



Centro de Investigación y de Estudios Avanzados del I.P.N.

Sección de Electrónica del Estado Sólido

Prototipo en FPGA de Emulador de Red Neuronal Pulsada

Presenta:

ING. ERIK JONATAN MORALES DE LA ROSA

Para obtener el grado de

MAESTRO EN CIENCIAS

DIRECTORES DE TESIS:

DR. FELIPE GÓMEZ CASTAÑEDA

DR. JOSÉ ANTONIO MORENO CADENAS



Objetivos

GENERAL

1. Implementar en FPGA un emulador de red neuronal pulsada para el aprendizaje y reconocimiento de caracteres simples.

PARTICULARES

1. Proponer un algoritmo no supervisado basado en STDP en software.
2. Proponer e implementar una arquitectura de hardware digital en FPGA, que incorpore las principales características de los sistemas neuromórficos digitales actuales.



Contenido

1.- Marco Teórico

- Redes Neuronales Artificiales
- Neurona
- Sinapsis
- STDP



2.-Implementación en Software

- Algoritmo
- Resultados



3.-Implementación en Hardware

- Arquitectura
- Resultados



4.- Conclusiones

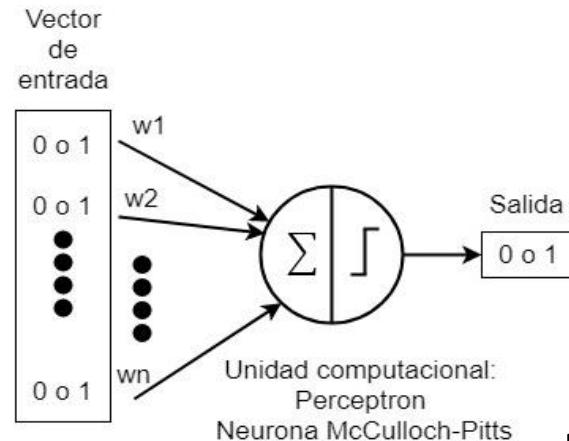
- Trabajo Futuro



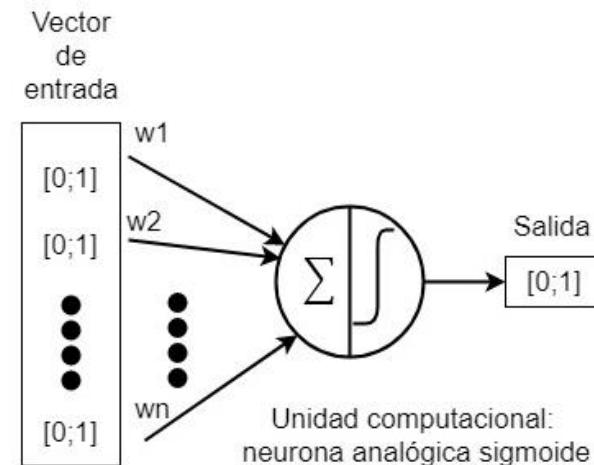


Redes Neuronales Artificiales

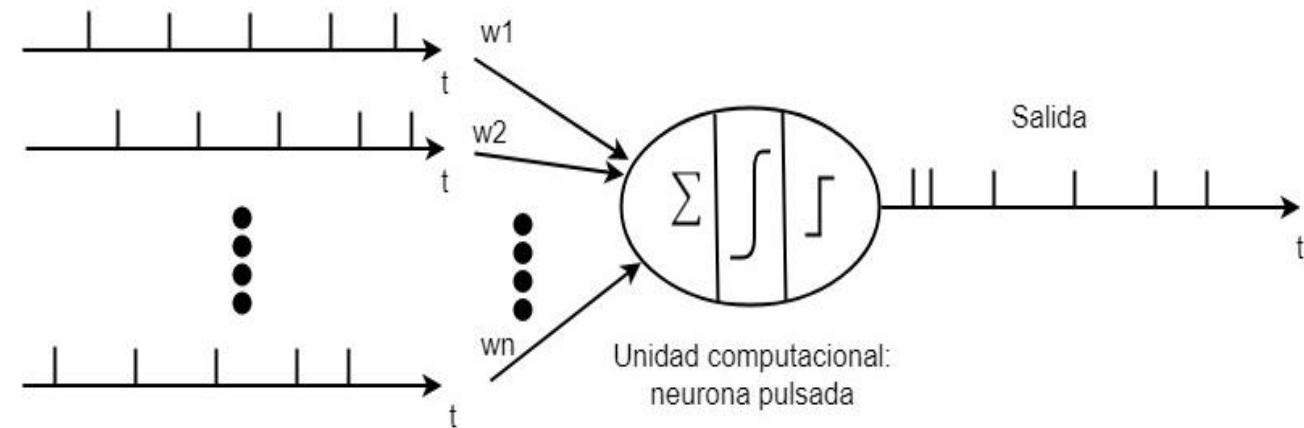
Primera Generación



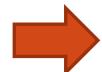
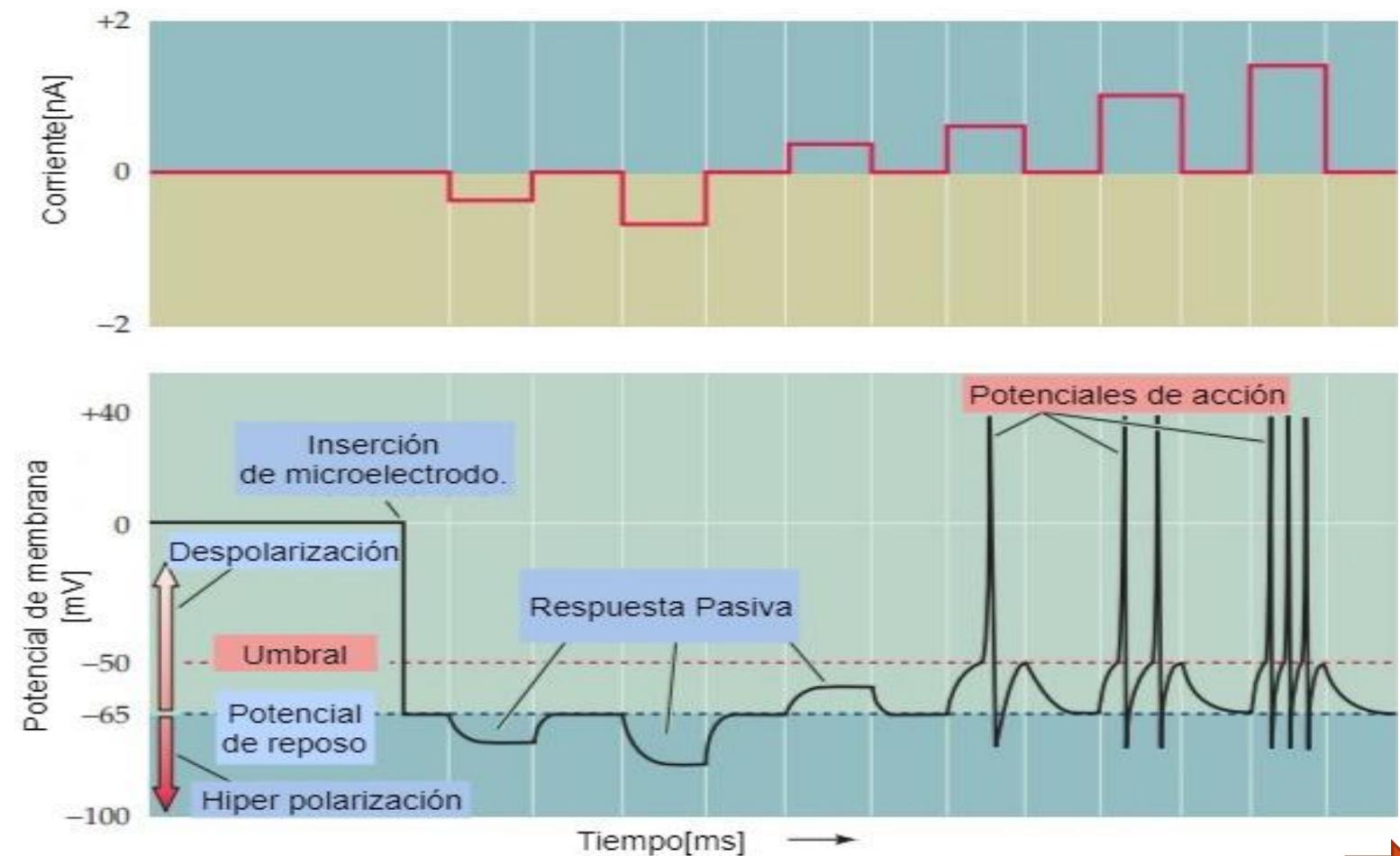
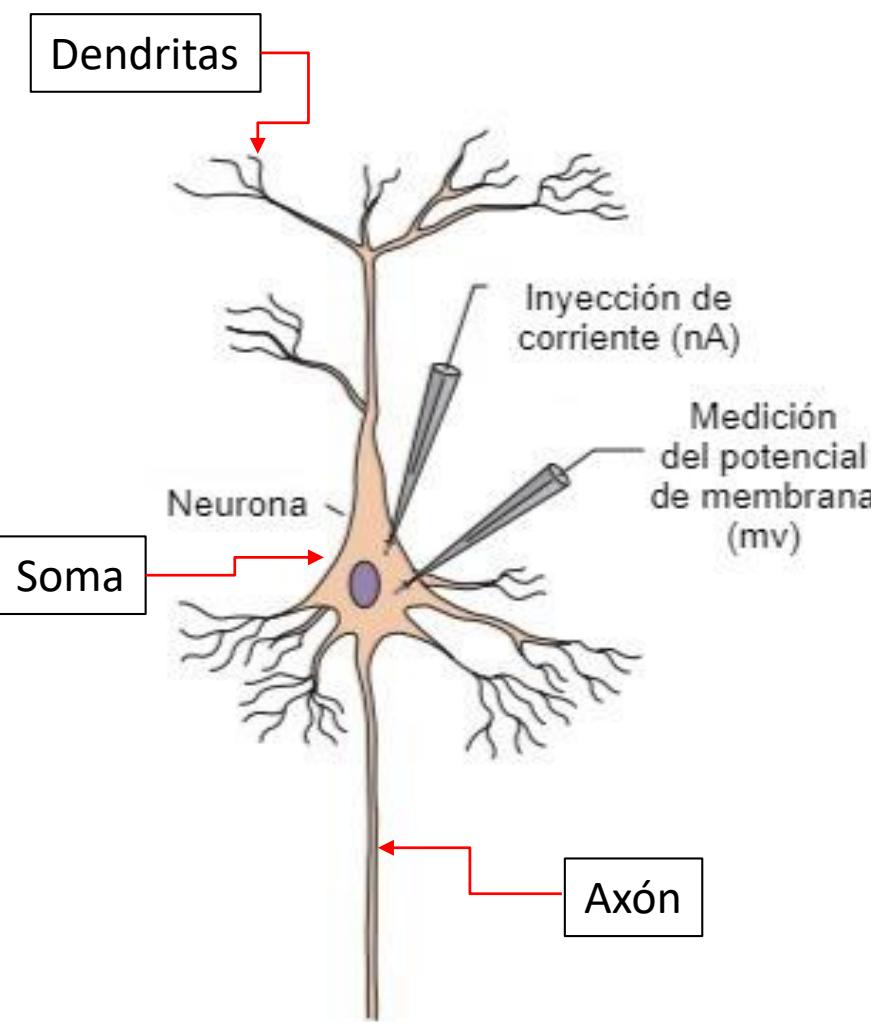
Segunda Generación



Tercera Generación

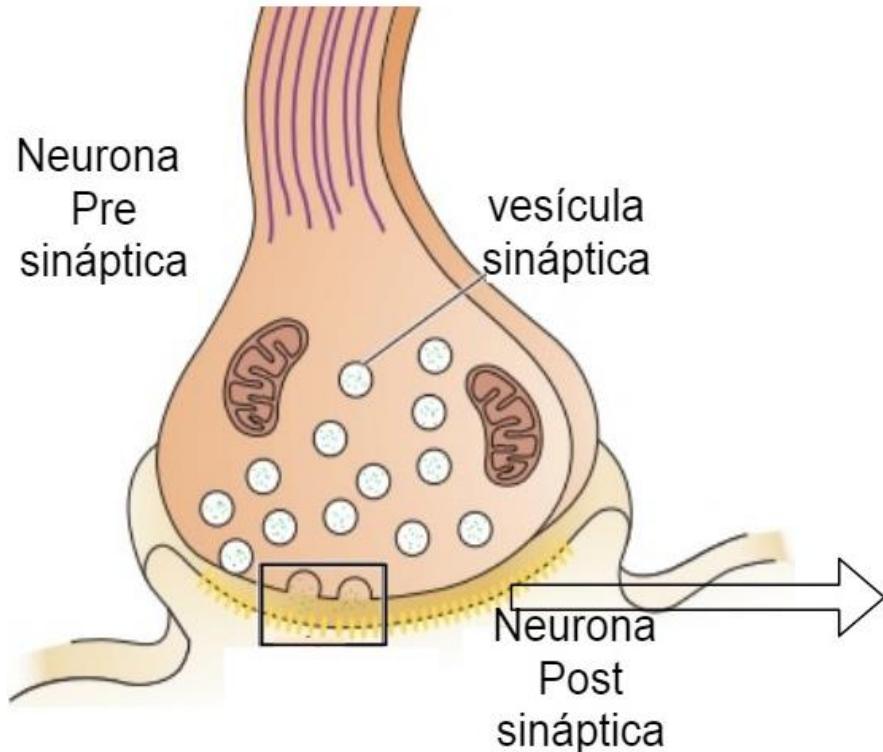


Comportamiento Biológico Neuronal

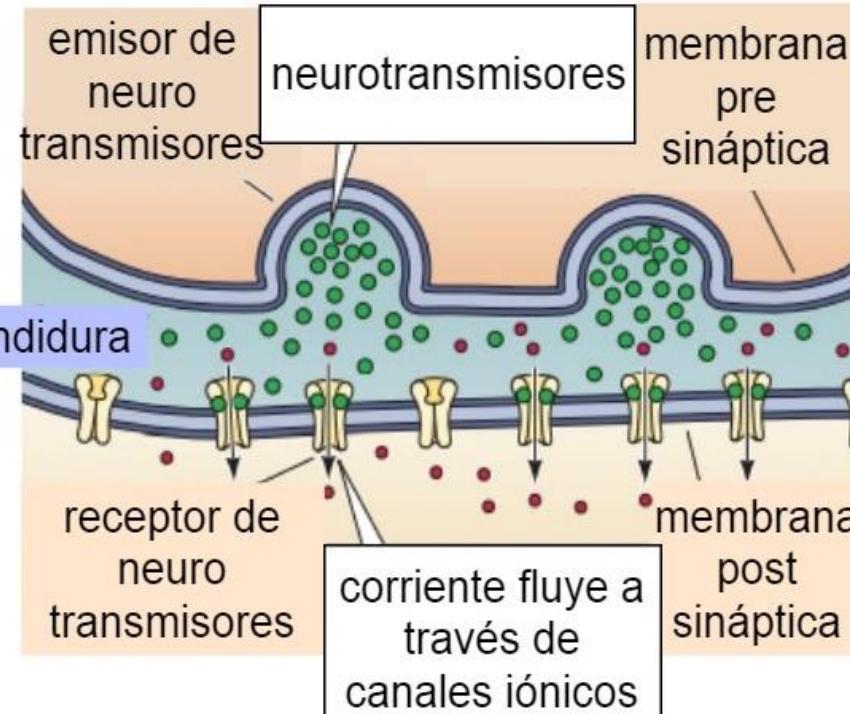


Comportamiento Biológico de la Sinapsis Química

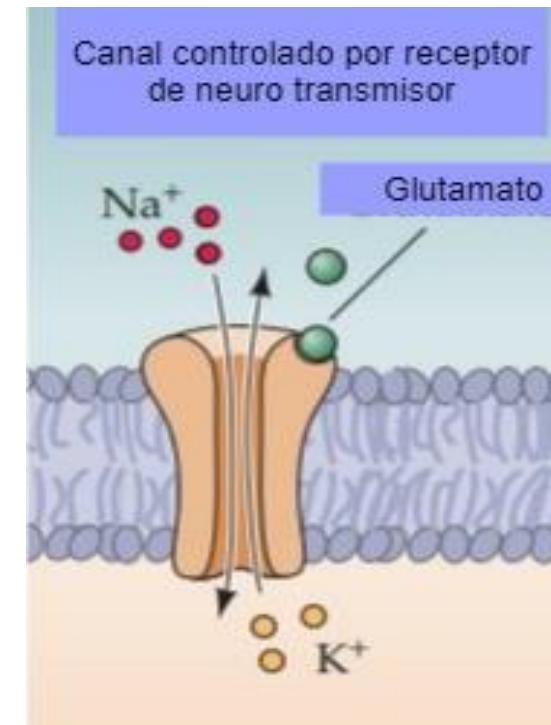
Comunicación



Modulada mediante



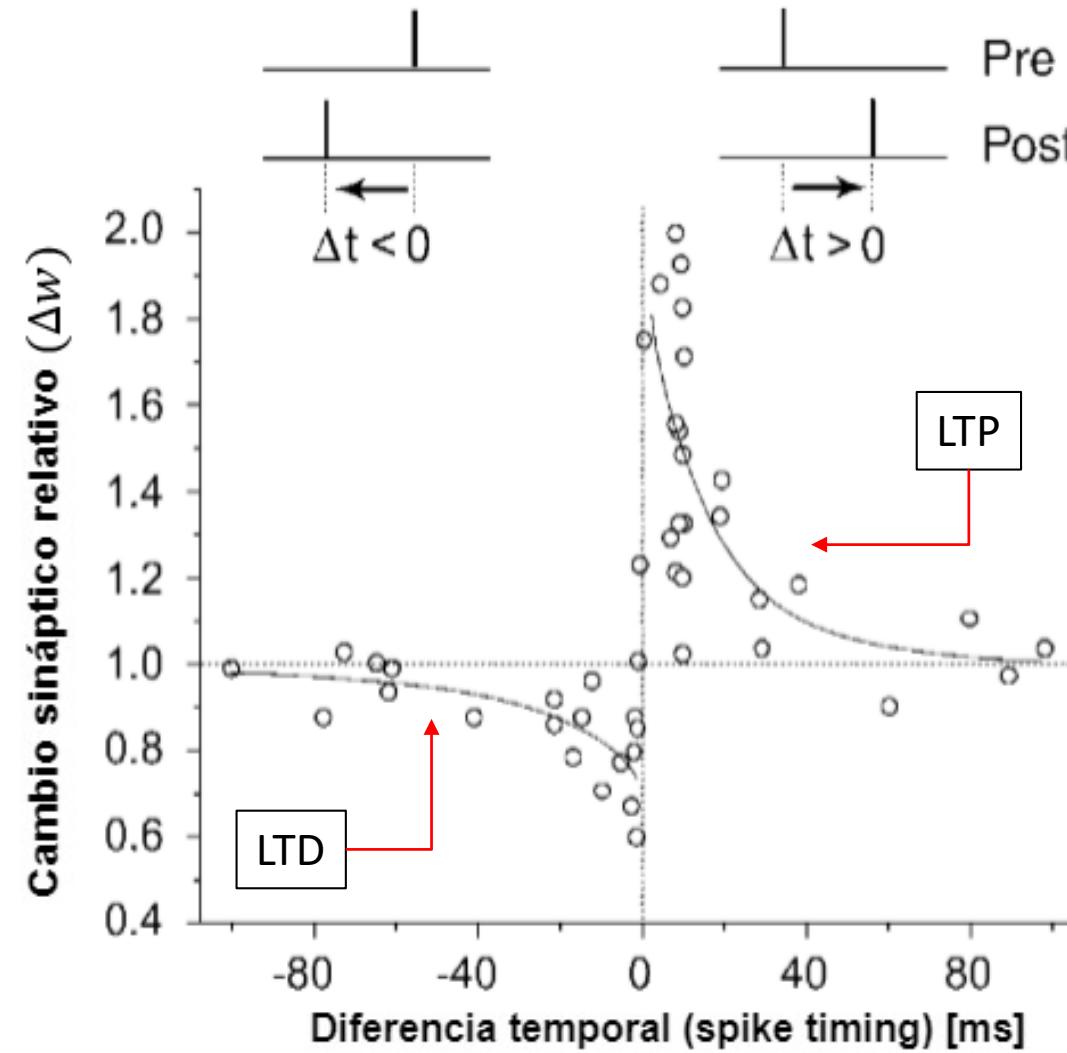
Corriente Sináptica variable



STDP (Spike Timing Dependent Plasticity)

