Chapter 23


Vyasa Sai, Ajay Ogirala, and Marlin H. Mickle

Contents

23.1 Introduction ............................................................................. 515
23.2 Low-Power Circuit Techniques in WSNs ............................................... 517
23.3 Novel Low-Power Data-Driven Coding Paradigm ...................................... 518
  23.3.1 Pulse Width Coding Scheme ................................................... 518
  23.3.2 Data-Driven Decoder Architecture ............................................. 520
23.4 Distributed Architecture Design for a WPSN Node Processor ......................... 521
  23.4.1 Exploring the 8051 Microcontroller and Its ISA for WPSN Applications ..... 523
23.5 Data-Driven Architecture Design Flow Methodology .................................. 525
23.6 Conclusion ............................................................................... 526
References ....................................................................................... 526

23.1 Introduction

Conventional wireless sensor networks (WSNs) are generally made up of a set of autonomous multifunctional sensor nodes distributed over a specific environment. These sensor nodes are used to collect environment data and transfer these data to the user through the network that can include Internet segments. Besides collecting data, a node may also need to perform computations on the measured data. In general, deployment of conventional sensor networks for environmental monitoring is mainly limited due to the active life span of the onboard non-rechargeable power source. As the sensors are battery powered, it becomes difficult to periodically monitor the manual
replacement of the batteries. There have been a lot of research efforts in the direction of prolonging the limited lifetime of WSNs through efficient circuit, architecture, and communication techniques [1, 2]. In summary, the use of a WSN system is strictly limited by the battery life of the sensor nodes [36]. There is a need for a new sensor network paradigm that is not based on an enhanced lifetime of a conventional WSN but is developed for a network that is free of any battery constraints.

A wireless passive sensor network (WPSN) is a non-disposable and a cost-efficient system that operates based on the incoming received power [3–7]. This system is considered to be an efficient and a novel solution for energy problems in WSN [3]. The concept to remotely feed a sensor node on the power from an external radio frequency (RF) source has led to the emergence of the WPSNs. This concept was first introduced to power a passive RF identification (RFID) tag. It is well known that passive RFID design blocks form the basis for passive sensor node (PSN) architectures [8]. PSN operating frequencies fall under the same industrial, scientific and medical (ISM) frequency bands as most RFID applications. The latest trend in environmental monitoring application is to have sensor nodes operating at power levels low enough to enable the use of energy harvesting techniques [9, 10]. This facilitates the deployed system, in theory, for continuous sensing for a considerable extended period of time reducing recurring costs.

Building blocks of typical wireless PSN architecture consist of a sensing unit, communication unit, a processing unit, and a power source as shown in Figure 23.1 [4, 5]. The sensing unit in most cases consists of a sensor (s) and an analog-to-digital converter (ADC) as components. A sensor is a device generally used to measure some physical quantity such as temperature, light, etc. The ADC is used to convert the received analog data signal into a digital signal so as to be processed by the microcontroller. The processing unit consists of a low-power microcontroller and a storage block. The microcontroller processes data and controls and coordinates other component functionalities. The communication unit consists of an RF transceiver module that transmits and receives data to/from other devices connected to the wireless network. The power unit mainly delivers the RF–DC converted power to the rest of node units and also stores additional power based on availability.

The major differences in the architectures of a conventional WSN node and a WPSN node are the hardware of the power unit and the transceiver [4]. The power unit of the conventional WSN generally consists of a battery along with a support block called the power generator. The power unit for a WPSN node is basically an RF-to-DC converter–capacitor network. The converted DC power is used to wake up and operate the node or is kept in a charge capacitor for future usage.

---

**Figure 23.1** General WPSN node architecture.
A short-range RF transceiver, typically a major power-consuming unit on the node, is used in a conventional WSN as compared to a much simpler transceiver for modulated backscattering in the WPSN node [4,11].

To minimize the power consumed at the sensor node, the simplest of all solutions is to eliminate any or at least most signal processing at the node by transferring sampled data from all nodes to a central server. But there exist several applications in real-time control and analysis of continuous data sampling processes, where real-time signal processing and conditioning are implemented on discrete time sampled data. In many cases, the sensor data at each node must be preprocessed or conditioned before it can be handled by the processor. In such a scenario, where each sensor node must include a processor, there is need for application based dedicated processor hardware implementations that improve power efficiency and allow fine-grained design optimization. This chapter introduces a low-power conceptual design for a distributed architecture of a single passive sensor processor.

