



# CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL IPN

Departamento de Ingeniería Eléctrica  
Sección de Electrónica del Estado Sólido



## “Prototipo en FPGA de Red Neuronal con Memristores”

*Tesis que presenta:*

Ing. Erick Moreno Cuellar

---

*Directores de Tesis:*

Dr. Felipe Gómez Castañeda

Dr. José Antonio Moreno Cadenas

# Contenido

---

- Objetivos
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Objetivos

---

## General

- Simular e implementar un modelo del Memristor en tecnología digital usando un sistema FPGA y mostrar diferentes aplicaciones.

## Específicos

- Implementar el modelo bipolar con umbral del memristor controlado por voltaje en la tarjeta Nexys 4DDR.
- Simular e implementar una neurona pulsada accesible computacionalmente para uso en FPGA.
- Reproducir el experimento Pavlov's Dog usando el memristor como sinapsis entre neuronas.
- Implementar una red neuronal pulsada con Memristores para el reconocimiento de caracteres.
- Realizar una red de Memristores usándolos como análogo del algoritmo de hormigas.
- Simular una red de Memristores usada para la detección de bordes en imágenes.

# Contenido

---

- *Objetivos*
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Introducción

El memristor, un dispositivo eléctrico propuesto por Chua plantea nuevas formas de interactuar con el resto de dispositivos electrónicos, puesto que combina dos funciones (almacenamiento y procesamiento) en un solo elemento.

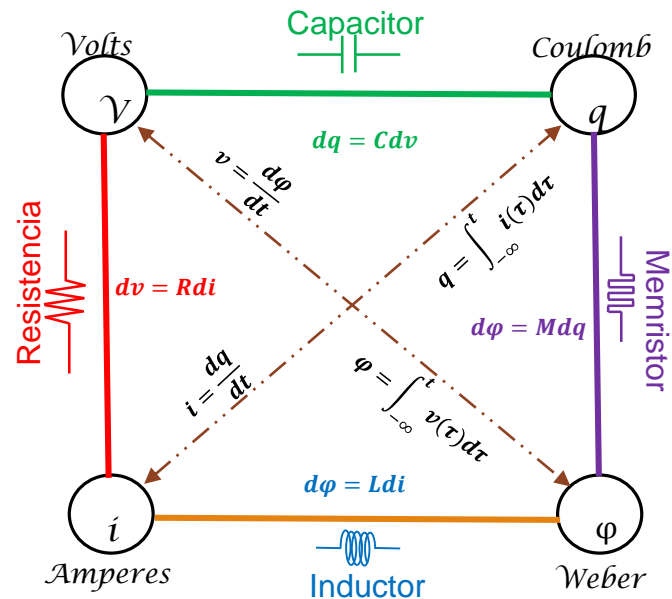


Figura 1 Elementos fundamentales donde se aprecia la simetría y relaciones entre las variables de cada uno.

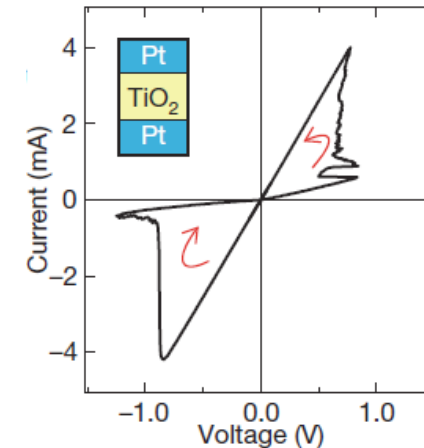
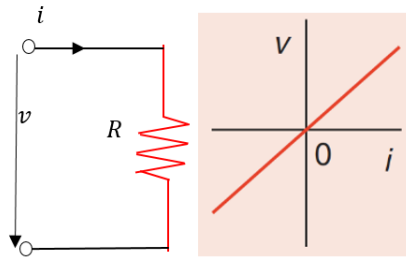
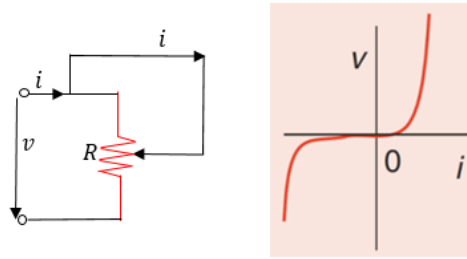


Figura 2 Memristor obtenido por "HP" proveniente de [1]

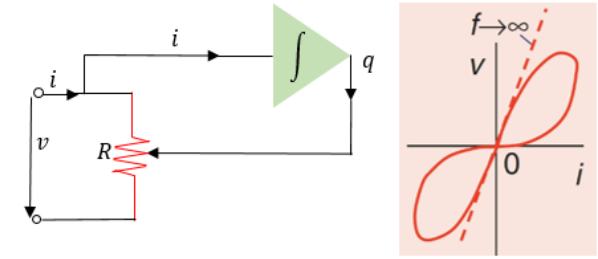
# Evolución de elementos resistivos controlados por corriente



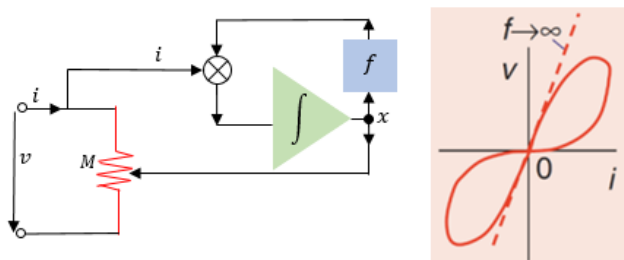
A) Resistencia



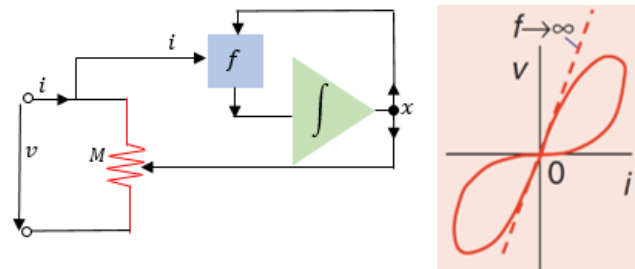
B) Resistencia no lineal



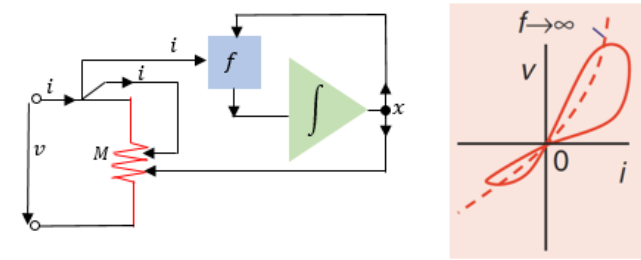
C) Memristor ideal



D) Memristor genérico ideal



E) Memristor genérico



Memristor extendido

Figura 3 Interpretación de la familia de memristores según Biolek, ref .[9]

# Contenido

---

- *Objetivos*
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Modelo del Memristor Usado

---

Un simulador del Memristor debe contar con ciertas características[2]:

1. Un amplio rango de Memristancia.
2. No volatilidad.
3. Configuración del estado inicial.
4. Operación flotante.
5. Operatividad para alta frecuencia y señales continuas de entrada.
6. Capaz de interconectarse con otros dispositivos.





# Modelo del Memristor Usado

*Modelo bipolar con umbral controlado por voltaje*

1.  $i(t) = M^{-1} \cdot v(t)$
2.  $\dot{M} = b \cdot v + \frac{1}{2}(a - b) \cdot (|v + v_T| - |v - v_T|) \cdot \theta(M - R_{ON}) \cdot \theta(R_{OFF} - M)$

Circuito utilizado para simular el aprendizaje de la ameba [3] y [4]

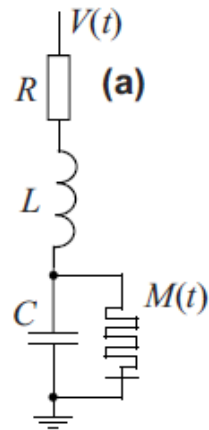


Figura 4 Circuito emulador del aprendizaje de la ameba

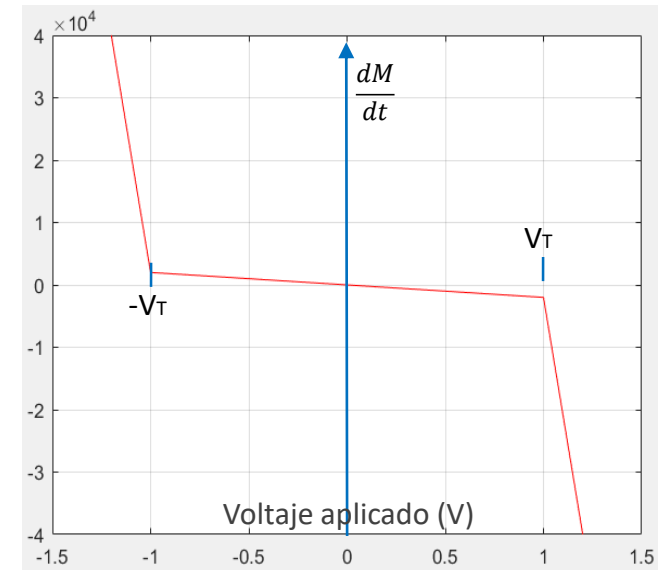


Figura 5 Cambio de  $\dot{R}$  dependiendo del voltaje aplicado

# Contenido

---

- *Objetivos*
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Redes Neuronales Pulsadas

## ➤ Sinapsis

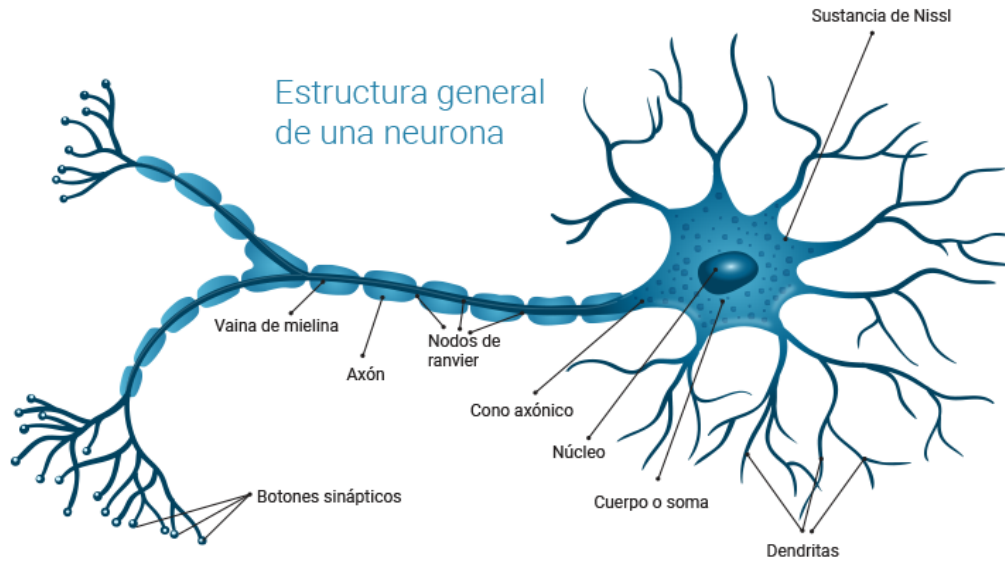


Figura 6 Partes de la Neurona

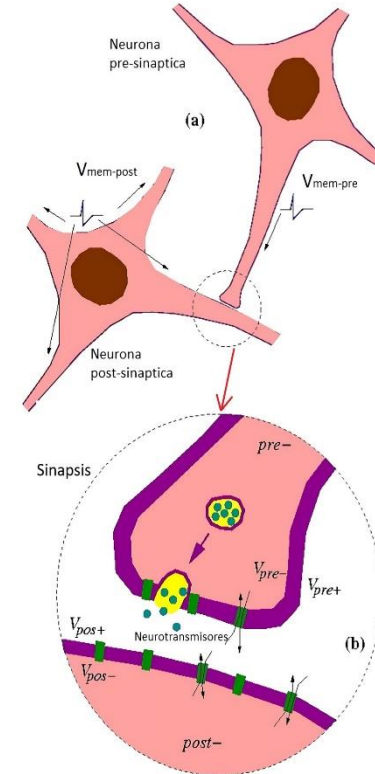


Figura 7 proceso de sinapsis entre neuronas tomado de [5]

