico original de la CNN propuesto por Chua y Yang [1] sigue la forma de la ecuación (1.0), y está en principio constituido por elementos eléctricos lineales y no lineales, como se muestra en la figura 1.6.



**Figura 1.6** Diagrama eléctrico de una neurona basada en el modelo Chua-Yang.

Cada neurona en el sistema contiene una fuente de voltaje  $E_{ij}$  como entrada, una fuente de corriente independiente  $I_{ij}$  como el umbral, una fuente de corriente dependiente  $I_{yx}$  que corresponde a la función no lineal (1.2) que depende de la variable de estado  $v_{xij}$  y que a través de la resistencia  $R_y$  determinan el voltaje de salida  $v_{yij}$ . Las fuentes lineales de corriente controladas por voltaje  $I_{xu}(i,j,k,l)$  e  $I_{xy}(i,j,k,l)$  son fuentes acopladas con las neuronas vecinas mediante el voltaje de entrada  $v_{uij}$  y el de salida  $v_{yij}$  de estas últimas, dichas fuentes constituyen las matrices de retroalimentación y de control respectivamente. Y el capacitor de estado  $C$  y la resistencia  $R_x$  constituyen el “núcleo” de la neurona.

## 1.5 Variantes del Modelo Original de CNN

Desde su definición en 1988 a la fecha, el modelo propuesto por L. O. Chua y L. Yang ha sufrido modificaciones que permiten tanto extenderlo para la resolución de problemas mas complejos a nivel de simulación en el tratamiento de imágenes, así como simplificarlo para facilitar su implementación en circuitos integrados.

### 1.5.0 CNN con Mascarillas No Lineales y de Tipo Retardado

La definición antes mencionada de L. O. Chua y L. Yang en su modelo eléctrico, contempla la utilización de fuentes de corriente lineales dependientes de voltaje para la implementación de las matrices de retroalimentación y de control, dicha característica hace definir a las mascarillas con valores fijos de sus elementos como *mascarillas lineales* por su dependencia lineal con los voltajes  $v_{uij}$  y  $v_{yij}$  de sus vecinos.

Debido a que este tipo de parámetros lineales no son capaces de resolver ciertas tareas de procesamiento de imágenes que involucran propiedades no lineales, las CNNs no lineales y con parámetros retardados fueron definidas en [6] de la forma:

$$\hat{A}_{ij;kl}(v_{ykl}, v_{yij}) + A_{ij;kl}^{\tau} v_{ykl}(t - \tau) \quad \text{sustituyendo} \quad A(i, j : k, l) v_{ykl}, y$$

$$\hat{B}_{ij;kl}(v_{ukl}, v_{uij}) + B_{ij;kl}^{\tau} v_{ukl}(t - \tau) \quad \text{sustituyendo} \quad B(i, j : k, l) v_{ukl}$$

en el modelo original, de manera que se obtiene una nueva ecuación de estado:

$$\frac{dv_{xij}(t)}{dt} = -v_{xij}(t) + \sum_{C(k,l) \in N_r(i,j)} \hat{A}_{ij;kl}(v_{ykl}(t), v_{yij}(t)) + \sum_{C(k,l) \in N_r(i,j)} \hat{B}_{ij;kl}(v_{ukl}(t), v_{uij}(t))$$

$$+ \sum_{C(k,l) \in N_r(i,j)} A_{ij;kl}^{\tau} v_{ykl}(t - \tau) + \sum_{C(k,l) \in N_r(i,j)} B_{ij;kl}^{\tau} v_{ukl}(t - \tau) + I_{ij}$$

Donde:  $\hat{A}_{ij;kl}$  y  $\hat{B}_{ij;kl}$  son las matrices con parámetros no lineales,  $A_{ij;kl}^{\tau}$  y  $B_{ij;kl}^{\tau}$  son las matrices con parámetros retardados.

La CNN con parámetros retardados ha tenido aplicación en la detección y estimación de ciertos tipos de movimiento [6] y su estabilidad es estudiada en [7],[8]. Ejemplos de tareas de procesamiento de imágenes que contemplan mascarillas no lineales pueden ser encontradas en [9] y [10].

### 1.5.1 CNN con Parámetros Espacialmente Variables.

En los casos mas generales, los parámetros  $A(i,j;kl)$ ,  $B(i,j;k,l)$  e  $I_{ij}$  se consideran invariantes en el espacio y en el tiempo. De este modo, cuando en una CNN sus parámetros antes mencionados poseen la característica de cambiar entre neuronas, entonces se habla de una CNN con parámetros espacialmente variables.

### 1.5.2 CNN con Parámetros Variables en el Tiempo.

En la CNN con parámetros variables temporalmente, se permite que los parámetros de las mascarillas cambien durante la evolución de la red. De [11], podemos obtener la definición de una CNN con este tipo de parámetros de la siguiente manera:

*Ecuación de estado:*

$$\frac{dv_{xij}(t)}{dt} = -v_{xij}(t) + \sum_{C(k,l) \in N_r(i,j)} A_{ij;kl}(t)v_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B_{ij;kl}(t)v_{ukl} + I_{ij}(t)$$

*Función de salida:*

$$v_{yij} = f(v_{xij}) = \frac{1}{2}(|v_{xij}(t) + 1| + |v_{xij}(t) - 1|)$$

En donde, a diferencia de (1.1), los términos  $A_{ij;kl}$ ,  $B_{ij;kl}$  e  $I_{ij}$  son funciones del tiempo. En [10], se demuestra cómo una estructura CNN de una sola capa con este tipo de parámetros puede realizar cualquier función lógica (de Boole) y, en [3], se propone una arquitectura multicapa que puede llevar a cabo tareas como “adelgazamiento”(skeletonization) y filtrado tipo “half-toning”.

### 1.5.3 CNN de Rango Completo

En el modelo de rango completo o FSR [12], la ecuación de estado para cada neurona es modificada en la forma:

$$\tau \frac{dv_{xij}(t)}{dt} = -g[v_{xij}(t)] + \sum_{C(k,l) \in N_r(i,j)} A(i,j:k,l)v_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i,j:k,l)v_{ukl} + I_{ij}$$

Donde  $g[v_{xij}(t)]$  es una función no lineal de la forma:

$$g[v_{xij}(t)] = \begin{cases} m(v_{xij} - 1) + 1 & v_{xij} > 1 \\ v_{xij} & |v_{xij}| \leq 1 \\ m(v_{xij} + 1) - 1 & v_{xij} < -1 \end{cases}$$

para valores de  $m$  grandes.

Este modelo permite, mediante  $g[v_{xij}(t)]$ , acotar la variable de estado en el mismo rango de variación que las entradas y salidas  $[-1,1]$ , lo cual, a diferencia del modelo de Chua y Yang donde la variable de estado alcanza valores mucho mayores a  $+1$  y menores a  $-1$ , este modelo permite reducir el rango dinámico del sistema simplificando la complejidad eléctrica, el área y el consumo de energía. Una generalización de este modelo es el llamado modelo de rango mejorado o ISR en el que  $m$  puede tomar cualquier valor. La convergencia y estabilidad del modelo FSR puede verificarse en [13].

#### 1.5.4 CNN de Tiempo Discreto

En la implementación integrada de una CNN, una de las desventajas del modelo eléctrico original propuesto por Chua y Yang es la necesidad de integrar capacitores y resistores. Según [14], esto representa un gran consumo de área ya que constantes de tiempo grandes deben ser consideradas en el diseño con el propósito de reducir la influencia de capacitancias y resistencias parásitas. A pesar de que dichos componentes pasivos son disponibles en tecnología CMOS estándar, éstos por lo regular consumen mucha área y son imprecisos para los requerimientos del modelo [15]. De esta manera surgieron las propuestas de CNNs en tiempo discreto para superar las desventajas antes mencionadas con implementaciones que sólo consideran el uso de transistores [16].

La ecuación de estado en tiempo discreto para una neurona  $C(i,j)$  se obtiene al aplicar el algoritmo de Euler en la discretización. La expresión (1.1) puede ser transformada utilizando pasos de tiempo  $t = kT$  (con  $n = 0,1,2,\dots$ ) quedando de la forma:

$$v_{xij}(k+1) = v_{xij}(k)(1-h) + h \left[ \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)v_{yij}(k) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{ukl} + I_{ij} \right]$$

Donde  $h$  es el tamaño del paso y determina la precisión con la que se aproxima la derivada, así como el número de iteraciones que debe realizar la red hasta alcanzar la condición de estado estable.



Este tipo de CNN ha encontrado gran aplicación en tareas donde las entradas son binarias y aunque muchos de los parámetros utilizados en una CNN de tiempo continuo producen los mismos estados estables que en una CNN de tiempo discreto, existen arquitecturas en tiempo discreto cuyas tareas no pueden ser resueltas en una CNN convencional de tiempo continuo.

### 1.5.5 CNN con Función de Salida de Rango Positivo.

En estructuras en donde sólo se contempla el procesamiento de imágenes binarias, la teoría original de CNN de Chua y Yang puede ser modificada en algunos aspectos para facilitar la implementación de arreglos de CNN con circuitos menos complejos que permitan la integración de un mayor número de neuronas.

Se ha demostrado que la función de salida de ganancia unitaria (función UG) propuesta en [1], puede ser modificada por una función de rango positivo [17]. Los parámetros definidos para redes con función de salida UG pueden ser expandidos a una función de rango positivo mediante un proceso de mapeo o transformación lineal. Siguiendo lo establecido en [17] tenemos:

Si se define  $g_{ij}$  como:

$$g_{ij} = \sum_{C(k,l) \in N_r(i,j)/C_{ij}} A(i,j;k,l)y_{kl} + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)u_{kl} + I \quad (1.4)$$

donde  $g_{ij}$  es independiente de la variable de estado de la neurona y su salida, por lo tanto puede ser interpretada como una variable independiente. La notación  $C(k,l) \in N_r(i,j)/C_{ij}$  hace referencia a todas las neuronas del vecindario de  $C(i,j)$  a excepción de la neurona misma  $C(i,j)$ . De tal modo que la ecuación de estado (1.1) queda como:

$$\frac{dv_{xij}}{dt} = -v_{xij} + A_0 v_{yij} + g_{ij}$$

donde  $A_0$  denota el elemento central de la mascarilla de retroalimentación. De [18] corolario 2, pueden obtenerse las condiciones para el funcionamiento de una CNN para casos donde la entrada, el estado inicial y la salida de la red son bipolares. Dichas condiciones son:

- 1.-  $(A_0 - 1) * (v_{yij} = 1) + g_{ij}^- > 0$
- 2.-  $(A_0 - 1) * (v_{yij} = 1) + g_{ij}^+ < 0$
- 3.-  $(A_0 - 1) * (v_{yij} = -1) + g_{ij}^- > 0$
- 4.-  $(A_0 - 1) * (v_{yij} = -1) + g_{ij}^+ < 0$

Donde los casos 1 y 4 son para salidas negro y blanco en estado estable, respectivamente. Los casos 2 y 3 son condiciones requeridas para que la salida entre a la

región lineal de la función UG.  $g_{ij}^-$  y  $g_{ij}^+$  son los límites inferior y superior de  $g_{ij}$  en algún instante.

Si la neurona se considera que evoluciona *monótonamente* en el tiempo, esto es que cuando el valor de la salida entra a la región lineal de la función UG, cruzará toda esta región hasta alcanzar eventualmente la región de saturación de signo opuesto, tal como se demuestra en [19] excepto para el caso del proceso CCD, los cuatro casos que definen el comportamiento de una CNN con una función de salida UG pueden quedar de la forma:

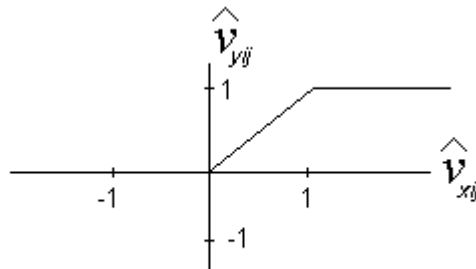
$$\begin{aligned} \text{BB: } & A_o + g_{ij}^- > 1 \\ \text{BW: } & A_o + g_{ij}^+ < 1 \\ \text{WB: } & -A_o + g_{ij}^- > -1 \\ \text{WW: } & -A_o + g_{ij}^+ < -1 \end{aligned}$$

donde la desigualdad BB establece la condición para que en una neurona con entrada en negro su salida permanezca en negro, WW para una entrada y salida en blanco. Y tanto BW como WB para que la neurona haga la transición de negro a blanco y blanco a negro, respectivamente.

Así, con las consideraciones antes mencionadas, puede proponerse el cambio de la función de salida de la CNN a un tipo de salida de rango positivo, donde dicha función se define como:

$$\hat{v}_{yij} = f(\hat{v}_{xij}) = \begin{cases} 0, & \hat{v}_{xij} < 0 \\ \hat{v}_{xij}, & 0 \leq \hat{v}_{xij} \leq 1 \\ 1, & \hat{v}_{xij} > 1 \end{cases}$$

esta función se ilustra en la figura 1.7



**Figura 1.7** Función de salida de rango positivo.

El cambio hacia esta función puede ser llevado a cabo mediante una *transformación lineal* de la entrada de la neurona y la variable de estado, haciendo:

$$\hat{v}_{xij} = \frac{1}{2}(v_{xij} + 1), \hat{v}_{uij} = \frac{1}{2}(v_{uij} + 1)$$

donde  $v_{xij}$  y  $v_{yij}$  son la variable de estado y la salida de (1.1) y (1.2), respectivamente;  $\hat{v}_{xij}$  y  $\hat{v}_{yij}$  son las variables transformadas. De tal modo que la nueva ecuación de estado es la misma que la de la ecuación (1.1) con excepción de las variables transformadas, quedando:

$$\frac{d\hat{v}_{xij}(t)}{dt} = -\hat{v}_{xij}(t) + \sum_{\substack{C(k,l) \in N_r(i,j) \\ v_{ykl}=1}} A(i, j : k, l) \hat{v}_{ykl}(t) + \sum_{C(k,l) \in N_r(i,j)} B(i, j : k, l) \hat{v}_{ukl} + \hat{I}_{ij}$$

en donde solamente el término  $\hat{I}_{ij}$  tiene que ser modificado. Para obtener  $\hat{I}_{ij}$  puede hacerse la siguiente consideración:

Si dividimos  $g_{ij}$  de la ecuación (1.4) en tres partes:

$$e_1 = \sum_{\substack{C(k,l) \in N_r(i,j) / C_{ij} \\ v_{ykl}=1}} A(i, j : k, l) + \sum_{\substack{C(k,l) \in N_r(i,j) \\ v_{ukl}=1}} B(i, j : k, l)$$

$$e_2 = \sum_{\substack{C(k,l) \in N_r(i,j) / C_{ij} \\ v_{ykl}=-1}} A(i, j : k, l) + \sum_{\substack{C(k,l) \in N_r(i,j) \\ v_{ukl}=-1}} B(i, j : k, l)$$

y el término  $I$ , y definimos tanto  $f_1$  y  $f_2$  de igual manera que  $e_1$  y  $e_2$  y además consideramos dos ecuaciones de estado, una en cada región de saturación de la función de salida UG, tendremos:

$$A_0 + e_1 - e_2 + I = 1, \quad -A_0 + f_1 - f_2 + I = -1 \quad (1.5)$$

Y para la función de salida de rango positivo serán.

$$A_0 + e_1 + \hat{I} = 1, \quad f_1 + \hat{I} = 0 \quad (1.6)$$

Como la salida de la función UG  $v_{yij} = -1$ , corresponde a un  $v_{yij} = 0$  en la función de rango positivo, entonces tanto  $e_2$  como  $f_2$  y  $-A_0$  se hacen cero. De esta manera  $\hat{I}$  puede ser determinada, de tal manera que las desigualdades en (1.6) se mantengan para cualquier combinación entrada-salida que satisfaga a las igualdades (1.5). Entonces  $\hat{I}$  puede ser escrita como:

$$\hat{I} = \frac{1}{2} \left[ I + 1 - \sum_{C(i,j) \in N_r(ij)} (A(i,j;k,l) + B(i,j;k,l)) \right]$$

De esta manera, las mascarillas utilizadas en modelos de CNN con función de salida UG pueden también ser utilizados para el caso de función de rango positivo a excepción del término de umbral  $\hat{I}$  que debe ser alterado de acuerdo a la ecuación anterior.

El beneficio directo de esta transformación lineal radica en que en el procesamiento de la neurona aumenta el número de operaciones que involucran el factor cero correspondiente a los píxeles blancos tanto de entrada como de salida. Si los circuitos empleados en el diseño analógico permite realizar tales operaciones con un muy alto grado de precisión, puede incrementarse la robustez de los parámetros de las mascarillas en un gran porcentaje como puede verificarse en [17]. Esto, desde el punto de vista de la implementación física, trae muchos beneficios que se reflejan principalmente en el área de silicio requerida [20]. Por una parte, si consideramos que la precisión de un multiplicador analógico es inversamente proporcional a las dimensiones de sus transistores, al tener los coeficientes de multiplicación un mayor porcentaje de tolerancia a variaciones, trae consigo que el diseño de los bloques para la multiplicación consideren transistores mucho más pequeños en comparación con otras implementaciones que siguen el modelo original. Por otro lado, la función de salida de rango positivo puede ser realizada con estructuras más compactas que aquellas utilizadas para la función de salida UG.

En particular una de la mayores aportaciones de Ari Paasio *et al.* [20], radica en que la función de rango positivo permite reducir los multiplicadores de cuatro cuadrantes del modelo original al empleo de multiplicadores de sólo dos cuadrantes. Además, hace notar que para el tratamiento de imágenes binarias, la operación de multiplicación puede ser reducida a un caso de selección como puede observarse también en [20]. Esta modificación también trae consigo un beneficio en un consumo de área mucho menor en comparación con otras aproximaciones.

### 1.5.6 Procesador Celular Analógico de Imágenes (CISP)

La arquitectura propuesta por G. Erten *et al.* [21], puede considerarse como una extensión de la idea original de la CNN, en donde las neuronas como elementos con capacidad de adquisición de imagen y de procesamiento basado únicamente en conexiones locales permiten la realización de tareas complejas globales en imágenes de manera paralela.

En la mayoría de la implementaciones físicas de la CNN, la etapa de la sumatoria de los productos que corresponden a las matrices  $A$  y  $B$  son llevadas a cabo en modo de corriente mediante la Ley de Corrientes de Kirchoff, debido a las ventajas que esto representa. Esto quiere decir que en un sólo nodo  $x$  inciden todas las corrientes resultantes de los productos haciéndose en este punto la suma total de ellas. Según Erten debido a los

valores intrínsecos de resistencia  $R$  grandes y los valores de capacitancia  $C$  muy pequeñas en el nodo  $x$ , la cantidad de carga requerida para mantener un voltaje particular en ese nodo es muy pequeña, por lo tanto, cualquier corriente proveniente de ruido es suficiente para alterar el voltaje en dicho nodo.

Para superar la consideración antes mencionada, en el modelo se introduce una estructura de retroalimentación autonormalizada para el cálculo de las entradas y la mascarilla de control. De manera, que las operaciones de las mascarillas tanto  $A$  como  $B$  permiten que el nodo  $x$  rápidamente acumule o pierda carga. Así, las ecuaciones dinámicas para cada neurona se modifican tomando la forma:

*Ecuación de estado.*

$$C \frac{dv_{xij}}{dt} = \frac{-v_{xij}}{R} + \sum_{kl \in N_r(ij)} A_{kl} v_{ykl}(t) + \sum_{kl \in N_r(ij)} B_{kl} \tanh \left[ \frac{qk}{2kT} (v_{ukl}(t) - v_{xij}(t)) \right] + I_{ij}$$

*Ecuación de salida*

$$v_y = f(v_x) = \tanh \left[ \frac{qk}{2kT} (v_x - V_{ref}) \right], y$$

$$I_{ij} = I$$

El modelo antes mencionado permite que la estructura estabilice la variable de estado en el nodo  $x$ , que de lo contrario tendería hacia valores extremos de magnitud  $V_{dd}$  ó  $V_{ss}$ , los cuales son los potenciales aplicados a la CNN. La estructura es mucho menos vulnerable a aspectos relacionados con la manufactura tal como el desacoplamiento (*mismatch*) entre transistores.

### 1.5.7 Máquina Universal CNN

Muchas tareas en el procesamiento de imágenes pueden ser resueltas con una CNN, pero por lo regular la resolución de tareas complejas resultan de la ejecución de un grupo de operaciones morfológicas de las imágenes. De ahí, la necesidad de una arquitectura que permita la programación algorítmica de mascarillas para la realización de operaciones morfológicas de manera secuencial.

La máquina universal CNN es precisamente, una arquitectura computacional capaz de implementar algoritmos compuestos de instrucciones analógicas (mascarillas) y lógicas,

donde la idea principal es que operaciones espacio-temporales predefinidas puedan ser ejecutadas con una sola instrucción y programadas en unidades que permitan la realización de tareas secuenciales.

La arquitectura de una máquina universal CNN puede encontrarse en [22] y está basada en dos partes principales: 1) El arreglo de neuronas CNN, y 2) La unidad analógica de programación global (GAPU).

A través del tiempo han existido muchas implementaciones de este tipo de estructuras en circuitos integrados. En la tabla 5.1 se citan algunas implementaciones y sus características.

### 1.6 Aplicaciones de las CNNs.

En esta sección, se dan algunos ejemplos de procesamiento de imágenes binarias ejecutados mediante una CNN. Aquí, se consideran sólo vecindarios de orden 1 para topologías rectangulares, que es la característica más común relacionada con implementaciones en circuitos integrados. Sólo se consideran mascarillas lineales invariantes en el tiempo y en el espacio.

#### Detector de Bordes

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -2 & 0 \\ -2 & 8 & -2 \\ 0 & -2 & 0 \end{bmatrix} \quad I = -1$$

La salida representa la extracción de bordes de la imagen de entrada



Imagen de Entrada



Imagen de Salida

**Removedor de Líneas Verticales**

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad I = -2$$

La salida representa a la entrada sin líneas verticales y aquellas partes que pueden ser interpretadas como tales.



Imagen de Entrada

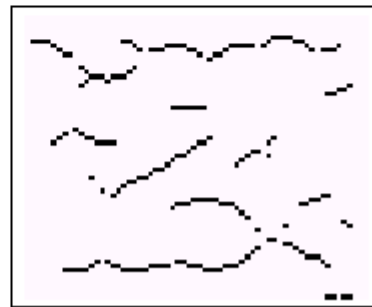


Imagen de Salida

**CCD o *Detector de Componentes Conectados***

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad I = -1$$

La imagen de salida muestra el número de componentes conectados en determinada dirección.

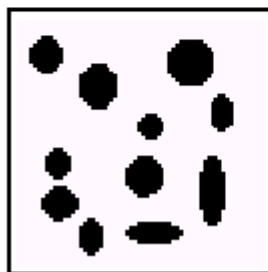
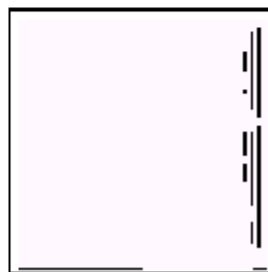


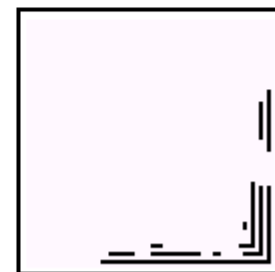
Imagen de Entrada



Salida Horizontal



Salida Vertical



Salida Diagonal

**Halftoning** caso  $3 \times 3$

$$A = \begin{bmatrix} -0.07 & -0.1 & -0.07 \\ -0.1 & 1 + \epsilon & -0.1 \\ -0.07 & -0.1 & -0.07 \end{bmatrix} \quad B = \begin{bmatrix} 0.07 & 0.1 & 0.07 \\ 0.1 & 0.32 & 0.1 \\ 0.07 & 0.1 & 0.07 \end{bmatrix} \quad I = 0$$

La salida es una imagen binaria que preserva las características principales de la imagen de entrada en escala de grises. La velocidad de convergencia es controlada por  $\epsilon \approx [0.1, \dots, 1]$ , donde a medida que  $\epsilon$  es mayor, la respuesta es más rápida pero la imagen es más áspera.



Imagen de Entrada



Imagen de Salida

**Umbral** *De Imagen Gris a Imagen Binaria*

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{array}{l} I = -I^* \\ -1 < -I^* < 1 \end{array}$$

La salida deseada es una imagen binaria donde los píxeles negros corresponden a localidades de la entrada donde la intensidad en la escala de grises es mayor que el umbral  $I^*$ .

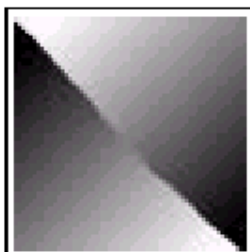
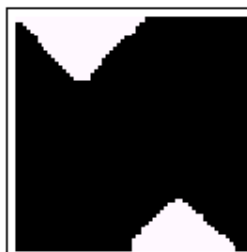
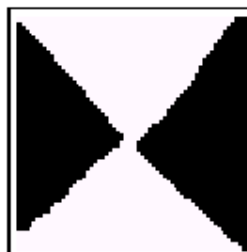


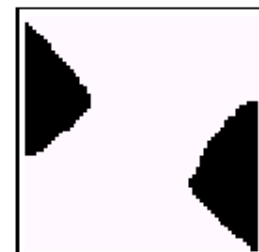
Imagen de Entrada



Salida  $I^* = -0.5$



Salida  $I^* = 0$



Salida  $I^* = 0.5$



**Rellenado de Huecos**

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad I = -4$$

La salida es el resultado de la imagen de entrada con los “huecos” ( representados por trayectorias cerradas ) rellenos con fondo negro.

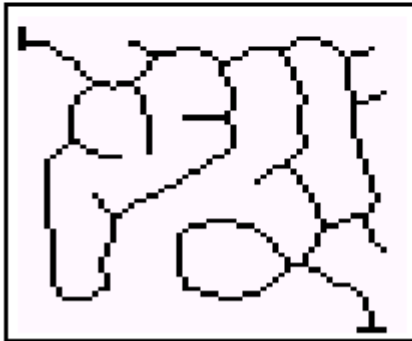


Imagen de Entrada

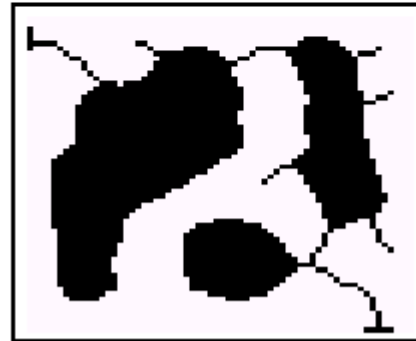


Imagen de Salida

**Sombreado** ◦ *Extractor de Sombras*

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad I = 0$$

La salida representa la “sombra” de la imagen de entrada con corrimiento hacia la izquierda.

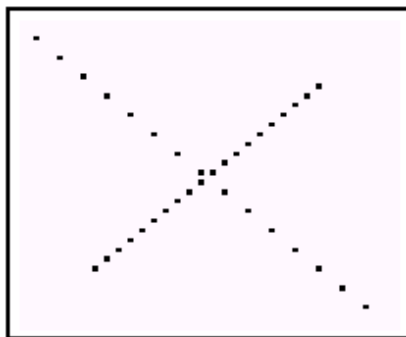


Imagen de Entrada



Imagen de Salida

**Detector de Esquinas** caso: *Detección de Esquinas Convexas*

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad I = -8.5$$

La salida es una imagen binaria donde los píxeles negros corresponden a las esquinas convexas de la imagen de entrada.

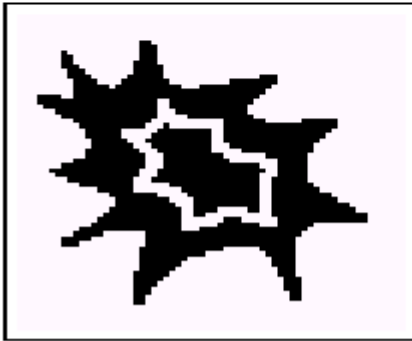


Imagen de Entrada

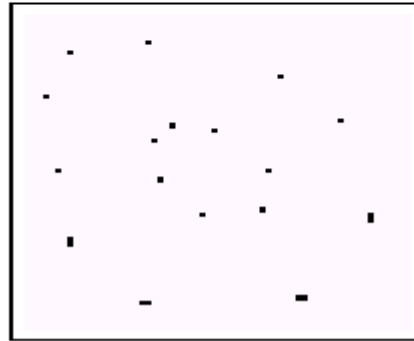


Imagen de Salida

## Referencias

- [1] Leon O. Chua and L. Yang, "Cellular Neural Networks: Theory" IEEE Trans. on Circuits and Systems, vol. 35, pp 1257-1272, Oct. 1988.
- [2] Leon O. Chua, "CNN: A Paradigm for Complexity" Series A, vol. 31 Ed. World Scientific Publishing 1998.
- [3] H. Harrer, "Multiple Layer Discrete-Time Cellular Neural Networks Using Time-Variant templates" IEEE Trans. On Circuits and Systems II vol. 40, no. 3 March 1993.
- [4] Servando Espejo, Rafael Dominguez-Castro, Gustavo Liñan, Angel Rodriguez-Vazquez, "A 64 x 64 CNN Universal Chip with Analog and Digital I/O" IEEE International Conference on Electronics Circuits and Systems (ICECS/98) pp. 203-206 Lisbon Sept. 1998.
- [5] Leon O. Chua and Tamas Roska, "Stability of a Class of Non-Reciprocal Cellular Neural Network" IEEE Trans. On Circuits and Systems, vol. 37 no. 12 December 1990.
- [6] T. Roska and L.O. Chua, "Cellular Neural Networks with Nonlinear and Delay-type Template Elements", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'90), pp.12-25, Budapest, 1990,
- [7] T. Roska, C.W. Wu, M. Balsi, L.O. Chua, "Stability and Dynamics of Delay-Type General and Cellular Neural Networks", IEEE Trans. on Circuits and Systems, I, 39(6), 487-490 (1992).
- [8] P. Civalleri, M. Gilli, L. Pandolfi, " On Stability of cellular Neural networks with Delay" IEEE Trans. On Circuits and Systems I, vol 40, no. 3 March 1993.
- [9] T. Roska, J. Vandewalle, "Cellular Neural Networks" John Wiley Sons 1993.
- [10] T. Roska, Angel Rodriguez-Vazquez, "Toward the Visual Microprocessor" John Wiley Sons 2001.
- [11] Z. Galias, " Designing Cellular Neural Networks for the Evaluation of Local Boolean Functions" IEEE Trans. On Circuits and Systems II, vol. 40, no. 3 March 1993.
- [12] Angel Rodriguez-Vazquez, Servando Espejo, Rafael Dominguez-Castro, J.L. Huertas and E. Sánchez-Sinencio, "Current Mode Techniques for the Implementation of Continuous and Discrete-Time Cellular Neural Networks" IEEE Trans on Circuits and Systems II, vol. 40, no. 3, March 1993.
- [13] S. Espejo, A. Rodríguez-Vazquez, R. Domínguez-Castro and R. Carmona, "Convergence and Stability of the FSR CNN Model" CNNA'94 Third IEEE International Workshop on Cellular Neural Networks and their Applications. Rome, 1994.
- [14] L. Yang, L.O. Chua, and K.R. Krieg, "VLSI implementation of cellular neural networks", Proceedings of IEEE International Symposium on Circuits and Systems, (ISCAS'90), pp.2425-2427, 1990.
- [15] H. Harrer, J. A. Nossek, R. Stelz, "An Analog Implementation on Discrete-Time Cellular Neural Network" IEEE Trans. On Circuits and Systems, vol.3,no. 3, May 1992.
- [16] A. Rodriguez-Vazquez, R. Dominguez-Castro and J.L. Huertas, "Accurate Design of Analog CNN in CMOS Digital Technologies", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'90), pp.273-280, Budapest, 1990,
- [17] A. Paasio and A. Dawidziuk, "CNN Template Robustness with Different Output Nonlinearities" Int. J. Circuit Theory Appl. Vol. 27, no. 1, pp 87-102, 1999.
- [18] L. O. Chua, P. Thiran, " An Analytic Method for Designing Simple Cellular Neural Networks" IEEE Trans. On Circuits and Systems, vol. 38, no. 11, November 1991.
- [19] I. Fajfar and F. Bratkovic, " Design of Monotonic Binary-Valued Cellular Neural Networks" CNNA'96, Fourth IEEE International Workshop on Cellular Neural Networks and their Applications. Seville, 1996.
- [20] A. Paasio and K. Halonen, " A New Cell Output Nonlinearity for Dense Cellular Nonlinear Network Integration" IEEE Trans. On Circuits and Systems I, vol. 48, no. 3 March 2001.
- [21] G. Erten and F. M. Salam, "Two Cellular Architectures for Integrated Image Sensing and Processing on a Simple Chip" Journal of Circuits, Systems and Computers, vol. 8, no. 5 y 6, pp. 637-659.
- [22] T. Roska, L. O. Chua, " The CNN Universal Machine: An Analogic Array Computer" IEEE Trans. on Circuits and Systems II, vol. 40, no. 3 March 1993.

# CAPÍTULO 2



DISEÑO DE MASCARILLAS DE  
LA RED NEURONAL CELULAR

## 2.0 Introducción

La tarea específica que desempeña una CNN sobre una imagen de entrada, como se señaló en el capítulo 1, está definida por los valores de los parámetros en las mascarillas  $A$ ,  $B$ , el umbral  $I$  y el estado inicial  $v_{xij}(0)$ , por lo que el diseño de una CNN se enfoca a encontrar dichos parámetros para una tarea específica de procesamiento.

En la CNN, a diferencia de otras estructuras neuronales, no existe una técnica universal para determinar los parámetros que permiten procesar una tarea específica. Los métodos desde el nacimiento de la CNN, sin contar con aquellos que se derivan del conocimiento empírico, pueden agruparse como: **métodos analíticos** [1],[2], los cuales se basan en una serie de reglas locales que caracterizan la dinámica de la neurona y forman un conjunto de desigualdades que deben ser resueltas mediante un método de optimización para obtener los valores de las mascarillas; **algoritmos de aprendizaje local** [3]-[5], en ellos se extienden los algoritmos de entrenamiento de otras estructuras neuronales a las CNN, tal como “*recurrent backpropagation*”, “*backpropagation through time*” y el aprendizaje de “*perceptron*”. Y **algoritmos de aprendizaje global** [6], que como el anterior utiliza también pares de datos entrada-salida, pero que son presentados de manera global a toda la red y no solamente a una neurona de forma local. Comúnmente, en este tipo de aprendizaje, se cuenta con metodologías de optimización que no consideran el uso de derivadas, tales como algoritmos genéticos y recocido simulado. La utilización de cada método depende del problema particular a resolver, ya que en cada uno de ellos existe una relación entre la exactitud y el costo computacional. Es necesario definir el método a seguir según nuestros requerimientos.

### 2.1 Diseño de Mascarillas Mediante el Método Simplex

Este método puede considerarse entre los *métodos analíticos* y de *aprendizaje local*, ya que reúne características de ambos. Fue propuesto por K. Nakai y A. Ushida [7] y el principio del diseño se basa en forzar la salida de cada neurona en respuesta a un patrón de entrada considerando solamente el estado inicial y el estado estable deseado para la neurona, de manera que se obtiene un sistema de desigualdades lineales que después es resuelto mediante el algoritmo Simplex.

#### 2.1.0 Desigualdades para el Estado Estable

Para una CNN de dimensión  $M \times N$  con vecindario de orden 1 y considerando la convención de valores extremos de imagen: +1 para un píxel negro y -1 para blanco, así como la condición de estabilidad (1.3). La ecuación de estado (1.1) en estado de equilibrio con  $dv_{xij}/dt = 0$  resulta en:

$$v_{xij} = \sum_{C(k,l) \in N_r(i,j)} A(i,j:k,l)v_{ykl} + \sum_{C(k,l) \in N_r(i,j)} B(i,j:k,l)v_{ukl} + I_{ij} \quad (2.0)$$

donde  $v_{xij}$  y  $v_{ykl}$  son estados estacionarios.

Considerando un patrón de entrada  $v_{uij}$ , y una salida deseada  $v_{yij}$ , las siguientes relaciones deben ser satisfechas como desigualdades:

- A) Siguiendo lo establecido por la función de salida UG (1.2), para una salida  $v_{yij} = 1$ , debe cumplirse  $v_{xij} \geq 1$  por lo que (2.0) debe satisfacer:

$$1 \leq \sum_{C(k,l) \in N_r(i,j)} A(i, j : k, l) v_{ykl} + \sum_{C(k,l) \in N_r(i,j)} B(i, j : k, l) v_{ukl} + I_{ij}$$

- B) Para una salida  $v_{yij} = -1$ , debe cumplirse  $v_{xij} \leq -1$  por lo que (2.0) debe satisfacer:

$$-1 \geq \sum_{C(k,l) \in N_r(i,j)} A(i, j : k, l) v_{ykl} + \sum_{C(k,l) \in N_r(i,j)} B(i, j : k, l) v_{ukl} + I_{ij}$$

### 2.1.1 Desigualdades para el Estado Inicial

Las desigualdades que deben de considerarse para el estado inicial son las siguientes:

- C) Cuando se desea la salida  $v_{yij}(\infty) = 1$ , es necesario que para el caso en que  $v_{xij}(0) < 1$ , sobre la región lineal de la función de salida UG, debe satisfacerse  $dv_{xij}/dt > 0$  para garantizar el cruce de la variable de estado por toda esta región y alcanzar el punto de saturación +1. Consecuentemente la ecuación de estado (1.1) para esta condición de estado inicial debe satisfacer:

$$0 < -v_{xij}(0) + \sum_{C(k,l) \in N_r(i,j)} A(i, j; k, l) v_{ykl}(0) + \sum_{C(k,l) \in N_r(i,j)} B(i, j; k, l) v_{ukl}(0) + I_{ij}$$

- D) Cuando se desea la salida  $v_{yij}(\infty) = -1$ , es necesario que para el caso en que  $v_{xij}(0) > -1$ , en la región lineal debe satisfacerse  $dv_{xij}/dt < 0$  para garantizar el cruce de la variable de estado por toda esta región. Consecuentemente (1.1) para este caso debe cumplir:

$$0 > -v_{xij}(0) + \sum_{C(k,l) \in N_r(i,j)} A(i, j; k, l) v_{ykl}(0) + \sum_{C(k,l) \in N_r(i,j)} B(i, j; k, l) v_{ukl}(0) + I_{ij}$$

donde:

$$v_{xij}(0) = v_{uij}$$

expresa la condición inicial del estado y la salidas para las condiciones (A), (B), (C) y (D) se consideran:

$$v_{ykl} = \begin{cases} v_{xij}(0) & k = i, l = j \\ v_{ykl}(\infty) & k \neq i, l \neq j \end{cases} \quad (2.1)$$

donde  $v_{ykl}(\infty)$  en (2.1), se refiere al hecho de que la salida de las demás neuronas han alcanzado el estado estable, tal consideración es valida como se menciona en [2] si las salidas de las demás neuronas tienen una evolución monótonica en el tiempo.

### 2.1.2 Análisis de la Dinámica de la CNN

Como (2.1) es válida para el análisis de CNN en el tratamiento de imágenes binarias y como las entradas  $v_{ukl}$  se consideran fijas en el tiempo, podemos considerar a (1.1) como:

$$\frac{dv_{xij}}{dt} = -v_{xij} + a_o v_{yij} + k_1 \quad (2.2)$$

Donde  $k_1$  (en estado estable) es definido como:

$$k_1 = \sum_{C(k,l) \in N_r(i,j)/C_{ij}} A(k,l)v_{ykl} + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{ukl} + I_{ij}$$

y por lo tanto sólo toma valores discretos.

La notación  $C(k,l) \in N_r(i,j)/C_{ij}$  denota que se incluyen todas las neuronas del vecindario de  $C_{ij}$  con excepción de ella misma, de tal forma que  $a_o$  en (2.2) es el elemento de la matriz de retroalimentación  $A(i,j;i,j)$  que corresponde a la salida de la neurona  $C_{ij}$ .

Si se toma la ecuación (2.2) en la región lineal de la función UG de salida, es decir, donde  $v_{yij} = v_{xij}$ , tenemos la ecuación diferencial en la forma:

$$\frac{dv_{xij}}{dt} = -v_{xij} + a_o v_{xij} + k_1$$

cuya solución esta dada por:

$$v_{xij}(t) = \frac{k_1}{1 - a_o} + C e^{(a_o - 1)t} \quad (2.3)$$

donde  $C$  es una constante.

En esta ecuación es fácil observar del por qué la necesidad de que el elemento de retroalimentación satisfaga  $a_o > 1$ , como es establecido en [8] para el caso de imágenes binarias, ya que ello garantiza que el término exponencial en (2.3) sea una función creciente

e independientemente del valor de  $k_1 / (1 - a_o)$ , se asegura que la variable de estado salga de la región lineal y se dirija hacia los puntos de saturación de la función de salida UG. En este sentido, las desigualdades para la condición inicial (C) y (D) de la sección anterior, hacen la función de determinar una desigualdad que establezca el signo del término  $Ce^{(a_o-1)t}$  propio para salir de la región lineal y alcanzar el punto de saturación deseado.

Para las regiones de saturación de la función de salida UG puede hacerse un análisis similar, esto es, ahora cuando  $v_{yij} = 1$ , por lo que podemos definir:

$$k_2 = k_1 a_o \tag{2.4}$$

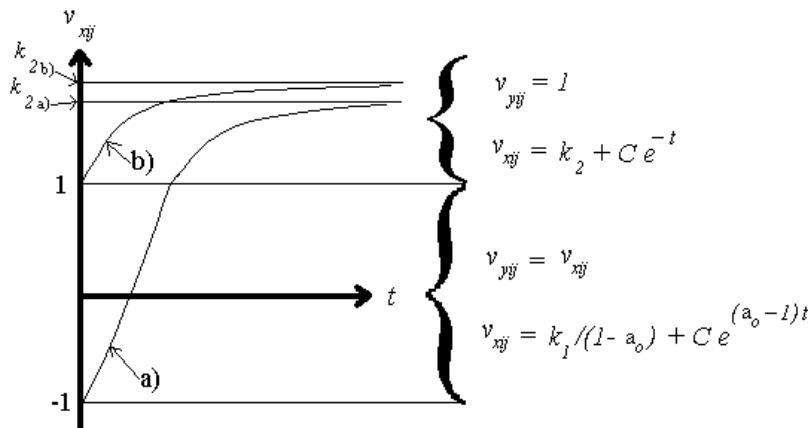
de tal forma que la ecuación (2.2) queda:

$$\frac{dv_{xij}}{dt} = -v_{xij} + k_2$$

y su solución está dada por:

$$v_{xij}(t) = k_2 + Ce^{-t} \tag{2.5}$$

Como se puede ver, el segundo término de (2.5) es una función decreciente que tiende asintóticamente al valor de  $k_2$ . En la figura 2.0 se da un ejemplo esquemático de la variable de estado de una neurona contra el tiempo para dos casos diferentes de salida negra con condiciones iniciales  $v_{xij}(0)=1$  y  $v_{xij}(0)=-1$ .



**Figura 2.0** Evolución de la variable de estado para dos casos, a)  $v_{xij}(0) = -1$  y b)  $v_{xij}(0) = 1$

Un sistema de desigualdades en la forma de (A), (B), (C) y (D) puede obtenerse del planteamiento de un problema específico a resolver, de manera que es necesario definir una función objetivo y un método para la solución del sistema. En la literatura existen muchas propuestas al respecto como la de [9], donde se utiliza el método de relajación para resolver sistemas de desigualdades.



