

Arquitectura de un nodo sensor para aplicaciones de supervisión ambiental. Implementación de un prototipo reconfigurable.

Corti R., Martínez R., D'Agostino E., Belmonte J., Giandomenico E.

Departamento de Sistemas e Informática – Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Rosario, Argentina

e-mail: {rcorti, romamar, estelad, belmonte, giandome}@fceia.unr.edu.ar

Resumen—Las redes de sensores se utilizan para medir variables del medio ambiente, con el objetivo de realizar la supervisión y control de diversos entornos y actividades. Estas redes suelen trabajar bajo fuertes restricciones de energía y esto tiene un gran impacto en la definición del soporte hardware para los nodos que las componen. Los procesadores, por razones de costo y consumo, han sido la base de las plataformas utilizadas tradicionalmente, pero el surgimiento de aplicaciones complejas con altos requerimientos de velocidad, flexibilidad y capacidad de procesamiento ha propiciado la incorporación de dispositivos reconfigurables en el soporte hardware. Este trabajo presenta la arquitectura de un nodo sensor en el dominio de las aplicaciones de supervisión ambiental, con medición periódica de variables del medio. La misma se definió en base a los requerimientos de las aplicaciones elegidas y las características de un algoritmo de encaminamiento distribuido, particularmente adecuado para el dominio de trabajo. La arquitectura propuesta se ensayó en una plataforma reconfigurable cuya flexibilidad permitió explorar distintas opciones de implementación de los módulos constitutivos y seleccionar la más conveniente. Los ensayos realizados demostraron la adecuación de la arquitectura propuesta a la funcionalidad requerida.

Palabras clave: Redes de sensores; Plataformas reconfigurables; Partición HW/SW; FPGA.

I. INTRODUCCIÓN

Las redes inalámbricas de sensores inteligentes (RISI) se utilizan para medir variables del medio, con el objetivo de realizar la supervisión y control de entornos y actividades de la más diversa índole. Están constituidas por nodos que se auto organizan para adaptarse a topologías cambiantes y colaboran entre ellos para hacer llegar sus mediciones hasta la estación base. Las RISI se integran en aplicaciones industriales, médicas, agrícolas, de preservación del medio natural o creación de ambientes inteligentes, entre otras. En muchas de las aplicaciones mencionadas, la adquisición de las variables de interés debe realizarse en ambientes distantes u hostiles que hacen muy difícil el cableado y la atención periódica de los dispositivos de medición [1]. Estas redes suelen trabajar bajo fuertes restricciones de energía y esto tiene un gran impacto en la definición del soporte hardware (HW) para los nodos que las componen [2]. En este sentido, la mayoría de las plataformas

HW para RISI que se han presentado en la última década, se basan en procesadores pequeños de 8 o 16 bits que, además de ser baratos, permanecen en estado de bajo consumo cuando no están realizando las tareas que tienen asignadas en la red [3]. Sin embargo, al ampliarse el campo de aplicación de las RISI e incorporarse áreas de trabajo de complejidad creciente, se han incrementado las demandas de capacidad de cómputo incorporándose para estos casos procesadores de mayores prestaciones, que traen aparejado un aumento importante en el consumo [4].

Hoy en día se encuentran aplicaciones que agregan a los requerimientos de bajo consumo y capacidad de procesamiento la necesidad de plataformas flexibles y rápidas capaces de adaptarse a cambios en el ambiente de trabajo o, incluso, alterar su comportamiento en base al análisis de la información recolectada. Por estos motivos, muchos investigadores han comenzado a explorar otras opciones para las plataformas HW incorporando dispositivos de lógica reconfigurable que, en muchos casos, reparten sus responsabilidades con los procesadores que tradicionalmente han sido la base para la implementación de los nodos sensores [5].

