

Estructuras diseñadas a medida para adquisición de datos de un sensor de visión dinámica usando el protocolo AER en un FPGA

Nicolás Maldonado Orozco

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
nicolas.maldonado.orozco@gmail.com

J.Roberto Reyes Barón

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
juan.raygoza@ucei.udg.mx

Juan José Raygoza P.

CUCEI, Universidad de Guadalajara, Departamento de Electrónica, México
Blvd. Marcelino García Barragán # 1421, C.P. 44430, Guadalajara, Jalisco, México
jose.reyes@red.cucei.udg.mx

Susana Ortega Cisneros

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
susana.ortega@gdl.cinvestav.mx

Juan Luis del Valle

Centro de Investigación y de Estudios Avanzados del I.P.N, CINVESTAV, Unidad Guadalajara
Av. del Bosque 1145, colonia el Bajío, Zapopan, C.P. 45019, Jalisco, México
juan.delvalle@cts-design.com

Resumen

La retina humana genera impulsos representados por los cambios en la reflectancia de los foto-receptores, cuenta con amplio rango dinámico que permite a la visión adecuarse a diferentes condiciones de iluminación. El Sensor de Visión Dinámica (DVS) cuenta con una matriz de pixeles que generan impulsos cuando hay cambios de iluminación. El DVS emula la función de la retina humana, procesa los cambios generados por la reflectancia en los pixeles, logra un bajo consumo de potencia y un amplio rango dinámico por el uso de configuraciones de transistores CMOS en nivel sub-umbral. Los impulsos generados por los pixeles, se transfieren a través del protocolo de comunicación asíncrono AER; Representación de eventos a través de direcciones (*Address Event Representation*). Este trabajo presenta el diseño de estructuras asíncronas utilizando la característica de modularidad, estas estructuras fueron probadas con diferentes configuraciones, variando profundidad en bloques de control asíncrono y el número de elementos de retardo, el diseño de un conjunto de bloques de control para flujo de datos asíncronos basada en el protocolo de riel simple a cuatro fases, un análisis del impacto de la profundidad de estos bloques y la fiabilidad encontrada en la transferencia de datos del circuito *self-timed* diseñado.

Palabras Claves: AER, DVS, FPGA, protocolo de riel simple

1. Introducción

1.1. Circuitos síncronos y asíncronos

La mayoría de los diseños de circuitos digitales están sincronizados con una señal de reloj global, la cual es necesaria para forzar los estados de transición de forma correcta. Conforme el área del circuito integrado aumenta, la longitud de la ruta global se incrementa así como la carga capacitiva, llegando a ser un problema importante a considerar, ya que puede causar que los flancos de la señal de reloj llegue en

diferentes tiempos a los distintos componentes o bloques del circuito, a este fenómeno se conoce como *Clock Skew* [1, 2].

En un sistema de lógica síncrona convencional, todas las partes de la arquitectura consumen potencia con cada flanco de reloj, desarrollen o no algún trabajo útil en dicho sistema. Los circuitos lógicos asíncronos puros son controlados por eventos [3]. Estos circuitos permanecen inactivos con un consumo de potencia mínimo o nulo [4], hasta que un evento lógico o una secuencia de eventos se transmiten a través de la red lógica, al terminar su proceso o tarea el circuito asíncrono regresa a la inactividad. Así, la lógica asíncrona permite una solución tanto a los problemas de desfase de la señal de reloj como al ahorro en el consumo de potencia [5].

1.2. Protocolo de riel simple a cuatro fases

En este protocolo los datos que se envían se encuentran en una sola línea de transmisión. Además, se emplean dos señales de control para realizar la comunicación entre los bloques emisor y receptor (Fig. 1(a)), estas señales son *request* (petición) y *acknowledge* (reconocimiento) [6, 7].

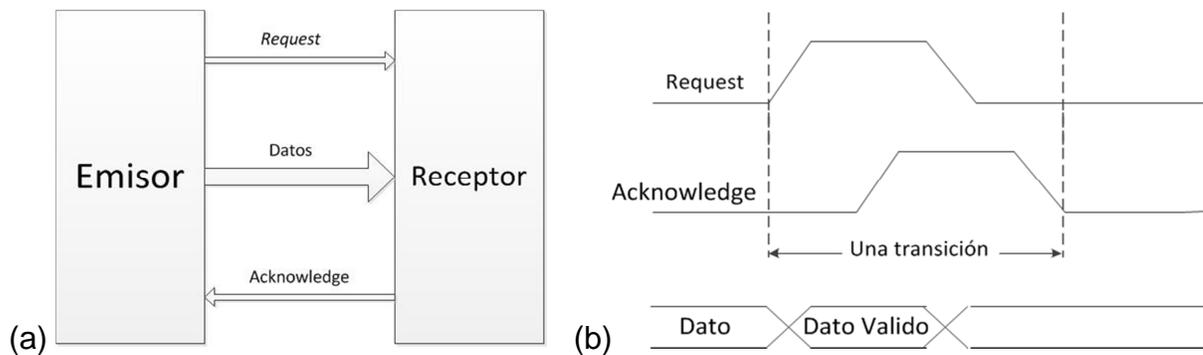


Fig. 1. (a) Comunicación con protocolo de riel simple (b) Protocolo de riel simple a 4 fases.

Los pasos necesarios para la comunicación en este protocolo son los siguientes:

1. El emisor coloca un dato en el bus de transmisión y pone la señal de *request* en nivel alto.
2. El receptor toma el dato y pone en nivel alto la señal de *acknowledge*.
3. El emisor responde poniendo la señal de *request* en nivel bajo.
4. El receptor termina la transferencia poniendo la señal de *acknowledge* en nivel bajo.

La representación de estos pasos se muestra en Fig. 1(b).

Para que se generen correctamente las señales, es necesario una fase de retorno a cero que restaure el estado que tenían las variables de control antes de una transferencia [7].

1.2.1. Bloques de Control Asíncrono BCA

Para la realización de este bloque de control es necesario el uso de Muller-C (Fig. 2(a)), estos elementos tienen el comportamiento de una compuerta AND de eventos [6]. Si se presenta el mismo nivel lógico en ambas entradas, la salida toma dicho nivel. Si alguna de las entradas cambia de nivel, la salida se mantiene, el diagrama del comportamiento de este elemento se presenta en Fig. 2(b).