Typically, the smaller the area of the processor used in WPSN node architecture, the lower is the price. Using small-area processors, which require minimum power to operate, to provide greater read distances is significant in this scenario. WPSN being an emerging research area, there is little documentation on all the power-efficient scenarios applicable to passive sensor devices. In Refs. [3,4,6,7] efficient antenna designs, low-power transceivers were introduced for WPSNs. Not only is it important to have energy-efficient front-end and power unit designs, but there is also a need to have low-power novel processor designs that allow greater ranges for WPSN nodes. This chapter forms the basis for the low-power passive distributed sensor node architecture providing an increased operating range of the passive device.

Consider an intelligent sensor network topology with a sink node in the center communicating with several nodes around it. The server and a single sensor combination can be viewed as a single instruction single data (SISD) processor or the intelligent sensor network as a whole can be viewed as a single instruction multiple data (SIMD) processor. The power consumption of such an SIMD system depends on the hardware complexity of the passive node processor, which in turn depends on the instruction set (IS) supported by the architecture.

An intelligent combination of circuit techniques, applications, and architecture support is required to build a low-power sensor node system. This chapter introduces and elaborates the key concepts of the low-power design of the WPSN node processor. Using the 8051-ISA as an example for the distributed design concept, application-based customization of the sensor processor architecture is also elucidated in the later sections.

### 23.2 Low-Power Circuit Techniques in WSNs

Energy has become a critical aspect in the design of modern wireless devices and especially in WSNs. There is a need for a new architecture that takes into account such factors especially for passive or battery-operated device applications. Energy is defined in general as the sum of switching energy plus the leakage current energy.

The energy consumption equation is given as follows [1,12]:

\[
E_{\text{total}} = V_{\text{dd}}(\alpha C_{\text{sw}} V_{\text{dd}} + I_{\text{leakage}} \Delta t_{\text{op}}).
\]  

(23.1)

Switching activity for 1 s is represented by \(\alpha\) and the amount of time required to complete an operation is denoted by \(\Delta t_{\text{op}}\) as in the Equation 23.1. \(V_{\text{dd}}, I_{\text{leakage}},\) and \(C_{\text{sw}}\) shown in Equation 23.1 represent the supply voltage, leakage current, and switching capacitance, respectively.
Conventional WSNs employ a variety of low-power design techniques, and a short overview of these approaches is presented in the following paragraphs. The following circuit techniques are most commonly classified under the following categories, which are used to minimize the power consumption in sensor networks [1].

Asynchronous designs are increasingly becoming an integral part of numerous WSNs [13–16] due to their low-power advantages. These designs are characterized by the absence of any globally periodic signals that act as a clock. In other words, these designs do not use any explicit clock circuit and hence wait for specific signals that indicate completion of an operation before they go on to execute the next operation. Low-power consumption, no clock distribution, fewer global timing issues, no clock skew problems, higher operating speed, etc., are advantages of asynchronous designs over synchronous designs.

Power supply gating is also a low-power circuit technique widely used to reduce the subthreshold leakage current of the system [17]. This process allows unused blocks in the system to be powered down in order to reduce the leakage current. This technique was used in the Harvard sensor network system [18].

A subthreshold operation technique allows supply voltages \( V_{dd} \) lower than threshold voltages \( V_{th} \) to be used for lowering the active power consumption. This technique was first used in the complete processor design for WSNs from University of Michigan [19–21].

The aforementioned techniques can also be extended to the WPSN based on the requirements of the application and the power available to a sensor node. The focus of this chapter will be on low-power solutions to wireless passive distributed sensor node architectures. The novel low-power techniques described in the following sections are applicable not only to WPSN but also to RFID systems and RFID sensor networks (RSNs).