Este trabajo presenta la arquitectura de un nodo para una RISI en el dominio de las aplicaciones de supervisión ambiental, con medición periódica de variables del medio. La misma se definió en base a los requerimientos de las aplicaciones elegidas y las características de un algoritmo de encaminamiento distribuido, particularmente adecuado para el dominio de trabajo. La arquitectura propuesta se ensayó en una plataforma reconfigurable basada en una FPGA (Field Programmable Gate Array), lo que permitió explorar distintas opciones de implementación de los módulos constitutivos y seleccionar la más conveniente.

El resto de la publicación se organiza de la siguiente forma, en la sección II se enumeran las ventajas e inconvenientes de las plataformas reconfigurables para RISI, la sección III fundamenta las características de la arquitectura de nodo sensor propuesta y la sección IV analiza las opciones exploradas para implementar el módulo SW de la misma. En la sección V se describen las pruebas realizadas con el prototipo reconfigurable

implementado y, finalmente, en la sección IV se obtienen conclusiones y se trazan líneas de trabajo futuro.

II. PLATAFORMAS RECONFIGURABLES PARA RISI

Los dispositivos de lógica reconfigurable poseen varias características que los vuelven muy atractivos para algunas aplicaciones de redes de sensores. Las mismas se refieren a su gran flexibilidad, alta velocidad de respuesta y elevada capacidad de procesamiento.

En la literatura se encuentra, por los motivos antes expuestos, un interesante número de propuestas ligadas a la incorporación de dispositivos reconfigurables en las plataformas HW para nodos sensores, entre las cuales se pueden mencionar los trabajos descriptos en [6][5][3].

La mayoría de los investigadores coinciden en que, pese a los avances de la tecnología que ha permitido desarrollar dispositivos con mejores prestaciones, las FPGA aún presentan el problema de mayor costo y consumo respecto de los microcontroladores, lo cual es un inconveniente importante para su utilización en redes de sensores. Sin embargo, considerando las ventajas que proporcionan en flexibilidad, velocidad y capacidad de procesamiento, varios de ellos apuestan al avance tecnológico y opinan que en un futuro cercano la utilización de estos dispositivos será más habitual en las plataformas HW para RISI [7][8][3].

En este trabajo se decidió utilizar una plataforma reconfigurable basada en una FPGA para los ensayos ligados a las pruebas de funcionamiento de la arquitectura de nodo sensor desarrollada. Esta decisión se fundamentó en la flexibilidad brindada por estos dispositivos para poder explorar distintas opciones de implementación de los módulos constituyentes sin necesidad de realizar cambios en la placa de trabajo.

III. ARQUITECTURA PARA EL NODO SENSOR

El nodo de una red de sensores está constituido por cinco módulos principales, esquematizados en la Fig. 1, a saber: adquisición de datos, adecuación de señal, procesamiento, comunicaciones y potencia.

La definición de los módulos de la arquitectura propuesta se fundamentó en los requerimientos del dominio de trabajo y en las características de un algoritmo de encaminamiento adecuado para el mismo [9][10].

- Se debe realizar la medición periódica de variables en toda el área bajo estudio. Los datos se recolectan, procesan y envían hacia la estación base una vez que se cumple un período T del orden de 10 a 15 minutos.
- El retardo en la llegada de la información a la estación base no es relevante, pero se deben diferenciar las mediciones correspondientes a distintas rondas de recolección.
- La red debe ser capaz de operar sin atención durante varios meses, manteniendo una calidad de servicio (QoS) definida como un porcentaje máximo admisible de pérdida de mediciones en cada ronda.

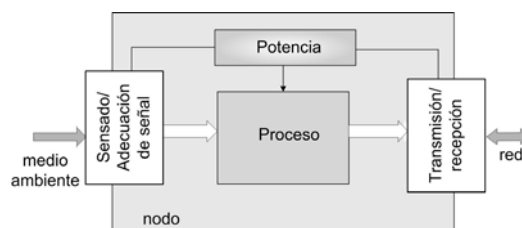


Fig. 1. Esquema de nodo sensor

El algoritmo de encaminamiento es distribuido y divide su funcionamiento en fases de trabajo en las cuales los nodos, homogéneos en recursos, asumen distintos roles. Esta característica tiene el objetivo de plasmar estrategias orientadas a disminuir el consumo y balancear la carga de trabajo entre los miembros de la red, colaborando en el logro de autonomía de funcionamiento.