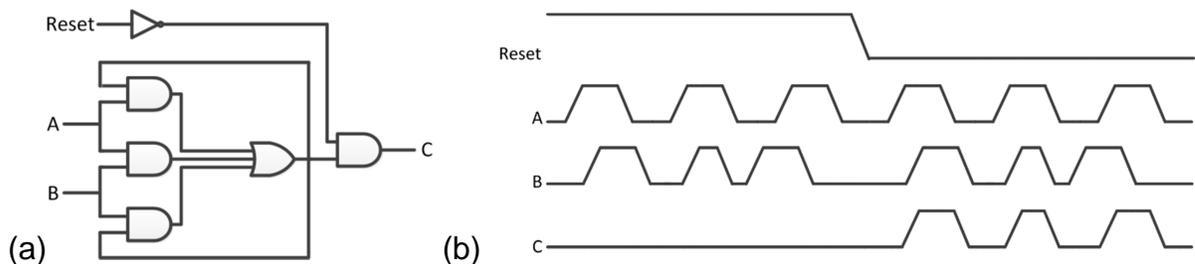


Fig. 2. (a) Diagrama lógico Muller-c (b) Comportamiento Muller-c.

La implementación de un BCA se muestra en Fig. 3(a), este utiliza elementos Muller-C y compuertas NOT, el diagrama de su comportamiento se muestra en Fig. 3(b).

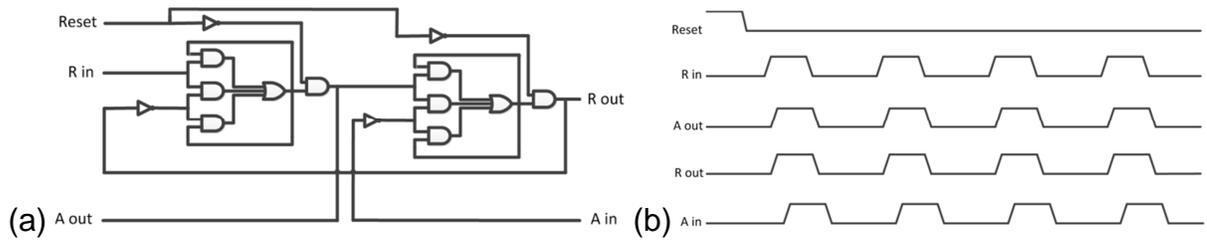


Fig. 3 (a) Diagrama lógico del BCA de 4 fases (b) Comportamiento BCA de 4 fases.

1.2.2. Pipeline

La técnica de *Pipelining* superpone múltiples procesos en transferencia de datos, además, las señales de control se transfieren de una etapa a otra, controlando así distintos procesos en un circuito. Este proceso de propagación de las señales de control se asemeja a una FIFO, primero en entrar, primero en salir (del inglés *First In, First Out*). En un *Pipeline* asíncrono la latencia del circuito depende de la propagación de la señal a través del mismo, esto provoca que cada etapa opere a su propia velocidad y solamente se requiere de la transferencia de las señales de control de su etapa vecina, dicha transferencia se logra mediante el uso de elementos de retardo que garantice la correcta señalización y lectura de datos entre cada uno de los módulos que integran el circuito. A diferencia de en un circuito síncrono la latencia es igual al periodo del reloj multiplicado por el número de etapas del circuito.

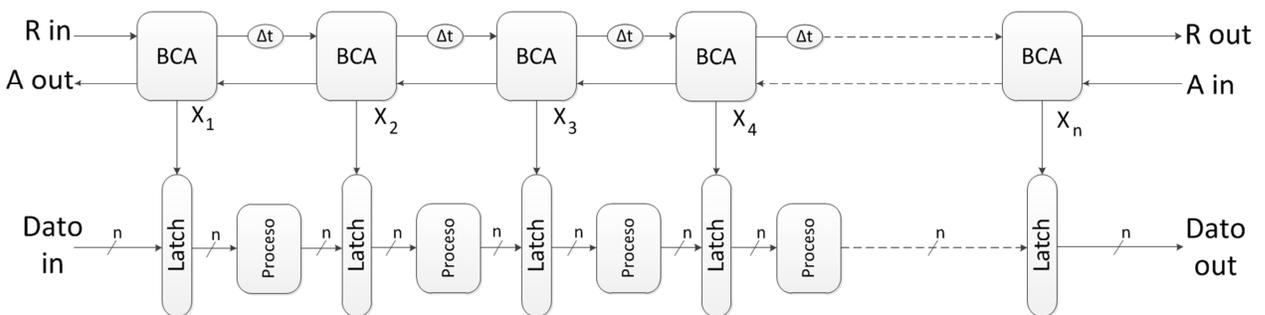


Fig. 4. Pipeline protocolo de riel simple de 4 fases.

Un *Pipeline* que utiliza comunicación asíncrona basada en BCAs se muestra en Fig. 4, se observa que las señales “X” activan los *Latches* con un tiempo de retardo dependiente de los bloques “ Δt ”, además se observa que la señal X_1 ingresa el dato a través de las etapas mientras que el X_n , entrega el dato procesado por el circuito.

1.3. Sensor de Visión Dinámico DVS

Este tipo de sensor es lo más parecido a una retina biológica, replicada de manera artificial, el cual no envía imágenes completas a una tasas de transferencia fijas, sino que solo se envían los cambios producidos en la escena por la variación de intensidad luminosa al momento que ocurren, a diferencia de los sensores de visión convencional que envían *frames* sucesivos con información que pudiese resultar redundante, lo cual conlleva a un mayor consumo tanto computacional como de potencia [8].

El DVS utilizado es el TmpDiff128 cuenta con una matriz de 128 x 128 pixeles, los cuales desarrollan la función de foto-receptores como los que internamente tiene la retina biológica, esta estructura está fabricada con transistores CMOS trabajando en la región de sub-umbral, con esto se obtiene un bajo consumo de potencia y la capacidad de emular la respuesta de la retina humana con un amplio rango dinámico. El resultado a la salida del DVS es un flujo de eventos basados en el protocolo AER, siendo éste el medio de comunicación entre el sensor y diferentes elementos externos [9, 10].

1.4. Representación de eventos a través de direcciones AER

AER (*Address Event Representation*) es un protocolo de comunicación entre circuitos, el cual emplea el multiplexado digital asíncrono, representando eventos basados en direcciones [11]. El funcionamiento de este protocolo asíncrono se observa en Fig. 5.