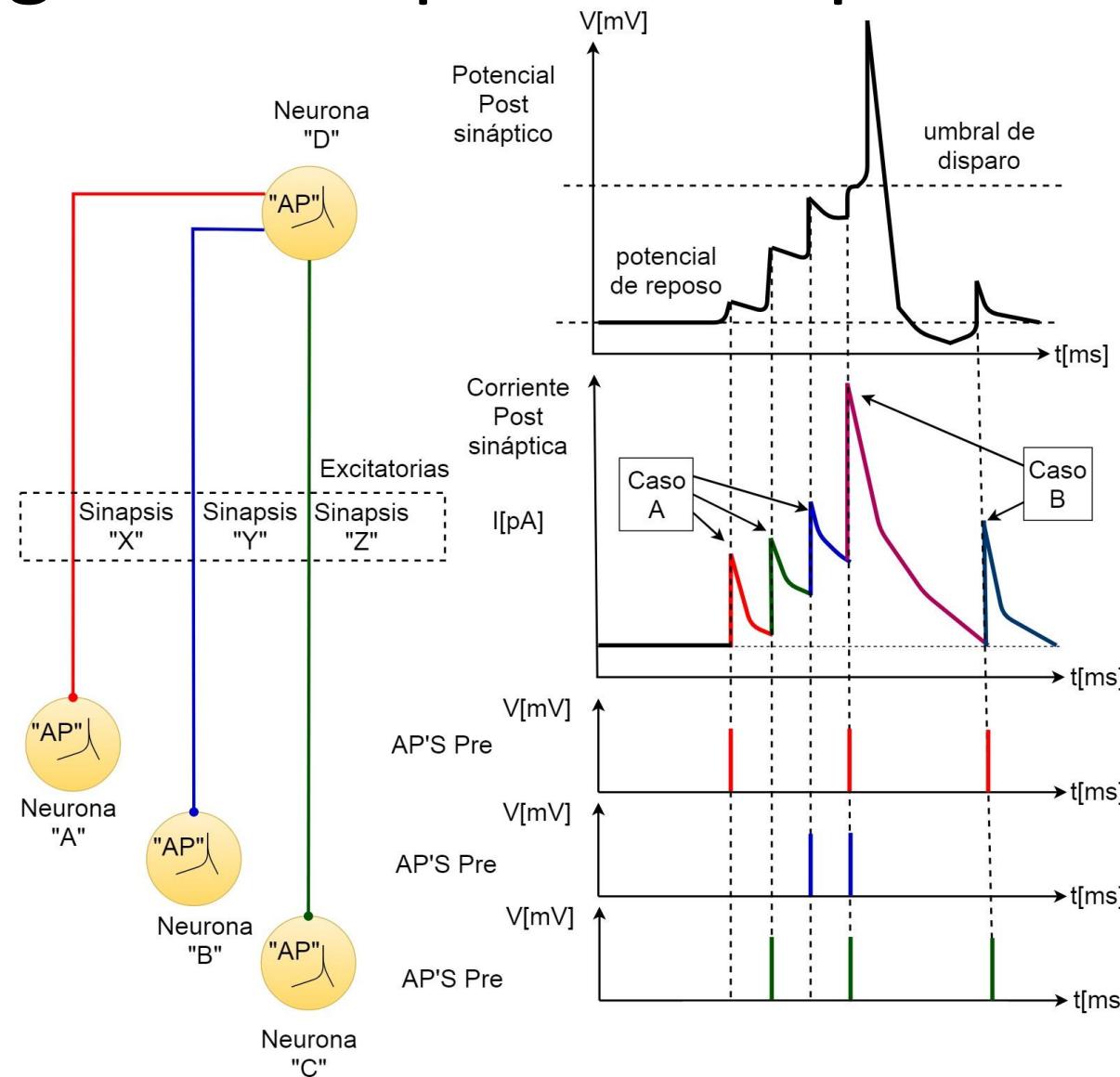


- Protocolo de asociación de pulsos
- Ventana de aprendizaje



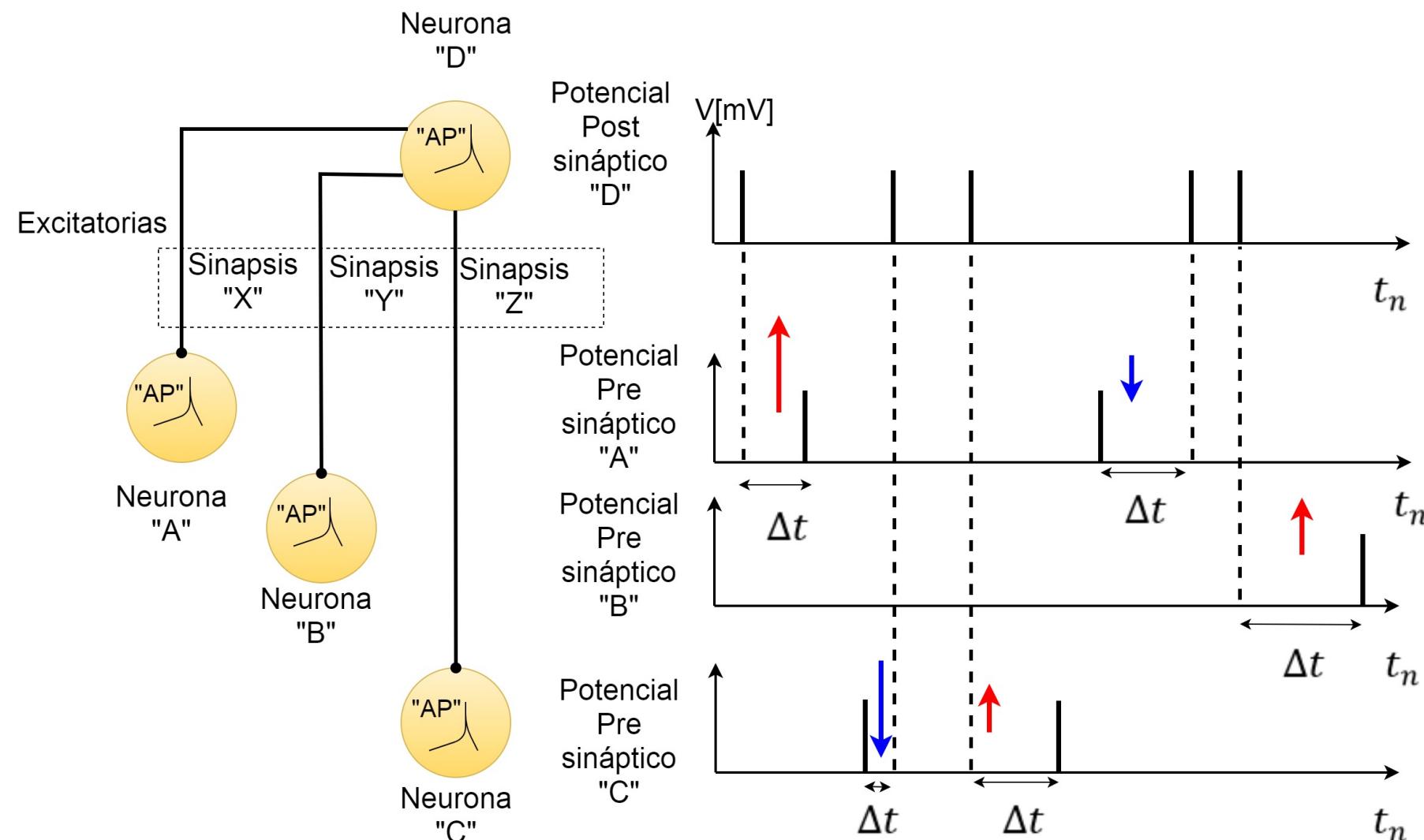


Integración Espacio-Temporal Sináptica



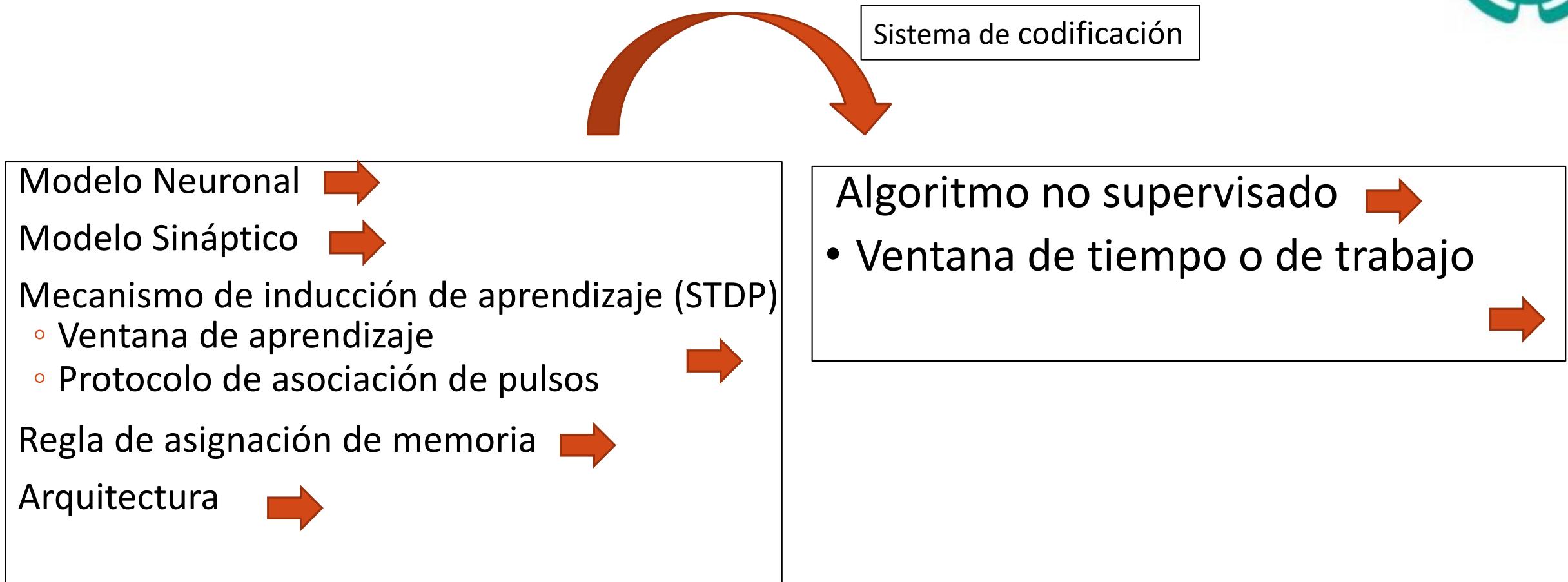


Integración Espacio-Temporal en STDP





Implementación en Software





Modelo Neuronal

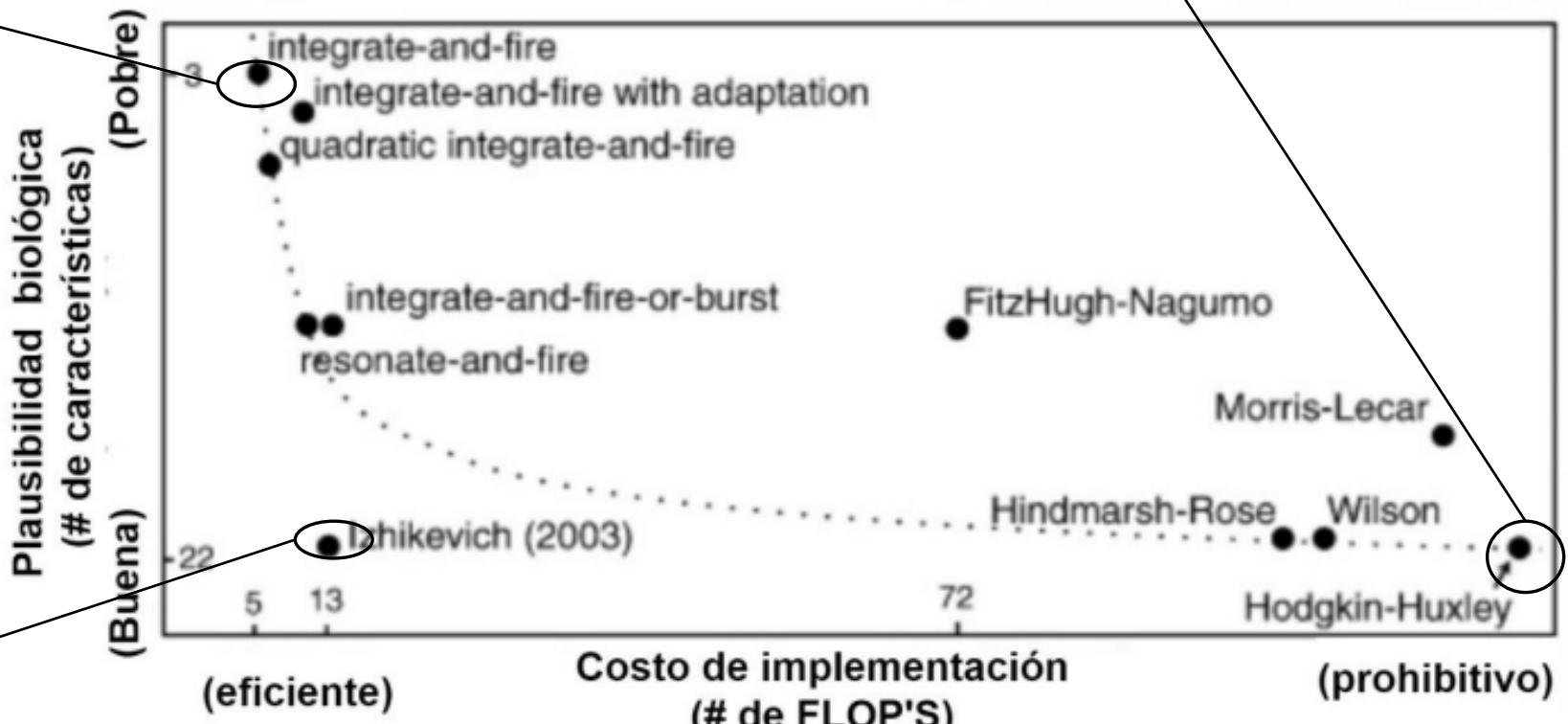
$$C_m \frac{dV_m}{dt} = -g_{Na} m^3 h(V_m)[V_m - E_{Na}] - g_K n^4 [V_m - E_K] - g_L [V_m - E_L] + I_m$$

$$C \frac{du}{dt} = -\frac{1}{R}(u - u_{rest}) + I(t)$$

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 + I$$

$$\frac{du}{dt} = a(bv - u)$$

Si $v = 30 \text{ mV}$ entonces $v = c, u = u + d$

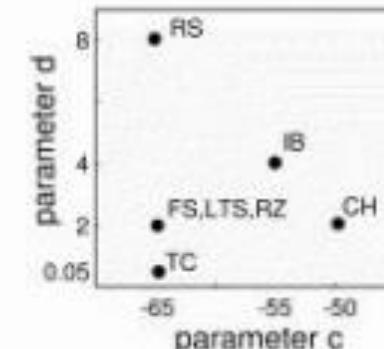
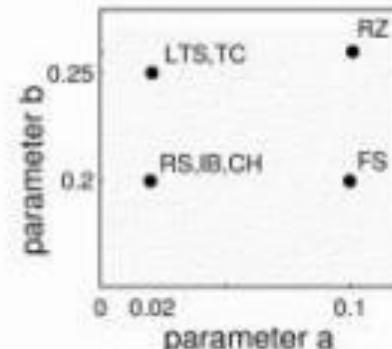
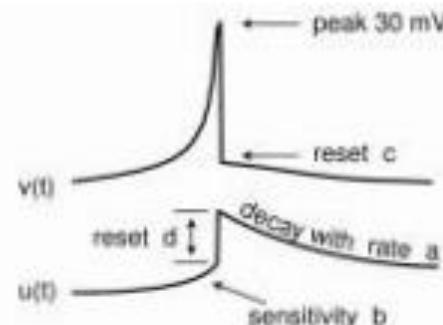




Modelo Neuronal

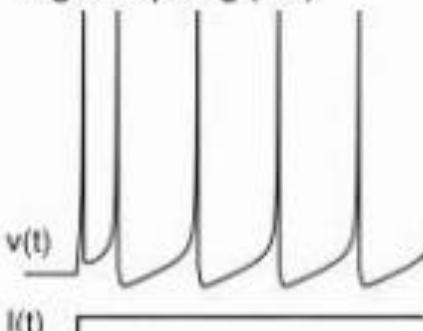
$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$

if $v = 30 \text{ mV}$,
then $v = c$, $u = u + d$

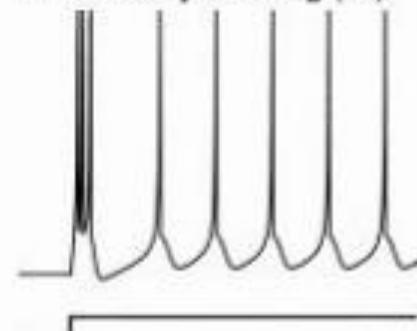


Parámetros	Valores
a	0.02
b	0.2
c	-65
d	8
dt	0.005 [ms]

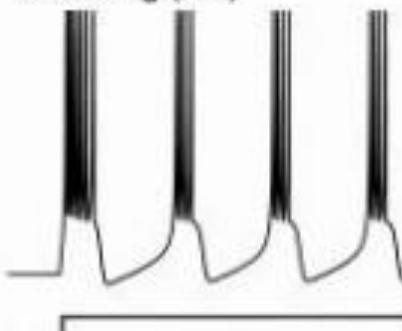
regular spiking (RS)



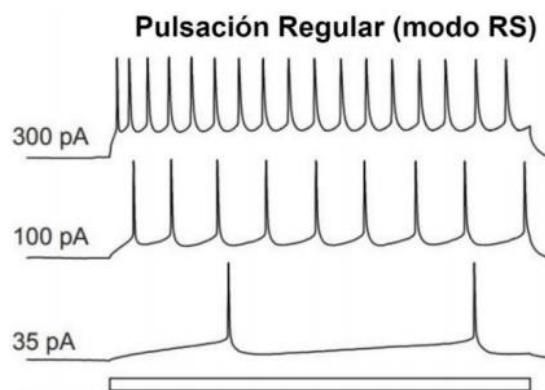
intrinsically bursting (IB)



chattering (CH)



fast spiking (FS)





Modelos Sinápticos

$$I_j(t) = q \sum_i w_{ij} \sum_k H(t - t_k^i)$$

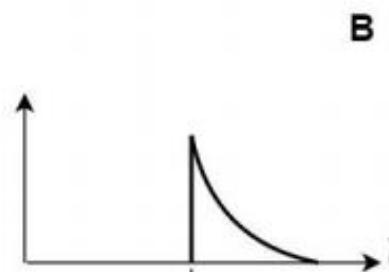
$$I_j(t) = \frac{q}{\tau_s} \sum_i w_{ij} \sum_k e^{\left(\frac{t-t_k^i}{\tau_s}\right)} H(t - t_k^i)$$

$$I_j(t) = \frac{q}{\tau_s - \tau_r} \sum_i w_{ij} \sum_k \left[e^{\left(\frac{t-t_k^i}{\tau_s}\right)} - e^{\left(\frac{t-t_k^i}{\tau_r}\right)} \right] H(t - t_k^i)$$

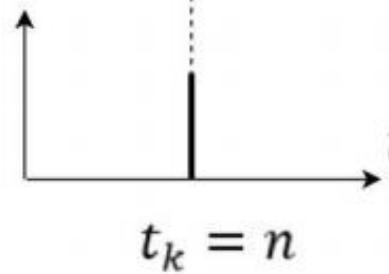


Corriente
post sináptica

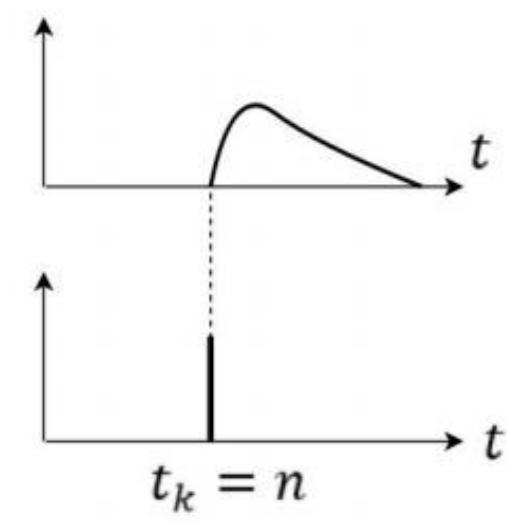
A



B



C



$$I_j(t) = \sum_i w_{ij} \sum_k H(t - t_k^i) [pA]$$

i-ésimos
"spikes"

$t_k = n$



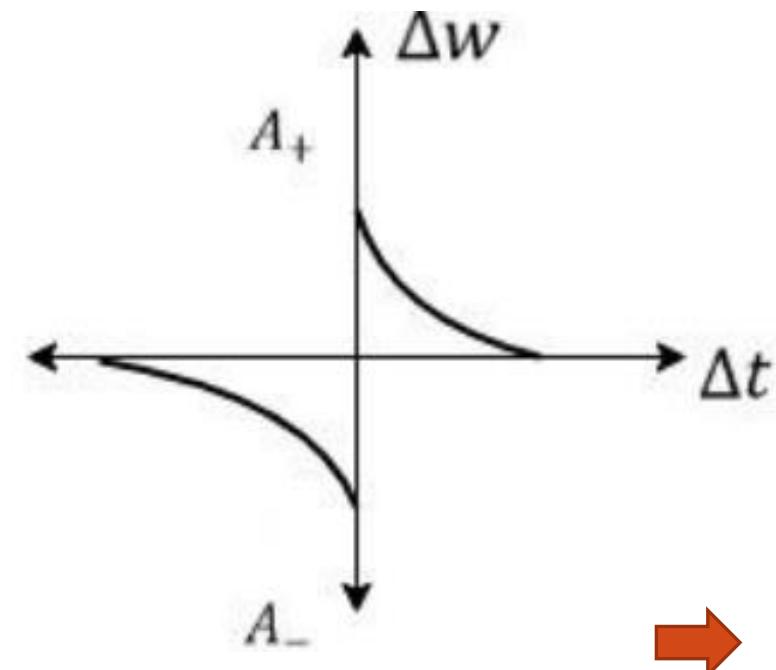
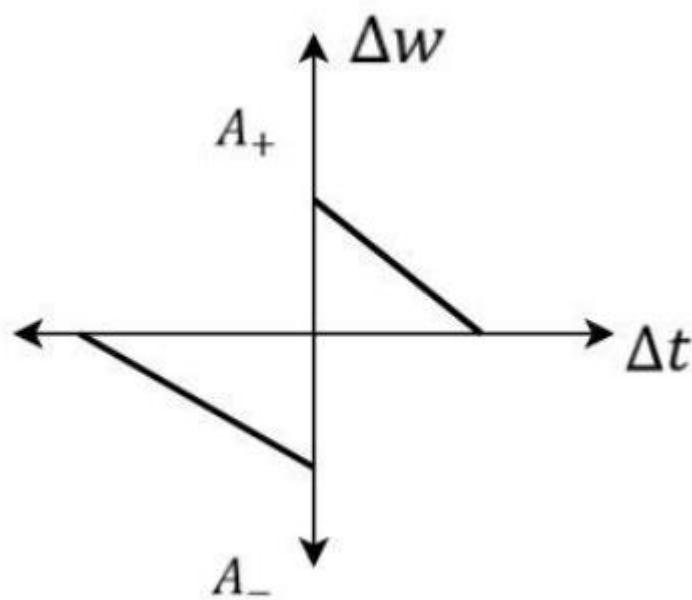
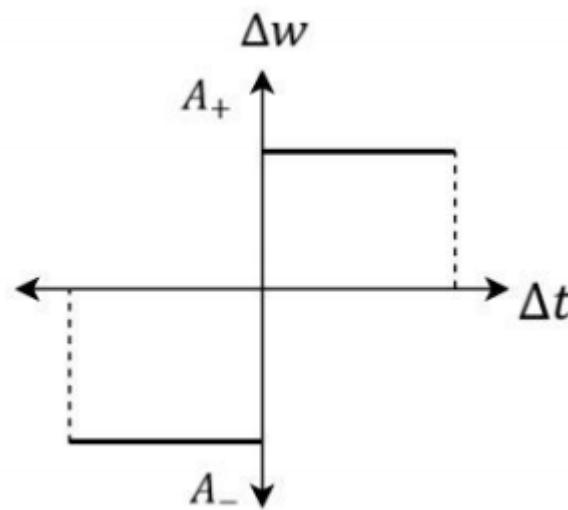
Ventanas de aprendizaje STDP



$$\Delta w = \begin{cases} +A_+ & \text{si } \Delta t \in [0, \tau_+] \\ -A_- & \text{si } \Delta t \in [-\tau_-, 0) \\ 0 & \text{de otra forma} \end{cases}$$

$$\Delta w = \begin{cases} +A_+ \left(1 - \frac{\Delta t}{\tau_+}\right) & \text{si } \Delta t \in [0, \tau_+] \\ -A_- \left(1 - \frac{\Delta t}{\tau_-}\right) & \text{si } \Delta t \in [-\tau_-, 0) \\ 0 & \text{de otra forma} \end{cases}$$

$$\Delta w = \begin{cases} +A_+ e^{\left(\frac{-\Delta t}{\tau_+}\right)} & \text{si } \Delta t \geq 0 \\ -A_- e^{\left(\frac{\Delta t}{\tau_-}\right)} & \text{si } \Delta t < 0 \end{cases}$$



Protocolo de pares en software (Todos contra todos)



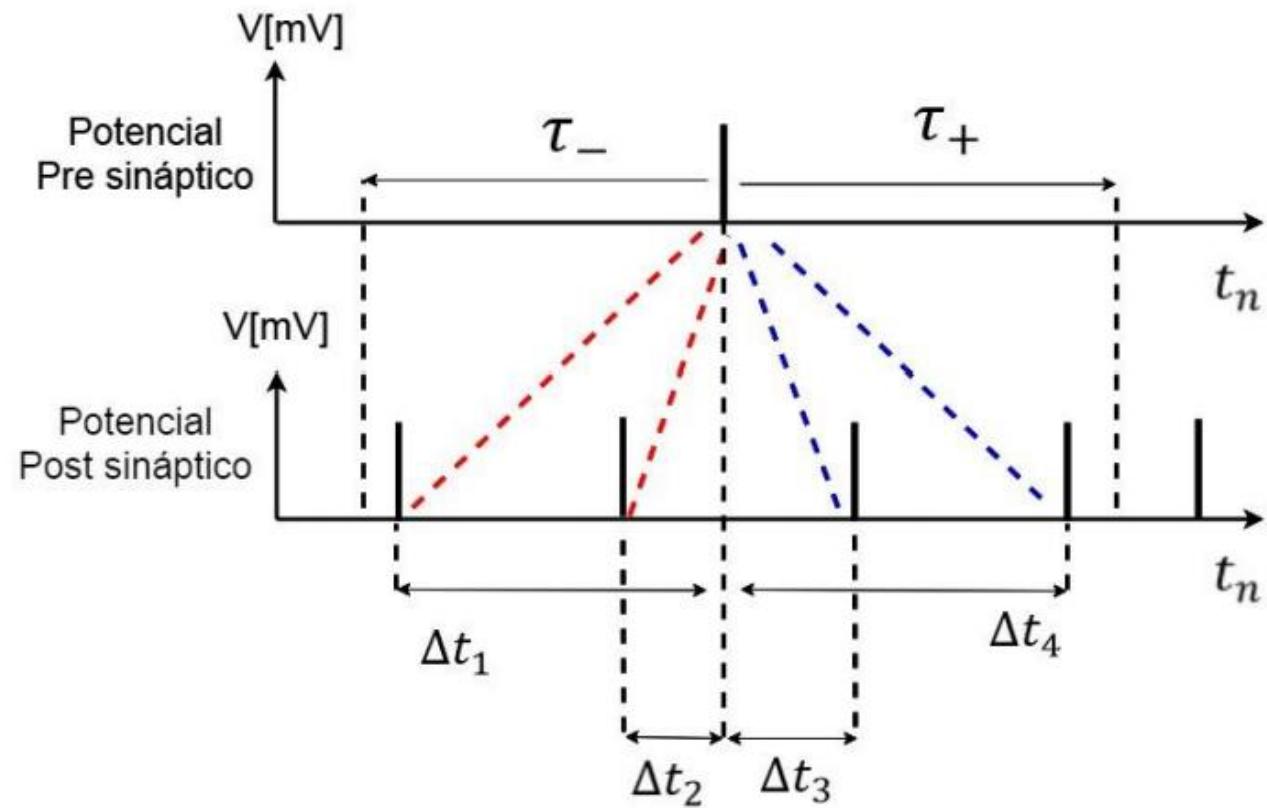
$$\Delta t = t^{post} - t^{pre}$$

$$W(\Delta t) = \begin{cases} +A_+ e^{\left(\frac{-\Delta t}{\tau_+}\right)} & \text{si } \Delta t \geq 0 \\ -A_- e^{\left(\frac{\Delta t}{\tau_-}\right)} & \text{si } \Delta t < 0 \end{cases}$$