- El nodo debe ser capaz de definir distintos tipos de mensajes para comunicarse con sus vecinos y de aplicar técnicas de agregación de datos a las mediciones colectadas. Esto se fundamenta en que las variables estudiadas están altamente correlacionadas espacial y temporalmente, por lo que la agregación de datos permite lograr un envío de información más eficiente y disminuye la congestión en la red.

El módulo de la unidad de proceso se ocupa de realizar las tareas inherentes al tratamiento local de la información y a la implementación del algoritmo de encaminamiento, definiendo los mensajes de datos y control asociados que luego transfiere al módulo de comunicaciones. Por lo tanto, considerando que se trabaja con un algoritmo distribuido que basa las decisiones de encaminamiento en información local, y que se aplicarán técnicas de agregación de datos, sus responsabilidades involucran una carga importante de trabajo. Esta funcionalidad podría haberse implementarse utilizando una máquina de estados finitos (MEF), pero la misma debería contar con un número importante de estados y una lógica de evolución compleja, conduciendo a un diseño y prueba dificultosos. Si el número de estados necesarios en una MEF resulta elevado y los requerimientos de tiempo de respuesta no son muy exigentes como en este caso, es mucho más conveniente asignar la tarea a un módulo software. Este fue el enfoque adoptado, y se asignaron a un procesador las decisiones ligadas al encaminamiento, tratamiento y envío de información, responsabilizándolo del procesamiento local de los datos y de la definición de los mensajes necesarios.

La medición periódica de variables en toda el área bajo estudio implica periodos donde todos los nodos de la red envían mediciones a la estación base y otros, prolongados para nuestro caso, donde no realizan tarea alguna. Por lo tanto, es importante colocar en modo de bajo consumo todos los bloques posibles del nodo, con el objetivo de ahorrar energía y cumplir con el requerimiento de autonomía de funcionamiento. En este sentido se definió un módulo de encendido, que permanece siempre activo y asume la responsabilidad de indicar al resto de los bloques del nodo el momento en el cual deben iniciar sus tareas.

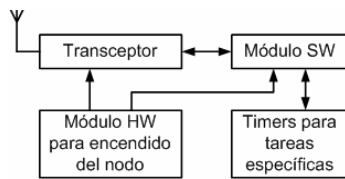


Fig. 2. Unidades de comunicación y procesamiento

El algoritmo de encaminamiento divide su funcionamiento en fases con tareas diferenciadas que deben ser estrictamente definidas. Para lograrlo, se definieron un conjunto de temporizadores encargados de determinar los momentos de toma de decisiones y cumplimiento de responsabilidades. Los mismos se implementaron como bloques HW, capaces de interrumpir al procesador para indicarle que ha llegado el momento de llevar adelante acciones específicas.

En resumen, las unidades de comunicaciones y procesamiento correspondientes a la arquitectura propuesta para el nodo sensor, están constituidas por los módulos de alto nivel que se muestran en la Fig. 2.

IV. EL MÓDULO SOFTWARE

Los altísimos niveles de integración existentes, y la heterogeneidad de las arquitecturas de los dispositivos reconfigurables actuales, convierten a las FPGA en integrados dotados de grandes capacidades de cómputo [11]. Actualmente existen numerosos núcleos software de procesadores que pueden ser embebidos en dichos dispositivos. Entre ellos se pueden mencionar, PicoBlaze de libre distribución [12], Nios II de la empresa Altera [13], y MicroBlaze de Xilinx [14].

