

Diseño de un prototipo de procesador soft-core para aplicaciones en nodos de WSN

Oswaldo Lucio Marianetti,ⁱ Luis Arceⁱⁱ & Alfredo Iglesiasⁱⁱⁱ

Resumen

Las redes de sensores inalámbricos son redes ad-hoc, ya que se despliegan según una amplia gama de aplicaciones. Están constituidas por nodos sensores que cuentan con procesadores de propósito general y una o más estaciones bases. Los procesadores de los nodos no están optimizados para las necesidades de las diferentes aplicaciones de estas redes.

Una alternativa a estos procesadores, es un procesador soft-core optimizado respecto de la problemática particular de los requerimientos de la WSN. Este trabajo tiene como objetivo el diseño de un procesador soft-core con esas características.

Palabras claves: WSN, FPGA, nodos, reconfigurable, soft-core.

ⁱ Universidad de Mendoza. Argentina. Facultad de Ingeniería. Secretario Académico. Profesor Titular.

ⁱⁱ Universidad de Mendoza. Argentina. Facultad de Ingeniería. Secretario Instituto de Microelectrónica. Profesor Adjunto.

ⁱⁱⁱ Universidad de Mendoza. Argentina. Facultad de Ingeniería. Instituto de Microelectrónica. Profesor Titular.

Abstract

Wireless sensor networks are ad hoc networks as they are deployed across a wide range of applications. They are made up of sensor nodes that have general purpose processors and one or more base stations. The processors of the nodes are not optimized for the needs of the different applications of these networks.

An alternative to these processors is a soft-core processor optimized for the particular problem of the requirements of the WSN. This work has as purpose to design a soft-core processor with these characteristics.

Keywords: WSN, FPGA, nodes, reconfigurable, soft-core.

--

Fecha de recepción 14/04/2017 - Fecha de aceptación 04/08/2017

Introducción

Redes de Sensores Inalámbricos

Las redes de sensores inalámbricos (WSN) están constituidas por grupos (clusters) de nodos de dispositivos embebidos con conexión inalámbrica (sensores, microcontrolador o procesador más un módulo transmisor/receptor). Además cuentan con estaciones bases que reciben la información de los nodos de la WSN. Las estaciones base también pueden realizar transferencias de la información recolectada en los distintos clusters mediante diferentes alternativas de comunicación. (Por ejemplo una pasarela TCP/IP). Fig. 1.

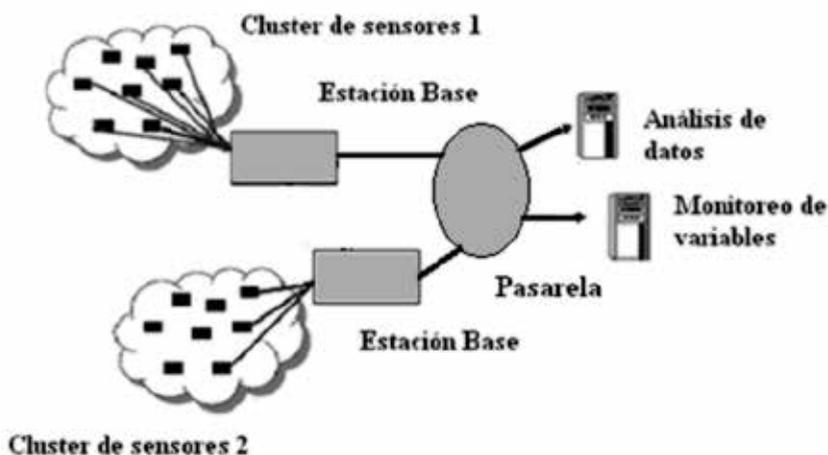


Figura 1. Modelo de una WSN

Los usos de las redes de sensores incluyen sensado ambiental y del entorno en el cual se despliegan, supervisión remota de pacientes, análisis climatológicos, vigilancia militar y de seguridad, seguimiento de productos y procesos de fabricación, control de seguridad y comprobación estructuras y emplazamientos de obra, hogares y oficinas inteligentes y otras tantas aplicaciones.