23.3 Novel Low-Power Data-Driven Coding Paradigm

Wireless digital transmission systems are known to use different data encoding techniques especially the variable pulse width encoding and Manchester encoding techniques. Many RF applications, such as RFID passive tags, RSNs, sensors, serial receivers, etc., use this type of encoding. Most well-known receiver decoder designs use an explicit clock to decode (Manchester or Pulse Interval) encoded data. On receiving encoded data from the transmitter, the clock is extracted from it. A classical decoding process for pulse width modulated signals is by oversampling with a clock [22,23]. The received signal is sampled at a much higher bit rate clock than the received signal in order to decode it as shown in Figure 23.2. Symbol-1 or symbol-0 of the received encoded data stream can be identified by counting the number of clock pulses within each symbol as shown in Figure 23.2 (2 for symbol-0 and 4 for symbol-1). The well-known architecture of this classical decoding scheme is shown in Figure 23.3. This architecture basically consists of high-frequency oscillator, fast-clocked counter, and a comparator. The major disadvantage of using high clock rate driven decoders is the significant increase in the power consumption at the receiver side [23–26].

23.3.1 Pulse Width Coding Scheme

A novel explicit clock-less coding scheme for communication receivers is shown in Figure 23.4 for reducing the power consumption on the receiver side [24,27,37]. Pulse width coding (PWC) data shown in Figure 23.4 represent the encoded input demodulated serial data as “01100110.” In Figure 23.4, the PWC data signal is sampled at every rising edge of the delayed version of
the PWC data signal in order to differentiate “1” and “0” for completing the decoding process. We can clearly see from Figure 23.2 that the decoded output bit is “1” whenever both the signals are high; otherwise it is a “0.” The most important power parameter in this decoding scheme is the delay ($\Delta$). The minimum possible delay required is about $PW_0$ and the maximum delay required is less than $PW_1$. In other words, for a successful PWC decoding, we need delay ($\Delta$) to satisfy the condition: $PW_0 < \Delta < PW_1$. The decoding mechanism proposed in Ref. [24] for this scheme is an extremely simple, low-power, and clock-less circuit realized using Complementary metal-oxide-semiconductor (CMOS) digital chip design techniques.

The PWC scheme can be implemented and easily integrated to other well-known synchronous and asynchronous design variants that have high-power-consuming decoder modules in serial data...
communication receivers. A well-known direct application is in the symbol decoding process of the passive RFID tag and RSN node systems while the encoding remains unchanged at the transmitter.

The delay generally in hardware translates to a buffer element. A buffer element is generally built using even number of inverters. Using an optimized library, there is a possibility to further optimize the design schematic generated and lower the power values for inverters that are used to interpret large delay values. Another alternative to designing low-power inverters is to individually model them based on the choice of parameters such as width, length, and target technology of the metal-oxide-semiconductor (MOS) layout designs [29]. The inverter can be designed from the transistor level using the CAD (computer-aided design) layout tools. This would also give the designer the flexibility to alter the width of individual P-type metal-oxide-semiconductor (PMOS) and N-type metal-oxide-semiconductor (NMOS) transistors to generate the necessary delays within the circuit [30,31] conforming to the low-power requirements.

23.3.2 Data-Driven Decoder Architecture

The PWC scheme introduced in the previous Section can be realized using the explicit clock-less architecture shown in Figure 23.5. This data-driven architecture does not use any explicit clock to drive its components such as the shift register and the comparator. The delayed encoded input acts as a clock to trigger the components of the decoder thus eliminating the need for any explicit clock. This architecture was successfully simulated using the standard CAD tools as a low-power and low-area data-driven decoder [24]. In Ref. [24], it has been reported that the post-layout power consumption of the data-driven chip was about one-fourth the power consumed by the conventional decoder design for the same data rate of 40 kHz. The cell area of the data-driven decoder is 69% smaller than that of the conventional design. Elimination of the high-frequency oscillator, fast-clocked counter, and an explicit clock has contributed to this significant reduction in both power consumption and the area occupied by the data-driven chip.

An example passive RFID post-layout CMOS design was successfully simulated to operate at very low power using a custom low-power asynchronous computer [28]. This design uses the data-driven decoder with an integrated counter to it as the major low-power component of the entire architecture. The power consumption of the post-layout simulation results of both the synchronous/asynchronous RFID designs is illustrated in Figure 23.6a for different data transmission rates. The switching power consumption comparisons for different data rates are also illustrated in Figure 23.6b. There is a consistent linear increase in switching power for the synchronous design when compared to the asynchronous design as the data rate increases. In other words, the switching power of the asynchronous design is lower when compared to the synchronous design at each data rate. These results are very encouraging especially at the typical data rate corroborating the

---

Figure 23.5  Novel data-driven decoder architecture.
concept that a passive RFID tag can be designed using an asynchronous design to significantly reduce power requirements and thereby increasing its read range. The same concept can also be applied to sensor node architecture especially for the case where the data-driven decoder can be integrated with a low-power processing unit. The low-power processing unit design concept will be discussed in the next section.