$$\Delta w_{ij} = \sum_{f=1}^N \sum_{n=1}^N W(t_j^f - t_i^n)$$

Parámetros	Valores
A_+	0.10
A_-	0.12
τ_+	20
τ_-	20

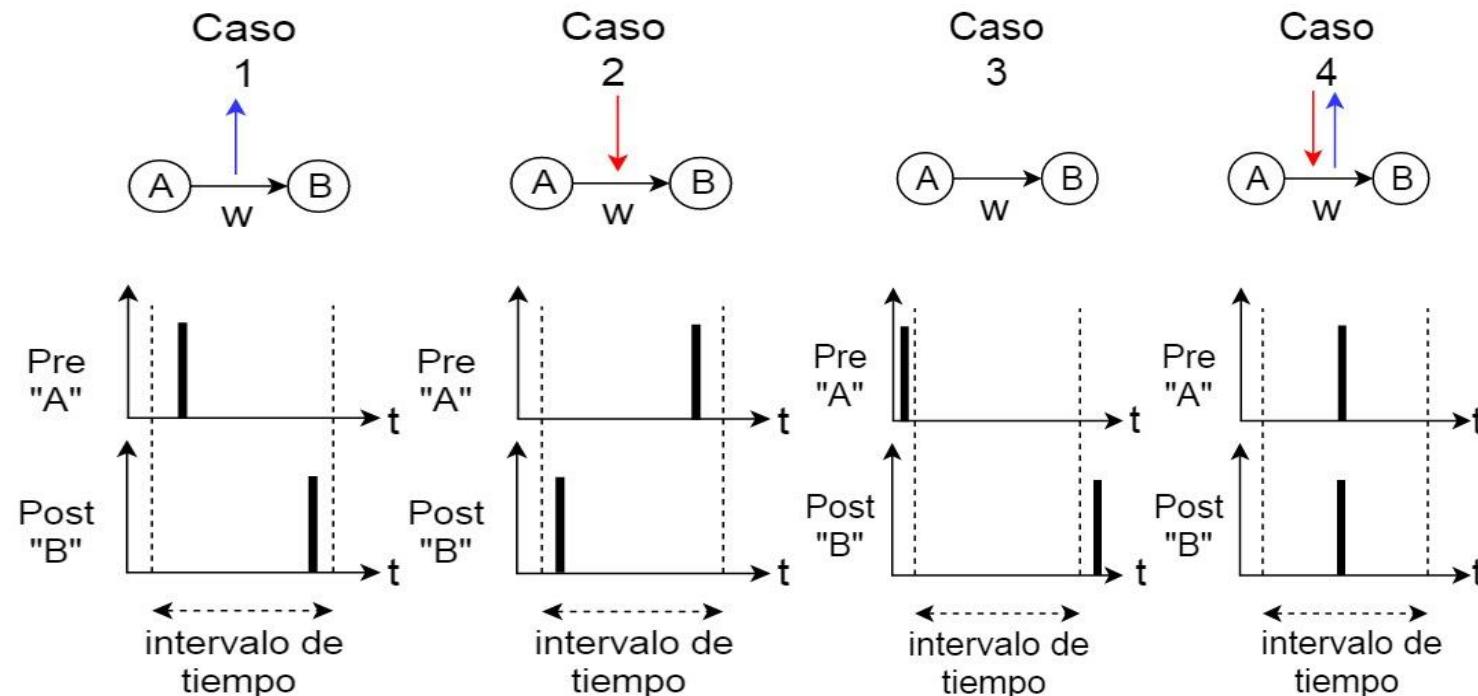
$$A_- \tau_- > A_+ \tau_+$$



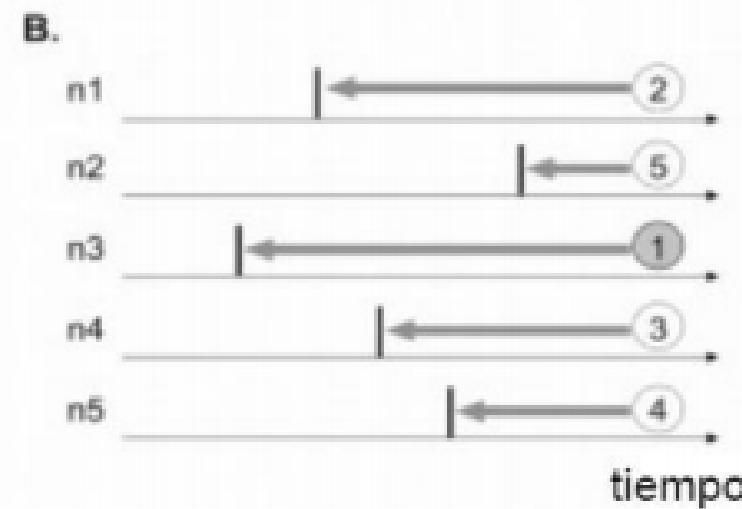
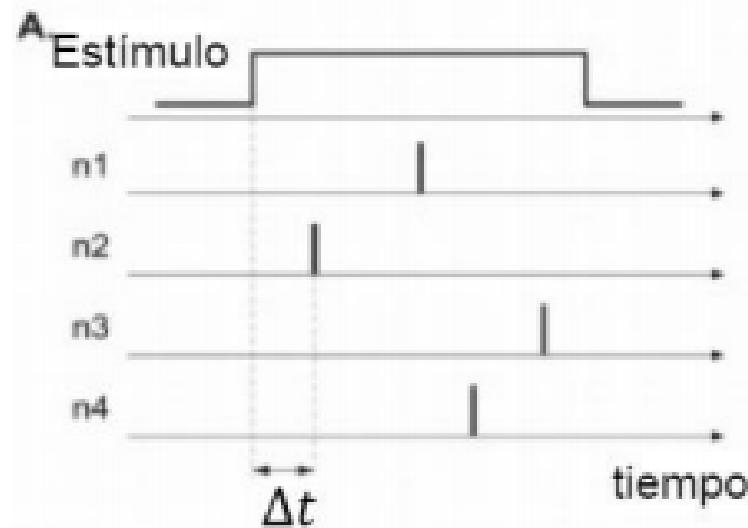
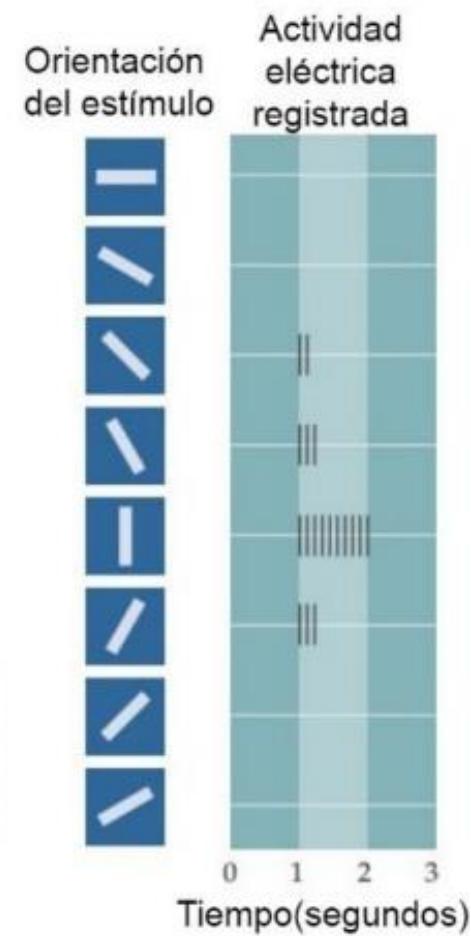


Regla de asociación de memoria.

$$w_{new} = \begin{cases} w_{old} + \eta \Delta w (w_{max} - w_{old}) & \text{si } \Delta w \geq 0 \\ w_{old} + \eta \Delta w (w_{old} - w_{min}) & \text{si } \Delta w < 0 \end{cases}$$



Codificación

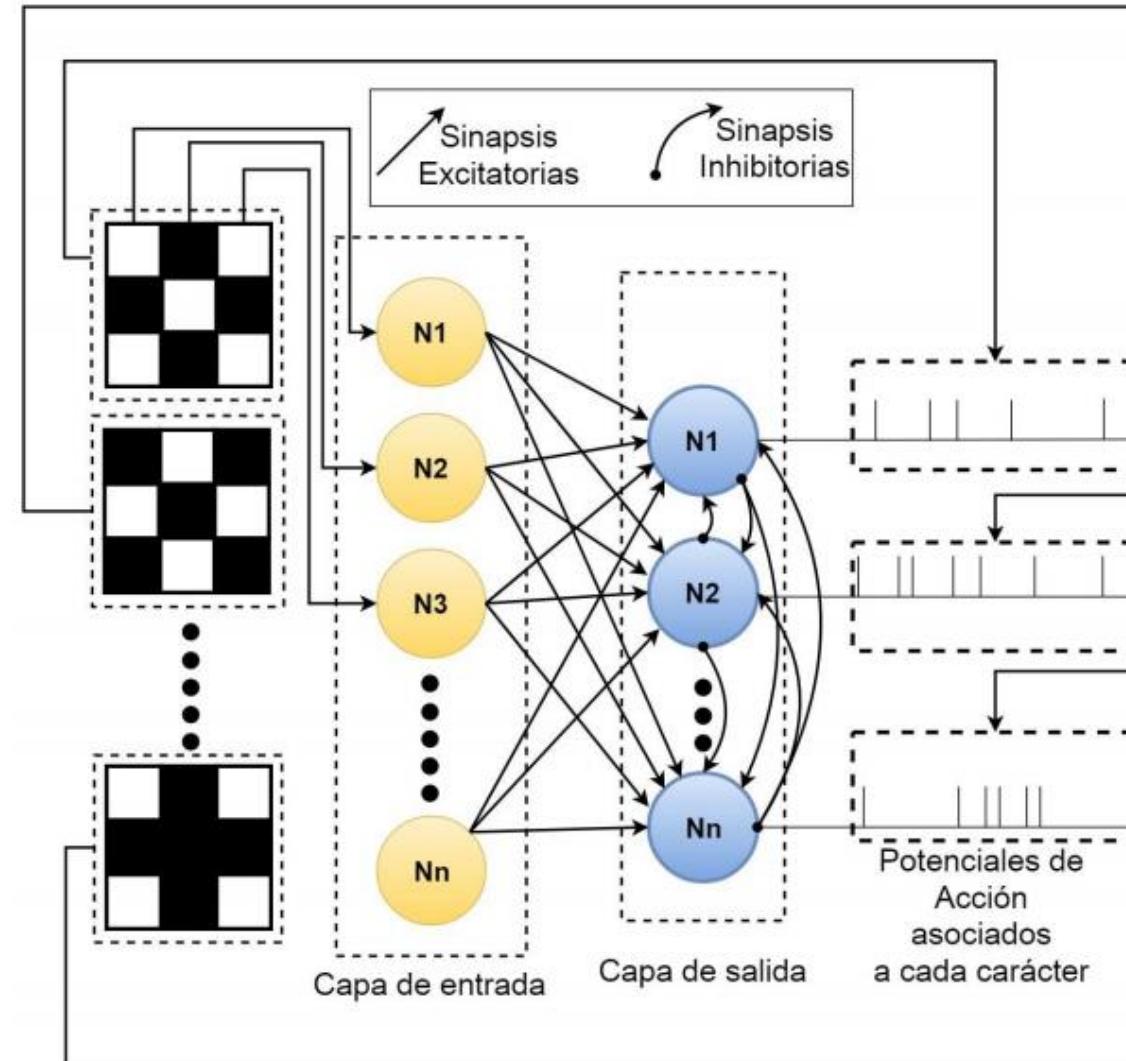




Arquitectura de Red Neuronal

Neurona por pixel
en la capa de entrada

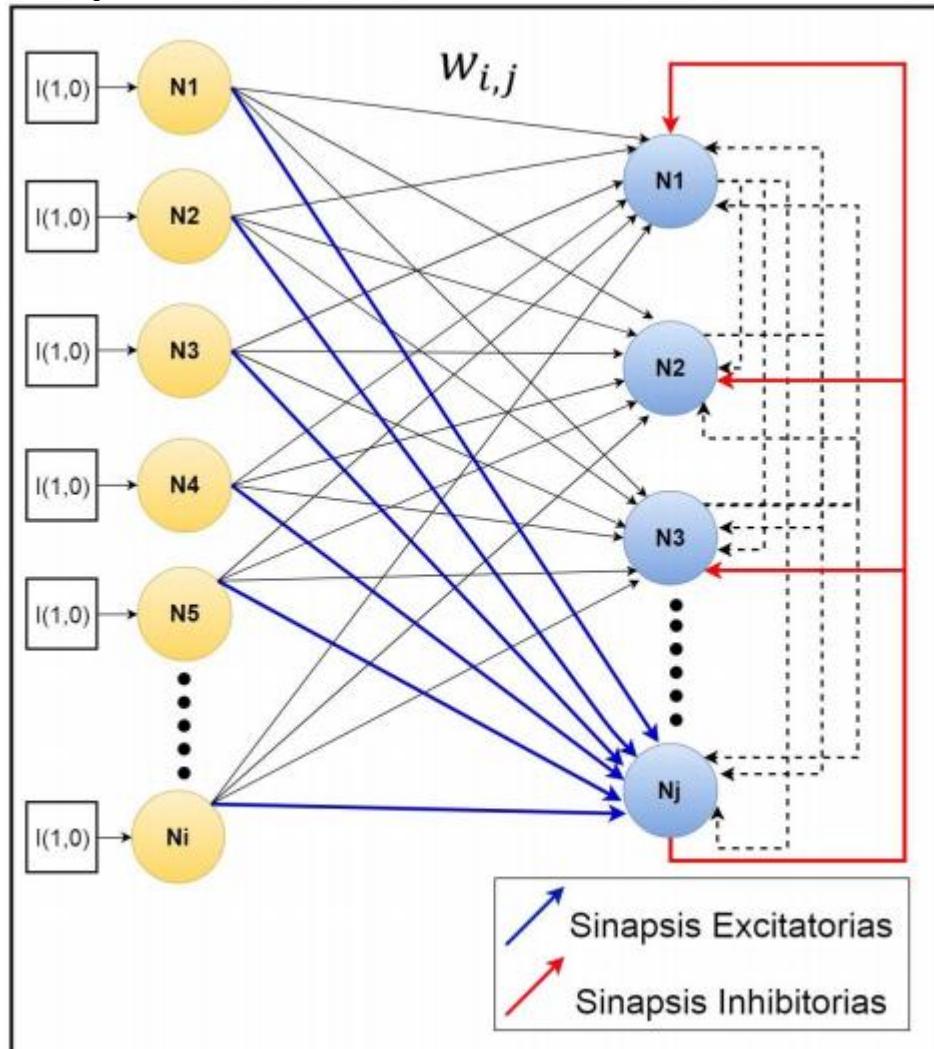
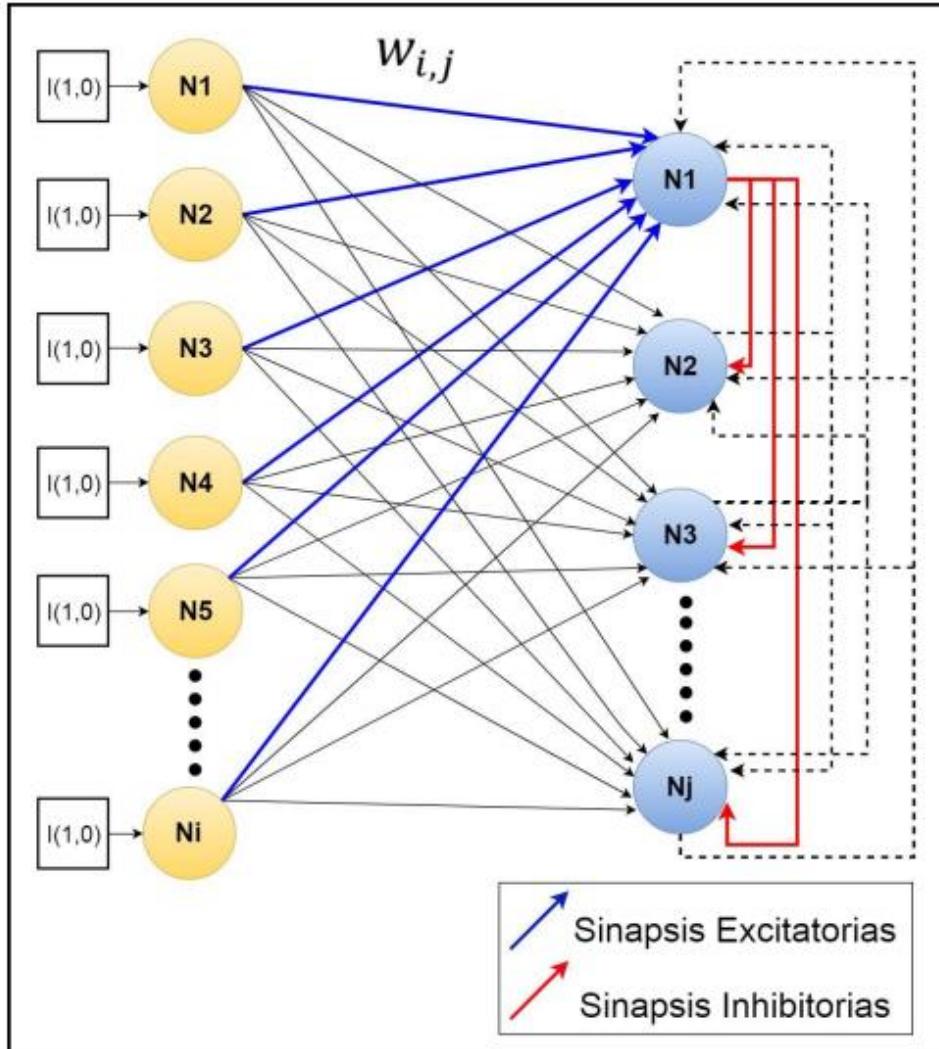
Pixel blanco = baja (0 pA)
Pixel Negro = alta (5 pA)



Neurona por imagen a
reconocer



Algoritmo de aprendizaje basado en el primero en disparar

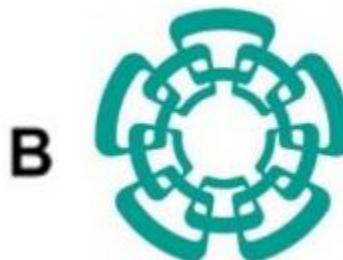


Pudiendo ser:
• Auto-organizado
• Selectivo

Reconocimiento en paralelo



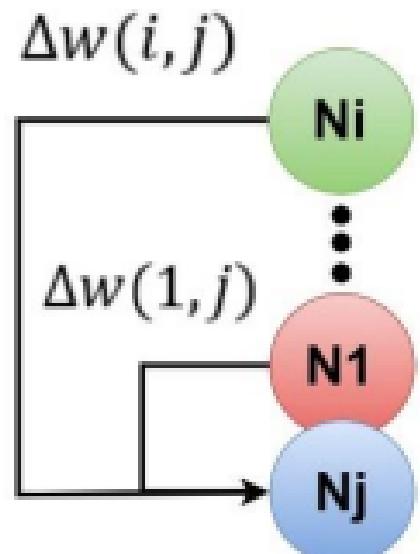
Ejecución del entrenamiento en software



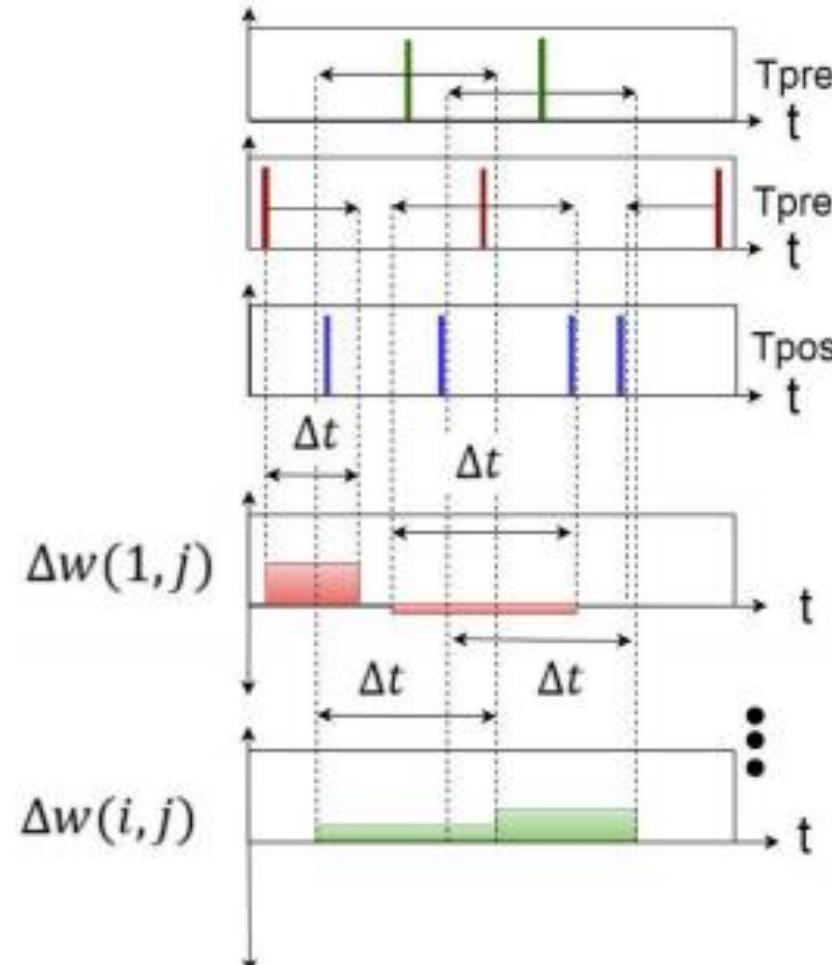
B

Entrenamiento

- Para una sola neurona post sináptica



STDP ($T_{pos} - T_{pre}$) Regla de modificación Sináptica

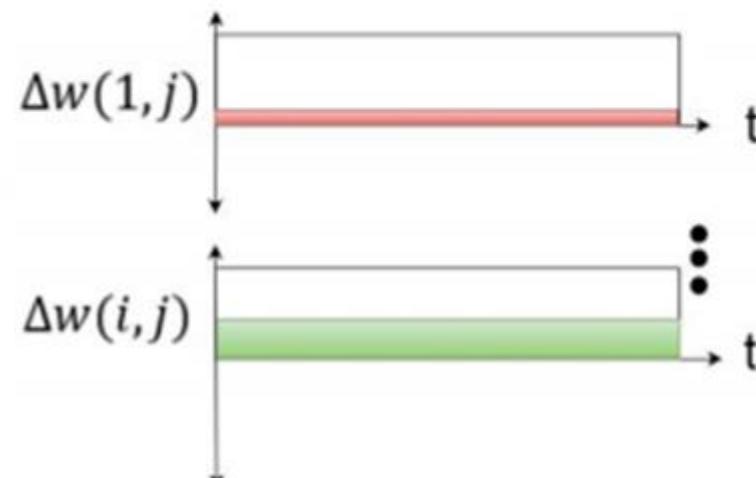


A

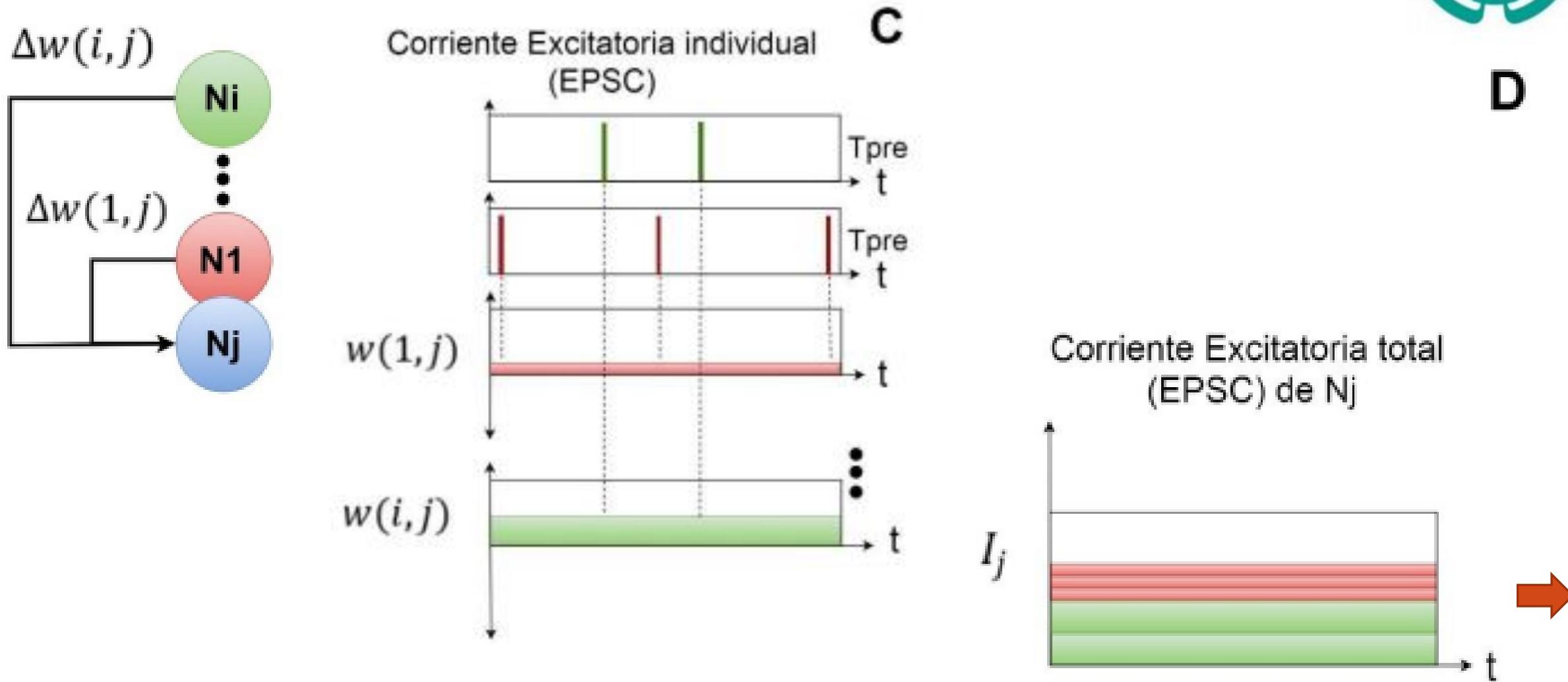
Ventana de tiempo

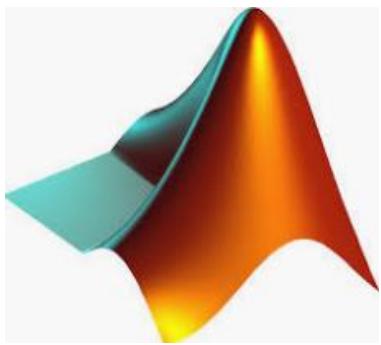


Integración de STDP

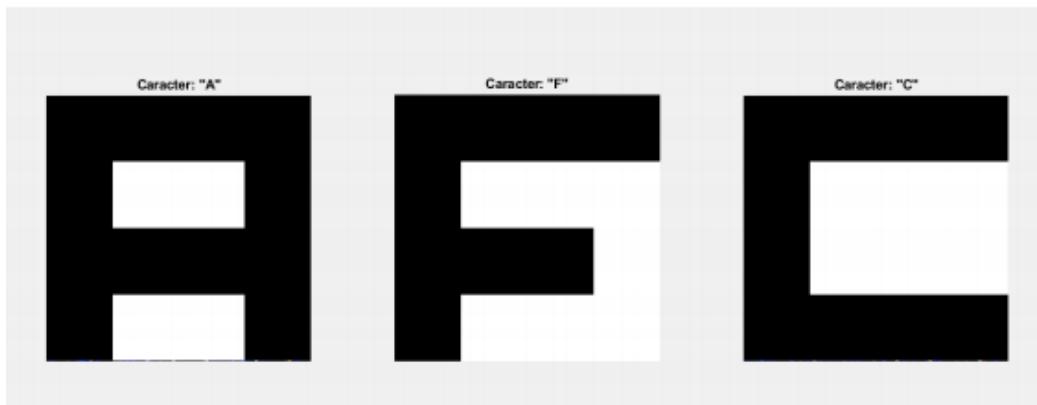


Ejecución del Algoritmo en software





Resultados Software

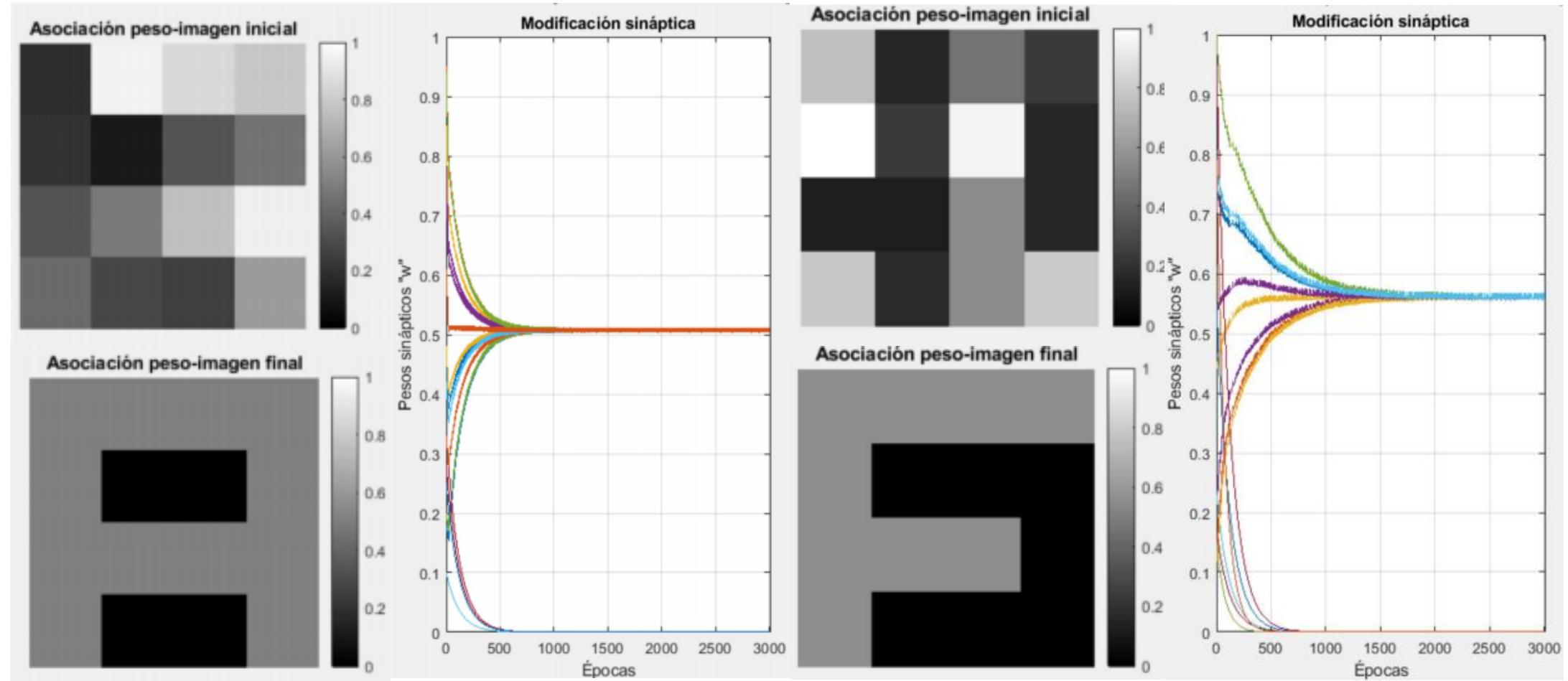


- Auto-organizado (método burbuja)
- imágenes 4x4 (capa de entrada)
- 3 imágenes a clasificar(capa de salida)
- 19 neuronas
- Ventana de tiempo= 250[ms]
- Epocas = 3000
- Velocidad de aprendizaje = 0.1
- Reconocimiento 100%