El despliegue o la cobertura del entorno a cubrir por la WSN se realiza en forma ad-hoc, es decir, la red se distribuye y organiza a partir de la aplicación para la cual se la va a utilizar. Una característica de las WSN es que una vez instalada, hay poca intervención humana sobre los nodos y la topología; elementos de la red que tienen elevada probabilidad de falla.

Los principales problemas a resolver en las WSN son:

Optimizar el consumo de energía disponible en los nodos, de modo de lograr

el máximo tiempo de vida útil de la red:

1) La energía disponible en cada nodo limita la cantidad de datos que pueden transmitirse entre nodos de la red, ya que durante la transmisión y recepción de datos es cuando mayor consumo de energía se produce. Este problema se complica aún más cuando la densidad de sensores y el tamaño de la red aumenta. El uso eficiente de la energía disponible requiere entre otras cosas, la optimización de la cobertura de la región de interés, colaboración entre nodos para extraer información del conjunto de sensores y minimizar la información redundante y el uso de protocolos y técnicas de ruteo que tengan en cuenta la energía disponible.

Para tratar estas problemáticas se han propuesto diferentes alternativas. Como por ejemplo estrategias basadas en cobertura eficiente en la organización de la red y el uso mínimo de la energía de cada nodo. Para la administración de esta energía disponible se han desarrollado protocolos MAC adaptativos y además técnicas dinámicas de ruteo. Algunos investigadores proponen el uso de técnicas de agregación de datos para minimizar los efectos de la redundancia en la información resultante de las mediciones de los sensores.

2) Optimizar los recursos de procesamiento:

Las WSN utilizan en su despliegue nodos con procesadores o microcontroladores de propósito general. Este diseño les da a los nodos la capacidad de procesamiento que requieren diferentes aplicaciones. Sin embargo, las arquitecturas de estos procesadores no están optimizadas para las necesidades particulares de cada una de las aplicaciones de redes de sensores. Los procesadores de propósito general tienen un diseño que admite prácticamente todos los tipos de aplicaciones, pero tienen un alto costo y el consumo de energía no está optimizado.

Para esta problemática se ha propuesto que los nodos cuenten con procesadores de “soft-core” (arquitectura configurable). Esto optimiza la arquitectura del procesador para que se pueda adaptar a las necesidades de las aplicaciones de las redes de sensores. Por lo tanto, el procesador puede contar con sólo aquellos componentes de hardware requeridos por la aplicación particular y así reducir el consumo de energía del procesador. Además, el procesador soft-core puede ser integrado en un chip con los otros componentes de hardware necesarios para desarrollar el nodo sensor. Esto reduce el tamaño de la placa de circuito impreso (PCB) requerido para implementar el nodo sensor y por lo tanto, reduce el costo del nodo.

3) Adecuar los nodos a la topología dinámica de la red:

Esta problemática se presenta ya que las WSN pueden contar con elementos móviles, nodos con alta probabilidad de fallo y nodos que entran y/o salen en el sistema. Este tipo de topología, se contradice con implementaciones ad-hoc, ya que las mismas presentan una cobertura estática y con poca capacidad de reorganizarse en función de las modificaciones que se presentan durante el periodo de funcionamiento de la WSN.

Una solución factible para esta problemática es recurrir a la tecnología de los dispositivos lógicos programables, en la particular las Field Programmable Logic Array (FPGA). Para esto se utiliza la reconfiguración parcial dinámica (DPR) de la FPGA, donde se reconfigura el nodo “header” de los clusters de nodos, para adaptar la configuración de los mismos.

Tecnología de Dispositivos Lógicos Programables Fpga

Los Field Programmable Logic Array (FPGA) consisten en un arreglo de celdas lógicas que pueden ser configuradas para implementar una función lógica determinada. Esto se logra a través de la carga de un archivo de configuración en la FPGA. Estos dispositivos permiten la reconfiguración estática cuya funcionalidad se encuentra determinada por el archivo de configuración cargado al inicializar el sistema.

