SISTEMA DE RECONOCIMIENTO DE CARACTERES DE ALTA VELOCIDAD BASADO EN EVENTOS

J. A. Pérez-Carrasco [(1)], B. Acha [(1)], C. Serrano [(1)], T. Serrano-Gotarredona [(2)], and B. Linares-Barranco [(2).]

jcarrasco@imse-cnm.csic.es, bacha@us.es, cserrano@us.es, terese@imse-cnm.csic.es, bernabe@imse-cnm.csic.es

[(1)] Dpto. Teoría de la Señal, ETSIT, Universidad de Sevilla. Avda de los Descubrimientos, s/n, CP41092

[(2)] Instituto de Microelectrónica de Sevilla (IMSE-CNM-CSIC) Avda. Reina Mercedes, s/n, Sevilla. CP41012.

***Abstract*- Spike-based processing technology is capable of very high speed throughput, as it does not rely on sensing and processing sequences of frames. Besides, it allows building complex and hierarchically structured cortical-like layers for sophisticated processing. In this paper we summarize the fundamental properties of this sensing and processing technology applied to artificial vision systems and the AER (Address Event Representation) protocol used in hardware spiking systems. Finally a four-layer system is described for character recognition. The system is slightly based on the Fukushima´s Neocognitron. Realistic simulations using figures of already existing AER devices are provided, which show recognition delays under 10µs.**

I. INTRODUCCIÓN

La tecnología actual permite la implementación de aplicaciones complejas a alta velocidad y con resultados bastante eficientes. Sin embargo, cuando se trata de implementar aplicaciones que resultan inmediatas para el cerebro, como actividades de reconocimiento, de seguimiento o movimiento de objetos, etc., los sistemas electrónicos actuales resultan ineficientes. En el caso de aplicaciones de visión, la mayoría de los sistemas actuales basan su funcionamiento en el procesamiento de fotogramas. Los sistemas de visión trabajan habitualmente capturando y procesando secuencias de fotogramas (frames), que son procesados, píxel a píxel hasta que alguna tarea determinada es conseguida. Este procesamiento basado en fotogramas es lento, especialmente si se necesitan varias convoluciones en secuencia para cada imagen de entrada. El cerebro no funciona con un esquema basado en fotogramas. En la retina, cada píxel envía pulsos (también llamados eventos) a la corteza cerebral cuando su nivel de actividad alcanza cierto umbral. Aquellos píxeles muy activos enviarán más pulsos que los menos activos. Todos estos pulsos son transmitidos a medida que están siendo producidos, y no esperan el tiempo artificial "*tiempo de frame*" antes de enviarlos a la siguiente etapa de procesamiento [1]. Las características extraídas son propagadas y procesadas etapa por etapa tan pronto como han sido producidas, sin esperar a finalizar la recolección y procesamiento de los datos de fotogramas completos. Un problema importante que encuentran los ingenieros cuando tratan de implementar sistemas de procesamiento de visión bio-inspirados es conseguir la masiva cantidad de interconexiones hacia delante y de realimentación que aparece entre las etapas neuronales existentes en el sistema

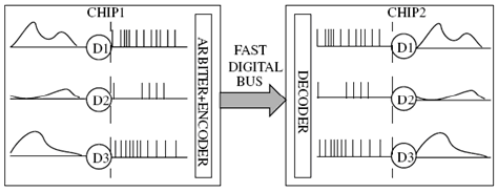

Fig. 1. Concepto de comunicación punto a punto basada en AER.

de procesamiento de visión humano. La representación de datos basada en direcciones de eventos AER (*Address Event Representation*) [2] es una posible solución. La Fig. 1 ilustra la comunicación en un enlace punto a punto AER tradicional.

En la figura, el estado continuo en el tiempo de las neuronas emisoras en un chip es transformado a una secuencia de pulsos digitales muy rápidos (eventos) de anchura mínima (del orden de *ns*) pero con intervalos entre pulsos del orden de *ms* (similar a las neuronas cerebrales). Este alto intervalo entre pulsos permite una potente multiplexación, y los pulsos generados por las neuronas emisoras pueden ser multiplexados en tiempo en un bus de salida común de alta velocidad. Cada vez que una neurona emite un pulso o evento, la dirección de esa neurona aparece en el bus digital, junto con sus señales *request* y *acknowledge*. Esto se conoce como *evento de dirección*. El chip receptor lee y decodifica las direcciones de los eventos entrantes y envía pulsos a las neuronas receptoras correspondientes, que integran esos pulsos y son capaces de reproducir el estado de las neuronas emisoras. Esta es la comunicación entre chips basada en AER más simple.

