IPASJ International Journal of Electrical Engineering (IIJEE)

A Publisher for Research Motivation........

Web Site: http://www.ipasj.org/IIJEE/IIJEE.htm

Email: editoriijee@ipasj.org

Volume 5, Issue 2, February 2017

ISSN 2321-600X

TRADITION SINGLE-PURPOSE EVENT PROCESSOR DESIGN: FOR LOW POWER WSN NODE

Mr. Bhavesh Pillay

PDM University, Bahadurgarh

ABSTRACT

A basic processor consists of a controller and a data path. The datapath stores and manipulates a system's data. The information within the case of WSN node is that the data noninheritable by the sensors of node when measurement the close conditions to be hold on, processed and compressed for transmission; controller is capable of moving data through datapath. The combinational and serial logic style techniques ar applied to build a controller and datapath for a customized single-purpose event processor for WSN Node. Such custom designed single-purpose event processor leads to several advantages in terms of quicker performance, smaller size lower power consumption. WSN Node consists of sensors, a radio circuit a processor/microcontroller and some variety of power source. Event processor is custom designed, simulated, tested and implemented for single purpose application i.e. WSN Node using VHDL. Active HDL is used for simulation; to look at the varied trade offs of victimization general purpose versus single purpose processor to implement necessary WSN node practicality.

I. INTRODUCTION

Wireless Sensor Networks (WSN) is speedily turning into well-liked and finding use in several areas of day to day life. The sensor nodes ar ready to monitor a large type of close conditions that embody the following: flow, temperature, pressure, humidity, moisture, noise levels, mechanical stress, speed, etc. There are various areas for the readying of wireless sensors: physiological watching, habitat watching, precision agriculture, forest fire detection, nuclear, chemical, biological attack detection, military, transportation, disaster relief, and environmental monitoring (air, water, and soil chemistry) etc. The basic components of a detector node ar shown in figure one. A typical detector node consist of a sensing unit having sensor for sensing close conditions, a process unit having a processor for processing collected information and a transmission unit having transceiver for communication with base. Other units relying upon application ar position finding system and a mobilizer all these units high-powered by an influence unit generally connected to an influence generator. Since maximum network life is desired and the quantity of energy accessible for every node is finite (nodes ar generally battery powered), low power implementations are a needed for longer lifetime of WSN node. In the proposed style a custom single purpose event processor is meant like to satisfy the particular necessities of the process unit of atypical processor node. Designing is done by decisive the systems design for WSN Node, and then mapping the functionality through activity and structural description thereto design. Designing a custom designed single-purpose event processor could end in increase in NRE price and time to promote in addition because it can end in reduction in power consumption, range of electronic transistor on chip and number of clock cycles (increasing speed).

II. MOTIVATION AND GOALS

Proposed design replaces most of the practicality of a general purpose microcontroller with associate event-driven system specifically optimized for watching applications in associate agricultural farm. A summary of style goals for the system design ar given below and elaborate discussions of goals and however the design meets these goals follows within the following section.

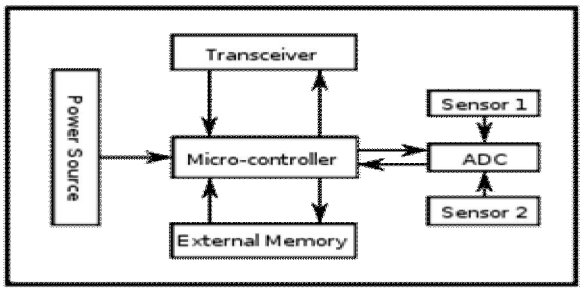

Fig:-1 Components of a Sensor Node

1. Event-Driven Computation: Eliminate unnecessary event-processing overhead with event-driven hardware system design.
2. Hardware Acceleration to Improve Performance and Power: Build a system composed of several elements that ar optimized for specific tasks.
3. Exploiting Regularity of Operations among associate Application: Optimize the common-case behavior within associate application that is special purpose condescend for single application (WSN node).
4. Optimization for a specific category of Applications: Optimize the common-case behavior of watching applications to scale back power, while still providing all-purpose process capability to change broad practicality.
5. Modularity: Provide associate simply protrusible system design that permits totally different sets of hardware elements to be combined into a bigger system targeting a specific application.
6. Fine-grain Power Management Based on process Requirements: offer express programmer- accessible commands for fine-grain resource and power management.