La reconfiguración completa del dispositivo solo se puede hacer dejando sin energía la FPGA, borrando el contenido y recargando un nuevo archivo de configuración en el dispositivo logrando una nueva funcionalidad. Otro modo de reconfiguración es el conocido como DPR (Dynamic Partial Reconfiguration). En este modo, se borran solamente unas partes de la FPGA y el resto del dispositivo mantiene la configuración original. Este proceso se realiza mientras la FPGA tiene alimentación (energía).

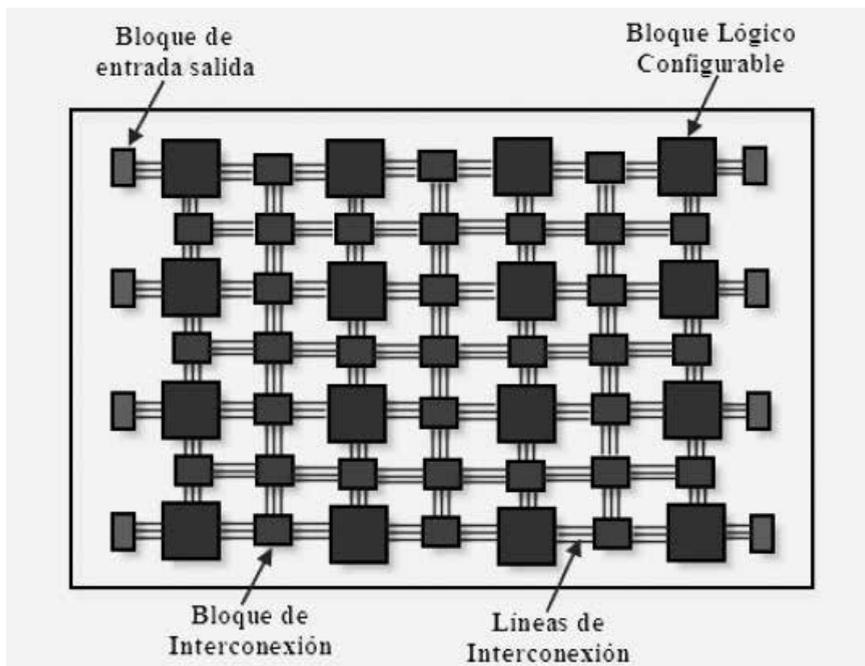


Figura 2. Arquitectura básica de una FPGA

La reconfiguración se realiza borrando el contenido de los Bloque Lógicos Configurables (CLB) y configurándolos con una nueva lógica. Por otra parte existe un módulo que permite el acceso a la reconfiguración parcial de la lógica de la FPGA (en el caso del fabricante de FPGA Xilinx, este módulo se conoce como Internal Configuration Access Port, ICAP). En la Fig. 2 se presenta la arquitectura básica de una FPGA.

La reconfiguración puede hacerse en tiempo real mientras el sistema está funcionando y se pueden incorporar nuevos componentes en función de la aplicación del sistema. Además, en los últimos años las FPGA se han reconocido como un dispositivo que permite incrementar la performance (velocidad de procesamiento) limitando el consumo de potencia. Para aplicaciones de WSN, otra capacidad importante de las FPGA es la posibilidad de reconfiguración remota y dinámica.

Durante el periodo de utilidad del sistema, es importante que cada nodo pueda ser reconfigurado si se requiere una nueva funcionalidad para el mismo. Para facilitar esta funcionalidad, los nodos de WSN basados en FPGA se deben desarrollar.

Según el texto “Future Wireless Networks and Information Systems”, Volumen 2. Editor: Ying Zhang. Springer, Jan 23, 2012; la utilización de configuración dinámica de FPGA puede aumentar la performance con menores espacios de memoria en aplicaciones de la industria automotriz y en aplicaciones de video. La configuración remota de FPGA se ha desarrollado en la industria aeroespacial. En el campo de las WSN pocos trabajos se han llevado a cabo donde se ha utilizado configuración estática en FPGA en redes ad-doc. Respecto de la reconfiguración dinámica remota pocos estudios se han realizado para aplicaciones de WSN.