Sin embargo, esta comunicación punto a punto puede ser extendida a un esquema multiemisor o multireceptor [3], donde rotaciones, traslaciones o procesamientos más complicados como convoluciones pueden ser implementados por chips de procesamiento que reciben estos eventos [4]. Además, la información puede ser trasladada o rotada fácilmente simplemente cambiando las direcciones de los eventos al tiempo que viajan de un chip al siguiente. Existe una creciente comunidad de usuarios del protocolo AER para el diseño de aplicaciones de visión y audición bio-inspiradas, robótica, seguimiento y reconocimiento de objetos, etc., como ha sido demostrado por el éxito en los últimos años de los participantes en las "*Neuromorphic Engineering Workshop series*" [5]. El éxito de esta comunidad es diseñar sistemas grandes jerárquicamente estructurados multichip

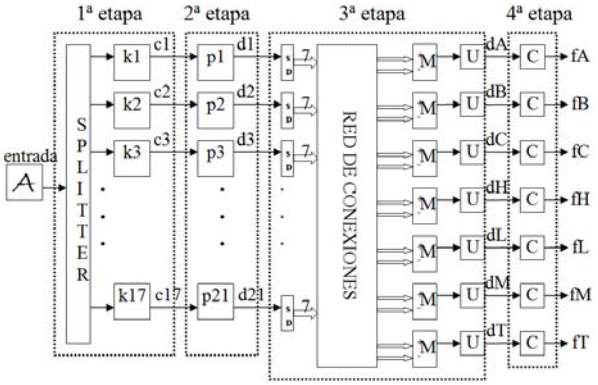

Fig. 2. Sistema de Reconocimiento de Carácteres basado en AER

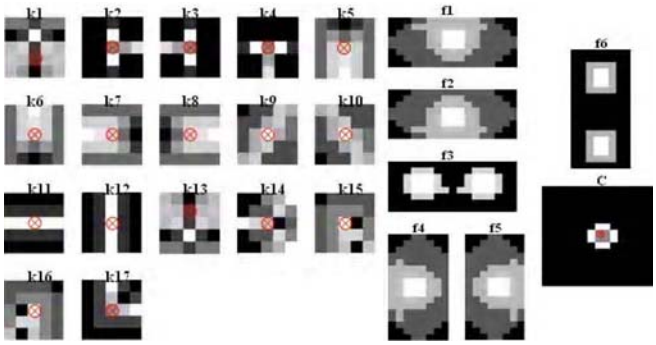

Fig. 3. Máscaras utilizadas en los diferentes módulos de convolución descritos en el sistema de la Fig. 2

multietapa capaces de implementar procesamientos complejos de matrices en tiempo real. Pero el principal hito que ha permitido un desarrollo exponencial de la tecnología AER fue el diseño de los primeros chips de convolución basados en AER libres de fotogramas [4][6]. En estos chips, cada vez que un evento es recibido a la entrada, una máscara de convolución almacenada como parámetro es añadida en un array de píxeles alrededor de la dirección codificada por el evento. Cuando un píxel en el array alcanza un umbral determinado programable, un evento codificando la dirección del píxel se envía a la salida y el píxel es reseteado.

En este artículo se presenta un sistema multichip multietapa basado en AER ligeramente inspirado en el Neocognitron de Fukushima [7][8] para el reconocimiento de caracteres.

II. SISTEMA DE RECONOCIMIENTO DE CARACTERES

El sistema AER multietapa multiprocesamiento implementado en este artículo corresponde a un sistema de reconocimiento de caracteres escritos a mano. En particular, se ha implementado una versión simplificada de la arquitectura de reconocimiento de caracteres Neocognitron [8]. La estructura implementada permite el reconocimiento de varios caracteres (‘A’,’B’,’C’,’H’,’L’,’M’,’T’) que además pueden presentar ligeras deformaciones.

El sistema se representa en la Fig. 2. La entrada al sistema es un estímulo visual de 16x16 píxeles codificado en eventos y que puede representar uno de los siete caracteres (‘A’,’B’,’C’,’H’,’L’,’M’,’T’). Cada pixel produce 10 eventos y la separación entre eventos es de 50*ns*. Como el número aproximado de píxeles activos es de 30, el estímulo completo tiene una duración de 15μs aproximadamente.