### 2.1.3 Función Objetivo

En nuestro caso, para la elaboración de una función objetivo adecuada, es necesario definir primeramente que la solución óptima para nuestro sistema será aquella que más se aproxime al centro del área solución de las desigualdades. Esto es debido a que en una implementación física, toda solución estará sujeta a la existencia de variaciones provocadas por cambios aleatorios de parámetros tecnológicos y tolerancias propias de los circuitos que constituyen el circuito integrado, por lo que es necesario que la solución a pesar de dichas variaciones permanezca dentro del área y asegure el buen funcionamiento del sistema. Si definimos una función objetivo adecuada, es posible encontrar una solución cercana al centroide del área solución que sea robusta a tales variaciones.

Este problema de optimización con restricciones, puede ser transformado a un problema de optimización sin restricciones mediante el uso de *funciones de penalización* [10]. Si reescribimos todas las desigualdades en la forma:

$$f_k(\mathbf{T}) \geq 0$$

donde  $\mathbf{T}$  son los coeficientes de las mascarillas.

La función objetivo puede ser definida como:

$$F(\mathbf{T}) = \sum f_k(\mathbf{T})^2$$

Para hacer posible que ésta busque la solución dentro del área, una constante  $\phi$  es introducida en la ecuación anterior, así como la función de penalización, de tal manera que queda en la forma:

$$\hat{F}(\mathbf{T}) = \sum \hat{f}_k(\mathbf{T})^2$$

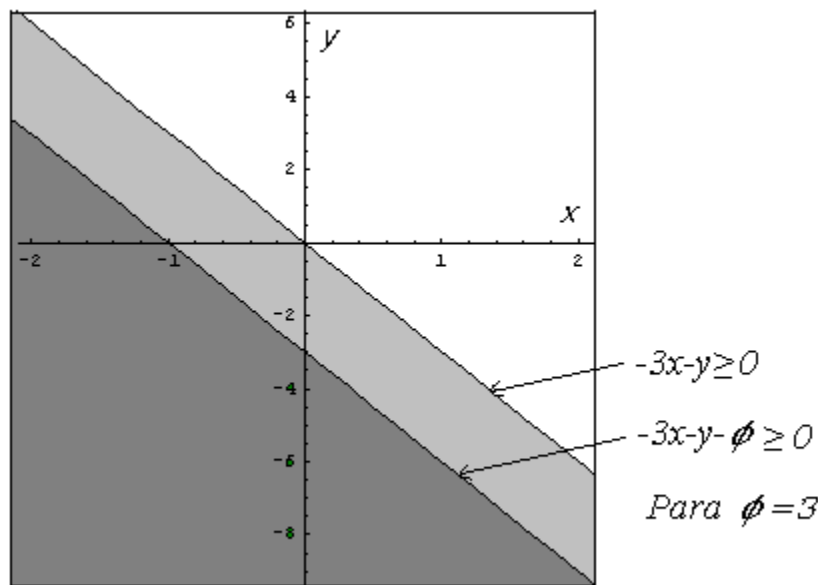
donde:

$$\hat{f}_k(\mathbf{T}) = \begin{cases} f_k(\mathbf{T}) - \phi & f_k(\mathbf{T}) - \phi \geq 0 \\ p + m \cdot (f_k(\mathbf{T}) - \phi) & f_k(\mathbf{T}) - \phi < 0 \end{cases} \quad (2.6)$$

con  $p$  y  $m$  como cantidades reales positivas grandes.

Para toda desigualdad en la forma  $f_k(\mathbf{T}) \geq 0$ , la sustracción de una constante  $\phi$  al término  $f_k(\mathbf{T})$  corresponde a un desplazamiento de la frontera reduciendo el área solución de

dicha desigualdad. Tal como se muestra en la figura 2.1 dicha característica permite que a valores de  $\phi$  grandes la solución se aproxime más al centroide del área solución.



**Figura 2.1** Reducción del área solución con la sustracción de la constante  $Z$ .

La función lineal  $p+mx$  definida en (2.6) es la función de penalización que hace que la función objetivo tome valores muy grandes (para el caso de una minimización) cuando  $\mathbf{T}$  no cumple con una o varias de las desigualdades.

#### 2.1.4 Algoritmo Simplex

Ahora, ya definida la expresión de la función objetivo, se hará la descripción del método Simplex (*Downhill Simplex Search*). Como se define en [11], este es un método de optimización multidimensional que no involucra el uso de derivadas en su algoritmo, su única similitud con aquél de nombre similar para programación lineal, radica en determinar un conjunto de puntos geométricos que son evaluados desde el punto de vista de la función objetivo. Lo simple de su algoritmo hace que sea fácilmente implementado mediante un programa, aunque no sea tan eficiente como aquellos que involucran el uso de derivadas.

La idea básica del método consiste primeramente en definir una colección de puntos  $(n+1)$  en el espacio  $n$ -dimensional. El punto mínimo de la función objetivo es buscado mediante la sustitución de aquellos puntos más desfavorables desde el punto de vista de la función objetivo, por otros más favorables. La continua iteración de lo anteriormente expuesto permite que la colección de puntos converja hacia el mínimo de la función.

Para el algoritmo es necesario definir el Simplex (colección de puntos) como:  $\mathbf{T}(i)$  para  $(i=1, \dots, n+1)$  y después de evaluarlos en la función objetivo es necesario definir:

1. Aquel punto que devuelve un valor máximo desde el punto de vista de dicha función, esto es:

$$\mathbf{T}_{H1} = \text{Max } \hat{F}(\mathbf{T}(i))$$

2. Aquel que devuelve el segundo valor mayor :

$$\mathbf{T}_{H2} = \text{Max } \hat{F}(\mathbf{T}(i)), (i \neq H1), \text{ y}$$

3. El que devuelve el valor mínimo :

$$\mathbf{T}_{min} = \text{Min } \hat{F}(\mathbf{T}(i))$$

La sustitución de puntos por otros más favorables se lleva a cabo mediante un algoritmo que involucra los tres siguientes tipos de operaciones :

I) **Reflexión** de  $\mathbf{T}_{H1}$ , considerando  $m_r > 0$  se define como:

$$\mathbf{T}(\mathbf{R}) = \mathbf{T}_{H1} + (1 + m_r)(\mathbf{T}_a - \mathbf{T}_{H1})$$

donde  $\mathbf{T}_a$  es el centroide de los puntos  $\mathbf{T}(i)$  calculado de la forma:

$$\mathbf{T}_a = \frac{\sum_{i=1}^{n+1} \mathbf{T}(i)}{n+1}$$

En la figura 2.2 se muestra una representación gráfica en contornos del mínimo de una función, en donde cuatro puntos alrededor de él definen el Simplex. En la gráfica se observa el desplazamiento del peor punto del Simplex (en negro) en dirección del vector de Reflexión .

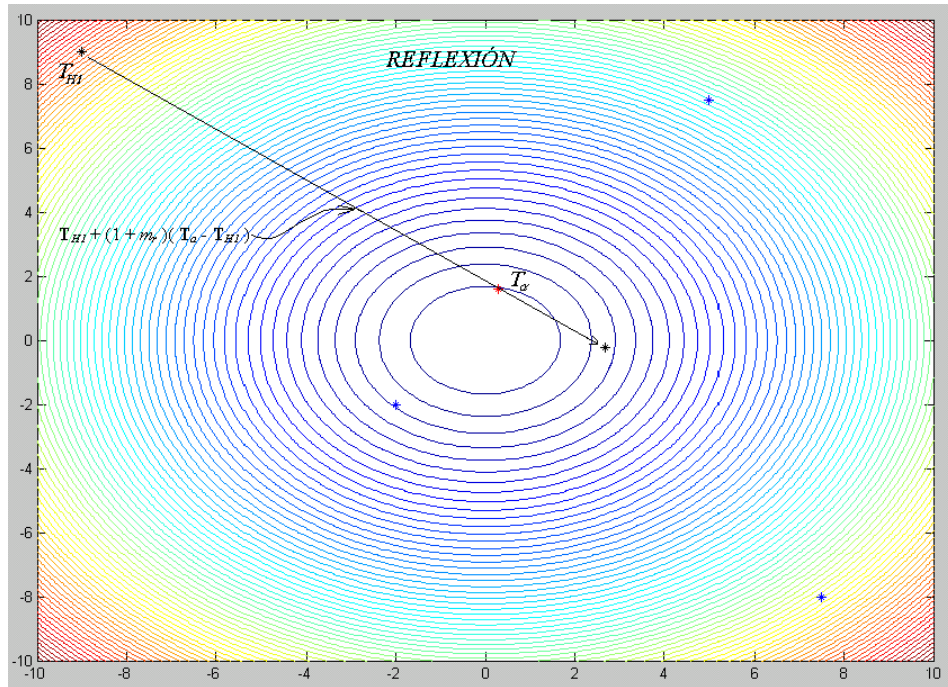


Figura 2.2 Reflexión.

II) **Expansión** de  $T_{HI}$ , considerando  $m_e > m_r$ , se define como:

$$\mathbf{T}(E) = \mathbf{T}_{HI} + (1 + m_e)(\mathbf{T}_a - \mathbf{T}_{HI})$$

En la figura 2.3 se muestra un ejemplo gráfico de la Expansión.

III) **Contracción** de  $T_{HI}$ , considerando  $0 < m_c < 1$ , se define:

$$\mathbf{T}(C) = \mathbf{T}_{HI} + m_c(\mathbf{T}_a - \mathbf{T}_{HI})$$

En la figura 2.4 se muestra una representación de la Contracción.

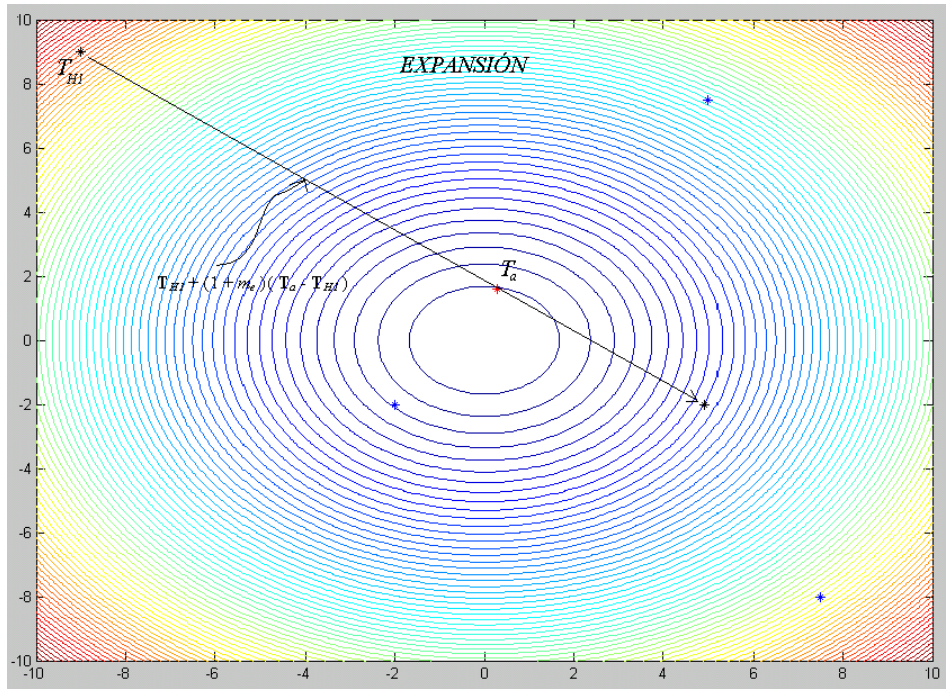


Figura 2.3 Expansión

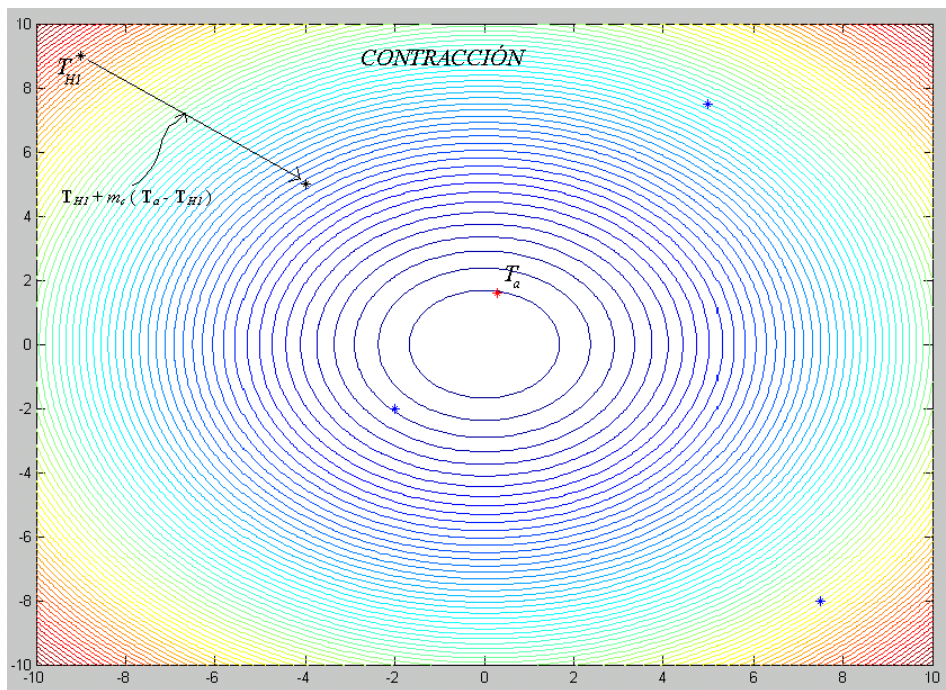
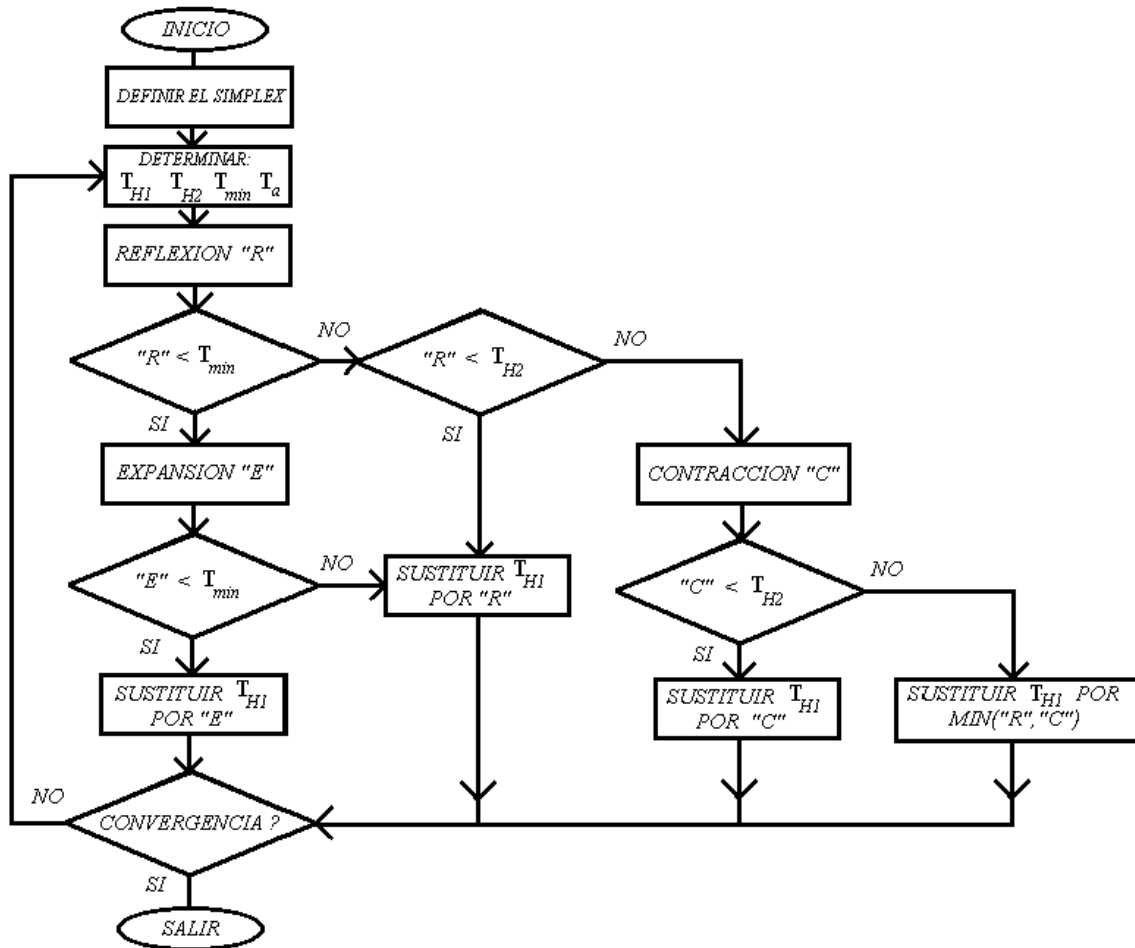


Figura 2.4 Contracción

Los valores óptimos de las constantes  $m_r$ ,  $m_e$  y  $m_c$  son dependientes de la aplicación en particular y la mejor manera de determinarlas es heurísticamente.

Con base en las tres operaciones antes mencionadas, el algoritmo del método se lleva a cabo de acuerdo con el siguiente diagrama de flujo.



### 2.1.5 Aplicación del Método Simplex a Sistemas de Desigualdades de Dos Variables

El algoritmo fue implementado satisfactoriamente en Matlab y fue probado primeramente con desigualdades lineales de sólo dos variables, para lo cual es posible observar de forma gráfica el desarrollo del algoritmo a través de las iteraciones.

### 2.1.5.0 Ejemplo 1.

Partiendo de las restricciones:  $x+y-2<0$ ,  $x-y>0$ ,  $x>0$  y  $y>0$  y siguiendo los pasos para la función objetivo con un  $\phi = 1$ , es posible obtener la siguiente función objetivo:

$$\hat{F}(\mathbf{T}) = [\hat{f}(x - y - 1)]^2 + [\hat{f}(-x - y + 2 - 1)]^2 + [\hat{f}(x - 1)]^2 + [\hat{f}(y - 1)]^2$$

Utilizando el algoritmo del método Simplex en Matlab, es posible localizar el mínimo de dicha función. En la figura 2.5 se muestran los resultados del programa en donde la inicialización del Simplex está representada por los puntos rojos alrededor del área solución delimitada por la forma triangular, cabe señalar que el punto negro en  $(0,1)$  hace referencia al  $\mathbf{T}_{HI}$  de la primera iteración. En la figura se observa como todos los puntos del Simplex convergen hacia el centro del área solución (mínimo de la función objetivo graficada en contornos) al punto  $(1,0.3393)$ , los centroides de cada operación están representados por los puntos en color violeta.

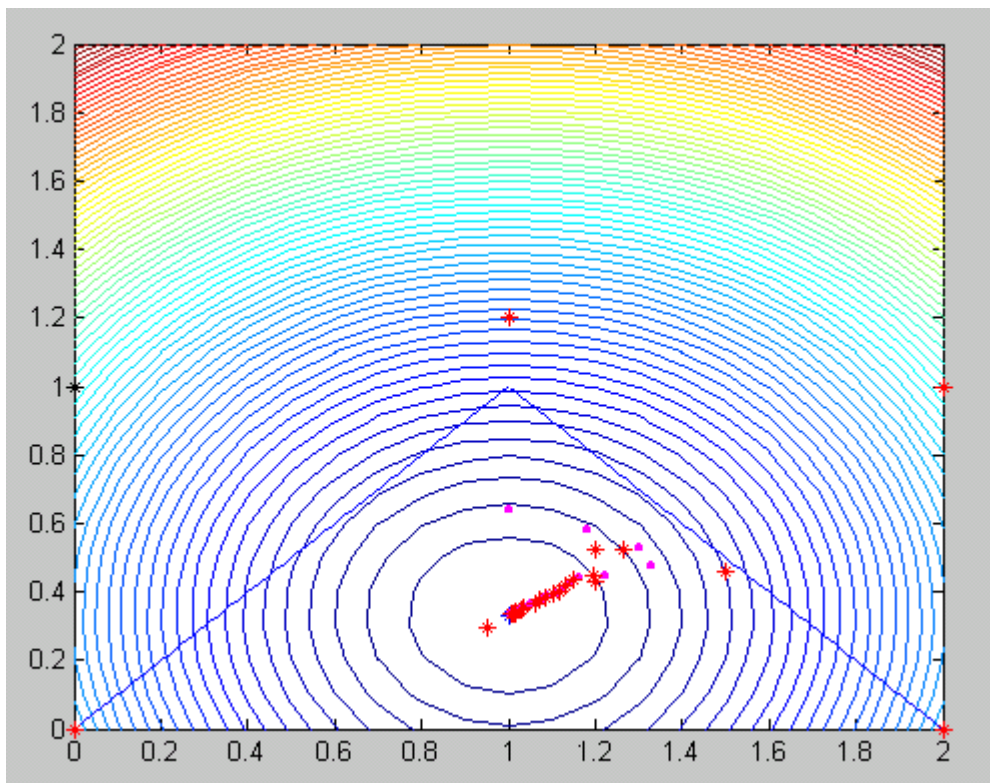


Figura 2.5 Resultados del método Simplex aplicado al ejemplo 1.

### 2.1.5.1 Ejemplo 2.

En este segundo ejemplo se consideran las desigualdades:  $6x+4y<24$ ,  $x+2y<6$ ,  $-x+y<1$ ,  $y<2$ ,  $x>0$ ,  $y>0$  y siguiendo los pasos para la función objetivo para un  $\phi= 2$  se obtiene:

$$\hat{F}(\mathbf{T}) = [\hat{f}(-6x-4y+24-2)]^2 + [\hat{f}(-x-2y+6-2)]^2 + [\hat{f}(x-y+1-2)]^2 + [\hat{f}(-y+2-2)]^2 + [\hat{f}(x-2)]^2 + [\hat{f}(y-2)]^2$$

En la figura 2.6 se observa el resultado del programa, la inicialización del Simplex, el mínimo de la función objetivo graficada en contornos, el área solución delimitada por líneas negras y la convergencia de los puntos hacia un punto cercano al centroide (en amarillo).

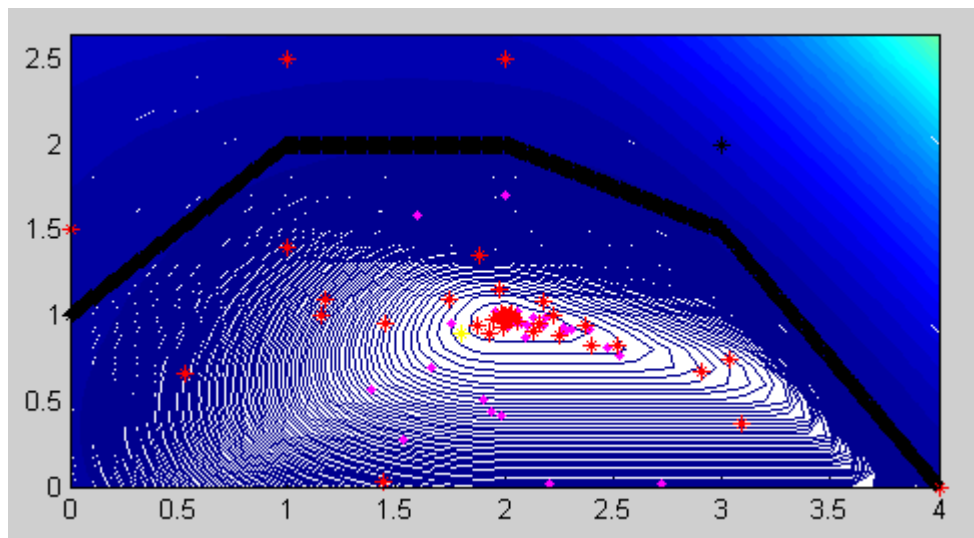


Figura 2.6 Resultados del método Simplex aplicado al ejemplo 2.



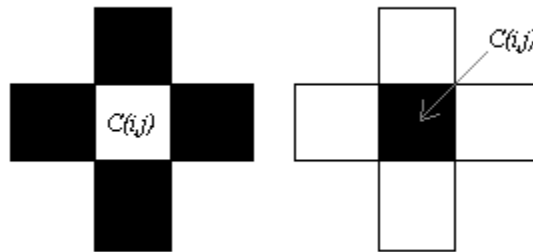
## 2.2 Aplicación del Método Simplex en el Diseño de Mascarillas de la CNN para Diferentes Tareas de Procesamiento

Ahora se procede a la aplicación del método Simplex para la obtención de los parámetros de las mascarillas en diferentes tareas de procesamiento.

### 2.2.0 Removedor de Ruido.

La característica propia de esta operación morfológica, radica en restaurar una imagen en blanco y negro cuando existe ruido en ella. Un píxel será considerado ruido cuando no tenga en sus vecinos inmediatos horizontales y verticales un píxel del mismo valor, de manera que deberá cambiar su salida al valor contrario. En la figura 2.7 se muestran las dos condiciones para el píxel considerado ruido.

En la literatura no existe procedimiento alguno para elegir la o las mascarillas propias para un problema en particular. Podría decirse de manera general que para aquellas tareas donde es necesaria la propagación de la información entre neuronas, las mascarilla  $A$  debe ser considerada, mientras que en aquellas para lo cual no es necesaria dicha propagación y sólo la información inicial de entrada es suficiente, puede proponerse utilizar la mascarilla  $B$ . En la literatura se encontrarán muchas soluciones diferentes para un mismo problema particular, donde algunas consideran el uso de las dos mascarillas  $A$  y  $B$  mientras que en otras sólo consideran el uso de una de ellas.



**Figura 2.7** Casos posibles para que el píxel  $C(i,j)$  sea considerado ruido.

Para este caso hacemos la propuesta de las mascarillas  $A$ ,  $B$  y el umbral  $I$  en una topología rectangular de orden uno en la forma:

$$A = \begin{bmatrix} 0 & a & 0 \\ a & a_0 & a \\ 0 & a & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I$$

donde  $a_0$ ,  $a$ ,  $b_0$  e  $I$  son los valores a determinar.

Ahora, si se considera todos los patrones de entrada posibles para una neurona y se define su salida deseada, tanto para neuronas regulares como de frontera (tomando en cuenta una frontera de tipo Dirichlet de valor cero), se puede llegar a un grupo de desigualdades que son definidas según el patrón de entrada y la condición inicial. Estas desigualdades se presentan en la tabla 2.1.

A continuación se muestra un ejemplo que describe como se obtiene cada una de las desigualdades:

Ejemplo.-

Para el caso 4 de la tabla 2.1 en negrillas, cuando un píxel tenga una entrada blanca con su respectiva condición inicial  $v_{xij}(0) = -1$  es necesario que se cumpla (C) para garantizar que la variable de estado cruce la región lineal de la función de salida UG hacia +1, por lo tanto:

$$0 < -v_{xij}(0) + \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)v_{ykl}(0) + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{ukl}(0) + I_{ij}$$

que para este caso queda:

$$0 < 1 + a_0 \cdot (-1) + a \cdot (1) + a \cdot (1) + a \cdot (1) + a \cdot (1) + b_0 \cdot (-1) + I$$

puesto que  $v_{yij}(0) = v_{xij}(0)$  como es definido en (2.1), simplificando se obtiene:

$$-1 < -a_0 + 4a - b_0 + I$$

Cuando la variable de estado alcance el valor de +1, es necesario garantizar que esta misma sobrepasará tal valor para la nueva salida de la neurona C(i,j) esto es, ahora para  $v_{yij} = +1$  y tenderá asintóticamente hacia un valor por encima de +1, de manera que para este mismo caso debe cumplirse también (A), esto es:

$$1 \leq \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l)v_{ykl} + \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l)v_{ukl} + I_{ij}$$

que para este caso toma la siguiente forma.

$$1 \leq a_0 \cdot (1) + a \cdot (1) + a \cdot (1) + a \cdot (1) + a \cdot (1) + b_0 \cdot (-1) + I$$

o sea,

$$1 \leq a_0 + 4a - b_0 + I$$

Y así de igual forma para los que tienen que hacer transición de negro a blanco. Para los casos donde la entrada y la salida deseada son del mismo valor, solamente se consideran las ecuaciones en estado estable, esto es debido a que para estos casos sólo es necesario que la variable de estado se mantenga en las zonas de saturación de la función de salida UG.

Tabla 2.1 Planteamiento para la obtención de las desigualdades del removedor de ruido.

CASO	Píxel de Entrada $V_{uij}$	Píxel de Salida $V_{yij}$	Neuronas del vecindario $C_{kl} \quad k \neq i, l \neq j$		Desigualdades
			No. de Negros	No. de Blancos	
(1)	Negro (+1)	Blanco(-1)	0	4	-ao - 4a + bo + l ≤ -1 B)
					ao - 4a + bo + l < 1 D)
			0	3	-ao - 3a + bo + l ≤ -1 B)
					ao - 3a + bo + l < 1 D)
(2)	Negro (+1)	Negro (+1)	1	3	ao - 2a + bo + l ≥ 1 A)
			1	2	ao - a + bo + l ≥ 1 A)
			2	2	ao + bo + l ≥ 1 A)
			2	1	ao + a + bo + l ≥ 1 A)
			3	1	ao + 2a + bo + l ≥ 1 A)
			3	0	ao + 3a + bo + l ≥ 1 A)
			4	0	ao + 4a + bo + l ≥ 1 A)
(3)	Blanco(-1)	Blanco(-1)	0	4	-ao - 4a - bo + l ≤ -1 B)
			0	3	-ao - 3a - bo + l ≤ -1 B)
			1	3	-ao - 2a - bo + l ≤ -1 B)
			1	2	-ao - a - bo + l ≤ -1 B)
			2	2	-ao - bo + l ≤ -1 B)
			2	1	-ao + a - bo + l ≤ -1 B)
			3	1	-ao + 2a - bo + l ≤ -1 B)
(4)	Blanco(-1)	Negro (+1)	3	0	ao + 3a - bo + l ≥ 1 A)
					-ao + 3a - bo + l > -1 C)
			<b>4</b>	<b>0</b>	<b>ao + 4a - bo + l ≥ 1 A)</b>
					<b>-ao + 4a - bo + l &gt; -1 C)</b>
Condición para Estabilidad					ao > 1

En Matlab se desarrolló un programa que ejecuta el método Simplex haciendo uso de interfaces gráficas de usuario que permiten crear un ambiente más amigable con el usuario. En dicho programa sólo es posible resolver problemas para combinaciones limitadas de mascarillas que pueden ser definidas en la interfaz, el valor de  $\phi$  puede ser ajustado a valores de 1, 2 ó 3 y el conjunto de desigualdades tiene que ser introducido en forma de matriz.

La matriz del removedor de ruido corresponde a las columnas 2 a 5 de la tabla 2.1, de manera que el programa determina las desigualdades correspondientes y resuelve el sistema según se le indique que elementos constituirán la mascarillas. Tal matriz, como debe ser introducida, queda en la siguiente forma tabular:

1	-1	0	4
1	-1	0	3
1	1	1	3
1	1	1	2
1	1	2	2
1	1	2	1
1	1	3	1
1	1	3	0
1	1	4	0
-1	-1	0	4
-1	-1	0	3
-1	-1	1	3
-1	-1	1	2
-1	-1	2	2
-1	-1	2	1
-1	-1	3	1
-1	1	3	0
-1	1	4	0

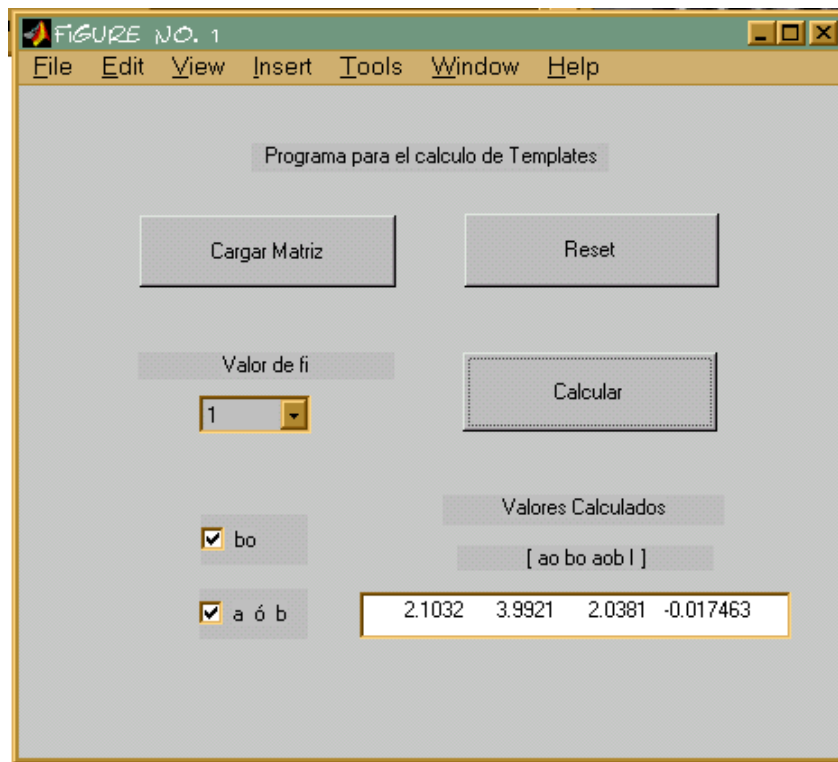
La pantalla a través de la cual se ingresan los datos hacia el programa en Matlab así como el resultado de éste se muestra en la figura 2.8.

Para verificar el resultado adecuado de los parámetros, se utilizó un simulador de CNN para imágenes binarias del Instituto de Procesamiento de Información y Señales de Zurich, con dirección electrónica <http://www.isi.ee.ethz.ch/~haenggi/CNNsim.html> con el que se hicieron los análisis de respuesta. En la figura 2.9 se muestra una de las pruebas en el simulador donde se observa la salida correcta de la CNN para la imagen de entrada. En el simulador existe la posibilidad de observar la trayectoria de evolución de la variable de estado en el tiempo mediante un visualizador de trayectoria (“trajectory viewer”), lo cual es de utilidad para observar la dinámica de neuronas específicas; en la figura 2.10 se muestra tal gráfica (donde  $x$  corresponde a  $v_{xij}$ ) en donde varias neuronas han sido señaladas en la imagen de entrada de la figura 2.9 mediante un triángulo cuyo color corresponde a cada trayectoria de la gráfica.

Debido a que los ejes de la gráfica de la variable de estado del visualizador de trayectoria se limita sólo al intervalo de  $v_{xij} = 5$  a  $v_{xij} = -5$ , no es posible observar en dicha gráfica la convergencia de cada una de las trayectorias para sus respectivos casos. Sin embargo, existe otro comando en el simulador que permite conocer para una neurona específica la variable de estado en cualquier momento. Considerando un tiempo grande en la simulación, el simulador devuelve los siguientes valores de convergencia para las diferentes neuronas:

- 1.- Azul = 14
- 2.- Negro = 6
- 3.- Rosa = -14
- 4.- Rojo = 6.2
- 5.- Verde = -6.2

Estos valores concuerdan exactamente con el cálculo que puede hacerse de  $k_2$  (ecuación (2.4)) para el caso que presenta cada neurona individual.



**Figura 2.8** Resultado del programa en Matlab para la mascarilla del removedor de ruido.

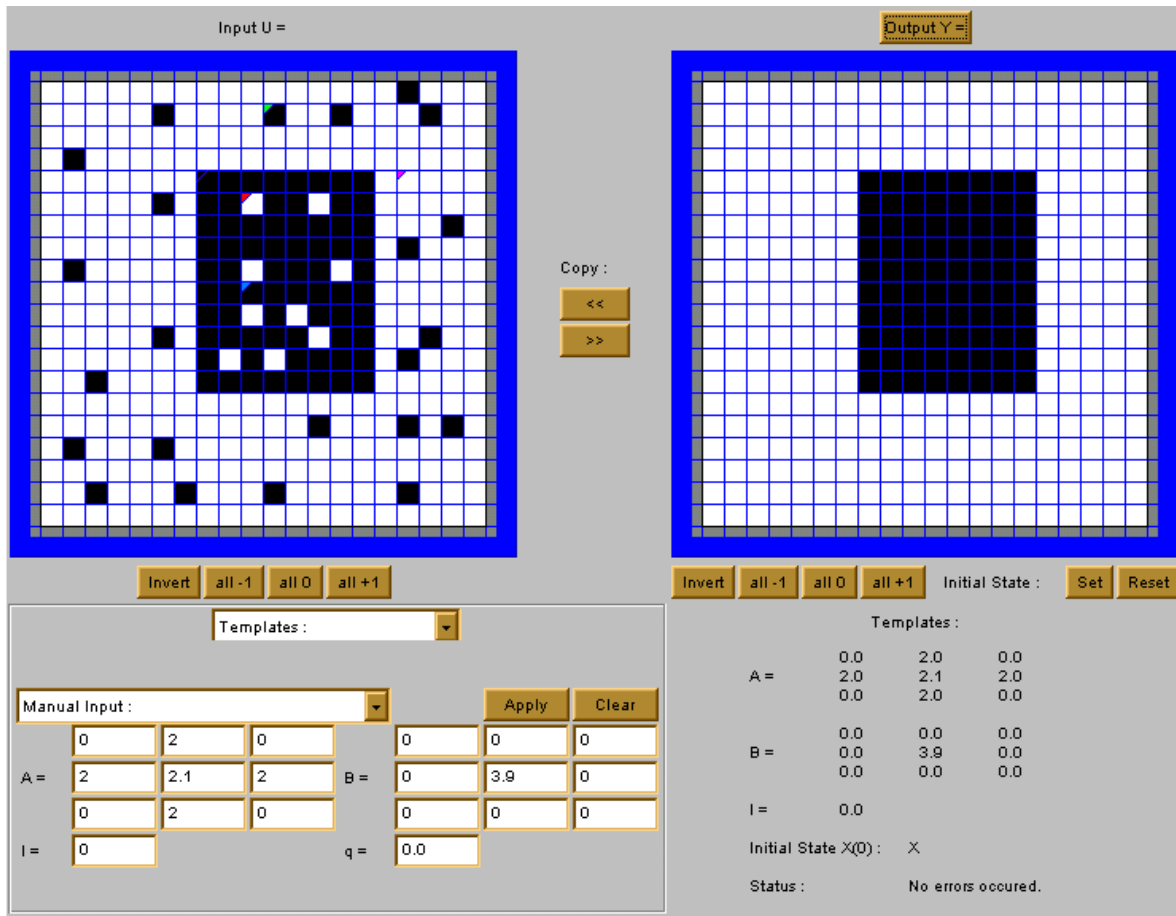


Figura 2.9 Resultados del simulador utilizando las mascarillas obtenidas mediante el método Simplex.

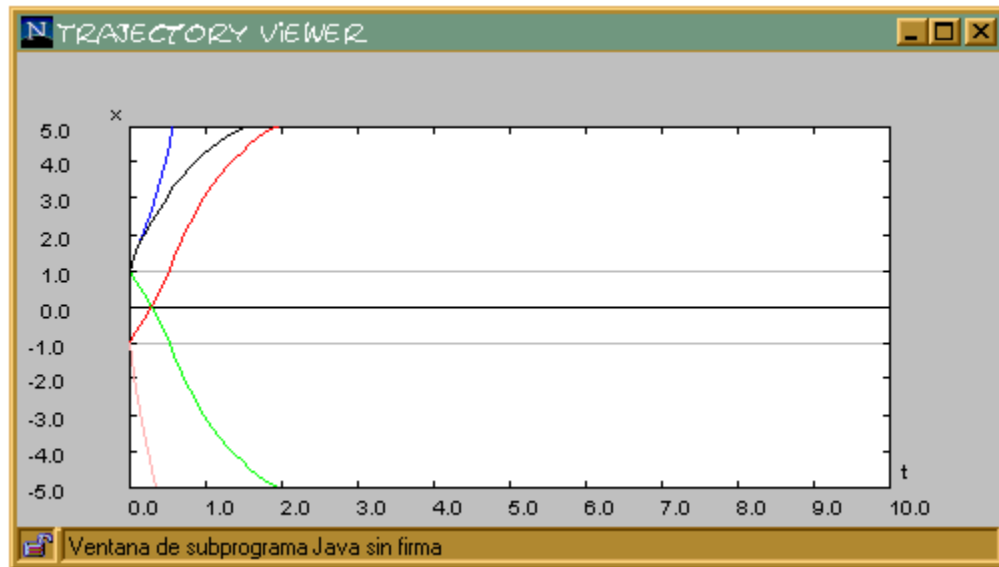


Figura 2.10 Gráfica que demuestra la evolución de la variable de estado de las neuronas señaladas.

### 2.2.1 Detector de Conectividad Global

El problema de la conectividad global fue un problema técnico importante en sistemas de procesamiento paralelo como se menciona en la referencia [12]. La propiedad intrínseca de la CNN para resolver este problema demuestra el gran potencial de la conexión local para hacer procesamiento global en imágenes.

El detector de conectividad global puede ser planteado como un problema de “Laberinto”, en donde la red tiene que encontrar el camino de la entrada hacia la salida desechando pseudorutas. La forma para solucionarlo se plantea de la siguiente forma:

- 1) El camino del laberinto siempre será blanco y su ancho de un píxel.
- 2) Si la entrada a una neurona es negra su salida siempre será negra.
- 3) Si la entrada a la neurona es blanca y existen tres vecinos con salida negra, la salida de ésta deberá ser negra.
- 4) Para todo caso de entrada blanca a excepción de (3) la salida deberá ser blanca.

Si tomamos en cuenta que es un problema que requiere de propagación de información entre neuronas, la mascarilla  $A$  para la solución de este problema tiene que ser considerada, de manera que las mascarillas que se proponen son como las del problema de la sección anterior, esto es, de la forma:

$$A = \begin{bmatrix} 0 & a & 0 \\ a & a_o & a \\ 0 & a & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I$$

Los resultados del programa en Matlab para la matriz que corresponde a la tabla 2.2, que contempla todos los casos de entrada-salida, se muestran en la figura 2.11. Una de las simulaciones correspondientes se muestra en la figura 2.12 para el caso de un laberinto, en donde se puede apreciar la salida correcta de la red. Otras simulaciones se hicieron para probar las mascarillas en problemas más complejos de conexión global, como aquel en la figura 2.13, en donde para los propios humanos resulta difícil determinar si las imágenes están globalmente conectadas o no.

Vale la pena señalar la diferencia entre la solución obtenida para este problema y la solución propuesta en [12], donde proponen el uso de más elementos para la mascarilla  $B$ . Debido al gran número de puntos estables en la CNN, no existe una solución única para los problemas, como se ha mencionado anteriormente.

Tabla 2.2 Planteamiento para la obtención de las desigualdades del detector de conectividad global.

