

Soft-Core Processor Design for Sensor Networks Nodes

Ali Elkateeb

Electrical & Computer Engineering Department
University of Michigan – Dearborn
USA

ABSTRACT

Existing sensor networks are using off-the-shelf general-purpose processor in their nodes design to support them with the processing capability that required by different applications. However, the architectures of these processors are not optimized for the needs of sensor networks applications. Since the general-purpose processors have a design that support all types of applications, these processors are generally have high cost and consume large power.

Supporting sensor nodes with soft-core processor will optimize the processor architecture to match the needs of the sensor networks applications. Therefore, the processor will support only those hardware components required by the target application and that will reduce the processor power consumption. In addition, the soft-core processor can be integrated on a chip with the other hardware components required to develop the sensor node and that will reduce the size of the Printed Circuit Board (PCB) required to implement the sensor node and hence, reduce the sensor node cost.

In this paper, we are presenting the design for the soft-core processor using Field Programmable Gate Array (FPGA) technology. The soft-core processor, instructions set, core testing, and performance are presented in this work.

Keywords: Sensor networks, soft-core processor, multi-cycle architecture, routing protocols, FPGA technology.

1. Introduction

The wireless sensor networks (WSNs) have become widely used to support large variety of applications. One of the key components in WSN design is the sensor node, also known as mote. Typically, the sensor networks nodes consist of sensor(s) to monitor the surrounding environment, a processing unit, a communication unit to communicate with neighboring nodes or base station, and an onboard power supply [1, 2, 10] (Figure 1).

The General-purpose microprocessors are normally working in an unimpeded power supply environment. Sensor networks, on the other hand, are constrained for power where they stress upon utilizing an existing low power processing unit [3]. Despite the fact of the constraint power feature of sensor network nodes, the microprocessors are commonly used in existing sensor networks nodes design due to their wide availability and their architecture that can fit the needs of most applications. Examples of these microprocessors that used in existing sensor nodes are Atmel AVR 8bit Microprocessor, Intel PXA271 "Bulverde", Texas Instruments MSP430, Atmel AtMega 1281, etc. The general-purpose microprocessors are typically packed with different features, which might not be required in the processing of the sensor networks applications and that will unnecessarily increase the power consumption of the sensor nodes [12].

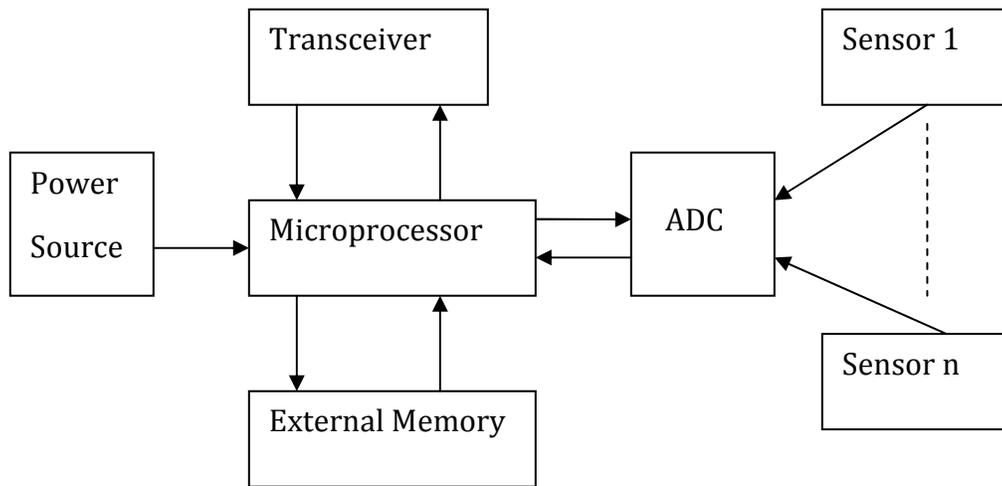


Figure 1: Sensor Node Organization

The advances in the reconfigurable computing, System-On-Chip (SOC), Field Programmable Gate Array (FPGA) technology, and the CAD design tools have made possible to develop a soft-core processor and other nodes' units in short period of time. The soft-core design can be optimized to match the processing needs of the sensor node applications. Also, the soft-core processor has the design flexibility that can be modified to support special processing needs by the sensor networks applications. One of the captivating features of the soft-core processor design is the capability to optimize its size and power to fit the needs of the target applications.