En este sentido, se observa un incremento significativo del uso de dispositivos reconfigurables con núcleos de procesadores embebidos, que en opinión de varios autores parece consolidarse como tendencia firme para el diseño electrónico [15]. En particular, las plataformas FPGA y los ambientes de desarrollo asociados, resultan muy convenientes para llevar adelante la tarea de partir e interconectar los módulos asignados a funciones hardware y software dentro de la misma pastilla [16].

La implementación del módulo SW de la unidad de proceso se realizó con el enfoque antes mencionado, incorporándose núcleos software de procesadores de distintas características a la arquitectura del nodo sensor. Los ensayos realizados fueron incrementales, se comenzó con prototipos cableados basados en un procesador muy sencillo, y se continuó con un prototipo inalámbrico de mayores prestaciones, capaz de soportar la arquitectura definida en la sección III.

A. Implementación con un procesador de 8 bits

La primera experiencia realizada sobre el módulo SW, tuvo como objetivo explorar el nivel de complejidad asociado con la implementación de las responsabilidades asignadas al bloque. Se trabajó con el microprocesador Picoblaze, un núcleo

software de propiedad intelectual (IP) optimizado para la familia Spartan 3 de Xilinx [12]. Se trata de un procesador RISC de arquitectura muy simple que, junto con un bloque RAM que contiene el programa de 1Kbyte como máximo, se embebe en la FPGA de la placa Spartan-3 Starter Kit Board [17]. Esta disposición permitió conectar el procesador con otros bloques HW de la arquitectura.

Es importante destacar que esta prueba preliminar, no incluyó el módulo HW de encendido ni los temporizadores para tareas específicas necesarios para la arquitectura general del nodo antes descripta. El trabajo se centró en el análisis de la opción de implementación del módulo SW con el procesador PicoBlaze, y su interconexión con los periféricos necesarios para su correcto funcionamiento. Por lo tanto, se implementaron dos nodos, uno encargado de enviar información y otro de recibirla y procesarla. La transmisión en radiofrecuencia, fue reemplazada por una comunicación serie con dos hilos, incorporándose al diseño el núcleo IP de una UART (Universal Asynchronous Receiver-Transmitter) de libre distribución [18], bajo el estándar RS-232, compatible con el puerto serie de la placa utilizada.

Se programó el microprocesador en código ensamblador, y se instanciaron, utilizando el ambiente ISE de Xilinx [19], los núcleos del PicoBlaze, las UARTs y los módulos en VHDL diseñados para lograr la integración de todos los componentes en el sistema final. Si bien la implementación resultó viable y las pruebas satisfactorias, se hicieron evidentes algunas dificultades y limitaciones. El procesador PicoBlaze utilizado, posee una arquitectura muy sencilla que demanda pocos recursos de pastilla, pero hace necesario incorporar al diseño los periféricos requeridos para su funcionamiento. Estos dispositivos basados en núcleos IP deben ser adaptados y conectados en el sistema, tarea que puede resultar ardua si la complejidad de la plataforma hardware es grande. Además, la programación del procesador se realiza en ensamblador, lenguaje de bajo nivel, que demanda un esfuerzo considerable si se pretende implementar el comportamiento completo del nodo.

Teniendo en cuenta las consideraciones mencionadas, se decidió migrar el diseño al núcleo IP de un procesador con mayores prestaciones que permite trabajar en un lenguaje de alto nivel y dispone de un ambiente de desarrollo que facilita la conexión de periféricos.

B. Implementación con un procesador de 32 bits

El procesador seleccionado fue el núcleo de software MicroBlaze de 32 bits de la empresa Xilinx, que cuenta con un ambiente de desarrollo integrado para personalizar la plataforma hardware incorporando distintos tipos de periféricos, que se disponen en bibliotecas con sus correspondientes drivers, lo que reduce significativamente la cantidad de módulos necesarios para adaptación e interface [20]. La programación software puede realizarse en un lenguaje de alto nivel como C, lo que facilita en gran medida la codificación de algoritmos complejos [14].