III. ARCHITECTURE DESCRIPTION

To fulfill design goals, the basic functionality of a all-purpose microcontroller is replaced with a modularized, event-driven system. The system architecture is illustrated in Figure two. There are 2 distinct divisions among the system in terms of the positions of the elements with respect to the system bus. We refer to the elements to the correct of the bus as slave elements and people to the left as master elements except the memory, which is a slave. The system bus has three divisions – information, interrupt, and power control. The slaves compete for the interrupt bus victimization centralized arbitration if a lot of than one slave has associate interrupt to signal. The slaves also respond to scan or write requests from the master facet on the info bus, thus permitting the masters to scan their data content and management their execution.

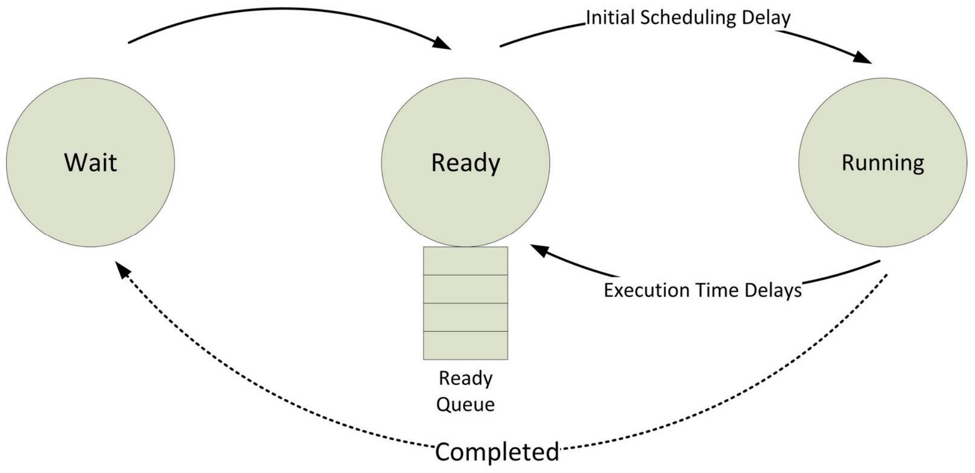

Figure 2 The system architecture

3.1 E v e n t -Driven System we tend to propose associate event-driven system in that all of the master elements ar involved event handling, and the slaves assist the master elements in their tasks and signal the incidence of events to trigger the master components. All external events, such as the start of radio packet reception, are expressed as interrupts by associate acceptable slave part. The slave components additionally raise interrupts for their internal events, such as completion of an assigned task. To the master components, there is no distinction between external and internal events. Also, since the occurrence of all events is signaled by interrupts, we can use the terms event and interrupt interchangeably. The system idles until one of the slaves signals the presence of an occurrence, and when all outstanding events have been processed, the system returns to its idle mode. Since all the system does is respond to events, there is no software overhead for interrupt handling.

3.2 H a r d w a r e Acceleration to Improve Performance and Power there's a all-purpose microcontroller in system designed. However, unlike different detector network device architectures, the intent of design is for the microcontroller to be the last resort for any computation, i.e. the microcontroller should be known as upon to perform a task solely if the remainder of the system doesn't have the requisite practicality. Specific tasks that ar thought of common to a wide type of application are offloaded to hardware accelerators, which will be a lot of power and cycle economical than the all-purpose microcontroller. Hence, the microcontroller can sometimes be high-powered down by gating the provide voltage. This not only reduces active power however additionally outflow, which will be a really vital supply of power consumption for low duty cycle operation. In the absence of a hardware timer, a software timer would have to be enforced within the microcontroller, requiring the microcontroller to always be active. There is a generic filter slave for basic processing. In our architecture, this block is a simple threshold filter with a programmable threshold. We additionally implement a message processor block to offload packet process and avoid wakening the microcontroller for common events like packet forwarding and sending packets of collected samples. The slave elements additionally embody essential detector network device components such as the radio, and a block of sensors and Analog-to- Digital Converters (ADCs).

3.3 Exploiting Regularity of Operations among associate Application All immediate interrupt handling is offloaded to the event processor block whereas the microcontroller is high-powered down. The event processor is a simple state machine which will be programmed to handle associate interrupt by transferring information blocks between the slave devices and putting in place management data for these devices to complete their tasks.

A regular event is one which will be processed wholly by the event processor and therefore the slave elements. An irregular event is one that needs the microcontroller. One of the tasks involved in mapping associate application to our system is to see the partitioning of events into regular and irregular events. The regularity of an event is decided by the practicality gift within the slaves and therefore the event processor. For a typical application, events such as sampling, transmitting samples, and forwarding packets would ideally be regular while application or network re-configurations would usually be classified as irregular.