CASOS	Píxel de Entrada $V_{uij}$	Píxel de Salida $V_{yij}$	Neuronas del vecindario $C_{kl} \quad k \neq i, l \neq j$		Desigualdades
			No. de Negros	No. de Blancos	
(1)	Blanco (-1)	Negro(1)	3	1	$ao + 2a - bo + l \geq 1$
			3	0	$-ao + 2a - bo + l > -1$
			4	0	$ao + 3a - bo + l \geq 1$
(2)	Negro (+1)	Negro (+1)	3	0	$-ao + 3a - bo + l > -1$
			4	0	$ao + 4a - bo + l \geq 1$
			4	0	$-ao + 4a - bo + l > -1$
			4	0	$ao + 4a + bo + l \geq 1$
			3	0	$ao + 3a + bo + l \geq 1$
			3	1	$ao + 2a + bo + l \geq 1$
			2	1	$ao + a + bo + l \geq 1$
			2	2	$ao + bo + l \geq 1$
			1	2	$ao - a + bo + l \geq 1$
(3)	Blanco(-1)	Blanco(-1)	1	3	$ao - 2a + bo + l \geq 1$
			0	3	$ao - 3a + bo + l \geq 1$
			0	4	$ao - 4a + bo + l \geq 1$
			2	2	$-ao - bo + l \leq -1$
			1	3	$-ao - 2a - bo + l \leq -1$
			0	4	$-ao - 4a - bo + l \leq -1$
			2	1	$-ao + a - bo + l \leq -1$
1	2	$-ao - a - bo + l \leq -1$			
0	3	$-ao - 3a - bo + l \leq -1$			
Condición para Estabilidad					$ao > 1$

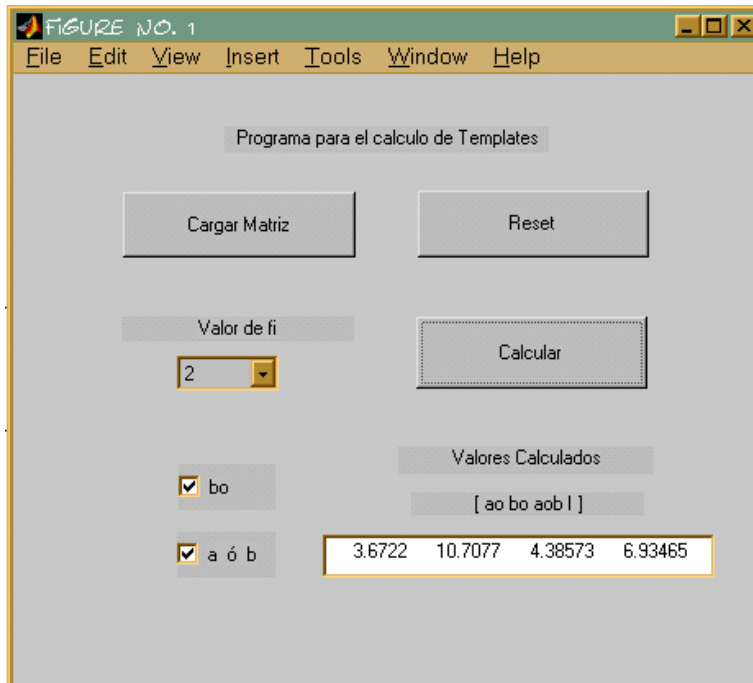


Figura 2.11 Resultados del programa en Matlab para la mascarilla del detector de conectividad global.



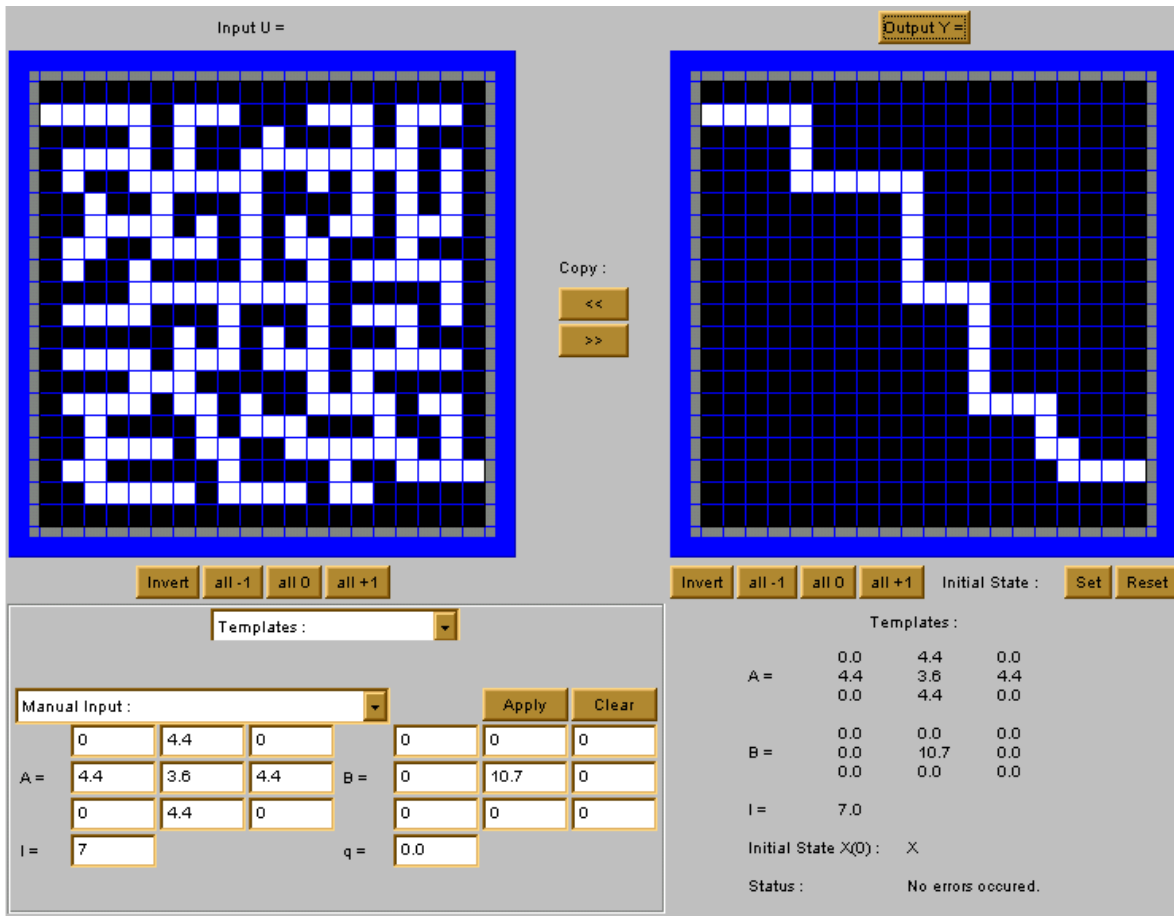


Figura 2.12 Resultados del simulador empleando las mascarillas obtenidas mediante el método Simplex.

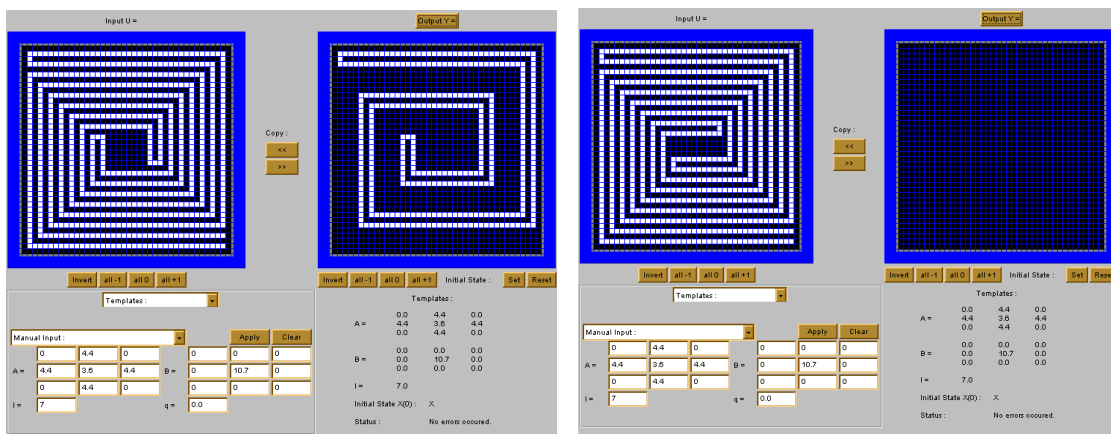


Figura 2.13 Otros patrones de entrada en el simulador para el detector de conectividad global.

### 2.2.2 Extractor de Bordes

La extracción de bordes es una de las tareas básicas que realizan sistemas sofisticados de procesamiento de imágenes, entre ellos los sistemas nerviosos biológicos como el cerebro humano. La importancia de esta operación morfológica radica en que permite la obtención de mucha información importante de los bordes de las imágenes, así como la compresión de grandes cantidades de información.

Para una imagen binaria, el planteamiento del problema puede quedar de la siguiente manera:

- 1) Buscar los parámetros de las mascarillas que lleven a cabo la extracción del borde de una silueta en negro.
- 2) Para este caso se requiere, que toda neurona que posea una entrada blanca su salida sea siempre blanca.
- 3) Para una neurona con entrada negra y la entrada de todas sus vecinas sea blanca, su salida deberá ser blanca.
- 4) Para todos los casos de entrada negra, con excepción de (3), la salida deberá ser negra.

Para este caso particular se proponen las mascarillas en la forma :

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & b & 0 \\ b & b_o & b \\ 0 & b & 0 \end{bmatrix}, \quad I$$

En donde, para que la condición (3) pueda ser llevada a cabo, no debe propagarse la salida de la información entre neuronas, de manera que los elementos de la matriz de retroalimentación  $A$  quedan descartados, con excepción de  $a_o$ .

Los diferentes casos de entrada para las neuronas y las salidas deseadas para la red se muestran en la tabla 2.3. La matriz correspondiente fue introducida en el programa de Matlab devolviendo los resultados mostrados en la figura 2.14. Una de las pruebas para estas mascarillas se muestra en la figura 2.15, donde se observa el desempeño deseado de la CNN.

Tabla 2.3 planteamiento para la obtención de las desigualdades para el extractor de bordes.

CASOS	Píxel de Entrada $V_{uij}$	Píxel de Salida $V_{yij}$	Neuronas del vecindario $C_{kl} \quad k \neq i, l \neq j$		Desigualdades
			No. de Negros	No. de Blancos	
(1)	Blanco (-1)	Blanco (-1)	4	0	$-ao + 4b - bo + l \leq -1$
			3	1	$-ao + 2b - bo + l \leq -1$
			2	2	$-ao - bo + l \leq -1$
			1	3	$-ao - 2b - bo + l \leq -1$
			0	4	$-ao - 4b - bo + l \leq -1$
			3	0	$-ao + 3b - bo + l \leq -1$
			2	1	$-ao + b - bo + l \leq -1$
			1	2	$-ao - b - bo + l \leq -1$
			0	3	$-ao - 3b - bo + l \leq -1$
(2)	Negro(1)	Negro(1)	3	1	$ao + 2b + bo + l \geq 1$
			2	2	$ao + bo + l \geq 1$
			1	3	$ao - 2b + bo + l \geq 1$
			0	4	$ao - 4b + bo + l \geq 1$
			3	0	$ao + 3b + bo + l \geq 1$
			2	1	$ao + b + bo + l \geq 1$
			1	2	$ao - b + bo + l \geq 1$
			0	3	$ao - 3b + bo + l \geq 1$
(3)	Negro (1)	Blanco(-1)	4	0	$-ao + 4b + bo + l \leq -1$
					$ao + 4b + bo + l < 1$
Condición para Estabilidad					$ao > 1$

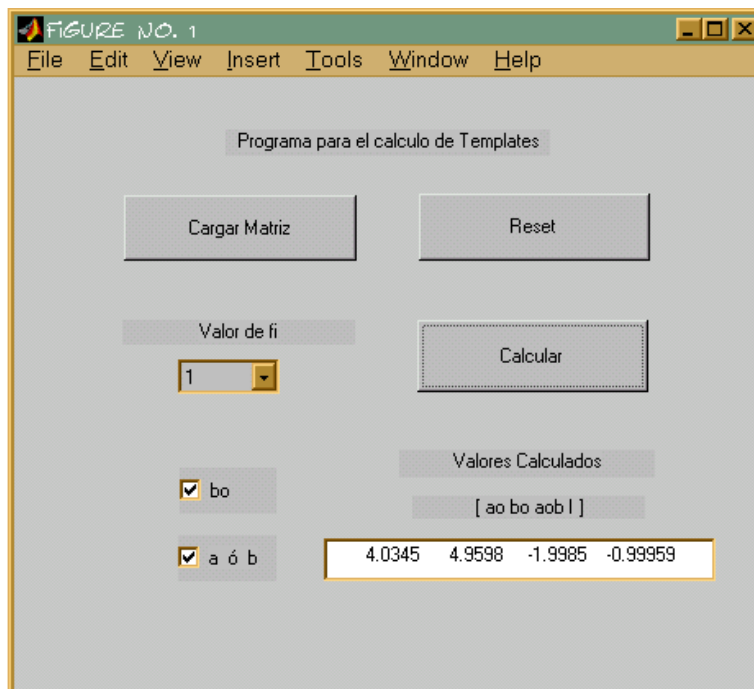


Figura 2.14 Resultados del programa en Matlab para las mascarillas del extractor de bordes.

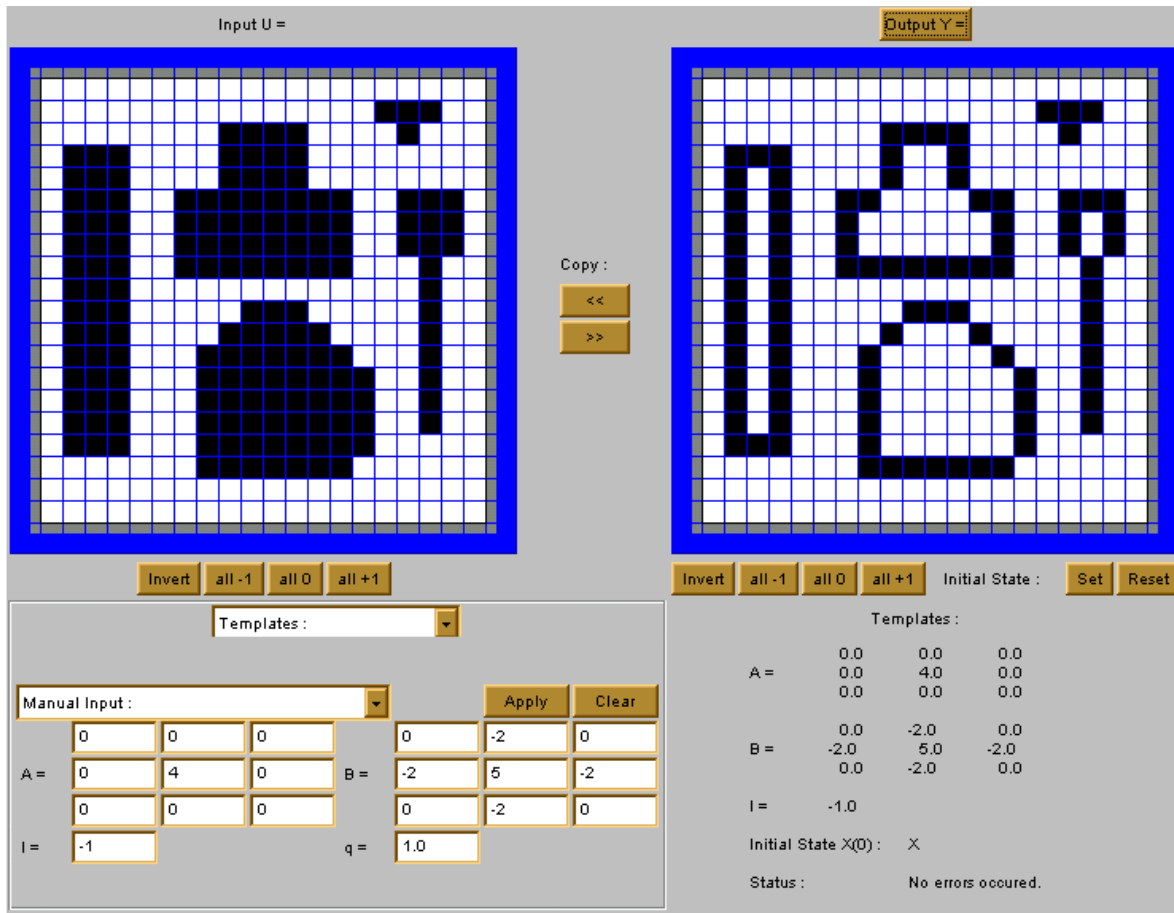


Figura 2.15 Resultados del simulador con las mascarillas obtenidas para el extractor de bordes.

### 2.2.3 Extractor de Sombras

La extracción de sombras en imágenes binarias consiste en hacer la proyección de los objetos negros de una imagen en alguna dirección. Esta operación permite extraer información de las imágenes que es utilizada en el reconocimiento de patrones [13].

El planteamiento del problema es el siguiente, para el caso de una proyección hacia la izquierda:

- 1) Cuando la entrada a una neurona es negra, su salida siempre tendrá que ser negra.
- 2) Cuando la entrada es blanca y la neurona vecina de la derecha tiene una salida blanca, la salida de la primera debe permanecer blanca.
- 3) Si la entrada es blanca y la neurona de la derecha posee salida negra, la salida de la primera tiene que ser negra.

Debido a que es un problema que requiere de propagación de información entre neuronas la mascarilla  $A$  es considerada en este caso, con lo cual se proponen las siguientes mascarillas:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_o & a \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, I$$

El planteamiento anterior en forma de casos se presenta en la tabla 2.4.

Tabla 2.4 Planteamiento para la obtención de las desigualdades para el extractor de sombras.

Casos	Píxel de Entrada	Píxel de Salida	Vecinos Negros	Vecinos Blancos	Desigualdad
(1)	Negro(1)	Negro(1)	1	0	$ao+a+bo+I \geq 1$
			0	1	$ao-a+bo+I \geq 1$
			0	0	$ao+bo+I \geq 1$
(2)	Blanco(-1)	Blanco(-1)	0	0	$-ao-bo+I \leq -1$
			0	1	$-ao-a-bo+I \leq -1$
(3)	Blanco(-1)	Negro(1)	1	0	$-ao+a-bo+I > -1$
					$ao+a-bo+I \geq 1$

De forma semejante a los problemas anteriores, al programa de Matlab sólo se introducen las columnas 2 a 5 de la tabla anterior. El resultado para las mascarillas se observa en la figura 2.16 y el resultado de una de las pruebas en el simulador se muestra en la figura 2.17 donde se observa el correcto desempeño de la tarea en la CNN.

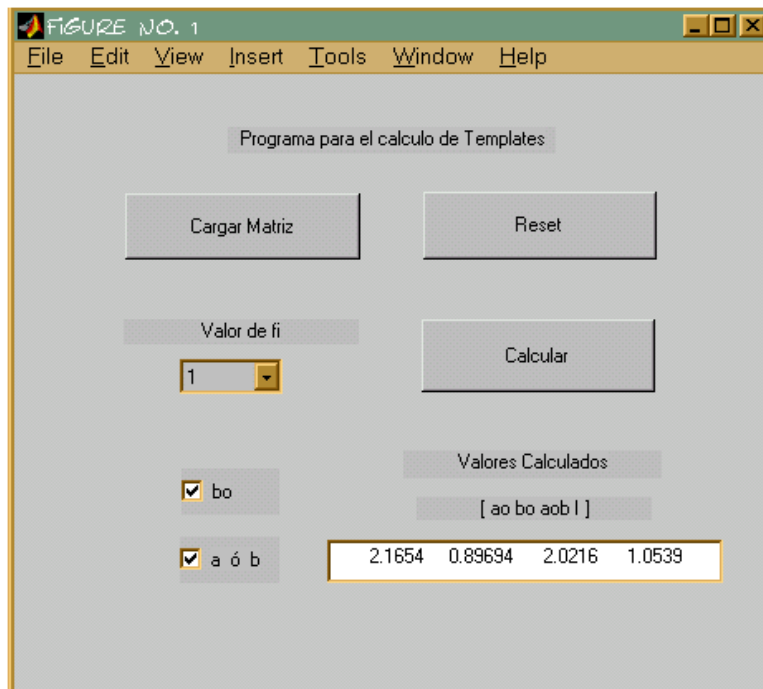


Figura 2.16 Resultados del programa en Matlab para las mascarillas del extractor de sombras.

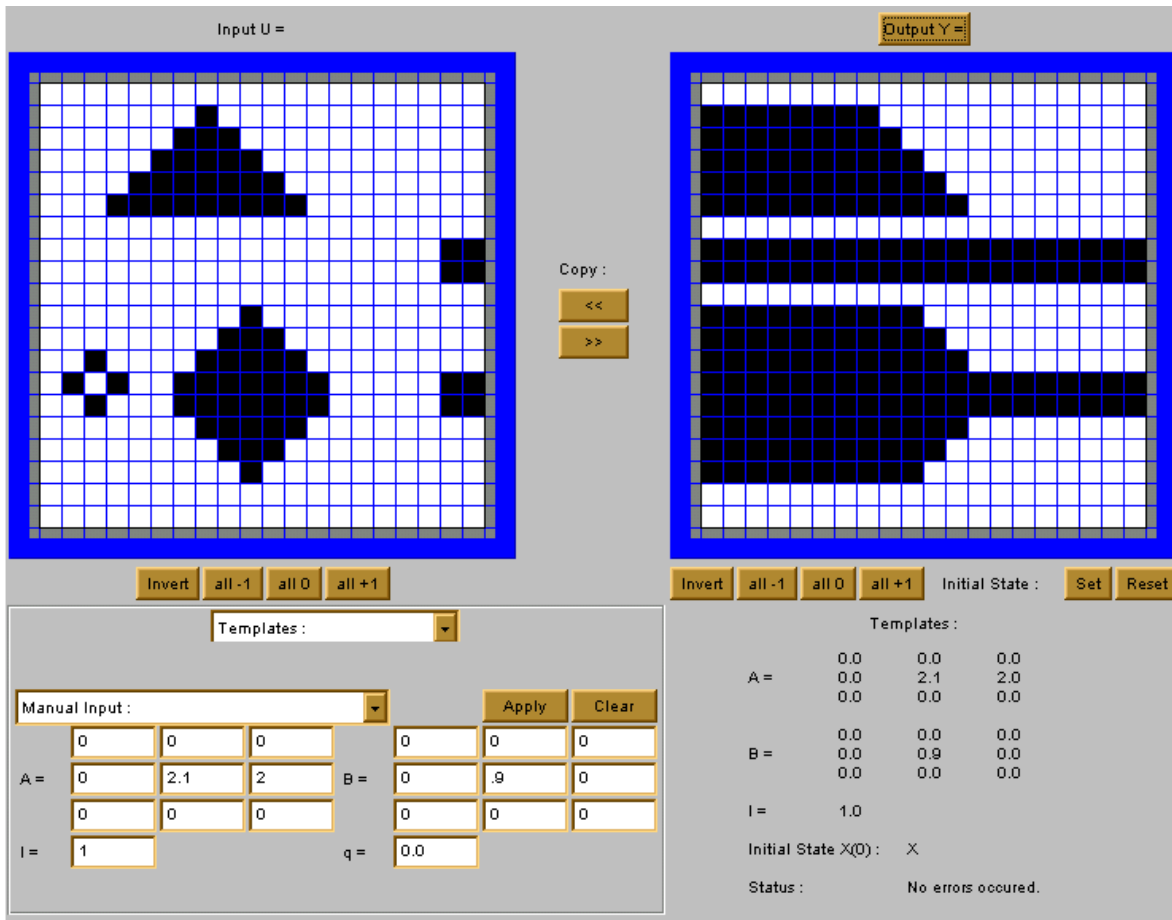


Figura 2.17 Resultados del simulador con las mascarillas obtenidas para el extractor de sombras.

### 2.2.4 Detector de Componentes Conectados.

El detector de componentes conectados (CCD) tiene la función de detectar el número de componentes que están conectados en un vector binario. Esta característica permite extraer información muy importante de las imágenes, lo cual ha permitido que sea empleada ampliamente para el reconocimiento de patrones y caracteres [14],[15].

El planteamiento del problema se puede hacer como sigue:

- 1) El problema se reduce solamente a procesamiento unidimensional, esto debido a que cualquier imagen binaria puede considerarse compuesta de un conjunto de vectores binarios.
- 2) Cuando la entrada a una neurona  $C(i,j)$  sea blanca, la vecina de la izquierda posea salida negra y la de la derecha blanca, la salida de  $C(i,j)$  deberá cambiar a negra.
- 3) Si la entrada a una neurona  $C(i,j)$  es negra, la salida de su vecina de la izquierda es blanca y la de la derecha es negra, la salida de  $C(i,j)$  deberá ser blanca.

- 4) Para cualquier otra combinación posible con excepción de (2) y (3), la salida de la neurona C(i,j) deberá corresponder a la de su entrada.

Dada la naturaleza de este tipo de problema, es necesario el empleo de mascarillas no simétricas, en este caso en la matriz de retroalimentación  $A$ , obedeciendo a la propagación de los componentes conectados hacia una dirección específica.

Las mascarillas que se proponen son :

$$A = \begin{bmatrix} 0 & 0 & 0 \\ a_1 & a_0 & a_2 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, I$$

El conjunto de casos de entrada y salida posibles, considerando los estados y las salidas de la neuronas de frontera como (-1) se muestran en la tabla 2.5. Como en este problema es necesaria la utilización de mascarillas no simétricas, el programa con la interfaz gráfica-usuario que calcula las ecuaciones automáticamente no puede ser utilizado ya que está limitado para emplearse sólo para mascarillas simétricas. Por lo tanto, el problema fue planteado de manera manual definiendo la función objetivo y utilizando el método Simplex, con lo cual se obtuvo el siguiente resultado:

$$\begin{aligned} a_0 &= 3.30 \\ a_1 &= 2.00 \\ a_2 &= -3.24 \\ I &= 0.05 \end{aligned}$$

Utilizando los valores de la mascarilla anterior, se procedió a hacer pruebas en el simulador, uno de los resultados de las pruebas se muestra en la figura 2.18, donde se puede observar el desempeño correcto.

Tabla 2.5 Planteamiento para la obtención de las desigualdades para el detector de componentes conectados.

Casos	Píxel de Entrada	Píxel de Salida	Vecinos Negros	Vecinos Blancos	Desigualdad
(1)	Blanco(1)	Negro(1)	$a_1$	$a_2$	$ao+a_1-a_2+I \geq 1$
					$-ao+a_1-a_2+I > -1$
(2)	Negro(1)	Negro(1)	$a_1$ y $a_2$	0	$ao+a_1+a_2+I \geq 1$
			0	$a_1$ y $a_2$	$ao-a_1-a_2+I \geq 1$
(3)	Blanco(-1)	Blanco(-1)	$a_1$ y $a_2$	0	$-ao+a_1+a_2+I \leq -1$
			0	$a_1$ y $a_2$	$-ao-a_1-a_2+I \leq -1$
(4)	Negro(1)	Blanco(-1)	$a_2$	$a_1$	$-ao+a_2-a_1+I \leq -1$
					$ao+a_2-a_1+I < 1$

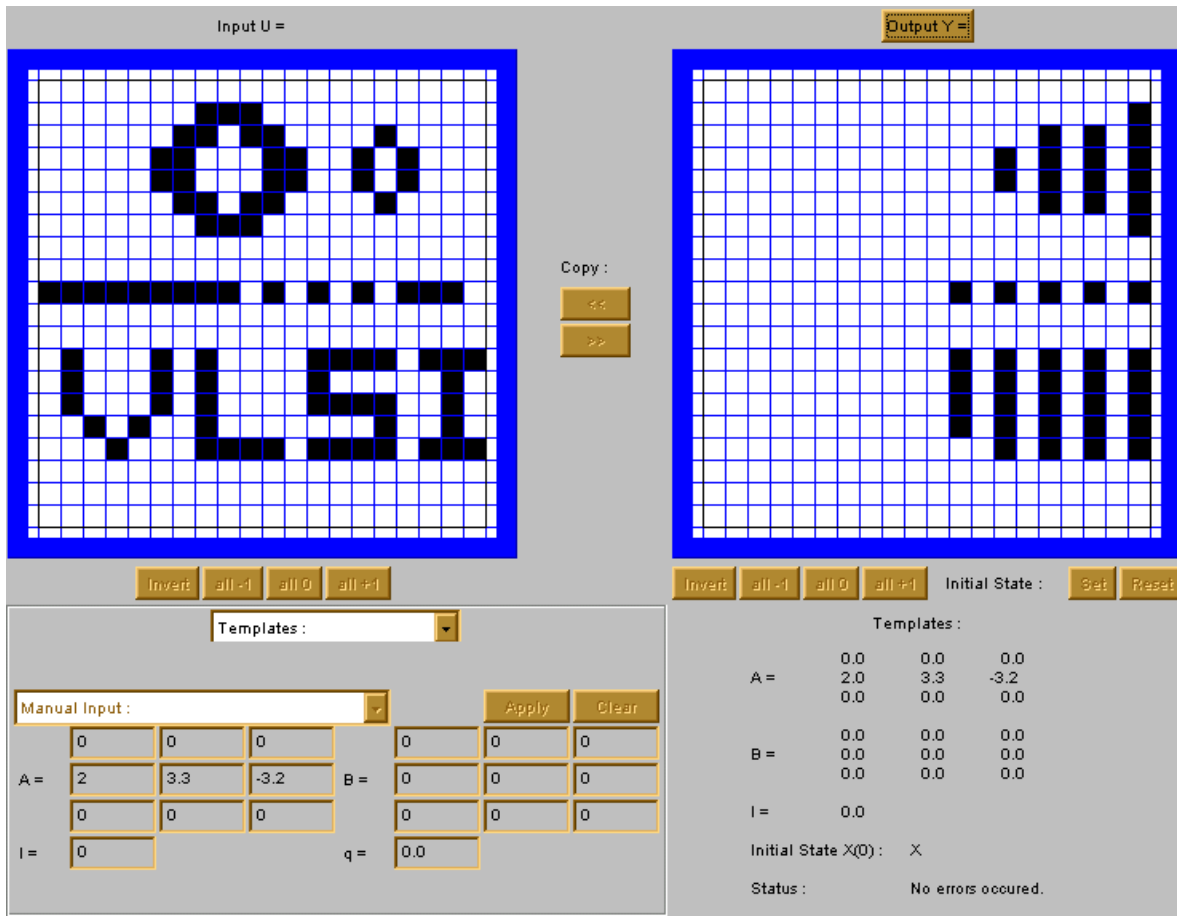


Figura 2.18 Resultado de la CNN en el simulador utilizando la mascarilla obtenida para el CCD.



---

## Referencias

- [1] L. O. Chua, P. Thiran, “ An Analytic Method for Designing Simple Cellular Neural Networks” IEEE Trans. on Circuits and Systems, vol. 38, no. 11, November 1991.
- [2] I. Fajfar and F. Bratkovic, “ Design of Monotonic Binary-Valued Cellular Neural Networks” CNNA’96, Fourth IEEE International Workshop on Cellular Neural Networks and their Applications. Seville, 1996.
- [3] J. A. Nossek, “ Design and Learning with Cellular Neural Networks” Proc. IEEE International Workshop on Cellular Neural Networks and their Applications. Rome, 1994 pp. 137-146.
- [4] C. Guzelis and S. Karamahut, “Recurrent Perceptron Learning Algorithm for Completely Stable Cellular Neural Networks” Proc. IEEE International Workshop on Cellular Neural Networks and their Applications. Rome, 1994 pp. 177-182.
- [5] B. Mirzai, Z. Cheng, and G. S. Moschytz, “Learning Algorithms for Cellular Neural Networks” Proc. IEEE Int. Symp. Circuits Systems, Monterey CA. June 1998, vol. 3, pp. 159-162.
- [6] T. Kozek, T. Roska and L.O. Chua, “Genetic Algorithm for CNN Template Learning “ IEEE Trans. On Circuits and Systems I, vol.40, pp. 392-402, June 1993.
- [7] K. Nakai and A. Ushida, “Design Technique on Cellular Neural Network” Electronics and Communications in Japan, Part 3, vol. 78, no. 3, 1995.
- [8] Leon O. Chua and L. Yang, “Cellular Neural Networks: Theory” IEEE Trans. on Circuits and Systems, vol. 35, pp 1257-1272, Oct. 1988.
- [9] F. Zou, S. Schwarz and J.A. Nossek, "Cellular Neural Network Design Using a Learning Algorithm", Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Their Applications, (CNNA'90), pp.73-81, Budapest, 1990.
- [10] S. C. Chapra, “Métodos Numéricos para Ingenieros” 3ra edición McGraw Hill 1999.
- [11] J. R. Jang, C. Sun, E. Mizutani, “Neuro-Fuzzy and Soft Computing: A computational approach to learning and machine intelligence” Prentice Hall 1997.
- [12] Leon O. Chua, “CNN: A Paradigm for Complexity” Series A, vol. 31 Ed. World Scientific Publishing 1998.
- [13] T. Matsumoto, L.O. Chua and H. Suzuki, “CNN Cloning Template: Shadow Detector” IEEE Trans. on Circuits and Systems, vol. 37, no. 8, pp. 1070-1073, August 1990.
- [14] T. Matsumoto, L.O. Chua and H. Suzuki, “CNN Cloning Template: Connected Component Detector” IEEE Trans. on Circuits and Systems, vol. 37, no. 5, pp. 633-635, May 1990.
- [15] T. Roska, Angel Rodriguez-Vazquez, “Toward the Visual Microprocessor” pp. 344-350 John Wiley Sons 2001.



# CAPÍTULO 3



MODELO DE CNN CON  
FUNCIÓN DE SALIDA ESCALÓN  
CON DESPLAZAMIENTO

### 3.0 Introducción

Hasta este momento, el trabajo se ha enfocado en el desarrollo de un método para el diseño de los parámetros de las mascarillas basándonos en la idea original propuesta por L. O. Chua y L. Yang.

Varias observaciones al trabajo de K. Nakai *et al.* [1], que se mencionan a continuación, sugieren la posibilidad de realizar algunas modificaciones al modelo de Chua y Yang que permiten simplificar los recursos en hardware para la implementación de la CNN.

#### 3.1 Red Neuronal Celular con Función de Salida Escalón.

Si se observan todas las tablas, esto es: 2.1, 2.2, 2.3, 2.4 y 2.5 del capítulo anterior, puede hacerse notar que todas las desigualdades para imágenes binarias, tienen la forma ( $>-1$ ), ( $<1$ ), ( $\leq -1$ ) y ( $\geq 1$ ). Matemáticamente, es posible llevar a todas las desigualdades a sólo dos formas posibles ( $\leq -1$ ) y ( $\geq 1$ ) si consideramos lo siguiente:

Para el caso particular en que se desea una salida  $v_{yij} = 1$  con una condición inicial  $v_{xij}(0) = -1$ , es necesario que la ecuación de estado en la región lineal cumpla  $dv_{xij}/dt > 0$  como lo establecido en (C) del capítulo anterior, de manera que la ecuación (2.2) queda de la siguiente forma:

$$0 < -v_{xij}(0) + a_0 v_{xij}(0) + k_I$$

que sustituyendo la condición inicial, obtenemos:

$$-1 < -a_0 + k_I \tag{3.0}$$

para lo cual despejando  $k_I$  se debe cumplir:

$$k_I > a_0 - 1 \tag{3.1}$$

Ahora modificando la expresión (3.0) a la forma ( $\geq 1$ ), implica:

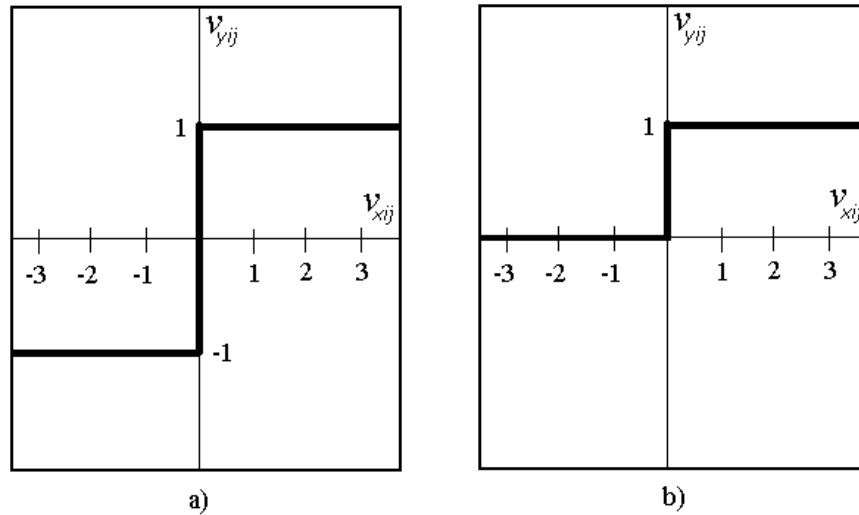
$$1 \leq -a_0 + k_I, \text{ lo cual para } k_I \text{ tenemos}$$

$$k_I \geq a_0 + 1 \tag{3.2}$$

lo cual debe cumplirse.

Si la desigualdad (3.2) se satisface, resulta evidente que (3.1) también se cumple, de manera que el modificar la desigualdad de la forma ( $> -1$ ) a ( $\geq 1$ ) resulta solamente en reducir la frontera solución de aquella con la forma ( $> -1$ ) en dos unidades. Lo mismo se puede considerar para el caso cuando se desea una salida  $v_{yij} = -1$  y la condición inicial es  $v_{xij}(0) = 1$ .

El hecho de que para todos los casos que deban cumplir una salida negra, las desigualdades pueden llevarse a la forma  $(\geq 1)$  y las de salida blanca a  $(\leq -1)$ , cualquier solución matemática a un sistema de este tipo, sugiere que las salidas blancas y negras son *linealmente separables* y es posible la existencia de un hiperplano que separe a los dos tipos de salidas. Entonces, la función de salida podría modificarse a una función de tipo escalón como las que se muestran en la figura 3.0 a).



**Figura 3.0** Dos tipos diferentes de función de salida escalón que podrían ser utilizadas en una CNN.

La función de la figura 3.0 b) también podría ser utilizada cambiando la convención de las salidas, ahora interpretando un 0 para píxeles blancos y +1 para negros. Este tipo de funciones traen beneficios directos en hardware ya que son funciones cuya implementación requiere de un número menor de transistores comparado con implementaciones de aquellas otras aproximaciones para la función de salida UG.

### 3.2 Eliminación del Elemento de Retroalimentación $a_0$

Cuando el sistema de desigualdades, es llevado a la forma  $\leq -1$  y  $\geq 1$  considerando una función de salida escalón para el sistema tal como el de la figura 3.0 (a), y tomando en cuenta los casos particulares que involucran transición en el valor del píxel, como es el caso (4) en negrillas de la tabla 2.1, las nuevas desigualdades quedarían de la siguiente forma:

<b>(4)</b>	Blanco(-1)	Negro (+1)	4	0	$ao + 4a - bo + l \geq 1$ A)
					$- ao + 4a - bo + l \geq 1$ C)

Donde la única diferencia entre la desigualdad (A) y (C) es el elemento central de la mascarilla de retroalimentación,  $a_0$ . Como puede verse, las dos desigualdades establecen que, independientemente de la propia salida de la neurona  $C(i,j)$ , las desigualdades tienen que satisfacer  $\geq 1$ , por lo que el elemento  $a_0$  para todos los casos del removedor de ruido que consideran una transición podría omitirse. Tal omisión puede resultar obvia desde el punto de vista de que al desaparecer la región lineal de la función de salida UG ya no es necesaria la retroalimentación positiva realizada por  $a_0$ , la cual garantiza que la variable de estado salga de dicha región. Esta observación puede llevarse a todos los demás casos planteados (aquellos que sólo consideran la desigualdad para el estado estable) de un problema particular de procesamiento si observamos el siguiente ejemplo de un sistema de desigualdades lineales:

$$\begin{aligned} x + y &> 10 \\ x + y &< 4 \end{aligned} \tag{3.3}$$

Puede observarse a simple vista que no hay existencia de intersección entre las regiones solución para ambas desigualdades, con lo cual se concluye que el sistema no tiene solución.

Volviendo a la tabla 2.1 y tomando en cuenta la eliminación del elemento  $a_0$  en todas las desigualdades, puede observarse que ningún par de ellas puede expresarse en una forma análoga a la antes planteada en (3.3). Por lo que tal consideración nos lleva a que para el nuevo sistema no existe la garantía de no solución.

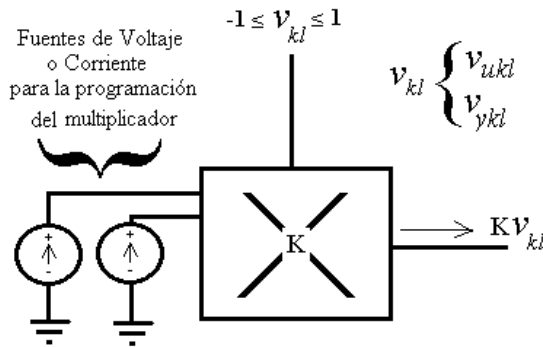
Los sistemas de desigualdades de las tablas 2.1, 2.2, 2.3 y 2.4 con las desigualdades en la forma  $>0$  y  $<0$  y la omisión de  $a_0$  fueron resueltos satisfactoriamente (para  $a$ ,  $bo$  e  $I$  respectivamente) mediante el método simplex (apéndice A), con lo cual se puede concluir que son problemas *linealmente separables* y una función de tipo escalón puede ser utilizada como función de salida; además, es posible omitir en los cuatro primeros casos el elemento  $a_0$  de la matriz de retroalimentación. Sin embargo, esta conclusión no puede ser hecha para el caso particular del detector de componentes conectados (CCD) ya que, la separabilidad lineal puede ser demostrada pero no la omisión de  $a_0$  (por la existencia de dos casos análogos a (3.3) que se muestran en la tabla 3.1 mediante \* y \*\*) y por lo tanto, es un caso en donde la salida propia de la neurona  $C(i,j)$  sí interviene en la dinámica de la neurona.

Tabla 3.1

Casos	Píxel de Entrada	Píxel de Salida	Vecinos Negros	Vecinos Blancos	Desigualdad
(1)	Blanco(1)	Negro(1)	$a_1$	$a_2$	
(2)	Negro(1)	Negro(1)	$a_1$ y $a_2$	0	$ao+a_1+a_2+I \geq 1^*$
			0	$a_1$ y $a_2$	$ao-a_1-a_2+I \geq 1^{**}$
(3)	Blanco(-1)	Blanco(-1)	$a_1$ y $a_2$	0	$-ao+a_1+a_2+I \leq -1^*$
			0	$a_1$ y $a_2$	$-ao-a_1-a_2+I \leq -1^{**}$
(4)	Negro(1)	Blanco(-1)	$a_2$	$a_1$	

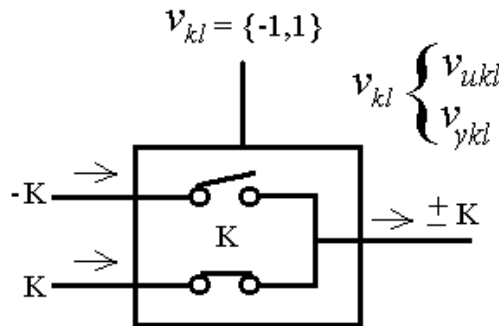
### 3.3 Desplazamiento de la Función Escalón al Semiplano Positivo

En el contexto de los circuitos integrados con arquitectura de CNN programable con función de salida UG, es necesaria la implementación de circuitos multiplicadores que permitan llevar a cabo las operaciones de las mascarillas, esto es, la realización de las multiplicaciones de la matrices  $A$  y  $B$  por las salidas y las entradas de la neuronas correspondientes tanto para la región lineal como para la de saturación en la función de salida UG. En la figura 3.1 se muestra el bloque de un multiplicador comúnmente utilizado para la implementación de una CNN. En los trabajos [2],[3], se observa la utilización de circuitos multiplicadores que emplean la celda de Gilbert.