Fig. 3. Ensayo del prototipo implementado

Estas características son muy ventajosas, ya que dependiendo de los requerimientos de cada aplicación, puede decidirse el número y tipo de periféricos a interconectar utilizándose sólo los recursos de pastilla necesarios.

Las características de MicroBlaze y su ambiente de desarrollo permitieron incorporar desde biblioteca, el módulo controlador de interrupciones, la UART, los temporizadores y los módulos GPIO (General Purpose Input/Output) para comunicarse con los recursos de la placa, que se utilizaron para ingresar la información de identificación del nodo y mostrar resultados del procesamiento local de mensajes. Además, se incorporó al prototipo un módulo de comunicación inalámbrica que reemplazó a la conexión de dos hilos implementada en las pruebas preliminares. El módulo de comunicación que se conectó con la placa Spartan 3 Starter Kit Board que implementa los módulos antes mencionados, fue el de la empresa Digi Internacional que soporta el estándar IEEE 802.15.4 elegido para las capas inferiores del protocolo de comunicaciones.

El correcto funcionamiento de la arquitectura propuesta se verificó mediante la implementación y ensayo de dos nodos idénticos, como el mostrado en la Fig. 3. El funcionamiento de los dispositivos se rigió por el algoritmo de encaminamiento

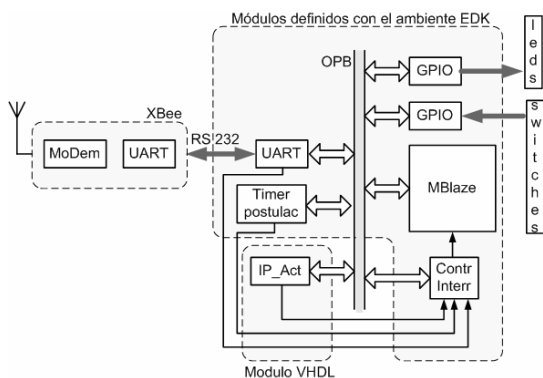


Fig. 4. Esquema de la arquitectura del nodo con MicroBlaze

TABLA I. RECURSOS DE PASTILLA UTILIZADOS

Recursos	PicoBlaze		Microblaze	
<i>Slice FF</i>	149 out of 3,840	3%	1,905 out of 3,840	49%
<i>4 input LUTs</i>	201 out of 3,840	5%	2,145 out of 3,840	55%
<i>Block RAMs</i>	1 out of 12	8%	8 out of 12	66%

mencionado en la sección II. La arquitectura implementada en cada nodo quedó constituida por los módulos mostrados en la Fig. 4. En la misma, además del procesador, pueden observarse varios periféricos personalizados en el ambiente EDK y también el bloque de activación del nodo conectados a través del bus OPB (On-chip Peripheral Bus) de la arquitectura CoreConnect de IBM [21].

- **Timer postulac:** Este módulo se ocupa de lanzar una rutina asociada con la fase de armado del árbol de encaminamiento mediante una interrupción al procesador
- **Módulos GPIO:** Implementados con IP disponibles en bibliotecas para conectar los recursos de la placa (llaves y diodos led) con el procesador embebido.
- **Módulo UART:** Implementado con un IP de biblioteca (UartLite) que se comunica vía RS-232 con la UART del módulo XBee, que implementa el bloque tranceptor.
- **IP_Act:** Es un módulo HW desarrollado en VHDL, que se ocupa de despertar en el momento apropiado los bloques que intervendrán en el procesamiento de la información y el intercambio de mensajes en la red (procesador y tranceptor).
- **Controlador de Interrupciones:** Implementado con un IP de biblioteca personalizado para recibir tres pedidos de interrupción (IP_Act, Timer postulac y UART).