Considerando la potencialidad de las diversas aplicaciones de las WSN, como así también la problemática intrínseca de las mismas, y teniendo en cuenta las posibilidades que presentan la tecnología y herramientas para el desarrollo basado en FPGA, es que se propone estudiar y diseñar un prototipo de nodo de WSN con características de reconfiguración remota, basándose en los dispositivos y elementos disponibles en el contexto local, de manera de lograr su transferencia al medio.

Arquitectura de prototipo de procesador soft-core

La arquitectura del prototipo de procesador soft-core desarrollada es simple. Se ha definido un bus de datos bidireccional de 8 bits. El bus de direcciones es también de 8 bits. Los registros internos del procesador son de 8 bits. Cuenta con un registro acumulador (A), un registro de instrucciones de 4 bits, que permite definir hasta 16 instrucciones, de las cuales el prototipo implementa solamente 11. Estas instrucciones están formadas por dos bytes. El primer byte contiene el código de operación y el segundo byte contiene el operando.

El set de instrucciones del prototipo de procesador soft-core es el siguiente:

Instrucción	Operación	Código Operativo
lda,(xx)	Carga el acumulador con el contenido de la dirección de memoria xx	0000
sta, (xx)	Carga en la dirección de memoria xx el contenido del registro A	0001
anda, (xx)	Realiza la operación AND entre el acumulador y el contenido de la dirección de memoria xx. El resultado se almacena en A	0010
adda, (xx)	Realiza la operación suma entre el acumulador y el contenido de la dirección de memoria xx. El resultado se almacena en A	0011
suba, (xx)	Resta el contenido del acumulador y el contenido de la dirección de memoria xx. El resultado se almacena en A	0100
ssi, (xx)	Salto condicional. Salta a la dirección xx si el contenido de A es cero	0101
sme, (xx)	Salto condicional. Salta a la dirección xx si el contenido de A es negativo (se verifica en el código VHDL de descripción. Se realiza en base una operación de resta)	0110
sma,(xx)	Salto condicional. Salta a la dirección xx si el contenido de A es positivo, distinto de cero (se verifica en el código VHDL de descripción. Se realiza en base una operación de resta)	0111
jmp,(xx)	Salto incondicional. Salta a la dirección xx.	1000
jr, (xx)	Salta a la dirección xx cuando se produce una interrupción	1001
ret	Retorno de interrupción.	1010