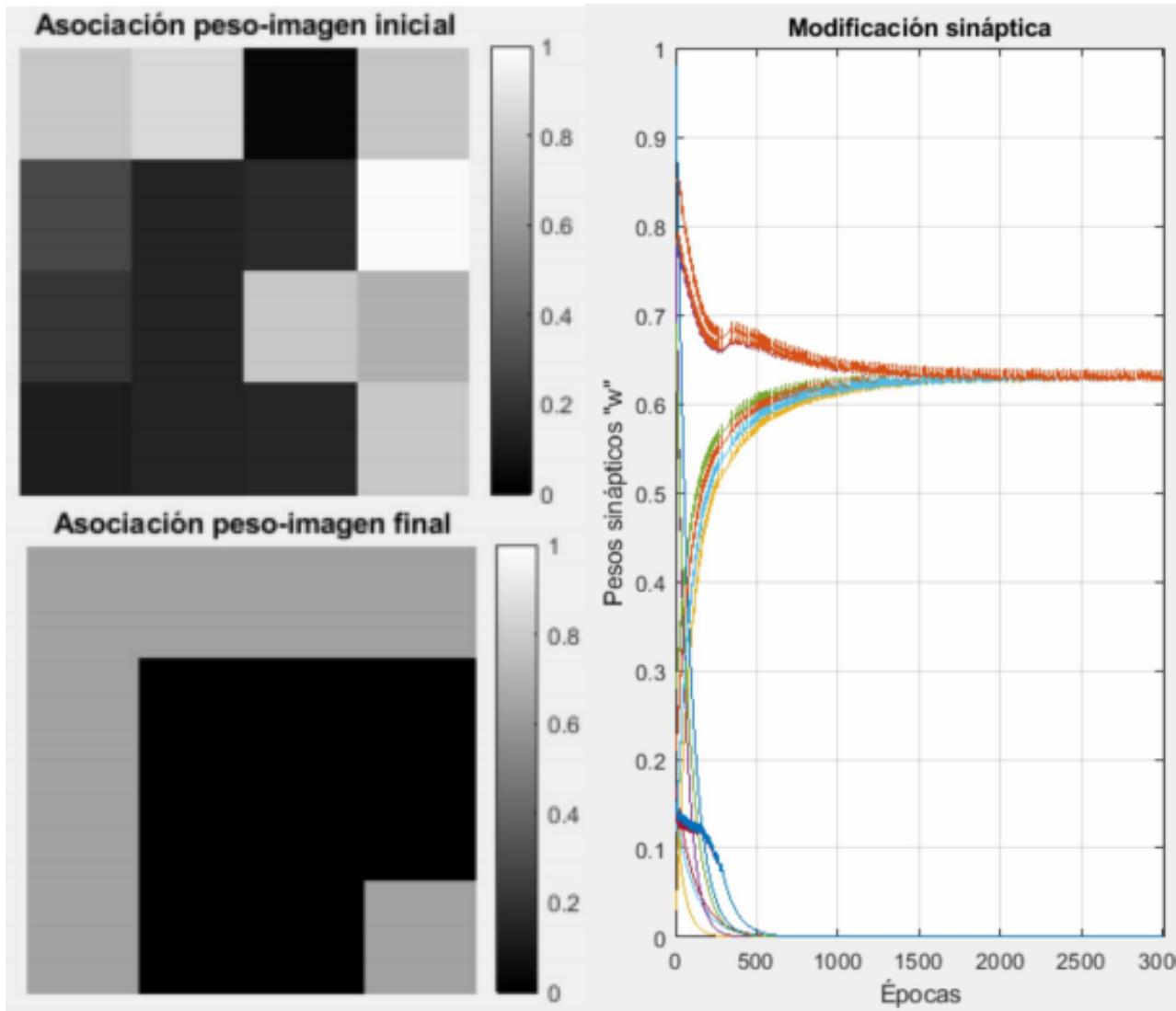


Aprendizaje Auto-organizado





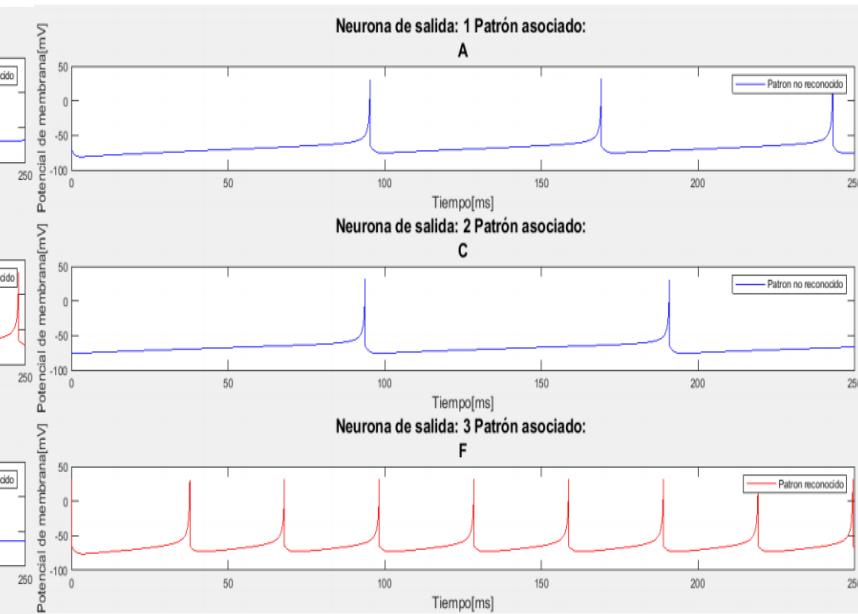
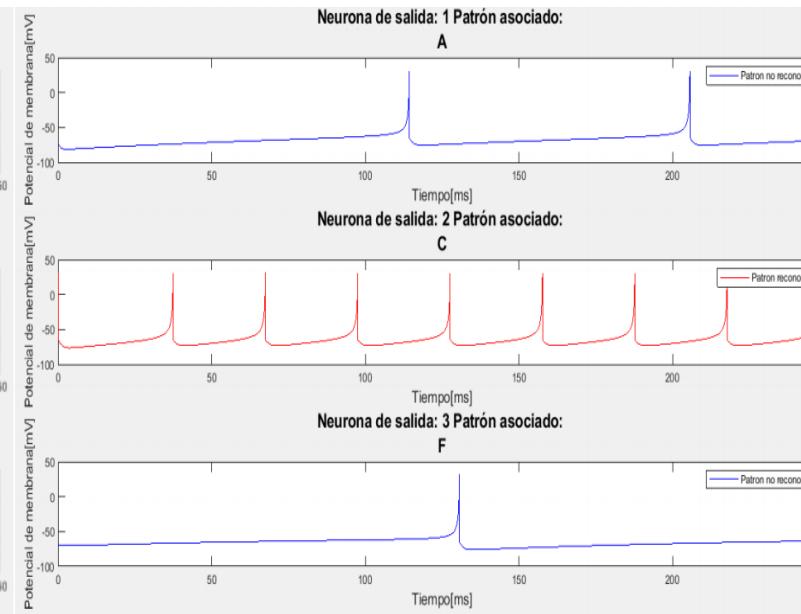
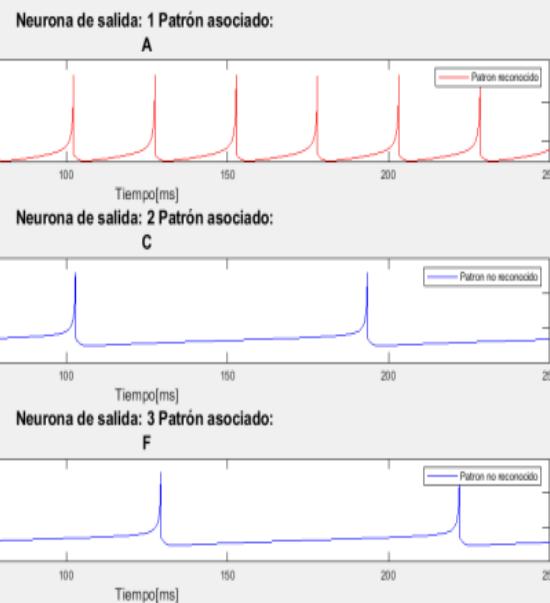
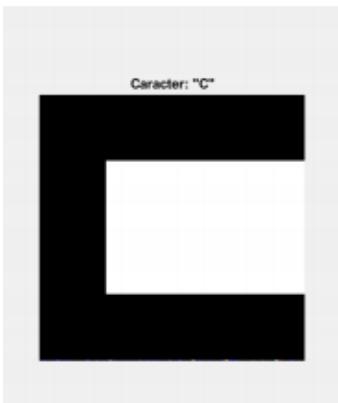
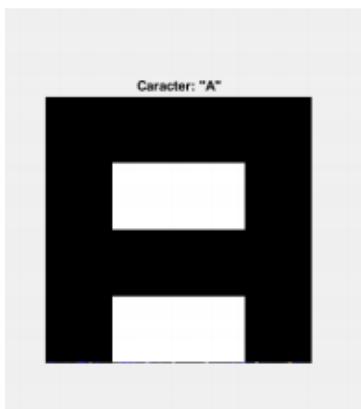
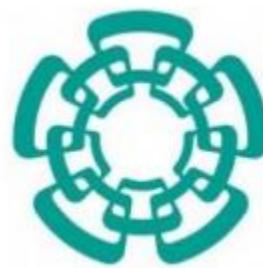
Aprendizaje Auto-organizado



Neurona Post sináptica	Caracter aprendido	EPSC [pA]
1	A	18.13
2	C	15.14
3	F	15.05



Reconocimiento





Implementación en Hardware

Caracteres de 36 píxeles (6x6) y reconocer 10 de ellos.

Arquitectura Neuromórfica →

Modelo Neuronal (simplificación de constantes) →

AER →

Modelo Sináptico →

Mecanismo de inducción de aprendizaje (STDP)

- Ventana de aprendizaje
- Protocolo de asociación de pulsos

Regla de asociación de memoria →

Representación Neuronal



Representación de procesamiento



Implementación Algoritmo como sistema Neuromórfica

Hardware digital →

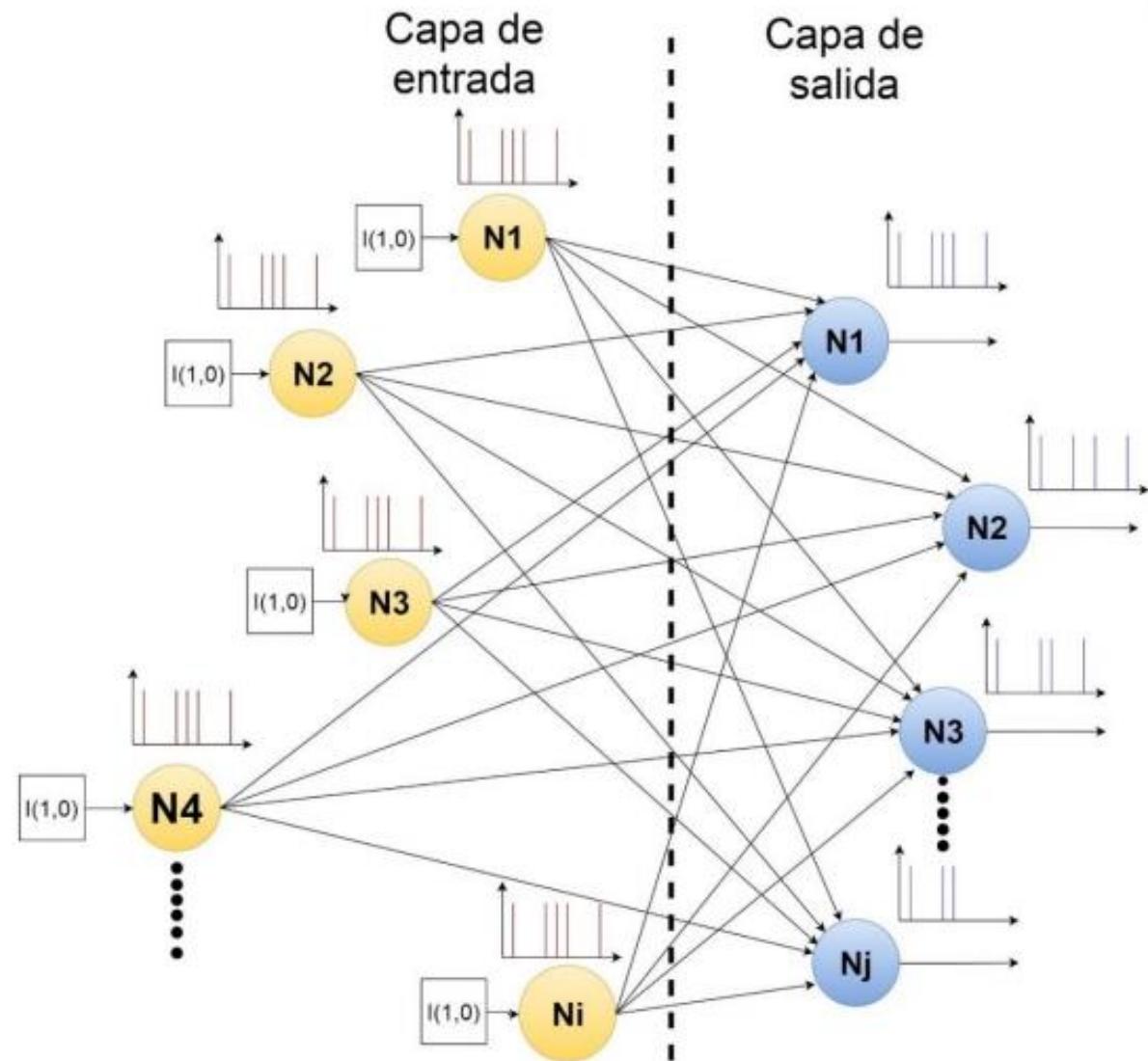
- Entry_Process
- Main_control
- STDP_Process

Resultados →

Arquitectura Neuromórfica

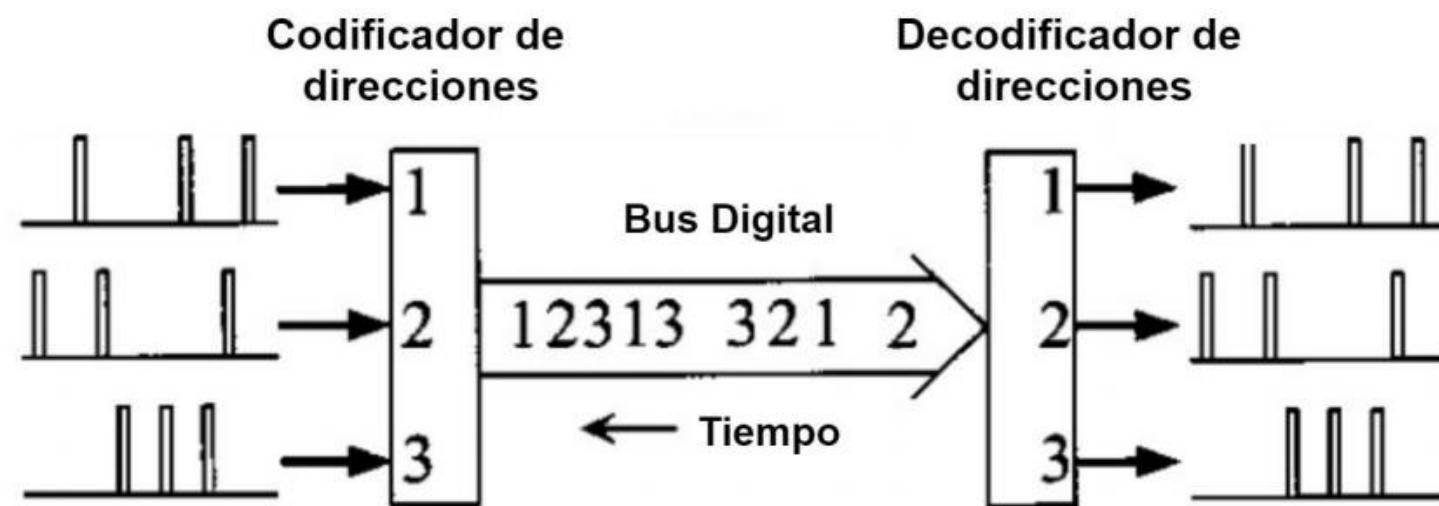
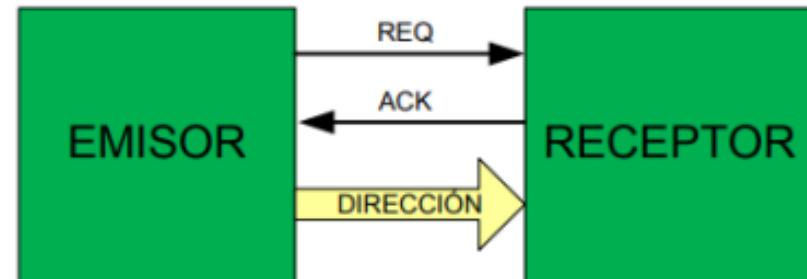


- Emular sistemas neuronales biológicos
 - Paralelismo entre procesos
 - Procesamiento en tiempo Real
 - Tolerancia a fallos
- Hardware (circuitos Neuromórficos)
 - Address Event Representation →
 - Sistema de retardos →



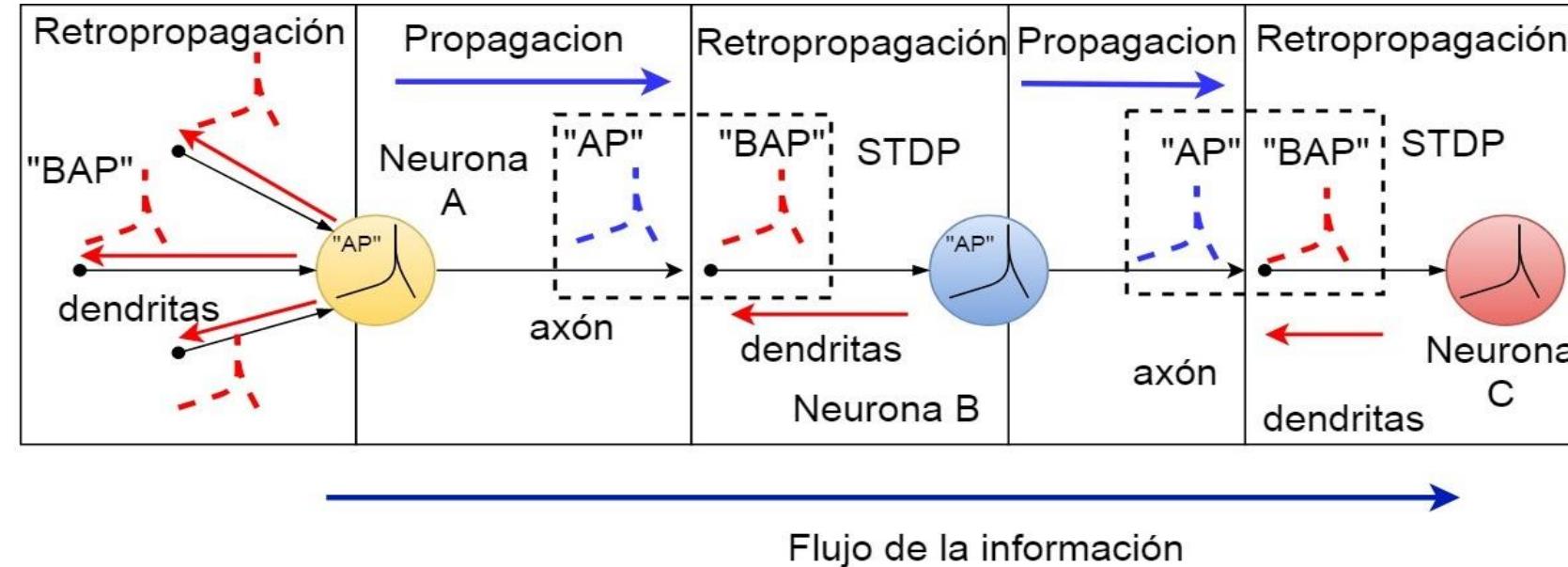


Address Event Representation (AER)





Sistema de propagación y retro-propagación

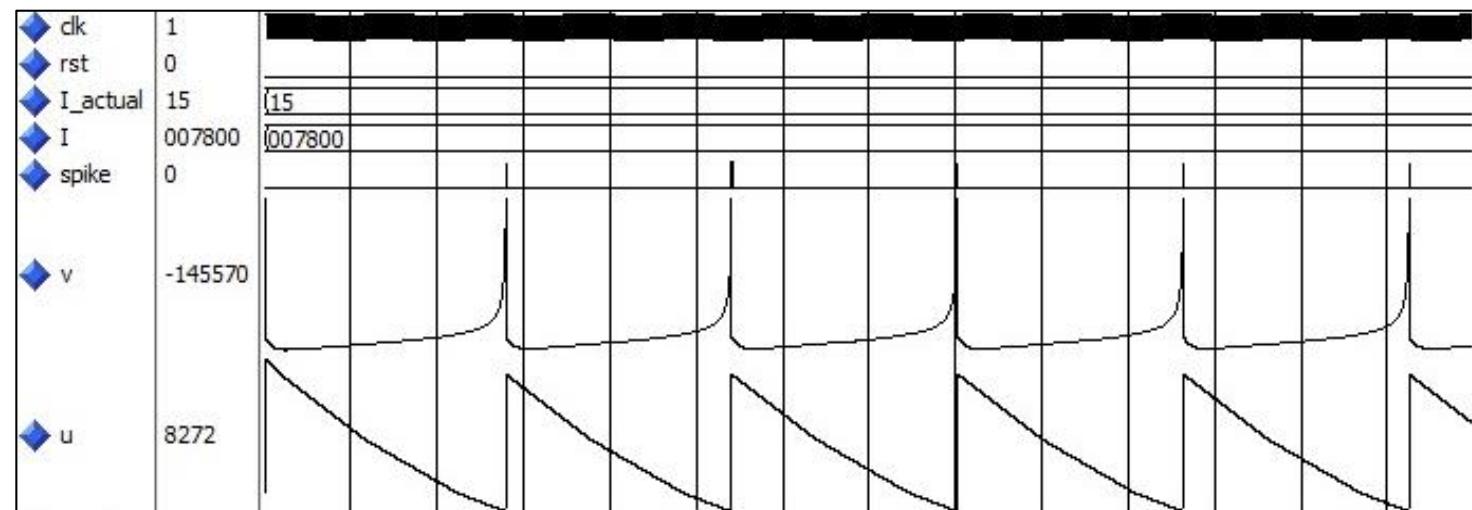
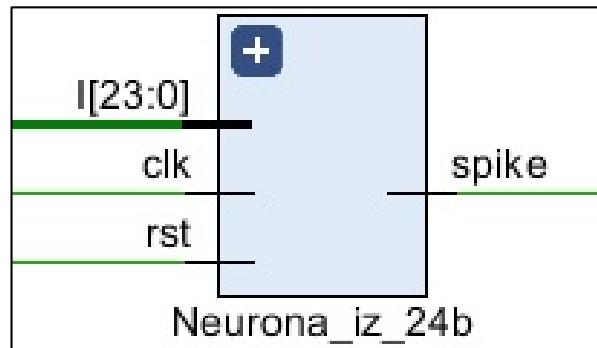