In this paper, a flexible soft-core processor architecture, design, and performance evaluation has been discussed. The next section will cover the related work section on hard and soft-core processors. Section three will investigate the proposed soft-core architecture, instructions set, and the interrupt scheme. Section four will highlight the sensor node units design such as the display and communication units. The soft-core processor implementation and performance evaluation have been discussed in section five and six respectively. The paper is concluded in section seven.

2. Related work

The design of the sensor network nodes are achieved by using the commercially available general-purpose embedded hard-core processors and some other required off-the-shelf components such as wireless transceiver, memory, sensors, etc. Typically, these hard-core processors are designed to support features and capabilities required by the general-purpose applications. However, many of their features and capabilities may not be fully require for WSNs applications. Therefore, the hard-core processors are not designed to be optimized for WSNs applications and they consume large energy.

To have a flexible processor in sensor nodes is important design feature to support the nodes to perform the processing required by the WSNs applications. The use of Very large Scale Integration (VLSI) or Application Specific Integrated circuit (ASIC) technologies in the design of hard-core processors has made the design of these processors inflexible. However, the soft-

core design approach by using the Field programmable Gate Array (FPGA) technology can produce nodes that have flexible design to fulfill the applications needs. The MicroBlaze [7] from Xilinx, Nios and Nios II processors [8] from Altera are some examples of the soft-core processors that designed for general-purpose applications. In addition to the design flexibility of the soft-core processors, they can be designed to achieve high performance.

Using FPGA in WSNs nodes can support the integration most, if not all, components that required for the sensor node design on a single FPGA chip, also known as Reconfigurable System-On-Chip (RSOC). For example, components such as processor, sensors' interfaces, small size of memory, memory interface, wireless transceiver, etc., can be integrated on a single FPGA chip. This integration will reduce or eliminate the use of some of the off-the-shelf components that used in nodes development and that will improve nodes' reliability, cost, and reduce the size of the sensor nodes.

The sensor networks nodes design require using nodes that consume low power, have flexible design, and provide good processing performance. The FPGA is the technology that provides an optimal balance between design flexibility and processing performance that required for sensor networks applications [13, 14, 15]. However, the FPGA power requirement is high comparing to the use of other technologies. The nodes' power consumptions can not be considered as a problem when these nodes used in some applications where the continuous power supply is exist, such as when nodes are used inside a building [9]. However, most of the WSNs applications are using these nodes on the remote applications fields where the nodes power consumption should be very low.

Different nodes design methods can be used to reduce their power consumptions such as using low-power FPGA chips, use an optimized soft-core processor architecture that has small amount of hardware components such as the use of multi-cycle architecture [4]. As one of the key components in sensor nodes design is their processors, the focus of this work is given to the development of a soft-core processor for sensor nodes using a multi-cycle architecture.

3. Soft-core Processor Design

3.1. Soft-Core Architecture

The architecture of the sensor node processor is required to be simple, consume minimum power, and support suitable instructions for the target application processing [11]. Also, the processor should provide the processing performance required by the application.

The sensor node processor design is a 16-bit soft-core multi-cycle architecture. This multi-cycle architecture allows using the processor's functional units more than once per instruction execution cycle. For example, the Arithmetic Logic Unit (ALU) can be used to compute the next program counter address in one cycle and perform the arithmetic/logic operation in another cycle of the instruction execution. Therefore, the multi-cycle architecture can helps to reduce the amount of required hardware for the soft-core processor design, which is very important for reducing the processor power consumption [4]. The use of multi-cycle architecture in sensor node design will produce a smaller processor core and that will free more space on the chip to integrate other functional units needed for the processing of the sensor network applications.

Despite that the multi-cycle based architecture will not provide high performance like the pipeline based architecture, the performance provided by the multi-cycle will be still enough to provide the processing required by the sensor networks applications (as will be discussed in section 4). However, using pipeline architecture in sensor node design will require more hardware and consume more power than multi-cycle architecture and therefore, the multi-cycle architecture is more appropriate to be used in sensor networks nodes than the pipeline architecture.

The soft-core processor is designed to execute basic logical and arithmetic instructions. The datapath of the soft-core consists of Instruction Register, Data Register, eight 16-bit Registers, Program Counter, Exception Program Counter, Arithmetic and logic unit (ALU), and the ALU out buffer located at the output of the ALU (Figure 2). The soft-core processor designed in this research is running at a frequency of 50MHz. The Instruction register holds the instruction read from the memory during the fetch clock cycle and provides it to the subsequent stages to complete the instruction execution.

