Wireless Sensor Nodes Processor
Architecture and Design

Ali El Kateeb, Aiyappa Ramesh, and L. Azzawi
Electrical and Computer Engineering Department
University of Michigan –Dearborn
USA

ABSTRACT
The sensor networks are intended to support a variety of applications. Providing sensor network with flexible nodes design will make possible to support a variety of applications. A detailed view of soft processor core design and the instructions set for flexible sensor network nodes is presented in this paper. This soft core design can be easily modified and integrated with other components to construct different types of sensor nodes. The soft core is implemented using Xilinx ISE design tools. The core has been tested using cycle accurate timing simulation. The performance of above 10 MIPS can be achieved with the designed soft core.

1. Introduction
Wireless Sensor Networks support wide applications in the field of distributed sensing. Each sensor network has many nodes (or motes) where the node consists of sensors, micro-controller to process information received from the sensors, communication unit to communicate and transfer information to neighboring nodes or base station, and an onboard power supply.

There are different microcontrollers are currently used in sensor nodes design such as Atmel AVR 8bit, Intel PXA271 "Bulverde", Texas Instruments MSP430, Atmel AtMega 1281, etc. Compared with general purpose processors, the sensor nodes consume much lower power and that achieved by trading off advanced computational features that typically used in every high performance processors such as floating point unit. Despite designing node processors to operate in a power constrained environment, these processors architecture are still not fully optimized to sensor networks applications need.

These sensors nodes are typically required to perform simple processing that involve reading and processing the sensing values, communicate these values to other nodes, etc. However, new sensor networks applications such as surveillance will require more processing capability than the nodes using in current sensor networks.

Using the soft core processors in sensor nodes will provide the advantage of allowing of simple modifications to reuse the same core as the application changes. These observations have led us to develop a "soft core processor” specifically for wireless sensor networks design. A soft core processor also has the advantage of being modified to include special processing unit that can be integrated with the soft core processor to support special processing that needed by certain applications.

2. Soft Core Processor Architecture
The flexible sensor network node that design in this work included the soft core processor and other interface units that required for supporting different sensor networks applications. Since this work is focusing on the surveillance applications, camera interface and VGA display units will be integrated with the soft core unit to provide the sensor node design (Figure 1).

The soft core processors architecture closely resembles the MIPS processor architecture. The architecture features a 16-bit data path and executes instructions over multiple clock cycles. The processor operates at a frequency of 50MHz.

The processor functional unit consists of basic sequential and combinational elements (Figure 2). Additional registers are made use to hold data form the previous stage as the instructions are executed over multiple cycles. The data path elements consists of Instruction Register, Data Register, Register File, Program Counter, Exception Program Counter, Arithmetic and logic unit and ALU OUT Register. 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 Register File consists of eight 16-bit wide registers. Registers R0 thru R5 are used to hold operands and results during instruction execution, registers R6 and R7 holds the base address for two different memory locations. On interrupt the states of registers R0 thru R5 are preserved in the Block RAM. The Data Register is used to hold 16-bit immediate address during Load, Store immediate and branch instructions. It also holds the data being read from the memory during Load instructions. The exception program counter (EPC) holds the address of the instruction being interrupted. Once the interrupting device is serviced the program counter is loaded with the address of the offended instruction from the exception program counter.

The multi cycle architecture allows the design to utilize a single Arithmetic Logic Unit more than once per instruction. The ALU computes the instruction address, effective address to fetch and 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 in order for the subsequent clock cycles to carry out instruction execution. The soft core processor ALU can perform AND, ANDI, OR, XOR, Add, Subtract, Branch Equal, Branch Less Than, Branch Greater Than operations, JAL and JR.

The soft core processor control unit design is based on Moore finite state machine. Inputs to the processor control unit comprise of a 7-bit opcode that comes from the instruction register a reset signal and an interrupt signal. The output from the control unit controls the various data selectors, registers, provides opcode to ALU, and handles interrupts. All registers are written on the falling (negative) edge and are read on the rising (positive) edge.

One of the captivating features of the Soft Core Processor is its compact size. The processor occupies approximately 15-25% of FPGA Spartan 3A chip area, leaving enough room for additional designs units (Figure 3). Advancements in FPGA technology has made it possible to have the soft core processor and other controller designs on the same chip. Other units required for the nodes design such as VGA controller, frame capturing from camera can be integrated with the soft core processor on the same chip. Clearly, the System-On-Chip (SOC) design approach can be used for the flexible sensor node design.

The architecture consists of 15 instructions of R-type, I-type and J-type format that used with MIPS architecture. The R-type instruction consists of basic logical and arithmetic instructions. The I-type format is used for Load, store, conditional branch, and jump and link instructions. The J-type format used for Jump instruction.

The cycles per instruction (CPI) varies between instructions. Of all the instructions the Load word immediate and Branch instruction requires the most no of clock cycles. The clock cycles for each instruction are:

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Clock Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>All R-Type instructions</td>
<td>4</td>
</tr>
<tr>
<td>LW, SW, SWI</td>
<td>5</td>
</tr>
<tr>
<td>LWI, BE, BGE, BLE</td>
<td>6</td>
</tr>
<tr>
<td>ANDI, JAL</td>
<td>5</td>
</tr>
<tr>
<td>JR</td>
<td>4</td>
</tr>
<tr>
<td>RET</td>
<td>2</td>
</tr>
</tbody>
</table>

The Return (RET) and Jump and Link (JAL) instructions are used for serving the Interrupt service routine.