- Conversor de pixel en corriente con retardo (sinapsis con retardo)
- Flip - Flop D (Retardo retro-propagado)
- FIFO

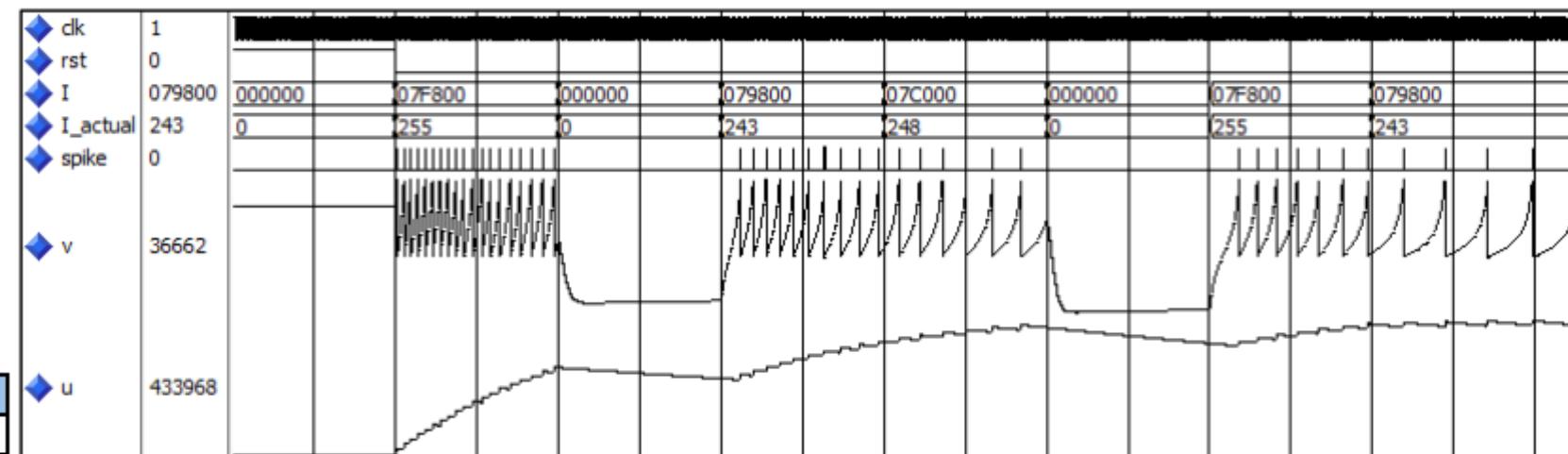




Neurona Izhikevich-VHDL

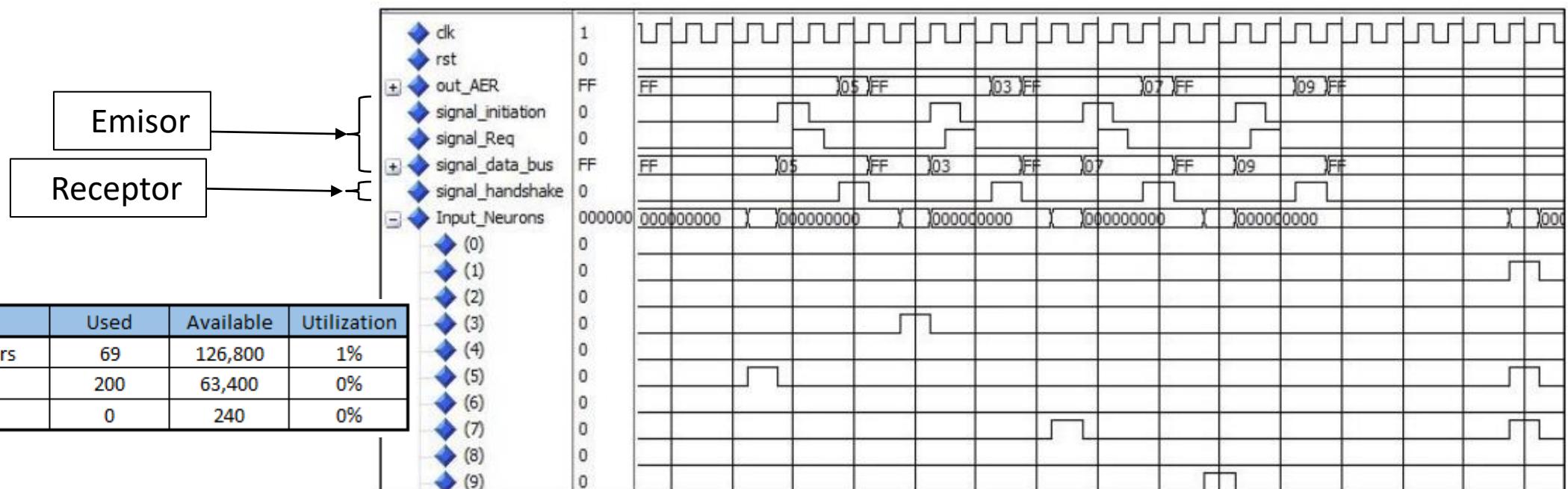
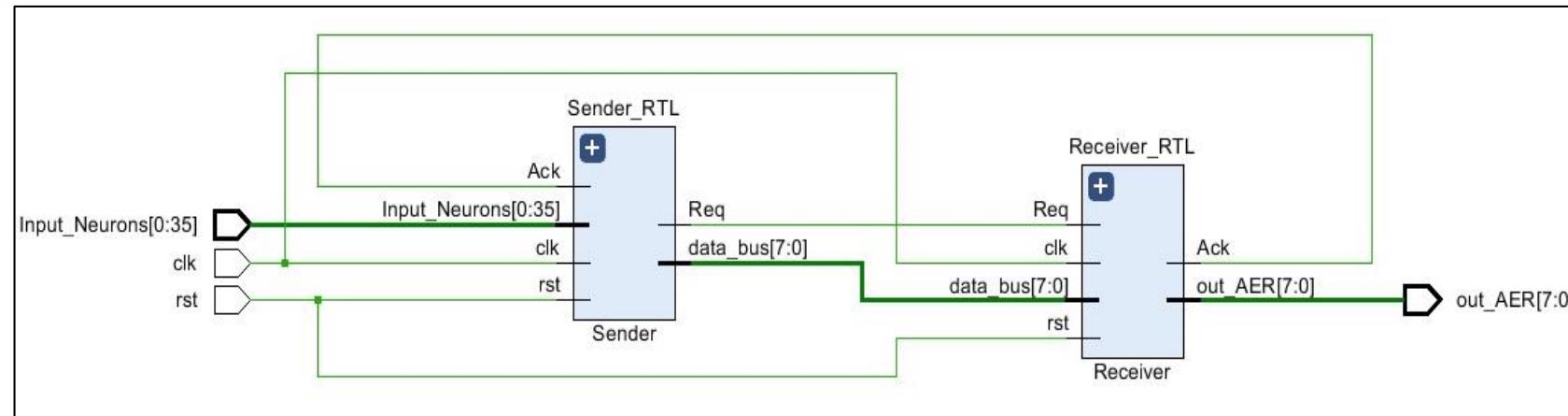


Parámetros	Valor Decimal	Valor Hexadecimal
a	0.015625	000020
b	0.125	000100
c	-65	FDF800
d	8	004000
dt	0.0078125 [ms]	000010



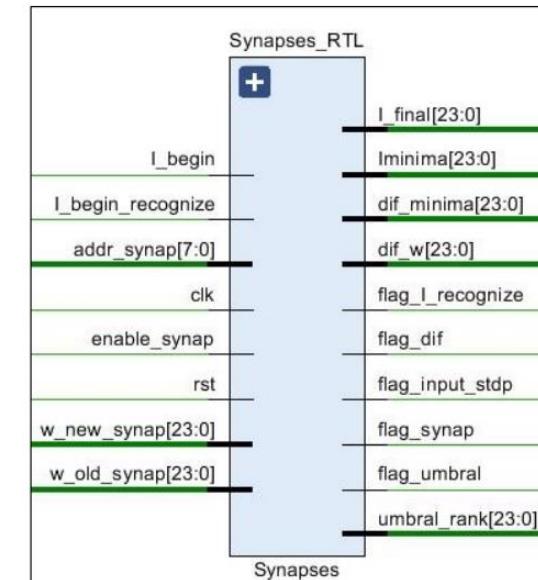
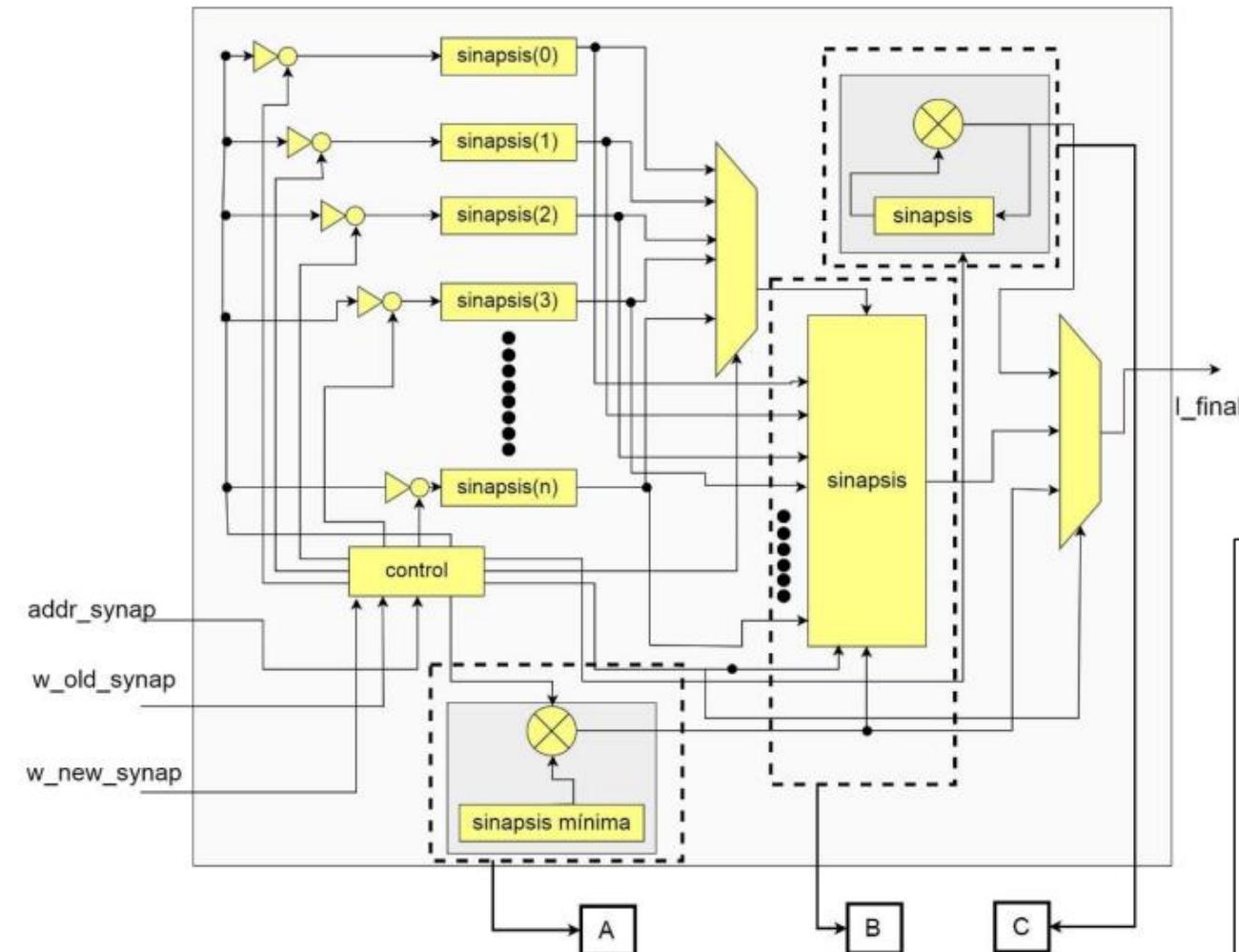
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	51	126,800	1%
Number of Slice LUTs	183	63,400	0%
Number of DSP48E1s	3	240	1%





Logic Utilization	Used	Available	Utilization
Number of Slice Registers	69	126,800	1%
Number of Slice LUTs	200	63,400	0%
Number of DSP48E1s	0	240	0%

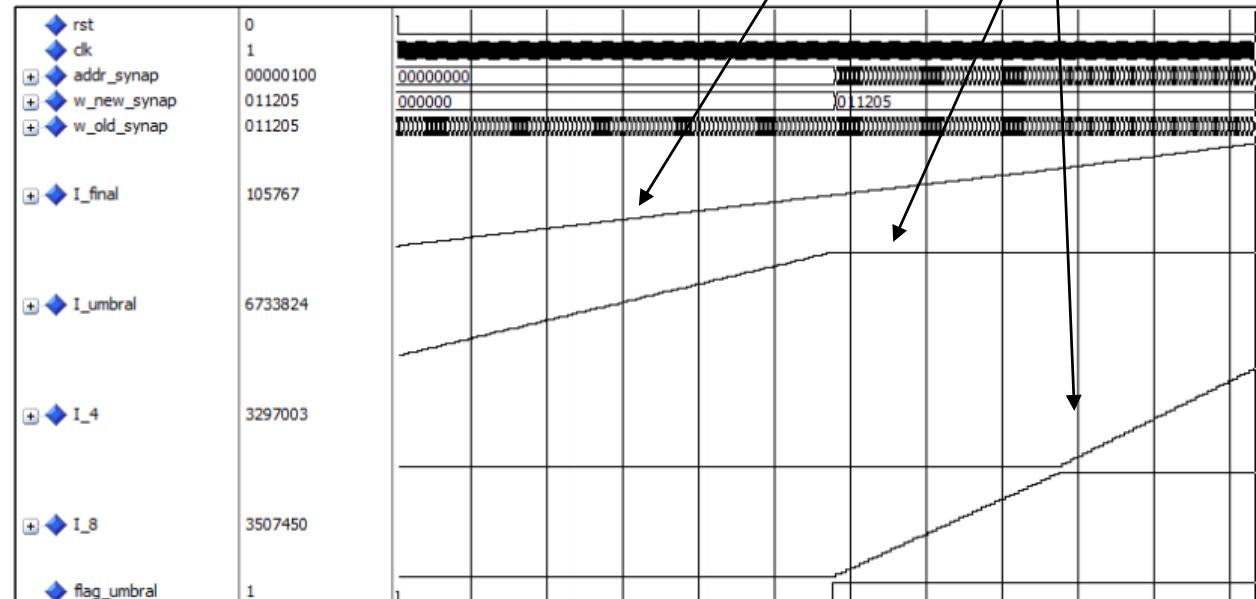
Sinapsis -VHDL



Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1,800	126,800	1%
Number of Slice LUTs	3,025	63,400	4%
Number of DSP48E1s	0	240	0%