La primera etapa de procesamiento implementa 17 convoluciones (con máscaras de convolución *ki* (i = 1,…,17) en paralelo para la extracción de carácterísticas en el estímulo visual. Cada máscara de convolución (kernel) en la etapa ‘1’ tiene como objetivo detectar características discriminatorias que ayuden a identificar los caracteres. Los kérneles utilizados en esta etapa se muestran en la parte izquierda de la Fig. 3, normalizados de ‘0’ a ‘1’. La cruz roja en las máscaras representa el origen de coordenadas.

En el sistema de la Fig. 2, cada módulo de convolución está configurado para no enviar eventos negativos. Sólo eventos positivos salen del chip, por tanto, cada módulo de convolución implementa una rectificación de media onda además de la convolución programada.

La máscara de convolución *k1* detecta la presencia y posición del pico superior en la letra ‘A’. La máscara *k2* detecta un segmento horizontal que termina en la izquierda y toca un segmento vertical. La máscara *k3* igual, pero terminando el segmento a la derecha, la máscara *k4* detecta un segmento vertical que termina en la parte superior y toca un segmento horizontal. La máscara *k5* detecta el tope inferior de un segmento vertical, la máscara *k6* igual pero para el tope superior. La máscara *k7* detecta el límite izquierdo de un segmento horizontal, la máscara *k8* igual pero para el límite derecho. La máscara *k9* detecta la curvatura superior de la letra ‘C’ y la máscara *k10* la curvatura inferior. La máscara *k11* detecta un segmento horizontal y la máscara *k12* uno vertical. La máscara k13 tiene como objetivo detectar el punto de cruce central entre los dos segmentos inclinados en la letra ‘M’, la máscara *k14* detecta el punto de cruce entre las dos curvaturas derechas de la letra ‘B’. La máscara *k15* detecta el pico izquierdo superior en la letra ‘M’. La máscara *k16* igual pero para el pico derecho superior. Finalmente la máscara *k17* detecta el punto de cruce entre los segmentos horizontal y vertical de la letra ‘L’. Como se puede observar, la primera etapa de convoluciones tiene como intención detectar un conjunto de 17 características geométricas que pueden ser usadas para detectar y discriminar entre las diferentes letras.

La letra ‘A’ debería producir actividad en las salidas {c1, c2, c3, c5,c11}, la letra ‘B’ en {c2, c11, c12, c14, c17}, la letra ‘C’ en {c8, c9, c10, c11, c12}, la letra ‘H’ en {c2, c3, c5, c6, c11, c12}, la letra ‘L’ produce salida en {c6, c8, c11, c12, c17}, la letra ‘M’ en {c5, c12, c13, c15, c16} y la letra ‘T’ en {c4, c5, c7, c8, c11, c12}.

La segunda etapa implementa 21 convoluciones en paralelo (con mascaras *pi (i =1,…, 21)*). Hay solamente seis máscaras diferentes en esta etapa, *fi (i=1,..,6)* que se muestran en la Fig. 3 normalizadas de ‘0’ a ‘1’. El objetivo de esta etapa es evaluar si la distribución espacial de las características detectadas en la primera etapa es significativa para el carácter que está siendo analizado. Por ejemplo, para el carácter ‘A’, el pico superior (detectado por la máscara *k1* en la primera etapa) debería estar por encima del resto de características. Por tanto, la máscara *p1* producirá una contribución positiva en la región justo debajo del pico detectado por *k1*, porque éste será el lugar donde debería estar el centro del carácter ‘A’. Del mismo modo, si hay actividad en el canal c2, el centro de la letra ‘A’ debería estar