# Redes Neuronales Pulsadas

## STDP

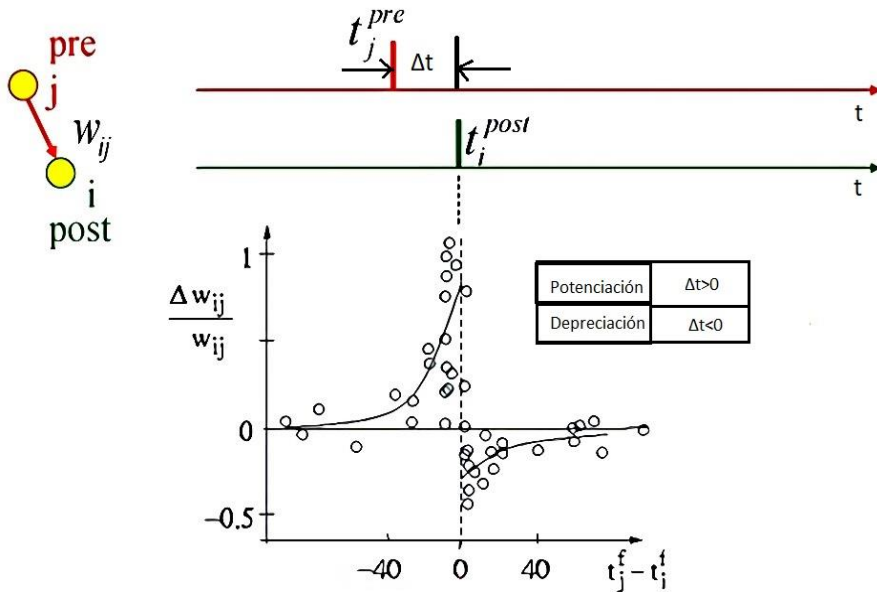


Figura 8 Proceso Spike-Timing-Dependent-Plasticity (STDP), mostrando el cambio de los pesos sinápticos (1998).

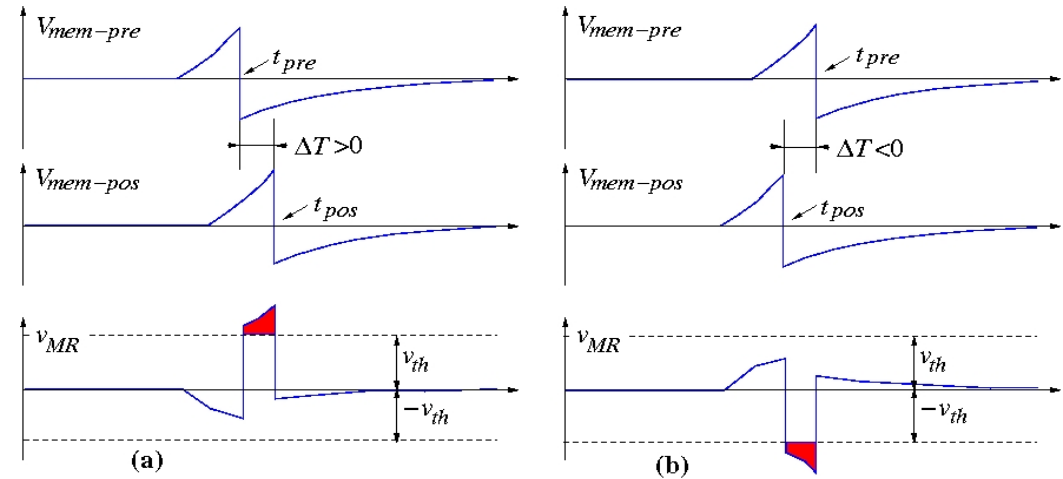


Figura 9 Proceso Spike-Timing-Dependent-Plasticity (STDP) colocando un memristor, tomado de [5]

# Contenido

---

- *Objetivos*
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Algoritmos Metaheurísticos

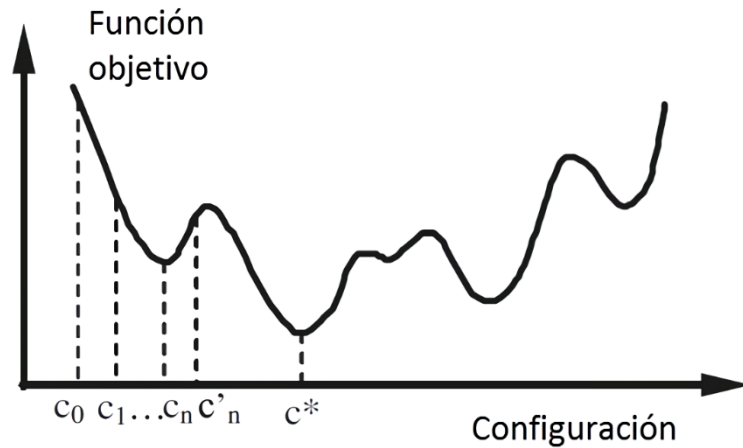


Figura 10 Función objetivo con mínimos locales, tomado de [6]

- Las Metaheurísticas distribuidas permiten la salida de una solución particular.
- Ayudan a realizar una mejor búsqueda contra los mínimos locales.

# Algoritmo Metaheurístico de Colonia de Hormigas

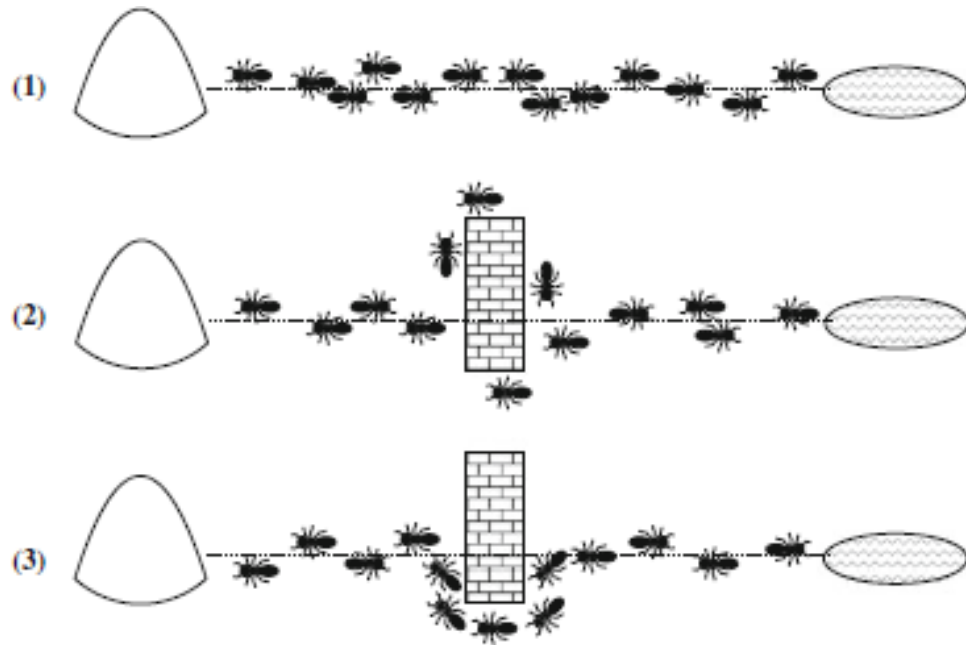


Figura 11 Ejemplo, algoritmo de hormigas proveniente de [6]

➤ Propagación estocástica de agentes (hormigas)

$$1.- \quad p_{ij,k} = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_m \tau_{im}^{\alpha} \eta_{im}^{\beta}},$$

$$2.- \quad \tau_{ij}(k+1) = (1 - \rho)\tau_{ij}(k) + \nu \frac{Q}{L_k},$$

# Algoritmo Metaheurístico de Colonia de Hormigas

Naturaleza	Algoritmo de Hormigas	Memcomputing
Hábitat natural	Grafo	Red de memristores
Hormigas	Hormigas artificiales	Electrones
Longitud del camino	Reciproco de la distancia	Conductancia
Feromonas	Feromonas artificiales	Memoria del dispositivo
Evaporación de feromonas	Evaporación artificial de feromonas	Estado de relajación del dispositivo
Comportamiento de búsqueda	Reglas de selección	Leyes de Kirchhoff

Tabla de equivalencia propuesta por Y. V. Pershin y M. Di Ventra [7].

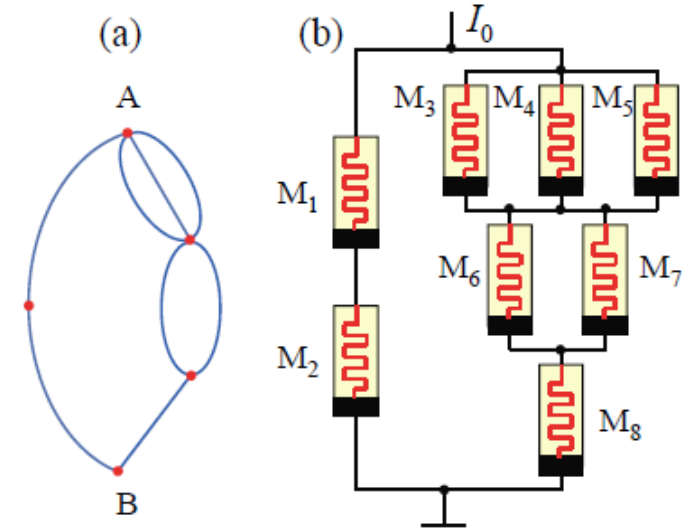


Figura 12 Red de Memristores, análogos al algoritmo de optimización de hormigas

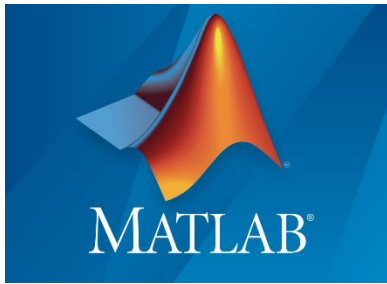


# Contenido

---

- *Objetivos*
- *Introducción*
- *Modelo Bipolar con Umbral del Memristor controlado por voltaje*
- *Redes Neuronales Pulsadas*
- *Algoritmos Metaheurísticos*
- *Resultados*
- *Conclusiones*
- *Referencias*