Se debe puntualizar que durante las pruebas realizadas todos los módulos permanecen activos, y que el encendido de los bloques se simula en el prototipo mediante interrupciones al procesador, que lanzan rutinas en los momentos precisos. La implementación efectiva del apagado y encendido de todos los bloques software y hardware posibles queda pendiente para una implementación futura.

Los recursos de pastilla utilizados en la implementación del diseño con MicroBlaze resultaron, como era de esperar, superiores a los demandados por la opción basada en PicoBlaze, según puede verse en la Tabla I que presenta los resultados comparativos. Es importante destacar que el prototipo no constituye una implementación definitiva de la arquitectura propuesta. El objetivo fue explorar distintas opciones para el módulo software, verificando que la arquitectura propuesta es adecuada para soportar la funcionalidad requerida.

V. RESULTADOS OBTENIDOS

Para realizar las correcciones necesarias y verificar el funcionamiento del prototipo se utilizó el analizador lógico

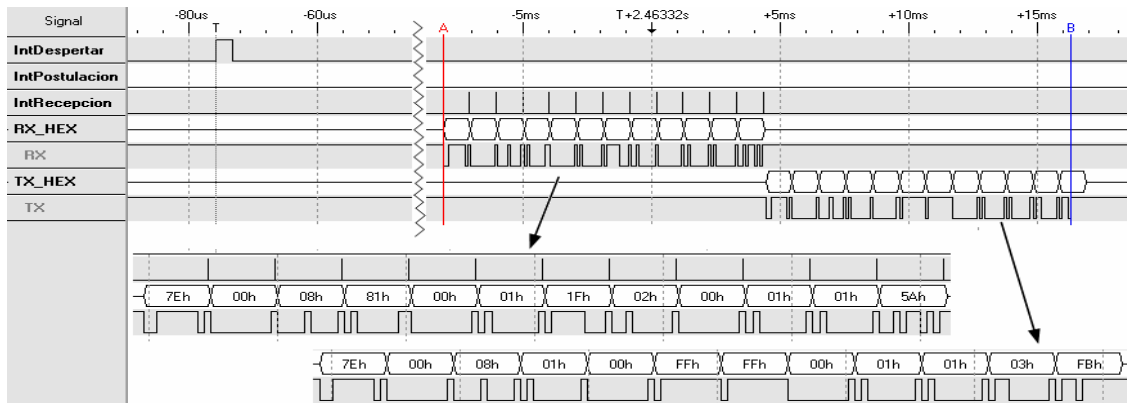


Fig. 5. Visualización de señales: Recepción de la postulación del Nodo 1 por parte del Nodo 3

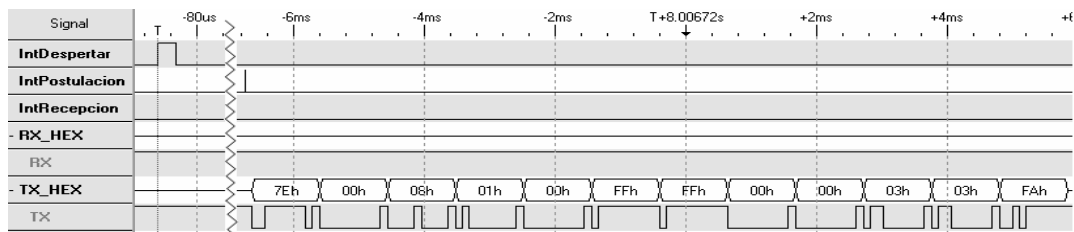


Fig. 6. Visualización de señales: Postulación del Nodo 3

LogicPort [22] para visualizar señales generadas a partir del intercambio de mensajes entre los dos nodos implementados. En el ensayo realizado se inicializó el sistema ingresando mediante las llaves de las placas el identificador de cada nodo y lanzando el módulo IP_Act. Luego de la inicialización, los nodos intercambiaron los mensajes correspondientes a situaciones ligadas a la definición del árbol de encaminamiento necesario para transmitir las mediciones hasta la estación base.