						ACTIVIDAD SALIDA (NÚMERO DE EVENTOS)						
LETRA	ESTÍMULO(μs)	T2:TIEMPO PRIMERA SALIDA (μs)	T1-T2(μs)	TIEMPO ÚLT. SALIDA (μs)	TASA RECONOC	fA	fB	fC	fH	fL	fM	fT
A1	15	7,45	7,55	33,23	100	69	0	11	0	0	0	0
A2	12,95	11,3	1,65	27,15	100	28	0	6	1	0	1	0
A3	14,45	7,97	6,48	32,02	100	72	0	4	0	0	0	0
B1	17,95	14,64	3,31	37,08	100	3	43	0	12	0	8	0
B2	19,95	16,58	3,37	37,91	100	5	21	0	9	0	0	0
B3	17,45	17,15	0,3	40,07	100	0	36	9	1	1	8	0
C1	9,95	7,74	2,21	25,9	100	0	8	72	9	12	0	0
C2	10,95	7,65	3,3	21,18	100	0	19	76	16	34	0	7
C3	7,95	7,71	0,24	28,44	100	1	0	33	0	0	0	0
H1	14,95	5,97	8,98	48,62	100	0	0	0	49	8	0	0
H2	12,95	14,76	-1,81	29	100	1	0	0	16	12	4	0
H3	12,45	16,23	-3,78	45,32	100	0	0	0	22	9	1	0
L1	9,45	5,44	4,01	23,88	100	0	12	0	8	47	0	0
L2	6,95	6,72	0,23	23,62	100	0	8	0	0	32	0	0
L3	6,45	7,65	-1,2	25,26	100	0	0	12	7	31	0	0
M1	15,45	5,41	10,04	27,83	100	0	1	0	9	0	83	0
M2	14,95	9,19	5,76	27,53	100	0	0	0	11	1	33	0
M3	13,95	5,74	8,21	38,68	100	0	0	0	0	6	50	0
T1	9,95	7,11	2,84	11,04	100	6	1	0	5	0	0	23
T2	8,45	7,21	1,24	14,19	100	7	10	0	7	0	0	18
T3	7,95	5,93	2,02	20,26	100	0	1	0	4	0	0	23
PROMEDIO	12,40	9,31	3,09	29,44	100,00							

Tabla 1. Tiempos y número de eventos obtenidos tras la simulación

a la derecha. Por tanto, la máscara *p2* producirá contribución en los píxeles en d2 que están a la derecha de esos que dispararon eventos en c2. La salida de c3 se trata simétricamente a la salida de c2. La máscara *k5* envía eventos a c5 si se detecta la parte inferior de un segmento vertical. Esto significa que el centro de la letra 'A' está en algún lugar por arriba, o a la izquierda o a la derecha. De esta contribución se encarga la máscara *p5*. Finalmente, si hay actividad en c11, el centro de 'A' debería estar en la misma posición (centro de la letra 'A'). De este modo, cuando la entrada es el carácter 'A', la actividad en {c1, c2, c3, c5, c11} estará en diferentes píxeles. Sin embargo, la etapa 2 hará que la actividad en {d1, d2, d3, d5, d11} esté en el centro de la letra 'A'. Algo similar ocurre con el resto de caracteres, cuyo centro quedará definido gracias a las máscaras *pi* de la etapa 2, que se encargan de implementar la ponderación espacial de las características detectadas en la etapa 1.

El propósito de la tercera etapa es combinar con pesos positivos o negativos las salidas de la segunda etapa. Por ejemplo, para la letra 'A', las salidas {d1, d2, d3, d5, d11} deberían contribuir positivamente, mientras que las salidas {d4, d6, d7, d8, d9, d10, d12, d13, d14, d15, d16, d17, d18, d19, d21} deberían inhibir. Algo similar ocurre con el resto de caracteres. Para ello, cada una de las 17 salidas obtenidas en la etapa 2 es replicada en 7 canales independientes. Cada uno de estos canales entra a un módulo merger de 17 entradas (módulo *M* en la Fig. 2). Como los eventos que proceden de la segunda etapa tienen todos signo positivo, los módulos merger tienen cableados los signos en las entradas, de modo que se le asigna signo positivo a los eventos que contribuyen positivamente para el reconocimiento del carácter y negativo a los que contribuyen negativamente. Los eventos con nuevo signo obtenidos de cada modulo merger son enviados a un módulo de convolución con una máscara de convolución 1x1 y con valor 1 (módulo *U* en la Fig. 2). Los parámetros de los módulos de convolución son ajustados de modo que hacen falta 3 eventos a la entrada codificando la misma dirección para producir la generación de un evento. De este modo, en la tercera etapa se implementa una suma de las actividades recibidas a la entrada.

Finalmente, la cuarta etapa consiste de un módulo de convolución para cada canal de salida de la etapa 3 (módulos

Fig. 4. Caracteres utilizados para evaluar el sistema AER.

C en Fig. 2). Hay un módulo *C* para cada carácter (siete en total) y el objetivo de estos módulos es analizar si los eventos que vienen de las etapas previas están más o menos agrupados, en lugar de dispersos. Si están agrupados (en el centro del carácter) significa que el carácter ha sido detectado.