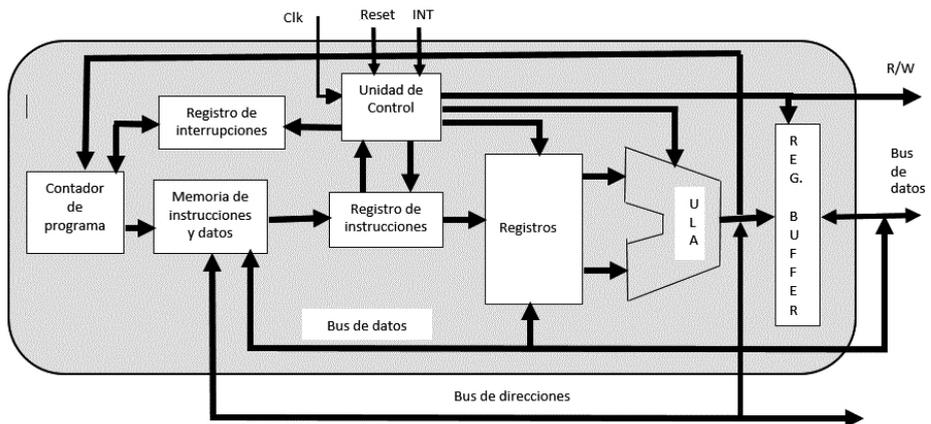


Figura 3. Arquitectura del procesador soft-core

En la Figura 3 se representa el diagrama en bloques del prototipo desarrollado. Se observan las señales de entrada de reloj (clk) y de inicialización (Reset). También se presenta una señal de control de lectura (un uno indica esta operación) y escritura (un cero indica esta operación). Además se encuentra el registro Contador de Programa y un Registro de Interrupciones. Esto permite que el procesador de soft-core soporte interrupciones para el servicio del procesamiento que requieran otros componentes conectados al procesador, como distintos sensores. Cuando recibe un requerimiento, se interrumpe el procesamiento para atender la acción solicitada por el dispositivo que interrumpe. El procesador maneja las interrupciones guardando los estados de los registros y almacenando la dirección de la instrucción interrumpida en el registro de interrupciones. Una vez que el procesador presta servicios al control de petición de interrupción, la ejecución de la instrucción Return (ret), retorna el control del programa a la instrucción interrumpida. Durante una interrupción, el control se transfiere a la rutina de servicio de interrupción ejecutando un salto (jr), que guarda la dirección de retorno.

El registro buffer permite la entrada o salida de datos, como así también la carga de instrucciones en la memoria. Este registro lo controla la Unidad de Control. Esta unidad está implementada con las instancias de los componentes que requiere su funcionamiento. La Unidad Lógico Aritmética permite la ejecución de las operaciones citadas en el conjunto de instrucciones del procesador

El modelo del procesador propuesto es el del tipo multiciclo, dependiendo de su complejidad unas instrucciones podrían ser ejecutadas en pocos ciclos y otras en más ciclos. Esto permite que las instrucciones tengan tiempos de ejecución diferentes, ya que emplean diferente número de ciclos de reloj. La máquina de estados del prototipo implementado se muestra en la Figura 4.

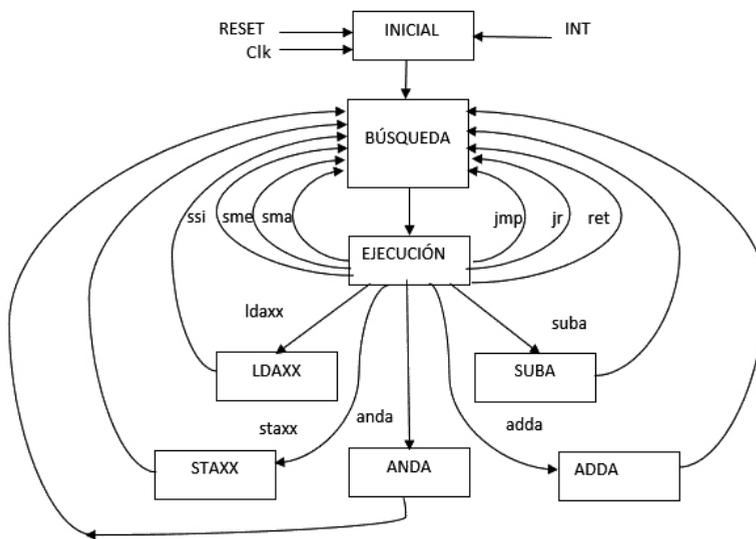


Figura 4. Máquina de estados del procesador soft-core

Síntesis de la descripción del prototipo

Para el desarrollo e implementación del procesador soft-core se utilizó el entorno de desarrollo Quartus II 10.1 Web Edition (32-Bit) de la empresa Altera y el kit DEO- Nano que contiene una FPGA Altera Cyclone IV EP4CE22F17C6N. Los resultados de los experimentos con estas herramientas se presentan a continuación.

