

## Clúster de Computación Científica de Bajo Coste y Consumo

Mariano Hernández<sup>1</sup>, Alberto A. Del Barrio<sup>1</sup>, Guillermo Botella<sup>1</sup>

<sup>1</sup> Departamento de Arquitectura de Computadores y Automática, Facultad de Informática de la Universidad Complutense de Madrid  
Madrid, España

{marianoh, abarriog, gbotella}@ucm.es

**Resumen.** En este trabajo se presenta la construcción de un clúster basado en FPGAs de gama baja, capaz de ejecutar programas muy intensivos en datos en el mismo o en menor tiempo que una estación de trabajo con 56 cores lógicos, de mucho mayor coste y consumo. Se ha generado una imagen personalizada basada en Debian 8 y se ha instalado en ella el software necesario para poder ejecutar códigos escritos en OpenCL y compilados con el Kit de desarrollo de software de Intel para FPGAs. Además, se ha realizado una comparativa de los tiempos de ejecución, coste y consumo energético, cuyo resultado ha sido que el clúster es casi 6 veces más barato y es capaz de obtener reducciones energéticas del 83%.

**Palabras Clave:** FPGA, clúster, OpenCL, bajo consumo, bajo coste.

**Abstract.** In this work we present the construction of a cluster based on low-end FPGAs. Such cluster is able to execute data-intensive applications in a similar amount of time as a 56-threaded workstation, which is much more power-hungry and expensive. In order to perform this task, an in-house Debian 8-based image has been developed, and the required software to run OpenCL kernels compiled with Intel FPGA SDK for OpenCLv16.0 has been installed. Moreover, several comparisons tackling execution time, cost and energy consumption have been performed. As a result, we concluded that the cluster is almost 6 times cheaper than the workstation, and reaching 83% energy reductions.

**Keywords:** FPGA, cluster, OpenCL, low-power, low-cost.

### 1 Introducción

Las grandes cantidades de datos que se generan a diario y la necesidad de procesarlas han aumentado la demanda de la capacidad de cómputo de nuestros sistemas informáticos. Una de las soluciones más importantes en los últimos años ha sido el agrupamiento o clustering de sistemas informáticos [1-2, 9-12], conectándose entre sí

por medio de una red de altas prestaciones, con el fin de que se comporten como una única computadora. Estos clústeres hacen uso de la programación distribuida para ejecutar algoritmos en el menor tiempo posible (altas prestaciones o HPC) o para proporcionar un servicio continuo capaz de recuperarse ante los fallos (alta disponibilidad o HTC). El Máster de Ingeniería Informática de la Universidad Complutense de Madrid (UCM) [3-8] dispone de diversas asignaturas en las cuales se profundiza en estos contenidos, aunque el presente trabajo se presentó como Trabajo de Fin de Máster en el curso 2016-2017 [17-18].

Uno de los paradigmas de programación distribuida más importantes hoy en día es OpenCL [19], que permite paralelizar aplicaciones tanto en GPUs, multicores como FPGAs. Habitualmente en la literatura se encuentran trabajos que aceleran computaciones con OpenCL sobre una FPGA [10, 12] o por medio de un clúster de FPGAs, pero sin OpenCL [11, 20, 22]. Aquí nace la idea y motivación del proyecto: el desarrollo de un clúster de FPGAs que, ejecutando algoritmos implementados en C y OpenCL, sea capaz de procesarlos en un tiempo aceptable, con un consumo muy inferior al de un ordenador personal o una estación de trabajo.

Como elemento nodo de este clúster hemos decidido utilizar la Altera Cyclone V DE1-SoC, dado que tienen un coste de \$179 para universidades. Son apropiadas para este proyecto ya que son placas que cuentan con un procesador físico de doble núcleo (ARM Cortex-A9 Dual Core) capaz de ejecutar un Sistema Operativo (SO) Unix/Linux, y una FPGA de la familia Cyclone V. Otra ventaja muy importante, es que permiten ser programadas en OpenCL [19] utilizando un kit de desarrollo software (SDK) creado por Altera, compañía recientemente adquirida por Intel. No obstante, esto exige disponer de un SO compatible con OpenCL y que permita una rápida interconexión para transferir datos. Tras estudiar la literatura nos dimos cuenta de la carencia de tal SO, por lo que en este trabajo presentaremos una imagen basada en Debian-8 y compatible con OpenCLv16.0. A continuación, presentaremos también resultados de tiempo, consumo y coste tras ejecutar diversos programas tanto en el clúster propuesto como en una workstation de 56-cores lógicos. Los resultados muestran que el clúster es energéticamente más eficiente, llegando a consumir un 83% menos, y con un coste casi 6 veces inferior.