**Figura 3.1** Bloque representativo de un multiplicador.

En nuestro modelo propuesto de CNN, en donde se considera la utilización de una función de salida tipo escalón, debido a la ausencia de la región lineal, la multiplicación puede ser cambiada a una función de selección. Tomando en cuenta que sólo existirán únicamente dos valores posibles para las salidas  $v_{yij}$  y las entradas  $v_{uij}$ , el multiplicador que originalmente es empleado en la CNN con función de salida UG, puede ser sustituido por un multiplexor analógico como el que se ilustra en la figura 3.2.



**Figura 3.2** Bloque representativo de un multiplexor.

Este multiplexor analógico ofrece el mismo resultado que si se utilizara un multiplicador solamente para  $v_{yij} = \{-1,1\}$ , pero con el beneficio directo de una reducción considerable en el número de transistores, lo cual beneficia directamente en la implementación de arreglos neuronales con mayor velocidad y mayor densidad de integración. Esta afirmación también es hecha en [4].

Partiendo de la figura 3.2, donde dicho bloque se considera para la realización de los coeficientes de las mascarillas, en una implementación programable donde la variable de estado es considerada en modo de corriente (como es más comúnmente visto en la mayoría de trabajos), la programación radica en la posibilidad de poder cambiar las corrientes de entrada  $-K$  y  $K$  en diferentes magnitudes y sentidos para la ejecución de diferentes tareas. Para poder llevar a cabo esto, es necesario el empleo de copiadore de corriente bilaterales (figura 3.3 ) como los que se mencionan en [5] o fuentes de corriente ajustables como las utilizadas en [6], que suministren la corriente deseada al multiplexor analógico.



Figura 3.3 Bloque de un copiador bilateral.

Si circuitos de este tipo tienen que ser implementados por cada coeficiente de las mascarillas de cada neurona, resulta en una gran cantidad de transistores que tienen que ser empleados para tal propósito, de tal manera que sería deseable encontrar un modelo que permitiera que todas las variables fueran siempre positivas, lo cual conduciría a que la implementación física se reduzca a contemplar sólo circuitos de una sola polaridad, como podrían ser copiadore de corriente como los que se muestran en la figura 3.4. Tal situación reduciría en forma aún mayor el número de transistores empleados.



Figura 3.4 Bloque de un copiador unilateral.



La idea de la cual se parte para obtener una solución matemática que considere todos los valores de los elementos de las mascarillas de la CNN positivos, sería primeramente modificar la función de salida escalón mediante un desplazamiento al semiplano positivo, como se muestra en la figura 3.5. De manera que se da lugar a que valores positivos de  $v_{xij}$  menores a  $X$  permitan una salida en bajo correspondiendo a un blanco y que valores mayores a  $X$  correspondan a un negro.

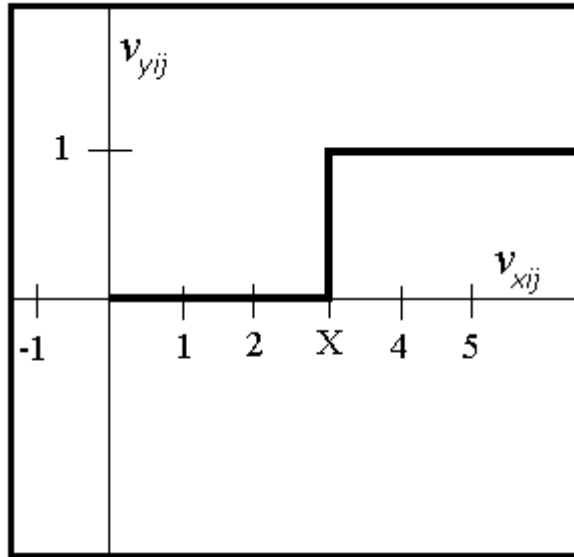


Figura 3.5 Función de salida escalón desplazada al semiplano positivo.

Es necesario considerar todos los coeficientes de las mascarillas como positivos, esto es, no se debe seguir la convención de  $-1$  para blanco y  $1$  para negro, sino considerar unidades diferentes para píxeles blancos y negros de cada mascarilla. Para hacer posible la solución matemática, el elemento  $I$  definido como umbral en el modelo original tiene que ser omitido y una nueva variable  $f$  es introducida; esta variable sólo es considerada para aquellas neuronas de frontera. El siguiente ejemplo ilustra esta idea para el caso de una neurona de frontera incluida en la tabla 2.1 para el removedor de ruido. El planteamiento original establece:

(2)	Negro (+1)	Negro (+1)	2	1	$a_0 + a + b_0 + I \geq 1$ (A)
-----	------------	------------	---	---	--------------------------------

Aplicando las sustituciones propuestas se llega a:

(2)	Negro (+1)	Negro (+1)	2	1	$n_{OB} + 2n_A + b_A + f \geq X$
-----	------------	------------	---	---	----------------------------------

Donde para el elemento central de la mascarilla  $B$ , que antes se consideraba  $+b_0$  para el caso de un píxel negro y  $-b_0$  para uno blanco, ahora se sustituye por las variables  $n_{OB}$  para uno negro y  $b_{OB}$  para uno blanco. De manera análoga se hace para los elementos no centrales de las mascarillas, donde para la mascarilla  $A$  se consideran las variables  $n_A$  para el negro y  $b_A$  para el blanco, así como para la mascarilla  $B$ , se consideran  $n_B$  y  $b_B$  respectivamente. El elemento  $X$ , denota el punto del semiplano positivo donde se realiza la transición de la función escalón, y es establecido mediante heurística.

Este nuevo planteamiento, con la consideración de la omisión del elemento  $a_0$  antes mencionada y con las consideraciones del método simplex, ha encontrado solución matemática satisfactoria para cuatro de las tareas de procesamiento antes señaladas que son: removedor de ruido, extractor de bordes, extractor de sombras y detector de conectividad global, las tablas y los resultados pueden observarse en el apéndice B.

Para el caso del detector de componentes conectados (CCD), como se mencionó en la sección anterior, el elemento  $a_0$  es necesario para el desarrollo de la tarea pero no  $b_0$ , la solución propuesta para este problema radica en sustituir al elemento  $+a_0$  por  $n_{OA}$  y  $-a_0$  por  $b_{OA}$  de manera análoga a  $n_{OB}$  y  $b_{OB}$  definidos anteriormente, pero la diferencia fundamental es que ahora éstas no dependen de la entrada sino dependen, como ya se definió para la mascarilla  $A$ , de la salida de la neurona. Las tablas de los casos correspondientes y su solución mediante el método simplex se muestra también en el apéndice B.

### 3.4 Ecuaciones del Modelo Propuesto de CNN

Tomando en cuenta los resultados numéricos de la sección anterior, es posible proponer una ecuación de estado que satisfaga las cinco tareas de procesamiento antes mencionadas en un nuevo modelo para el cual es posible contemplar sólo una mascarilla, ya sea la de control o la de retroalimentación para sus elementos no centrales y  $a_0$  o  $b_0$ .

Puesto que ahora sólo existirán dos valores posibles para las entradas  $v_{uij}$  y las salidas  $v_{yij}$  correspondiendo a los valores lógicos  $\{0,1\}$ , la **ecuación de estado** para el modelo viene dado en la forma:

$$\frac{dv_{xij}}{dt} = -v_{xij} + k_3 \quad (3.4)$$

donde  $k_3$  es definido como

$$k_3 = b_{Ox_1} \bar{v}_{ij} + n_{Ox_1} v_{ij} + \sum_{C(k,l) \in Nr(i,j)/C_{ij}} (b_{x_2}(k,l) \bar{v}_{kl} + n_{x_2}(k,l) v_{kl}) + f \quad (3.5)$$

con:

$$b_{Ox_1}, n_{Ox_1}, b_{x_2}(k,l), n_{x_2}(k,l), f \geq 0$$

Donde el subíndice  $x_1$  puede ser  $A$  si se trata de un elemento central de la mascarilla de retroalimentación, o  $B$  para un elemento central de las mascarilla de control. De igual forma el subíndice  $x_2$  puede considerarse como  $A$  si se trata de elementos no centrales de la mascarilla de retroalimentación, o  $B$  para el caso de la de control.

La señal  $v_{ij}$  estará representada por la señal de salida  $v_{yij}$  para el caso de que  $x_1$  corresponda al elemento central de las mascarilla  $A$  o estará representada por la señal de entrada  $v_{uij}$  para el caso de que  $x_1$  corresponda a la mascarilla  $B$ . De forma similar, la variable  $v_{kl}$  representará a la salida de la demás neuronas  $v_{ykl}$  en el caso de que  $x_2$  corresponda a la mascarilla  $A$  o a  $v_{ukl}$  en el caso de la mascarilla  $B$ .

Y los elementos  $\bar{v}_{ij}$  y  $\bar{v}_{kl}$  representan los complementos lógicos de dichas variables respectivamente.

La solución de la ecuación (3.4) esta dada de igual manera a la ecuación (2.5), esto es, en la forma:

$$v_{xij} = k_3 + Ce^{-t/h} \quad (3.6)$$

donde  $h$  esta dada por el producto RC asociado a la entrada del subcircuito destinado a la implementación de la suma de los elementos de  $k_3$ .

Y, finalmente, la **función de salida** que como ya se comentó anteriormente es la función escalón desplazada al semiplano positivo, la cual está dada en la forma:

$$v_{yij} = F(v_{xij}) = \begin{cases} 0 & v_{xij} < X \\ 1 & v_{xij} \geq X \end{cases} \quad (3.7)$$

para  $v_{xij}$  y  $X$  siempre positivos.

Y dada la característica de la ecuación (3.6) de que  $v_{xij}$  siempre tenderá hacia  $k_3$  para valores grandes de  $t$  independientemente de su condición inicial  $v_{xij}(0)$ , esto hace que las condiciones iniciales consideradas en (2.1) puedan ser modificadas a la forma:

$$v_{ykl} = \begin{cases} v_{uij} & k = i, l = j \\ v_{ykl}(\infty) & k \neq i, l \neq j \end{cases} \quad (3.8)$$


En el siguiente capítulo se da inicio a las consideraciones eléctricas en la implementación de este modelo.

### Referencias

- [1] K. Nakai and A. Ushida, "Design Technique on Cellular Neural Network" *Electronics and Communications in Japan, Part 3*, vol. 78, no. 3, 1995.
- [2] M. Anguita, F. J. Pelayo, A. Prieto and Julio Ortega, "Analog CMOS Implementation of a Discrete Time CNN with Programmable Cloning Templates" *IEEE Trans. on Circuits and Systems II*, vol. 40, no. 3, March 1993.
- [3] G. F. Dalla Betta, S. Graffi, Zs. M. Kovacs and G. Masetti, "CMOS Implementation of an Analogically Programmable Cellular Neural Network" *IEEE Trans. on Circuits and Systems II*, vol. 40, no. 3, March 1993.
- [4] A. Paasio and K. Halonen, "A New Cell Output Nonlinearity for Dense Cellular Nonlinear Network Integration" *IEEE Trans. On Circuits and Systems I*, vol. 48, no. 3 March 2001.
- [5] Angel Rodriguez-Vazquez, Servando Espejo, Rafael Dominguez-Castro, J.L. Huertas and E. Sánchez-Sinencio, "Current Mode Techniques for the Implementation of Continuous and Discrete-Time Cellular Neural Networks" *IEEE Trans on Circuits and Systems II*, vol. 40, no. 3, March 1993.
- [6] A. Paasio, A. Dawidziuk, V. Porra, "Compact VLSI implementation of Cellular Neural Network" *The Midwest Symposium on Circuits and Systems*, Ames, 1996



# CAPÍTULO 4



IMPLEMENTACIÓN  
ELÉCTRICA DE LA RED  
NEURONAL CELULAR

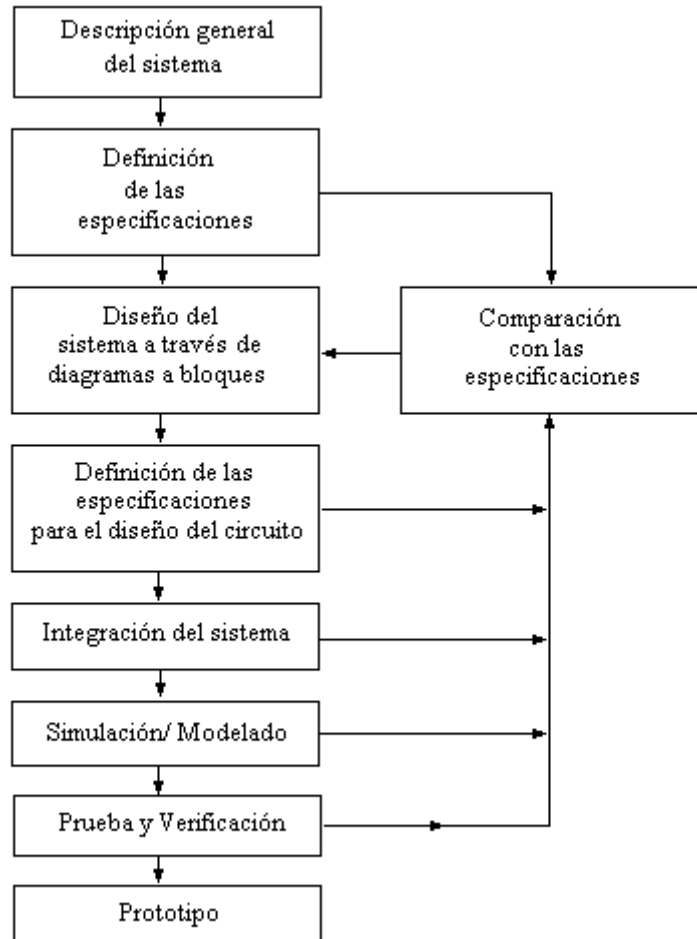
## 4.0 Introducción

La forma en que se procesa una señal en un sistema electrónico depende de la naturaleza de las señales de entrada, de las especificaciones en los parámetros de salida y de los requerimientos globales de funcionamiento. Actualmente el uso de técnicas de diseño de circuitos integrados con la capacidad de combinar funciones tanto analógicas como digitales, ha permitido un gran avance en el desarrollo de sistemas electrónicos más complejos. En este sentido, esta tendencia ha venido a exigir actualmente a los diseñadores de circuitos integrados el conocimiento en ambos dominios de la electrónica: lo analógico y lo digital, de tal manera que muchos de los sistemas electrónicos hoy en día pueden ser implementados sobre un mismo sustrato de silicio y disponer de las grandes ventajas que ofrece la integración, tales como la reducción de costos en producción masiva, fiabilidad, área reducida, mayor velocidad de operación y bajo consumo de energía.

A medida que los sistemas en ingeniería aumentan en complejidad, es necesario desarrollar las metodologías para poder analizar, sintetizar y diseñar tales sistemas. Un diseño transforma las especificaciones derivadas de los planteamientos en circuitos que las llevan a cabo. Antes de comenzar con los aspectos del diseño electrónico del modelo, consideramos oportuno citar una definición precisa acerca de la idea general del diseño. El Comité de Acreditación para la Ingeniería y Tecnología (ABET) establece una definición al respecto:

*“El diseño de ingeniería es el proceso mediante el cual se crea un sistema, componente o proceso para satisfacer ciertas necesidades. Es un proceso de toma de decisiones en el que se aplican las ciencias básicas, las matemáticas y las ciencias de la ingeniería para convertir los recursos de manera óptima y cumplir un objetivo establecido. Entre los elementos fundamentales en el proceso del diseño se encuentran el establecimiento de los objetivos y los criterios, la síntesis, el análisis, la construcción, la prueba y la evaluación. El diseño en ingeniería debe incluir la mayor parte de las siguientes características: desarrollar la creatividad del estudiante, utilizar problemas abiertos, desarrollar y utilizar la teoría y la metodología del diseño, formular las bases y especificaciones del problema a resolver, considerar soluciones alternativas, considerar la factibilidad, procesos de producción, diseño concurrente de ingeniería y descripciones detalladas del sistema. Además, es esencial incluir una variedad de restricciones reales como son los factores económicos, la seguridad, la confiabilidad, la estética, la ética y el impacto social”.*

Específicamente, en el área de diseño de sistemas VLSI, la elaboración de un sistema electrónico puede ser orientado mediante un proceso de pasos jerárquicos en el que los sistemas se diseñan primero con la ayuda de diagramas funcionales de bloques y posteriormente se prosigue con el diseño de los circuitos y los dispositivos. En la figura 4.0 se muestra una metodología para la implementación de un prototipo electrónico y aunque algunos pasos y su orden pueden ser alterados u omitidos, resulta útil disponer de una línea que oriente la labor del diseñador. A lo largo de este trabajo de tesis, se ha pretendido mostrar de una manera global los aspectos generales necesarios en la elaboración de un prototipo que permita la ejecución de determinadas tareas de procesamiento en imágenes dentro del marco de una arquitectura definida como red neuronal celular.



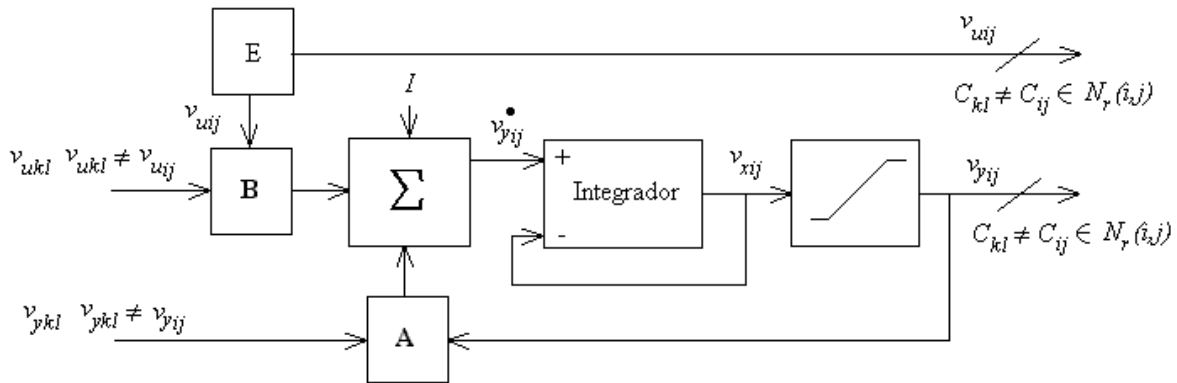
**Figura 4.0** Diagrama a bloques para la elaboración de un sistema electrónico.



### 4.1 Diagrama a Bloques de una Neurona CNN

La implementación física de una CNN consiste en llevar a cabo a través de circuitos eléctricos cada una de las partes que constituyen las ecuaciones dinámicas y conexiones que determinan el comportamiento de la red.

Estos circuitos pueden ser deducidos a partir de las ecuaciones de estado y de salida, así como del vecindario definido para una neurona. En base al modelo propuesto por Chua y Yang, considerando parámetros invariantes, se puede construir el diagrama a bloques que se muestra en la figura 4.1



**Figura 4.1** Diagrama a bloques de una neurona CNN basada en el modelo Chua-Yang.

La neurona en este diagrama está constituida por diferentes bloques que son:

- A y B, que se refieren a las matrices de retroalimentación y control. En las implementaciones programables, estas etapas se llevan a cabo mediante el uso de circuitos multiplicadores.
- Un sumador para las señales de las mascarillas y el umbral  $I$ .
- Un bloque integrador para determinar la variable de estado  $v_{xij}$ .
- Un bloque para la función de salida UG.
- Y el bloque E, que define la entrada de la neurona, la cual puede provenir, ya sea de un fotodetector o de algún dispositivo de almacenamiento.

Para el modelo que se propone en este trabajo, siguiendo la establecido en las ecuaciones (3.5) y (3.7), se obtiene el diagrama a bloques de la figura 4.2.

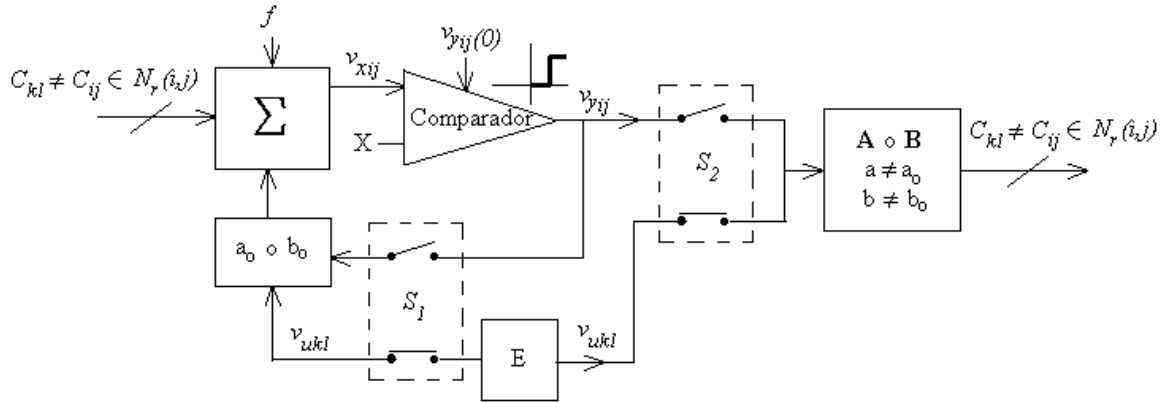


Figura 4.2 Diagrama a bloques de una neurona del modelo propuesto.

De esta representación gráfica se obtiene:

- Un bloque sumador, en donde se acumulan todas las señales originadas por las mascarillas.
- Un comparador que permite la implementación de la función escalón desplazada, donde se debe cumplir la condición  $X > 0$ .
- Un bloque para la ponderación de las señales  $v_{yij}$  o  $v_{uij}$ , como elementos centrales de las mascarillas,  $a_0$  o  $b_0$ .
- El elemento  $E$  que, al igual que en el modelo Chua-Yang, define la señal de entrada hacia la neurona.
- Una etapa para ponderar las señales de entrada o salida que serán comunicadas a las demás neuronas, es decir, los elementos no centrales de las mascarillas  $A$  o  $B$ .
- Y los interruptores  $S_1$  y  $S_2$ , que se encargan de seleccionar el flujo de señales que determinan si los elementos no centrales de las mascarillas en la red pertenecen a la de control o a la de retroalimentación y si se utiliza el elemento  $a_0$  o  $b_0$ .

Como puede observarse en este modelo, el integrador y la función de salida están incluidos en el comparador, en donde la variable de estado  $v_{xij}$  es el resultado directo del sumador para un tiempo aproximado de  $t \approx 5h$ , como puede verse en (3.6), con la constante de tiempo definida por la entrada del comparador.

Este modelo, a diferencia del propuesto por Chua-Yang solamente puede trabajar con los elementos no centrales, ya sea de la mascarilla  $A$  o  $B$  y los elementos centrales  $a_0$  o  $b_0$ , pero nunca con ambos (esto en base a los resultados numéricos obtenidos para las mascarillas propuestas en las diferentes tareas de procesamiento mostradas en el apéndice B). A pesar de que aparentemente esto pudiese ser una desventaja, es necesario señalar que la mayoría de mascarillas definidas para el procesamiento de imágenes binarias hacen uso de una sola mascarilla como puede verificarse en [1]. Por otra parte, esto permite reducir notablemente los elementos de circuito ya que el número total de coeficientes en las mascarillas se reduce notablemente.

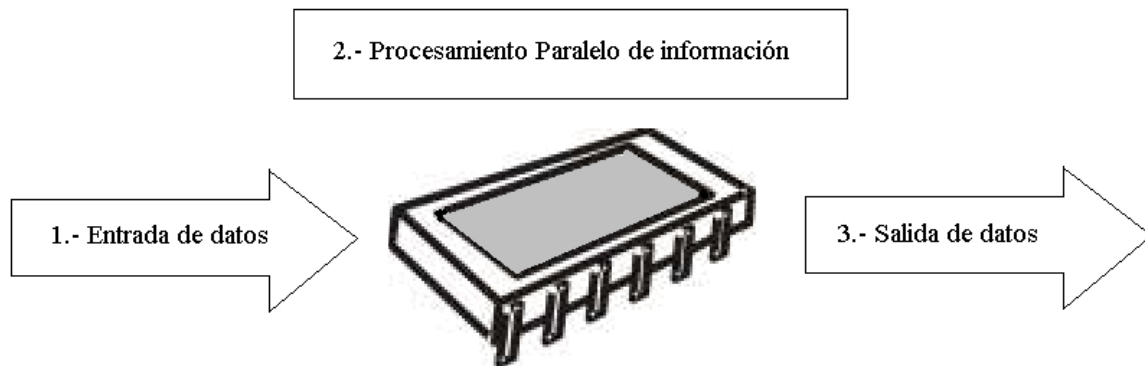
## 4.2 CNN Programable

A través del tiempo han existido muchas y diferentes propuestas para la implementación eléctrica de una CNN. Algunos trabajos se enfocan a la realización de estructuras destinadas a la ejecución de una sola tarea, esto es, sólo son capaces de resolver una tarea de procesamiento. Tales implementaciones llamadas CNN de mascarillas fijas (*fixed-template*) son las que reportan el menor número de transistores por neurona [2], [3], y por lo tanto mayor densidad de integración ( $\approx 230$  neuronas/mm<sup>2</sup>) [4] con un tiempo de convergencia corto.

En este trabajo, como se mencionó en la sección 3.3, se enfoca al diseño de una CNN programable, es decir una estructura que permite eléctricamente cambiar los valores de los elementos de las mascarillas para la ejecución de diferentes tareas de procesamiento. Todo el diseño del arreglo neuronal esta basado en la utilización de tecnología CMOS estándar.

### 4.3 Entrada, Procesamiento y Salida de Datos de una CNN

La mayor ventaja que ofrece la CNN con respecto a los sistemas computacionales convencionales, como ya fue mencionado en el capítulo 1, se obtiene del *Procesamiento Paralelo*. De tal forma que la manera ideal para explotar al máximo tal potencial de procesamiento en un circuito integrado, es necesario que las etapas de entrada de información, procesamiento y salida sean de carácter paralelo (figura 4.3).



**Figura 4.3** Entrada de datos, procesamiento y salida de datos de un CNN chip.

Para implementaciones de arreglos neuronales muy pequeños ( $\approx 3 \times 3$  neuronas), es posible que tanto las entradas como las salidas (siendo señales eléctricas), puedan ser llevadas a cabo de manera paralela siempre y cuando el número de terminales disponibles permitan tal condición. A medida que el tamaño del arreglo aumenta, la entrada de datos a la red generados por algún sensor de imagen tiene que ser organizada en grupos para poder ser introducida secuencialmente. De igual manera, la salida de la red es leída mediante el uso de buses que descargan la información procesada “línea por línea”. Aunque existen

trabajos en donde la entrada y salida de datos es realizada de esta manera [5], esta forma de ingreso y lectura de datos afecta considerablemente el tiempo de procesamiento del sistema.

Actualmente, dada la posibilidad de integrar fotodetectores en tecnología CMOS [6]-[8], es posible crear sobre un mismo sustrato las etapas de adquisición de imagen y procesamiento, superando así el tiempo de demora producido por la entrada secuencial de información [9]-[12], pero aun con la desventaja de conservar la salida multiplexada “línea por línea“ de información procesada, lo cual a pesar de ello, dichas estructuras reportan un desempeño elevado.

Por otra parte, existen algunas propuestas que sugieren la posibilidad de implementar las CNN en otras tecnologías donde es posible llevar a cabo tanto la entrada como la salida de manera óptica [13], lo cual sería muy deseable desde el punto de vista de velocidad de procesamiento, sin embargo, es necesario mencionar que para realizar un arreglo CNN con la capacidad de una máquina universal, se requiere de una gran versatilidad en circuitos que van desde memorias analógicas y digitales hasta lógica digital y funciones analógicas. Tal posibilidad hoy en día puede ser llevada a cabo eficientemente con tecnología CMOS y posiblemente en el futuro existan tecnologías que permitan obtener mejores desempeños de procesamiento que los que se obtienen actualmente.

#### **4.4 Variables Físicas Utilizadas en la Representación de Señales en una CNN**

Las señales internas y externas de una CNN pueden ser representadas como voltajes, corrientes o frecuencias, dependiendo de los criterios de carácter tecnológico para su implementación. En implementaciones donde se pretende obtener una CNN expandible modularmente, la representación en frecuencia es la más apropiada para las señales que se intercambian entre neuronas [14].

Para implementaciones que buscan optimizar la densidad de integración, en la mayoría de los casos es necesario seleccionar el tipo de señal de acuerdo a los circuitos que permitan ser implementados con el menor número de transistores. Esta consideración es tomada en cuenta para el diseño realizado en este trabajo como será mencionado en las secciones posteriores para cada uno de los bloques que constituirán la CNN.

#### **4.5 Desarrollo Electrónico de los Bloques que Constituyen una Neurona CNN Programable**

Las consideraciones tecnológicas del diseño de la red neuronal celular en este trabajo, fueron tomadas del proceso AMI CMOS, ABN 1.2 micras, pozo N, disponible a través de MOSIS, para un área de silicio de 2.2mm X 2.2mm. Las simulaciones se hicieron con PSpice y el modelo utilizado del transistor MOS corresponde a BSIM3 ver. 3 (nivel 8) cuyo conjunto de parámetros fue obtenido de la propia organización MOSIS.

### 4.5.0 Bloque Sumatoria

La representación de señales en modo de corriente ha sido ampliamente utilizada para la implementación de redes neuronales artificiales. Esta característica ofrece ventajas sobre otros tipos de señales eléctricas, siendo una de las más importantes que la suma de varias señales puede llevarse a cabo mediante la utilización de la Ley de Corrientes de Kirchoff, es decir, mediante la concurrencia de las señales de corriente a un solo nodo, lo que permite reducir el número de elementos activos en su implementación. Otras de las ventajas que ofrece, es que permite incrementar la frecuencia de operación debido al uso de nodos internos de baja impedancia y facilita las condiciones para que los transistores MOS trabajen en un amplio rango de operación, que abarca desde inversión débil hasta inversión fuerte.

De esta manera, para la realización del bloque de suma de la figura 4.2, se considera que todos los elementos que constituyen la ecuación (3.5) son llevados a señales representadas por corriente. De tal forma que el bloque sumatoria estará representado por un solo nodo como el que se muestra en la figura 4.4

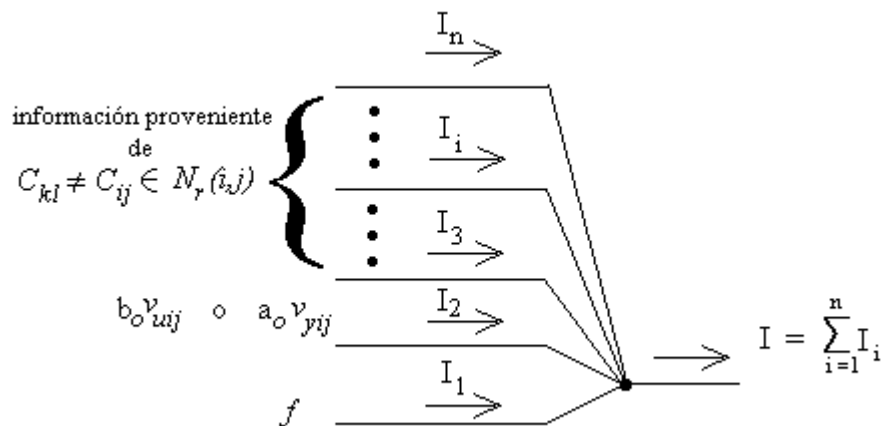


Figura 4.4 Nodo sumatoria.

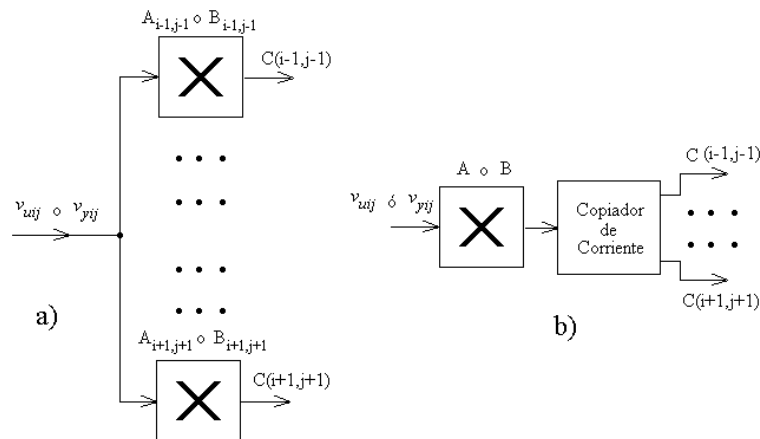
### 4.5.1 Implementación de las Mascarillas A o B

En la implementación eléctrica de una CNN, la realización de las multiplicaciones asociadas a las mascarillas es la etapa que puede consumir la mayor cantidad de área en una neurona. Si la implementación es llevada a cabo siguiendo de manera estricta las ecuaciones del modelo Chua-Yang, los bloques correspondientes a las operaciones de las matrices A y B de la figura 4.1 tendrían que ser implementados haciendo uso de un circuito multiplicador por cada elemento.

Tomando en cuenta, como se mencionó en la sección anterior, que todos los elementos que conforman  $k_3$  serán representados mediante una señal de corriente, para los casos en donde todos los elementos que constituyen las matrices del bloque A o B (con

$a$  y  $b$  de la figura 4.2 son diferentes, la implementación tiene que estar constituida como se muestra en la figura 4.5 (a). Esto es, independientemente del tipo de señal de  $v_{yij}$  o  $v_{uij}$  la multiplicación (ponderación) debe realizarse de manera individual por cada coeficiente, ofrecer su salida en modo de corriente y posteriormente ser llevada a las respectivas neuronas del vecindario.

Por otra parte, muchas de las mascarillas existentes para el tratamiento de imágenes en blanco y negro sugieren soluciones en donde todos los elementos no centrales de cada mascarilla son iguales, tal condición ayuda en el diseño de la CNN reduciendo considerablemente el número de circuitos multiplicadores utilizados. Esto puede ser observado para el mismo caso del bloque A o B de la figura 4.2, en donde si todos los elementos no centrales son iguales, la implementación puede llevarse a cabo como se ilustra en la figura 4.5 (b). Esto es, haciendo uso de un sólo multiplicador (cuya función se reemplaza por un multiplexor analógico) y realizando una copia para cada neurona del vecindario.



**Figura 4.5** Dos formas posibles de implementar el bloque A o B: a) Utilizando un multiplicador por cada elemento b) Utilizar un solo multiplicador y realizar una copia para cada neurona vecina.

Esta característica permite reducir el número de circuitos multiplicadores o ponderadores por cada neurona de la CNN.

De las cinco tareas de procesamiento estudiadas en este trabajo, cuatro de ellas pueden ser realizadas mediante esta última estructura: Removedor de ruido, extractor de bordes, detector de conectividad global y el extractor de sombras, siendo necesario la incorporación de dos etapas de ponderación (multiplexores) para el detector de componentes conectados.

Para la construcción de una estructura CNN capaz de resolver las cuatro tareas de procesamiento mencionadas anteriormente, es necesario el diseño de un copiador de corriente que permita llevar el resultado de la multiplicación (ponderación) hacia las entradas de las demás neuronas. En este caso el diseño se enfoca a un espejo de corriente sencillo. Para el caso particular de la tarea del detector de componentes conectados, como

puede ser visto en su mascarilla en el apéndice B, no cumple con tal condición, por lo que se requiere del empleo de dos multiplexores para la realización de la etapa A o B como se verá posteriormente en la sección 4.6.

#### 4.5.1.0 Espejos de Corriente

El hecho de que todos los coeficientes de las mascarillas sean positivos permite, como ya fue mencionado en el capítulo 3, que la estructura CNN pueda ser implementada a base de circuitos unipolares, lo que reduce notablemente el número de transistores utilizados. Esta característica hace posible que con espejos sencillos de corriente pueda realizarse tanto la etapa para las mascarillas A o B mencionada en la sección anterior, como las fuentes de corriente destinadas hacia las entradas de los multiplexores analógicos encargados de la realización de la ponderación (sección 4.5.2).

Las fuentes y sumideros de corriente son circuitos ampliamente utilizados en el diseño analógico de circuitos integrados, de manera ideal, estos pueden ser vistos como componentes de dos terminales capaces de suministrar o sustraer una corriente independientemente del voltaje que exista en sus terminales. Sin embargo, en la práctica los valores finitos de impedancia de salida y la necesidad de un voltaje mínimo necesario en las terminales que garantice a los transistores operar en la región de saturación, puede limitar el desempeño.

En la figura 4.6 se muestra el espejo de corriente básico. El principio de funcionamiento se basa en que si tanto M1 como M2 son transistores MOS idénticos y además operan en la región de saturación, un mismo potencial  $V_{gs}$  aplicado en sus compuertas debe propiciar que sus respectivas corrientes de drenador sean iguales.

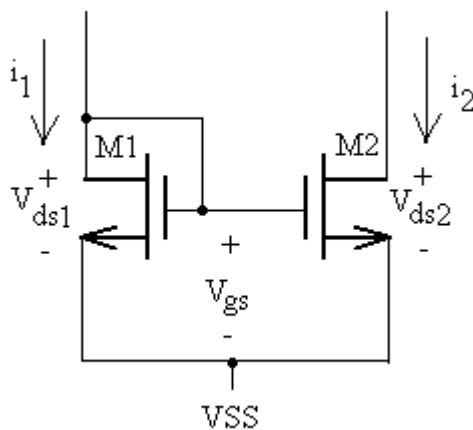


Figura 4.6 Espejo de corriente.

Es evidente que M1 se encuentra en saturación, y por otro lado suponiendo que  $V_{ds2} \approx V_{gs2} - V_{T2}$  y  $V_{gs} > V_{T2}$ , pueden utilizarse las ecuaciones de primera aproximación en la región de saturación para  $i_1$  e  $i_2$  y definir una razón entre ambas:

$$\frac{i_2}{i_1} = \frac{K_2 W_2 / L_2 (V_{gs2} - V_{T2})^2 (1 + \lambda V_{ds2})}{K_1 W_1 / L_1 (V_{gs1} - V_{T1})^2 (1 + \lambda V_{ds1})} \quad (4.0)$$

Normalmente los parámetros físicos como  $V_T$ ,  $K$ , entre otros, en una oblea de silicio varían estadísticamente en razón de las dimensiones y la distancia entre transistores [15], pero haciendo la suposición de que para ambos dispositivos, tales parámetros son idénticos y que además se cumple  $V_{ds1} = V_{ds2}$ , entonces (4.0) puede reducirse a:

$$\frac{i_2}{i_1} = \frac{W_2 / L_2}{W_1 / L_1}$$

lo cual establece que la razón entre las corrientes está dada en función de las relaciones de aspecto geométrico de los transistores.

Para el desempeño correcto del espejo, es necesario que siempre M2 opere en saturación, esto es  $V_{ds2} \approx V_{gs2} - V_{T2}$  o se garantice un voltaje mínimo  $V_{min} = V_{ds(sat)} = V_g - V_T$ . La resistencia de salida del espejo está dada por la resistencia de salida de M2, esto es:

$$r_o = \frac{1}{\lambda i_2} \quad (4.1)$$

donde  $\lambda$  es el factor de modulación de longitud de canal que está dado por:

$$\lambda = \frac{1}{L_{elec}} \frac{dX_{dl}}{dV_{ds}} \quad (4.2)$$

donde  $X_{dl}$  es función de  $V_{ds}$  y representa el ancho de la zona de deserción que reduce el canal eléctrico  $L_{elec}$  en el punto de oclusión (*pinch off*). En forma analítica:

$$L_{elec} = L_{eff} - X_{dl} \quad (4.3)$$

donde  $L_{eff}$  es la longitud de canal efectiva definida en el diseño geométrico.



Ahora, considerando la modulación de longitud de canal para dos transistores de dimensiones idénticas, es decir, para una razón  $W_2L_1/W_1L_2$  unitaria y tomado en cuenta un  $V_T$  y  $V_{gs}$  igual para ellos, de (4.0) se obtiene la siguiente expresión:

$$\frac{i_2}{i_1} = \frac{1 + \lambda V_{ds2}}{1 + \lambda V_{ds1}}$$

Suponiendo que  $\lambda$  es igual para los dos transistores, la ecuación muestra que una diferencia entre los voltajes drenador-fuente de ambos transistores causa una desviación no deseada en la ganancia. Este efecto puede ser atenuado si se disminuye directamente el valor de  $\lambda$  mediante la utilización de transistores de longitud de canal grande (4.3),(4.2), con lo cual se aumenta la resistencia de salida (4.1).

Además del efecto de la modulación de longitud de canal, también la variación aleatoria de parámetros físicos inherentes a cualquier proceso de fabricación tiene efecto en el desempeño del espejo. Esta situación será tratada más adelante en la sección 4.10.

#### 4.5.1.1 El Interruptor MOSFET

Un componente muy importante en cualquier circuito integrado analógico o digital es el interruptor. Un interruptor en tecnología CMOS puede ser implementado mediante el uso de un sólo transistor, ya sea mediante un PMOS o un NMOS como el que se ilustra en la figura 4.7

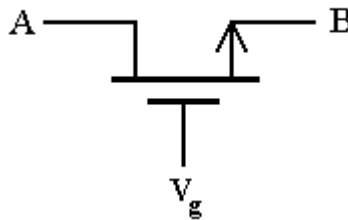


Figura 4.7 Interruptor NMOS.

El interruptor es controlado a través de una señal de voltaje aplicada a la compuerta que determina su estado de encendido o apagado. Las terminales A o B pueden actuar como la fuente o como el drenador del transistor, dada la naturaleza bidireccional de este dispositivo.

El principio de funcionamiento consiste en que para una señal de compuerta  $V_g > V_T$  y  $V_{ds} < V_{gs} - V_T$  entre las terminales, el dispositivo se encontrará en la región lineal de operación, con lo cual un canal de portadores de resistencia  $R_{on}$  unirá a las regiones de drenador y fuente dejando fluir una corriente a través de éstas; tal condición mantendrá al dispositivo en estado de “encendido”.

Teniendo en cuenta que en la región lineal de un transistor MOS, para valores pequeños de  $V_{ds}$  alrededor de cero, la corriente de drenador depende de manera quasi-lineal con el voltaje  $V_{ds}$ , el transistor puede ser tomado como una resistencia. Si consideramos la ecuación para la corriente de drenador  $i_d$  en la región lineal, despreciando los términos de segundo orden, se llega a la expresión:

$$i_d = \frac{KW}{L} \left[ (V_{gs} - V_T)V_{ds} - \frac{V_{ds}^2}{2} \right]$$

Para valores pequeños de  $V_{ds}$  el término cuadrático puede eliminarse, quedando:

$$i_d = \frac{KW}{L} (V_{gs} - V_T)V_{ds}$$

de tal manera que la resistencia  $R_{on}$  de “señal-grande” es:

$$R_{on} = \frac{V_{ds}}{i_d} = \frac{L}{KW(V_{gs} - V_T)}$$

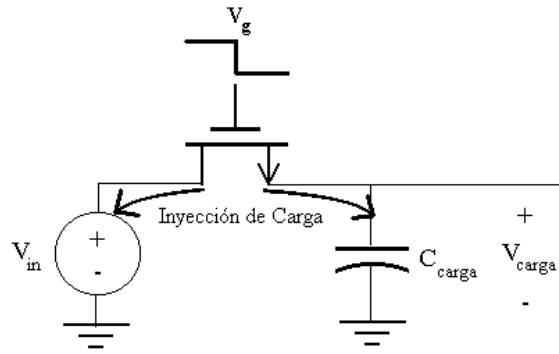
Puede observarse que  $R_{on}$  depende directamente de las dimensiones del transistor.

Cuando el voltaje en la compuerta es  $V_g < V_T$ , el transistor se encuentra en corte y puede ser interpretado como una resistencia infinita entre los puntos A y B, para la cual el interruptor se encuentra en estado de “apagado”. La corriente de fuga en este estado representa el parámetro más importante a considerar, ya que puede alterar el estado de carga de capacitancias parásitas y del diseño.