Corriente Post sináptica total

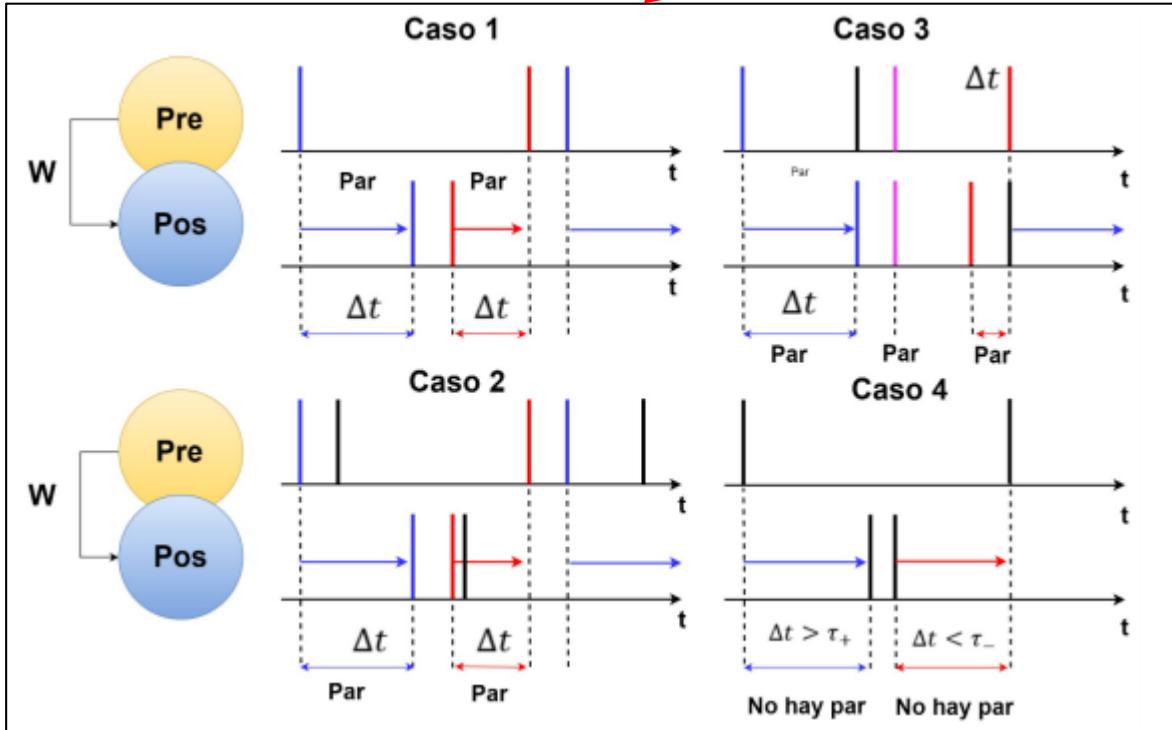
Suma paralela





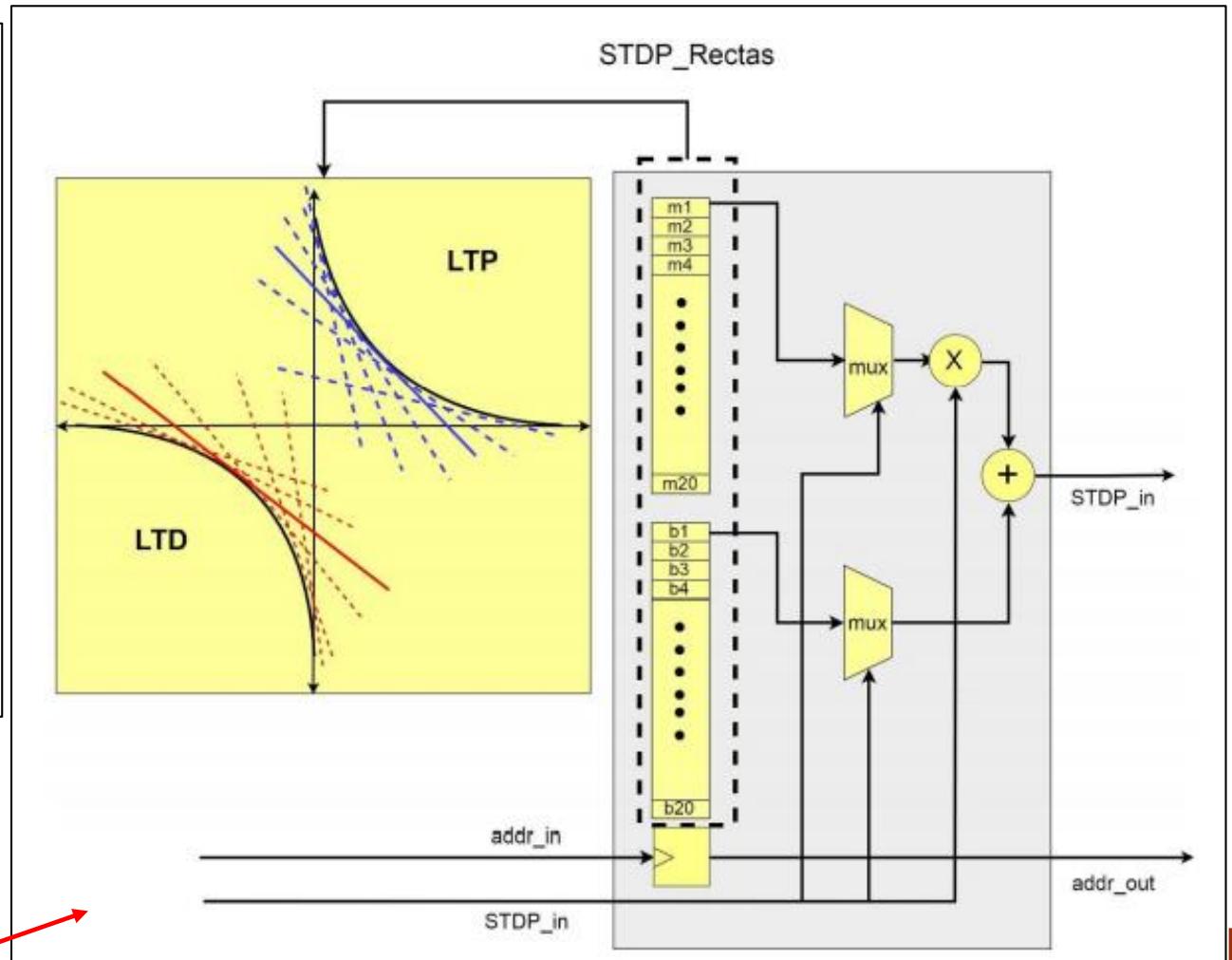
STDP Neuromórfico

Reglas de asociación propuestas

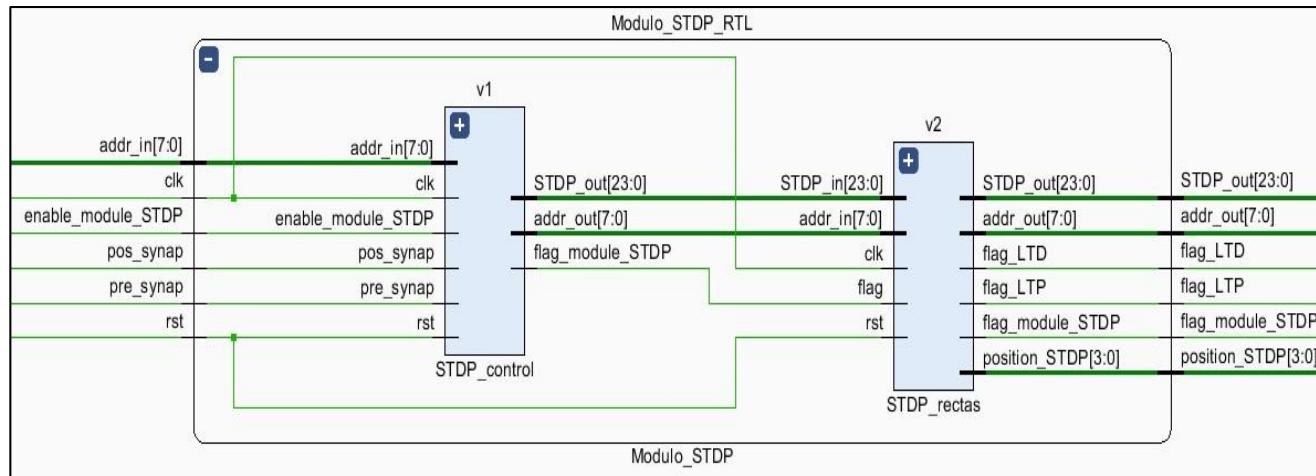


dt 0.0078125 [ms] 000010

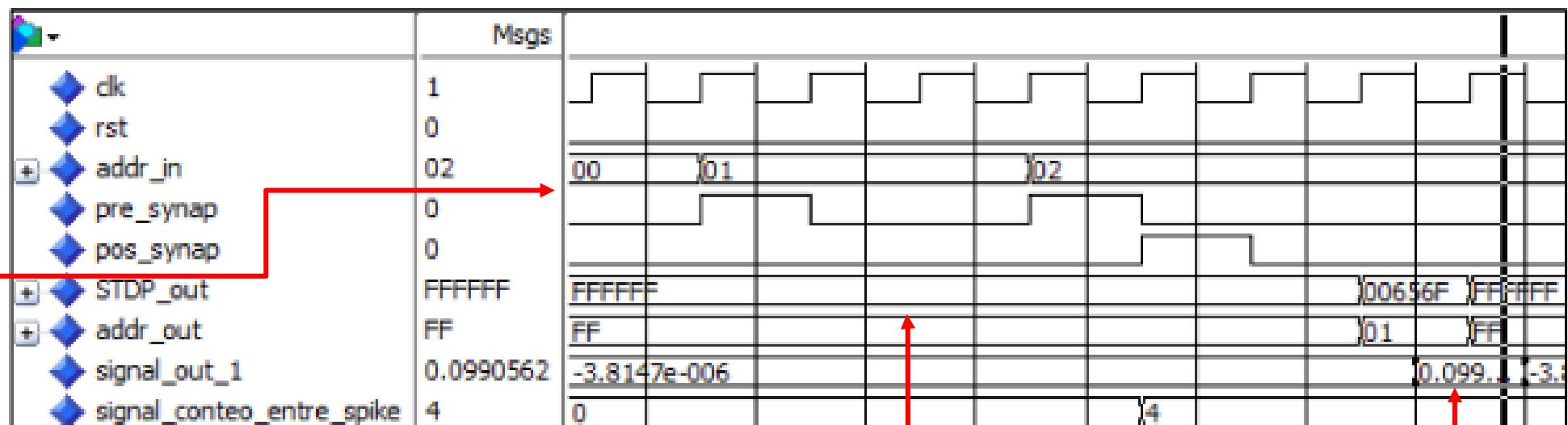
Ventana de aprendizaje



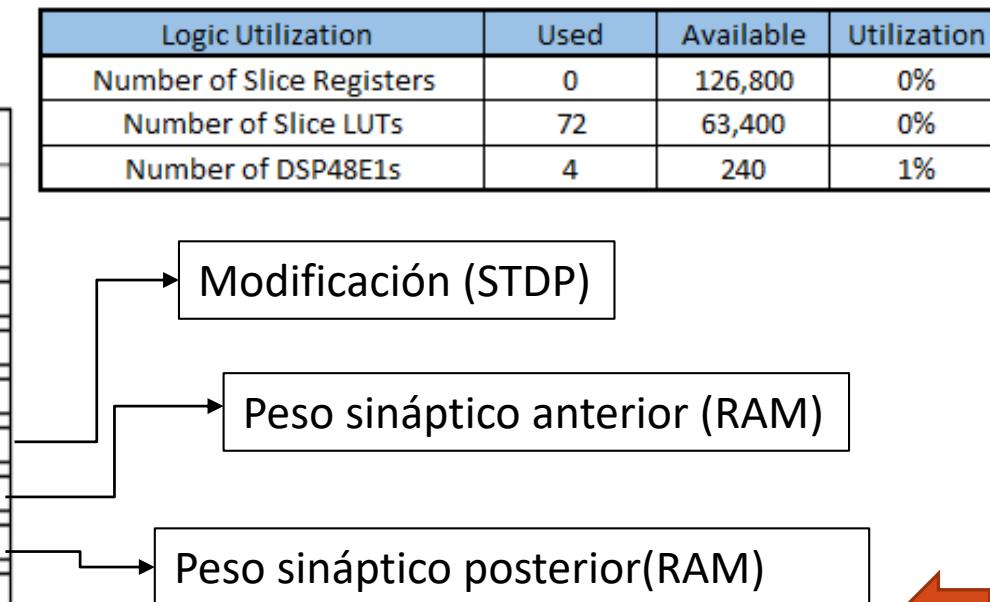
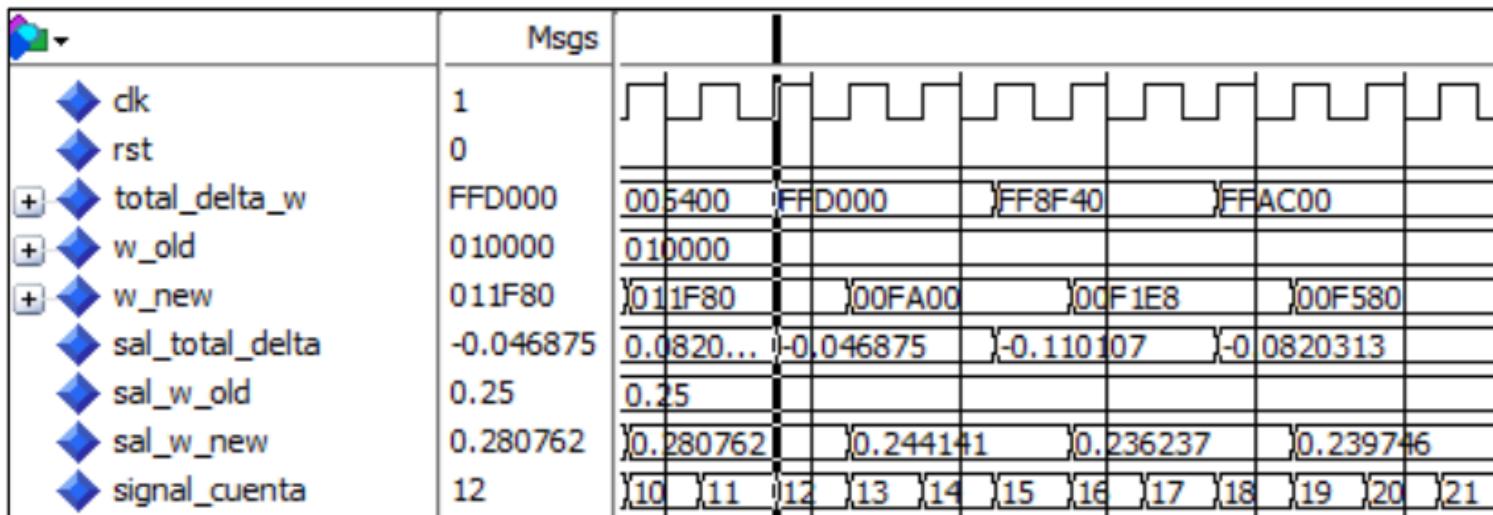
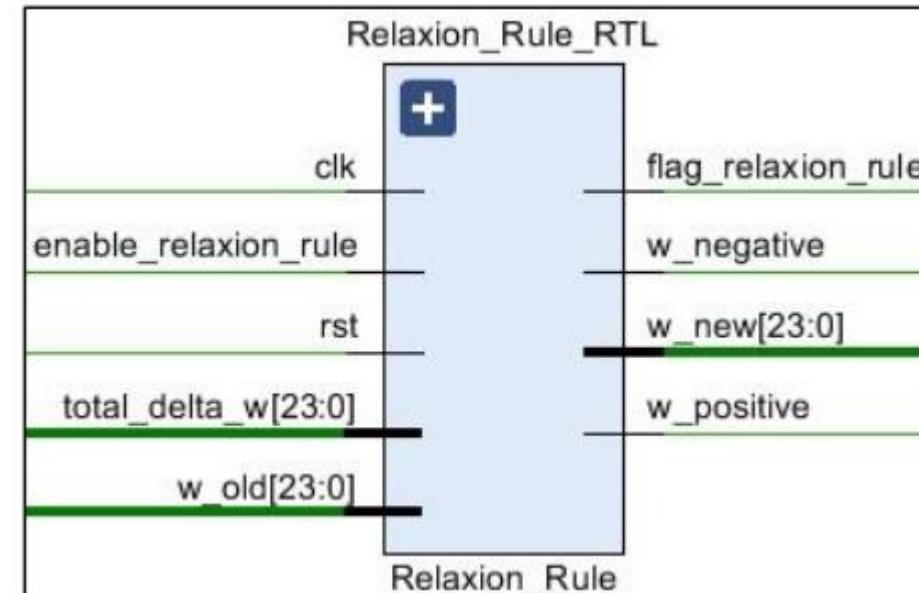
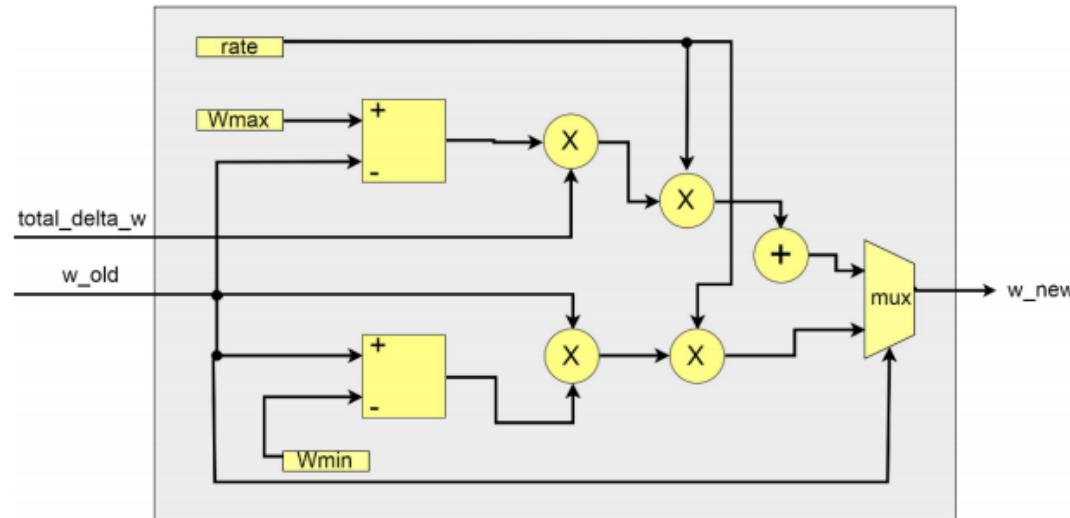
STDP -VHDL



Logic Utilization	Used	Available	Utilization
Number of Slice Registers	623	126,800	0%
Number of Slice LUTs	569	63,400	0%
Number of DSP48E1s	4	240	1%



Regla de relajación



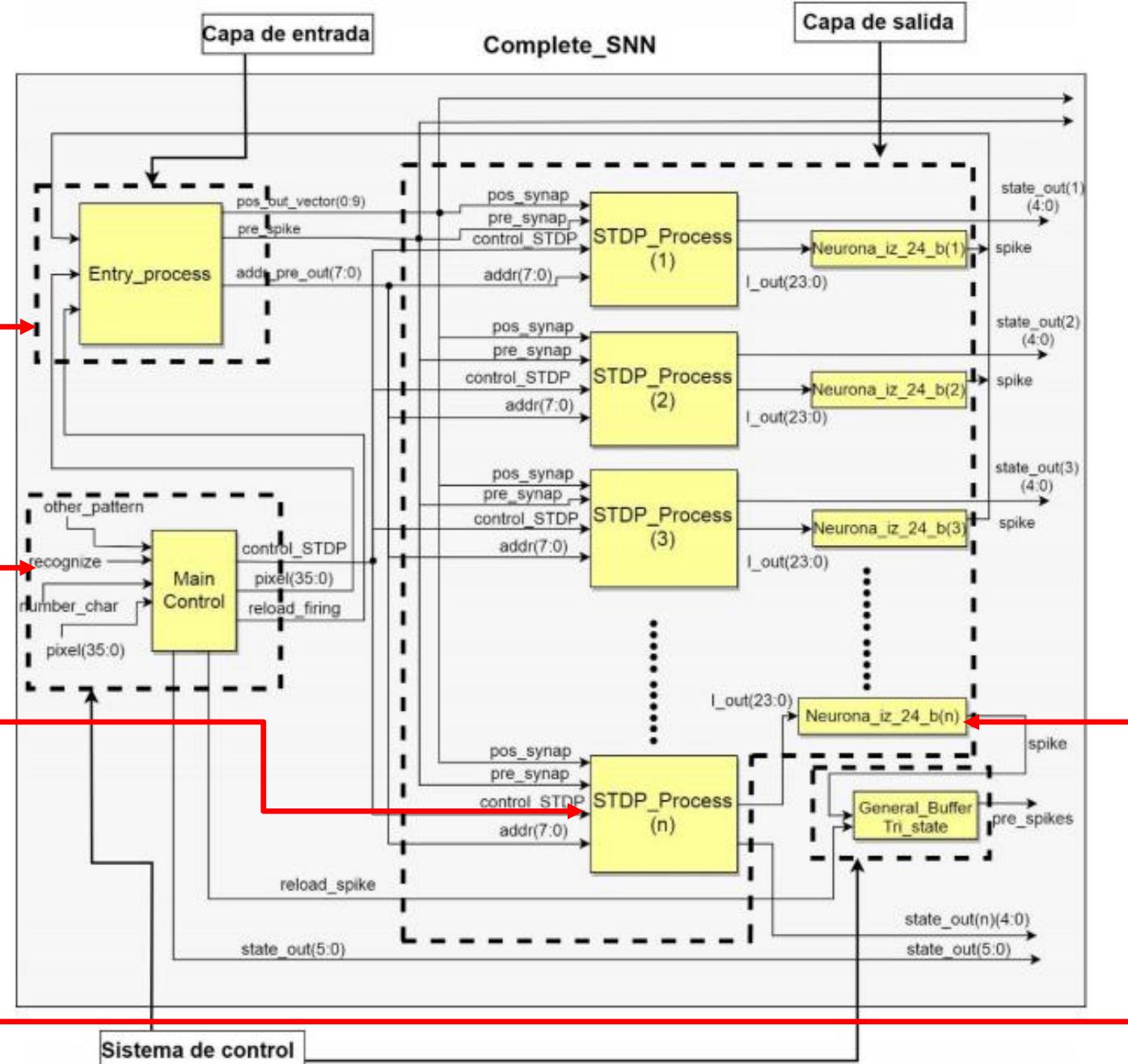


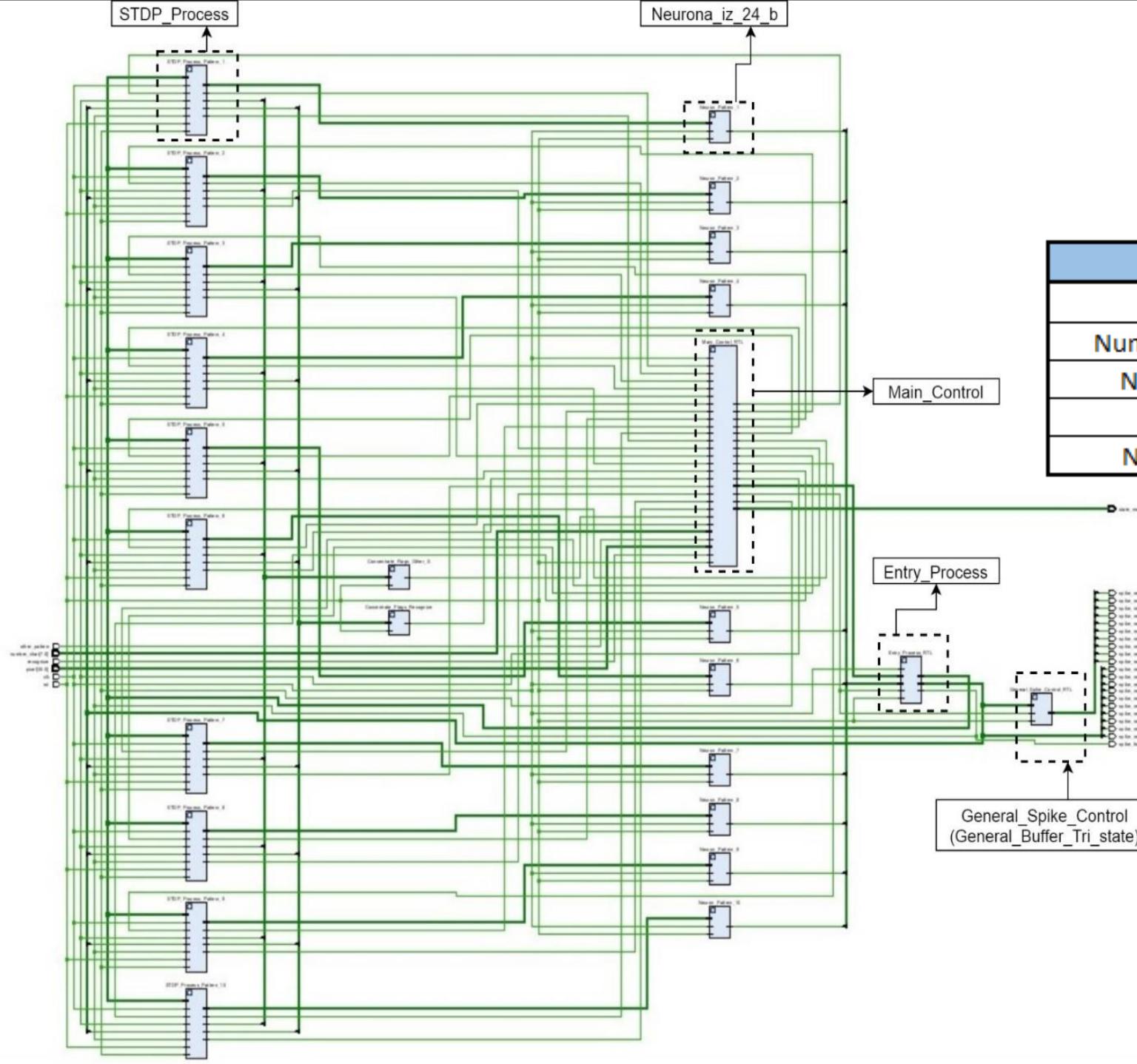
- 36 Neuronas de entrada
- Conversor de pixel con corriente (Retardo propagado)
- AER
- Flip Flop D(Retardo retro-propagado)
- FIFO(Sistema de compensación de tiempo)

- Algoritmo selectivo

- Sinapsis
- Regla de relajación
- Ventana de aprendizaje y protocolo de asociación.

- 10 Neuronas de salida





VIVADO[®]
HLx Editions



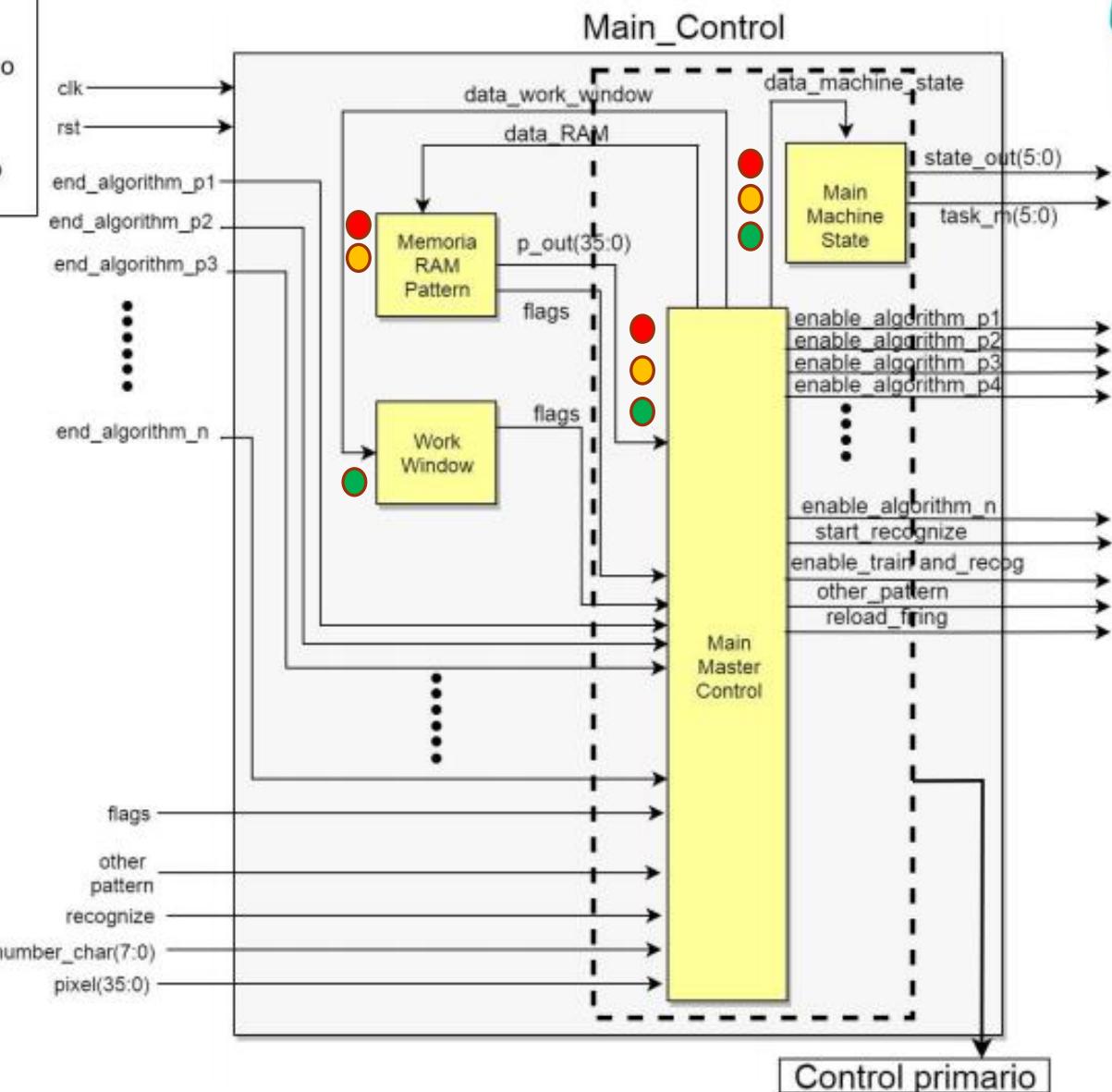
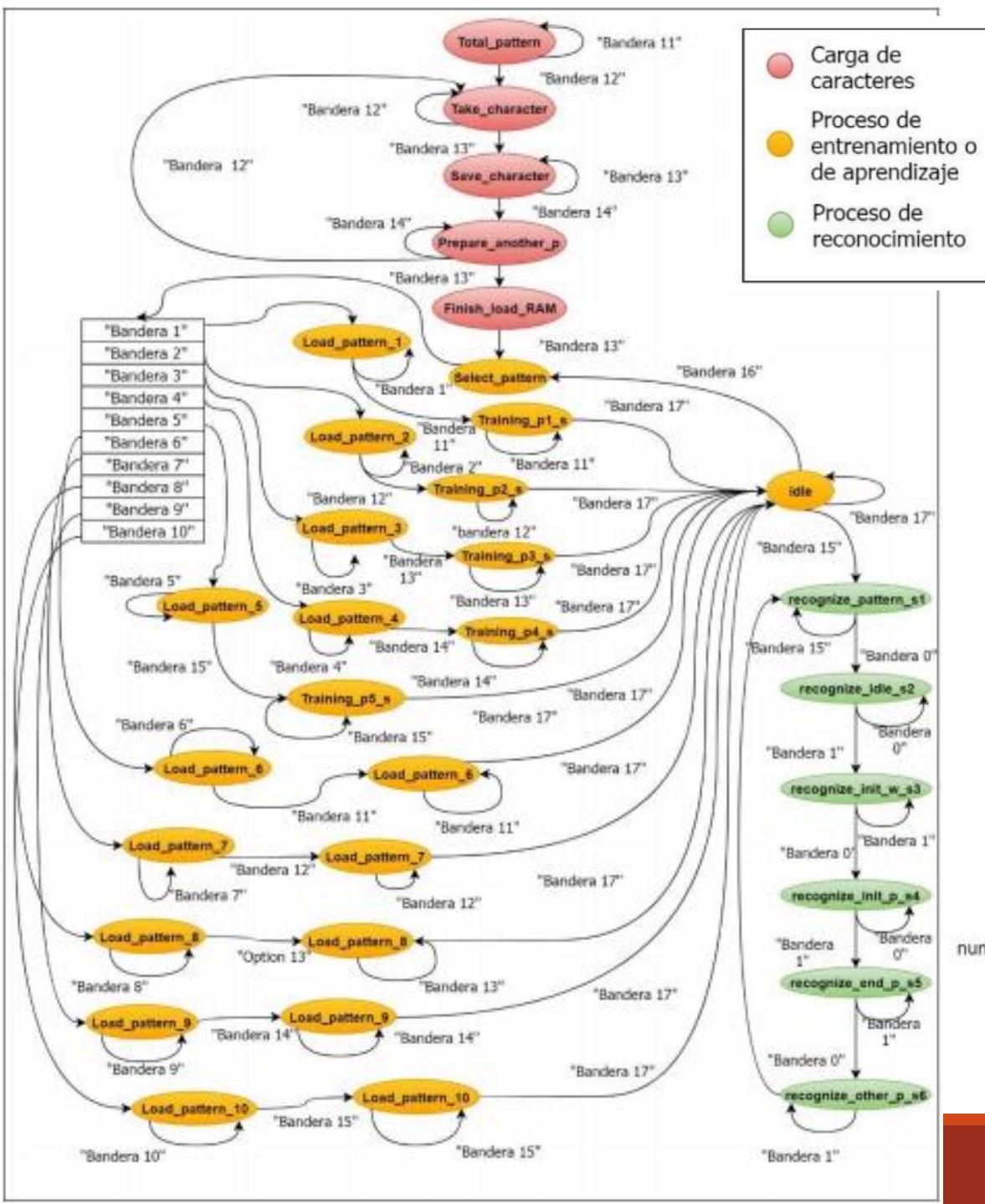
Logic Utilization	Used	Available	Utilization
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	29,593	126,800	23%
Number of Slice LUTs	59,534	63,400	93%
Number of IOBs	69	210	32%
Number of DSP48E1s	218	240	90%