El kernel utilizado en los módulos *C* se muestra en la parte inferior derecha de la Fig. 3. De nuevo la cruz roja muestra el origen de coordenadas del kernel.

III. RESULTADOS

Debido a que la mayoría de los dispositivos AER hardware existentes en la actualidad son prototipos, se dispone de un número escaso de ellos y resulta inviable montar un sistema como el descrito en la Fig. 2. Por ello se ha simulado el sistema propuesto con una herramienta de simulación AER desarrollada en C++ y validada [9][10] que utiliza características y valores reales de dispositivos ya fabricados. Al estar implementado todo el sistema con módulos AER, todo el procesamiento es en paralelo y en tiempo real, siendo los eventos enviados de etapa a etapa en cuestión de *ns*. El sistema multi-chip (68 módulos de convolución) multi-etapa (4 etapas) ha sido testeado usando tres versiones ligeramente modificadas de cada uno de los 7 caracteres. Los 21 caracteres se muestran en la Fig. 4.

Los resultados obtenidos tras testear el sistema con cada uno de los 21 flujos de eventos se muestran en la Tabla 1. Se puede observar que en todos los casos, el sistema es capaz de detectar qué letra está presente en menos de 9,3μs (equivalente a procesar 100000 imágenes por segundo aproximadamente en un sistema que estuviera basado en fotogramas) desde que el primer evento de entrada es recibido por el sistema. Este retraso es incluso menor que la duración del estímulo de entrada visual (12,4μs). Por tanto, el sistema es capaz de reconocer el carácter a la entrada del sistema antes de haber recibido y procesado todos los

eventos. En una versión implementada en fotogramas, habría que esperar el tiempo correspondiente a un fotograma para recoger todos los valores de los píxeles correspondientes a un carácter, y después de eso, deberiamos procesar toda los píxeles de la imagen secuencialmente con los 68 módulos de convolución en el sistema. Si suponemos un esquema de codificación de 25 fotogramas por segundo, tendriamos siempre una limitacion de 40*ms* para procesar cada carácter (sin considerar los tiempos de procesamiento de los modulos de convolución).

La Fig. 5 muestra los eventos obtenidos en diferentes canales del sistema cuando el carácter codificado a la entrada es 'A'. Nótese cómo el sistema produce los eventos de salida antes de haber procesado el flujo de entrada completo.

IV. CONCLUSIONES

A lo largo de este trabajo hemos presentado un sistema de reconocimiento de caracteres multichip multietapa basado en eventos (y por tanto libre de fotogramas). Los módulos de convolución utilizados, basados en dispositivos físicos reales, permiten implementar convoluciones y procesamiento en tiempo real, de modo que los píxeles en los arrays almacenados en los módulos de convolución generan eventos a la salida que son transmitidos a las siguientes etapas aún sin haber recibido estos módulos el resto de eventos a la entrada.

El sistema proporciona en todos los casos una tasa de reconocimiento del 100% y es capaz de producir eventos positivos indicando la detección del carácter en sólo 9.3μs, inferior al tiempo medio de duración del estímulo de entrada (12.4μs).

El sistema ha sido probado con un conjunto de siete caracteres elegidos al azar para probar la robustez de la detección de las características. Sin embargo, sería bastante fácil modificar el sistema para detectar más caracteres diferentes mediante la adición de más módulos en paralelo, en todas las etapas. La adición de nuevos módulos en las etapas 1 y 2 tendría como objetivo la adición de nuevas características. Sin embargo, como la etapa 3 implementa combinaciones de las características extraídas, sólo sería necesario añadir un número reducido de características a detectar, ya que la combinación de todas ellas permite un grado de libertad para la detección de un conjunto grande de caracteres. El número de módulos de convolución en las etapas 3 y 4 crecería de forma lineal con el número de caracteres a detectar (nótese que por ejemplo la etapa 4 tiene un módulo de convolución por cada carácter). Todo ello sería sin penalizar los retrasos, que es lo más importante de la tecnología AER en el sistema propuesto.

El sistema descrito consta de 68 módulos de convolución. Este elevado número de módulos hace que el sistema no pueda ser implementado aún en la actualidad, pero gracias a la herramienta de simulación de sistemas AER existente es posible proponer y simular sistemas antes de que la tecnología electrónica esté disponible para ello. Para las simulaciones se han empleado medidas reales de las características físicas y electrónicas de los dispositivos (principalmente chips de convolución y módulos *splitters* y *mergers*).