Mientras que los interruptores basados en un solo transistor MOS tienen el beneficio de una área reducida para su implementación, este tipo de interruptores poseen efectos no ideales que reducen su aplicabilidad, específicamente en circuitos analógicos de muestreo de información. Estos efectos son: Inyección de Carga y “feedthrough”.

#### 4.5.1.2 Inyección de Carga

Observando la figura 4.8, cuando el interruptor MOS se encuentra en estado de encendido, existe una carga debajo de la compuerta debido al canal invertido. Una vez almacenada la información en  $C_{carga}$  asociada a  $V_{in}$  y el interruptor es apagado, dicha carga del transistor se transfiere tanto a la fuente  $V_{in}$  como al capacitor  $C_{carga}$ , lo que ocasiona un cambio de voltaje a través de este último alterando su información .

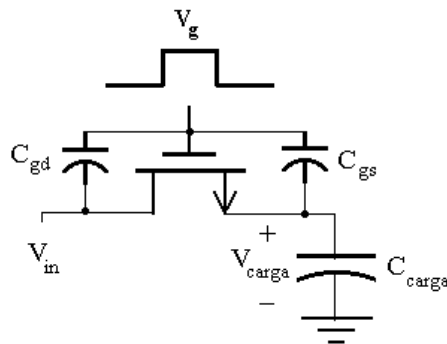


**Figura 4.8** Inyección de carga al condensador  $C_{carga}$ .

Este problema es muy común en circuitos de muestreo y se ha demostrado que la carga puede ser distribuida de manera equitativa tanto para la fuente como para el capacitor cuando se emplean señales de reloj con tiempos de transición pequeños.

#### 4.5.1.3 Efecto “*Feedthrough*”

Para un circuito como el de la figura 4.9, cuando en la señal de reloj de la compuerta tiene a cabo una transición, parte de su señal es transmitida a través de las capacitancias parásitas de compuerta a drenador/fuente en forma de corriente. Esta corriente “transmitida” (*feedthrough*), viene a alterar el voltaje almacenado en  $C_{carga}$  cambiando la información almacenada.



**Figura 4.9** Capacitancias parásitas compuerta-drenador y compuerta-fuente.

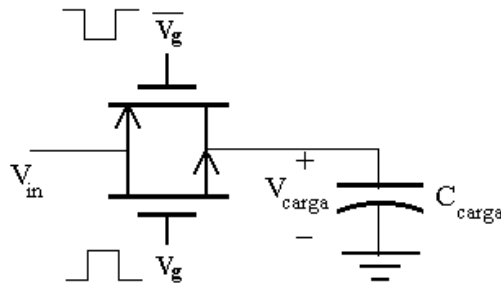
La corriente a través de  $C_{gs}$  puede ser estimada mediante la expresión:

$$i = C_{gs} \frac{d(V_g - V_{carga})}{dt} \tag{4.4}$$

Donde  $C_{gs}$  se calcula a partir de:

$$C_{gs} = C'_{ox} \cdot LD \cdot W$$

Existen métodos para atenuar este efecto, uno de los más usuales es el de sustituir el interruptor MOS por una compuerta de transmisión CMOS (TG) como la mostrada en la figura 4.10



**Figura 4.10** Compuerta de transmisión.

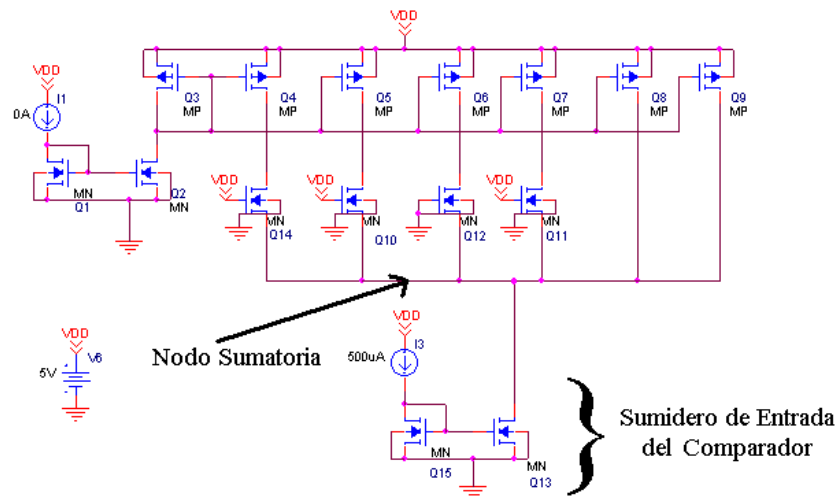
Este interruptor mejora notablemente el desempeño, ya que las señales complementarias utilizadas en las compuertas de los transistores apoya a que las señales de “*feedthrough*” se eliminen mutuamente. Este tipo de interruptor ofrece también un beneficio en la disminución de la resistencia de encendido  $R_{on}$ , la cual es el resultado de la combinación en paralelo de los transistores complementarios. Además, en el área digital, permite transmitir niveles lógicos sin caída de voltaje alguno a través de ella.

#### 4.5.1.4 Bloque A o B

El coprador de corriente de la figura 4.5 (b) fue implementado mediante los espejos de corriente del circuito de la figura 4.11. Como este circuito representa la información que cada neurona compartirá con sus vecinas, los espejos deben ser capaces de proporcionar una buena copia de la corriente de entrada, en este caso  $I_1$ . El rango dinámico de los copiadores debe extenderse a los límites de corriente que establecen las soluciones para las cuatro tareas antes señaladas. Si se toma en cuenta  $1\mu A$  por cada unidad de las soluciones proporcionadas en el apéndice B, el rango dinámico es de 0 a 70  $\mu A$ .

Para determinar las dimensiones apropiadas de los transistores, es necesario tener presentes las dos etapas subsecuentes, esto es, el nodo sumatoria, donde se acumularán todas las corrientes que conforman la ecuación de estado, y el sumidero que corresponde a

la etapa de entrada del comparador (figura 4.11). La condición que hay que tomar en cuenta es que, cuando la corriente a través de todos los transistores que conforman los espejos sea la máxima de 70 aA, el voltaje de drenador de Q13 debe garantizar que los transistores de los espejos permanezcan en la región de saturación. La utilización de los interruptores MOS permitirá cambiar la configuración de las mascarillas para las diferentes tareas a resolver.



**Figura 4.11** Circuito basado en espejos de corriente e interruptores NMOS.

Las dimensiones escogidas para los transistores de este circuito, expresada como cocientes ( $W/L$ ) en micras son:

$$\begin{aligned} Q1=Q2 &= 7.2/5.4 \\ Q3=Q4=Q5=Q6=Q7=Q8=Q9 &= 15/5.4 \\ Q11=Q12=Q13=Q14 &= 9/1.2 \\ Q10=Q15 &= 7.2/5.4 \end{aligned}$$

El voltaje de alimentación es  $V_{DD}=5V$  y la corriente de entrada, como se mencionó anteriormente, estará dada en el intervalo  $[0,70aA]$ .

La figura 4.12 muestra las gráficas de la simulación hecha con PSpice del circuito para una corriente de entrada  $I_1$  de 0 a 70aA. En la gráfica superior se observa que para todo el rango dinámico, los transistores de los espejos mantienen la condición  $V_{ds} \geq V_{gs} - V_T$  que asegura su operación en saturación (esto representado para el  $V_{ds}$  y  $V_{gs}$  del transistor Q4). Además, el voltaje  $V_{ds}$  del interruptor MOS Q14 cumple con la condición  $V_{ds, 0}$  definida en la sección 4.5.1.1.

En la gráfica inferior se observan las copias de corriente de los transistores Q3 a Q9 cuyo interruptor esta en estado de “encendido”. El error de copia es de 7% para el peor caso y la corriente es cero para el transistor Q6 que tiene el interruptor “apagado”.

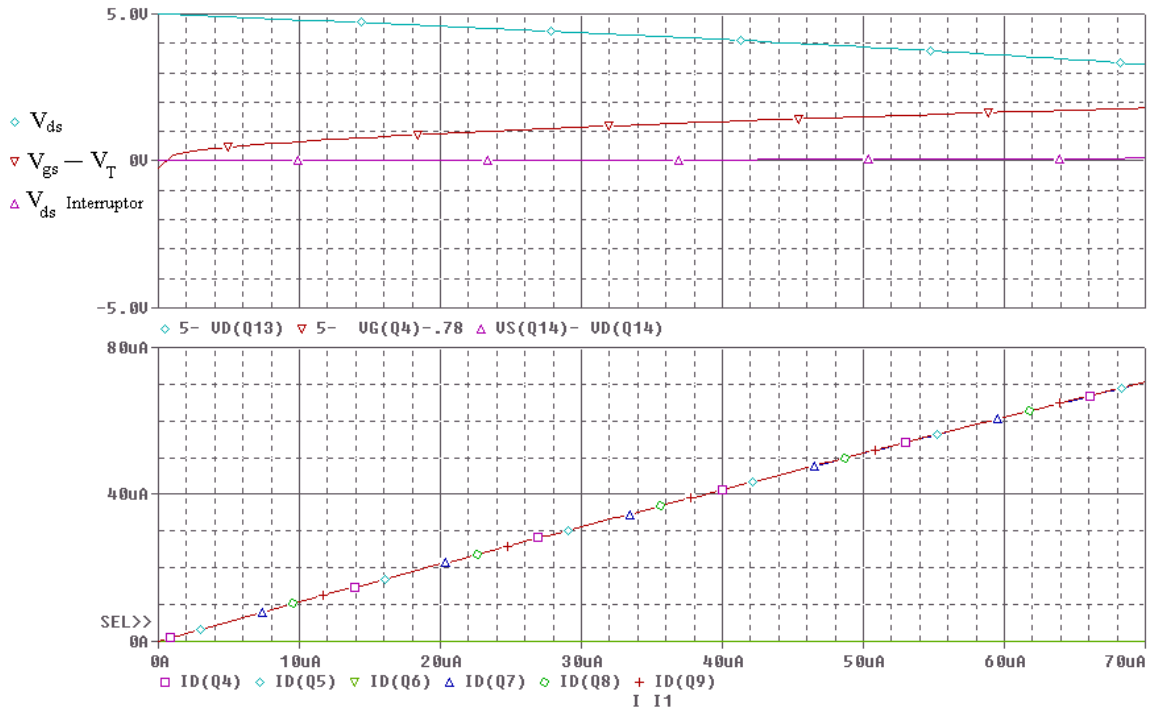


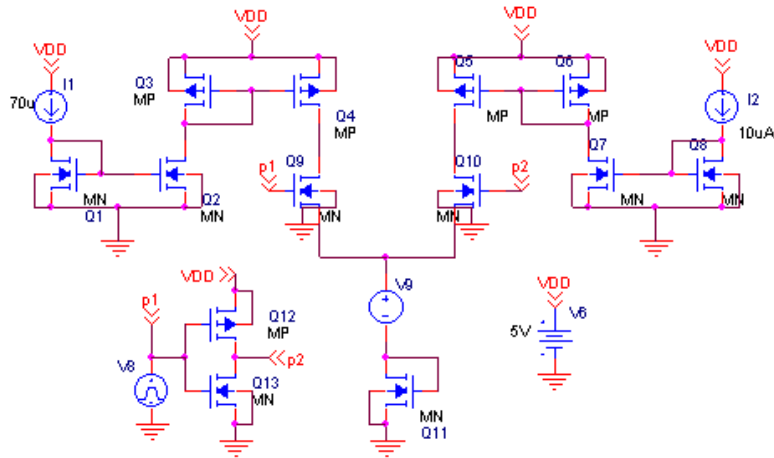
Figura 4.12 Resultados de la simulación en PSpice del circuito de la figura 4.11

### 4.5.2 Bloque Multiplicador

La multiplicación para el modelo CNN con función de salida escalón (como se mencionó en el capítulo 3), puede reducirse a un caso de selección. Un multiplexor analógico como el de la figura 3.2 puede funcionar como circuito multiplicador en la realización de los bloques A o B de la figura 4.5 b) y el  $a_0$  o  $b_0$  de la figura 4.2. Puesto que la salida de este bloque ponderador debe ser en modo de corriente, la idea más sencilla para llevar a cabo el multiplexor es mediante el uso de interruptores MOS, esto es, sustituir los interruptores de la figura 3.2 por transistores y las variables  $K$  y  $-K$  por fuentes de corriente unipolares y variables.

En la figura 4.13 se muestra el circuito, en donde los transistores Q9 y Q10 son los interruptores y Q4 y Q5 las fuentes de corriente variables. Como puede observarse, las fuentes de corriente son básicamente los mismos espejos que fueron empleados para el circuito del bloque A o B de la sección anterior. Para el correcto funcionamiento del circuito, es necesario que las señales  $v_{uij}$  o  $v_{yij}$  ( $V8$  en la figura 4.13) de la neurona

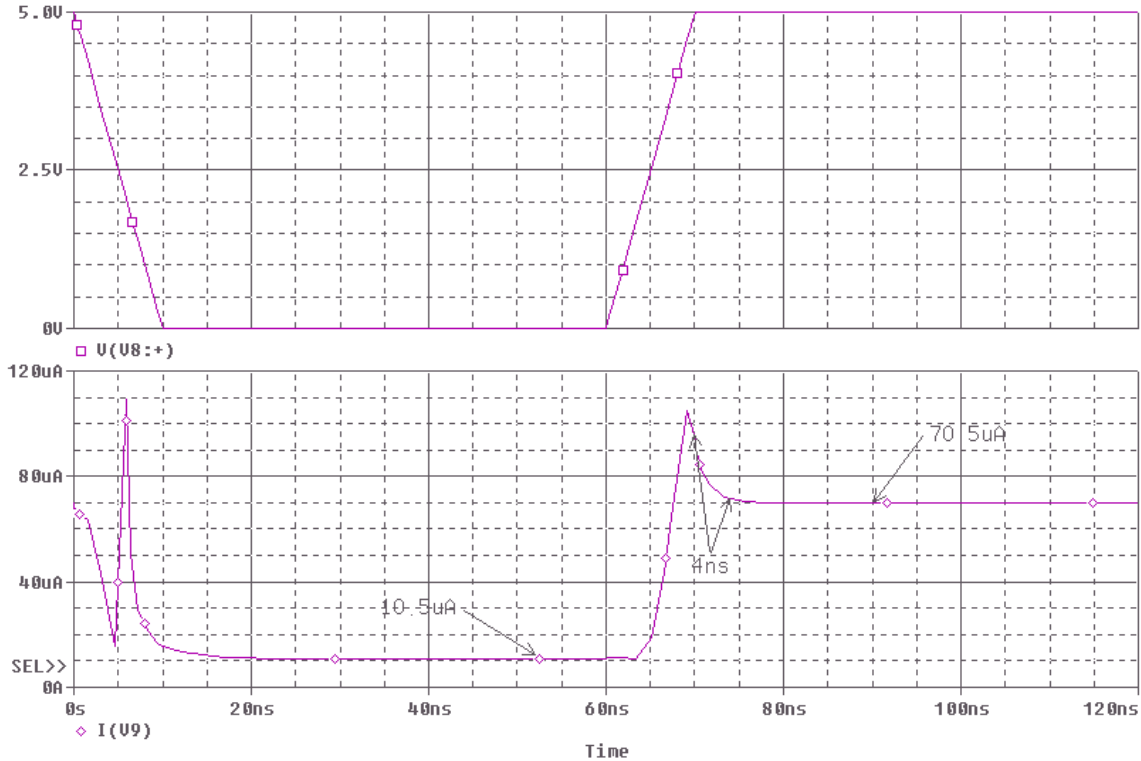
correspondan a niveles de voltaje lógico CMOS, de esta manera dichas señales con su complemento (el cual se obtiene con el inversor formado por Q12 y Q13) “encienden” el interruptor que deja fluir la corriente correspondiente para una salida  $v_{yij}$  o entrada  $v_{uij}$ , blanca o negra.



**Figura 4.13** Circuito de un multiplexor analógico constituido por Q9 y Q10.

La figura 4.14 muestra las gráficas de la simulación del circuito en PSpice para esta configuración. Se utilizó un sólo ciclo de una señal cuadrada para encender alternadamente ambos interruptores y probar que las corrientes a través del transistor Q11 (que representa la entrada de la siguiente etapa) fueran las adecuadas. La señal cuadrada se ilustra en la gráfica superior, donde los tiempos de subida y bajada son de 10ns. Cuando la señal cuadrada se encuentra en estado bajo, el interruptor Q9 es llevado a corte, mientras Q10 es llevado a la región lineal permitiendo el flujo de la corriente que proporciona el transistor Q5, la señal obtenida se ilustra en la gráfica inferior de la figura 4.14. De igual forma cuando la señal cuadrada se encuentra en estado alto, el interruptor que se enciende es Q9 y Q10 se apaga, obteniéndose la corriente de la copia del transistor Q4. En esta misma gráfica, puede observarse el efecto de “feedthrough” ocasionado por la señal cuadrada aplicada a las compuertas de los interruptores MOS. En éstos, siguiendo lo establecido en la ecuación (4.4), la corriente se transmite durante los flancos de subida y bajada de la señal cuadrada a través de las capacitancias parásitas, y altera el valor de la corriente en Q11, la cual vuelve a estabilizarse después de 4ns del término del flanco.

Este tiempo de recuperación de la corriente permite que la implementación pueda ser llevada a cabo mediante este tipo de interruptores, sin necesidad de utilizar dispositivos adicionales u otros circuitos que eliminen estos efectos no ideales.



**Figura 4.14** Gráficas de la simulación en Pspice: Arriba la señal de entrada; abajo, la corriente de salida del multiplexor (V9).

### 4.5.3 Bloque Comparador

La implementación eléctrica del bloque correspondiente al comparador de la figura 4.2 debe reunir dos características fundamentales: Primero, debe permitir el acoplamiento con el bloque sumatoria, esto es, la entrada al comparador debe estar dada en modo de corriente y poseer resistencia de entrada baja. Y segundo, la salida la cual es definida como  $v_{yij}$ , debe corresponder a niveles lógicos CMOS necesarios para el funcionamiento de la etapa subsecuente de los multiplexores.

El circuito de la figura 4.15 reúne estas características. Esta estructura tiene la parte de entrada de un transportador de corriente (*current-conveyor*) de 2ª generación [16], salvo que en lugar de utilizar un amplificador diferencial con un voltaje de referencia  $V_{ref}$ , éste emplea un inversor CMOS (transistores Q5-Q6) como amplificador [17]. Dado que los transistores Q3-Q4 actúan como un amplificador (buffer) de voltaje clase B, el voltaje de entrada  $V_{in}$  que corresponde al voltaje de transición  $V_{tran}$  (para el cual la salida del circuito cambia de estado, ya sea de nivel alto a bajo o viceversa) del circuito, se obtiene por las características del inversor cuando sus dos transistores operan en saturación, esto es, cuando la corriente a través de ellos es la misma:



$$\frac{\beta_n}{2}(V_{tran} - V_{Tn})^2 = \frac{\beta_p}{2}(VDD - V_{tran} - V_{Tp})^2$$

De manera que el voltaje de entrada  $V_{in}$ , para el cual la salida cambia de nivel lógico está dado por:

$$V_{tran} = \frac{\sqrt{\frac{\beta_n}{\beta_p}} \cdot V_{Tn} + VDD - V_{Tp}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}}$$

donde se observa que dicho voltaje depende de parámetros tecnológicos.

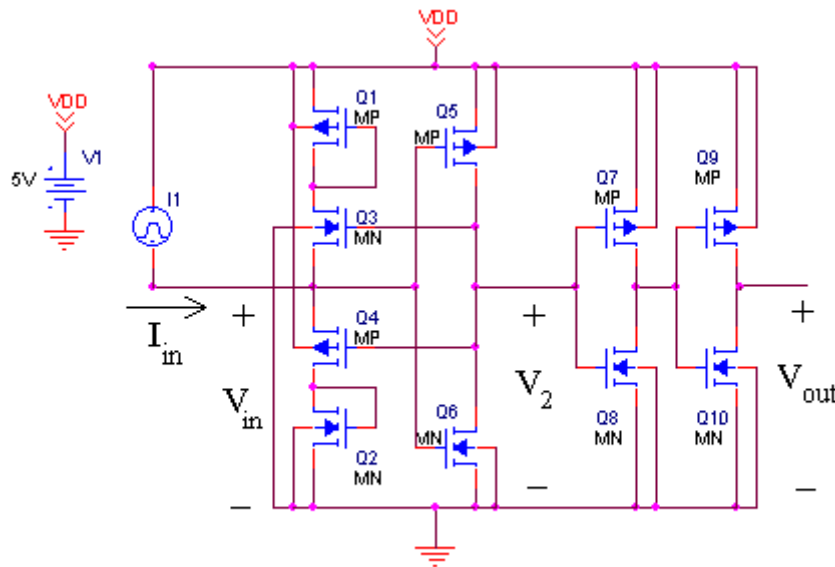


Figura 4.15 Circuito comparador.

El circuito tiene tres modos de operación:

1.- Cuando  $I_{in}$  es positiva (en el sentido mostrado en el diagrama), el voltaje de entrada  $V_{in}$  es llevado hacia un nivel por encima de  $V_{tran}$ , el cual a través del inversor Q5-Q6 hace que  $V_2$  vaya hacia un nivel bajo, lo cual provoca que Q3 vaya a corte (ya que el  $V_{gs}$  de Q5 y Q6 es negativo) y Q4 a saturación. En este estado,  $V_{in}$  es un nodo de baja impedancia puesto que la corriente de entrada  $I_{in}$  es drenada fácilmente a través de Q4.

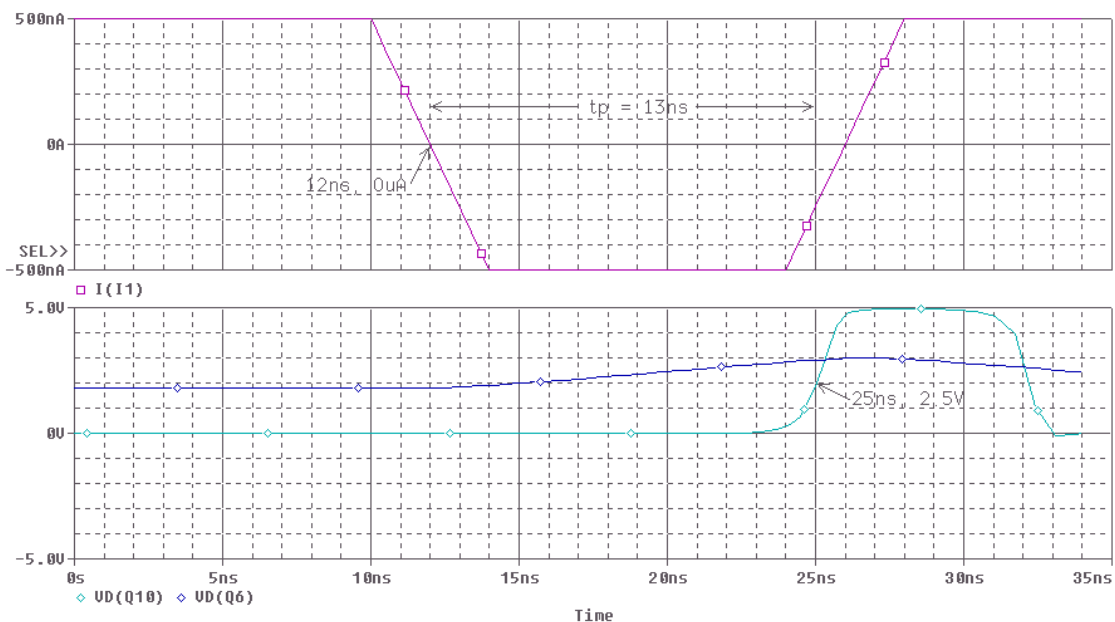
2.- Cuando la corriente de entrada cambia de signo, temporalmente no hay manera de suministrar la corriente demandada, por lo que el nodo de entrada  $V_{in}$  es de alta impedancia.

3.- Cuando  $I_{in}$  es negativa y  $V_{in}$  ha alcanzado un valor por debajo de  $V_{tran}$ ,  $V_2$  va a nivel alto y la corriente ahora es suministrada por Q3 manteniendo la baja impedancia de entrada.

Este circuito, como es mencionado en [18], posee una resistencia de entrada baja de aproximadamente  $1/g_m$ , lo cual permite que espejos de corriente simples puedan ser utilizados en su entrada. Sin embargo, una de sus desventajas es que la señal de voltaje en la entrada del inversor de retroalimentación positiva Q5-Q6, no cambia completamente de VDD a GND, ocasionando que no se “apaguen” completamente sus transistores. Esta condición provoca que, además del consumo de potencia en estado estable, sea necesario el empleo de dos inversores adicionales en cascada Q7-Q10 para garantizar una salida en niveles lógicos CMOS.

Por otra parte, el empleo de los transistores Q1 y Q2 como cargas activas, permite desacoplar los voltajes de polarización de los transistores Q3 y Q4, disminuyendo asimismo el voltaje  $V_{ds}$  a través de ellos. Esta situación permite disminuir el efecto de  $\lambda$  (modulación de longitud de canal) en la corriente de drenador de ambos transistores, con lo que se tiene una mejoría en la sensibilidad del comparador sin alterar la resistencia de entrada.

En la figura 4.16 se muestran las simulaciones del circuito en PSpice, con las siguientes dimensiones de transistores:  $Q1=Q2=9/5.4$ ,  $Q3=Q6=Q8=Q10=2.4/1.2$ ,  $Q4=Q5=Q7=Q9=7.2/1.2$ . La gráfica superior representa la señal de entrada y la inferior la respuesta del circuito. Ya que la señal de entrada corresponde a la magnitud mas pequeña de corriente que el comparador puede detectar, la sensibilidad del circuito es de  $0.5\text{aA}$ , mejorando casi dos veces la reportada en [18], pero con un tiempo de propagación mayor  $t_p = 13\text{ns}$ .



**Figura 4.16** Resultados de la simulación del circuito comparador, arriba la señal de corriente de entrada, abajo el voltaje  $V_{out}$  de salida del comparador.

En este sentido es necesario mencionar que según [8], el circuito presenta un entrada predominantemente capacitiva para corrientes de entrada pequeñas, y una entrada predominantemente resistiva para corrientes de entrada grandes. Tal situación concuerda con los resultados obtenidos en la simulación, ya que para señales de entrada con una amplitud  $\gg 0.5\mu\text{A}$  los resultados de la simulación muestra una reducción en el tiempo de propagación en aproximadamente 3ns.

Para poder realizar la función escalón desplazada mediante este circuito, es necesario incluir un sumidero de corriente a la entrada del circuito de la figura 4.15, de manera que la corriente de entrada al comparador sea el resultado de una sustracción de corrientes. El sumidero se basa en un espejo de corriente sencillo y se muestra en la figura 4.17. El resultado de la simulación en PSpice para una corriente de entrada de 0 a 60 $\mu\text{A}$  se muestra en la figura 4.18, donde puede observarse el escalón de salida del comparador  $V_{\text{out}}$  para una corriente del sumidero igual a  $X=30\mu\text{A}$ , la salida del inversor Q5-Q6 y el voltaje de entrada  $V_{\text{in}}$ .

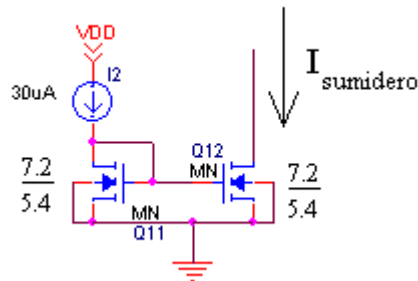


Figura 4.17 Sumidero de corriente para la implementación de la función escalón con desplazamiento.

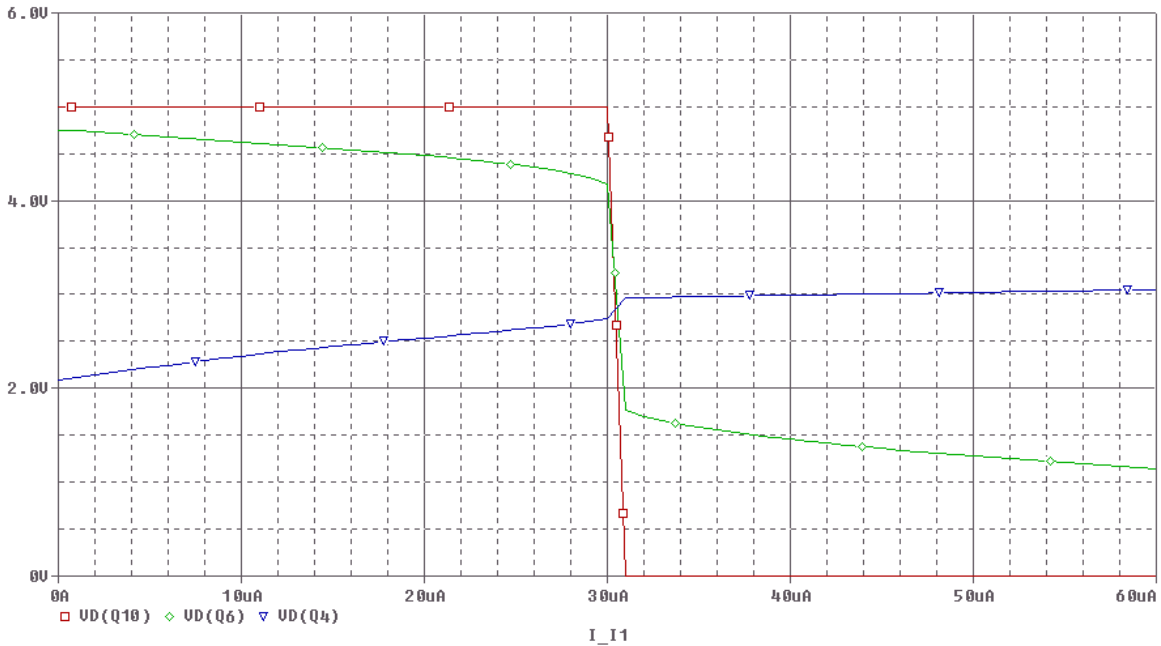


Figura 4.18 Simulación en PSpice de la función escalón con desplazamiento

En este bloque del comparador, es necesario incluir circuitos adicionales que permitan cargar las condiciones iniciales hacia la neurona, en este sentido, como se estableció en (3.8), en el modelo de esta tesis es suficiente con que se cumpla la condición:

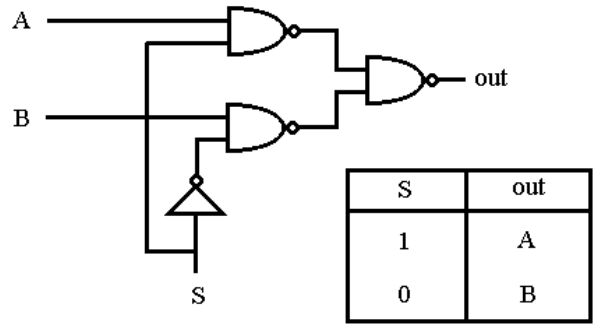
$$v_{yij}(0) = v_{uij} \tag{4.5}$$

para cualquier valor de  $v_{xij}(0)$ . Esta condición puede ser llevada a cabo con la introducción de interruptores MOS en la entrada del inversor Q7-Q8, como se verá más adelante en la sección 4.6.

#### 4.5.4 Bloques S<sub>1</sub> y S<sub>2</sub>

Los bloques de los interruptores S<sub>1</sub> y S<sub>2</sub> de la figura 4.2 solamente están destinados a la tarea de permitir el cambio de flujo de información a través de la neurona, esto es, que los bloques correspondientes a los coeficientes de las mascarillas consideren la salida  $v_{yij}$  o la entrada  $v_{uij}$  de la neurona.

Ya que tanto  $v_{yij}$  como  $v_{uij}$  son señales que van a estar dadas en niveles de voltaje lógico, los bloques S<sub>1</sub> y S<sub>2</sub> pueden ser implementados mediante un multiplexor digital de dos entradas como el que se muestra en la figura 4.19 con su correspondiente tabla de verdad.



**Figura 4.19** Multiplexor digital de dos entradas realizado mediante compuertas nand y su tabla de verdad.

El circuito eléctrico del multiplexor digital se muestra en la figura 4.20, donde las dimensiones para los transistores NMOS son 2.4/1.2 y para los PMOS 7.2/1.2. Los resultados de una de las simulaciones en PSpice se muestra en la figura 4.21, donde se observa la salida del circuito para los valores A=0, B=1 y una señal lógica de entrada S. El tiempo de propagación de la señal para el peor caso es de 1.95ns, considerando una capacitancia de carga de 40fF, la cual es aproximadamente la que manejará el multiplexor en la implementación global de una neurona.

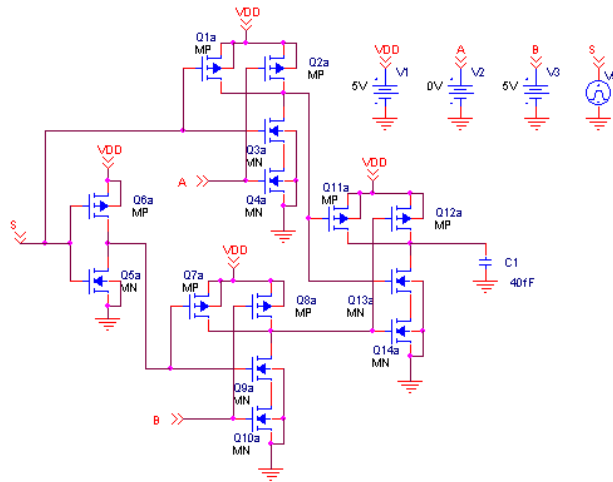


Figura 4.20 Circuito CMOS del multiplexor digital.

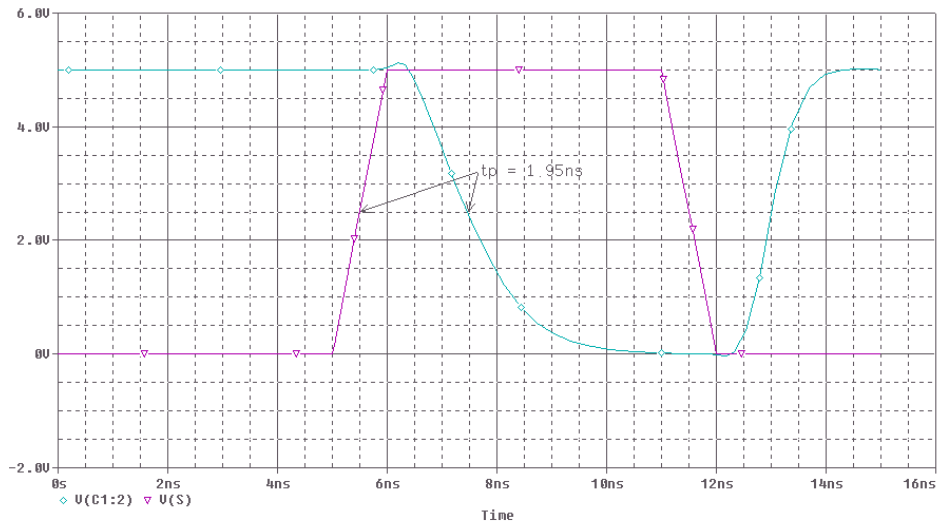
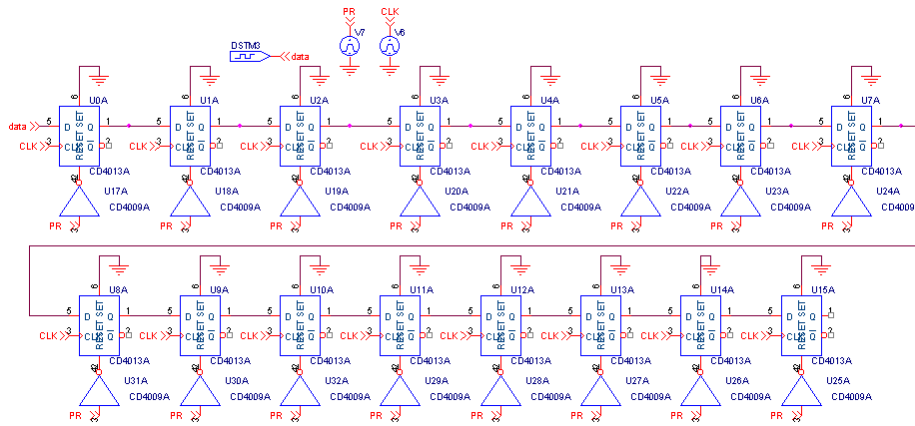


Figura 4.21 Resultado de una de las simulaciones del multiplexor digital.

### 4.5.5 Bloque de Entrada

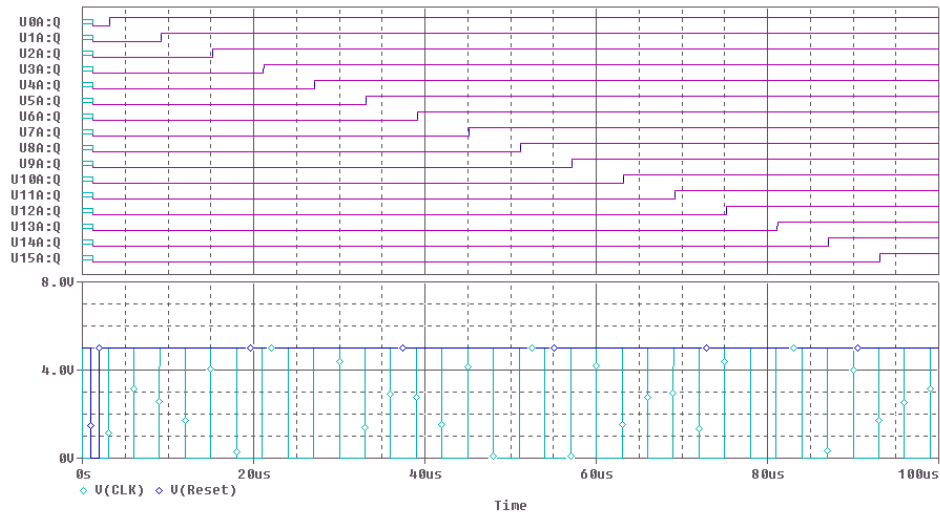
Como se comentó antes, el diseño debe realizarse en un área de silicio de 2.2mm x 2.2mm, lo cual lo limita a solamente 40 terminales disponibles en un encapsulado DIP. Dada la limitante en terminales de entrada/salida, se determinó que la entrada hacia la red fuera en modo serial mediante un registro de corrimiento implementado con Flip-Flops tipo D.

En la figura 4.22 se muestra el esquema del circuito, donde se observa que se utilizan flip-flops tipo D de la biblioteca CD4000 de PSpice para la simulación.



**Figura 4.22** Registro de corrimiento implementado mediante flip-flops tipo D.

En a figura 4.23 se muestra el resultado de una de las simulaciones en PSpice considerando una entrada permanente al registro en nivel alto, a una frecuencia de reloj de 167kHz.



**Figura 4.23** Simulación del registro de corrimiento para una entrada sincrónica siempre en “alto”.

### 4.5.6 Etapa de Salida

Tomando en cuenta la cantidad de terminales disponibles, es posible destinar 16 para la salidas de cada neurona y así obtener la salida del procesamiento de manera paralela. Además, se prevee que el diseño disponga de una etapa de salida que permita mediante el equipo de mediciones eléctricas del laboratorio, leer la información sin producir un retraso considerable de la señal.

Si la capacitancia de las puntas del equipo de prueba es de aproximadamente 10.8pF, es necesario que la etapa de salida pueda cargar y descargar esa capacitancia en el menor tiempo posible. En este sentido, se puede utilizar un buffer fuera del chip que permita hacer parte de la labor, el buffer 74AC11244 de Texas Instruments, comercialmente disponible, permite manejar cargas de hasta 50pF con un tiempo de propagación típico de 5ns y una capacitancia de entrada de 4pF. Así, tomando en cuenta este buffer externo, es necesario el diseño de un buffer integrado propio para cada neurona que permita manejar esta capacitancia de 4pF (despreciando la capacitancia de los pads r 0.1pF) en un tiempo breve.

El buffer integrado consiste en una serie de inversores en cascada donde cada inversor posee dimensiones mas grandes de W que el previo en un factor de A (ver figura 4.24). Usando las expresiones de primera aproximación de [19] y los parámetros del proceso, es posible calcular un buffer de tres etapas N=3 y un factor de A=6, cuyo tiempo de retraso sea aproximadamente 1.25ns. El esquema del circuito se muestra en la figura 4.25

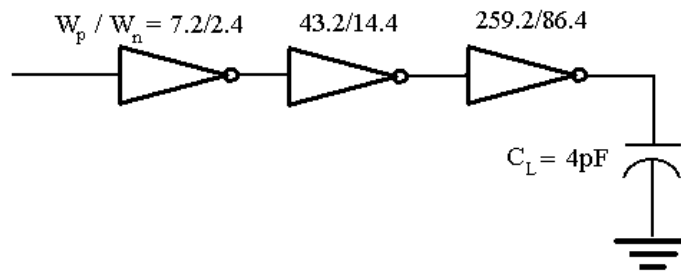


Figura 4.24 Buffer de salida en base a una cadena de inversores.

Todos los transistores utilizados tienen  $L = 1.2\mu\text{m}$ . Los resultados de la simulación (figura 4.26) muestran que para una misma señal de entrada un inversor de mínimas dimensiones, a diferencia del buffer, no es capaz de cargar rápidamente la capacitancia de 4pF.

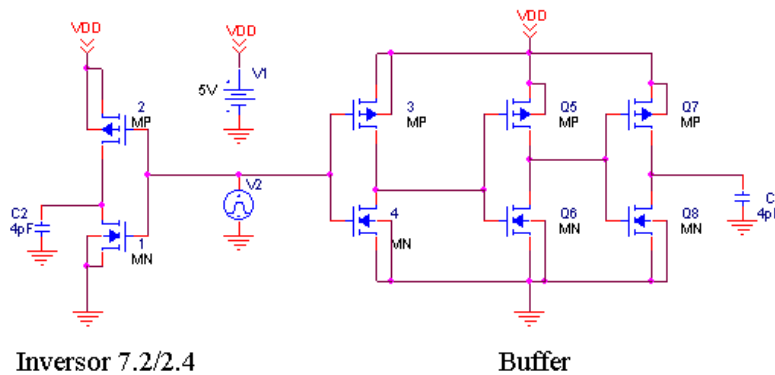


Figura 4.25 Circuito compuesto del buffer de salida (derecha) y un sólo inversor (izquierda).

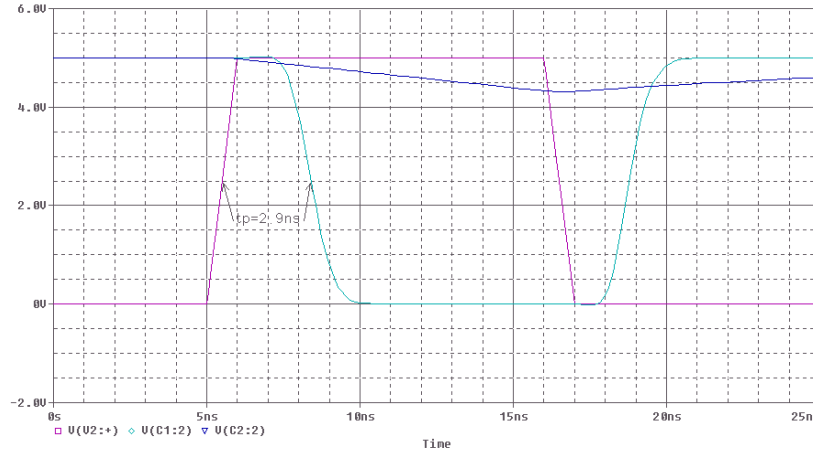


Figura 4.26 Resultados de la simulación del buffer y el inversor con la capacitancia de carga de 4pF.