23.4 Distributed Architecture Design for a WPSN Node Processor

Microcontroller design choice for a sensor node leads to a trade-off between speed and energy efficiency. In most cases, power constraints dominate, which in turn leads to significant computational constraints. In general, a microcontroller consists of a controller, volatile memory for data storage, ROM/EPROM/EPRPM, parallel I/O interfaces, clock generator, serial communication interfaces, etc. In Ref. [6], a general-purpose low-power 16 bit programmable microcontroller (MSP430F2132) is used for managing the entire node. The microcontroller design can be
Intelligent Sensor Networks

tailor-made for applications to further reduce the power requirements at the node. A significant contribution toward achieving a low-power sensor node processor is introduced in this chapter that highlights customizing a processor based on its subset instruction set architecture (ISA) for the specific target application [27].

SIMD is a well-known class of parallel computers in Flynn’s taxonomy. SIMDs have the ability to perform the same operation on multiple data simultaneously for processors with multiple processing units. The need for synchronization between processors is not required. The proposed architecture presents a PSN(s) that can be replicated to produce an SIMD architecture. The passive units are powered and controlled by RF energy that enables convenient reconfiguration due to the ability to address nodes individually or in groups that can be simply and conveniently changed using RF communications. Thus, bits within the passive node processors can be set to perform or ignore commands thus allowing dynamic reconfigurability of the units composing the SIMD.

Consider an intelligent sensor network topology with a base station (sink node) in the center communicating with several wireless PSNs around it as shown in Figure 23.7 [27]. In a WPSN, a PSN is passive and is powered by the impinging RF wave, which is also used for communication, from a sink node. The major change with respect to architecture of a PSN is only the processing unit as shown in Figure 23.1. The sink node acts as the RF source assumed to have unlimited power that feeds the PSN with RF power. The sink node transmits RF power to the randomly deployed PSN nodes for processing, sensing, and data collection activities. This sink node wirelessly transmits commands to the PSN that executes these commands and responds back to the sink node. A PSN can be implemented as a CMOS chip that provides logic to respond to commands from a sink node. Thus, the sink node and the PSN combination can be viewed as a complete processor or as multiple processing units [38]. This will form the basis of our distributed concept that will be introduced in the following paragraphs.

The sink node (Control and Memory [C&M]) is an RF equipped control and storage base station, and the PSN processor is an execution unit with minimal storage capacity (e.g., registers) as shown in Figure 23.8. The sink node is allowed the flexibility to be a classical von Neumann–or Harvard-type architecture that consists of an interrogation control unit along with a program and memory units. Commands will be stored on the powered sink node that transmits the commands wirelessly to the PSN. The intent is to keep the PSN processor as simple as possible so as to maintain

![Figure 23.7 WPSN topology with PSN fed by an RF source–sink node.](image-url)

low-power requirements and/or extend read range from the sink node. Any unnecessary complexity on the PSN processor will be moved onto the powered sink node. This would significantly reduce the hardware on the PSN side, thus reducing the overall power consumption of the node.

Many other circuit design techniques can be applied to the sensor node architectures to reduce power consumption. One of the main power reduction techniques that can be employed in the proposed PSN processor design is to eliminate the use of an explicit clock overhead [28]. The entire circuitry of the PSN processor is to be asynchronous with the remote command execution controlled by the sink node making the system programmable and reconfigurable. The design uses a clock-less data-driven symbol decoder introduced earlier as a low-power component instead of the conventional input clocked data decoding process used at the sensor nodes [24]. Another technique commonly used to reduce power consumption is scaling down the supply voltage of the system or part of the system [1]. Any combination of these energy-efficient techniques can be used in addition to the distributed architecture concept based on application-specific requirements.