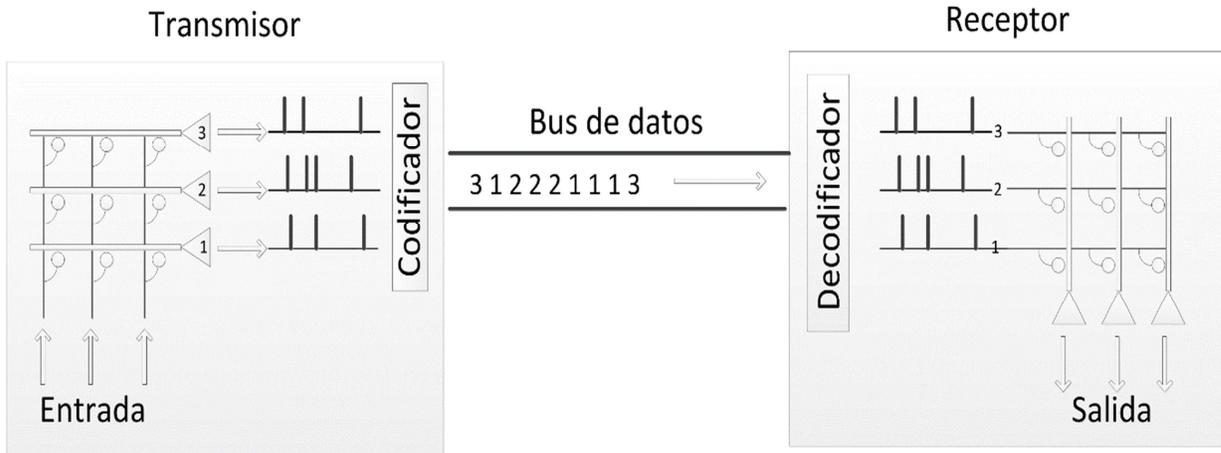


Fig. 5. Representación del protocolo AER.

Cada uno de los pixeles del chip TmpDiff128 tiene una dirección digital con un identificador único. Cuando ocurre un evento en un pixel, se transmite la dirección del pixel además de su polaridad (1,0), donde “1” indica un incremento en la iluminación y “0” un decremento, con respecto a un umbral dado. El orden de los bits del mensaje transmitido establece que el bit [0] representa la polaridad del evento, los bits [7:1] corresponden a la dirección en X, y los bits [14:8] corresponden a la dirección en Y y el bit [15] es un bit reservado para sincronización que es raramente utilizado, una de las aplicaciones de este bit es empleando dos DVS en visión estereoscópica.

2. Desarrollo

2.1. Elementos de retardo

Dentro del diseño de sistemas asíncronos, el elemento de retardo es uno de los que tienen mayor importancia, debido a que su desempeño permite a los otros bloques procesar el flujo de datos en el sistema. Este retardo debe ser calculado basado en el tiempo requerido para procesar los datos en algún punto dado del circuito. Los módulos fueron construidos en el software Xilinx ISE 14.6 utilizando macros y la tarjeta de desarrollo ML605 que incluye un FPGA XC5VSX50T [12], donde señales son forzadas

a pasar a través de una LUT del FPGA, como resultado la latencia es el tiempo requerido por las señales para pasar a través de los elementos internos (Fig. 6).

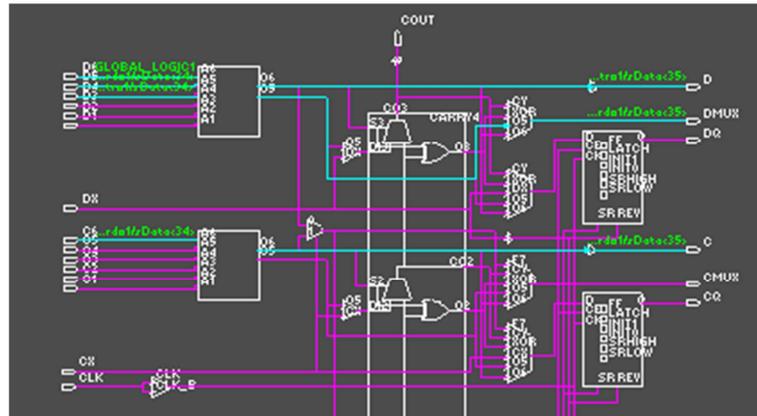


Fig. 6. Restricciones de usuario para generar elementos de retardo.

El tiempo de retardo de estos módulos se calcula como:

$$\Delta_{TOT} = T_{PI} + T_{LUT} + T_{PO} + T_{RUT}. \quad (1)$$

Donde:

- Δ_{TOT} es el retardo total.
- T_{PI} es la combinación de retardos entre la entrada y la salida del módulo T_{IOPI}
- T_{LUT} es la combinación de retardos entre entradas y salidas de las LUTs.
- T_{PO} es la combinación de retardo entre la entrada y salida del módulo T_{IOOP} .
- T_{RUT} es el tiempo de propagación existente de la conexión a otros módulos

2.2. Estructuras de adquisición de datos

El conjunto de estructuras propuestas es mostrado en Fig.7, las variables en estas estructuras son profundidad en los bloques de control asíncrono y número de elementos de retardo, Fig. 7a corresponde a cuatro bloques de control, Fig. 7b corresponde a ocho bloques de control, Fig. 7c corresponde a doce bloques de control y finalmente en Fig. 7d se muestra la estructura con dieciséis bloques de control, cada

una de ellas cuenta con cuatro cantidades diferentes de número de elementos de retardo, estos son; 10, 20, 30 y 40.