# Desarrollo de prototipo en FPGA (proceso)



Código



Programación  
gráfica ideal

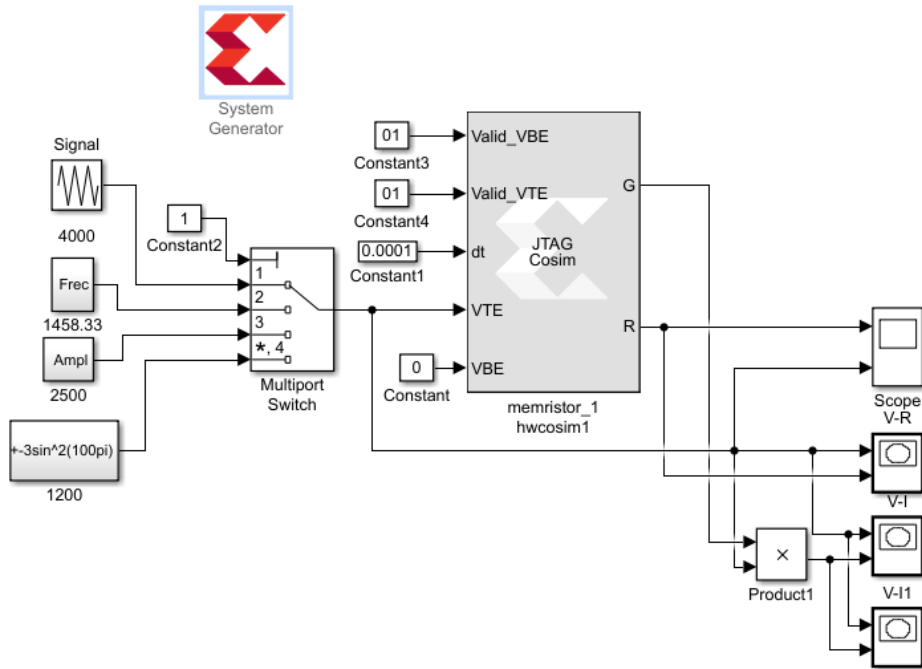


Simulación



Co-Simulación

# Resultados: Simulación e Implementación del Memristor



## Implementación del Memristor en el FPGA

Nexys 4 DDR Artix-7		Punto flotante 32 bits
DSPs	20 (8.3%)	
LUTs	3141 (5%)	
Registers	38 (<0.6%)	

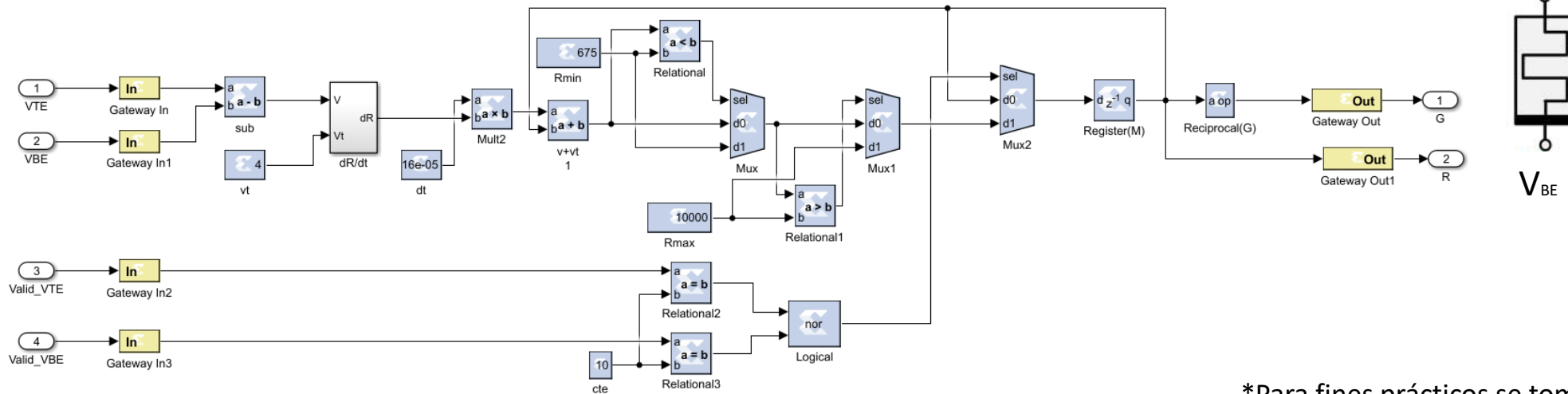
  

Nexys 4 DDR Artix-7		Punto fijo 18 bits
DSPs	18 (7.5%)	
LUTs	3487 (5.5%)	
Registers	76(<0.6%)	

# Resultados: Simulación e Implementación del Memristor

*Modelo bipolar con umbral controlado por voltaje*

1.  $i(t) = M^{-1} \cdot v(t)$
2.  $\dot{M} = b \cdot v + \frac{1}{2} (a - b) \cdot (|v + v_T| - |v - v_T|) \cdot \theta(M - R_{ON}) \cdot \theta(R_{OFF} - M)$



\*Para fines prácticos se toma  $M \equiv R$

# Resultados: Simulación e Implementación del Memristor

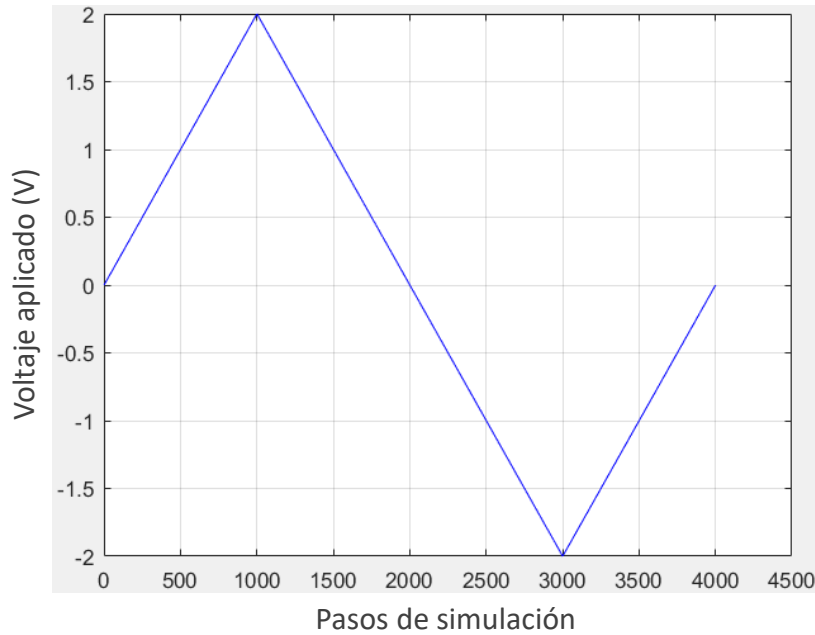


Figura 13 Voltaje aplicado

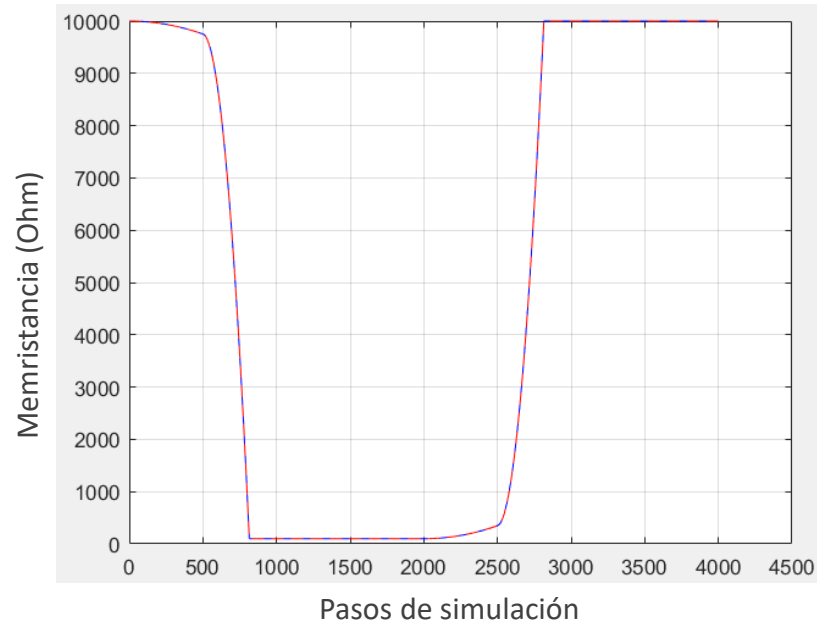


Figura 14 Respuesta de la Memristancia

— Simulink  
- - - System Generator y Nexys 4 DDR

Valores utilizados

$$a = -2000 \frac{\Omega}{V \cdot s}, \quad b = -190\,000 \frac{\Omega}{V \cdot s}, \quad \Delta t = 0.0005s, \quad V_T = 1V, \quad R_{min} = 100\Omega, \quad R_{max} = 10k\Omega$$