The ALU of the soft-core processor is designed to compute the next instruction address, effective address to fetch operand, store data during a memory operation, and performs basic logical operations. This approach requires the output of the ALU to be saved during every clock cycle at the ALU out buffer in order to carry out instruction execution at the subsequent clock cycles.

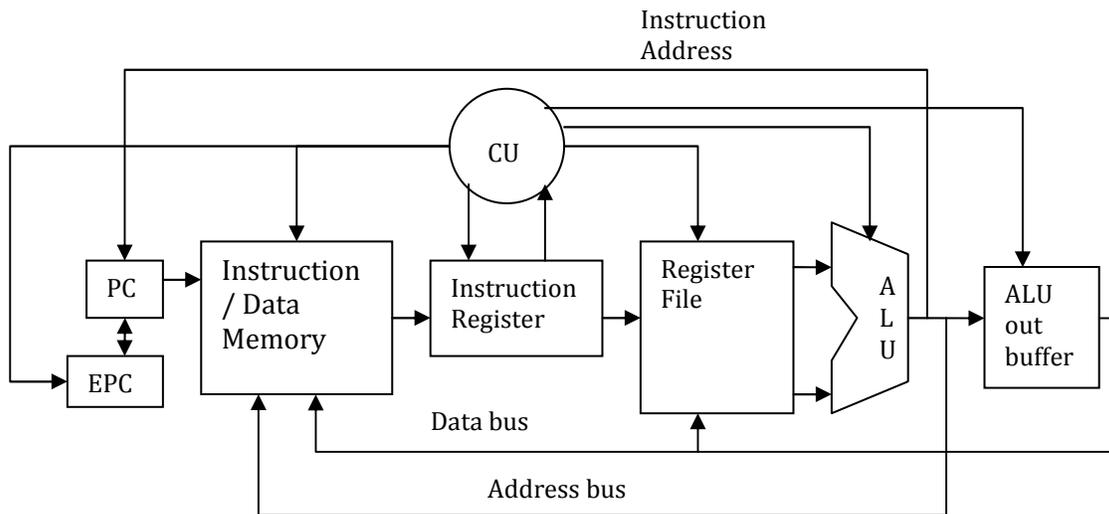


Figure 2: Soft Core Processor Architecture

The soft core control unit is based on Moore Finite State Machine. Inputs to the processor control unit comprise of a 7-bit operation code (opcode) that comes from the instruction register, a reset signal, and an interrupt signal (Figure 2). The output from the control unit controls the various data selectors, registers, provides the ALU with the required operations, and handles interrupts. The register file is designed to work with both edged of the clock by support writing on the falling (negative) edge and read on the rising (positive) edge of the clock.

3.2 The Instructions Set

The architecture consists of 15 instructions of three different types: register, immediate, and jump. The register type instructions consist of basic logical, arithmetic, load, and Store instructions. The immediate type instructions include Load word immediate, Store word immediate, and AND immediate. The jump type instructions include Branch Equal (BE), Branch Less Than (BLT), Branch Greater Than (BGT), Jump and link (JAL), Jump return (JR), and return (RET).

These instructions are of fixed format 16-bit wide composed of 4 fields: 7 bit Opcode, two 3 bit source registers & one 3-bit destination register. For Load & Store instructions, the two source registers provide the base & offset addresses (Figure 3). For the store instruction the destination register field specifies the register that contains the memory address. The immediate type instructions such as Load word immediate (LWI), Store word immediate (SWI), and Branch instructions are 32-bit long. The bit organization is similar to the register type instructions except that an additional 16-bit for the immediate data will be included in the instruction format.