IV. SYSTEM COMPONENTS

Details of other system elements ar represented in following section:

4.1 System Bus As mentioned earlier, the system bus comprises the information bus, the interrupt bus, and power control lines. The data bus has address, data, and control signals indicating scan and write operations. In current implementation the address bus has 16 lines, the data bus has eight lines, and there is one control signal every for scan and write operations. The address space for memory-mapped design is so 64K. The address and control lines will be driven solely by the event processor and therefore the microcontroller in mutual exclusion as determined by the bus arbiter, which is presently simply a mux. The data lines ar driven by the slave that determines that this request lies in its address vary, and are demultiplexed to the mastermind of the request, i.e., the event processor or the microcontroller.
The interrupt bus has 6 address lines and management signals for arbitrating the writing of interrupts by slaves. The system is therefore capable of handling sixty four interrupts in the current model. The event processor has control signals in the interrupt bus to point once it's scan this interrupt address. The power control lines ar handshaking pairs for every slave or memory phase controlled. The handshake is relevant only if a part is turned on, to determine the time once the part may be used. The system currently makes no assumptions regarding the time taken to wake up for the elements over that express power management is exercised.

4.2 Microcontroller The microcontroller is used to handle irregular events, as discussed in previous sections, such as system initialization and reprogramming. The microcontroller is a simple nonpipelined microcontroller. It implements an 8-bit Instruction Set design (ISA). The system uses available process cores with necessary modifications for low-power options needed for the system.

4.3 E v e n t Processor The event processor is basically a programmable state machine designed to perform the repetitive task of interrupt handling. Figure 3 illustrates a simplified version of the actual state machine within the event methodor The event processor idles within the prepared state till there's associate interrupt to process. When associate interrupt is signaled, the event processor transitions to the LOOKUP state if the information bus is on the market, i.e., the microcontroller is not awake. If not, the event processor transitions to the WAIT BUS state and waits until the microcontroller relinquishes the information bus. In the LOOKUP state, the event processor looks up the ISR address corresponding to the interrupt. The lookup table is hold on in memory, and the starting location of the table, offset by an quantity proportional to the interrupt address, contains the address of the event processor ISR. When the search is complete, the event processor transitions to the FETCH state, in which the primary instruction at the ISR address discovered within the search state is fetched. The event processor stays in the FETCH state until all the words of this instruction are fetched, and then it transitions to the EXECUTE state. The instructions among associate event processor ISR may be one amongst the subsequent – SWITCHON, SWITCHOFF, READ, WRITE, WRITEI, TRANSFER, TERMINATE, or WAKEUP. Table 1 provides a outline of the operations corresponding to the directions. The event processor has one register used to store temporary data. The op- codes are every three bits and the directions vary within the range of words they span.