# Resultados: Simulación e Implementación del Memristor

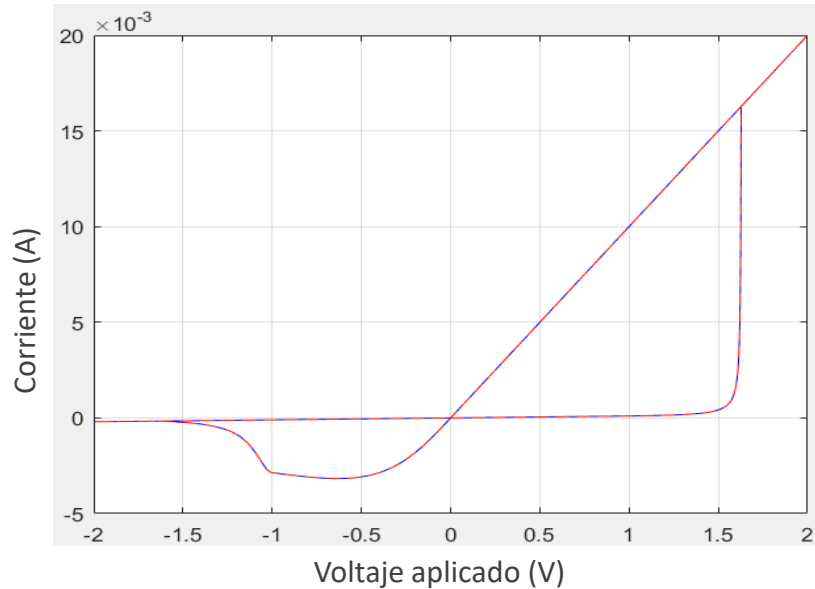


Figura 15 Histéresis característica de la curva I-V

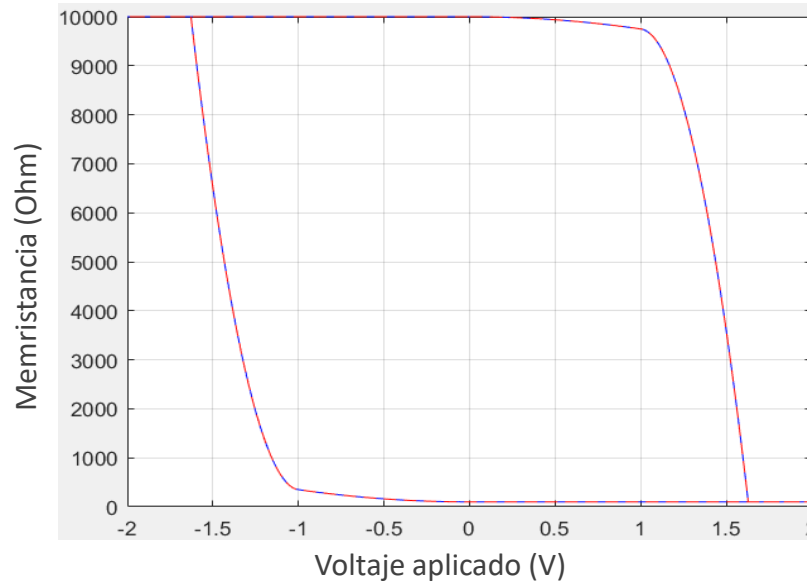
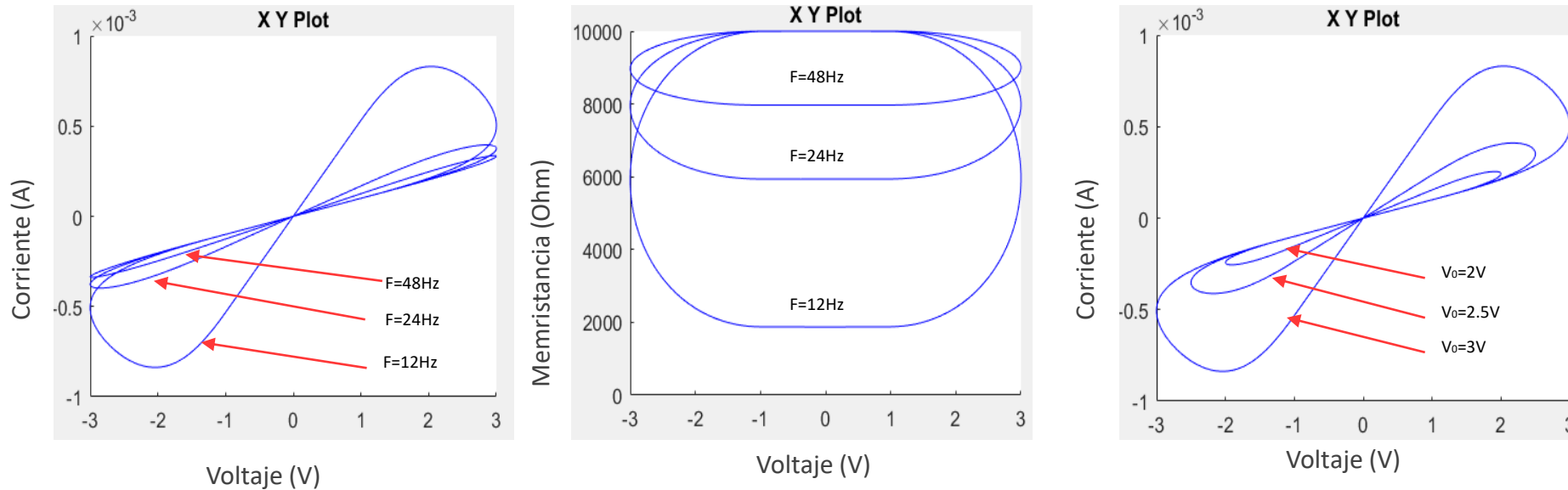


Figura 16 Cambio de Memristancia dado el voltaje aplicado

— Simulink  
- - - System Generator y Nexys 4 DDR

Comparación entre la simulación realizada con Simulink (línea puntada roja) y los resultados obtenidos en la implementación realizada con System Generator y el FPGA.

# Resultados: Verificación de los Fingerprints



System Generator en co-simulación con la tarjeta Nexys 4 DDR Artix 7 FPGA

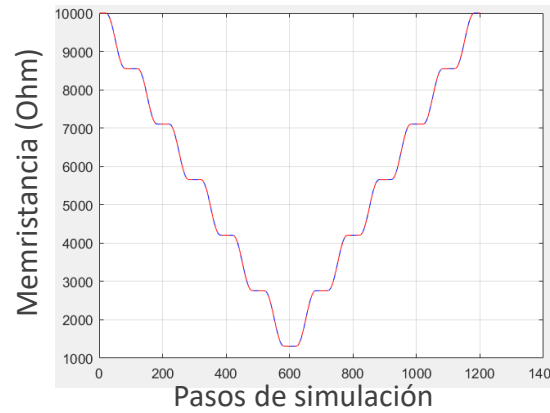
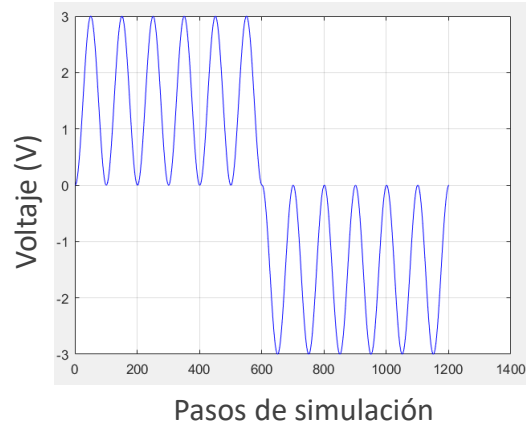
Figura 17 Verificación de los fingerprints en System Generator

Valores utilizados en todos los casos

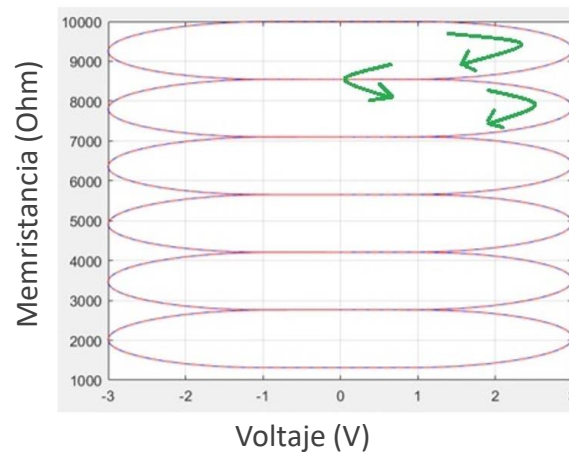
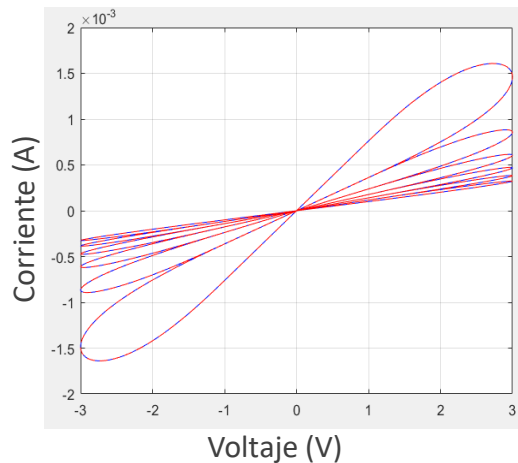
$$a = -2000 \frac{\Omega}{V \cdot s}, \quad b = -190\,000 \frac{\Omega}{V \cdot s}, \quad \Delta t = 0.0001s, \quad V_T = 1V, \quad R_{min} = 100\Omega, \quad R_{max} = 10k\Omega, \quad R_{init} = R_{max}$$

El voltaje aplicado es:  $v(t) = 3\sin(2\pi ft)$

# Resultados: Capacidad de ajuste multi-estado.



— Simulink  
- - - System Generator y Nexys 4 DDR

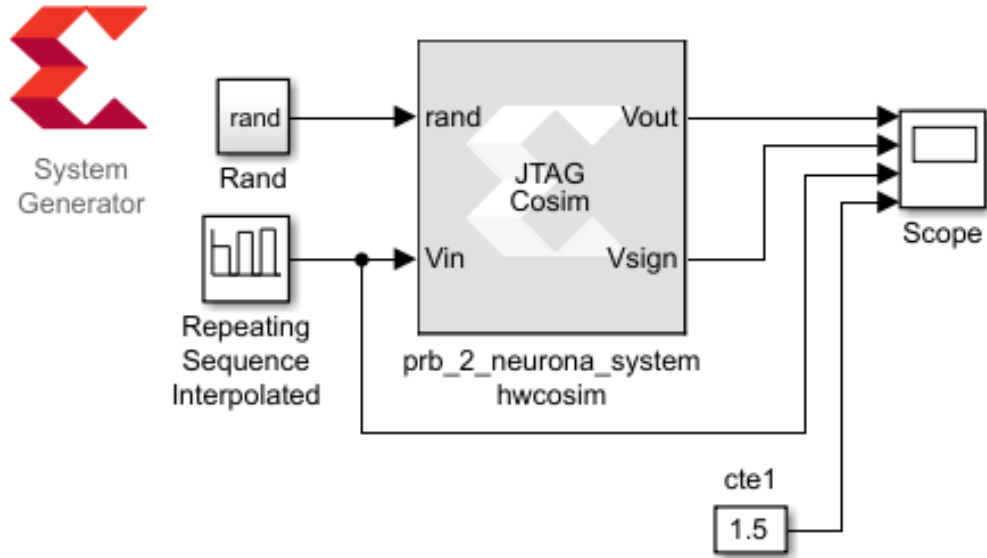


Valores utilizados  
 $a = -2000 \frac{\Omega}{V \cdot s}$ ,  $b = -190\,000 \frac{\Omega}{V \cdot s}$ ,  $\Delta t = 0.0001s$ ,  $V_T = 1V$ ,  $R_{min} = 100\Omega$ ,  $R_{max} = 10k\Omega$ ,  $R_{init} = R_{max}$ ,  $v(t) = \pm 3\sin^2(100\pi t)$

Figura 18 Comparación entre resultados de Simulink y System Generator



# Resultados: Simulación e Implementación de la Neurona



## Implementación de la neurona en el FPGA

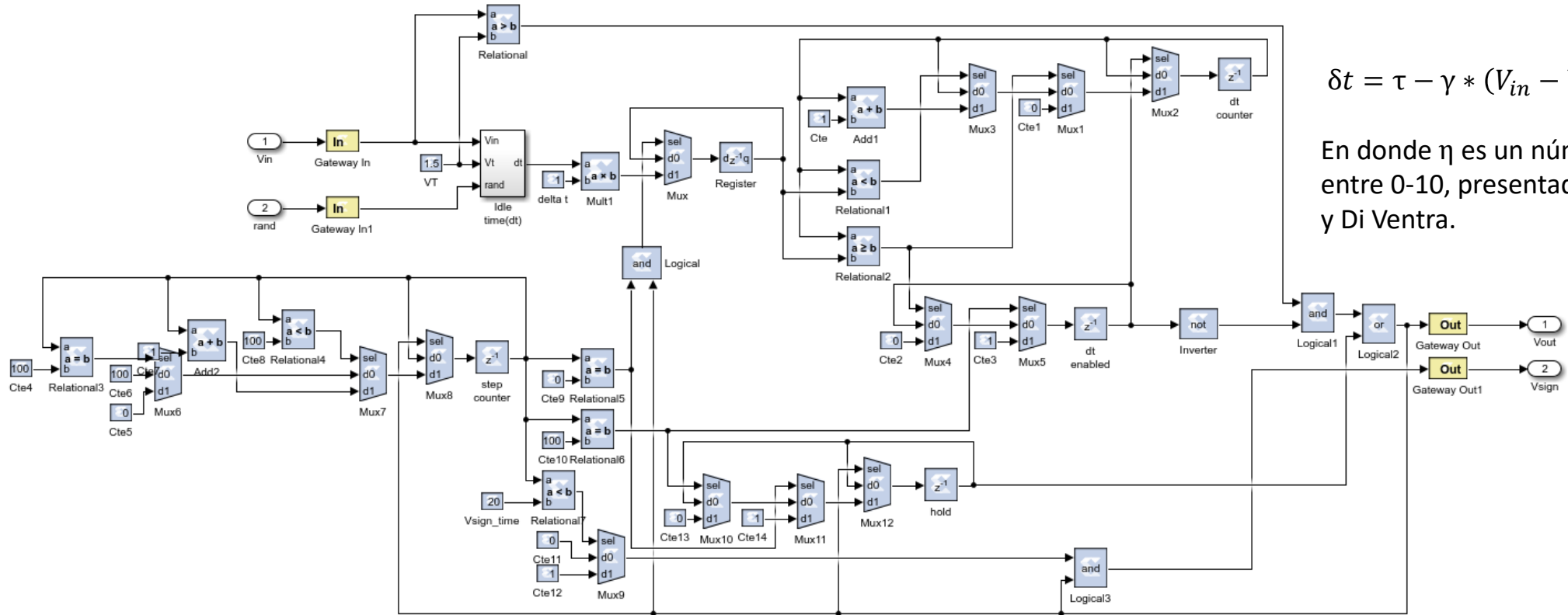
Nexys 4 DDR Artix-7	
DSPs	9 (4%)
LUTs	2814 (4.5%)
Registers	98 (<1%)

Punto flotante 32 bits

Nexys 4 DDR Artix-7	
DSPs	0 (0%)
LUTs	190 (<1%)
Registers	52(<1%)

Punto fijo 18 bits

# Resultados: Simulación e Implementación de la Neurona



$$\delta t = \tau - \gamma * (V_{in} - V_T) + \lambda(\eta - 5)$$

En donde  $\eta$  es un número aleatorio entre 0-10, presentado por Pershin y Di Ventra.