La Fig. 5 muestra la interrupción generada por el módulo IP_Act del nodo 3, que simula el encendido del procesador por parte del módulo de activación. A continuación, el nodo 3 recibe un mensaje del nodo 1, que indica que se postula como coordinador de un cluster en la estructura de encaminamiento. Como respuesta, el nodo 3 envía un mensaje anunciando que adopta al nodo 1 como su enlace en el cluster para transmitirle sus mediciones.

En la Fig. 5 también pueden verse las interrupciones producidas por el módulo UART del nodo cada vez que ingresa un byte enviado por el nodo 1 (IntRecepción). Los mensajes intercambiados son transmitidos por los módulos en su formato API (Application Programming Interface), y se muestran en las líneas RX y TX.

La figura Fig. 6 muestra otra situación en el nodo 3. La interrupción del módulo IP_Act despierta el nodo, que en este caso no recibe ningún mensaje de su vecino. Por lo tanto, una vez cumplidos los tiempos, se produce la interrupción generada por el temporizador de postulación (IntPostulación) y el nodo

3 envía (TX) un mensaje con el cual se propone como coordinador de un cluster de la estructura de encaminamiento.

Las pruebas realizadas se enfocaron en el buen funcionamiento del prototipo en las situaciones planteadas, sin establecer resultados respecto del consumo de energía de los nodos. La plataforma reconfigurable utilizada no constituye una implementación definitiva y está basada en dispositivos FPGA, muy adecuados para los objetivos planteados, pero que la hacen poco conveniente para un estudio del consumo asociado con su funcionalidad.

VI. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo presenta la arquitectura de un nodo para una red de sensores en el dominio de las aplicaciones de supervisión ambiental, con medición periódica de variables del medio. La misma se definió en base a los requerimientos de las aplicaciones elegidas y las características de un algoritmo de encaminamiento distribuido, particularmente adecuado para el dominio de trabajo. Para implementar un prototipo capaz de soportar la arquitectura definida y realizar las pruebas correspondientes se eligió una plataforma reconfigurable basada en una FPGA.

Los ensayos realizados permiten afirmar que la arquitectura propuesta es adecuada para soportar la funcionalidad descrita de un nodo sensor en el dominio elegido. Las pruebas se centraron en el intercambio de mensajes correspondientes a la definición del árbol de encaminamiento y permitieron

demonstrar su buen funcionamiento, resultando promisorias para avanzar en la implementación de la funcionalidad completa del algoritmo utilizado.

La plataforma de prueba elegida demostró ser sumamente flexible para realizar los cambios y correcciones que se llevaron adelante al ensayar el comportamiento de los nodos. Sus características permitieron evaluar distintas opciones de implementación de los bloques de la arquitectura y elegir la más conveniente.