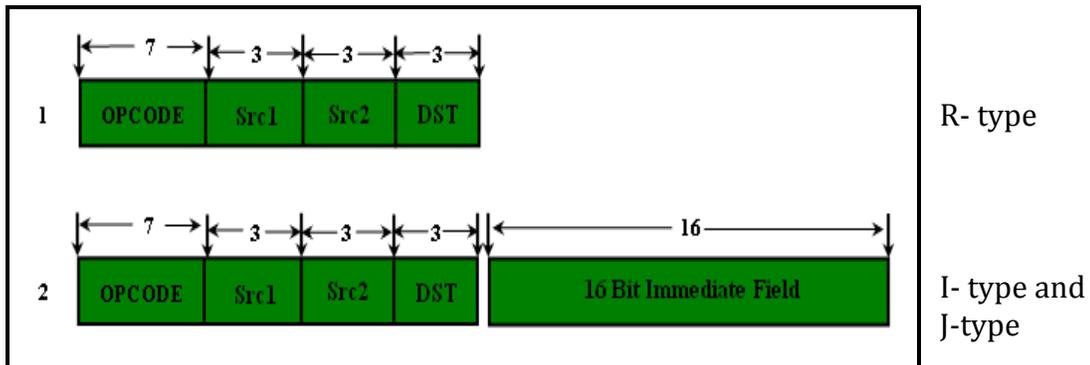


Figure 3: Soft Core Processor Instruction Format

The clock per instruction (CPI) varies from one instruction to others. Of all the instructions, the Load word immediate and Branch instructions require the most number of clock cycles. The clock cycles for each instruction are:

All register type required	4 clock cycles
LW	5 clock cycles
LWI	6 clock cycles
SW	5 clock cycles
SWI	5 clock cycles
BE	6 clock cycles
BGT	6 clock cycles
BLT	6 clock cycles
ANDI	5 clock cycles
JAL	5 clock cycles
JR	4 clock cycles
RET	2 clock cycles

3.3 Interrupts

The soft-core processor support Interrupts to service the communication unit and other processing components attached to the soft-core processor, such as the CMOS pixels sensor. The communication unit of the sensor node transmits and receives information and commands with the neighboring nodes and base station. When the communication unit receives a request for data, it interrupts the soft-core processor to take appropriate action. The processor handles the interrupts by saving the states of the registers and preserves the address of the instruction interrupted. The address of the interrupted instruction is preserved in the Exception Program Counter (EPC).

Once the processor services the interrupt request control, it returned to the interrupted instruction by executing the Return (RET) instruction. During an interrupt after saving the register states, the control is transferred to the Interrupt service routine by executing the Jump and Link instruction (JAL), which saves the address of the instruction that marks the beginning of the program that restores the register contents. The link address is stored in the register R5 of the register file. The soft-core processor is supported with a non-preemptive interrupt to service all units supported by the sensor node.

4. Sensor Node Support Units

The sensor node will integrate other units with the soft-core processor to provide the support required by the sensor networks applications. In addition to the processing of the basic operations, the soft-core processor will support the control function for the other sensor nodes, units. For example, to provide the support for security surveillance applications, sensor nodes need to be provided with different units such as LCD for displaying pictures and characters, wireless channel interface to transmit/ receive data, routing protocol unit to support nodes communications, CMOS pixels sensor for picture capturing, etc. However, we explore the use of the soft-core processor in supporting the display and routing protocol units design. The other units design such as the CMOS pixels sensor unit will be included with future publication on Sensor Node System-On-Chip (SNSOC) design.

4.1 Display Unit

The task of the soft-core processor is not designed to only support the nodes basic operations but also to support other nodes' units to perform their processing tasks and coordinate their operations. We have used Xilinx Prototyping board that based on Spartan 3A chip XC3S700A for developing and testing the sensor node design [6]. The LCD picture/video display and the LCD for characters display interfaces units have been integrated with the soft-core processor on the same Spartan FPGA based node design. Also, the VGA controller has been developed using the board generic DB15 VGA port that exist on Spartan board. Since the operating frequency of the soft-core processor is high comparing to the pixel frequency, a Digital Clock Manager (DCM) is used to reduce the VGA Controller frequency from 50MHz to the frequency required by the VGA controller which is 25MHz. The designed VGA controller has been tested by using the DB15 connector for the VGA monitor that exist on the Xilinx prototype board (Figure 4).

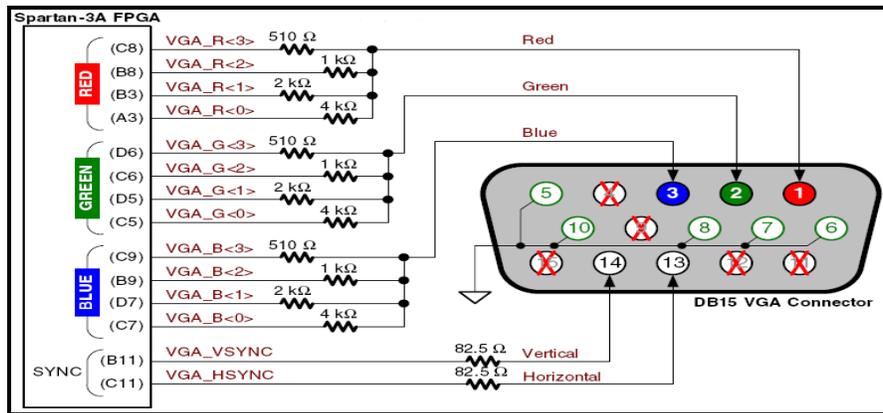


Figure 4: DB15 VGA Interface on Spartan-3A board (Source: Xilinx Spartan-3A Manual)