El resto del artículo se organiza de la siguiente forma: la Sección 2 describe los SOs analizados, mientras que la Sección 3 presenta la arquitectura de nuestra propuesta. La Sección 4 describe los benchmarks utilizados y finalmente las Secciones 5 y 6 muestran los resultados y las conclusiones del trabajo, respectivamente.

## 2 Estudio del Estado del Arte

Tal y como muestra la Figura 1, el SO se ejecutará en el ARM A9 de la DE1-SoC y recibirá el kernel OpenCL compilado por otra máquina. En esta sección se ha realizado un estudio sobre las imágenes de UNIX/Linux que existen actualmente para las placas DE1-SOC, con el fin de determinar la más adecuada para lograr el objetivo del proyecto. Estas imágenes deben cumplir dos requisitos indispensables:

- Deben tener una conexión de red rápida, que pueda funcionar sin necesidad de introducir la contraseña a través del protocolo SSH. Esto es necesario para enviar y

recibir la información a las placas de la forma más rápida posible y para evitar que, desde el nodo maestro, haya que escribir las credenciales de acceso para cada conexión.

- Deben tener instalado el software necesario para ejecutar aplicaciones escritas en C y OpenCL compiladas con la versión 16.0 del Intel FPGA SDK for OpenCL, que es la versión más reciente en la fecha que se empezó a escribir este Estado del arte y es la versión que se ha instalado en la estación de trabajo utilizada para compilar.

Linux Angstrom es una distribución de Linux basada en Debian creada especialmente para ser ejecutada en sistemas empotrados como las DE1-SoC. Es compatible con el proyecto Yocto, que es un proyecto de software libre que proporciona las plantillas, mecanismos y herramientas necesarias para crear distribuciones de Linux para sistemas empotrados. Esta imagen no cumple con ninguno de los requisitos que hemos impuesto. Por un lado, la red presenta problemas de velocidad de conexión que no pueden solucionarse al tratarse de una distribución personalizada y muy limitada en cuanto a configuración, lo que podría ser un problema en un futuro de cara a la instalación de software. Por otro lado, se presenta un problema relacionado con falta de librerías que impide ejecutar un programa compilado con el Intel FPGA SDK for OpenCL.

Por otro lado, se evaluaron cuatro imágenes de Linux BSP que se pueden descargar también desde la web de Altera:

- Linux Console. Se trata de una distribución basada en el proyecto Angstrom como la anterior, pero a diferencia de ella viene sin el módulo necesario para ejecutar OpenCL, aunque podría instalarse manualmente. No obstante, la conexión es lenta.
- Linux Console con framebuffer. Es igual que la anterior, solo que se ha utilizado la FPGA para que a través del puerto VGA de la placa se obtenga imagen. Esto posibilita conectar la placa a un monitor VGA y visualizar la terminal en él, pero tiene