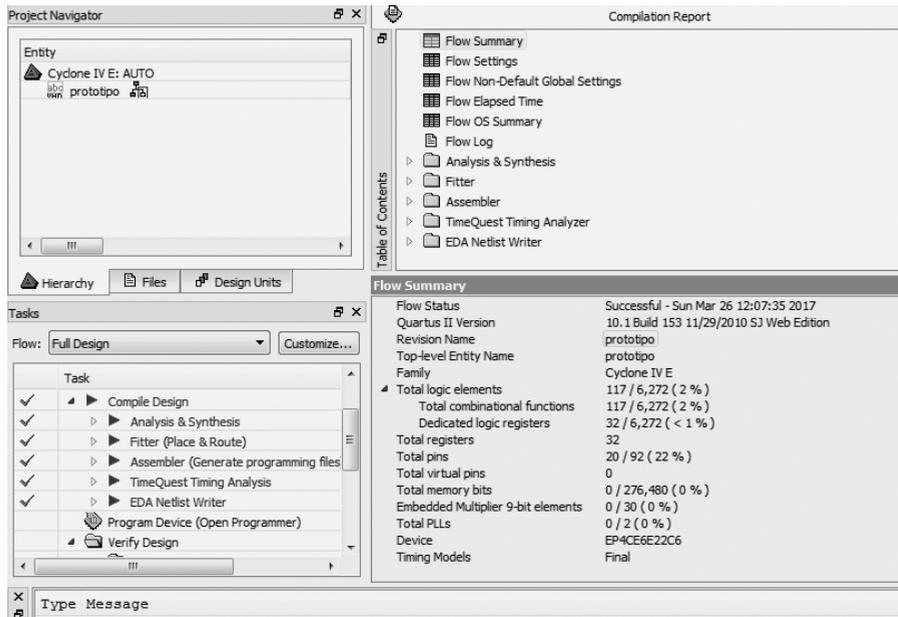


Figura 5. Resultados de la compilación del diseño del procesador soft-core

En la Figura 5 se observan los resultados de la compilación del código VHDL que describe el prototipo. Se observa el bajo porcentaje de recursos usados respecto de los que dispone el dispositivo utilizado (Cyclone IV).

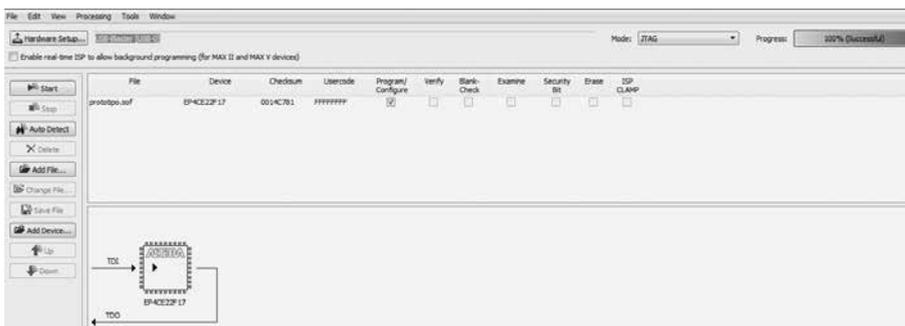


Figura 6. Resultados de la programación del diseño del procesador soft-core en el kit DEO- Nano

Conclusiones

Se desarrolló la descripción mediante código VHDL del prototipo de procesador soft-core. Se concretó la síntesis de este diseño y se programó en un dispositivo FPGA. El funcionamiento de la implementación se verificó con entidades de test con estímulos. Estos correspondían a carga de registros y accesos a memoria, como así también operaciones aritméticas.

Según Tomás Arredondo Vidal, en el documento “Estructuras de Computadores Digitales. Diseño de un Procesador Multiciclo”, se asume que un programa de prueba está constituido por la siguiente porciones de instrucciones: 25% de cargas, 11% de almacenamientos, 30% de sumas, 9% de restas, 4% de operaciones lógicas, 15% de bifurcaciones y 6% de saltos.

Para un procesador multiciclo el tiempo de ejecución se puede considerar que el valor promedio de CPI (ciclos por instrucción) para esta prueba es de 3,98.

En el caso del prototipo, el ciclo de reloj es 0,02 microsegundos (Frecuencia del clock 50 Mhz). Esto significa que el valor promedio de CPI será de 0,0796 microseg.

Se puede comparar este valor con el que corresponde a un procesador como el Atmel-0856-AVR, que se utiliza en la implementación de módulos XBee ZigBee para WSN, con un conjunto de instrucciones de 13 modos de direccionamiento y 128 instrucciones, con un clock de 8 Mhz. En este caso el valor promedio de CPI será de 0.4975 microseg.

El mismo cálculo se puede llevar a cabo con otro microcontroladores de propósito general que se aplican para nodos de WSN, tales como:

Texas Instruments MSP430. Con un set de 51 instrucciones con tres formatos y siete modos de direccionamiento adicionales para expandir el rango de direccionamiento de memoria. Con una clock de 8 Mhz.