FPGA Artix-7 XC7A100-T



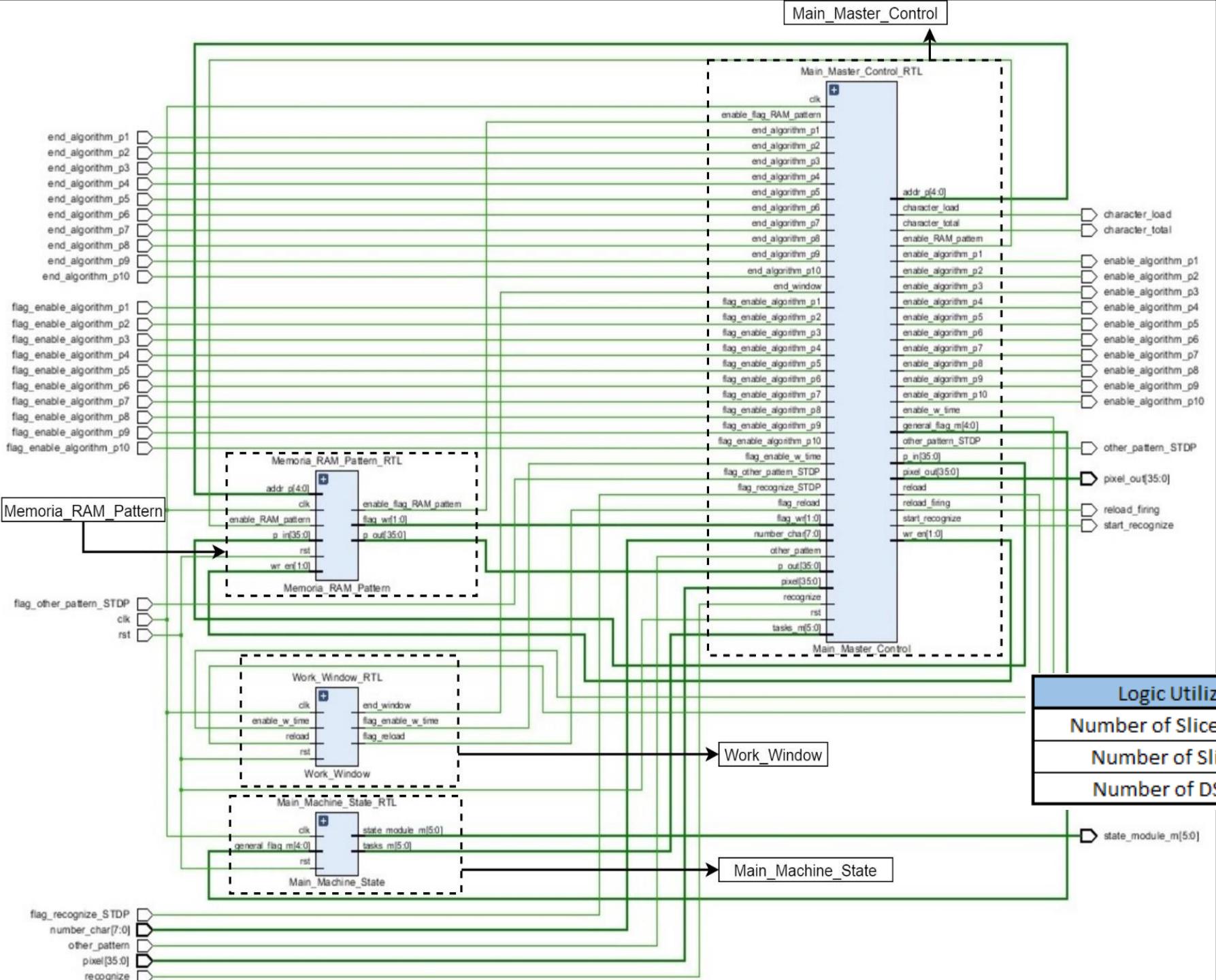
$f_{operación} = 10 \text{ MHz}$



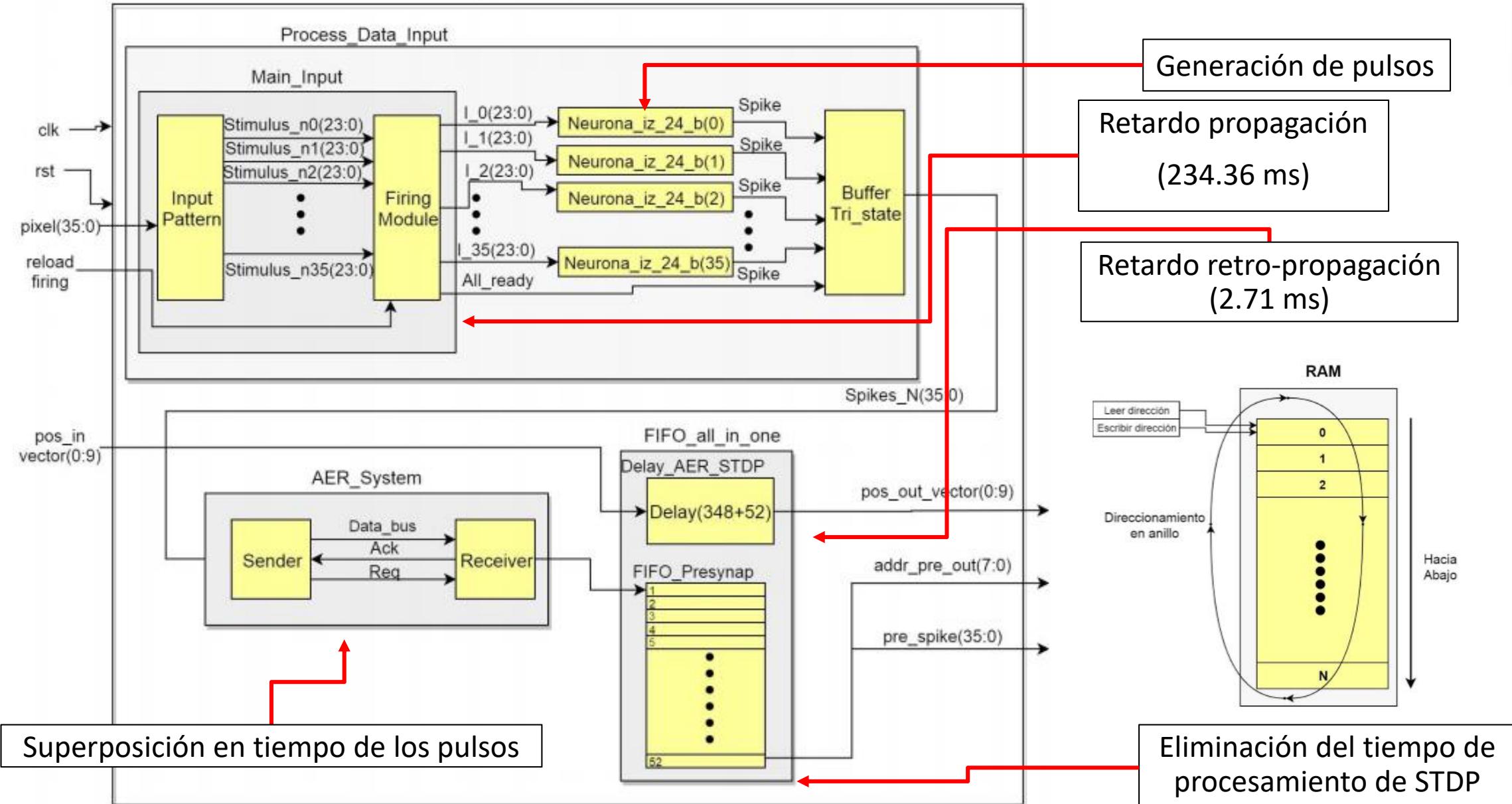




VIVADO®
HLx Editions

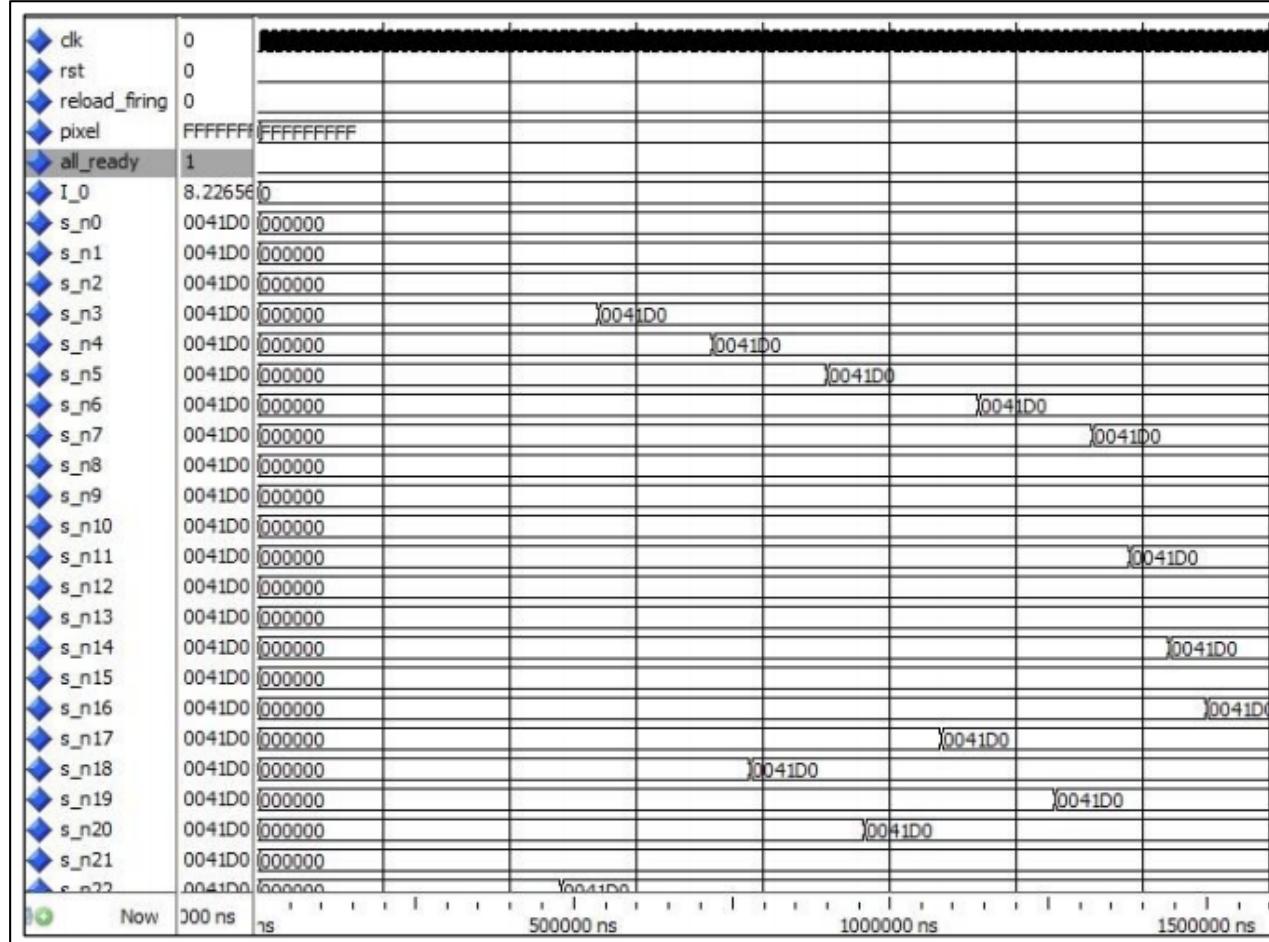


Entry_process

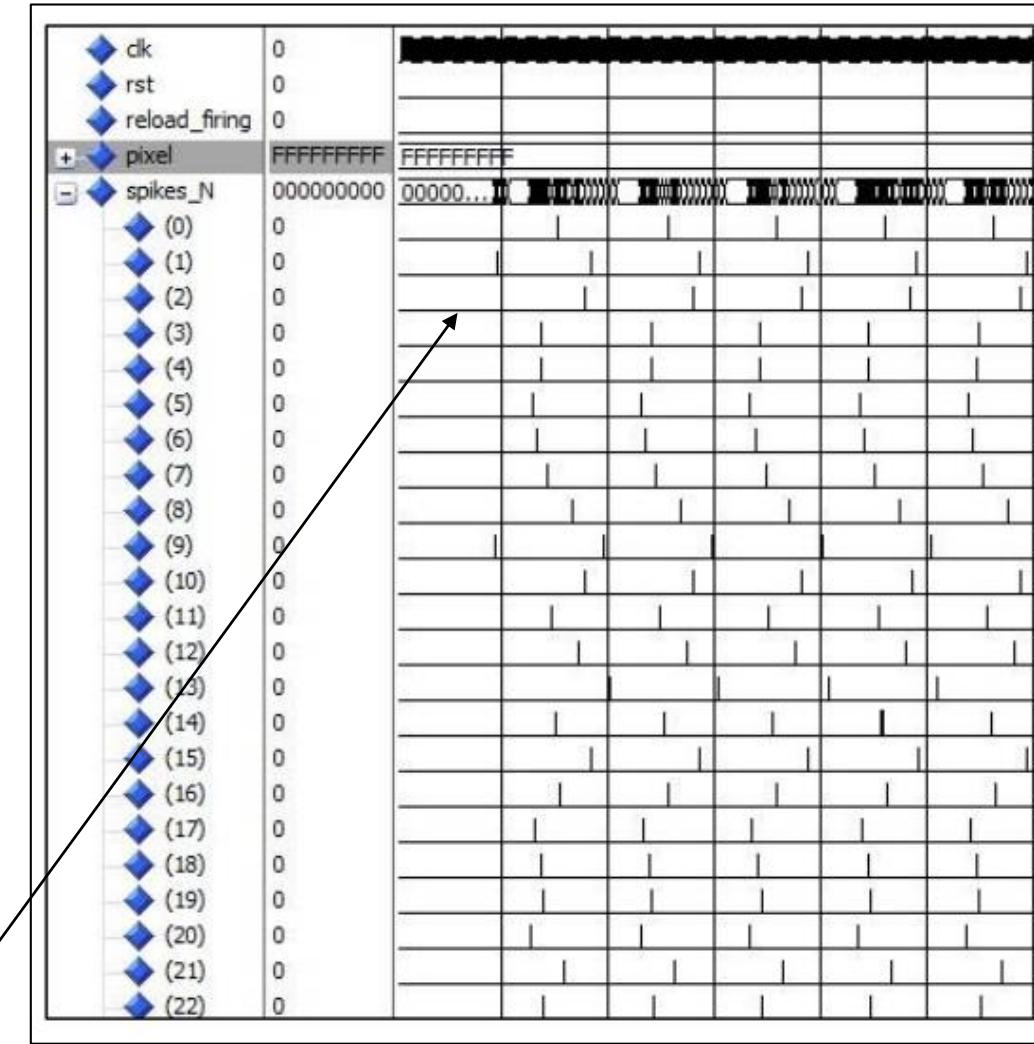




Process_Data_Input



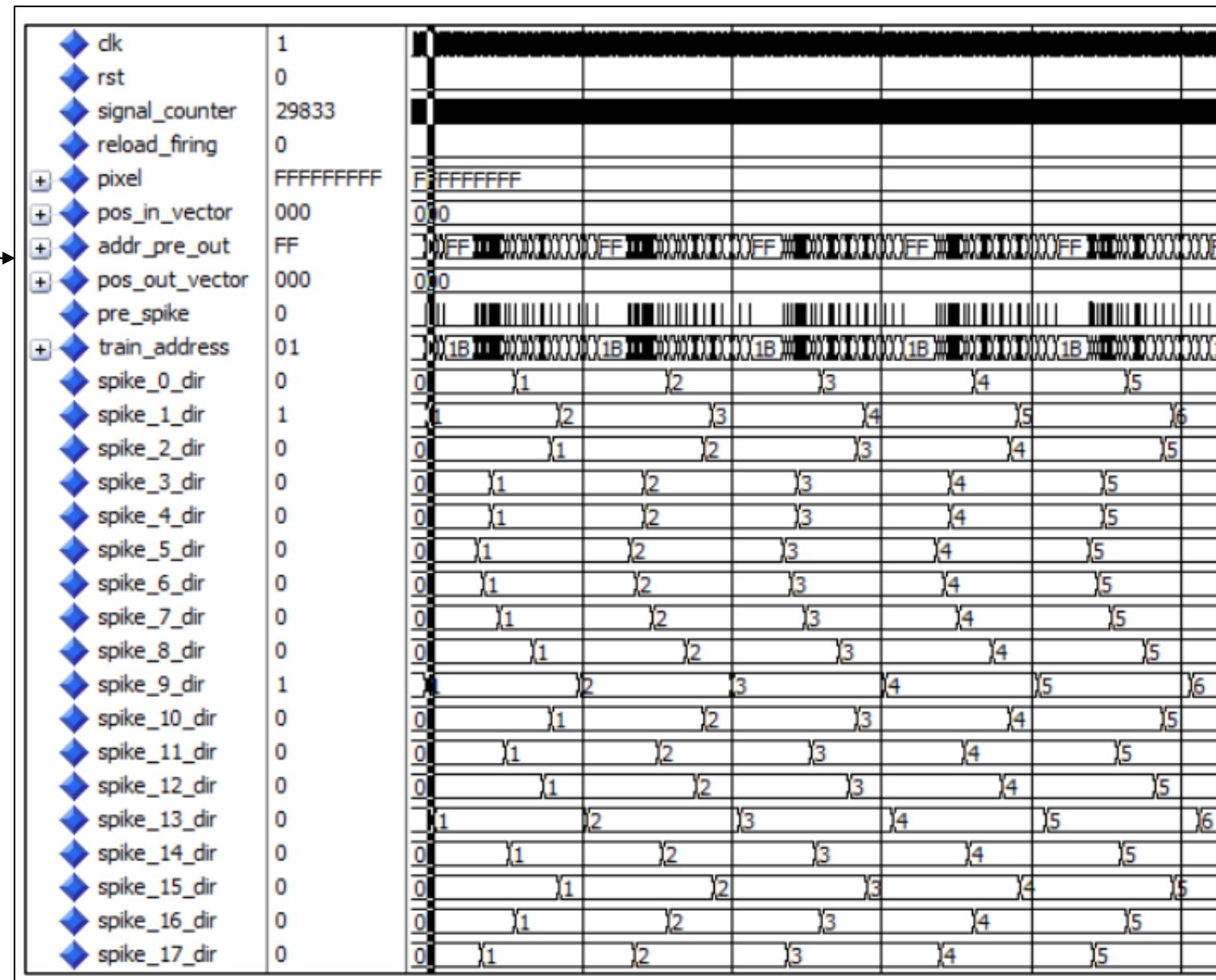
$$H(t - t_k^i - d_{ij})$$



Entry_Process

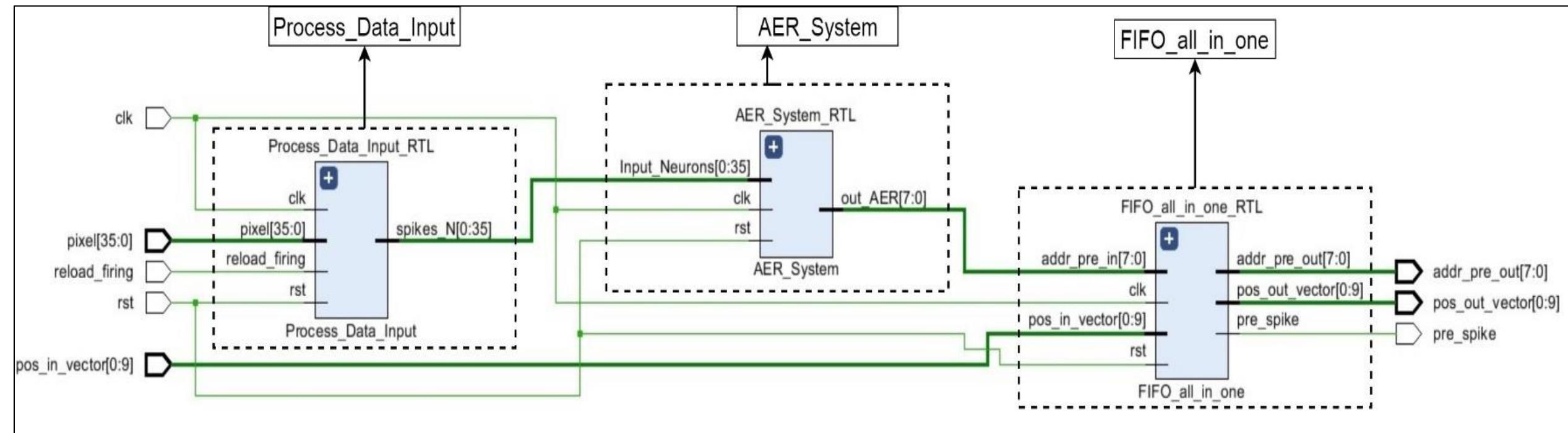
Tren de pulsos construido a base de 36 neuronas pre sinápticas (incluye retardo de propagación y FIFO)

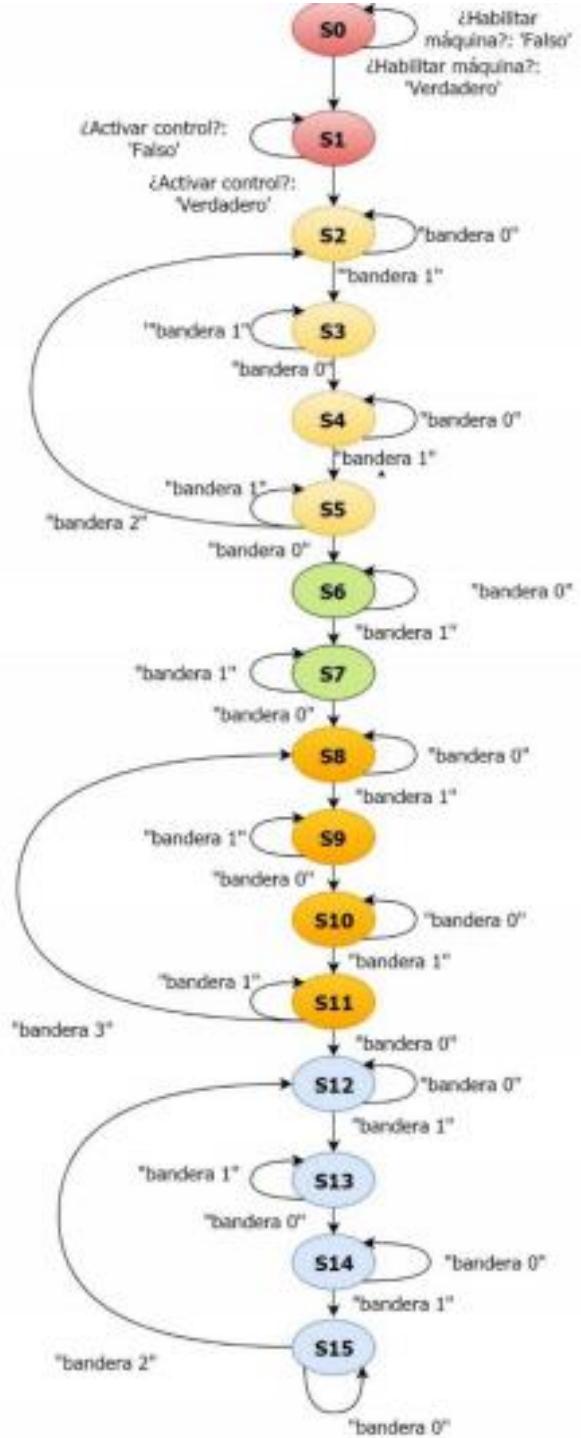
Decodificador desde banco de pruebas



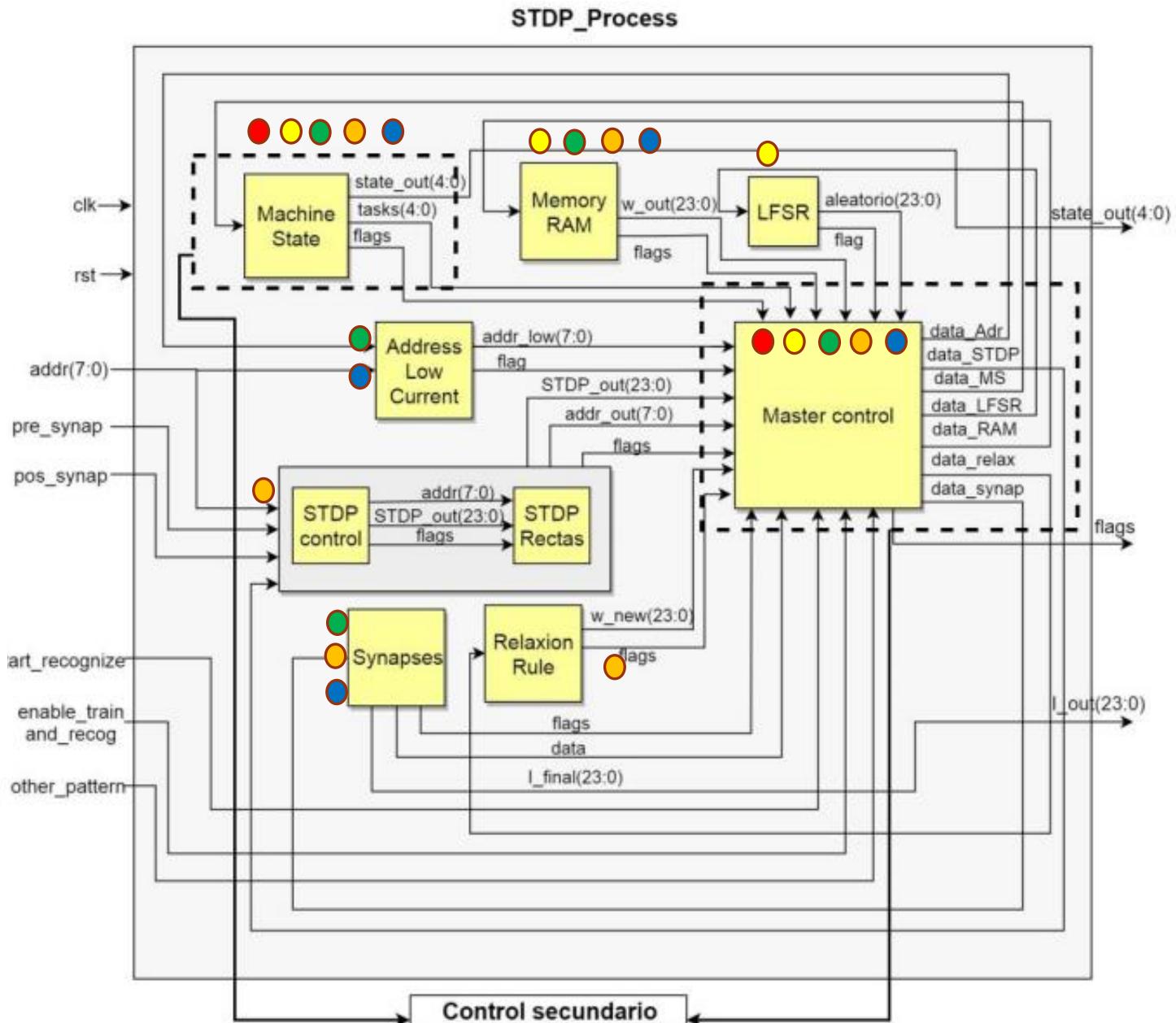


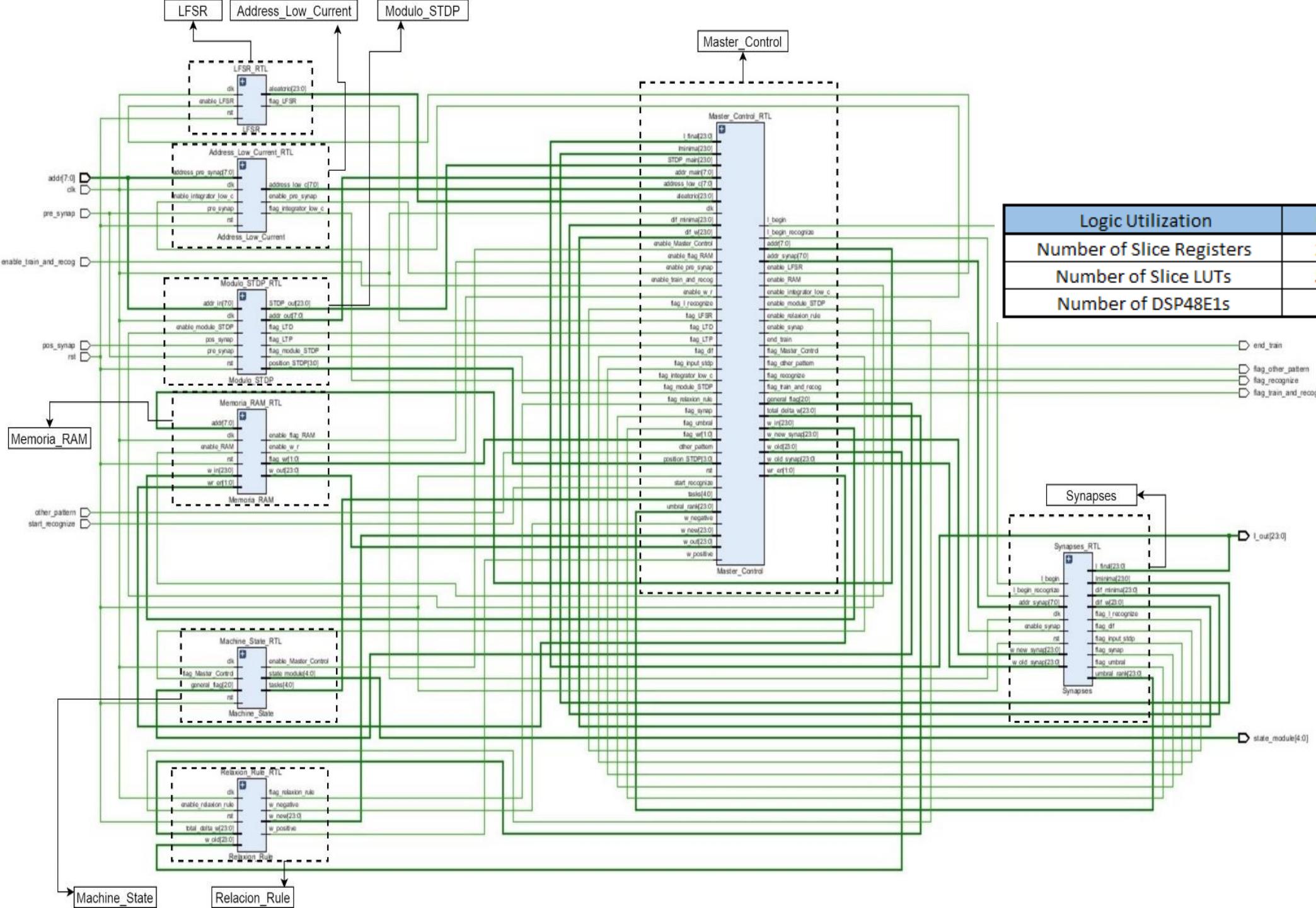
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2,654	126,800	2%
Number of Slice LUTs	7,010	63,400	11%
Number of DSP48E1s	108	240	45%





- Habilitar sistema
- Proceso memoria RAM
- Proceso de baja corriente
- Proceso de entrenamiento o aprendizaje
- Proceso de reconocimiento.