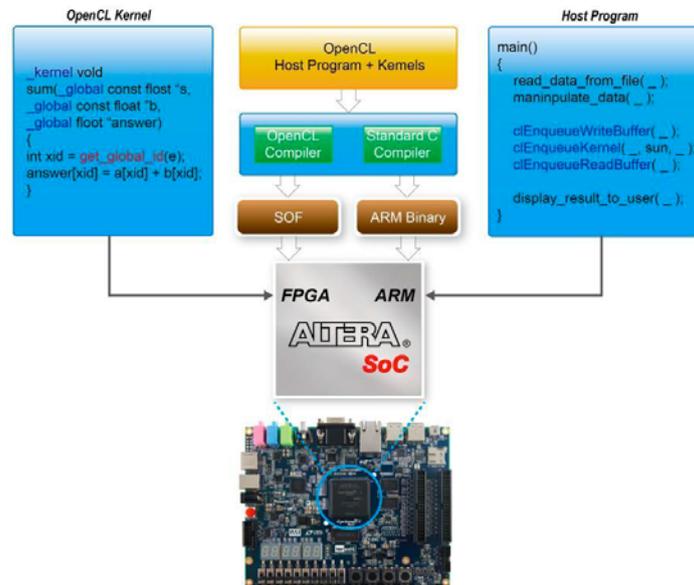


Figura 1. Integrando OpenCL y la DE1-SoC

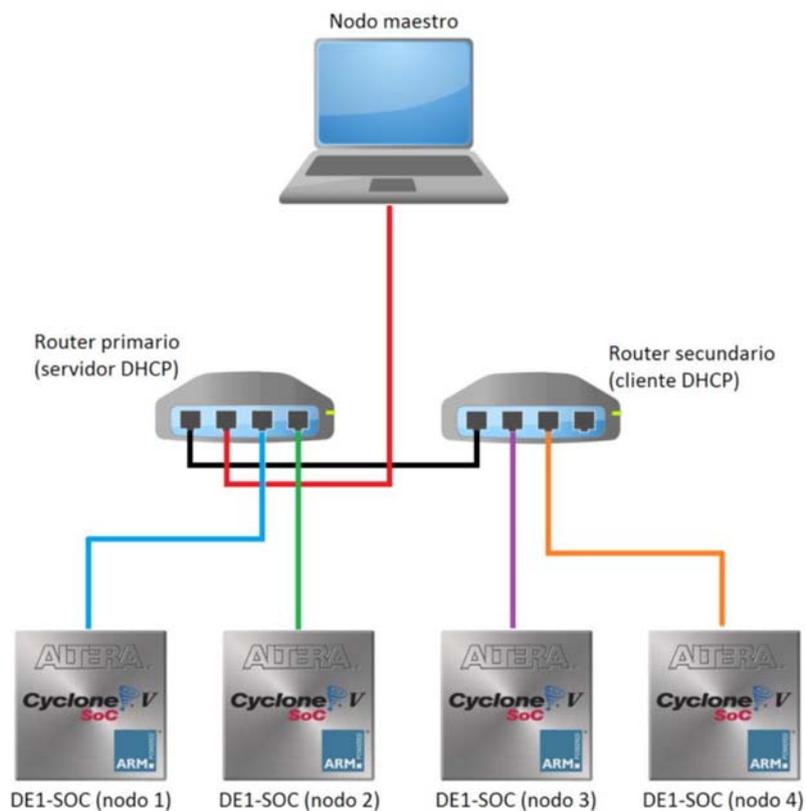
un gran inconveniente para nuestro proyecto: la FPGA queda inutilizada para ejecutar los algoritmos en OpenCL.

- Linux LXDE Desktop. Se trata de una distribución basada en LXDE con interfaz gráfico a través del puerto VGA. Como en el caso anterior, la FPGA se utiliza para comunicar la señal de video a través del puerto VGA y por lo tanto queda inutilizada para ejecutar OpenCL. No obstante, la red funciona sin problemas: la conexión es rápida y el intercambio de ficheros se hace a la máxima velocidad que permite la red.

- Linux Ubuntu Desktop. Igual que la anterior, pero basada en Ubuntu. La red también funciona sin problemas, pero la FPGA no puede utilizarse para acelerar los algoritmos.

Por último, se probó con la imagen de Linux incluida en el paquete Intel FPGA SDK for OpenCL. Se trata de la imagen de UNIX/Linux con base OpenBSD que viene incluida junto a la instalación del software de Intel necesario para realizar la compilación cruzada de los programas. Configuramos la red mediante el protocolo DHCP, pero la conexión por SSH no funcionó.

Dado que no encontramos una imagen que cumpliera todos los requisitos, decidimos desarrollar una propia basada en Debian 8. Los pasos detallados para hacer esto se encuentran en [17-18].



**Figura 2.** Arquitectura del sistema con 4 DE1-SoCs

### 3 Arquitectura del Sistema

La Figura 2 muestra la arquitectura del sistema con 4 DE1-SoCs, que es el máximo número de placas que teníamos disponibles. Para realizar la conexión utilizamos 2 enrutadores de 4 puertos Fast Ethernet. No se trata de routers de alto rendimiento, sino que son los típicos que proporcionan los Proveedores de servicios de Internet (ISP en su sigla en inglés). En nuestro caso son un Amper Xavi 7868r de Movistar y un TD5130 de ONO. Necesitamos conectarlos entre sí en la misma red local puesto que necesitamos que los nodos esclavos se puedan comunicar con el maestro. Esto hace un total de 5 conexiones (el nodo maestro más 4 nodos esclavos). Por ser más moderno, el TD5130 será el router maestro y, por lo tanto, será el servidor DHCP. Por otro lado, el Amper Xavi 7868r será el router secundario y será cliente de dicho servidor DHCP. Además, necesitamos cables Fast Ethernet y un computador funcionando como nodo maestro para enviar los kernels OpenCL así como los datos de entrada, y para recibir los datos de salida proporcionados por las DE1-SoCs.