Un caso a considerar es el del procesador Philips Semiconductors LPC2131/2132. Basado en una arquitectura ARM7TDMI-S. Tiene dos conjuntos de instrucciones:

El standard de 32-bit ARM set y uno de 16-bit Thumb set. Con un clock de 30 Mhz.

Otro caso es el Analog Devices ADuC836 MicroConverter®, Dual 16-Bit Sigma-Delta ADCs with Embedded 62 kB Flash MCU. Basado en el núcleo del microcontrolador de Intel 8051 con 256 instrucciones y un clock de 12 Mhz.

También se puede citar el microcontrolador HCS08 con más de 80 instrucciones y sus diferentes modos de direccionamiento y un clock de 40 MHz.

El prototipo ofrece en principio una mayor velocidad de procesamiento que un microcontrolador de propósito general, con una implementación más simple, si bien no es menor, que deben considerarse mejoras en la arquitectura, como instrucciones de manejo de bits e incorporar módulos de interface para comunicación con sensores.

Referencias

- Alcides Montoya, Diana Carolina Restrepo and Demetrio Arturo Ovalle, “Artificial Intelligence for Wireless Sensor Networks Enhancement”, disponible en: http://www.intechopen.com/books/smart_wireless-sensor_networks/artificialintelligence_for_wireless_sensor_networks_enhancement.
- Andre Mota, Leonardo B. Oliveira, Georgia P. Safe, Felipe F. Rocha, Ramon Riserio, Antonio A. F. Loureiro, Claudionor J.N. Coelho Jr., Hao Chi Wong1 and Eduardo Nakamura, WISENEP: “A Network Processor for Wireless Sensor Networks”, Federal University of Minas Gerais (UFMG), Brazil.
- Juan Valverde, Andres Otero, Miguel Lopez, Jorge Portilla, Eduardo de la Torre and Teresa Riesgo, “Using SRAM Based FPGAs for Power-Aware High Performance Wireless Sensor Networks”, ISSN 1424-8220, www.mdpi.com/journal/sensors, Sensors 2012.
- Lucas Iacono, Pablo Godoy, Ricardo Cayssials, Osvaldo Marianetti, Carlos García Garino, “Programmable Logic Devices in Sensor Networks: A Survey”, ITIC, Instituto Universitario para las Tecnologías de la Información y las Comunicaciones.
- Manoranjan Das, Banoj Kumar Panda, “Prototyping a Wireless Sensor Node using FPGA for Mines Safety Application”, ACEEE Int. J. on Electrical and Power Engineering, Vol. 02, No. 02, August 2011.
- Minh Duy Nguyen, “Reconfigurable Wireless Sensor Networks: A Survey and Future Works”, Embedded and Distributed Systems & Software Technical University Darmstadt, 21.12.2007.
- S. Commuri, V. Tadigotla and M. Atiquzzaman, “Reconfigurable Hardware Based Dynamic Data Aggregation in Wireless Sensor Networks”, International Journal of Distributed Sensor networks, 4: 194-212, 2008.
- Y. E. Krasteva, J. Portilla, J. M. Carnicer, E. de la Torre, T. Riesgo, “Remote HW-SW Reconfigurable Wireless Sensor Nodes”, 978-1-4244-1766-7/08/2008 IEEE.
- Yan Sun, Le Li, Hong Luo, “Design of FPGA-Based Multimedia Node for WSN”, Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference.
- Yibin Li, Zhiping Jia, Fucai Liu, “Remotely Run-Time FPGA Based WSN Node”, Future Computing, Communication, Control and Management Lecture

Notes in Electrical Engineering Volume 144, 2012, pp 57-63.

Ying Zhang, “Design of FPGA-Based Multimedia Node for WSN”, Future Wireless Networks and Information Systems, Volume 2, Springer, Jan 23, 2012.

Arredondo Vidal Tomás. Estructuras de Computadores Digitales. Diseño de un Procesador Multiciclo. Material de apoyo del texto de David Patterson, John Hennessy, “Computer Organization & Design”, (segunda y tercera edición),Morgan Kaufmann, CA. 2005