### 4.6 Arquitecturas de las Neuronas Propuestas

Con base a los circuitos que conforman los bloques antes mencionados, se diseñaron dos arquitecturas diferentes de neurona con el propósito de verificar el desempeño en la ejecución de las diferentes tareas.

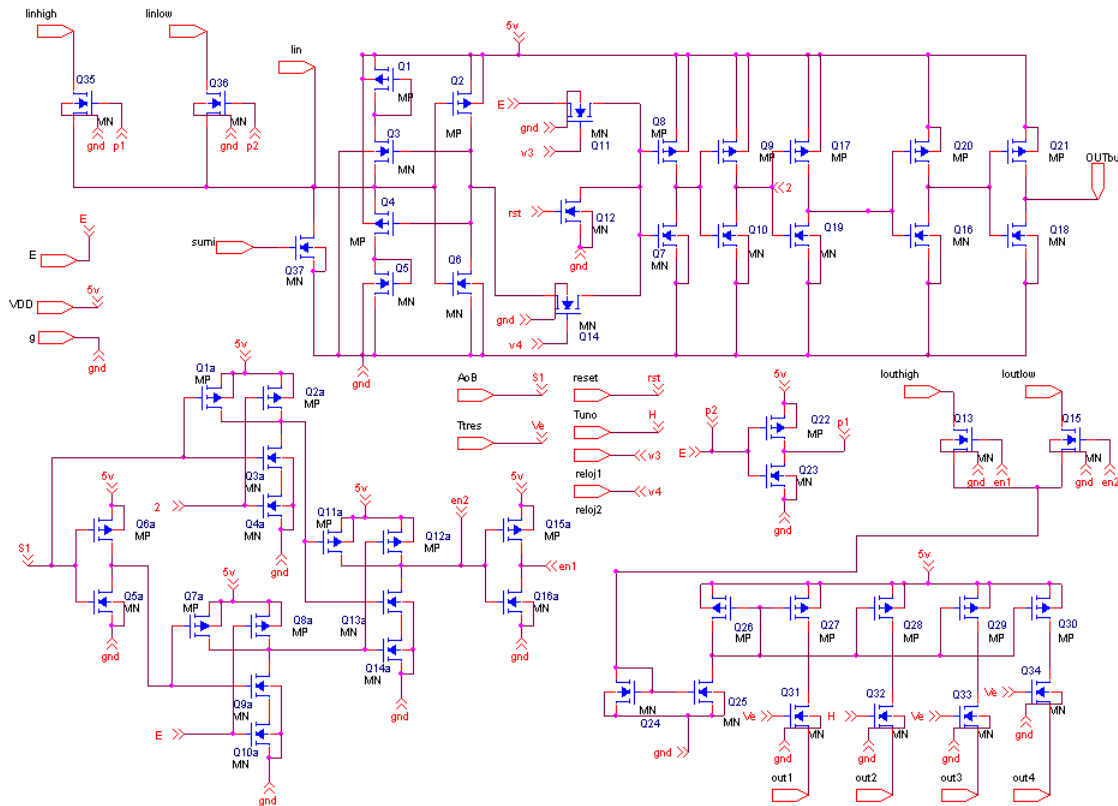


Figura 4.27 Arquitectura de una sola neurona para la resolución de las cuatro primeras tareas de procesamiento del apéndice B.



La primera se muestra en la figura 4.27, esta estructura fue diseñada con el propósito de tener un primer prototipo de CNN de 4x4, la cual es capaz de resolver solamente aquellas tareas donde los elementos no centrales de las mascarillas A o B sean iguales (sección 4.5.1) y no requieren del elemento  $a_0$  (4 primeras del apéndice B). Por lo tanto, utiliza un solo multiplexor digital «  $S_2$  ». Dentro del arreglo de 4x4 puede observarse el empleo de tres interruptores MOS antes de la etapa de salida del comparador. El transistor Q11 hace la función de permitir en un lapso conectar la entrada “E” de la neurona con las compuertas de Q7 y Q8 cargando la condición inicial (4.5); posteriormente al apagarse, Q14 restablece la conexión “normal” del comparador. La función de Q12 es permitir inicializar eléctricamente la entrada del inversor a un cero lógico. Dada la configuración en esta arquitectura es necesario señalar que el patrón de entrada a procesar debe ser introducido a la red en forma de su complemento.

Con el propósito de observar el desempeño del modelo para la tarea del detector de componentes conectados, se elaboró una segunda arquitectura que se muestra en la figura 4.28. Ésta, a diferencia de la primera, utiliza dos multiplexores analógicos en la etapa A o B y los dos multiplexores digitales en la implementación de  $S_1$  y  $S_2$ .

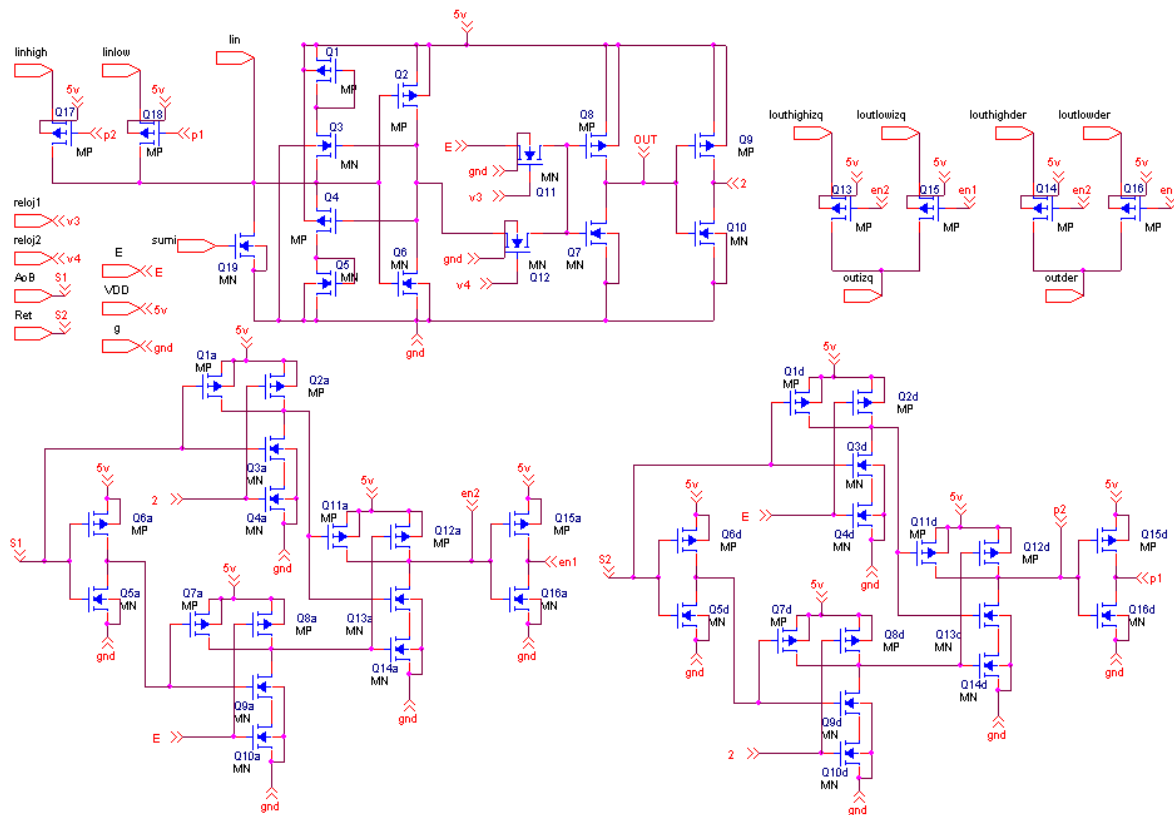


Figura 4.28 Arquitectura de la neurona para la resolución del problema de CCD.

## 4.7 Neuronas de Prueba

Dentro de un circuito integrado prototipo, es recomendable agregar circuitos de apoyo que permitan hacer pruebas funcionales del circuito central. Estas pruebas pueden clasificarse en *globales* y *locales* [20].

Las pruebas *globales* consisten en una aplicación sucesiva de patrones de entrada a la CNN para una interpretación posterior de las salidas y los estados. Obviamente estos patrones dependen de las tareas de procesamiento específicas de la CNN y su número depende de la complejidad de la misma. Por otro lado, las pruebas *locales* se refieren al hecho de poder separar cada neurona del arreglo y ya sea de manera simultánea o secuencial, probar cada una de ellas mediante un mismo patrón de entrada.

En este sentido dentro de la CNN de 4x4 se contemplan dos neuronas (una regular y una de frontera) que poseen un circuito como el de la figura 4.29 (a), el cual hace posible que a través de dos terminales externas del chip se pueda medir la corriente que incide en el nodo sumatoria. Esta característica permite hacer mediciones en estática (DC) para conocer el desempeño de cada uno de los bloques que conforman la neurona. Además se incluye un sumidero adicional con salida externa para medir la calidad de su respectiva copia de corriente (figura 4.29 (b)).

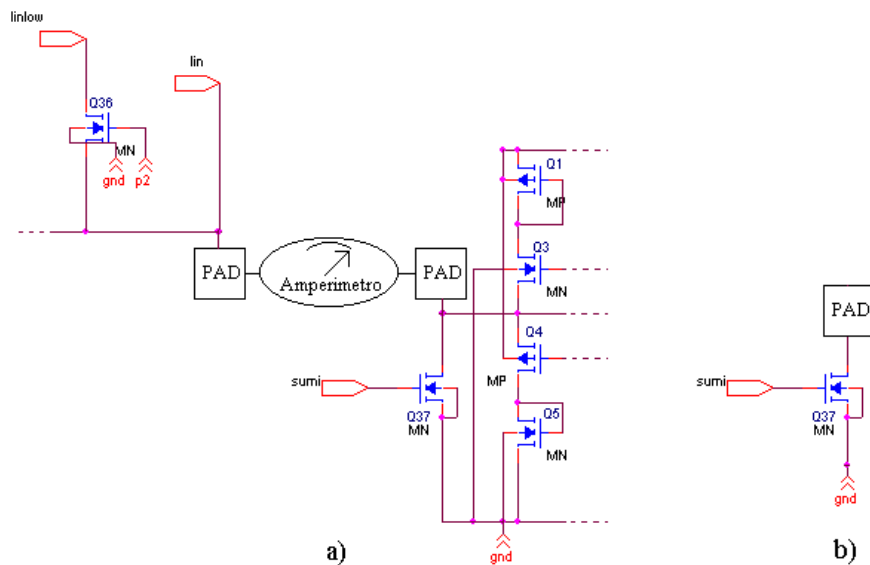


Figura 4.29 Arreglo interno de las neuronas de prueba.

Esta consideración para la ejecución de pruebas *locales*, dada la capacitancia inherente de los pads y las características eléctricas del amperímetro, hace que sólo puedan llevarse a cabo cuando la CNN ha alcanzado el estado estable. Por esta razón, aun cuando la dinámica de la red se realice con un cortocircuito entre ambas terminales, existirá un aumento en la capacitancia del nodo sumatorio. Según los datos tecnológicos proporcionados por MOSIS, las capacitancias de los pads contribuirán para este caso, con

un aumento de 0.33pF en dicho nodo. Este aumento de capacitancia debe tomarse en cuenta para las neuronas de prueba durante el análisis en simulaciones.

### 4.8 Simulaciones Eléctricas de una CNN de 4x4 para la Ejecución de las Diferentes Tareas de Procesamiento

A continuación se muestran algunos de los resultados obtenidos de la estructura CNN propuesta para la resolución de las cuatro primeras tareas del apéndice B.

#### 4.8.0 Diagrama Esquemático del Circuito

La simulación del arreglo neuronal estuvo basada en bloques jerárquicos en donde cada bloque de la figura 4.30 representa una neurona como la de la figura 4.27. Pueden observarse las entradas de voltaje y corriente que definen la tarea de cada neurona, así como la capacitancia de carga de 4pF a la salida del buffer de cada una de ellas.

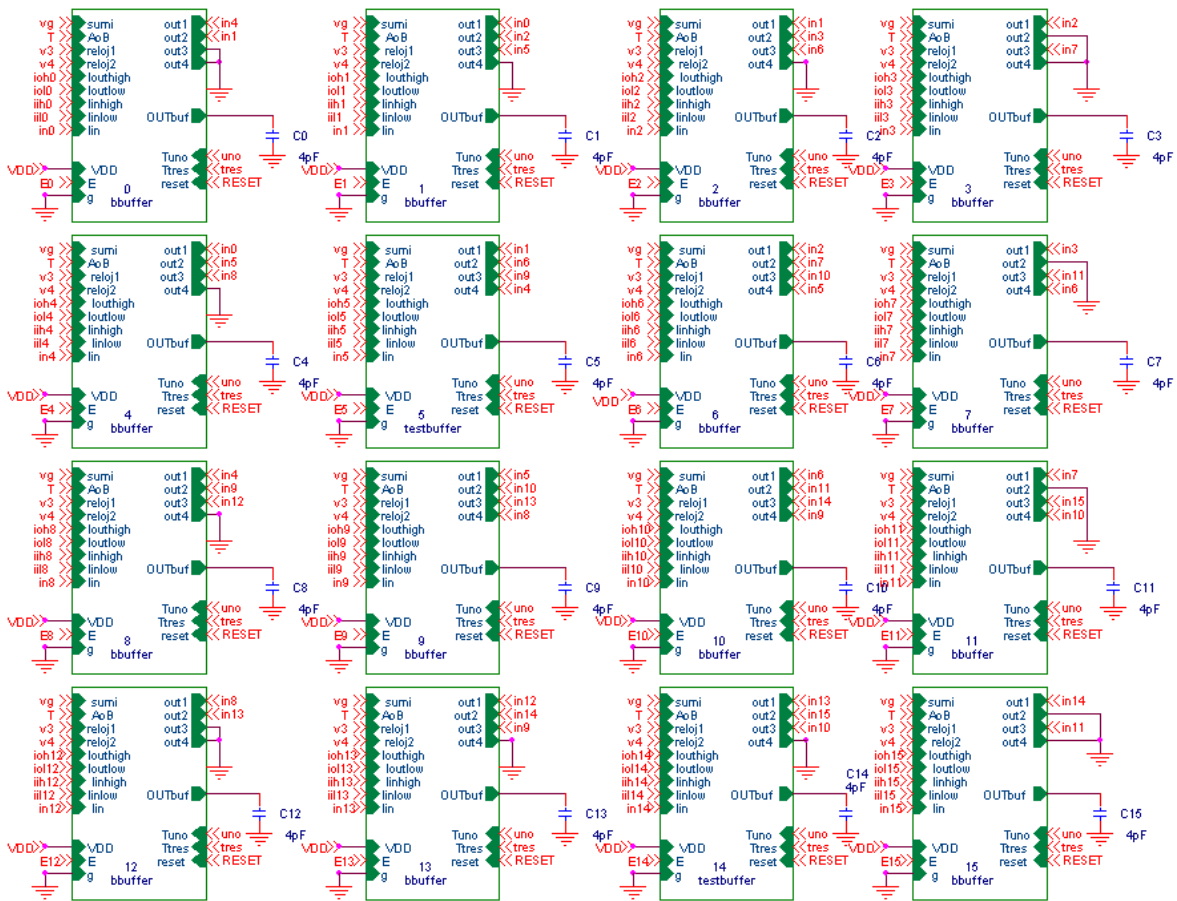
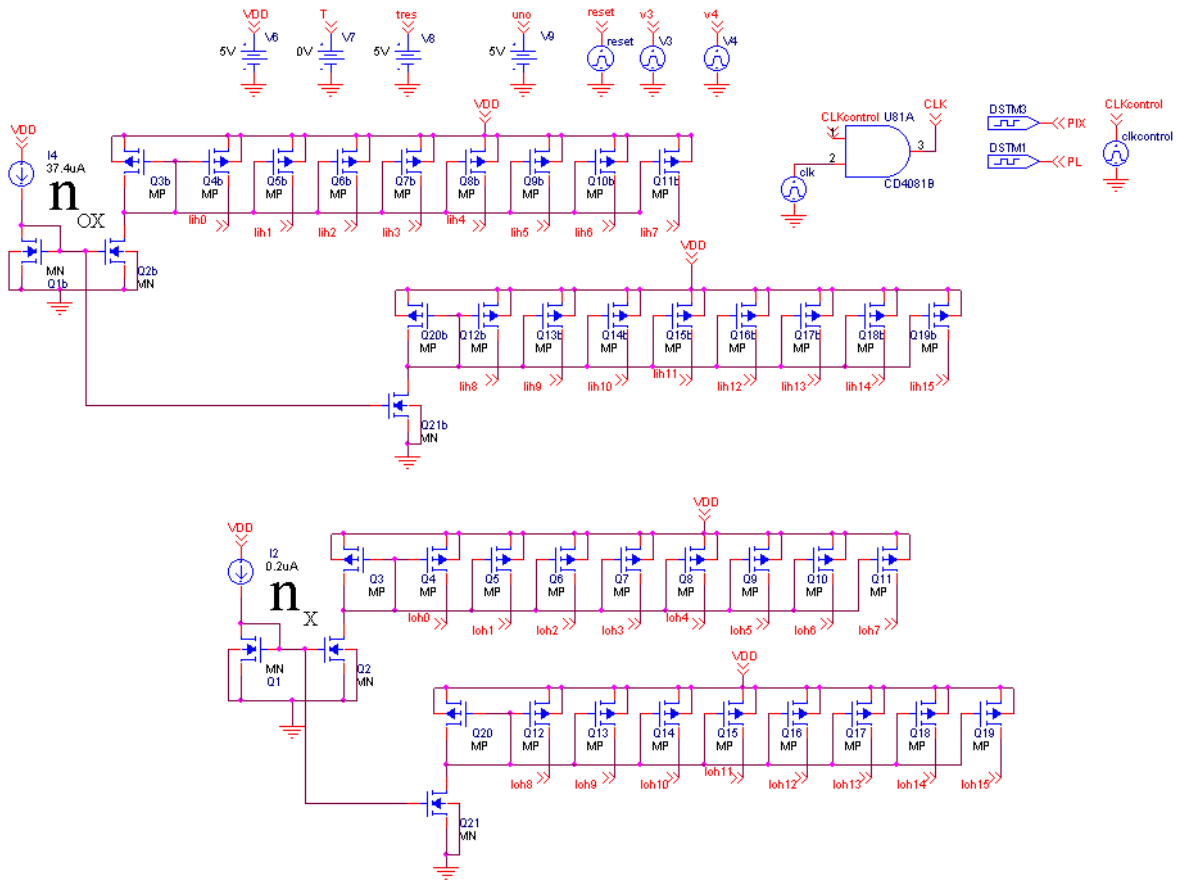


Figura 4.30 Arreglo CNN de 4x4.

En la figura 4.31 y 4.32 se muestran los copiadores de corriente que suministran la corriente a los multiplexores analógicos y programan de manera “analógica” la tarea de

procesamiento que desempeñará la red. La fuente de corriente independiente de cada copiadore simula a la corriente que debe ser suministrada de manera externa al chip y corresponde con las soluciones numéricas para las diferentes tareas mostradas en el apéndice B.



**Figura 4.31** Esquema donde se muestran los copiadores de corriente destinados a la implementación de los coeficientes de las mascarillas y las fuentes de voltaje.

Las fuentes de voltaje V7, V8 y V9 de la figura 4.31 permiten cambiar la configuración de la CNN, esto es, permiten definir el tipo de mascarilla y las conexiones con las neuronas del vecindario; reset, V3(reloj1) y V4(reloj2) son los pulsos que inicializan eléctricamente la red, cargan las condiciones iniciales y habilitan el procesamiento de la red, respectivamente. PIX es la información serial correspondiente a los patrones de entrada que es transmitida y almacenada a través del registro de corrimiento, PL corresponde al *clear* de los flip-flops y CLKCONTROL es la señal que permite controlar la secuencia en que la información es transmitida al registro de corrimiento.

En la figura 4.32 puede observarse la fuente de corriente I6 que tiene por objeto establecer el voltaje de referencia “vg” que se aplica a los sumideros de corriente. El

registro de corrimiento en la figura 4.33 se encarga de transferir el patrón de entrada hacia la entrada respectiva de cada neurona .

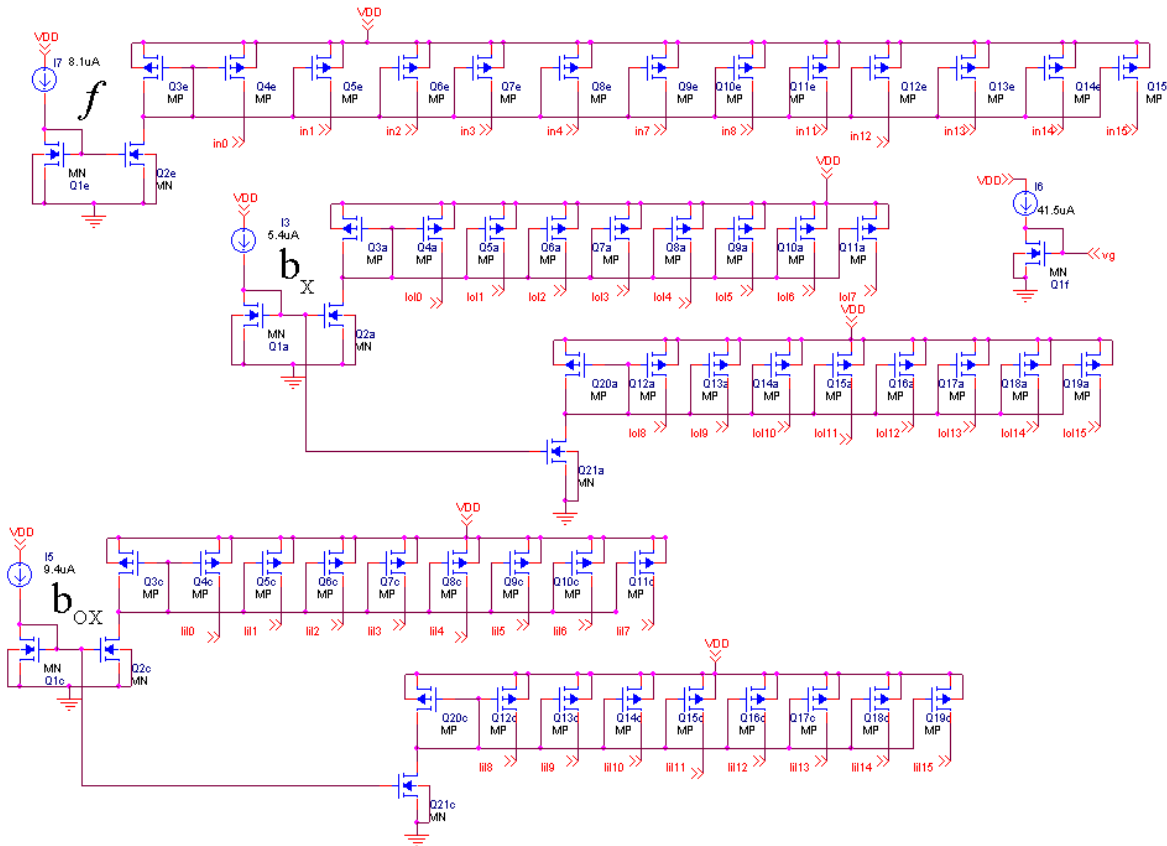


Figura 4.32 Esquema donde se muestran los copiadores de corriente para los coeficientes y el voltaje de referencia para los sumideros.

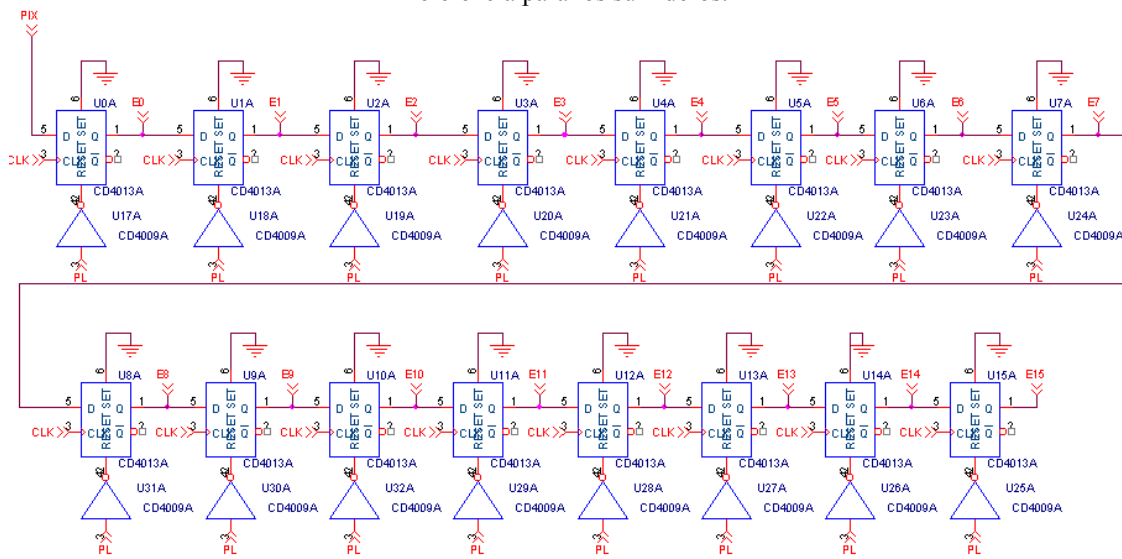


Figura 4.33 Registro de corrimiento implementado con flip-flops tipo D.

### 4.8.1 Removedor de Ruido

En las simulaciones para el removedor de ruido así como para todas las demás tareas, se utilizó la información del apéndice B con respecto a las mascarillas y sus coeficientes para programar las neuronas en la ejecución de cada tarea.

Después de considerar 150as para cargar la información correspondiente del patrón de entrada al registro de corrimiento (a una frecuencia de 167kHz). Las primeras simulaciones de la CNN contemplaron señales de control (reset, V3 y V4) con 50ns de ancho de pulso y 10ns de flancos de subida y bajada, ya que éstas representan las que se generan con el equipo disponible en el laboratorio. Se hizo un número considerable de simulaciones para diferentes patrones de entrada a fin de verificar el cumplimiento de todas las desigualdades de la tabla 1 del mismo apéndice. En la figura 4.34 se muestra uno de los patrones de entrada y su salida correspondiente. En la figura 4.35 se muestran los voltajes de salida de cada neurona sobre la capacitancia de 4pF. Si definimos el tiempo de convergencia como el intervalo de tiempo entre el momento en que las condiciones iniciales son cargadas y la CNN alcanza el estado estable, el circuito para esta tarea muestra un tiempo de convergencia de aproximadamente 100ns para el peor caso. El valor de la disipación de potencia de la red puede ser obtenida a través del archivo de salida de PSpice, el cual establece para este caso un consumo de 9.3mW por neurona.

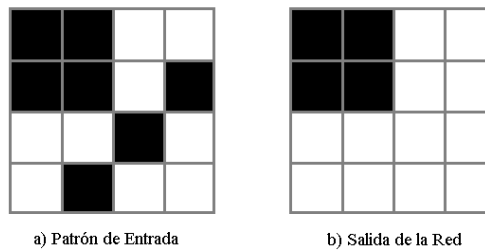


Figura 4.34 Removedor de ruido.

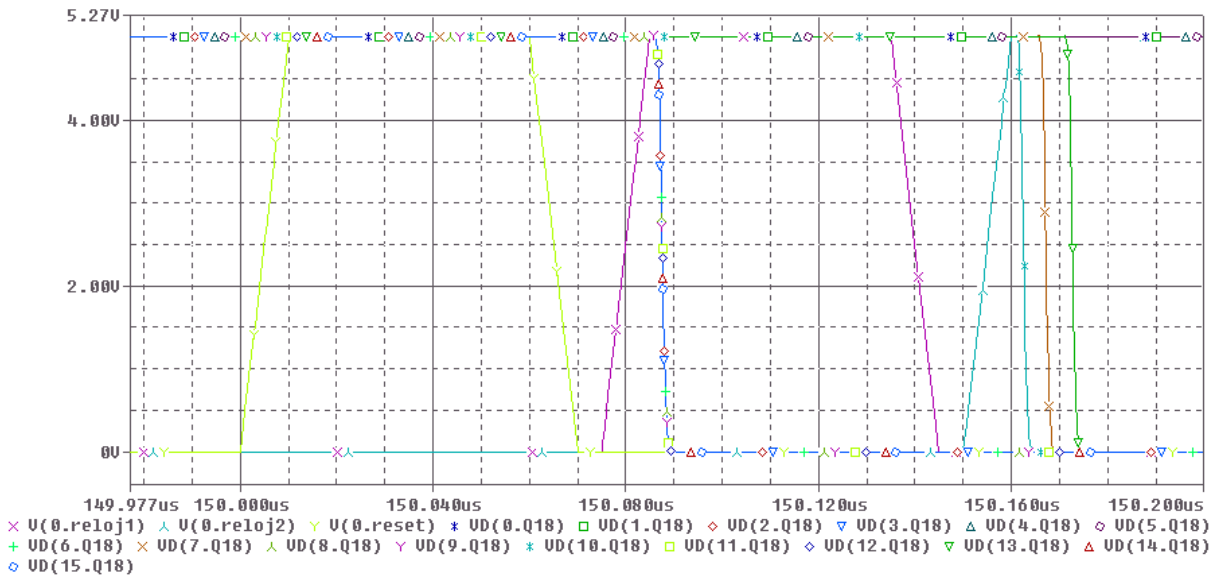


Figura 4.35 Resultados de la simulación en PSpice para el removedor de ruido

### 4.8.2 Detector de Conectividad Global

En este caso, también la CNN fue programada para resolver el problema del “laberinto” considerando las mismas señales de control, nuevamente se simularon una variedad de patrones de entrada de manera que se pudiera verificar el cumplimiento de todas las desigualdades definidas en la tabla 2 del apéndice B. En la figura 4.36 se muestra uno de los patrones que fueron considerados y la respectiva salida de la red. En la figura 4.37 se muestra el oscilograma correspondiente a este patrón en donde se observa un tiempo de convergencia de 162ns y una potencia disipada de 9.3mW por neurona.

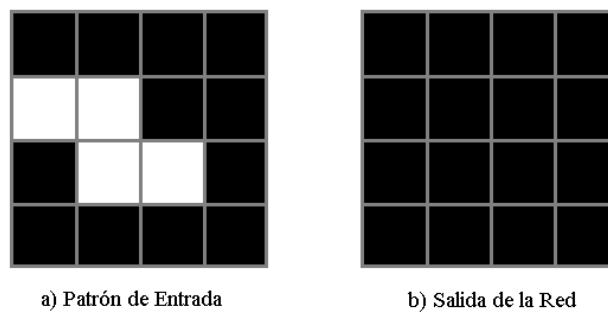


Figura 4.36 Detector de conectividad global.

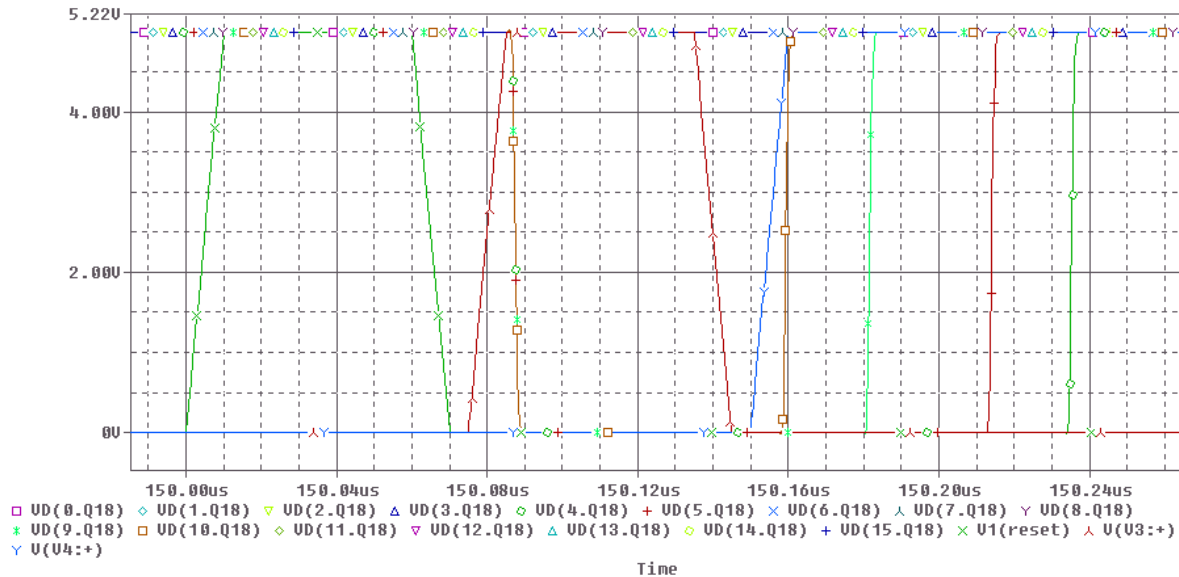


Figura 4.37 Resultados de PSpice para el detector de conectividad global.

El mismo circuito fue probado en una segunda simulación con señales de control de 30ns de ancho y flancos de 3ns, con el propósito de llevar a la CNN al límite de operación en frecuencia, para lo cual se obtuvo un tiempo de convergencia del circuito para el peor de los casos en 100ns.

### 4.8.3 Extractor de Bordes

También mediante la información del apéndice B, se hicieron simulaciones para un gran número de patrones de entrada con el objeto de verificar que la red cumpliera con la tarea del extractor de bordes, así como con cada una de las desigualdades de la tabla 3 del mismo apéndice.

En la figura 4.38 se muestra uno de los patrones introducidos a la CNN y su correspondiente salida. En la figura 4.39 se muestran las señales digitales del registro de corrimiento cuando la información del patrón de entrada es cargado, esto a una frecuencia de 167kHz. En la figura 4.40 se muestran los voltajes de salida de la CNN, donde se observa que el tiempo de convergencia de la red para el patrón de entrada, considerando señales de control de 50ns de ancho y 10ns de flancos, es de 88ns, con un consumo de potencia de 9.3mW por neurona.

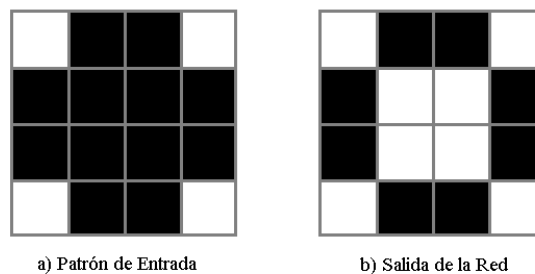


Figura 4.38 Extractor de bordes.

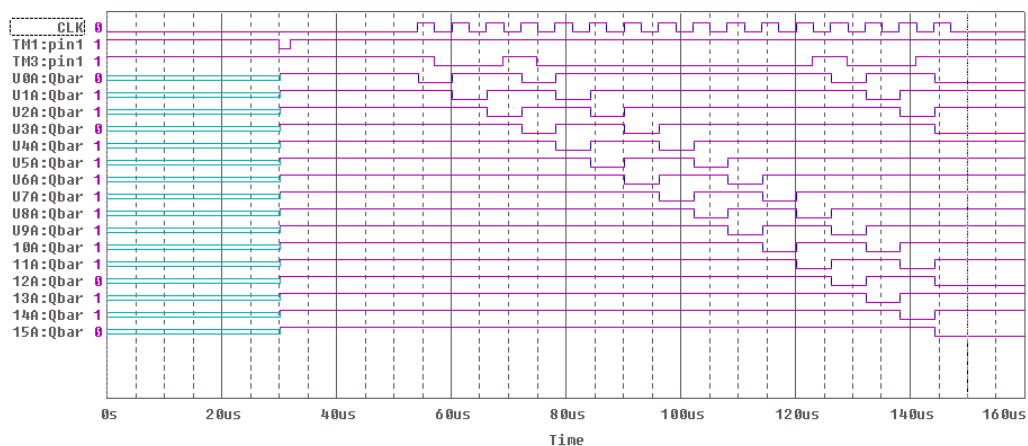
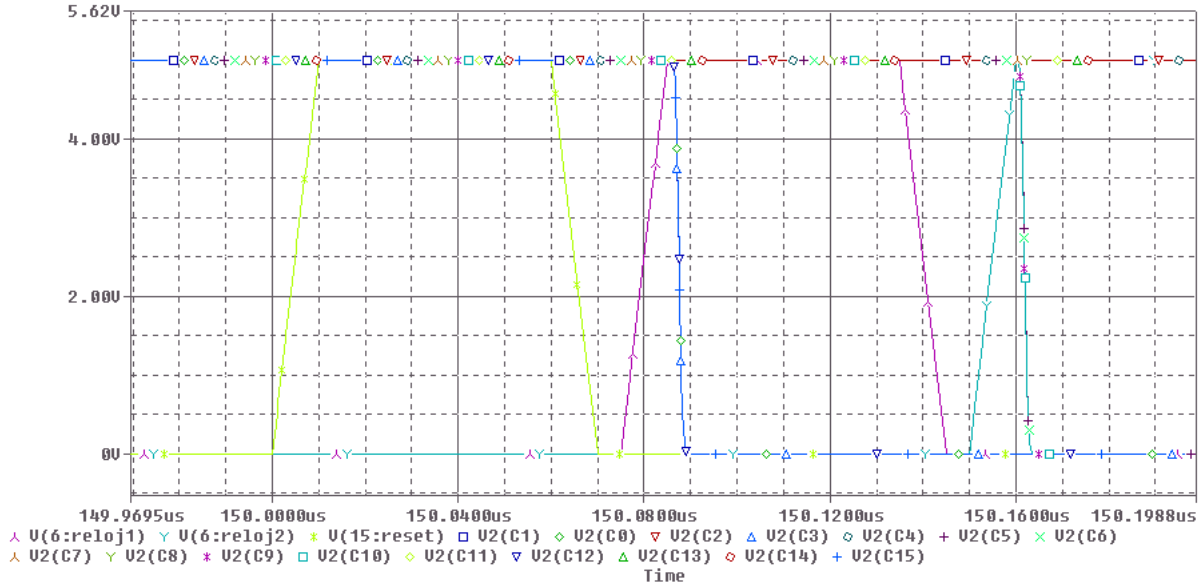


Figura 4.39 Evolución de la señal del patrón de entrada a través del registro de corrimiento.



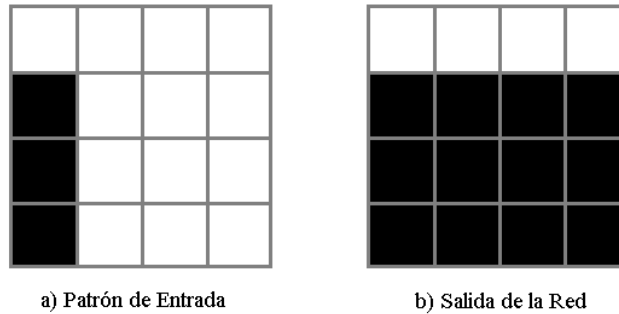


**Figura 4.40** Resultados de la simulación en PSpice para el extractor de bordes.

De igual forma que en la tarea anterior, el circuito fue probado con diferentes periodos de las señales de control. Para un ancho de pulso de 15ns y flancos de 3ns se observó el límite de operación de la red, la cual mostró un tiempo de convergencia de 30ns.

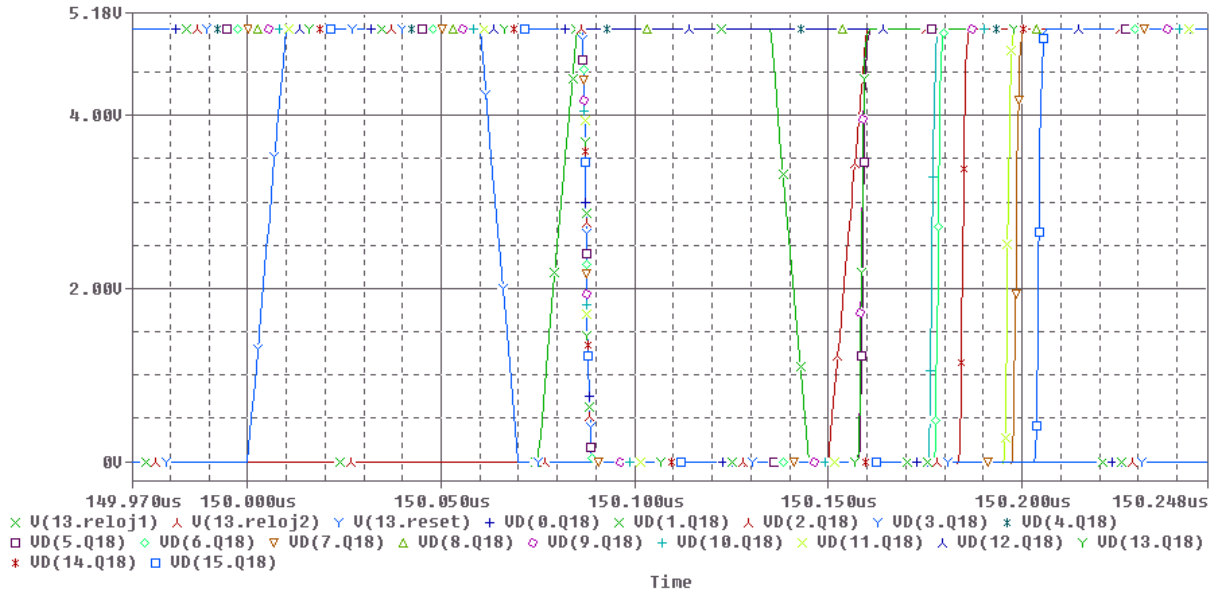
#### 4.8.4 Extractor de Sombras

Para esta tarea en particular, así como para las anteriores, la CNN procesó de manera correcta todos los patrones de entrada. Uno de estos patrones para el extractor de sombras y su respectiva salida se muestra en la figura 4.41. Los resultados de la simulación se muestran en el oscilograma de la figura 4.42. Vale la pena señalar el defasamiento irregular entre las señales de cada fila debido a la capacitancia incluida en el nodo sumatorio de las neuronas de prueba.



**Figura 4.41** Extractor de sombras.

En esta figura también se puede medir el tiempo de convergencia, el cual es de 132ns, con una disipación de potencia de 9.5mW por neurona para el caso de las señales de control de 50ns de ancho y 10ns en flancos de subida y bajada.



**Figura 4.42** Resultados de PSpice para el extractor de sombras.

Haciendo simulaciones con señales de control con pulsos de 15ns de ancho y flancos de 3ns, la CNN se llevó al límite en frecuencia lográndose un tiempo de convergencia de 83.5ns para el peor caso.

## 4.9 Simulación Eléctrica de una CNN de 1x8 para la Ejecución del Detector de Componentes Conectados (CCD)

### 4.9.0 Diagrama Esquemático del Circuito

La simulación para la tarea de procesamiento detector de componentes conectados, se hizo mediante un arreglo neuronal de 1x8 (figuras 4.43 y 4.44). Se emplearon neuronas con la arquitectura de la figura 4.28, con las mascarillas y coeficientes definidos en el apéndice B.