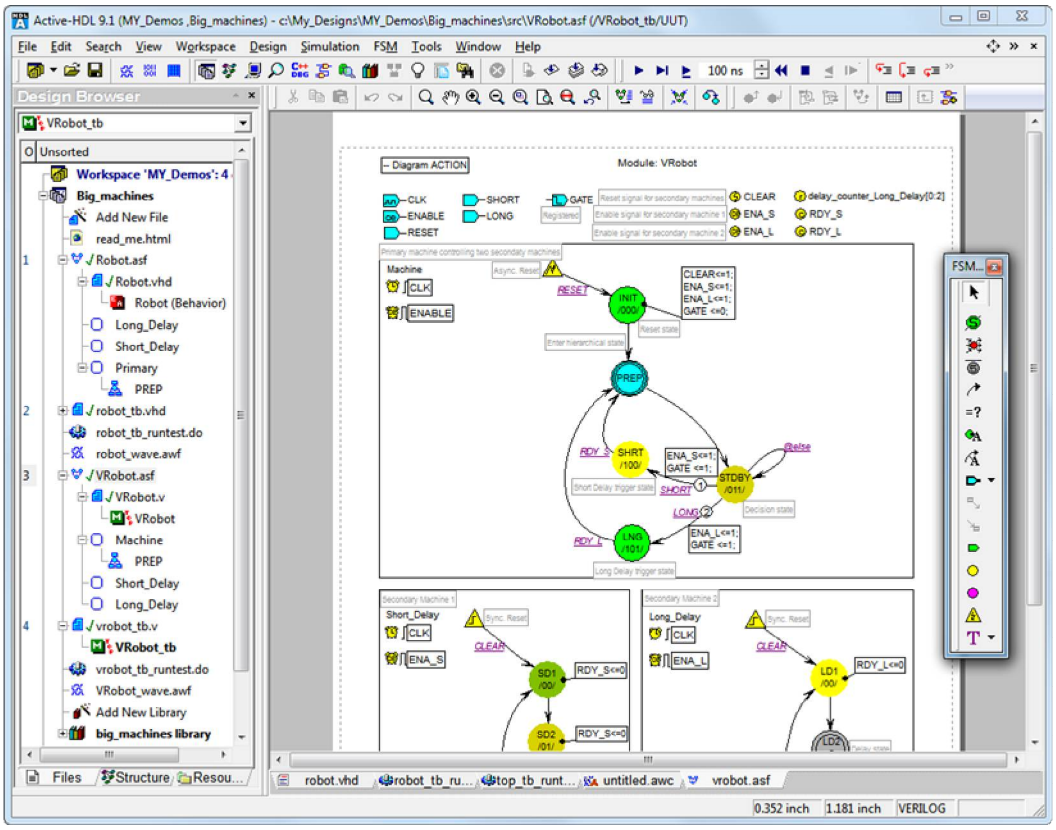

Figure 3 Event processor state machine diagram

V. SIMULATION SETUP

A synthesizable implementation of all the blocks of the figure 3 can be captured at the register transfer level (RTL) and written in VHDL, and integrated in to a simple system. The controller fetches instruction from its read solely program memory and decodes them victimization the decoder elements. The ALU component is used to really execute operation. The source and destination of these operations ar registers that reside within the internal RAM of the processor. A VHDL simulator / compiler Active alpha-lipoprotein is used to compile VHDL program. The read-only storage is generated victimization the special in build module in Active alpha-lipoprotein that outputs the VHDL description of the desired specification of the ROM. Now all the elements of the system ar prepared to be connected along to form WSN Node. Now all the software system modules ar compiled and coupled to acquire VHDL possible program, which ar simulated, tested and finally converted into gate level netlist. The simulation is done using Active alpha-lipoprotein, VHDL simulator software system. A design flow diagram of Active alpha-lipoprotein is shown in figure four.

REFERENCES

[1]. Beutel, Jan. Geolocation in a PicoRadio Environment. Masters Thesis. 2000, 9.
[2]. Cetintemel, Ugar, Flinders, Andrew, and Sun, Ye. Power-Efficient Data Dissemination in Wireless detector Networks. International Workshop on information Engineering for Wireless and Mobile Access Proceedings of the third ACM international workshop on Data engineering for wireless and mobile access. 2003, 1-8. [3] Chen, Wei-Peng and Sha, Lui. An Energy –Aware Data-Centric Generic Utility based mostly Approach in Wireless detector Networks. Information process In detector Networks Proceedings of the third international conference on scientific discipline in detector networks. 2004, 215-224.
[3]. Dousse, Olvier, Mannersalo, Petteri, and Thiran, Patrick. Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms. International Symposium on Mobile Ad Hoc Networking &amp; Computing Proceedings of the fifth ACM international conference on Mobile unintended networking and computing. 2004, 109-1200.
[4]. Hill, Jason, Horton, Mike, Kling, Ralph, and Krishnamurthy, Lakshman. The Platforms Enabling Wireless detector Networks. Communications of the ACM June 2004/ Vol47. No. 6. ,41-46.
[5]. Polastre, Joseph, Hill, Jason, and Culler, David. Versatile Low Power Media Access for Wireless Sensor Networks. Conference On Embedded Networked Sensor Systems Proceedings of the 2d international conference on Embedded networked detector systems. 2004, 95-107.
[6]. Rabaey, Jan M., Ammer, M. Josie, Silva, Julio L. da, Jr., Patel, Danny, Roundry, Shad. PicoRadio Supports AdHoc Ultra-Low Power Wireless Networking. Computer Magazine (July 2000), 42-44.
[7]. Savarese, Chris. Robust Positioning Algorithms for Distributed Ad-Hoc Wireless detector Networks. Masters Thesis. 2002, entire thesis.
[8]. Savarese, Chris, Rabaey, Jan M., Beutel, Jan. Locationing in Distributed Ad-hoc Wireless Sensor Networks. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). 2001, 2037-2040.
[9]. Seada, Karim, Zuniga, Marco, Helmy, Ahmed, and Krishnamachari, Bhaskar. Energy-Efficient Forwarding Strategies for Geographic Routing in lossy Wireless detector Networks. Conference On Embedded Networked Sensor Systems. Proceedings of the 2nd international conference on Embedded networked detector systems. 2004, 108-121.