El diseño de sistemas como el descrito en este trabajo se está convirtiendo en una realidad ya que los sistemas

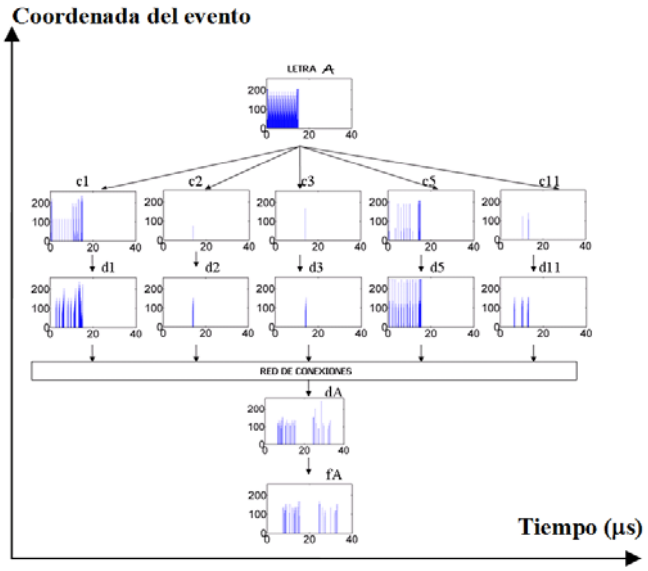

Fig. 5. Eventos obtenidos en diferentes canales cuando la entrada es 'A'.

multichip pueden ser implementados eficientemente con las tecnologías recientemente desarrolladas híbridas CMOS/noCMOS de escala nanométrica [11].

AGRADECIMIENTOS

Este trabajo ha sido financiado en parte por el proyecto TEC2009-10639-C04-01 (VULCANO) y el proyecto andaluz P06-TIC-01417 (Brain System). JAPC ha sido financiado por el proyecto andaluz P06-TIC-01417 (Brain System).

REFERENCIAS

[1] G. M. Shepherd, The Synaptic Organization of the Brain, Oxford University Press, 3rd Edition, 1990.

[2] K. Boahen, "Point-to-Point Connectivity Between Neuromorphic Chips Using Address Events," IEEE Trans. On Circuits and Systems Part-II, vol. 47, No. 5, pp. 416-434, May 2000.

[3] [50] R. Serrano-Gotarredona, et al., "AER Building Blocks for Multi-Layers Multi- Chips Neu-romorphic Vision Systems", in Advances in Neural Information Processing Systems, Vol. 18, Y. Weiss and B. S. and J. Platt (Eds.), (NIPS'06), MIT Press, Cambridge, MA, 1217-1224, (2006).

[4] T. Serrano-Gotarredona, A. G. Andreou, and B. Linares-Barranco, "AER image filtering architecture for vision-processing systems," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol. 46(9), pp. 1064-1071, Sep. 1999.

[5] A. Cohen, et al. "Rep. 2004 Workshop on Neuromorphic Eng.", Telluride, CO, Jun. 27 to Jul. 17 2004.

[6] L. A. Camunas-Mesa, A. Linares-Barranco, A. J. Acosta-Jim‼ enez, T. Serrano-Gotarredona, B. Linares Barranco, "Improved Aer Convolution Chip for Vision Processing With Higher Resolution and New Functionalities," Conference on Design of Circuits and Integrated Systems 2009. Num. 21. Barcelona. DCIS. 2009. Pag. 1-6.

[7] K. Fukushima: ''Visual feature extraction by a multilayered network of analog threshold elements,'' IEEE Transactions on Systems Science and Cybernetics, SSC-5 (4), pp. 322-333, Oct. 1969.

[8] K. Fukushima, "Analysis of the process of visual pattern recognition by the neocognitron", Neural Networks, vol. 2, pp. 413-420, 1989..

[9] J. A. Pérez-Carrasco, et al., "Advanced vision processing systems: Spike-based simulation and processing", LNCS, vol. 5807, pp. 640–651, 2009.

[10] J.A. Pérez-Carrasco, et al. "Simulador de sistemas AER basado en eventos", Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2008) Madrid, España.

[11] D. B. Strukov and K. K. Likharev, "Cmol FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," Nanotechnology, vol. 16, no. 6, pp. 888–900, 2005.