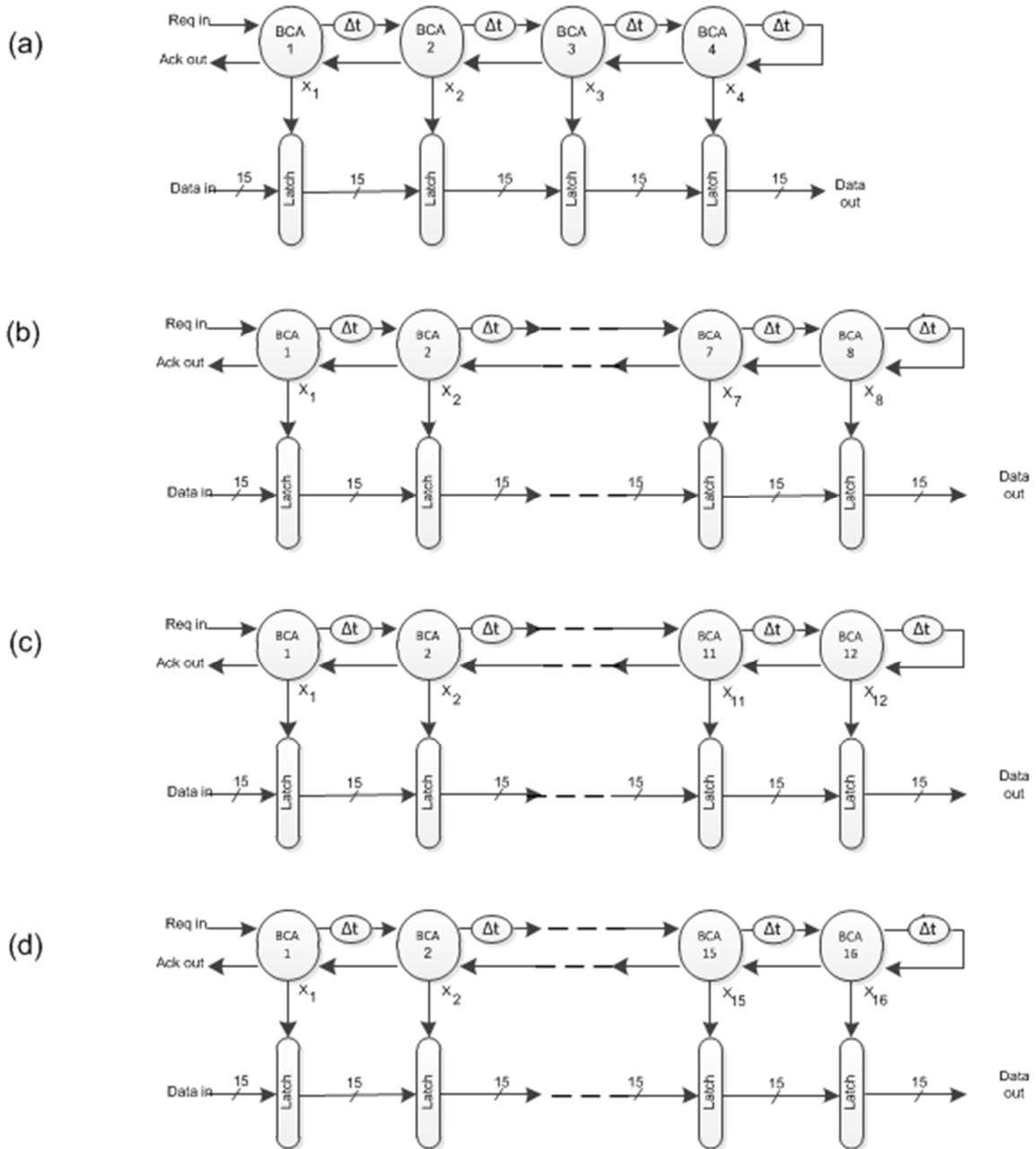


Fig. 7. Estructuras de pruebas diseñadas.

3. Resultados

3.1. Circuito generador de direcciones

La validación del circuito *self-timed* (ST) fue desarrollada mediante un circuito generador de direcciones, el cual emula el comportamiento del DVS usando un bus de datos de 15 bits y señales de control correspondientes al protocolo de riel simple de cuatro fases, su principal funciones proveer un flujo de datos al circuito *ST*, el cual genera direcciones correspondientes a una diagonal principal de una estructura de 128 x 128, las direcciones son enviadas y repetidas haciendo un ciclo en direcciones transmitidas, la frecuencia de operación de este generador es de 844.59 KHZ. El diagrama a bloques del circuito generador de direcciones se muestra en Fig. 8.

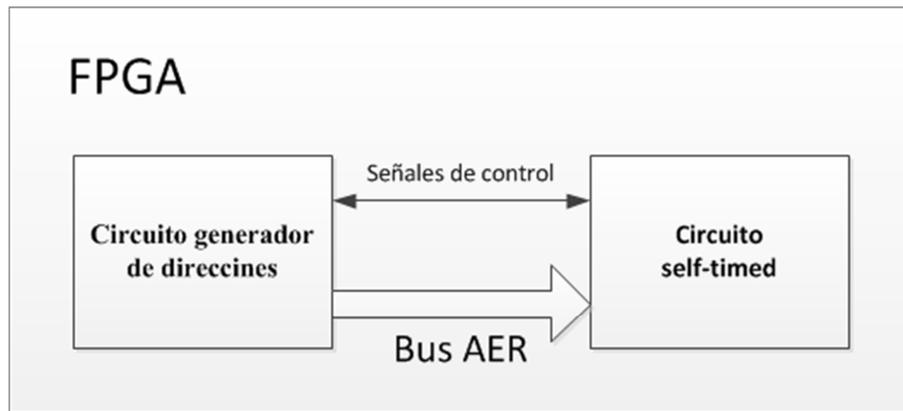


Fig. 8. Circuito generador de direcciones para pruebas con circuito *self-timed*.

El comportamiento del circuito generador de direcciones fue simulado utilizando las herramientas de Xilinx ISE 14.6. El resultado de la simulación (Fig.9) demuestra el correcto funcionamiento de este módulo, el cual maneja la señal de *request* y un bus de 15 bits, cambiando la dirección que será transmitida con cada petición.

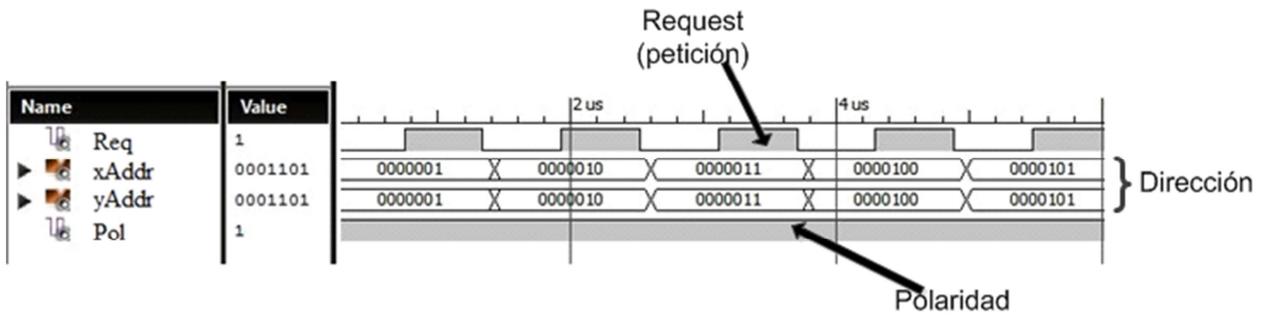


Fig. 9. Comportamiento en simulación del circuito generador de direcciones.