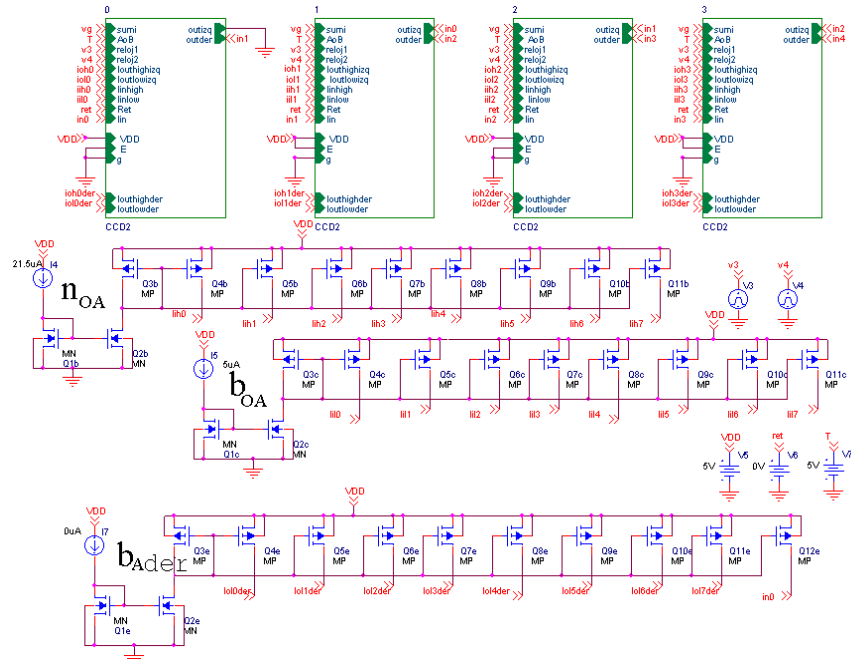


Figura 4.43 Cuatro neuronas del arreglo de 1x8, tres de los cinco copiadore de corriente para los coeficientes y las señales de voltaje.

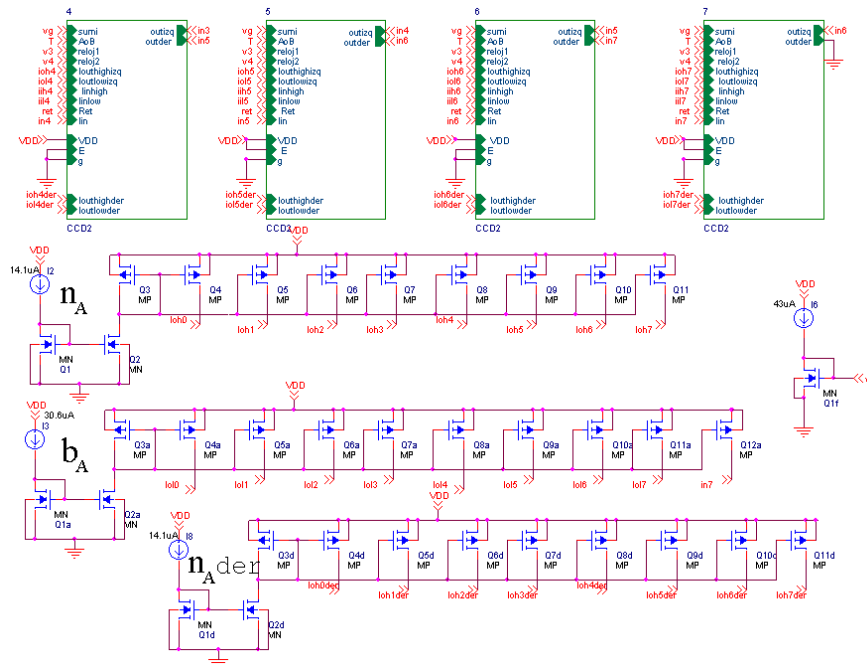


Figura 4.44 Últimas cuatro neuronas del arreglo de 1x8 y los restantes copiadore, así como el voltaje de referencia de los sumideros.

### 4.9.1 Resultados de la Simulación

Al igual que en la CNN de 4x4, se obtuvo el desempeño deseado en base a simulaciones extensivas. En esta estructura no se contempló la utilización de un registro de corrimiento para introducir los datos de entrada a las neuronas. En la figura 4.45 se muestra uno de los patrones de entrada y la salida correspondiente de la CNN.



Figura 4.45 Detector de componentes conectados.

El oscilograma de la señal de salida se muestra en la figura 4.46, en donde puede observarse que se utilizaron señales de control con pulsos de 25ns de ancho y 3ns en flancos de subida y bajada. El tiempo de convergencia de la CNN es de 90ns para el peor caso, con una potencia disipada de 0.368 mW por neurona.

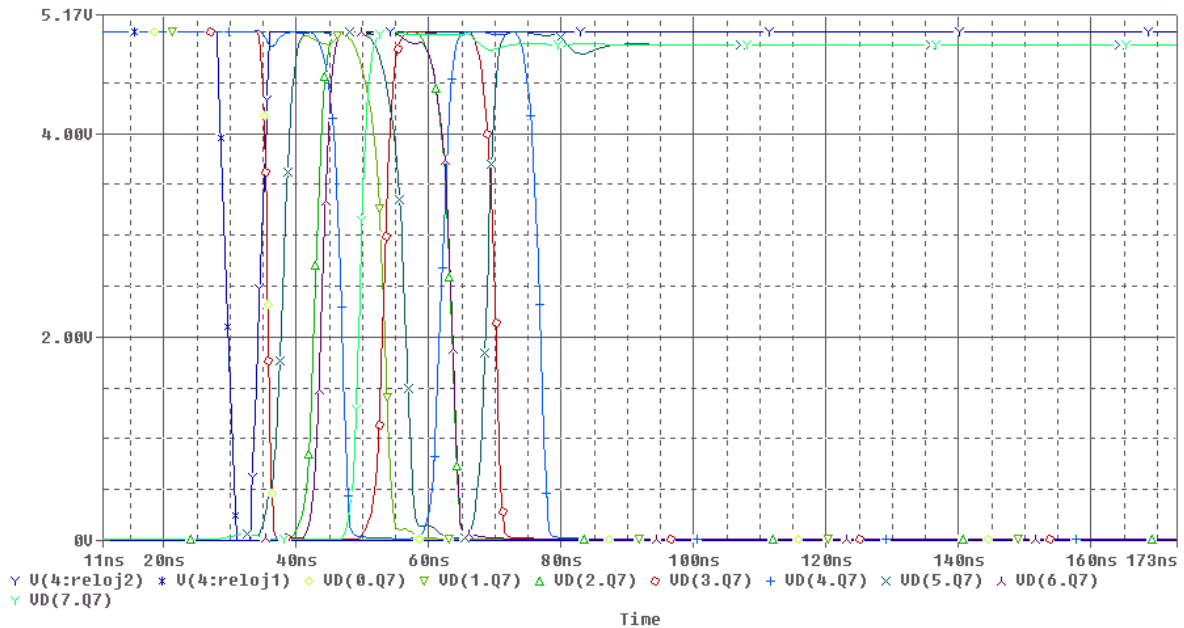


Figura 4.46 Resultados en PSpice para el CCD.

## 4.10 Desacoplamiento entre Transistores

Aún cuando las compañías dedicadas a la fabricación de circuitos integrados poseen procesos estables y bien controlados, en ellos siempre existirán variaciones aleatorias en los parámetros tecnológicos de sus dispositivos. Esta condición provoca que un diseño que contempla el uso de dos o más transistores de iguales dimensiones, bajo las mismas condiciones de polarización, manifiesten diferencias en sus corrientes de drenador. Este fenómeno se conoce comúnmente como desacoplamiento entre transistores (*mismatch*) y es un factor que limita el desempeño de diseños tanto analógicos como digitales. Consecuentemente, es necesario hacer estimaciones estadísticas de los parámetros tecnológicos y su influencia en el circuito con el fin de obtener diseños robustos que aseguren la funcionalidad y el buen desempeño en la implementación.

En este sentido, el modelo de Pelgrom *et al.* [15], formula las características estadísticas del desacoplamiento en función del área y distancia de los dispositivos, del cual se obtiene un número reducido de parámetros que describen de una manera general las desviaciones que pudiesen tener las corrientes de drenador de los transistores. Aunque existen otros modelos que describen estas desviaciones de maneras mucho más precisas [21], en ellos es necesario el empleo de técnicas sofisticadas de caracterización y medición; de tal forma que el modelo propuesto por Pelgrom es uno de los más ampliamente utilizados debido a su sencillez y aunque es impreciso, permite al diseñador tener una idea general de los grados de libertad disponibles para atenuar los efectos del desacoplamiento.

### 4.10.1 Modelo de Pelgrom

Haciendo un análisis de acuerdo a la teoría en [15], las ecuaciones que describen la desviación estándar para  $V_{TO}$  (voltaje de umbral sin polarización del sustrato) y  $\gamma$  (parámetro de efecto de cuerpo) están dadas por:

$$\sigma^2(V_{TO}) = \frac{A_{V_{TO}}^2}{WL} + S_{V_{TO}}^2 D^2 \quad y$$

$$\sigma^2(\gamma) = \frac{A_{\gamma}^2}{WL} + S_{\gamma}^2 D^2$$

donde  $W$ ,  $L$  son el ancho y longitud de canal,  $D$  representa la distancia entre los centros geométricos de los transistores y tanto  $A_x$  como  $S_x$ , representan coeficientes de ajuste que son dependientes del proceso. De tal forma que el desacoplamiento depende sustancialmente de las dimensiones y la separación entre dispositivos. De igual manera puede evaluarse la desviación del parámetro de transconductancia  $V$  mediante:

$$\sigma^2(\beta) = \frac{A_{\beta}^2}{WL} + S_{\beta}^2 D^2$$

De acuerdo a los experimentos de Pelgrom, los parámetros  $V_{TO}$ ,  $\beta$  y  $V$  muestran una notable dependencia lineal al factor  $1/\sqrt{WL}$ . Y, en cuanto al desacoplamiento debido a la distancia entre dispositivos, se observa que éste es sólo importante para dispositivos de área grande, esto es, transistores con área de compuerta mayores a  $100\mu\text{m}^2$ .

Considerando lo anterior, la desviación de la corriente de drenador entre los transistores puede ser aproximada mediante la desviación estándar de estos parámetros a través de:

$$\frac{\sigma^2(I_D)}{I_D} = \frac{4\sigma^2(V_{TO})}{(V_{GS} - V_{TO})^2} + \frac{\sigma^2(\beta)}{\beta^2}$$

en donde se observa que la desviación de la corriente depende fuertemente del voltaje compuerta-fuente y por lo tanto puede ser minimizada empleando voltajes de compuerta elevados con respecto a  $V_{TO}$ .

La idea general del desacoplamiento puede ser evaluada mediante la simulación de análisis de sensibilidad y Monte Carlo en PSpice, en donde los parámetros críticos son modificados de manera aleatoria siguiendo una distribución probabilística específica.

#### 4.10.1 Simulación de Análisis Monte Carlo

En este trabajo, con el propósito de obtener una idea general de lo susceptible del diseño al desacoplamiento entre transistores, especialmente en los espejos de corriente, se realizó un análisis Monte Carlo de 25 pruebas tomando en cuenta las desviaciones estándar reportadas de los parámetros  $V_{TO}$  y  $V$  en [15] para un proceso de 25nm de espesor de óxido y una longitud mínima de canal de  $1.6\mu\text{m}$ . Dicho análisis fue realizado en la CNN considerando la tarea del removedor de ruido en donde el patrón de entrada es el más representativo para poner a prueba las desigualdades más sensibles a los errores del desacoplamiento.

En la figura 4.47 se muestra el patrón de entrada y la salida que debe presentar la red, mientras que en la figura 4.48 se muestran los resultados del análisis Monte Carlo para las salidas de todas la neuronas que tienen una entrada blanca. Se puede apreciar que la neurona 8 en sólo una de las 25 pruebas no hace la correspondiente transición a un estado negro.

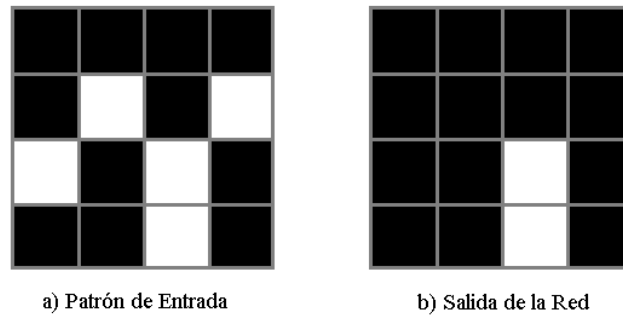


Figura 4.47 Patrón de entrada para el análisis Monte Carlo.

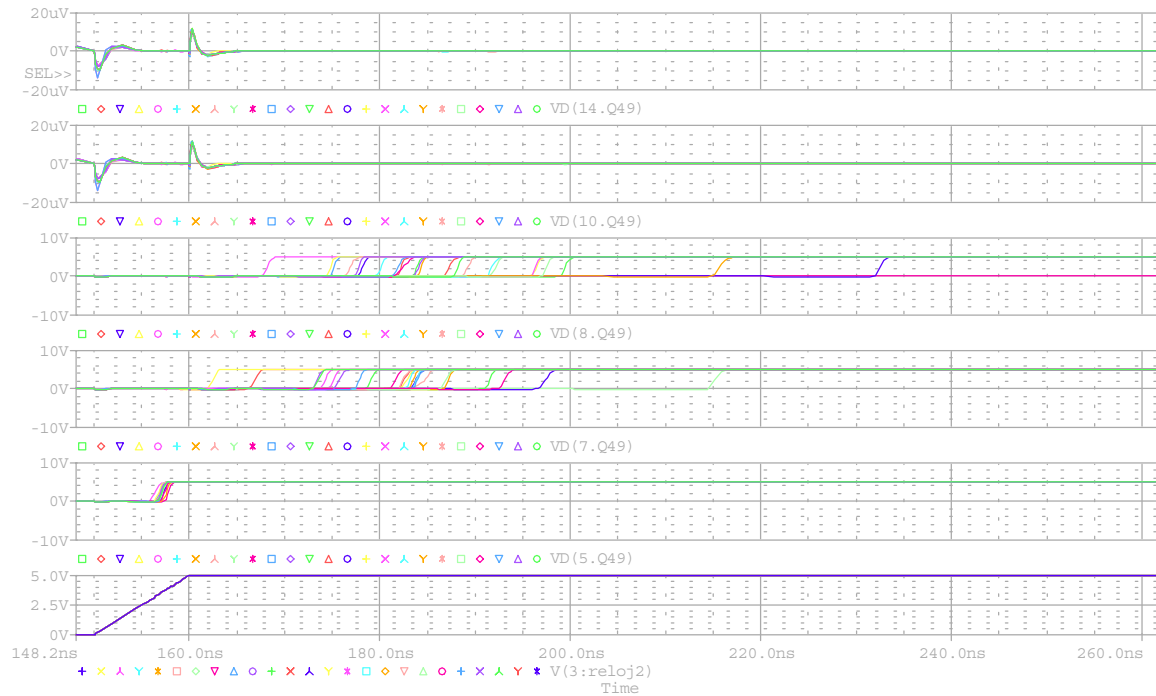


Figura 4.48 Resultados de PSpice para el análisis Monte Carlo con 25 pruebas.

Aunque estos resultados son una aproximación del desempeño del diseño, se tiene una medida probabilística aceptable que indica que el circuito funciona correctamente.

## Referencias

- [1] ACE4K: System-Level Measurement versión 2 Instituto de Microelectrónica de Sevilla –Centro Nacional de Microelectrónica.
- [2] F. Lobato-Lopez, J. Silva-Martinez, E. Sanchez-Sinencio, " A Very Fast CMOS Artificial Cellular Neural Network ," IEEE Int. Symposium on Circuits and Systems, Atlanta, Georgia, May 1996.
- [3] Ari Paasio, Asko Kananen, Kari Halonen, "Very fast and compact fixed template CNN realizations for B/W processing", IEEE International Symposium on Circuits and Systems (ISCAS'99), pp. 595-598.
- [4] M. Anguita, F.J. Pelayo, I. Rojas, A. Prieto . "Area Efficient Implementation of Fixed-Template CNNs", in IEEE Transactions on Circuits and Systems Part.I. vol.45, no.9, pp. 968-973, September 1998.
- [5] J.M. Cruz, L.O. Chua, " A 16x16 Cellular Neural Network Universal Chip: The first Complete Single-Chip Dynamic Computer Array with Distributed Memory and Gray-Scale Input-Output" Analog Integrated Circuits and Signal Processing, 15,227-237, 1998.
- [6] C. Huat and B. A. Wooley, "A 128x128 pixel standard-CMOS Image Sensor with Electronic Shutter" IEEE Journal of Solid State Circuits, vol. 31, no. 12, December 1996.
- [7] S. Mendis, Sabrina E. Kemeny, and Eric R. Fossum, " CMOS Active Pixel Image Sensor" IEEE Trans. on Electron Devices, vol. 41, no. 3, March 1994.
- [8] T. Roska, Angel Rodriguez-Vazquez, "Toward the Visual Microprocessor" John Wiley Sons 2001.
- [9] M. Anguita, F.J. Pelayo, E. Ros, D. Palomar and A. Prieto, " Focal-Plane and Multiple Chip VLSI Approaches to CNNs" Analog Integrated Circuits and Signal Processing, 15, 263-275, 1998.
- [10] S. Espejo, A. Rodríguez-Vázquez, R. Domínguez-Castro, J.L. Huertas, E. Sánchez Sinencio, "Smart-Pixel Cellular Neural Networks in Analog Current-mode CMOS Technology" IEEE Journal of Solid State Circuits, vol. 29, no. 8, August 1994.
- [11] R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. A. Carmona, P. Földesy, A. Zarándy, P. Szolgay, T. Szirányi and T. Roska, "A 0.8-um CMOS Two-Dimensional Programmable Mixed-Signal Focal-Plane Array Processor with On-Chip Binary Imaging and Instruction Storage", IEEE Journal of Solid-State Circuits, Vol. 32, No. 7, pp. 1013-1025, July 1997.
- [12] Servando Espejo, Rafael Dominguez-Castro, Gustavo Liñan, Angel Rodriguez-Vazquez, "A 64 x 64 CNN Universal Chip with Analog and Digital I/O" IEEE International Conference on Electronics Circuits and Systems (ICECS/98) pp. 203-206 Lisboa Sept. 1998.
- [13] M. Balsi, I. Ciancaglioni and V. Cimagalli, "Optoelectronic Cellular Neural Network Based on Amorphous Silicon Thin Technology" Proc. 3<sup>rd</sup> IEEE Int. Workshop on Cellular Neural Networks and Their Applications, Rome, 1994, pp. 399-403.
- [14] A. Mortara and E. A. Vittoz, " A Communication Architecture Tailored for Analog VLSI Artificial Neural Networks: Intrinsic Performance and Limitations" IEEE Trans. on Neural Networks, vol, 5, no. 3, May 1994
- [15] Marcel J. M. Pelgrom, J. Duinmaijer and A. P. G. Welbers, "Matching Properties of MOS Transistors" IEE Journal of Solid State Circuits, vol. 24, no. 5, October 1989.
- [16] C. Toumazou, F.J. Lidgley and D.G. Haigh, "Analogue IC Design: the current mode approach" IEE Circuits and Systems series 2, 1990.
- [17] Iluminada Baturone Castillo, "Implementación VLSI de Controladores Difusos" Tesis Doctoral, Instituto de Microelectrónica de Sevilla, CNM, Sevilla Dic. 1996.
- [18] H. Traff, " Novel Approach to High Speed CMOS Current Comparator" Electronic Letters, 30 January 1992, vol. 28, no. 3, pp. 310-312.
- [19] R.J. Baker, H.W. Li, D. E. Boyce, "CMOS Circuit Design, Layout, and Simulation" IEEE Press Series on Microelectronic Systems, 1998.
- [20] A. Rueda, J.L. Huertas, " Testability in Analogue Cellular Neural Networks" Proc. Of the 1990 International Workshop on Cellular neural Networks and Their Applications CNNA'90, Budapest.
- [21] C. Michael, M. Ismail, "Statistical Modeling of Device Mismatch for Analog MOS Integrated Circuits" IEEE Journal of Solid State Circuits, vol. 27, no. 2, February 1992.



# CAPÍTULO 5













DISEÑO GEOMÉTRICO  
DE LOS CIRCUITOS

## 5.0 Introducción

Después de realizado el diseño eléctrico de la CNN, se lleva a cabo el diseño geométrico (*Layout*) de todo el circuito, con lo cual se genera la información tecnológica para su fabricación. Este diseño consiste en determinar de manera geométrica las dimensiones e interconexiones, la ubicación y orientación que todos los dispositivos que conforman el circuito tendrán sobre el silicio.

El diseño geométrico de la CNN de 4x4, fue desarrollado con el programa L-Edit de la compañía Tanner Research Inc., empleando las reglas de diseño proporcionadas por MOSIS para una tecnología de 1.2 $\mu$ m, pozo N, dos niveles de polisilicio y dos de metales. Este programa permite obtener el diseño en un archivo de formato estándar CIF (*CalTech Intermediate Form*), el cual sirve como medio para describir el diseño completo de una manera universal.

Dentro de la representación gráfica del diseño, diferentes colores son asociados a los distintos materiales que constituyen los dispositivos. La representaciones de éstos en código de colores es:

 Zona Activa	 Contacto
 Poly 1	 P Select
 Poly 2	 N Select
 Metal 1	 Pozo N
 Metal 2	 Vía

Además, el programa L-Edit incluye bibliotecas de celdas prediseñadas (*Standard Cells*) tanto analógicas como digitales, que se encuentran a disposición del diseñador.

## 5.1 Celdas Básicas

En base a las figuras 4.30, 31, 32 y 33 es posible establecer tres celdas fundamentales que deben ser diseñadas: la *neurona CNN*, los *copiadores de corriente* destinados para los coeficientes de las mascarillas y los *flip-flops tipo D* que conforman el registro de corrimiento.

### 5.1.0 Neurona CNN

Tomando en cuenta el circuito de una sola neurona de la figura 4.27, que constituye el arreglo de 4x4, se llevó a cabo el diseño geométrico de la misma. En el diseño se debía tener presente la distribución espacial de sus componentes de manera que fuera lo más simétrica posible y que los conductores de entrada y salida facilitaran la conexión entre neuronas. El diagrama para esta celda se muestra en la figura 5.0, el cual tiene un área aproximada de  $33905\text{am}^2$  ( $204\text{am} \times 166\text{am}$ ) permitiendo con esto una densidad de integración de  $29.5$  neuronas/ $\text{mm}^2$ .

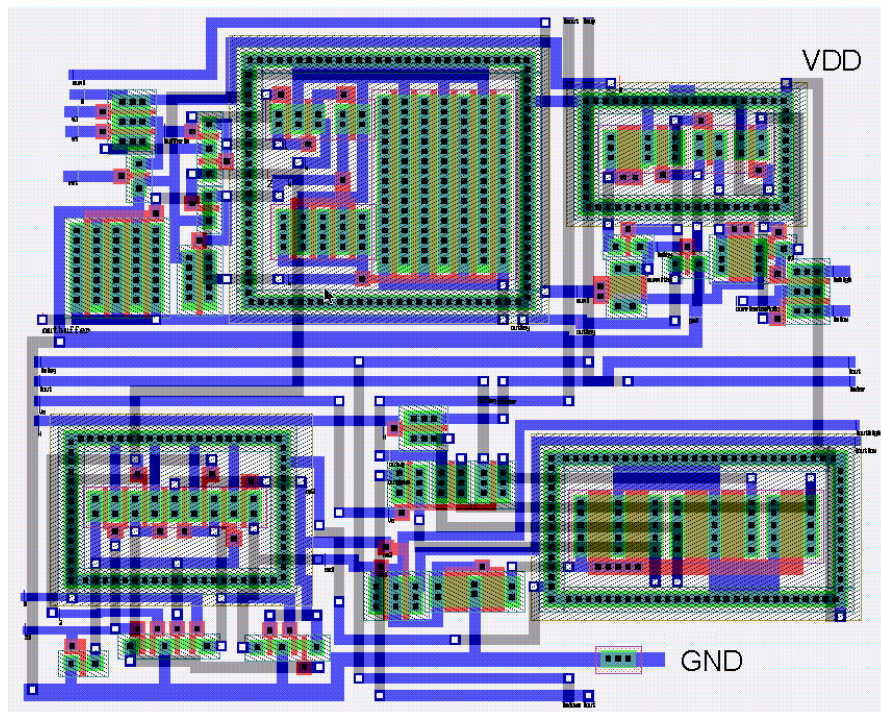
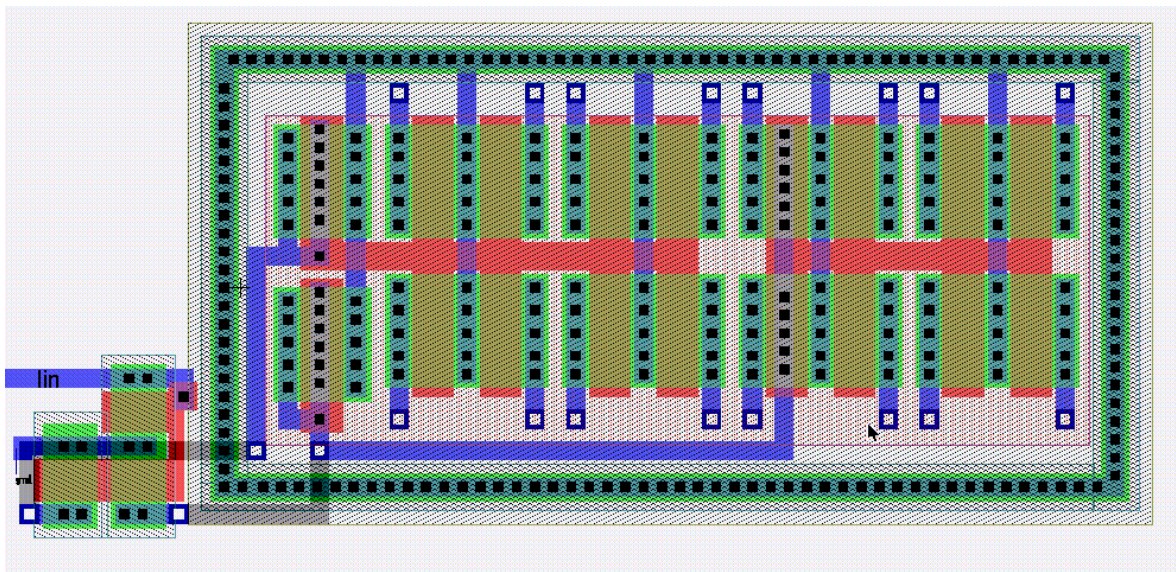


Figura 5.0 Diseño geométrico de una sola neurona.

### 5.1.1 Copiadores de Corriente

Las celdas de los espejos de corriente también se diseñaron tomando en cuenta las consideraciones de la sección 4.5.1.0. En cada circuito copiadore, es necesario que el número de copias corresponda al número total de neuronas con excepción de aquellas destinadas para la implementación del elemento  $f$ . De tal forma que el diseño geométrico de uno de los copiadores de la figura 4.31 ó 4.32 se muestra en la figura 5.1, donde es posible apreciar la terminal de corriente de entrada  $I_{in}$  y donde cada una de las “vias” en los transistores PMOS, corresponde al drenador de los transistores “espejos”.



**Figura 5.1** Diseño geométrico de un sólo copiadore con 16 espejos de corriente.

En este diseño particular, ya que se trata de transistores cuya área de compuerta es menor a  $100\text{am}^2$ , el factor de la distancia entre los dispositivos como elemento de desacoplamiento es despreciable [1] y por lo tanto el diseño no toma en cuenta una topología de centroide común u alguna otra con el objeto de disminuir tales efectos.



### 5.1.2 Flip-Flops tipo D

El desarrollo del diseño del registro de corrimiento de la figura 4.33 se llevó a cabo mediante el uso de una celda prediseñada de la biblioteca SCMOSLib ver. 2.0 del programa L-Edit, la cual es un flip-flop tipo D con *clear* (DFFC). El símbolo y su correspondiente tabla de verdad se muestra en la figura 5.2. El diseño geométrico de un sólo flip-flop se muestra en la figura 5.3; está compuesto por 29 transistores y tiene dimensiones de 79.2am x 44.4am.

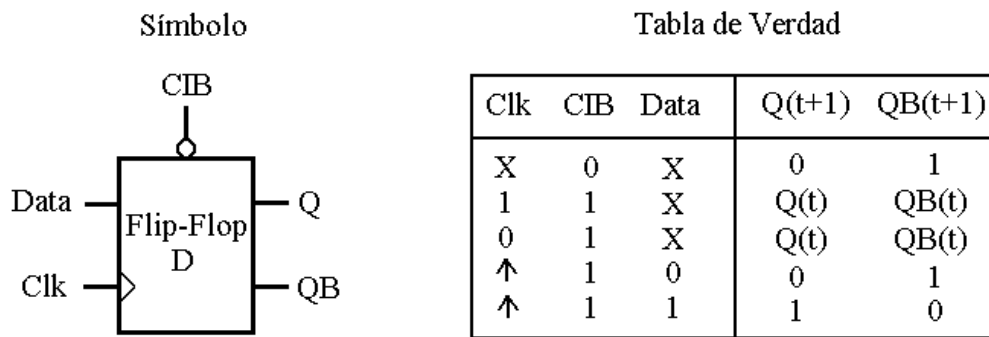


Figura 5.2 Flip-flop con *clear* disponible en la librería y su correspondiente tabla de verdad.

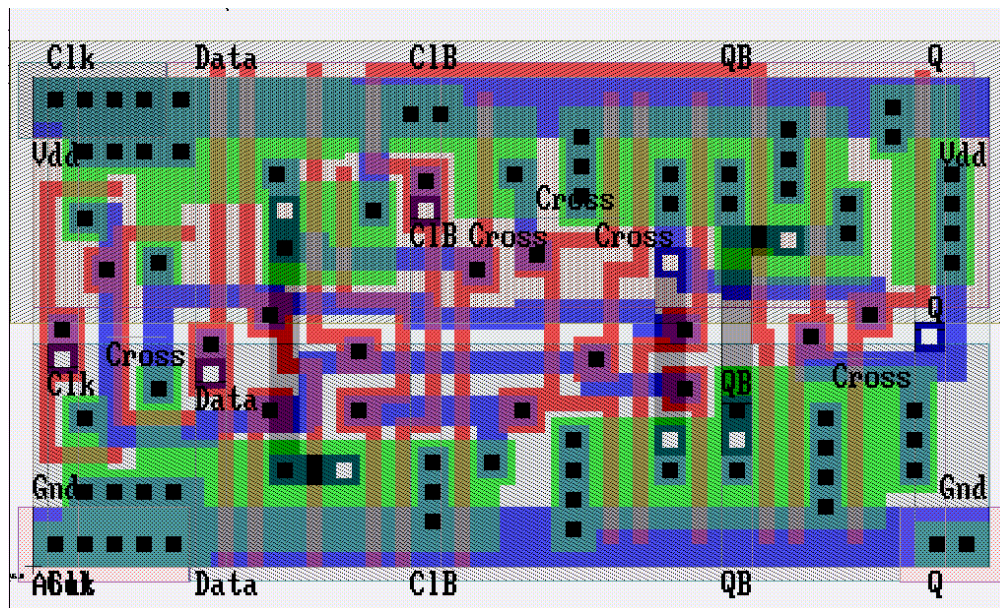


Figura 5.3 Diseño geométrico del flip-flop con *clear*.

## 5.2 Circuito Integrado Completo

Utilizando las celdas anteriores, se elaboró el circuito integrado completo, el cual está formado como se muestra en las figuras 4.30, 31, 32 y 33 por 16 neuronas CNN, 5 copiadore de corriente, de los cuales 4 poseen 16 espejos de corriente y uno sólo 12, y 16 flip-flops tipo D para el registro de corrimiento. Otras celdas que se consideraron de la biblioteca SCMOSSLib ver 2.0 fueron las referentes a las terminales (*Pads*) que conectan el dado de silicio con las terminales externas (Pines) del circuito integrado. En este caso, se utilizaron cinco terminales diferentes:

- 1.- PadIOnoESD.
- 2.- PadInC.
- 3.- Pad\_noconect.
- 4.- PadVDD.
- 5.-PadGND.

La primera es una terminal bidireccional que no cuenta con circuitos de protección antiestática (ESD) y es utilizada para las salidas y entradas analógicas del circuito integrado. PadInC es una terminal de entrada la cual esta provista de protección antiestática (ESD), que emplea una resistencia y un buffer compuesto de dos inversores digitales. Esta es empleada en aquellas terminales destinadas a las entrada digitales del circuito integrado. Pad\_noconect es una terminal que no tiene contacto con el circuito y es empleada en la única terminal del sistema que no tiene función. Y por último, PadVDD y PadGND que son las terminales empleadas para la alimentación.

El circuito completo se muestra en la figura 5.4, el cual incluye las terminales de entrada y salida. Por último, en la tabla 5.1 se muestra el número de transistores utilizados por celda y el total de ellos en el circuito integrado.

Tabla 5.1

Celda	No. de transistores	No. de celdas	Total de transistores
Neurona CNN	53	16	848
Copiadores de corriente	22	5	110
flip-flop con <i>clear</i>	29	16	464
			Total 1422

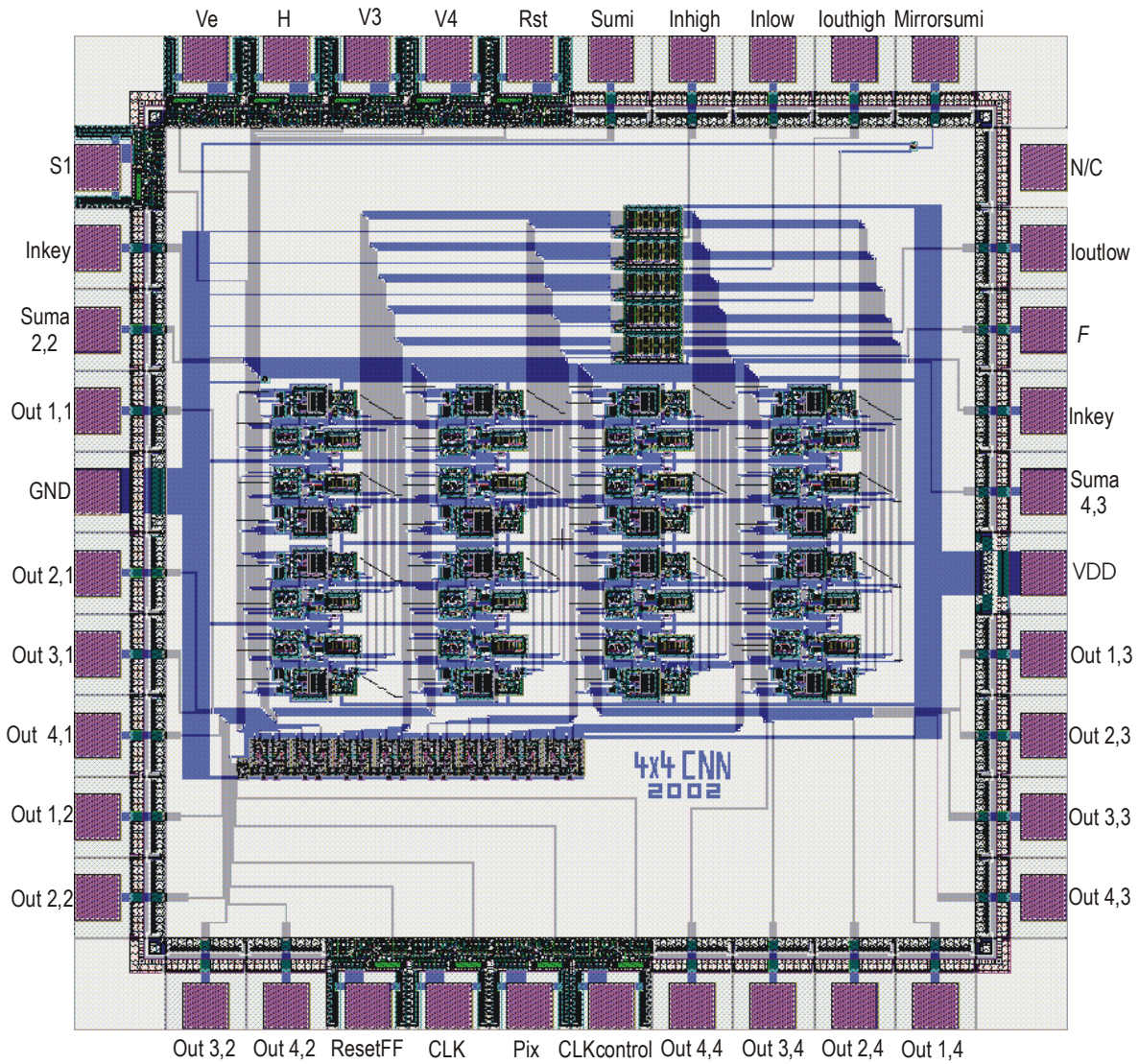


Figura 5.4 Circuito integrado completo.

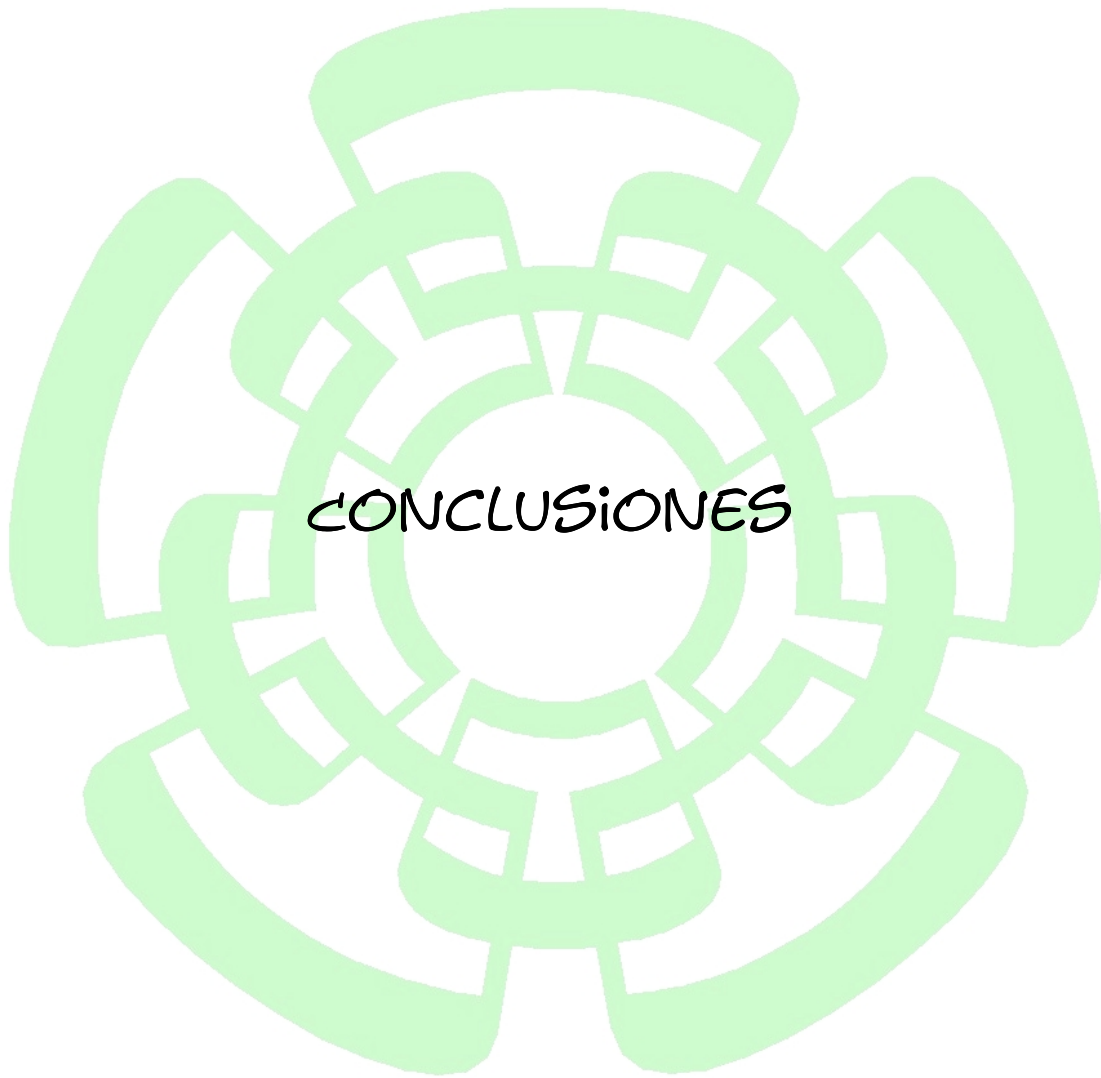
## Referencias

- [1] Marcel J. M. Pelgrom, J. Duinmaijer and A. P. G. Welbers, "Matching Properties of MOS Transistors" IEE Journal of Solid State Circuits, vol. 24, no. 5, October 1989.





# CAPÍTULO 6



## 6.0 Conclusiones

Dada la importancia de la electrónica en el desarrollo de sistemas complejos de visión artificial, es importante el estudio de nuevas propuestas de procesamiento que permitan la ejecución de tareas básicas de visión en tiempo real. Es así, que en este trabajo se evaluaron diferentes arquitecturas de la CNN existentes y se propone un nuevo modelo de CNN para el procesamiento de imágenes binarias.

Para obtener un panorama general del trabajo realizado en esta tesis, considerando los resultados obtenidos en las simulaciones del circuito integrado diseñado, tanto a nivel funcional como a nivel eléctrico, se puede realizar una comparación con otras propuestas similares realizadas por otros investigadores, y así poder evaluar las ventajas y desventajas que presenta el modelo de CNN propuesto.

En la tabla 6.1 se muestra un grupo de sistemas integrados CNN programables representativos de la evolución que ha tenido esta estructura en los últimos años. Resulta difícil realizar una comparación eficaz entre todas las propuestas, ya que en cada trabajo particular se reportan parámetros que son propios de cada propuesta, tanto de tipo digital como analógico. Sin embargo, como se puede observar, el diseño y la arquitectura de esta tesis es competitiva. Por ejemplo, en las tareas de extracción de bordes y detección de componentes conectados, el tiempo de convergencia obtenido es el menor para una CNN programable.

Una de las ventajas más significativas del modelo propuesto en este trabajo de tesis con respecto a otros es que su salida corresponde a niveles lógicos CMOS, lo cual permite acoplar de manera directa el circuito integrado con otros sistemas convencionales digitales, sin la necesidad de incorporar interfaces adicionales de conversión analógico-digital.

El número de transistores por neurona, así como la densidad de neuronas por milímetro cuadrado obtenidas en este trabajo, muestran cifras muy alentadoras para implementaciones futuras en tecnologías submicrométricas. Si consideramos que la densidad de integración es proporcional a  $\lambda^{-2}$ , (donde  $\lambda$  es el factor de escalamiento) en una tecnología con longitud mínima de canal de 0.6 $\mu$ m se podría obtener una densidad de 120 neuronas por milímetro cuadrado aproximadamente y, probablemente, aún mayor si tomamos en cuenta un tercer o cuarto nivel de metalización, los cuales están disponibles regularmente sobre estas tecnologías. Estos ayudarían notablemente a disminuir gran parte del área ocupada por las interconexiones entre neuronas, de programación y salida de la red.

Un punto importante del diseño de la CNN aquí propuesto, es que puede aún ser optimizado para disminuir el consumo de potencia, lo que implicaría trasladar las ideas fundamentales hacia un nuevo circuito integrado que fuese diseñado empleando técnicas de baja potencia y bajo voltaje.