Displaying information on the LCD is useful in some applications when information could be displayed on the LCD at the base station and on some sensors nodes to inform users about the activities on different nodes. The Spartan-3A FPGA control the character LCD screen that has two-lines, 16-character via a 4-bit data interface (Figure 5). The interface for this LCD is controlled by the soft-core processor on the sensor node that has been used to test three LCD operation states for display initialization, display characters, and clearing the display. The soft-core display controller is processing a state machine that cycle between the above three states.

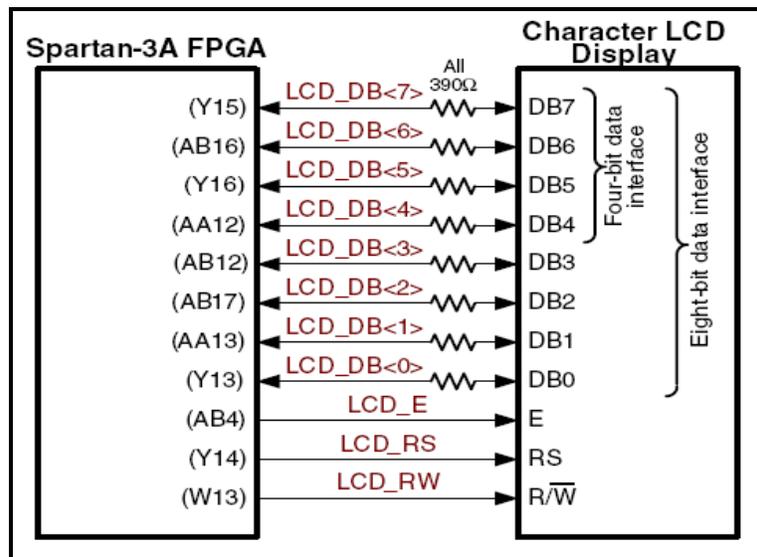


Figure 5: Character LCD & Spartan-3A Interface (Source: Xilinx Spartan-3A Manual)

In addition, the interface for image LCD that displaying pictures and video has been designed and soft-core processor can process the picture before displaying it on the LCD on the sensor node. The LCD image has 140 by 140 pixels resolution and it can be very useful for security monitoring application, and especially on the base station and on some of the sensor network nodes. Also the interface for the VGA display is design to be used to support the base station with VGA monitor as an alternative method to display pictures on the based station if it is not equipped with LCD display.

4.2 Routing Protocol Unit

The soft-core processor is required to perform the routing protocols processing to achieve nodes communications. There are many routing protocols used for wireless sensor networks such as classic flooding, node centric communication protocol, Sensor Protocol for Information via Negotiation (SPIN) [5], etc. However, the SPIN protocol is used to explore the instructions set capability to achieve protocols processing efficiently by the soft-core processor.

SPIN Protocol is a Negotiation based Information Dissemination Protocol that overcomes the limitations of the Classic Flooding and Gossiping protocols. Before transmitting data the protocol perform a negotiation stage between nodes to ensure that only relevant information is transmitted. SPIN uses three types of messages to communicate and negotiate: i) Advertise (ADV): when a sensor node detects new data that it intends to share, it sends out an Advertise message containing the meta-data of the sensed data to its neighbors; ii) Request (REQ): Upon receiving the advertise message the receiving node runs a check to determine if it carries a copy of the data with a matching meta-data. If the node carries a copy of the data it ignores the advertise message else it sends back a request message to the source node that broadcasted the advertise message; iii) DATA: Once the source node receives the request message it sends the actual data along with the meta-data header to the requesting node.

In order to simulate the processing of this protocol, the following assumptions have been made:

- i) When data is transferred to the requesting node, this node responds with an Acknowledge message.
- ii) A Termination message is sent to indicate the final chunk of data.