### 23.4.1 Exploring the 8051 Microcontroller and Its ISA for WPSN Applications

The choice of the Intel 8051 (i8051) is justified by the fact that it is still one of the most popular embedded processors. Furthermore, due to its small size and low cost, it has numerous applications where power efficiency is necessary. The most commonly used 8051 microcontroller in sensor nodes will be considered as an example for exploring its ISA and its application to the proposed conceptual distributed design.

The 8051 is an 8 bit microcontroller that includes an IS of 255 operation codes. The 8051 architecture consists of five major blocks, namely, control unit, ALU, decoder, ROM, and RAM. Based on the distributed design concept introduced as shown in Figure 23.8, the WPSN node processor consists of two major blocks with respect to 8051: 8051 compatible execution unit and the minimum number of temporary storage registers required. The execution unit is mainly an 8051 ALU. The number of instruction supported by the execution unit depends on the target application.

The sink node will transmit the program instructions to the WPSN node that executes these instructions and returns the results back to the sink node. The WPSN node, for example, will have the capabilities to perform functions like OR, XOR, AND, ADD, etc., that are compatible with 8051 depending on the application. This sink node and the WPSN node together form a complete processor. As the program to be executed by the WPSN node is stored in the sink node, the need
for program memory at the passive node is eliminated. There still may be a need for local scratch pad memory at the WPSN node although the number of bytes is drastically reduced in order to satisfy the power requirements. The WSPN node executes the instructions wirelessly as issued by the sink node.

Figure 23.9 represents a high-level sequence diagram for an ADD operation. Let us consider an ADD operation: ADD A, R1 \((A = A + R1)\), where R1 denotes one of the eight \((R0–R7)\) 8 bit 8051 working registers for a selected register bank and A denotes the 8-bit accumulator register. The sink node sends out the R1 values to load and store it in the temporary storage on the WPSN node processor unit. On receiving the ADD instruction, the passive node processor’s execution unit performs the addition operation on the already existing value in the accumulator and the new R1 value. The computed result on the accumulator register is sent back to the sink node. The sink node will contain main memory that acts as the major storage area for the majority of data items.

Sensor applications require special-purpose hardware suitable to cater to a different set of requirements. Characteristics of the target applications and the utility of the sensors make it important to choose applicable hardware for sensor networks on a case-by-case basis. The power requirements of a WPSN limit the requirements needed for different applications used. Some of the well-known basic core algorithms form a class of simple applications such as the sum-array (sum of all values in a list), Top10 (finds top 10 values in a list), majority consensus (finds the majority values in a list), min-max finder (finds minimum and maximum values in a list), Binary search (typical search algorithm for a sorted list), Matrix Multiplication (matrix multiplication for small size matrices), etc. [32].

Generally, sensor networks employ only data filtering at the node so that every sensor sample need not be transmitted on the radio so as not to consume all the wireless bandwidth available to the network. By transmitting only necessary sensor data readings over the radio allows saving the available stored energy on the node. Let us consider a simple application scenario, for example, using sum-array application using 8051-ISA. The amount of temporary storage and the ALU capabilities of the WPSN node processor will be chosen to maintain low-power requirements. Assume that the only temporary memory space available for the execution unit is the register R7–R0 of a selected single register bank of 8051. The major function is an ADD operation and hence the choice of the arithmetic instructions that would be part of the execution unit on the WSPN node are ADD A, Rn; ADDC A, Rn. The minimal data transfer instruction necessary would be the MOV A, Rn;
MOV Rn, A; and MOV Rn, #DATA (8 bit). The WSPN node processor will support only those features required to interface and communicate with the sink node. Therefore, the branch, comparison, load, and store instructions will be implemented on the sink node side rather than on the passive side. This ISA will be compatible with the i8051-ISA. Additional instructions can be added to enhance the capability of the execution unit and also depending on the application.

To arrive at an energy-efficient computation solution, there is always a trade-off between the communications from and computation on a sensor node. Hence the choice of a design for a sensor node architecture depends not only on the low-power technique but also on the application space.