Al terminar la validación, el circuito generador de direcciones fue conectado al circuito ST para verificar el funcionamiento de ambos. La simulación (Fig. 10) presenta una correcta adquisición de datos por parte del circuito ST, donde la dirección que es enviada hacia el circuito ST (Dirección generador) fluye a través del circuito self-timed y éste presenta a su salida la misma dirección entrante (Dirección self-timed) con un retardo debido a la propagación de los datos por la estructura asíncrona, además el circuito ST genera la señal de *acknowledge* que servirá para responder a la señal de control del DVS, esta señal tiene un tiempo de respuesta determinado solo por la lógica del primer BCA, debido a esto, parece que se genera al mismo tiempo que la señal de *request* proveniente del circuito generador, pero este efecto es debido a la escala de medición.

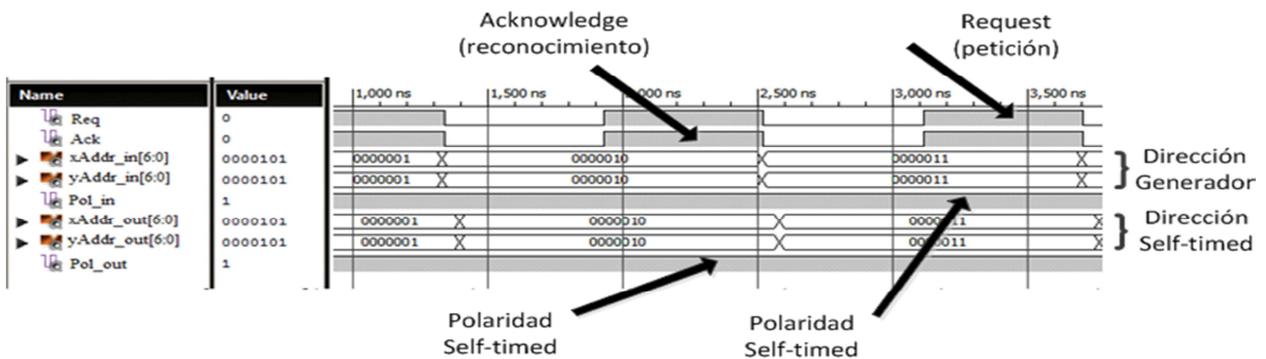


Fig. 10. Comportamiento en simulación del circuito generador de direcciones conectado al circuito self-timed.

La prueba en tiempo real del circuito generador de direcciones fue realizada utilizando un analizador lógico TLA5204B de Tektronix y es mostrada en Fig 11. El comportamiento en tiempo real resulto satisfactorio, debido a que su funcionamiento no presenta alteraciones en la generación de direcciones ni en la generación de la señal de control que servirán para estimular al circuito ST.

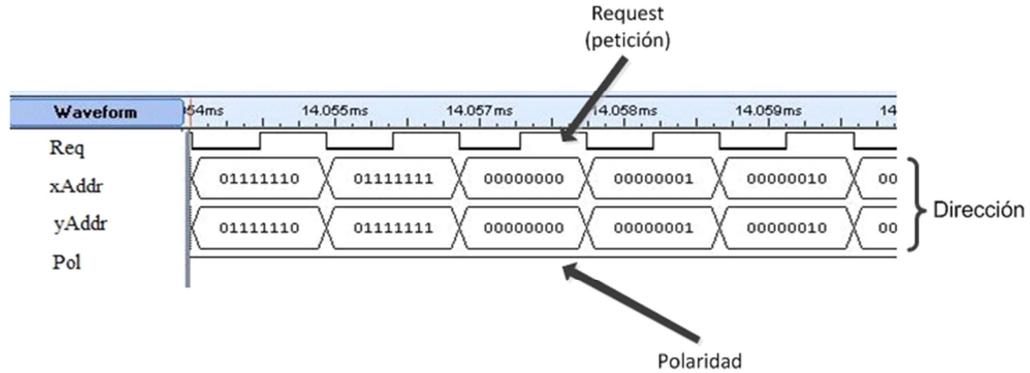


Fig. 11. Comportamiento en tiempo real del circuito generador de direcciones.

La prueba en tiempo real utilizando el circuito generador de direcciones conectado al circuito ST (Fig. 12), demuestra la capacidad del circuito asíncrono de manejar el flujo de datos utilizando la señal de control proveniente de una fuente externa como medio de señalización.

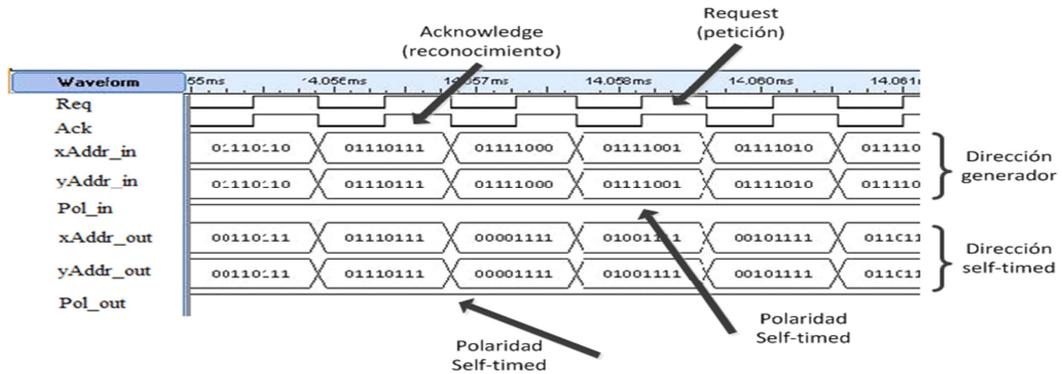


Fig. 12. Comportamiento en tiempo real del circuito generador de direcciones y self-timed.