Además de la configuración con 4 nodos, la configuración con 3 DE1-SoCs es particularmente interesante, ya que permite eliminar uno de los routers, disminuyendo el consumo final notablemente, como podrá verse en la Sección 5.

### 4 Benchmarks utilizados

En esta sección describiremos los programas utilizados para probar la eficiencia del sistema, que son los siguientes:

- El filtro de Sobel [21]. Es un filtro para detectar los bordes de una imagen.
- El filtro de Laplace [21]. Al igual que el anterior, es otro filtro para detectar los bordes de una imagen.
- El algoritmo Gaussian blur [21]. Se trata de una transformación para eliminar el ruido de una imagen, por ejemplo.
- El algoritmo de Lucas-Kanade [21]. Es un algoritmo de flujo óptico que detecta la variación de la posición de los elementos pertenecientes a una secuencia de imágenes (frames de un vídeo). Dicho algoritmo, toma 2 imágenes y genera una mostrando el movimiento de los píxeles ocurrido por medio de distintos colores.

Los 3 primeros programas se han probado paralelizando *en espacio*, es decir, dada una imagen de entrada, ésta se fragmenta y cada nodo es responsable de aplicar el kernel sobre el fragmento correspondiente. Para completar las pruebas se han utilizado diferentes resoluciones en las imágenes de entrada: Full HD, 2K, 4K, 5K and 8K [14-15].

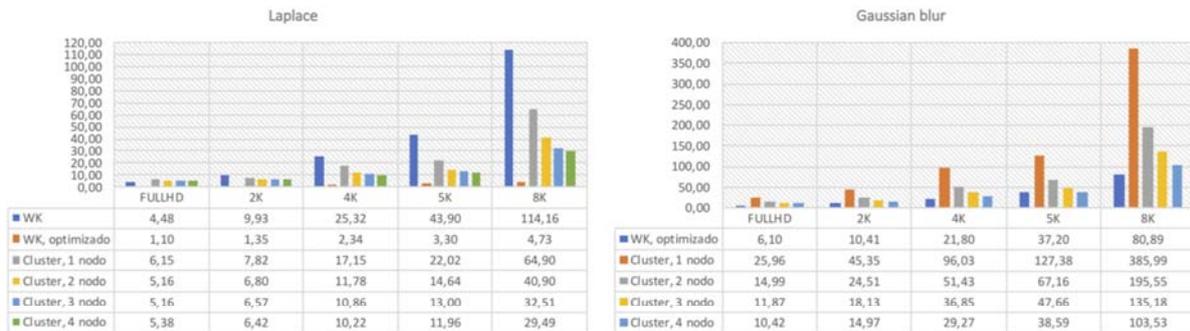
Por el contrario, el algoritmo de Lucas-Kanade se ha paralelizado *en el tiempo*, es decir, dada una secuencia de frames, ésta se divide y cada subconjunto es procesado por un nodo diferente. En este caso, se han probado conjuntos de 5, 10, 15, ... hasta 50 imágenes.

Los algoritmos necesarios para realizar dichas paralelizaciones y preparar los kernels OpenCL pueden encontrarse en [17-18].

Por último hay que mencionar que los kernels OpenCL se generan en una máquina distinta a las placas, lo que se conoce como compilación cruzada. Dado que la