Hence, the protocol implementation has included the Acknowledge and Terminate messages in addition to the Advertise, Request and Data. The protocol frame of the transmitted message is configured to contain the control and data bits. The control bits comprise the Requesting node address, Source node address and relevant message of type: Adv, Req, Data, Ack or Terminate. The size of the data bits is left flexible and can be controlled by the protocol designer's discretion. The transmitted frame is designed to be located in memory buffer and accessible to the transceiver unit. The processor responds to a message appropriately by saving the response message in buffer locations and signals the transceiver unit to transmit the message. The upper 8 bits of the control bits are designed to be the source and destination node address, and the lower 8 bits are the broadcast messages.

The simulation of the SPIN protocol has been conducted using the instructions set of the soft-core processor. This simulation process has shown that the designed soft-core instructions are suitable for processing sensor network SPIN protocol.

5. Soft-core Processor Implementation

The Soft-Core Processor architecture was developed using FPGA Xilinx Integrated Software Environment (ISE) design tools. The VHDL language has been used in developing the code of the designed soft-core processor. The design has been tested for behavior and timing functionality. Design implementation that includes translating, mapping, placing & routing stages are conducted using ISE design tools. The mixed modeling is used in this work

where some parts of the soft-core are developed with structural modeling such as the ALU part, where behavioral modeling is used in developing other parts such as the register file. We have used the ModelSim simulation tools version XE III 6.3c to perform behavioral & timing simulations.

The final soft-core implementation has been achieved by synthesis the VHDL code of the soft-core design. The IMPACT tool, which is integrated with ISE design tools, has been used to download the configuration bits to the FPGA chip. Also, the configuration bits are loaded to board Flash memory that helped to download the code of the soft-core to the FPGA chip automatically at board power-up phase. In addition to the testing of the soft-core processing of the communication routing protocol, the core is tested to display information on the character and image LCD display and on the VGA monitor that connected to the DB15 connector of the Spartan 3A board.

The soft-core processor occupies small portion of the Spartan 3A chip as shown in Figure 6. Clearly, there is enough space on the FPGA chip to hold more supporting hardware to accommodate the components required to develop a complete sensor node. However, a smaller and inexpensive FPGA chip than Spartan 3A can still be used to develop a low-cost sensor node.

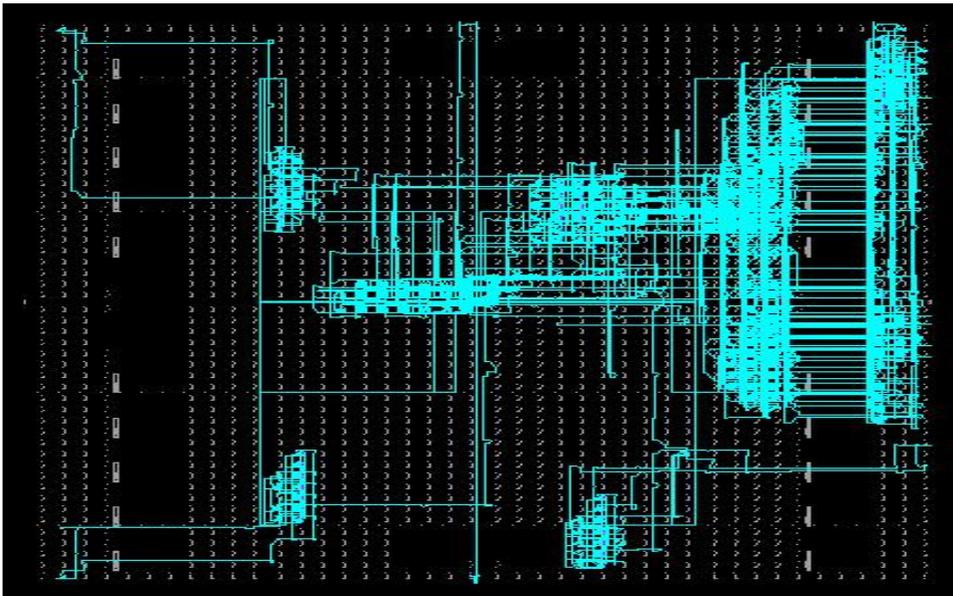


Figure 6: The Soft Core Processor Design on FPGA Chip

6. Performance Evaluation

We have use a function from the SPIN protocol as the test bench to evaluate the performance of the soft-core processor. Clearly, using one function will not give an accurate evaluation result and therefore, more functions will be developed in future for measuring the soft-core performance. The performance was measured in terms of MIPS (Millions of Instructions per Second).