# Resultados: Simulación e Implementación de la Neurona

System Generator en co-simulación con la tarjeta Nexys 4 DDR  
Artix 7 FPGA

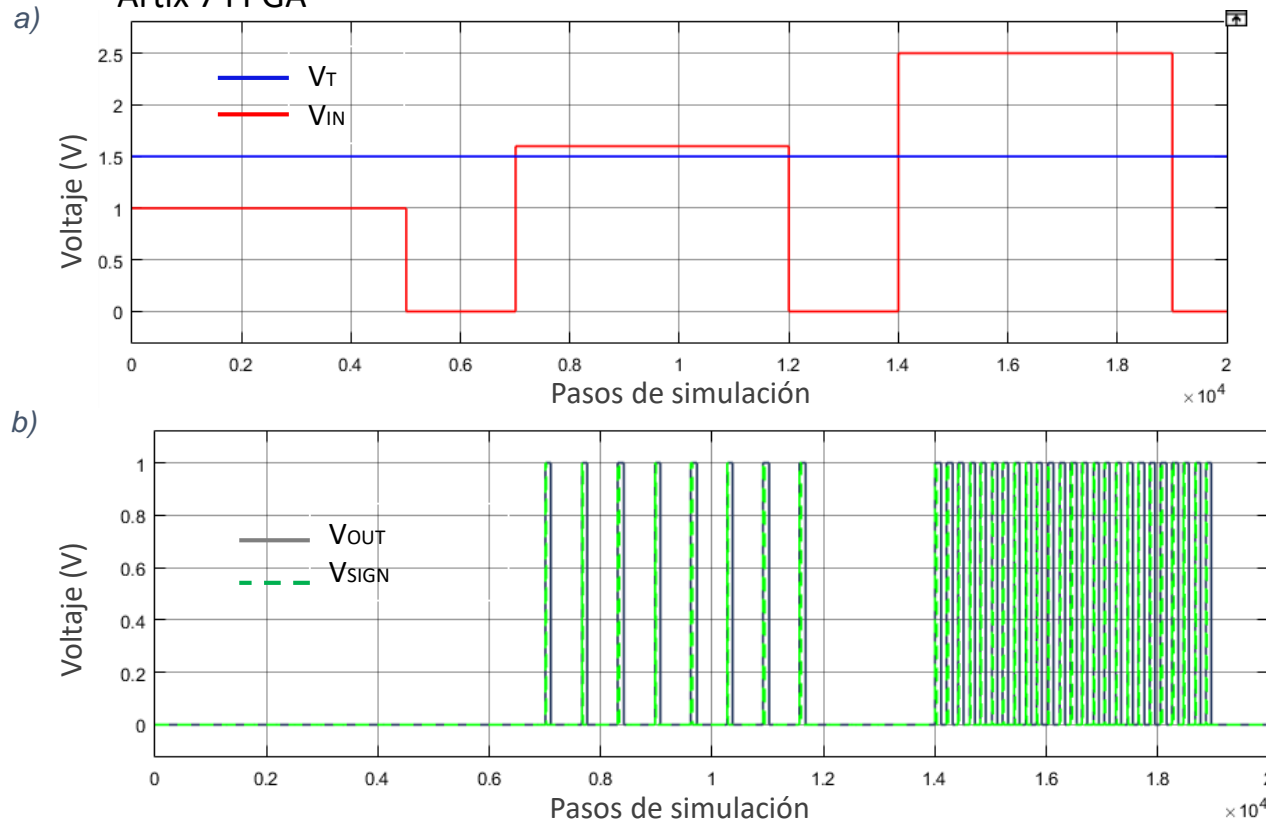


Figura 19, a) Voltaje aplicado, b) Respuesta de la neurona

Valores utilizados

$$\tau = 500ms, \gamma = 512 \frac{ms}{V} \quad V_T = 1.5V, \lambda = 1ms$$

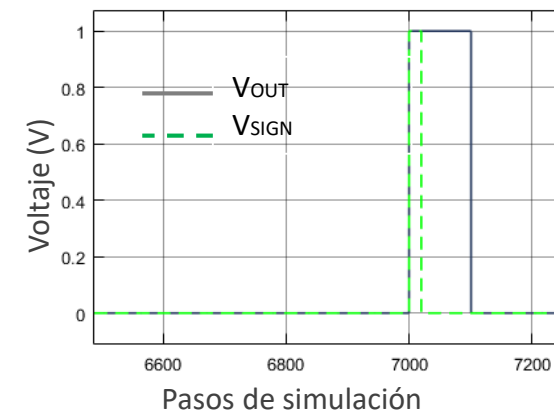


Figura 20 Ampliación figura 16a para visualizar  $V_{SIGN}$

# Resultados: "Pavlov's dog"

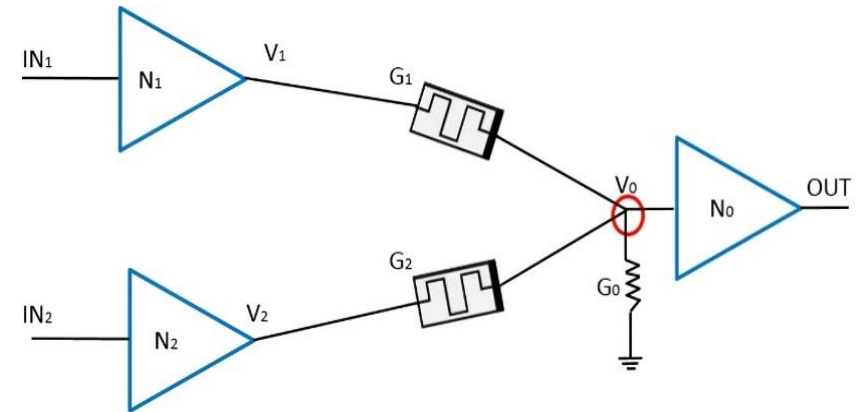
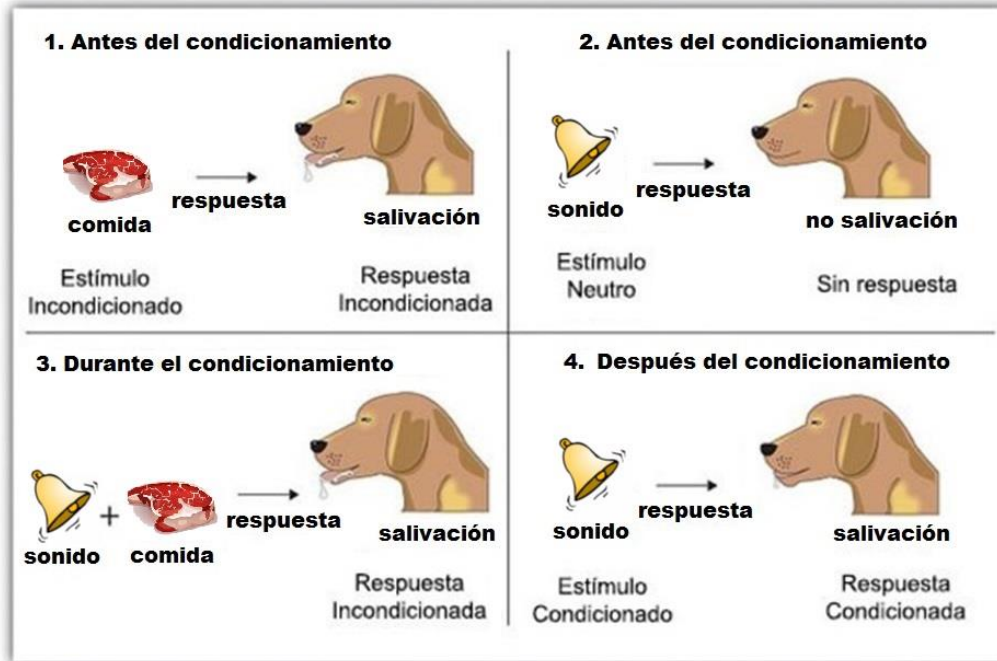
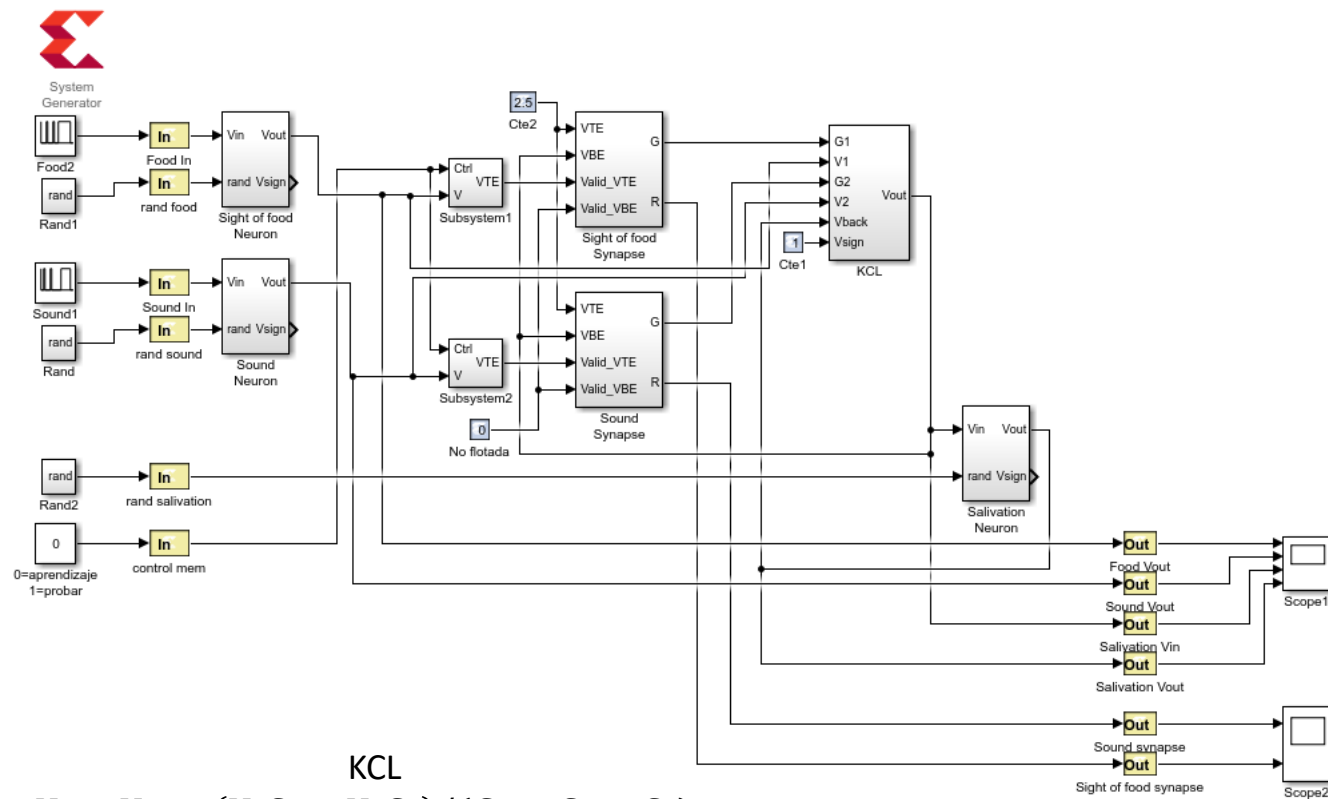


Figura 21 Ejemplo del experimento Pavlov's dog y diagrama a implementar.