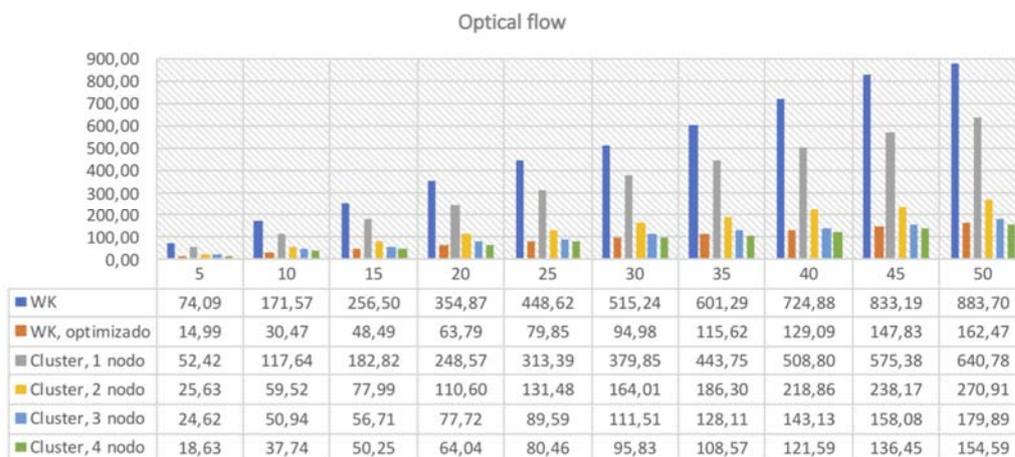
compilación de OpenCL requiere de un hardware con mucha memoria RAM y muchos accesos a disco, se ha decidido utilizar una workstation cuyo acceso ha sido facilitado por el Departamento de Arquitectura de Computadores y Automática (DACYA) de la UCM. Esta workstation cuenta con 32 cores, 32GB de RAM y un SSD de alto rendimiento de Intel. El software que ha sido necesario instalar en ella ha sido el siguiente:

- Quartus Prime Design Suite Release 16.0.
- Intel FPGA SDK para OpenCL versión 16.0.
- Altera SoC EDS.
- DE1-SOC OpenCL Board Support Package (BSP).



(a) Laplace

(b) Gaussian blur



(c) Lucas-Kanade

**Figura 3.** Tiempos de ejecución (s): workstation vs Clúster

**Tabla 1.** Potencia consumida (W) por la workstation [17]

Elemento	Potencia/unidad (W)	Cantidad	Potencia (W)
Intel Xeon E5	120	2	240
8GB DDR3 RAM	4,35	2	8,7
1 TB HDD	7,75	1	7,75
150 GB SSD	1,7	2	3,4
Placa base	62,5	1	62,5
Total			322,35

**Tabla 2.** Potencia consumida (W) y coste estimado (\$) del clúster [17]

Elemento	Precio/unidad (\$)	Potencia/unidad (W)	Cantidad	Coste (\$)	Potencia (W)
DE1-SoC	179	6,36	4	716	25,44
Router Xavi	46	15	1	46	15
Router ONO	46	12	1	46	12
Cables Ether.	6	0	5	30	0
Mem. Micro SD	16	0	4	64	0
Portátil	600	18	1	600	18
Total				1502	70,44

## 5 Experimentos

En esta sección presentaremos los resultados que demuestran la eficiencia del clúster construido. En primer lugar, tras compilar los kernels OpenCL como se mencionó en la Sección 4, estudiamos el tiempo de ejecución. En la Figura 3 se muestran los resultados para Laplace, Gaussian blur y Lucas-Kanade. Ha de notarse que los resultados de Sobel no aparecen por problemas de espacio, pero son similares a los de Laplace. Además, de los resultados del clúster con distintas configuraciones dependiendo del número de nodos, se muestran los resultados de ejecución en una workstation que cuenta con dos Intel Xeon E5-2695 v3, cada uno compuesto de 14 cores físicos, y con 2 threads/core, es decir, un total de 56 hilos o cores lógicos. Además tiene 16 GB de RAM y un disco SSD. Por último, en el caso de la workstation se muestran dos resultados: con OpenCL (WK) y con OpenCL y optimizado (WK, optimizado). Dicha optimización consiste en adaptar el kernel OpenCL al procesador de la workstation, utilizando para ello el número de grupos de trabajo dado por el compilador.

**Tabla 3.** Energía (kJ) consumida por la workstation (versión optimizada) y el clúster con 3 y 4 nodos [17]

		workstation	Cluster-3	Cluster-3 ratio	Cluster-4	Cluster-4 ratio
Laplace	FullHD	0.35	0.25	0.71	0.37	1.06
	2K	0.43	0.32	0.74	0.45	1.05
	4K	0.75	0.53	0.71	0.71	0.95
	5K	1.06	0.64	0.60	0.84	0.79
	8K	4.52	1.6	0.35	2.07	0.46
Sobel	FullHD	0.39	0.27	0.69	0.39	1.00
	2K	0.52	0.33	0.63	0.46	0.88
	4K	0.89	0.53	0.60	0.72	0.81
	5K	1.2	0.63	0.53	0.85	0.71
	8K	1.71	1.54	0.90	2.07	1.21
Gaussian blur	FullHD	1.96	0.58	0.30	0.73	0.37
	2K	3.35	0.89	0.27	1.05	0.31
	4K	7.02	1.81	0.26	2.06	0.29
	5K	11.99	2.34	0.20	2.71	0.23
	8K	26.07	6.63	0.25	7.29	0.28
Lucas-Kanade	25 frames	25.74	4.4	0.17	5.66	0.22
	50 frames	52.37	8.83	0.17	10.88	0.21