REFERENCES

- [1] S.K. Singh, M.P. Singh and D.K. Singh, "Routing protocols in wireless networks – A survey," in International Journal of Computer Science & Engineering Survey (IJCES), vol.1 pp. 63-83, November 2010.
- [2] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," in Computer Networks: The International Journal of Computer and Telecommunications Networking (Elsevier), vol. 52 pp. 2292-2330, August 2008.
- [3] J.Portilla Berruoco, "Plataforma modular e interfaces genéricas de transductores para redes de sensores inalámbricas," Tesis Doctoral. Madrid, España. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, 2010. 224 p.
- [4] Z. Pei, Z. Deng, B.Yang and X. Cheng, "Application-Oriented wireless sensor network communication protocols and hardware platforms: a survey," in Proceedings of The 2008 IEEE International Conference on Industrial Technology (2008, Chengdu, China). IEEE Press, 2008, pp. 1-6.
- [5] J. Wilder, V. Uzelac, A. Milenkovic and E. Jovanov, "Runtime hardware reconfiguration in wireless sensor networks," in Proceedings of The 40th Southeastern Symposium on System Theory (New Orleans, USA). IEEE Press, 2008, pp. 154-158.
- [6] S. Commuri, V. Tadigotla, M. Atiquzzaman, "Reconfigurable hardware based dynamic data aggregation in wireless sensor networks," in International Journal of Distributed Sensor Networks (ACM), vol. 4 pp. 194-212, April 2008.
- [7] R. Garcia, A. Gordon-Ross and A.D. George, "Exploiting partially reconfigurable FPGAs for situation-based reconfiguration in wireless sensor networks," in Proceedings of The 17th IEEE Symposium on Field Programmable Custom Computing Machines (Napa, USA). IEEE Press, 2009, pp.243-246.
- [8] H. Hinkelmann, A. Reinhardt, S. Varyani and M. Glesner, "A reconfigurable prototyping platform for smart sensor networks," in Proceedings of The 4th Southern Conference on Programmable Logic (San Carlos de Bariloche, Argentina). SPL 2008. IEEE Press, pp. 125-130..
- [9] E. D'Agostino, R. Corti, E. Giandoménico, J. Belmonte y R. Martínez, "Clustering dinámico para tiempo de encendido mínimo en redes inalámbricas de sensores (CLUDITEM). Definición del árbol de encaminamiento," en Libro de Trabajos del XIV Congreso Argentino de Ciencias de la Computación (Chilecito, Argentina). CACIC 2008.
- [10] R. Corti, E. D'Agostino, E. Giandoménico, R. Martínez y J. Belmonte, "Clustering dinámico para tiempo de encendido mínimo en redes inalámbricas de sensores (CLUDITEM). Análisis de las fases de envío de datos.," en Libro de Resúmenes de la 36ª Conferencia Latinoamericana de Informática (Asunción, Paraguay). CLEI 2010. pp. 49-50. Trabajo completo en el CD del evento.
- [11] J. Ou and V. K. Prasanna, "Rapid energy estimation for hardware-software codesign using FPGAs," in EURASIP Journal on Embedded Systems, vol. 2006, 2006, pp. 1-11.
- [12] XILINX INC. PicoBlaze 8-bit Embedded Microcontroller User Guide. Disponible en: http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf
- [13] ALTERA CORPORATION. Nios II Processor Reference Handbook San José, USA. Disponible en http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf
- [14] XILINX INC. MicroBlaze Processor Reference Guide. Disponible en: http://www.xilinx.com/support/documentation/sw_manuals/edk82i_mb_ref_guide.pdf
- [15] K. Olukotun and L. Hammond, "The future of microprocessors," ACM Queue Magazine, vol.3, pp. 26-34, 2005.
- [16] B. Fletcher, "FPGA embedded processors: revealing true system performance," in Proceedings of The 2005 Embedded Systems Conference (San Francisco, USA), 2005, pp. 2-18.
- [17] XILINX INC. Spartan-3 Starter Kit Board User Guide. Disponible en: http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD_3.pdf
- [18] K. Chapman, "200 MHz UART with Internal 16-Byte Buffer", XILINX INC. USA, 2008. Disponible en: http://www.xilinx.com/support/documentation/application_notes/xapp223.pdf
- [19] XILINX INC. ISE WebPack Design Software. Disponible en: <http://www.xilinx.com/tools/webpack.htm>
- [20] XILINX INC. Embedded System Tools Reference Manual. Disponible en: http://www.xilinx.com/support/documentation/sw_manuals/edk82i_est_rm.pdf
- [21] XILINX INC. CoreConnect Architecture - On-chip Peripheral Bus. Disponible en: http://www.xilinx.com/ipcenter/processor_central/coreconnect/coreconnect_opb.htm
- [22] INTRONIX. 34 Channel LA1034 Logicport Logic Analyzer-Home Page. Disponible en: <http://www.pctestinstruments.com/index.htm>