El resultado de los datos de prueba, valida la funcionalidad de ambos circuitos; generador de direcciones y *ST*, donde muestra al generador de direcciones proporcionar un flujo de datos al circuito asíncrono, el cual tiene una generación de la señal de control y una captura de datos. Los resultados también completan la fase de prueba de la estructura de adquisición de datos.

3.2. Flujo de datos asíncronos provenientes del DVS

La cama de pruebas con la que se realizaron las mediciones en tiempo real se muestra en Fig. 13, donde se presenta la conexión de la cámara (Fig. 13c), la tarjeta de desarrollo (Fig. 13a) y el equipo necesario para la captura de información (Fig. 13b). El analizador lógico utilizado fue el encargado de generar una base de datos correspondientes al flujo proveniente del DVS así como el entregado por el circuito asíncrono.

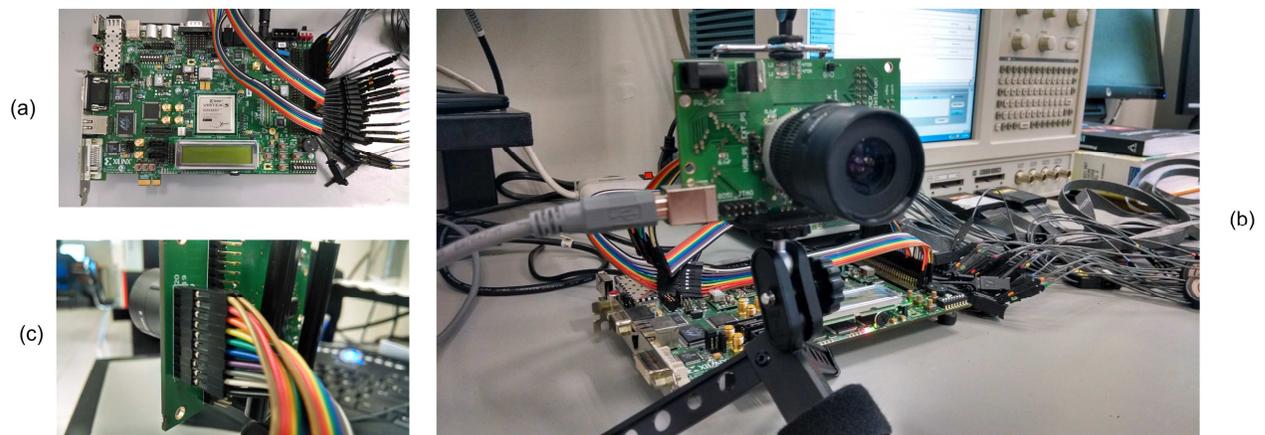


Fig. 13. (b) Conexión al equipo para captura de información (a) Tarjeta de desarrollo (c) Conexión de la cámara.

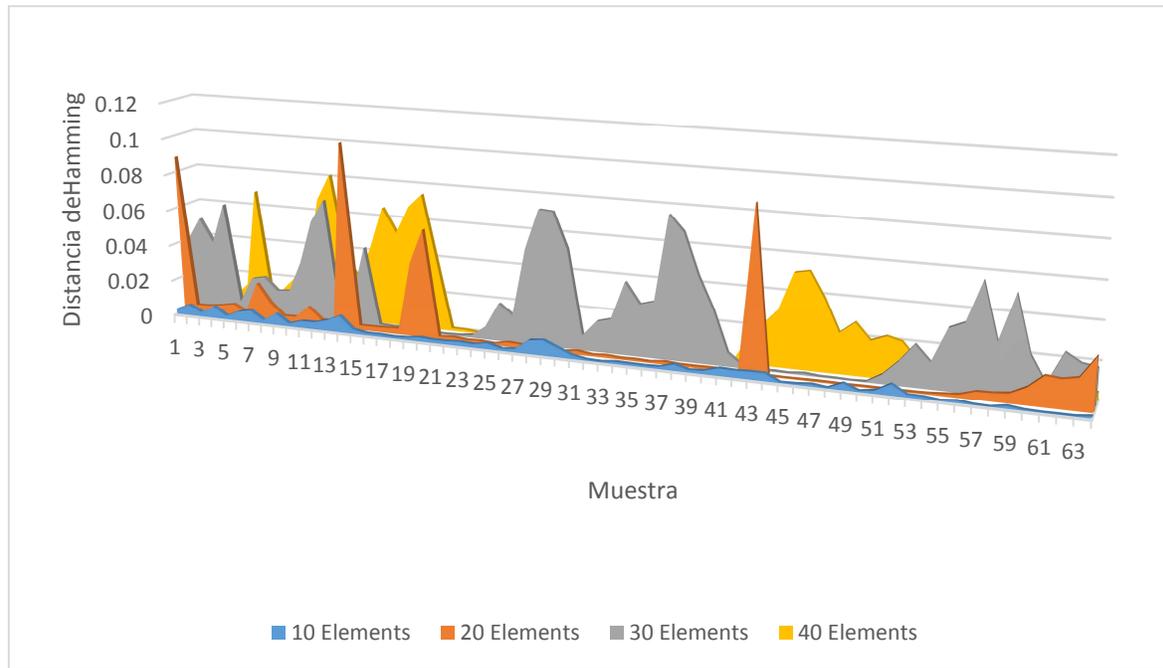
Los datos capturados en cada uno de los diseños fueron procesados con ayuda del software MATLAB, donde se realizó una comparación de la salida del DVS con respecto a la salida del circuito *ST*, con la finalidad de analizar que configuración tiene

una mejor respuesta (menor error), entre los datos generados por el DVS y el flujo de datos asíncronos a través de la estructura *Pipeline*.