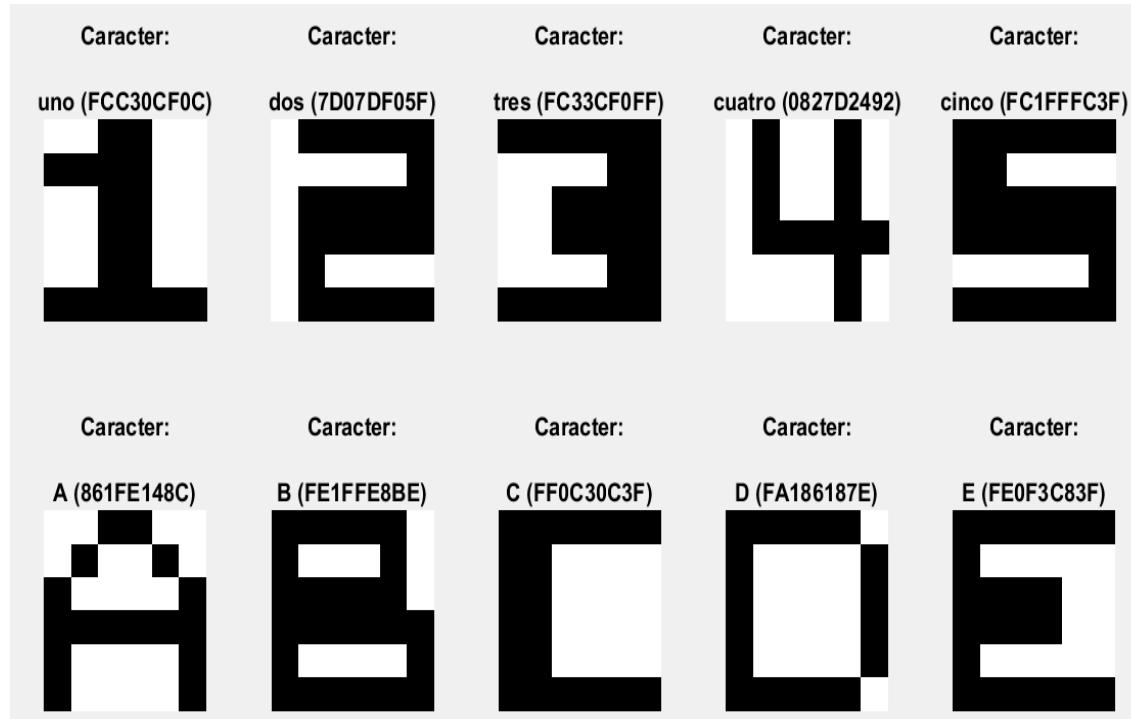
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	2,587	126,800	2%
Number of Slice LUTs	5,011	63,400	7%
Number of DSP48E1s	8	240	3%

VIVADO[®]
HLx Editions





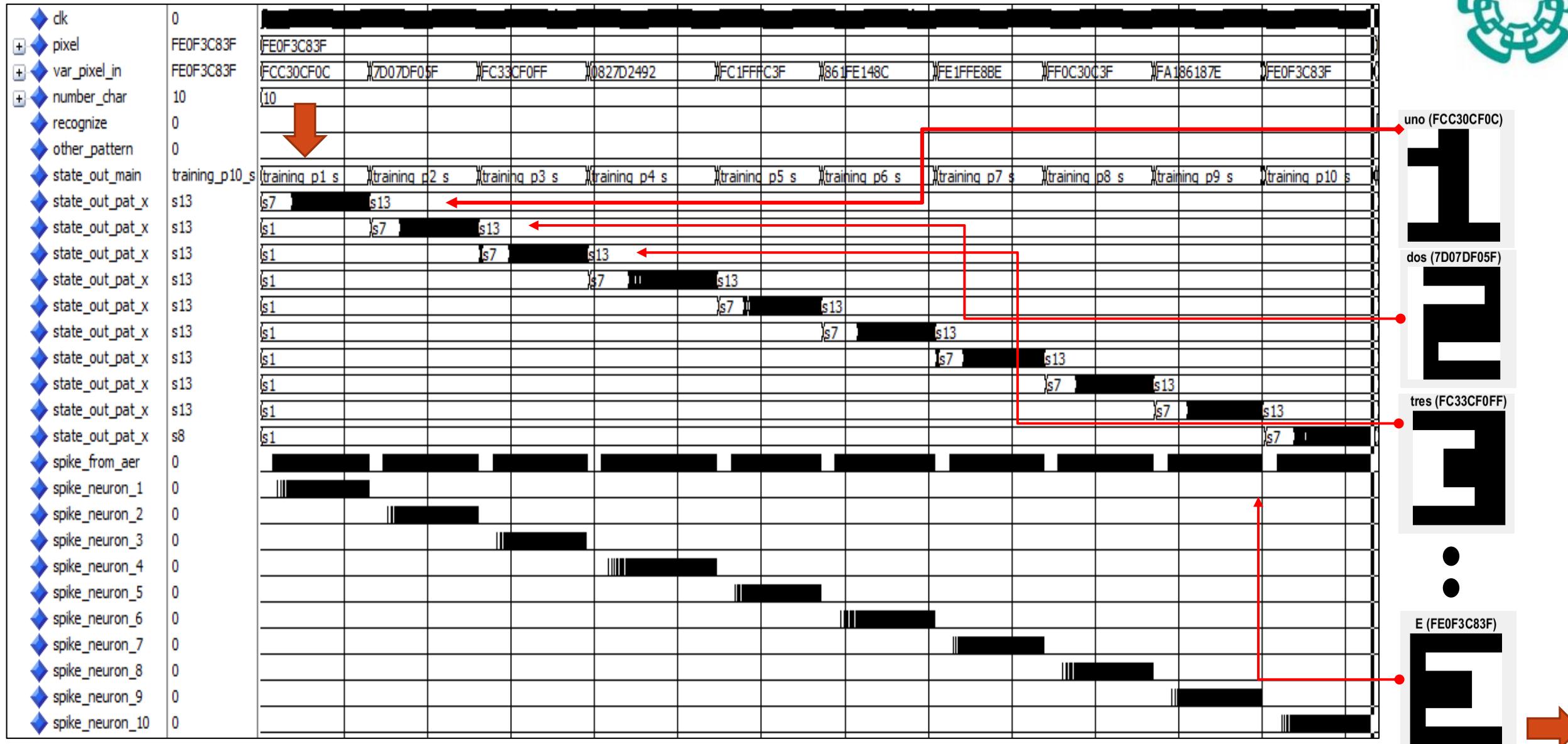
Resultados Hardware



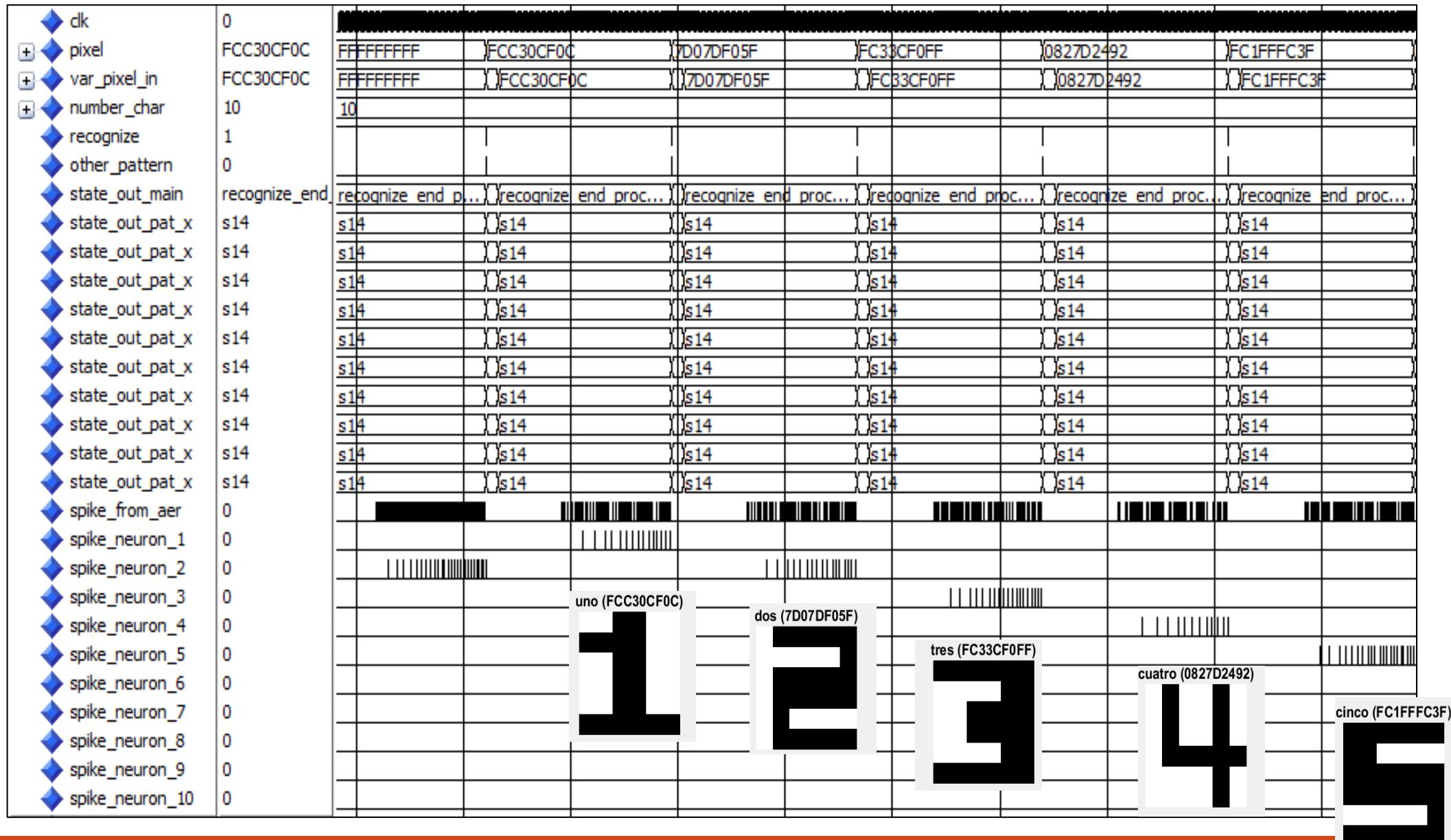
- Selectivo
- imágenes 6x6 (capa de entrada)
- 10 caracteres a clasificar
- 46 neuronas
- Épocas (pares en LTP)= 100
- Velocidad de aprendizaje = 0.5
- Reconocimiento 100%



Entrenamiento



Reconocimiento





clk	0						
pixel	FCC30CF0C	861FE148C	FE1FFE8BE	FF0C30C3F	FA186187E	FE0F3C83F	FCC30CF0C
var_pixel_in	FCC30CF0C	861FE148C	FE1FFE8BE	FF0C30C3F	FA186187E	FE0F3C83F	FCC30CF0C
number_char	10	10					
recognize	1						
other_pattern	0						
state_out_main	recognize_end	recognize end proc...					
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
state_out_pat_x	s14	s14	s14	s14	s14	s14	s14
spike_from_aer	0						
spike_neuron_1	0	A (861FE148C)	B (FE1FFE8BE)	C (FF0C30C3F)	D (FA186187E)	E (FE0F3C83F)	uno (FCC30CF0C)
spike_neuron_2	0						
spike_neuron_3	0						
spike_neuron_4	0						
spike_neuron_5	0						
spike_neuron_6	0						
spike_neuron_7	0						
spike_neuron_8	0						
spike_neuron_9	0						
spike_neuron_10	0						

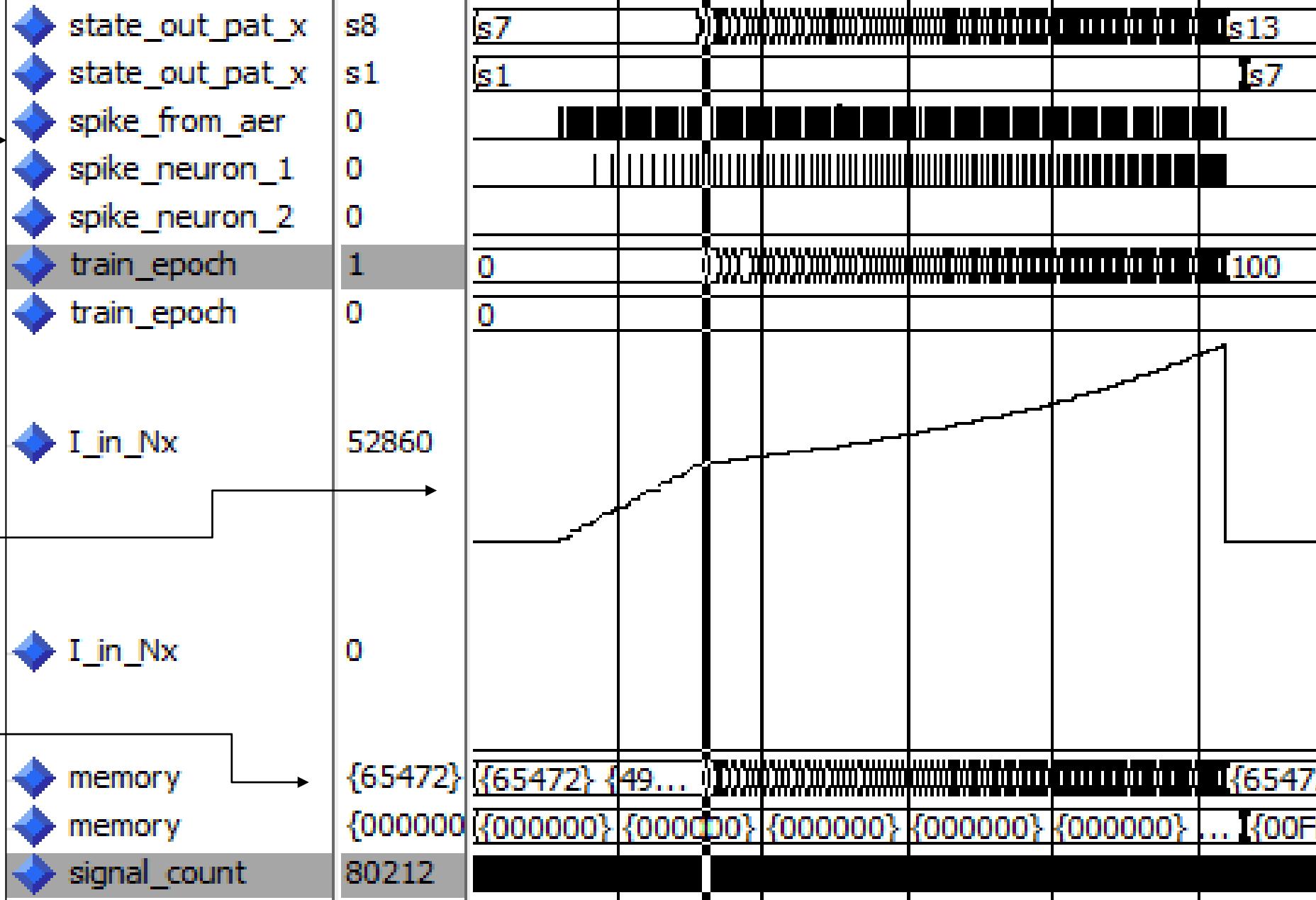


Pulsos
Pre sinápticos
(36 neuronas)

Pulsos
Post sinápticos
(1 neurona
neuronas)

Corriente
Post sináptica

Memoria RAM
36 localidades



Pares
Potenciados

17.94 ms
tiempo real



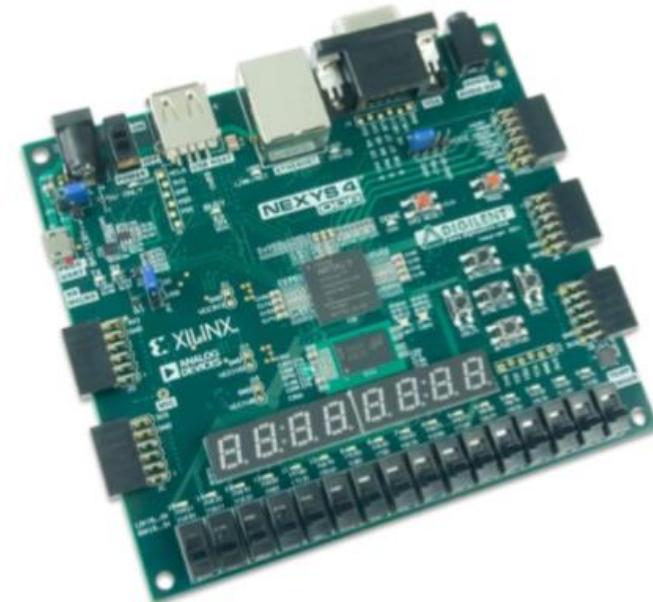
Conclusiones

- 1. Algoritmo y arquitectura propuestas son viables para el reconocimiento de caracteres simples (neurona, sinapsis, protocolo de STDP y algoritmo).**
- 2. Ambos sistemas (software y hardware) no son equivalentes a pesar de tener las mismas ventanas de aprendizaje y ejecutar el mismo algoritmo.**
- 3. Por el tipo de esquema de sistema de codificación, y dado que el algoritmo es no supervisado el sistema puede fallar.**
- 4. Necesidad de profundizar el tópicos de neurociencias y hardware neuromórfico para desarrollar sistemas de reconocimiento mas eficientes.**



Trabajo Futuro

- 1. Estudiar el rendimiento del sistema propuesto.**
- 2. Implementar etapa de interacción con usuario (PC o como sistema embebido).**
- 3. Solucionar problema de escalabilidad.**
- 4. Desarrollo de hardware Neuromórfico digital .**



The Nexys4 DDR



Referencias

- S. Daves., *Learning in spiking neural networks.*, Manchester.: Phd Thesis. University of Manchester.
- W. Mass, *Networks of Spiking Neurons: The Third Generation of Neural Networks Models.*, Austria: Institute for Theoretical Computer Science Technische Universität Fraz Klosterwiesgasse, 1997.
- J. V. Tranquillo, «Quantitative Neurophysiology. Synthesis Lectures on Biomedical Engineering #21,» Morgan & Claypool., 2008.
- H. P.-M. & S. Bohte, *Computing with Spiking Neuron Networks*, Universit de Lyon Laboratoire de Recherche en Informatique - INRIA - CNRS bat. 490, Universit Paris-Sud and Centrum Wiskunde & Informatica, Amsterdam, The Netherlands.
- E. M. Izhikevich, «Simple Model of Spiking Neurons,» *IEEE transactions on neural networks.*, vol. 14, nº 6, 2003.
- E. M. Izhikevich, «Polychronization: Computation with Spikes,» *Neural Computation MIT*, 2005.
- F. Ponulak, «Analysis of the ReSuMe Learning process for spiking neural networks.,» *International Journal of Applied Mathematics and Computer Science.(AMCS)*, vol. 18, nº 2, 2008
- Y. Z. R. W. & J. Z. Quansheng Ren, «Optical spike-timing-dependent plasticity with weight-dependent learning window and reward modulation,» *Optical Society of America*, vol. 23, nº 19, 2015.
- D. J. Heeger, *Poisson Model of Spike Generation*, Lecture Notes Center for Neural Science, New York University, 2005.
- W. G. a. W. M. Kistler, «Spiking Neuron Models: Single Neurons, Populations, Plasticity.,» de *Chapter 4. Formal Spiking Neuron Models.*, Cambridge University Press, 2002.
- A.-M.-M.-M. a. J. V. F.-V. Taras Lakymchuk, «Simplified spiking neural network architecture and STDP learning algorithm applied to image classification,» *EURASIP Journal on Image and Video Processing. Springer open journal.*, 2015..
- M.-m. P. Guo-qiang Bi, «Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type,» *The Journal of Neuroscience*, 1998.
- E. M. Izhikevich, «“Simple Model of Spiking Neurons”» *IEEE Transaction on Neural Networks*, vol. 14, nº 6, 2003.
- N. S. D. Eugene M. Izhikevich, «Relating STDP to BCM» *Neural Computation. MIT*, vol. 15, p. 1511–1523, 2003.

- H. N. Sou Nobukawa, «Synchronous spike propagation in Izhikevich neuron system with spike-timing dependent plasticity» *2012 Proceedings of SICE Annual Conference (SICE), IEEE*, pp. 453-458, 20-23 Aug. 2012.
- A. J. H. a. T. M. Kadhim, «Spiking Versus Traditional Neural Networks for Character Recognition on FPGA Platform» *Journal of Telecommunication, Electronic and Computer Engineering*, Vol. 10, Nº 3, pp. 109-115, July 2018.
- W. G. Jean-Pascal Pfister, «Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity,» *The Journal of Neuroscience*, Vol. 26, Nº 38, p. 9673-9682, 2006.
- T. M. K. Amjad J. Humaidi, «Recognition of English Characters Using Spiking Neural Networks» *International Journal of Engineering and Technology (IJET)*, Vol. 5, Nº 9, Oct-Nov 2017
- R. G. & S. J. T. Timothée Masquelier, «Competitive STDP-Based Spike Pattern Learning» *Neural Computation*, MIT, Nº 21, p. 1259–1276, 2009
- V. A. Pedroni, "Circuit Design with VHDL," MIT Press, 2004.
- D. P. Morton, *Hardware Modeling and Top-Down Design Using VHDL*, Boston, E.U.A.: M.Sc. Thesis. Massachusetts Institute of Technology, June 1991.
- W. Maass, "On the relevance of time in neural computation and learning" *Theoretical Computer Science*, Elsevier., p. 157–178, 2001.
- K. A. Boahen, "Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events" *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 47, No. 5, pp. 416-434, 2000.
- S.-C. L. T. D. R. D. G Indiveri, "Neuromorphic Systems" Elsevier, 2009.
- T. E. P. B. E. D. S. R. J. S. P. Catherine D. Schuman, "Computer Science (Cornell University) >Neural and Evolutionary Computing. A Survey of Neuromorphic Computing and Neural Networks in Hardware", 19 May 2017. [Online]. Available: <https://arxiv.org/abs/1705.06963>. [Accessed June 2019].
- S.-C. L. Giacomo Indiveri, "Memory and Information Processing in Neuromorphic Systems" *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379 - 1397, August 2015.
- S. H. M. E. S. C. M. R. L. S. F. Johannes Partzsch, "A Fixed Point Exponential Function Accelerator for a Neuromorphic Many-Core System," *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4, 2017.
- S. D. P. K. ,. A. A. Andrew Cassidy, "FPGA Based Silicon Spiking Neural Array" *IEEE Biomedical Circuits and Systems Conference*, pp. 75-78, 2007.
- J. C. V. Eduard-Guillem Merino Mallorquí, Digital System for spiking neural network emulation, Grau en Enginyeria Electrònica Industrial i Automàtica.Universitat Politècnica de Catalunya, June 2017.