The performance was computed by considering the number of instructions of the SPIN program and the clock cycles for each instruction as stated in the following equation [4]:

*Execution Time = Clock Cycles for Program * Clock Cycle Time*

*The total Clock Cycles for program execution = $\Sigma CPI_i * N_i$,*

Where, CPI_i is the Clock per Instruction,
 N_i is the number of instructions ($i \rightarrow 1$ to n)

Since the total number of instruction in the test bench program = 77

Therefore, the total number of Clock Cycles required to implement SPIN Protocol = 357

The clock rate of the soft-core processor is 50MHz, i.e., cycle time = 20ns

Therefore, the total execution time = $357 * 20ns = 7.14\mu sec$

Since $MIPS = 10^6 * (Instruction\ Count / Execution\ Time)$

Therefore, soft-core performance = $10^6 * 77 / 7.14\mu sec = 10.78 MIPS$

7. Conclusion

In this paper, we presented a soft-core processor designed to support the sensor network nodes processing. The soft-core processor design is optimized to the processing needs of the sensor node used for surveillance application and therefore, such design will minimize the node's size and its power consumption. We believe that the use of multi-cycle architecture is suitable for the soft-core design since it will minimize the hardware required for the processor design, reduce the power consumption, and provide enough processing power required by most, if not all, sensor networks applications. Therefore, we did not support the soft-core with pipelining architecture or other architecture to increase the performance of the soft-core processor.

In addition to the basic operations processing, the soft-core has the support for other processing components such as the interface components to support character and graphic LCD display and VGA monitor. Such support will make the soft-core suitable for monitoring environment, security, and surveillance applications. We are planning to provide more processing components support in future such as CMOS pixels sensor, encryption, other communication protocols, etc., to enhance the soft-core capability to support more applications. As the soft-core developed with VHDL code, a flexible and portable core design can be produced to be implemented over different FPGA product such as those chips from Xilinx, Altera, Actel, etc.

8. References

1. Dragos, Niculescu, NEC Laboratories America, "Communication Paradigms for Sensor Networks", IEEE comm. March 2005.
2. Ian Downes, Leili Baghaei Rad, & Hamid Aghajan, "Development of a Mote for Wireless Image Sensor Networks", In Proc. of Cognitive Systems and Interactive Sensors (COGIS), Paris, March 2006.

3. Siva D. Muruganathan, Daniel C. F. MA, Rollyi. Bhasin, & Abraham O.Fapojuwo, "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks", IEEE Radio Comm. March 2005.
4. David A. Patterson, John L. Hennessy, "Computer Organization and Design", University of California, Berkley and Stanford University.
5. Joanna Kulik, Wendi Rabiner, Hari Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", Massachusetts Institute of Technology, Cambridge, MA 02139.
6. Spartan-3A Starter Kit Board User Guide, UG330 (v1.1) February 15, 2007, Xilinx Inc. http://www.xilinx.com/support/documentation/boards_and_kits/ug330.pdf
7. Xilinx, Inc. MicroBlaze Soft Processor,; August 2004. http://www.xilinx.com/xlnx/xil_prodcat_product.jsp?title=microblaze
8. Altera Corporation. Embedded Processor Solutions Overview, January 2005. http://www.altera.com/technology/embedded/embedded/embedded_processor_solutions.html.
9. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, pp. 102-114.
10. G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Commun. ACM*, vol. 43, no. 5, May 2000, pp. 551-558.
11. A. Boulis, C. Han, and M. Srivastava, "Design and implementation of a framework for efficient and programmable sensor networks", *Proceedings of the first International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, 2003.
12. L. Horton, and et. al., "The intel mote platform: a Bluetooth-based sensor network for industrial monitoring", *IPSN 2005*, April 2005, pp. 437-442.
13. Y. Krasteva, and et. al., "Virtual Architectures for Partial Runtime Reconfigurable Systems Application to Network on Chip based SOC Emulation", *Proc. of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON 08)*, Nov. 2008.
14. Y. Krasteva, and et. al., "Remote HW-SW reconfigurable Wireless Sensor nodes", *Proc. of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON 08)*, Nov. 2008.
15. Simon Han, and et. Sl., Reconfigurable Sensor Networks with SOS, *UCLA Electrical Engineering Department Annual Research Review (ARR) for 2004*, University of California, Los An , October 2004.