# Resultados: ANN con Memoria Asociativa



KCL

- $V_0 = V_{pulse}(V_1G_1 + V_2G_2)/(G_0 + G_1 + G_2)$
- $G_0 = 1/R_0$

## Implementación de ANN en el FPGA

### Nexys 4 DDR Artix-7

DSPs	81 (34%)
LUTs	15359 (22%)
Registers	403 (<1%)

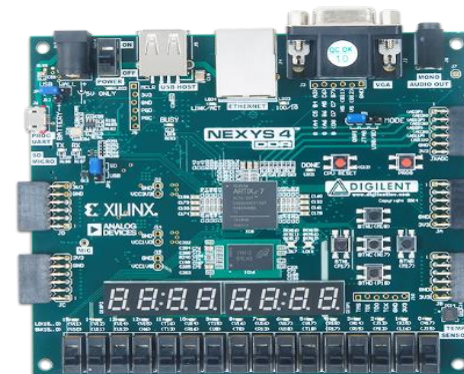


Figura 22 Nexys 4 DDR Artix-7

# Resultados: ANN con Pavlov's dog

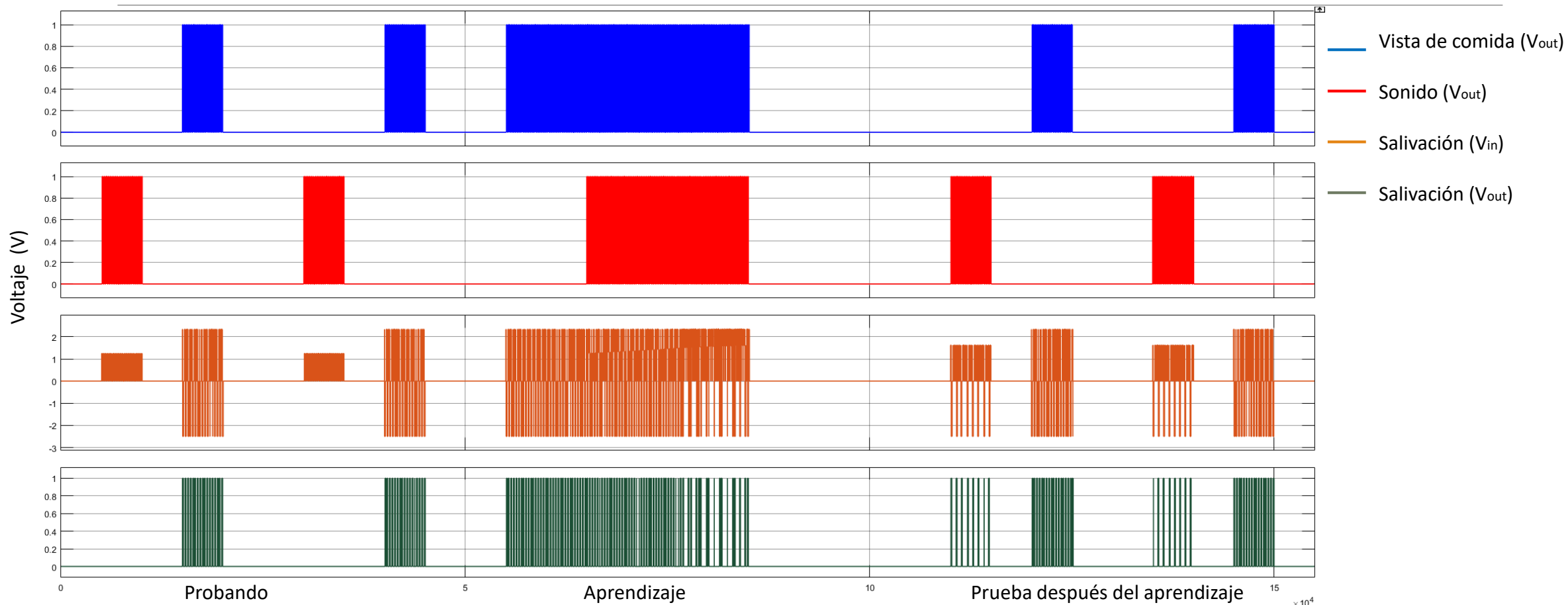


Figura 23 Resultados del experimento Pavlov's dog obtenidos en el FPGA

# Resultados: ANN con Pavlov's dog

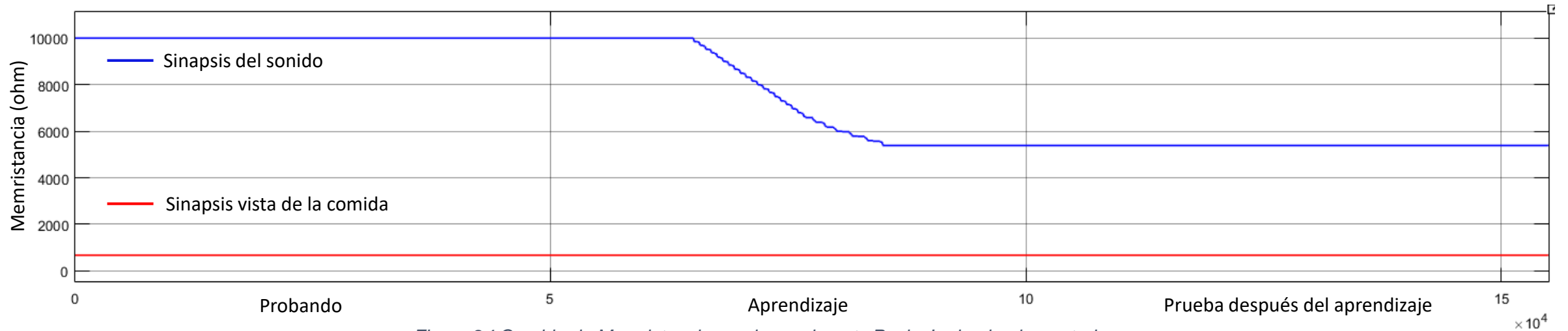


Figura 24 Cambio de Memristancia en el experimento Pavlov's dog implementado

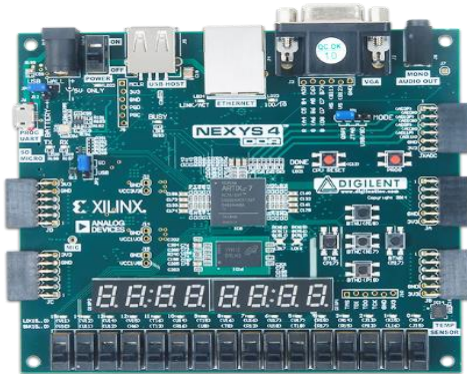
Valores utilizados en la ANN

$$a = 0 \frac{\Omega}{V \cdot s}, b = -15\,000 \frac{\Omega}{V \cdot s}, \Delta t = 0.0001s, V_T = 4V, R_{min} = 675\Omega, R_{max} = 10k\Omega, R_{init\ sonido} = R_{max}, R_{init\ comida} = R_{min}$$

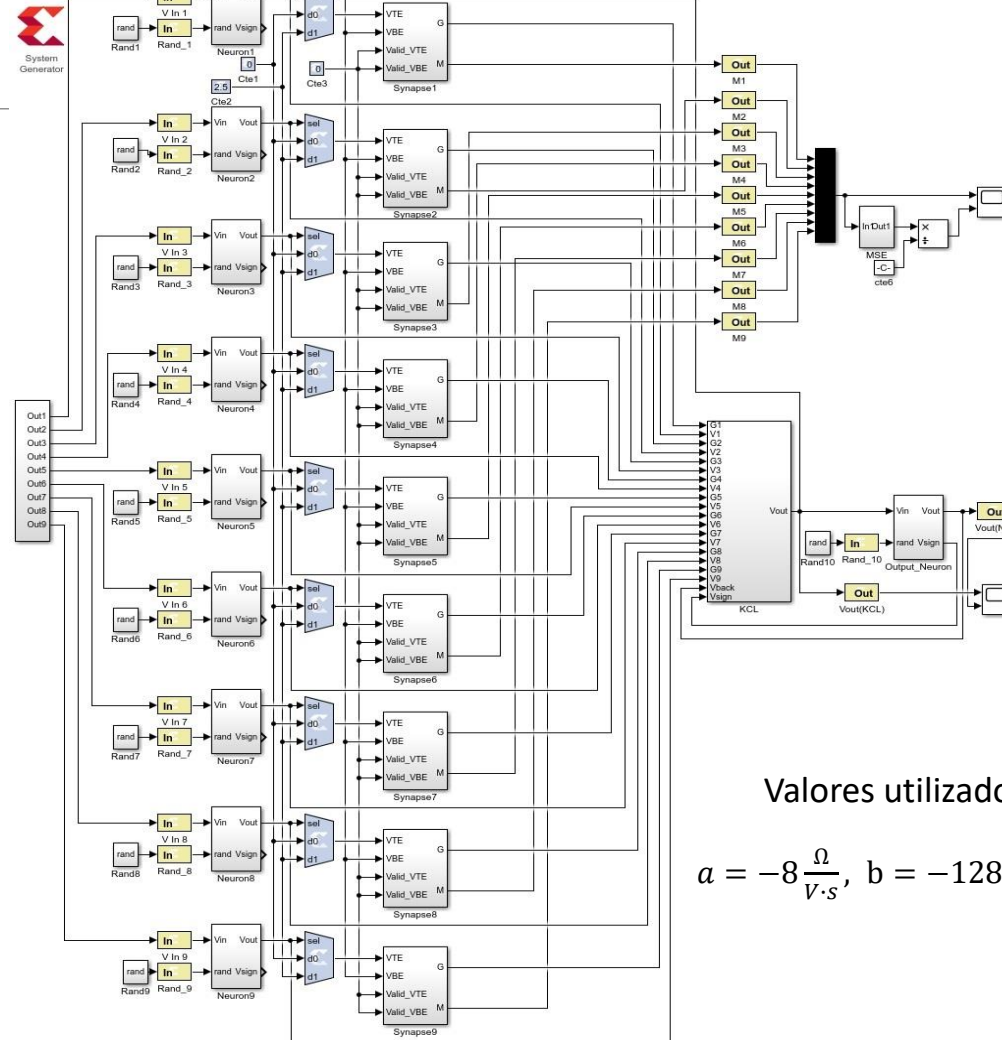
# Resultados: Reconocimiento de caracteres en una matriz dinámica