TABLA 6.1 ALGUNAS IMPLEMENTACIONES DE REDES NEURONALES CELULARES PROGRAMABLES

	Dimensiones y Tipo de CNN	Tecnología	No. de Transistores	Tiempo de Convergencia	Densidad de Integración	Consumo de Potencia
[1] H. Harrer 1992	4x4 CNN-DT	1.5am	106/Neurona	Máx. frec. de operación 3.3Mhz, 2as	12 Neuronas/mm <sup>2</sup>	0.9mW/Neurona
[2] P.Kinget 1995	4x4 CNN	2.4am	-	, 50as	3.8 Neuronas/mm <sup>2</sup>	0.4mW/Neurona @ ± 5V
[3] K. Halonen 1990	4x4 CNN Logic	2.0am	519/Neurona	, 3as	1 Neurona/mm <sup>2</sup>	-
[4] M. Salerno 1998	6x6 CNN	1.5am	-	, 30as	-	13.36mW/IC @5V
[5] M. Anguita 1998	8x8 CNN ISR	1.2am	-	6as No Propagación 150as Propagación	10.5 Neuronas/mm <sup>2</sup>	12aW/Neurona
[6] G.F. Dalla Betta 1993	10x10 CNN	1.5am	8000/IC	Edge, 5as Noise, 2as Shadow, 15as CCD, 10as	-	60mW/IC @ ± 5V
[7] C. Ravezzi 1998	8x8 CNN FSR	1.2am	-	Noise, 2as Edge, 2.2as	5.6 Neuronas/mm <sup>2</sup>	Noise 5mW/IC @ ± 2.5V
<b>J. E. Molinar 2002</b>	<b>Sim 4x4 CNN</b>	<b>1.2 m</b>	<b>53/Neurona</b>	<b>Noise 100ns Edge 30ns Shadow 80ns GCD 100ns CCD 1x8 90ns</b>	<b>29.5 Neuronas/mm<sup>2</sup></b>	<b>9.2mW/Neurona @5V</b>
[8] A. Paasio 1997	16x16 CNN PHS – UM	0.8am	-	CCD 1x5, 200ns	74 Neuronas/mm <sup>2</sup>	-
[9] A. Paasio 2001	Sim 1x4 CNN PHS	0.5am	-	CCD 1x4, 100ns	-	@ 3V
[10] A. Paasio 1998	48x48 CNN PHS-UM	0.5am	220,200/IC	, 250ns	295 Neuronas/mm <sup>2</sup>	100-200mW/IC @2.8V
[11] J.M Cruz 1998	16x16 CNN gray scale	0.8am	-	, 250ns	16.5 Neuronas/mm <sup>2</sup>	300mW/IC @5V
[12] R. Domínguez Castro 1998	20x22 CNN-UM	0.8am	-	, 2as	27.5 Neuronas/mm <sup>2</sup>	1.1 W @ 5V
[12],[13] Gustavo Liñan 1999	64x64 CNN-UM gray scale A/D I/O	0.5am	, 1'000,000/IC 172/Neurona	, 5as	82 Neuronas/mm <sup>2</sup>	1.2W/IC @3.3V
Angel Rodriguez Vazquez 2001	128x128 CNN-UM gray scale A/D I/O	0.35am 5M 1P	, 3'748,170/IC 198/Neurona	, 4as	180 Neuronas/mm <sup>2</sup>	< 4W @3.3V

## 6.1 Trabajo Futuro

Una de las desventajas importantes de la CNN para imágenes binarias, es que se encuentran confinadas a una resolución de un bit, esto es, no tienen la capacidad de producir señales en escala de grises, lo cual las limita todavía a un tipo de procesamiento primitivo si se toma en cuenta la idea de procesar imágenes a colores en tiempo real. La propuesta hecha en [13] es quizá hoy en día, de las únicas que poseen sensores ópticos distribuidos *on-chip* y opera con entradas y salidas en escala de grises con una resolución equivalente de 7 bits. Esta característica le da la posibilidad, mediante la descomposición tricromática de poder responder al espectro visible y en un futuro cercano ser una de las primeras arquitecturas CNN en un sólo circuito integrado que procese imágenes a color. Sin embargo, una de sus notables desventajas es que la mayoría de las mascarillas para escala de grises son del tipo no-lineal, por lo que es necesario recurrir a técnicas de descomposición [14] que hagan posible realizar tales tareas mediante algoritmos que utilizan exclusivamente mascarillas lineales. Además, la mayoría de los métodos de entrenamiento para esta red presentan muchas dificultades para generar mascarillas en escala de grises, por lo cual la mayoría de ellas son obtenidas heurísticamente.

Si por un lado, las CNN para imágenes binarias son las más veloces (tabla 6.1), y además se toma en cuenta que la gran mayoría de mascarillas existentes son para imágenes binarias, sería deseable explotar tales condiciones de este tipo de redes y encontrar alguna técnica que permitiera extender las propiedades de procesamiento de estas últimas hacia el caso de imágenes en escala de grises. Partiendo del modelo de CNN propuesto en esta tesis, podrían incluirse las características eléctricas de los transistores de compuerta flotante para lograr este objetivo.

Considerando la arquitectura de la neurona de la figura 4.25, en la parte donde se establecen las condiciones iniciales (figura 6.0), el inversor conformado por Q7-Q8 podría ser sustituido por un dispositivo de compuerta flotante como el de la figura 6.1 (a), el cual, de acuerdo con la teoría expuesta por T. Shibata *et al.* [15], podría ser un elemento que discriminase niveles de voltaje entre 0 y 5V que constituyen el valor de un píxel gris de entrada, de tal forma que, mediante los voltajes externos P1 y P2, se le podría indicar a la red qué partes de la imagen reconozca como negro o blanco. De esta manera una imagen de entrada en niveles de gris puede ser descompuesta en una colección de imágenes binarias, en donde cada una representa un nivel de gris determinado. En la figura 6.2 se muestra un ejemplo de este planteamiento. Esta idea ha sido desarrollada a nivel sistema en Matlab y ha presentado resultados interesantes que valdría la pena evaluar en un proyecto futuro.

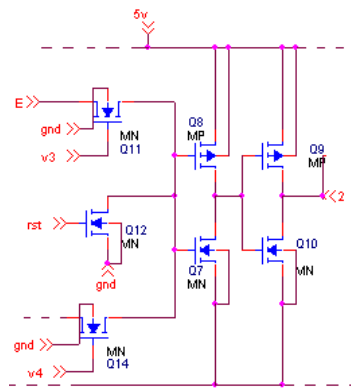


Figura 6.0 Las condiciones iniciales E son cargadas a través del inversor Q7-Q8.

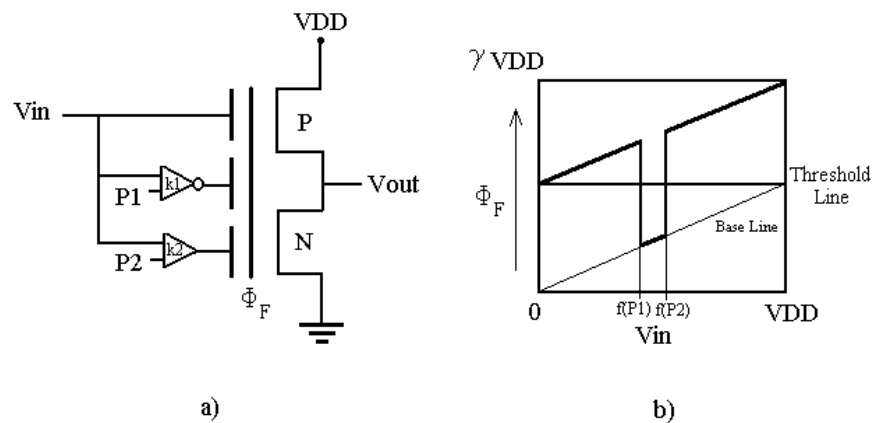
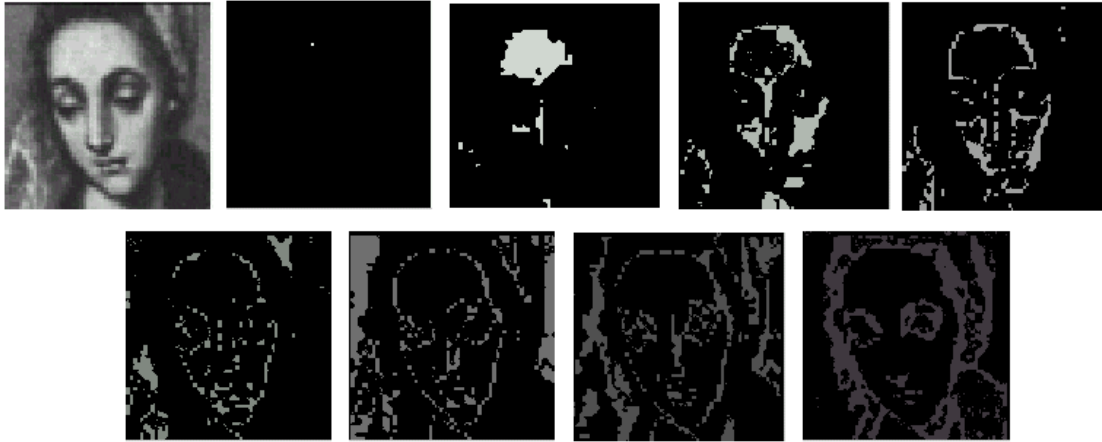


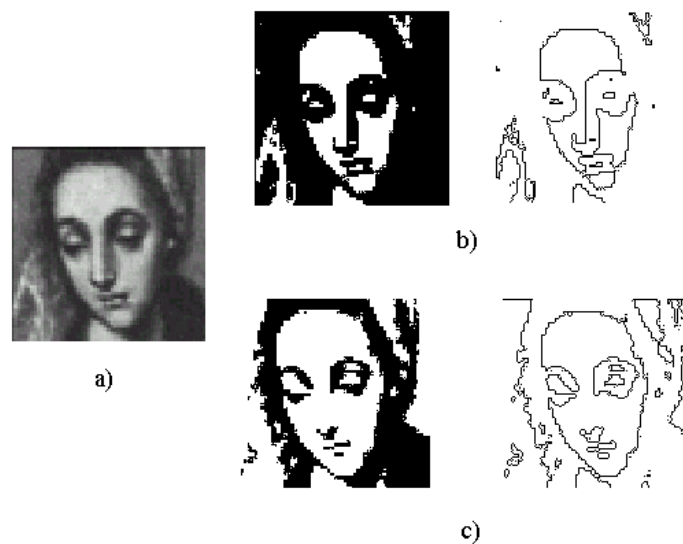
Figura 6.1 a) Estructura basada en vMOS. b) Diagrama del potencial de compuerta flotante.

Tomando en cuenta la descomposición de una imagen, sería posible realizar un grupo de operaciones morfológicas a cada nivel específico de gris, lo cual permitiría abarcar procesos de análisis de imágenes tales como lo relacionado a la segmentación y, posteriormente, unir todos los resultados mediante operaciones aritméticas y lógicas con el fin de obtener una imagen global en escala de grises compuesta de todos los niveles.



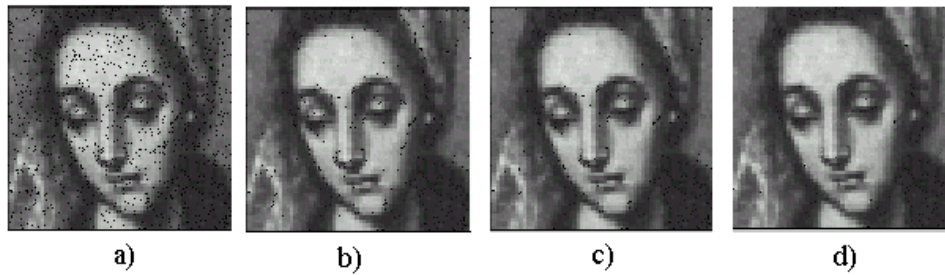
**Figura 6.2** Separación de los diferentes niveles de gris que componen una imagen.

En la figura 6.3 se muestra una operación de extracción de bordes de una imagen para dos niveles de gris diferentes. Esta operación es llevada a cabo únicamente a través de la descomposición y la utilización de la mascarilla lineal para imágenes binarias.



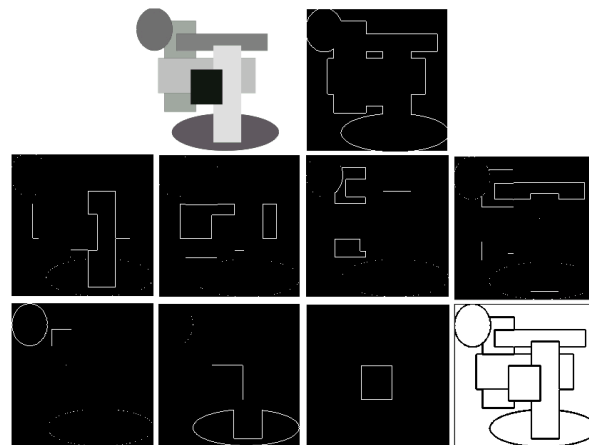
**Figura 6.3** a) Imagen de entrada. b) Extracción de bordes para una sola región de la escala de gris cercana a la mitad. c) Extracción de bordes para una región de gris por debajo de  $\frac{1}{4}$  de la escala.

En la figura 6.4 se muestra un ejemplo relativo al removedor de ruido extendido para imágenes en escala de grises, donde el resultado final (d), que en parte acumula los resultados de los procesamientos anteriores (b) y (c), demuestra una restauración considerable de la imagen de entrada.



**Figura 6.4** a) Imagen de entrada contaminada por ruido, b) Aplicación del removedor de ruido para una descomposición de 9 niveles de gris diferentes, c) Aplicación del removedor para 4 niveles, y d) Aplicación del removedor para 2 niveles.

Como último caso, el detector de contornos de imágenes en escala de grises (figura 6.5), donde la imagen de entrada es separada en 9 niveles de gris diferentes, se procesa cada nivel mediante la mascarilla lineal del detector de bordes y, posteriormente, cada resultado es unido mediante una operación lógica NOR.



**Figura 6.5** Detección de contornos en imágenes en escala de gris.

La operación que aquí se define como “descomposición”, equivale a la que es realizada en [14] como “función umbral” a nivel imagen. La diferencia entre ambas radica en la manera de como se llevan a cabo. El resultado final de la figura 6.5 puede ser comparado con aquél que se reporta en [14], donde para obtener el mismo resultado es necesario realizar un algoritmo con 16 mascarillas diferentes.



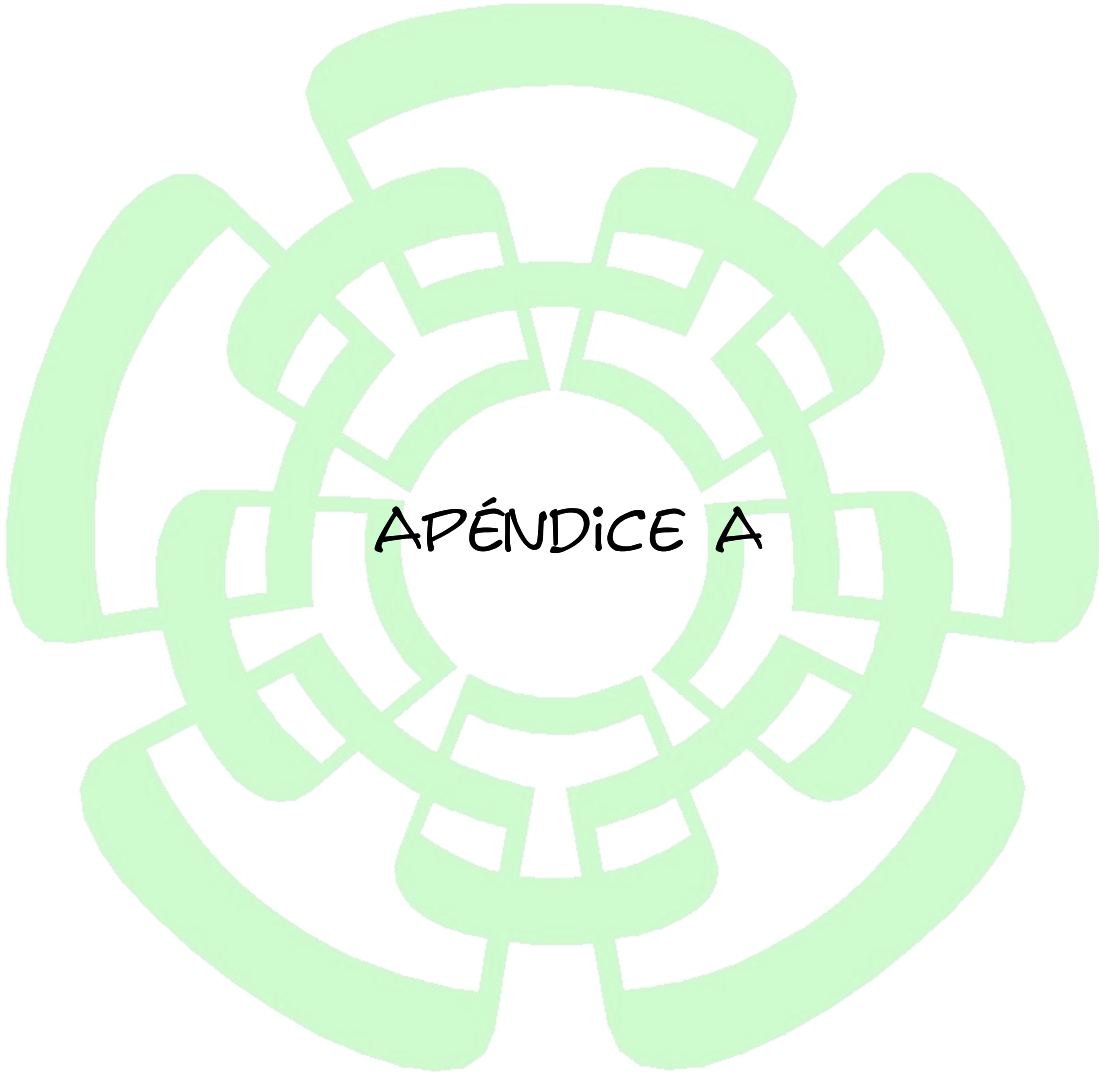
Desde nuestro punto de vista, a pesar de que dicha descomposición de imágenes en escala de grises puede resultar en un número excesivo de imágenes binarias (principalmente en aquellos procesos de alta resolución), la rapidez característica de la CNN “binaria” podría llegar a superar dicha limitante, ofreciendo buenas velocidades de operación con respecto a las otras aproximaciones.

Esta nueva estructura basada en transistores de compuerta flotante, representa un candidato potencial para la implementación de circuitos integrados con arquitectura CNN, en donde todas las mascarillas existentes para imágenes en blanco y negro podrían ser utilizadas para el análisis de imágenes en niveles de gris. Incluso permitiría realizar operaciones morfológicas sólo en ciertos niveles como un proceso fundamental en el reconocimiento de patrones.

---

## Referencias

- [1] H. Harrer, J. A. Nossek, R. Stelz, "An Analog Implementation on Discrete-Time Cellular Neural Network" IEEE Trans. On Circuits and Systems, vol.3,no. 3, May 1992.
- [2] P. Kinget and M. S. J. Steyaert, " A Programmable Analog Cellular Neural Network CMOS Chip for High Speed Image Processing" IEEE Journal of Solid State Circuits, vol. 30, no. 3, March 1995.
- [3] K. Halonen and V. Porra, " Programmable Analogue VLSI CNN Chip with Local Digital Logic" CNNA'90, Budapest, Hungary.
- [4] M. Salerno, F. Sargeni and V. Bonaiuto, " A 6x6 Cells Interconexión-Oriented programmable Chip for CNN" Analog Integrated Circuits and Signal Processing, 15, 239-250, 1998.
- [5] M. Anguita, F.J. Pelayo, E. Ros, D. Palomar and A. Prieto, " Focal-Plane and Multiple Chip VLSI Approaches to CNNs" Analog Integrated Circuits and Signal Processing, 15, 263-275, 1998.
- [6] G. F. Dalla Betta, S. Graffi, Zs. M. Kovacs and G. Masetti, "CMOS Implementation of an Analogically Programmable Cellular Neural Network" IEEE Trans. on Circuits and Systems II, vol. 40, no. 3, March 1993.
- [7] L. Ravezzi, G. F. Dalla Betta and G. Setti, " A New Current Mode Programmable Cellular Neural Network" Proc. CNNA'98, London, England.
- [8] A. Paasio, A. Dawidziuk, V. Porra, "Compact VLSI implementation of Cellular Neural Network" The Midwest Symposium on Circuits and Systems, Ames, 1996
- [9] A. Paasio and K. Halonen, "A New Cell Output Nonlinearity for Dense Cellular Nonlinear Network Integration" IEEE Trans. On Circuits and Systems I, vol. 48, no. 3 March 2001.
- [10] Ari Paasio, Asko Kananen, Kari Halonen, Veikko Porra "A 48 by 48 CNN Chip Operating with B/W Images" Fifth International Conference on Electronics, Circuits & Systems ICECS'98.
- [11] J.M. Cruz, L.O. Chua, " A 16x16 Cellular Neural Network Universal Chip: The first Complete Single-Chip Dynamic Computer Array with Distributed Memory and Gray-Scale Input-Output" Analog Integrated Circuits and Signal Processing, 15,227-237, 1998.
- [12] T. Roska, Angel Rodriguez-Vazquez, "Toward the Visual Microprocessor" John Wiley Sons 2001.
- [13] Servando Espejo, Rafael Dominguez-Castro, Gustavo Liñan, Angel Rodriguez-Vazquez, "A 64 x 64 CNN Universal Chip with Analog and Digital I/O" IEEE International Conference on Electronics Circuits and Systems (ICECS/98) pp. 203-206 Lisbon Sept. 1998.
- [14] G. Liñan, P. Foldesy, A. Rodriguez-Vazquez, S. Espejo and R. Domínguez Castro, " Implementation of Non-Linear Templates Using a Descomposition Technique by a 0.5um CMOS CNN Universal Chip" ISCAS 2000, Geneva, Switzerland.
- [15] T. Shibata and T. Ohmi, " Neuron MOS Binary-Logic Integrated Circuits- Part I: Design Fundamentals and Soft-Hardware Logic Circuit Implementation" IEEE Trans. on Electron Devices, vol, 40, no. 3 March 1993.



APÉNDICE A

## Apéndice A

A continuación se muestran las tablas con las desigualdades correspondientes de los diferentes casos de entrada hacia la neurona sin el elemento  $a_0$  (a excepción del CCD), así como la salida deseada y los resultados de las desigualdades para los valores solución de  $a$ ,  $b$ ,  $b_0$  e  $I$  para cada tarea de procesamiento particular. Los sistemas de desigualdades fueron satisfactoriamente resueltos mediante el método Simplex, de manera que tal resultado sugiere que es factible la utilización de una función de salida escalón como la de la figura 3.0.

Tabla 1 Removedor de ruido

Salida Deseada	Desigualdades	Resultados de las desigualdades para $a=2.2$ , $b_0=5.5$ e $I=1$
Blanca	$-4a + b_0 + I \leq 0$	-2.3000
	$-3a + b_0 + I \leq 0$	-0.1000
Negra	$-2a + b_0 + I \geq 0$	2.1000
	$-a + b_0 + I \geq 0$	4.3000
	$b_0 + I \geq 0$	6.5000
	$a + b_0 + I \geq 0$	8.7000
	$2a + b_0 + I \geq 0$	10.9000
	$3a + b_0 + I \geq 0$	13.1000
	$4a + b_0 + I \geq 0$	15.3000
Blanca	$-4a - b_0 + I \leq 0$	-13.3000
	$-3a - b_0 + I \leq 0$	-11.1000
	$-2a - b_0 + I \leq 0$	-8.9000
	$-a - b_0 + I \leq 0$	-6.7000
	$-b_0 + I \leq 0$	-4.5000
	$a - b_0 + I \leq 0$	-2.3000
	$2a - b_0 + I \leq 0$	-0.1000
Negra	$3a - b_0 + I \geq 0$	2.1000
	$4a - b_0 + I \geq 0$	4.3000

Tabla 2 Extractor de sombras

Salida Deseada	Desigualdad	Resultados de las desigualdades para $a=2$ , $b_0=2$ e $I=1$
Negra	$a+b_0+I \geq 0$	5.0458
	$-a+b_0+I \geq 0$	1.0417
	$b_0+I \geq 0$	3.0437
Blanca	$-b_0+I \leq 0$	-1.0021
	$-a-b_0+I \leq 0$	-3.0042
Negra	$a-b_0+I \geq 0$	0.9999

Tabla 3 Detector de componentes conectados (CCD)

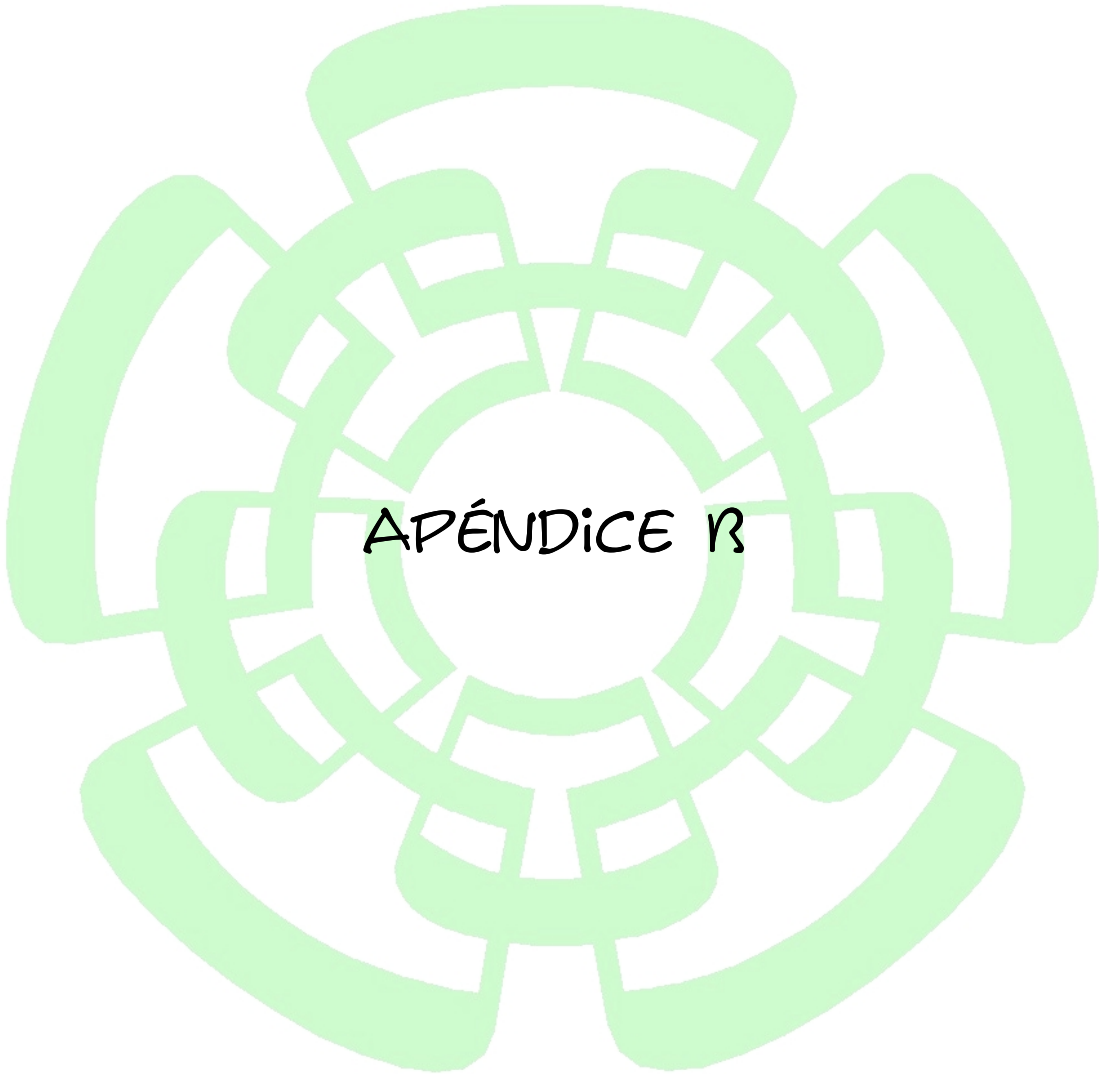
Vecinos Blancos	Desigualdad	Resultados de las Desigualdades para $a_0=1.2$ , $a_1=1.2$ $a_2=-1.1$ $I=0.1$
Negra	$a_0+a_1-a_2+I \geq 0$	3.7078
	$-a_0+a_1-a_2+I \geq 0$	1.3111
	$a_0+a_1+a_2+I \geq 0$	1.3919
	$a_0-a_1-a_2+I \geq 0$	1.3097
Blanca	$-a_0+a_1+a_2+I \leq 0$	-1.0048
	$-a_0-a_1-a_2+I \leq 0$	-1.0870
	$-a_0+a_2 -a_1+I \leq 0$	-3.4029
	$a_0+a_2-a_1+I \leq 0$	-1.0062

Tabla 4 Detector de conectividad global

Salida Deseada	Desigualdades	Resultados de las desigualdades para $a=2.3$ , $bo=6.9$ e $I=3.5$
Negra	$2a - bo + I \geq 0$	1.2000
	$3a - bo + I \geq 0$	3.5000
	$4a - bo + I \geq 0$	5.8000
	$4a + bo + I \geq 0$	19.6000
	$3a + bo + I \geq 0$	17.3000
	$2a + bo + I \geq 0$	15.0000
	$a + bo + I \geq 0$	12.7000
	$bo + I \geq 0$	10.4000
	$-a + bo + I \geq 0$	8.1000
	$-2a + bo + I \geq 0$	5.8000
	$-3a + bo + I \geq 0$	3.5000
	$-4a + bo + I \geq 0$	1.2000
	Blanca	$-bo + I \leq 0$
$-2a - bo + I \leq 0$		-8.0000
$-4a - bo + I \leq 0$		-12.6000
$a - bo + I \leq 0$		-1.1000
$-a - bo + I \leq 0$		-5.7000
$-3a - bo + I \leq 0$		-10.3000

Tabla 5 Extractor de bordes

Salida Deseada	Desigualdades	Resultados de las desigualdades para $b=-2.54$ , $bo=10$ e $I=-1.14$
Blanca	$4b - bo + I \leq 0$	-21.3000
	$2b - bo + I \leq 0$	-16.2200
	$-bo + I \leq 0$	-11.1400
	$-2b - bo + I \leq 0$	-6.0600
	$-4b - bo + I \leq 0$	-0.9800
	$3b - bo + I \leq 0$	-18.7600
	$b - bo + I \leq 0$	-13.6800
	$-b - bo + I \leq 0$	-8.6000
	$-3b - bo + I \leq 0$	-3.5200
Negra	$2b + bo + I \geq 0$	3.7800
	$bo + I \geq 0$	8.8600
	$-2b + bo + I \geq 0$	13.9400
	$-4b + bo + I \geq 0$	19.0200
	$3b + bo + I \geq 0$	1.2400
	$b + bo + I \geq 0$	6.3200
	$-b + bo + I \geq 0$	11.4000
$-3b + bo + I \geq 0$	16.4800	
Blanca	$4b + bo + I \leq 0$	-1.3000



APÉNDICE B

## Apéndice B

Para la resolución de los siguientes sistemas de desigualdades, también se toma en cuenta la consideración hecha en la sección 2.1.3 referente al elemento  $f$  utilizado para reducir la frontera solución de las desigualdades en la función objetivo. A mayores valores de  $f$ , la función objetivo exige al método la localización de soluciones matemáticas con un mayor margen de tolerancia (ver figura A.0). Esto es, que los resultados de las desigualdades para blancos y negros se alejen en mayor grado del punto de transición de la función escalón. Esto permite, como ya fue mencionado, que a pesar de las variaciones tecnológicas de parámetros y márgenes de error en elementos eléctricos, se asegure el correcto desempeño de la implementación.

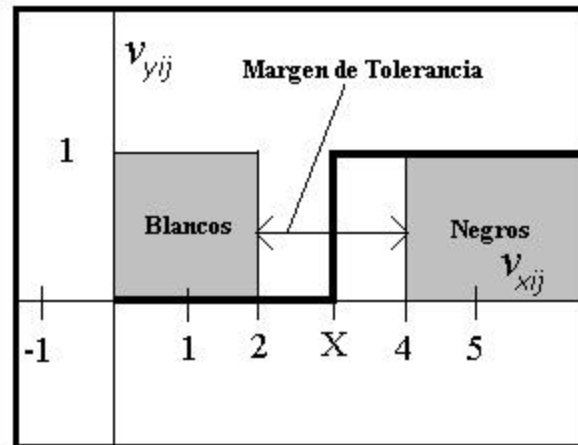


Figura A.0

A continuación se muestran las tablas de las desigualdades y su solución para las diferentes tareas de procesamiento consideradas en este trabajo. Se observa que el margen de tolerancia es diferente para cada tarea particular dada la susceptibilidad de la solución matemática para los diferentes procesos. El más pequeño en este sentido es el removedor de ruido con un valor cercano a 2.3 unidades y el mayor es el del detector de componentes conectados con 6 unidades.

### TAREA 1. Removedor de Ruido para X=45

$$A = \begin{bmatrix} 0 & a & 0 \\ a & 0 & a \\ 0 & a & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, f$$

TABLA 1

Desigualdades	Resultado para $n_{OB}=28.1$ $b_{OB}=1.8$ $n_A=12.5$ $b_A=2.2$ $f=7.8$
$n_{OB} + 4b_A < X$	37.0140
$n_{OB} + 3 b_A + f < X$	42.6389
$n_{OB} + n_A + 3 b_A > X$	47.3611
$n_{OB} + n_A + 2 b_A + f > X$	52.9860
$n_{OB} + 2 n_A + 2 b_A > X$	57.7082
$n_{OB} + 2 n_A + b_A + f > X$	63.3331
$n_{OB} + 3 n_A + b_A > X$	68.0553
$n_{OB} + n_A + f > X$	73.6802
$n_{OB} + 3 n_A + f > X$	78.4023
$b_{OB} + 4 b_A < X$	10.6949
$b_{OB} + 3 b_A + f < X$	16.3198
$b_{OB} + n_A + 3 b_A < X$	21.0420
$b_{OB} + n_A + 2 b_A + f < X$	26.6669
$b_{OB} + 2 n_A + 2 b_A < X$	31.3891
$b_{OB} + 2 n_A + b_A + f < X$	37.0140
$b_{OB} + 3 n_A + b_A < X$	41.7362
$b_{OB} + 3 n_A + f > X$	47.3611
$b_{OB} + 4 n_A > X$	52.0833



**TAREA 2. Detector de Conectividad Global para X=30**

$$A = \begin{bmatrix} 0 & a & 0 \\ a & 0 & a \\ 0 & a & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, f$$

TABLA 2

Desigualdades	Resultado para $n_{OB}=44.1$ $b_{OB}=0$ $n_A=11.4$ $b_A=0$ $f=1.6$
$b_{OB} + 3n_A + b_A > X$	34.2000
$b_{OB} + 3 n_A > X$	35.8000
$b_{OB} + 4 n_A > X$	45.6000
$n_{OB} + 4 n_A > X$	89.7000
$n_{OB} + 3 n_A > X$	79.9000
$n_{OB} + 3 n_A + b_A > X$	78.3000
$n_{OB} + 2 n_A + b_A > X$	68.5000
$n_{OB} + 2 n_A + 2 b_A > X$	66.9000
$n_{OB} + n_A + 2 b_A > X$	57.1000
$n_{OB} + n_A + 3 b_A > X$	55.5000
$n_{OB} + 3 b_A > X$	45.7000
$n_{OB} + 4 b_A > X$	44.1000
$b_{OB} + 2 n_A + 2 b_A < X$	22.8000
$b_{OB} + n_A + 3 b_A < X$	11.4000
$b_{OB} + 4 n_A < X$	0
$b_{OB} + 2 n_A + b_A < X$	24.4000
$b_{OB} + n_A + 2 b_A < X$	13.0000
$b_{OB} + 3 b_A < X$	1.6000

---

**TAREA 3. Extractor de Bordes para X=41**

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & b & 0 \\ b & b_o & b \\ 0 & b & 0 \end{bmatrix}, f$$

TABLA 3

Desigualdades	Resultado para $n_{OB}=37.4$ $b_{OB}=9.4$ $n_B=0.2$ $b_B=5.4$ $f=8.1$
$b_{OB} + 4n_B < X$	10.2000
$b_{OB} + 3 n_B + b_B < X$	15.4000
$b_{OB} + 2 n_B + 2 b_B < X$	20.6000
$b_{OB} + n_B + 3 b_B < X$	25.8000
$b_{OB} + 4 b_B < X$	31.0000
$b_{OB} + 3 n_B < X$	18.1000
$b_{OB} + 2 n_B + b_B < X$	23.3000
$b_{OB} + n_B + 2 b_B < X$	28.5000
$b_{OB} + 3 b_B < X$	33.7000
$n_{OB} + 3 n_B + b_B > X$	43.4000
$n_{OB} + 2 n_B + 2 b_B > X$	48.6000
$n_{OB} + n_B + 3 b_B > X$	53.8000
$n_{OB} + 4 b_B > X$	59.0000
$n_{OB} + 3 n_B > X$	46.1000
$n_{OB} + 2 n_B + b_B > X$	51.3000
$n_{OB} + n_B + 2 b_B > X$	56.5000
$n_{OB} + 3 b_B > X$	61.7000
$n_{OB} + 4 n_B < X$	38.2000

**TAREA 4. Extractor de Sombras para X=30**

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & a \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & 0 \end{bmatrix}, f$$

TABLA 4

Desigualdades	Resultado para $n_{OB}=35$ $b_{OB}=5$ $n_A=35$ $b_A=5$ $f=0$
$n_{OB} + f > X$	35
$n_{OB} + n_A > X$	70
$n_{OB} + b_A > X$	40
$b_{OB} + n_A > X$	40
$b_{OB} + b_A < X$	10
$b_{OB} + f < X$	5

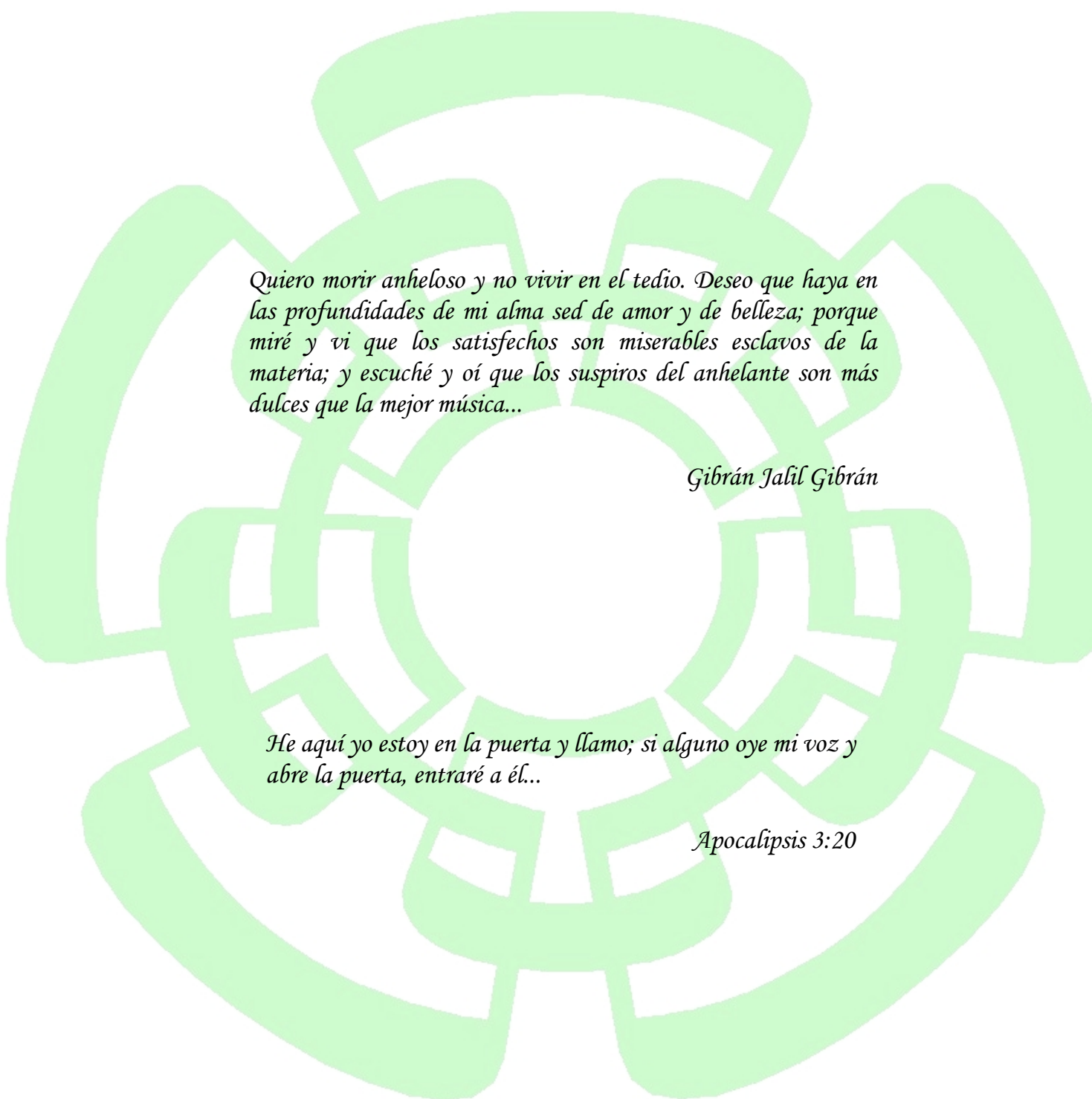
**TAREA 5. Detector de Componentes Conectados para X=42**

$$A = \begin{bmatrix} 0 & 0 & 0 \\ a_2 & a_o & a_1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

TABLA 5

Desigualdades	Resultado para $n_{OA}=21.5$ $b_{OA}=5$ $n_A=14.1$ $b_{A1}=30.6$ $b_{A2}=0$
$b_{OA} + n_A + b_{A1} > X$	49.8621
$n_{OA} + b_{A2} + n_A < X$	35.7931
$n_{OA} + b_{A2} + b_{A1} > X$	52.3448
$n_{OA} + n_A + b_{A1} > X$	66.4138
$n_{OA} + 2n_A > X$	49.8621
$b_{OA} + b_{A2} + b_{A1} < X$	35.7931
$b_{OA} + b_{A2} + n_A < X$	19.2414
$b_{OA} + 2n_A < X$	33.3104

Para el detector de componentes conectados, el sistema no fue resuelto con la variable “f”, ya que se trata de un problema que puede ser analizado de manera unidimensional y el adecuado desempeño de las neuronas de frontera (cumpliendo con lo establecido en la sección 2.2.4) se alcanza suponiendo una contribución de salida blanca para las fronteras derecha e izquierda.



*Quiero morir anheloso y no vivir en el tedio. Deseo que haya en las profundidades de mi alma sed de amor y de belleza; porque miré y vi que los satisfechos son miserables esclavos de la materia; y escuché y oí que los suspiros del anhelante son más dulces que la mejor música...*

*Gibrán Jalil Gibrán*

*He aquí yo estoy en la puerta y llamo; si alguno oye mi voz y abre la puerta, entraré a él...*

*Apocalipsis 3:20*