3. Simulation

The VHDL model for the soft core processor architecture was developed using Xilinx ISE design tools. The design was tested for behavior functionality using ModelSim XE III 6.1e simulation software. A Post Route Simulation was also carried out to check the timing correctness of the processor. The cycle-accurate simulation is used for testing every instruction using ModelSim simulator. The design was mapped to Spartan 3A prototyping board for further testing. The board offers state of the art XC3S700A FPGA chip on which the processor soft-core was tested. The FPGA has an internal Block Ram of 360 Kbytes which used to store the program that used for processor testing [3].

The layout of the Soft Core Processor on Xilinx XC3S700A chip is shown in (Figure 3). The image also highlights the amount of chip space occupied by
the processor, leaving enough space to accommodate other designs.

4. Soft Core Processor Testing
Besides testing the core on the processing instructions, the processing of one of the important wireless sensor network communication protocol was conducted using the developed soft core. The Soft Core Processor and the processor instruction set are designed to implement the SPIN (Sensor Protocol for Information via Negotiation) Protocol [1]. The SPIN protocol is a negotiation based protocol that negotiates with neighboring nodes when transmitting data. This ensures that only relevant information/data is transmitted and thereby help conserve much needed power. The protocol makes use of Meta Data that describes the information (image data) being transmitted. The receiving node then checks the Meta data in order to prevent duplication of data.

The Input Output operation between the Soft Core Processor and the communication unit is Memory Mapped (MMIO). The processor makes use of the Load and Store Immediate instructions to store and retrieve data from the I/O device address space. The Memory Mapped I/O approach was chosen since implementing Port Mapped I/O meant including IN and OUT instructions. The Soft Core Processor is built for image processing application, Memory Mapped I/O system is ideal. Since the data to be transferred is of the order of Kilo bytes it can be easily transferred by loading the data into the I/O device address space and then signaling the I/O device.

The Soft Core Processor is built to perform image capturing process that required for further processing. It involves detection of potential security breach or environmental changes, etc. Once the image is buffered the processor can then operate on it and relay the processed image results to the base station.

5. Processor Performance
The sensor mote processor performance was determined by considering the SPIN protocol as a test bench. The performance was measured in terms of MIPS (Millions of Instructions per Second). An implementation of SPIN protocol using assembly level instructions was used for the purpose (Figure 2).

The performance of the processor in MIPS using the SPIN protocol was computed by considering the clock cycles for each instruction in the program and the number of instructions in the SPIN program. The equation used to compute processor performance in MIPS is given by [2, 4]:

\[
\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}
\]

\[
\text{Execution Time} = \frac{\text{Clock Cycles for Program} \times \text{CycleTime}}{	ext{Clock Rate Assumed}}
\]

\[
\text{Total Clock Cycles} = \sum CPI_i \times C_i \quad (i \rightarrow 1 \text{ to } n)
\]

Total no of Clock Cycles required to implement SPIN Protocol = 357

\[
\text{Clock Rate Assumed} = 50\text{MHz} = 20\text{ns}
\]

\[
\text{Total No of Instruction in the program} = 77
\]

\[
\text{Execution Time} = 357 \times 20\mu\text{sec} = 7.14\mu\text{sec}
\]

\[
\text{MIPS} = \frac{77}{7.14\mu\text{sec}} \times 10^6 = 10.78 \text{ MIPS}
\]

6. Conclusion
In this paper, we presented a concise soft core processor designed for wireless sensor networks with enough processing power & capabilities to perform basic image capturing and processing. The processor was tested by executing part of a SPIN communication protocols. Although the developed core was intended to support images capturing and perform simple image processing, this core can be ported easily to support the design of sensor network node that required to support other applications.

7. References
Figure 1: The 16-bit Soft Core Processor Architecture

Advertise: Branch address (BAI)

XOR R3, R3, R3 --- Clear contents of register R3
LWI R2, R5, (16 bit Addr of data loc) --- Get Advertised Metadata to compare & determine whether the node has a copy of the image advertised
LW R6, R5, R2 --- Get Metadata of the image in memory to compare with the Metadata received
BE R5, R2 (Link) --- if Advertised image is present then the message is ignored
ANDI R4, R5("FF00") --- Extract source node address into register R5
OR R5, R1, R2 --- Combine Node address + Request Message (R1 contains 1)
SWT R3, R2, (16 bit Addr of control loc) --- Load contents of register R2 into the Control location
SWT R3, R1(16 bit Addr) --- save ("0001") in the location polled by the communication unit. "0001" indicates the Communication unit that the data is available to broadcast
(Link) LW R3, R0, R5 --- Load Link Address
JR R5 --- Jump to ISR

Figure 2
Part of the SPIN Assembly code implementation. The code checks to see if the Meta data affixed to the ADV (Advertise) message is present in the memory or not

Figure 3: Chip layout on Xilinx Spartan 3A