Implementación matriz 3x3 en FPGA

Nexys 4 DDR Artix-7	
DSPs	194 (81%)
LUTs	33075 (52%)
Registers	954 (<1%)



Nexys 4 DDR Artix-7



Valores utilizados en la ANN

$$a = -8 \frac{\Omega}{V \cdot s}, \quad b = -128 \frac{\Omega}{V \cdot s},$$



# Resultados: Reconocimiento de caracteres en una matriz dinámica



Figura 25 Alfabeto y números en matriz 3x3

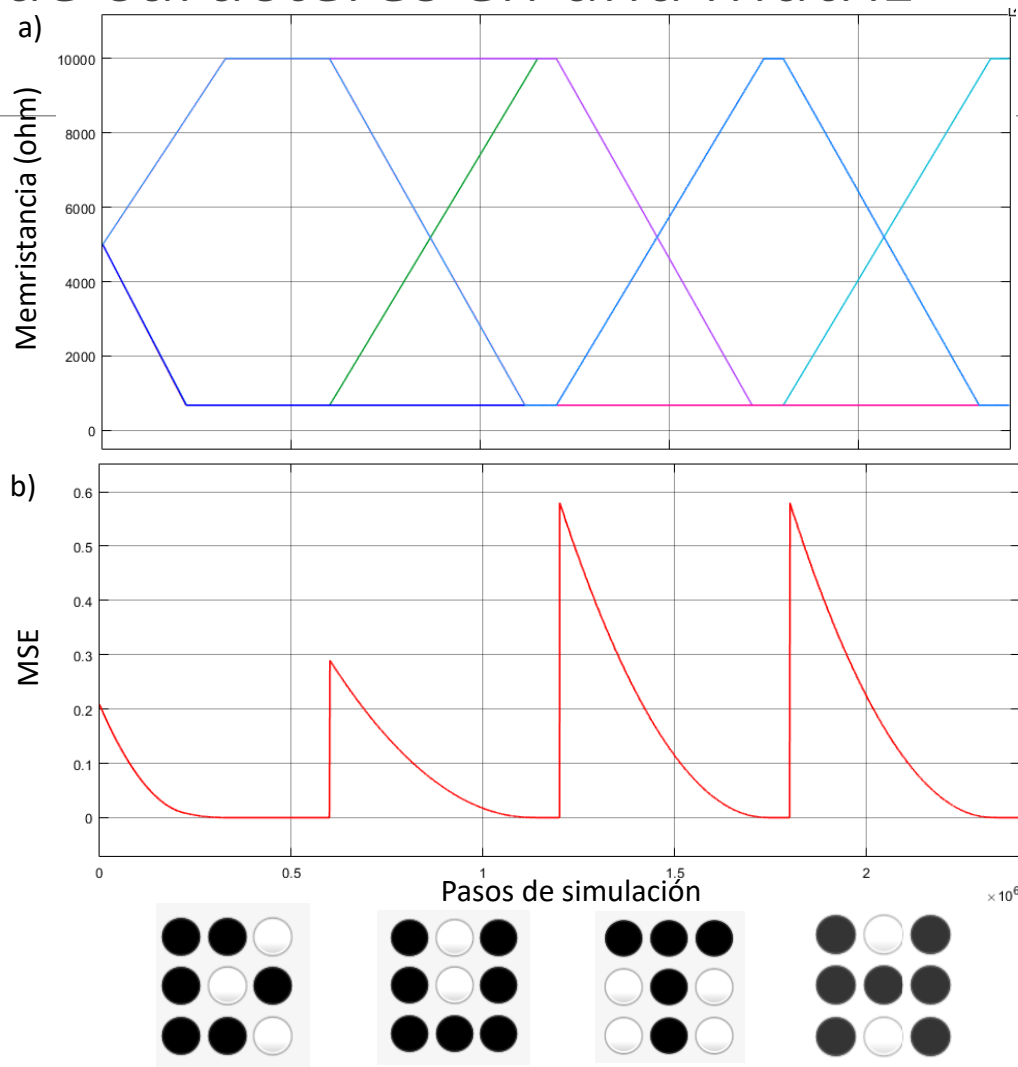


Figura 26 Resultados a) Cambio de pesos sinápticos (Memristancia), b) MSE al reconocer los caracteres

# Resultados: Uso del memristor en redes para realizar un análogo del algoritmo de hormigas

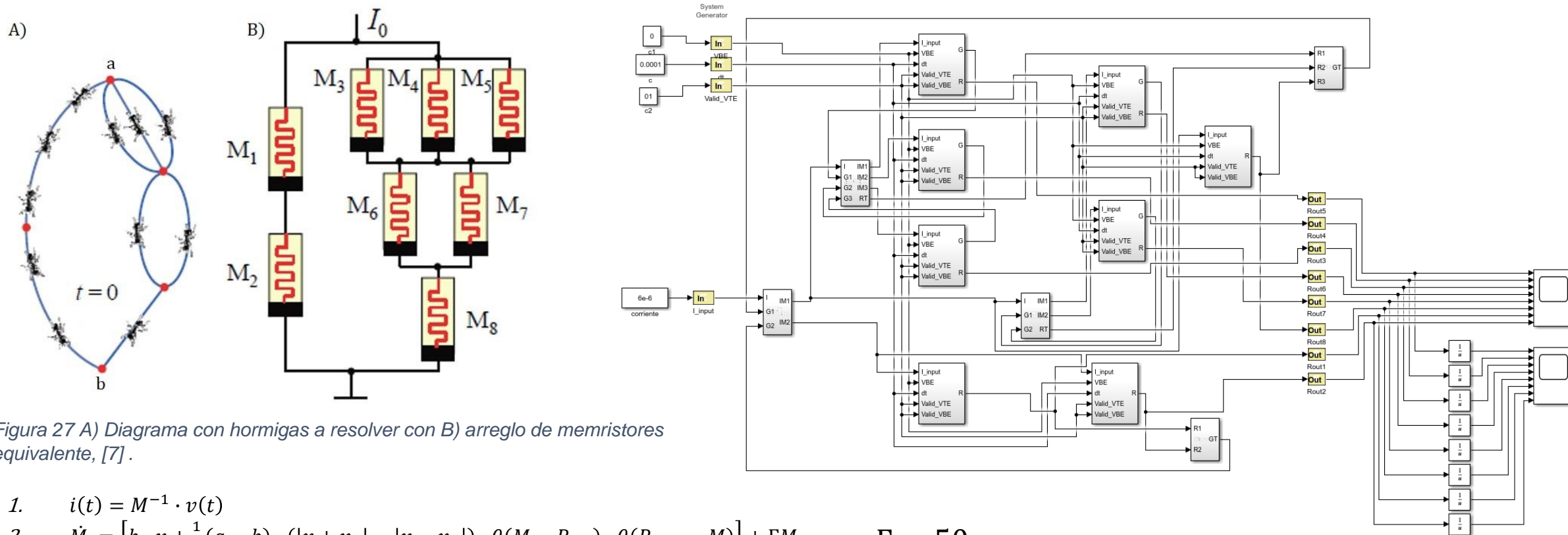


Figura 27 A) Diagrama con hormigas a resolver con B) arreglo de memristores equivalente, [7].

1.  $i(t) = M^{-1} \cdot v(t)$
2.  $\dot{M} = \left[ b \cdot v + \frac{1}{2}(a - b) \cdot (|v + v_T| - |v - v_T|) \cdot \theta(M - R_{ON}) \cdot \theta(R_{OFF} - M) \right] + \Gamma M \quad \Gamma = 50$

# Resultados: Uso del memristor en redes para realizar un análogo del algoritmo de hormigas

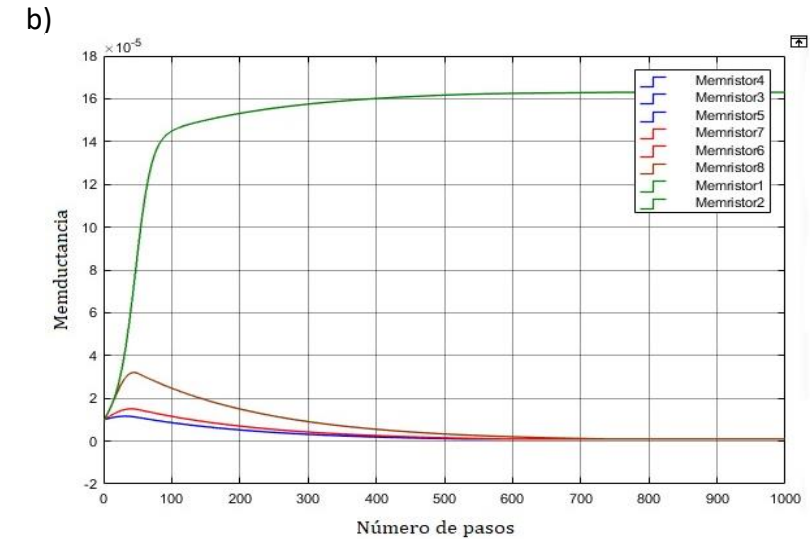
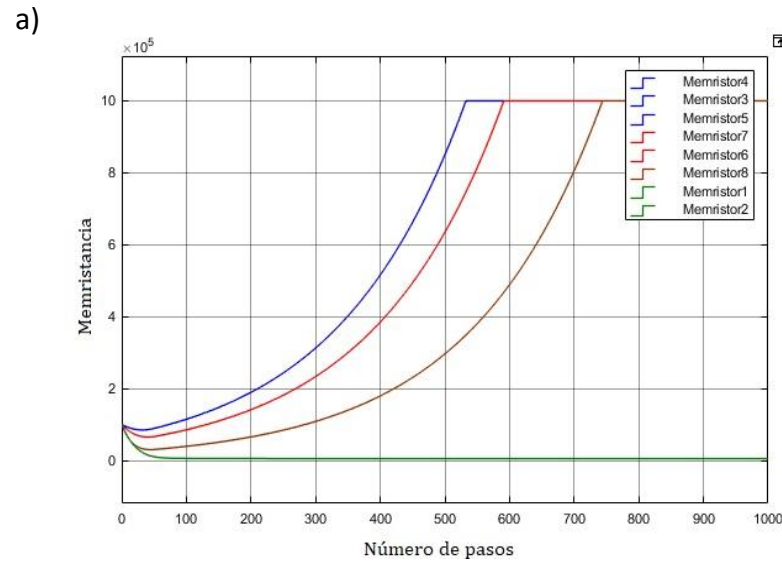
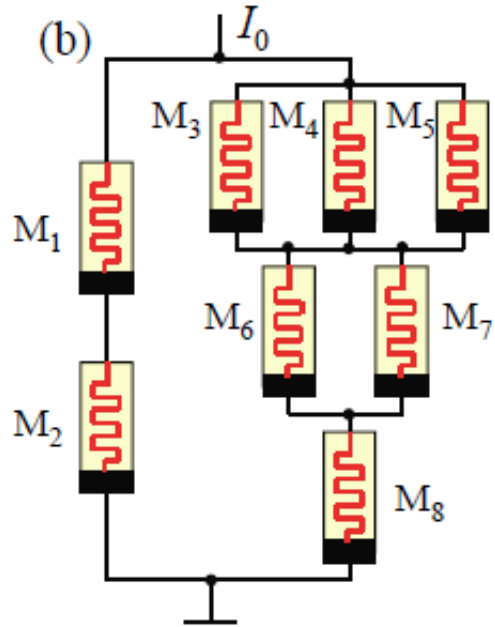