Como puede observarse, la versión optimizada sobre la workstation es la más rápida en general. Sin embargo, a medida que el clúster va aumentando el número de nodos, obtiene unos resultados bastante comparables e incluso ligeramente mejores, como en el caso de Lucas-Kanade. No obstante, la mayor ventaja proporcionada por el clúster es su bajo coste y su bajo consumo, como se verá en la siguiente subsección.

### 5.1 Coste y consumo

Las Tabla 1 muestra el consumo de la workstation de 56 cores lógicos, el cual es mayor de 322W. El coste estimado de acuerdo de dicha máquina, de acuerdo a [24], es de \$8482. Por otro lado, la Tabla 2 muestra el consumo de potencia y el coste en \$ del clúster propuesto en su configuración más completa, es decir, 4 DE1-SoCs. Como puede observarse, el consumo es de 70.4W y el precio total de \$1,502 (\$902 sin el portátil). Es decir, la workstation consume 4.57X más y cuesta 5.64X más (9.4X sin el portátil).

No obstante, para estudiar la eficiencia del sistema hay que tener en cuenta el consumo total de energía [13, 16, 23]. La Tabla 3 muestra el consumo de energía (kJ)

para el clúster configurado con 3 y 4 nodos, y para la workstation en su versión optimizada. En el caso de Laplace, Sobel y Gaussian blur se muestran los resultados para todas las resoluciones estudiadas, mientras que para Lucas-Kanade se muestran solo dos casos (25 y 50 frames). Las columnas Cluster-3 ratio y Cluster-4 ratio muestran la relación entre la energía consumida por el clúster con 3 y 4 nodos, respectivamente, y la energía consumida por la workstation. Por tanto, un valor por debajo de 1 significará que el clúster consume menos energía que la workstation.

Como puede observarse en la Tabla 3, casi todos los casos del clúster con 4 nodos y todos los casos del clúster con 3 nodos producen ahorros de energía, llegando al 83% en el mejor de los casos. A pesar de que el clúster con 4 nodos siempre es más rápido, la configuración con 3 nodos ha demostrado ser muy eficiente, ya que como comentamos anteriormente el segundo router no es necesario, bajando así el consumo total.

## 6 Conclusiones

En este artículo hemos presentado el resultado del Trabajo de Fin de Máster: un clúster de muy bajo coste y consumo, basado en FPGAs de gama baja, como la presente en la placa DE1-SoC, y OpenCL. Para conseguir la máxima eficiencia del sistema se ha desarrollado una versión propia de SO, basada en Debian 8 y compatible con la versión 16.0 del Intel FPGA SDK para OpenCL. Además, se han paralelizado diversas aplicaciones de procesamiento de imágenes para probar la velocidad y la eficiencia energética del sistema. Tal y como muestran los resultados, se han conseguido unos tiempos de ejecución cercanos a los de una workstation con 56 cores lógicos y casi 6 veces más cara que el clúster. A nivel energético, el clúster ha demostrado ser mucho más eficiente, logrando reducciones del 83%.

En el futuro, el sistema podría mejorarse escalándolo con más nodos y con una red más rápida, ya que el clúster propuesto estaba basado en Fast Ethernet y sería sencillo y no mucho más caro emplear Gigabit Ethernet, por ejemplo.

**Agradecimientos.** Nos gustaría agradecer al proyecto del MINECO con Fondos FEDER y número TIN2015-65277, así como al proyecto PR26-16/20B-1, financiado por la UCM y el Banco Santander.