23.5 Data-Driven Architecture Design Flow Methodology

A data-driven architecture is a design paradigm that uses no explicit clock to drive its components [24]. Either data or local signals are used to drive components of the processor. This type of an asynchronous design uses no global periodic signals to synchronize its operations. Lack of strong support of commercial CAD tools is a major hurdle for synthesis of explicit clock-less designs. Asynchronous (very-high-speed integrated circuits) hardware description language (VHDL) designs generally are known to use non-synthesizable delay constructs such as wait, delay, etc., for their implementation in the absence of a global periodic signal for synchronization. Standard VHDL compilers are not known to synthesize VHDL code that implements an asynchronous design. Based on the conventional hardware descriptor languages, most asynchronous design methodologies [33–35] that have been proposed are not accessible to standard high-level design tools. The high-level data-driven design flow will be described in this section that requires minimum changes to a traditional synchronous flow [24,27,39].

**Step 1:** The data-driven design is first written in VHDL along with the necessary non-synthesizable delay constructs. This VHDL design is then simulated using Mentor Graphic’s ModelSim. A customized test bench is used to verify the correct functionality of the design in ModelSim.

**Step 2:** A synthesized netlist is generated for the data-driven design using the Synopsys Design Compiler upon the successful verification of the VHDL design in step 1.

The “dc_shell” command interface provides a script execution environment based on Tool Command Language (TCL). The basic directives of a TCL script include setup environment variables, constraints, basic compilation directives, etc. The major modification will be to eliminate any clock in the script. All the statements that involve the non-synthesizable VHDL delay constructs are identified and removed. Synthesizable delay commands need to be separately inserted into the TCL script during the synthesis process. The available delay commands are set_max_delay and set_min delay. These commands have several options, generally needs a -from set of start points, -to set of end points along with a fixed target delay value. The start and end points refer to specific cells and their corresponding input and output pins in the schematic of the design. Identification of the necessary start/end points in the design is the key for the accurate working of the design as insertion of these delay changes the timing graph of the design.

The next step is to compile, simulate, and verify the generated design netlist Verilog file along with the target technology library using ModelSim.

**Step 3:** After the successful post-synthesis verification, the design layout along with the post place-and-route netlist is generated using a VLSI Cadence layout tool known as Cadence Encounter. The post place-and-route design netlist is simulated and verified for the expected operation using ModelSim along with a delay file format. After the successful post-layout simulation of the netlist, the layout verification of the data-driven design is said to be complete.
The final step is to estimate the power consumed by the data-driven design using Cadence Encounter. A switching activity file is generated from the initial test bench using ModelSim. During the final stage of the place-and-route process, the activity file is used in the power rail analysis option available with Encounter to produce the power report.

The advantage of using data-driven designs is not only for achieving low power but also for allowing flexibility to implement explicit clock-less designs using standard CAD tools. The data-driven symbol decoder introduced in Ref. [24] was implemented using this methodology. This methodology can also be applied to the distributed WPSN 8051-node architecture when implemented as a data-driven design. The design flow used for data-driven designs can also be extended to a variety of design paradigms such as synchronous, asynchronous, globally asynchronous locally synchronous, and globally synchronous locally asynchronous [24].

23.6 Conclusion

Conventional WSNs are a disposable system as they are dependent on the limited lifetime of their batteries. WPSN system of passive nodes does not face this problem as they are remotely powered by an RF source, but do have limited ranges. This chapter discusses novel low-power solutions to increase the range of WPSN nodes. This chapter illustrates the importance of the data-driven architecture using a novel clock-less symbol decoder architecture and its low-power applications that include synchronous and asynchronous design variants that have high-power-consuming modules in communication receivers. A detailed power analysis of the post-layout simulation power results of the data-driven symbol decoder and the conventional clocked symbol decoder for passive RF receiver systems has been analyzed. This chapter introduces the elements and concepts of the design of a WPSN node processor as a distributed architecture that operates remotely and wirelessly from the sink node. A high-level asynchronous design flow that can be used to implement the data-driven design using synchronous CAD tools is also discussed. This design flow will provide the reader with sufficient guidelines to design and implement application-specific data-driven processor architectures.

This research has the potential to realize WPSN node applications for environmental, structural, and medical fields especially while providing the basis for a programmable, reconfigurable, and a low-power passive processing unit for distributed computing.

Currently, our research work is focused on developing a low-power distributed 8051-SIMD architecture as a clock-less design based on the concepts described in this chapter.

References


Intelligent Sensor Networks


27. V. Sai, A low power distributed architecture: A passive remote execution processor, PhD dissertation (in process), unpublished.