Figura 28 a) Cambio de Memductancia, b) Cambio de Memristancia

$$a = 0 \frac{\Omega}{V \cdot s}, b = -190000 \frac{\Omega}{V \cdot s}, \Delta t = 0.0001s, V_T = 0.035V, R_{min} = 400\Omega, R_{max} = 1M\Omega, R_{init} = 100K\Omega.$$

Nexys 4 DDR Artix-7	
DSPs	240(100%)
LUTs	34165(54%)
Registers	638(>1%)

# Resultados: Reconocimiento de bordes en imágenes

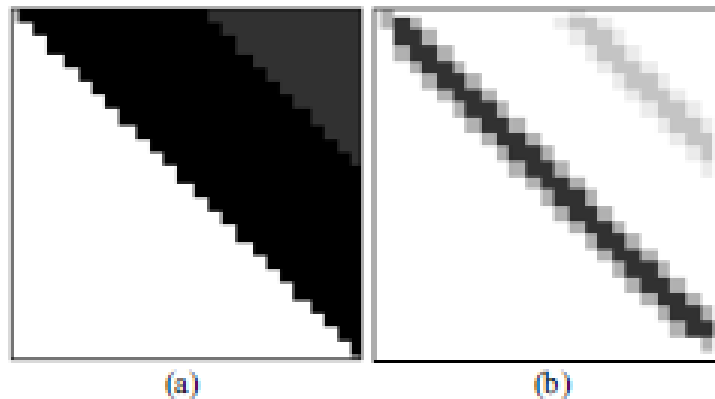


Figura 29 Imagen con bordes a) original, b) detección de bordes

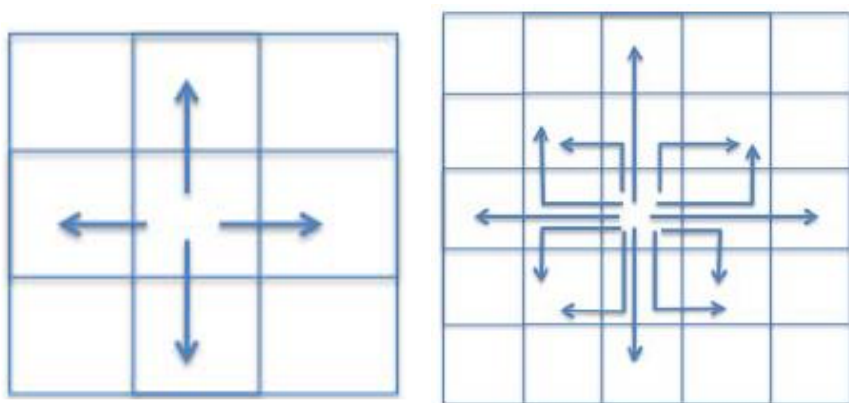


Figura 30 Posibles caminos a recorrer por las hormigas [8]

$$1).- \eta(i,j) = \frac{1}{I_{\text{Max}}} \left[ |I(i, j - 1) - I(i, j + 1)| + |I(i - 1, j) - I(i + 1, j)| \right]$$

$$2).- p_{\text{path}_m} = \frac{\prod_{(i_t, j_t) \in \text{path}_m} \tau_{(i_t, j_t)}^\alpha (1/L_{\text{path}_m})^\beta}{\sum_{f=1}^M \prod_{(i_t, j_t) \in \text{path}_f} \tau_{(i_t, j_t)}^\alpha (1/L_{\text{path}_f})^\beta}$$

$$L_{\text{path}_m} = \sum_{(i_t, j_t) \in \text{path}_m} \eta_{(i_t, j_t)}^{-1}$$

$$3).- \tau_{(i,j)}(k + 1) = (1 - \rho)\tau_{(i,j)} + \frac{vQ}{L_{\text{path}_m}}$$

# Resultados: Reconocimiento de bordes en imágenes

1).- 
$$\eta(i,j) = \frac{1}{I_{Max}} \left[ \begin{array}{l} |I(i,j-1) - I(i,j+1)| \\ + |I(i-1,j) - I(i+1,j)| \end{array} \right]$$

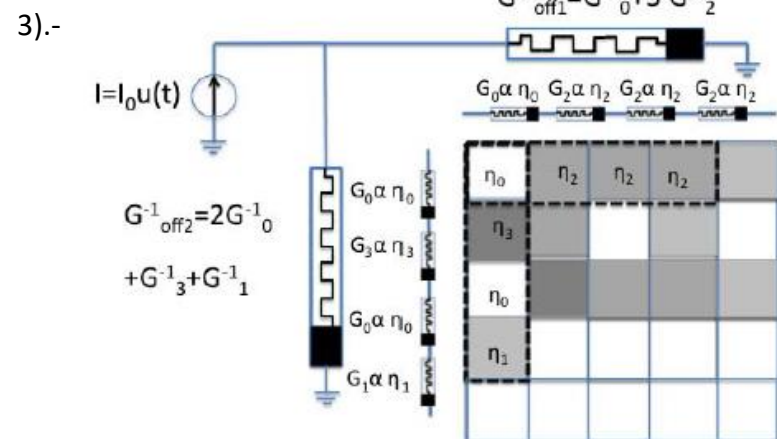


Figura 31 Simulación de los caminos a recorrer con Memristores tomado de [8]

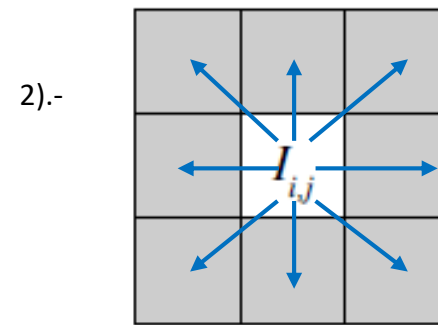


Figura 32 Caminos propuestos a recorrer



Figura 33 Imagen elegida para realizar el algoritmo

# Resultados: Reconocimiento de bordes en imágenes



Figura 34 Imagen original



Figura 35 Feromonas, 7t, ini2, 2s

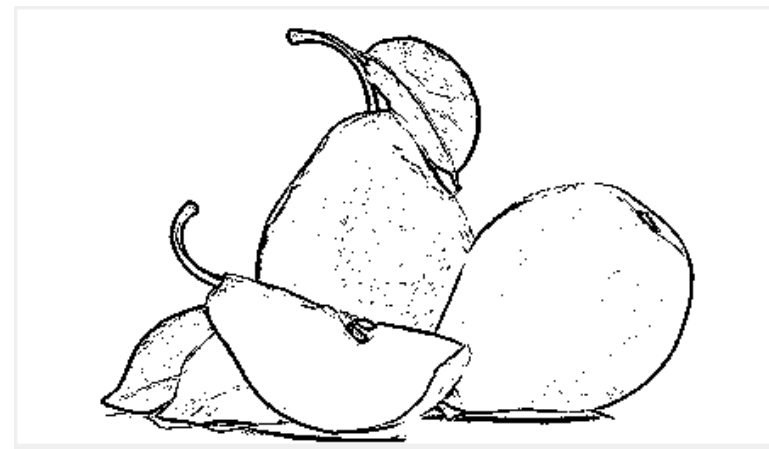


Figura 36 Memristores 0.01t (100t). 3uA, ini2, 41s

$$a = 0 \frac{\Omega}{V \cdot s}, b = -196\,000\,000 \frac{\Omega}{V \cdot s}, \Delta t = 0.0001s, V_T = 0.035V, R_{min} = 400\Omega, R_{max} = 1M\Omega, \Gamma = 50, I = 3\mu A.$$

# Resultados: Reconocimiento de bordes en imágenes



Figura 37 Imagen Lenna original



Figura 38 Feromonas, 7t, ini1, 2s



Figura 39 Feromonas, 7t, ini2, 2s



Figura 40 Memristores  
0.01t.(100t) 3uA, ini2. 18s



Figura 41 Memristores  
0.01t.(100t) 6uA, ini2, 18s

# Resultados: Reconocimiento de bordes en imágenes



Figura 42 Imagen original



Figura 43 Feromonas, 7t, ini2, 3s



Figura 44 Memristores 0.01t(100t), 3uA,  
ini2, 62s

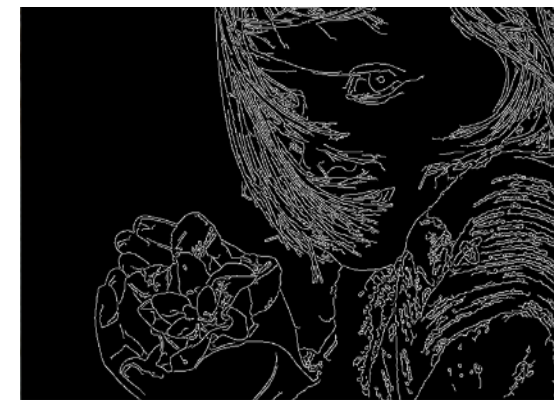


Figura 45 Algoritmo Canny, 3s



# Conclusiones

---

- Se concluye que el modelo del memristor bipolar con umbral controlado por voltaje es viable para utilizarse en otras implementaciones, ya que demostró ser capaz de ajustarse a los distintos usos que se le dieron.
- Dada la versatilidad del modelo del memristor, este resulta útil para ser modificado y obtener comportamientos mas complejos del memristor.
- Los recursos utilizados por el memristor pueden ser optimizados dependiendo de la aplicación.

# Conclusiones

---

- La neurona simplificada funciona correctamente, y es posible optimizarla para que los recursos consumidos del FPGA sean mínimos sin perder sus características.
- Se comprobó la propuesta del memristor como análogo del algoritmo de hormigas.
- Se obtuvieron mejores resultados del procesamiento de imágenes con memristores vs el algoritmo de hormigas clásico.

# Referencias

---

- [1] D. B. Strukov, G. S. Snider, D. R. Stewart and R. S. Williams, “*The Missing Memristor Found*”, Nature 453, 80-83, 2008.
- [2] V. Ntinias, I. Vourkas, A. Abusleme, G. CH. Sirakoulis and A. Rubio, “*Experimental Study of Artificial Neural Networks Using a Digital Memristor Simulator*”, Transactions on Neural Networks and Learning Systems, VOL. 29, NO. 10, 2018.
- [3] Y. V. Pershin, S. La Fontaine and M. Di Ventra, “*Memristive Model of Amoeba Learning*”, Physical Review E **80**, 021926 2009.
- [4] T. Saigusa, “*Amoebae Anticipate Periodic Events*”, Physical Review Letters ,January 2008.
- [5] B. Linares-Barranco and T. Serrano-Gotarredona “*Memristance Can Explain Spike-Time-Dependent-Plasticity in Neural Synapses*”, NaturePrecedings, 2009.
- [6] P. Siarry (Editor), “*Metaheuristics*”, Springer International Publishing, Switzerland, 2016.
- [7] Y. V. Pershin and M. Di Ventra, “*Memcomputing and Swarm Intelligence*”, eprint arXiv:1408.6741, 2014.
- [8] Z. Pajouhi and K. Roy, “*Image Edge Detection Based on Swarm Intelligence Using Memristive Networks*”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 9, pp. 1774-1787, 2018
- D. Biolek and Z. Biolek, “*About Fingerprints of Chua's Memristors*” IEEE Circuits and Systems Magazine, vol. 18, no. 2, pp. 35-47, 2018.

# Datos

---

$\mu v$ : movilidad iónica promedio  $10^{-14} \text{ m}^2/\text{S}^{-1}\text{V}^{-1}$

Lenna 220x220

Canny 510x341

Peras 450x253