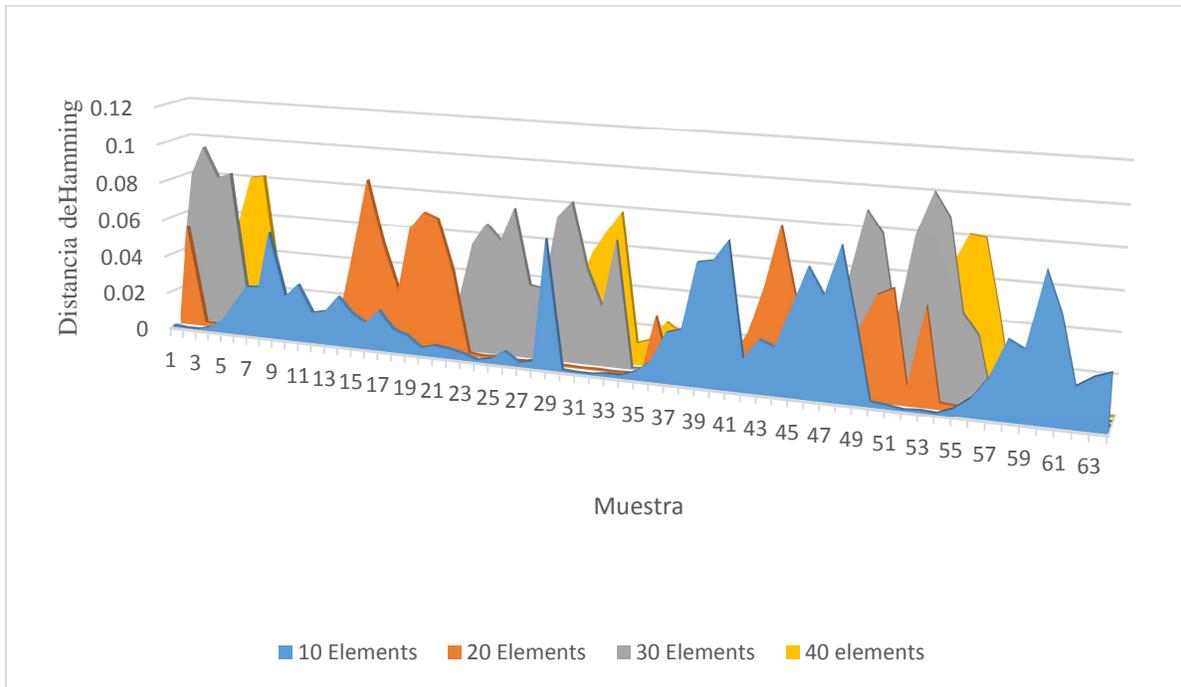
La comparación fue determinada por la Distancia de Hamming que corresponde a la ecuación (2).

$$H.D = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N x(i,j) \oplus y(i,j) \quad (2)$$

Para realizar el análisis se partió de una base de datos con dos millones de muestras, que se seccionó en 64 bloques, con fin de realizar cuadros de imagen y realizar la comparación (DVS vs ST). El resultado de esta comparación se presenta en la Fig. 14, donde se presentan las cuatro configuraciones de profundidad en bloques de control asíncrono y diferente cantidad de elementos de retardo.

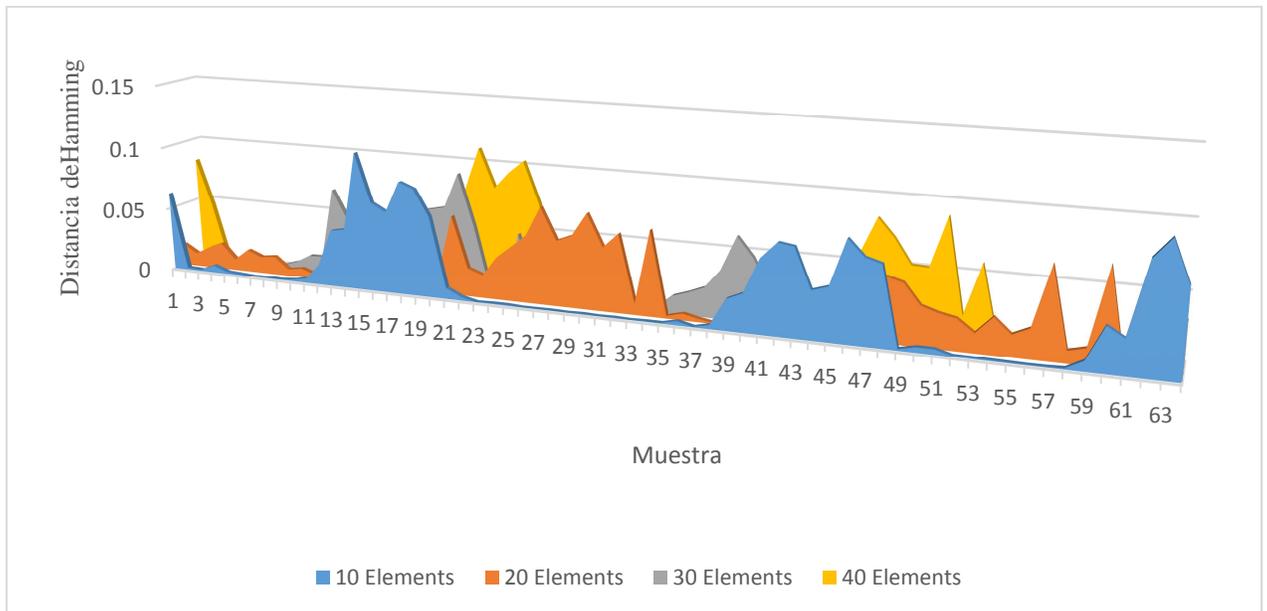


(a)

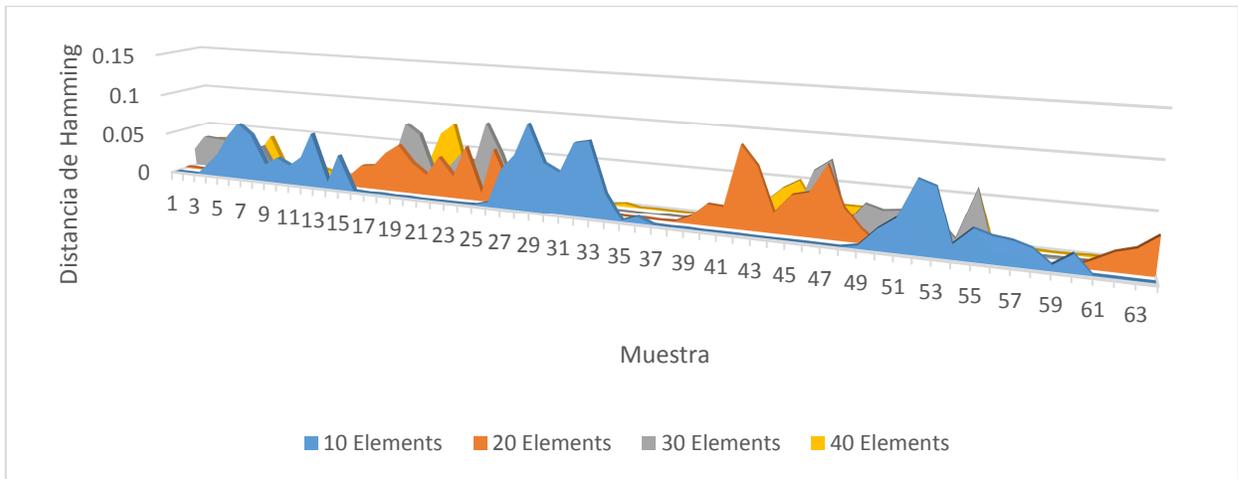


(b)