## Referencias

1. Michael Feldman 2017. "Microsoft's Plan for FPGAs in Azure". <https://www.top500.org/news/microsofts-plans-for-fpgas-in-azure-should-worry-traditional-chipmakers/>. [Online; accedido 09-Abril-2018].
2. George Leopold 2016. "Intel's FPGAs Target Datacenters, Networking". <https://www.hpcwire.com/2016/10/06/intels-fpgas-target-datacenters-networking/>. [Online; accedido 09-Abril-2018].
3. Máster en Ingeniería Informática de la Universidad Complutense de Madrid, <http://informatica.ucm.es/estudios/2017-18/master-ingenieriainformatica>

4. D. Lora et al., “Sistema de Seguridad Basado en una Plataforma Heterogénea Distribuida“, *Enseñanza y Aprendizaje de Ingeniería de Computadores*, 5: 29-38 (2015).
5. F. Párrales et al. “Una Orquesta Sinfónica como Ejemplo de Aplicación de un Sistema Empotrado Distribuido”, *Enseñanza y Aprendizaje de Ingeniería de Computadores*, 5: 115-124 (2015).
6. I.M. Laclaustra et al. “Sistema Domótico Distribuido para Controlar el Riego y el Aire Acondicionado en el Hogar”, *Enseñanza y Aprendizaje de Ingeniería de Computadores*, 6: 87-102 (2016).
7. H. Ivanov et al. “Bomberman modo multijugador”, *Enseñanza y Aprendizaje de Ingeniería de Computadores*, 7: 53-68 (2017).
8. Jesús Martín Alonso et al., 2016. A distributed HW-SW platform for fireworks. Proceedings of the *Summer Computer Simulation Conference (SCSC '16)*. Society for Computer Simulation International, Montreal, artículo 17, 7 pages.
9. A. A. Del Barrio et al., "A Distributed Clustered Architecture to Tackle Delay Variations in Datapath Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 419-432, 2016.
10. Bai, X. et al. 2017. “Acceleration of RSA processes based on hybrid ARM-FPGA cluster”. *IEEE Symposium on Computers and Communications (ISCC)*, pp. 682–688.
11. Castillo, J. et al. 2009. “Hardware accelerated monte carlo financial simulation over low cost FPGA cluster”. *IEEE International Symposium on Parallel Distributed Processing*, pp. 1–8.
12. Domingo, R. et al. 2017. “High-level design using Intel FPGA OpenCL: A hyperspectral imaging spatial- spectral classifier”. *International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pp. 1–8.
13. A. A. Del Barrio and R. Hermida, "A slack-based approach to efficiently deploy radix 8 booth multipliers," *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, Lausanne, 2017, pp. 1153-1158.
14. Fernández, D. et al. 2018. “Complexity reduction in the HEVC/H265 standard based on smooth region classification”. *Digital Signal Processing* vol. 73, pp. 24 – 39.
15. Fernández, D. G. et al. 2018. “Fast and effective CU size decision based on spatial and temporal homogeneity detection”. *Multimedia Tools and Applications* vol. 77 (5), pp. 5907–5927.
16. A. A. D. Barrio et al. 2016, "A Partial Carry-Save On-the-Fly Correction Multispeculative Multiplier," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3251-3264.
17. Hernández, M. 2017. “Creación de un clúster de computación científica basado en FPGAs de bajo coste y consumo”. Master’s thesis, Universidad Complutense de Madrid, Spain. <http://eprints.ucm.es/43993/>.
18. Mariano Hernández 2017. “Debian-8 with OpenCL for DE1-SoC”. <https://github.com/mariano2AA3/tfm-de1soc-opencl-cluster>. [Online; accedido 20-Marzo-2018].
19. Khronos Group 2018. “OpenCL: The Open Standard for Parallel Programming of Heterogeneous Systems”. [www.khronos.org/opencl](http://www.khronos.org/opencl). [Online; accedido 06-Febrero-2018].
20. Mencer, O. et al. 2009. “Cube: A 512-FPGA cluster”. *Southern Conference on Programmable Logic (SPL)*, pp. 51–57.
21. Meyer-Baese, U. 2014. *Digital Signal Processing with Field Programmable Gate Arrays*. 4<sup>th</sup> ed. Springer.
22. Sass, R. et al. 2007. “Reconfigurable Computing Cluster (RCC) Project: Investigating the Feasibility of FPGA-Based Petascale Computing”. *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2007)*, pp. 127–140.
23. Kim, M. S. et al. 2018. “Low-power implementation of Mitchell’s approximate logarithmic multiplication for convolutional neural networks”. *Asia and South Pacific Design Automation Conference (ASP-DAC)*, pp. 617–622.

24. Outervision. "Power Calculator". <https://outervision.com/power-supply-calculator>. [Online; accedido 20-Marzo-2017].