Fig. 14. (a) y (b) Relación de distancia de Hamming en las estructuras de adquisición de datos: (a) 4 BCAs, (b) 8 BCAs.



(c)



(d)

Fig. 14. (c) y (d) Relación de distancia de Hamming en las estructuras de adquisición de datos: (c) 12 BCAs y (d) 16 BCAs.

La comparación de las cuatro configuraciones se muestra en la Fig. 15, esta vez se llevó a cabo utilizando la información de la muestra completa, es decir, se realizó la comparación, uno a uno la salida de datos del DSV y salida de datos del circuito *ST* y se utilizó (2).

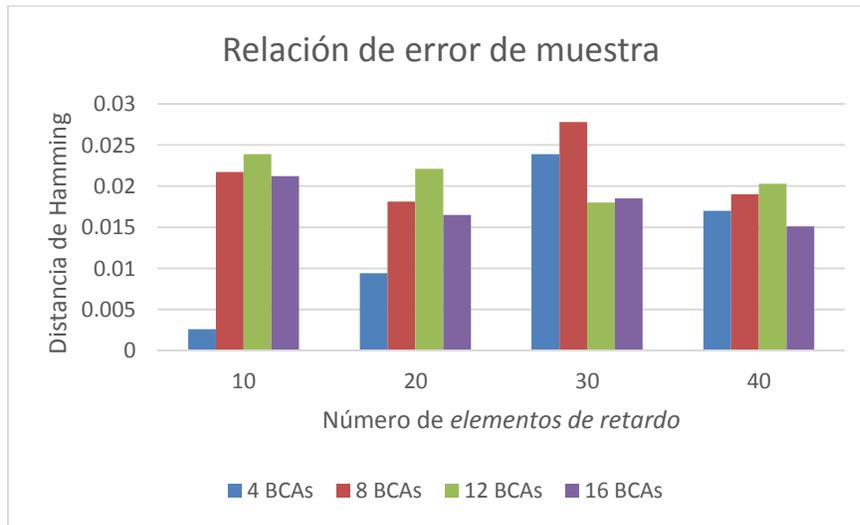


Fig. 15. Comparación de error con las cuatro configuraciones diseñadas.

4. Conclusiones

El protocolo AER emplea el flujo de datos asíncronos, lo cual requirió del diseño de elementos de retardo para trabajar con un circuito *self-timed*, éste circuito fue probado con un circuito generador de direcciones, con el fin de verificar su funcionamiento de forma independiente al DVS. En las pruebas en tiempo real con el sensor de visión dinámico, las estructuras diseñadas fueron capaces de realizar la captura de información generada por el chip TmpDiff128 y la relación de error obtenido se observan en Fig.14 y Fig.15. El comportamiento de las estructuras fue validado, y los resultados mostraron fiabilidad, además, se logró una buena adquisición del flujo de datos asíncronos utilizando el protocolo de riel simple de cuatro fases. Las características del DVS y el suministro de datos en una estructura asíncrona propician su empleo en áreas tales como medicina, biomedicina y militar, donde se requiere de un análisis óptico, por ejemplo, en la contabilidad de las células o partículas, velocimetría de seguimiento del flujo de partículas, análisis morfológico o la búsqueda de objetivos.

5. Referencias

- [1] H. Seo, J. Heo, T. Kim, "Clock skew optimization for maximizing time margin by utilizing flexible flip-flop timing". In Quality Electronic Design (ISQED), 16th International Symposium. IEEE. 2015. 35-39 pp.
- [2] Z. Xia, M. Hariyama, M. Kameyama, "Asynchronous Domino Logic Pipeline Design Based on Constructed Critical Data Path". Very Large Scale Integration (VLSI) Systems, IEEE. Vol. 23. No. 4. 2015. 619-630 pp.
- [3] P. A. Beerel, "Asynchronous circuits: an increasingly practical design solution". In Quality Electronic Design. Proceedings. International Symposium. 2002. 367-372 pp.

- [4] C. H. Van Berkel, M. B. Josephs, S. M. Nowick, "Applications of asynchronous circuits". Proceedings of the IEEE. Vol 87. No. 2. 1999. 223-233 pp.
- [5] S. Hauck, "Asynchronous design methodologies: An overview". Proceedings of the IEEE. Vol. 83. No. 1. 1995. 69-93 pp.
- [6] S. Ortega, J. Raygoza, E. Boemo, "Sincronización self-timed: protocolo de 4 fases". JCRA Jornadas de computación reconfigurable y aplicaciones. 2003. 517 – 528 pp.
- [7] J. Spars, S. Furber, "Principles Asynchronous Circuit Design". Kluwer Academic Publishers, 2002.
- [8] T. Delbruck, "Frame-free dynamic digital vision". In Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society. 2008. 21-26 pp.
- [9] P. Lichtsteiner, "An AER temporal contrast vision sensor" Doctoral dissertation, Diss., Naturwissenschaften, Eidgenössische Technische Hochschule ETH Zürich. No. 16898. 2006.
- [10] Overview of Dynamic Vision Sensors. www.inilabs.com/support/overview-of-dynamic-vision-sensors. Febrero 2015.
- [11] S. C. Liu, T. Delbruck, "Neuromorphic sensory systems". Current opinion in neurobiology. Vol. 20. No. 3. 2010. 288-295 pp.
- [12] Xilinx, Constraints guide. www.xilinx.com/itp/xilinx10/books/docs/cgd/cgd.pdf. Diciembre 2014.