International Journal of Engineering In Advanced Research Science and Technology ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:79829-35

LOW-POWER DIGITAL SIGNAL PROCESSOR ARCHITECTURE FOR WIRELESS SENSOR NODES

1.M.NAGA RAJU,2. K.SRINIVASA RAO
1. PG Scholar, Dept of ECE, Anubose Institute of Technology, Palwancha, khammam
2. Dept of ECE, Anubose Institute of Technology, Palwancha, khammam

ABSTRACT: The goal of this paper is to design low power WSN digital processor using parallel prefix technique. Wireless data acquisition, storing, performing arithmetic operations are three main key steps involved in this project. In this project, design and implementation of newly proposed folded tree architecture is presented for an efficient construction of DSP processors. Folded tree architecture has two phases. They are trunk and twig phase. Both phases are effectively utilized and designed in this project. This paper presents an overview of the key technologies required for low-energy distributed micro sensors. These include power aware computation/communication component technology, low-energy signalling and networking, system partitioning considering computation and communication trade-offs, and a power aware software infrastructure.

KEYWORDS: Micro sensors, folded tree architecture, Wireless Sensor Nodes, computation, micropower

INTRDUCTION:

The design of micropower wireless sensor systems has gained increasing importance for a variety of civil and military applications. With recent advances in MEMS technology and its associated interfaces, signal processing, and RF circuitry, the focus has shifted away from limited macrosensors communicating with base stations to creating wireless networks of communicating microsensors that aggregate complex data to provide rich, multi-dimensional pictures of the environment. While individual microsensor nodes are not as accurate as their macrosensor counterparts, the networking of a large number of nodes enables high quality sensing networks with the additional advantages of easy deployment and fault-tolerance. These characteristics that make microsensors ideal for deployment in otherwise inaccessible environments where maintenance would be inconvenient or impossible [1][2][3]. The potential for collaborative, robust networks of microsensors has attracted a great deal of research attention. The WINS [5] and PicoRadio [6] and projects, for instance, aim to integrate sensing, processing and radio communication onto a microsensor node. Current prototypes are custom circuit boards with mostly commercial, off-the-shelf components. The Smart Dust [4] project seeks a minimum-size solution to the distributed sensing problem, choosing optical communication on coin-sized "motes." The prospect of thousands of communicating nodes has sparked research into networkprotocols for information flow among microsensors, such as directed diffusion [7]. The unique operating environment and performance requirements of distributed microsensor networks require fundamentally new approaches to system design. As an example, consider the expected performance versus longevity of the microsensor node, compared with current battery-powered portable devices. The node, complete with sensors, DSP, and radio, is capable of a tremendous diversity of functionality. Throughout its lifetime, a node may be called upon to be a data gatherer, a signal processor, and a relay station. Its lifetime, however, must be on the order of months to years, since battery replacement for thousands of nodes is not an option. In contrast, much less capable devices such as cellular telephones are only expected to run for days on a single battery charge. High diversity also exists within the environment and user demands upon the sensor network. Ambient noise in the environment, the rate of event arrival, and the user's quality requirements of the data may vary considerably over time. A long node lifetime under diverse operating conditions demands power-aware system design. In a power-aware design, the node's energy consumption displays a graceful scalability in energy consumption at all levels of the system hierarchy, including the signal processing algorithms, operating system, networkprotocols, and even the integrated circuits themselves. Computation and communication are partitioned and balanced for minimum energy consumption. Software that understands the energy-quality tradeoff collaborates with hardware that scales its own energy consumption accordingly. Using the MIT

International Journal of Engineering In Advanced Research Science and Technology ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:79829-35

μAMPS project as an example, this paper surveys techniques for system-level power-awareness.

The transition from the 20th to 21st century observed the emergence of lowcost, low-power, and miniature size electronics, enabling attractive solutions for numerous new application areas to be created as well as facilitating several existing ones to improved. One such example is the development of Wireless Sensor Network (WSN), providing a low-cost alternative to both manual monitoring solutions and traditional infrastructure-based monitoring solutions, in which both the power and the data are required to be transported over a physical media, such as cables. In contrast to an infrastructure-based monitoring network, a WSN is comprised of spatially distributed sensor nodes that, in addition to sensing the environment, are capable of communicating wirelessly to transport the acquired data to a desired destination. In addition, the wireless communication in a WSN also provides the means to establish a self-organizing wireless monitoring network. A WSN finds its applications in numerous fields spanning from home automation to industrial monitoring, and to battlefield tracking etc. [1]-[7]. Some of the notable applications include environmental monitoring [8]-[10], fire detection in forests [11]-[13], structural health monitoring for buildings and bridges [14]-[15], health care monitoring [16]-[17], industrial condition monitoring [18]-[19], battlefield monitoring [20]-[22], precision agriculture [23] and logistics monitoring [24]. In comparison to infrastructure-based monitoring networks, the advantages associated with WSNs are low-cost, ability to self-organize, scalability and ease of deployment [25]-[28]. As the sensor nodes in a WSN communicate wirelessly and, the means of• energy (for example, batteries) are integrated within them, the cost associated with the development and maintenance of communication and power related infrastructure is low as compared to infrastructure-based networks. In addition, typically, sensor nodes have a simple design that leads to the low-cost development and thus, enables the realizing of costeffective WSNs.

CHARACTERISTICS OF WSN:

Several specific characteristics, unique to WSNs, need to be considered when designing data **Processor Architecture for WSNS**.

Data-Driven: WSN applications are all about sensing data in an environment and translating this into useful information for the end-user. So virtually all WSN applications are characterized by local processing of the sensed data.

Many-to-Few: Since radio transmissions are very expensive in terms of energy, they must be kept to a minimum in order to extend node lifetime. Data communication must be traded for on-the-node computation to save energy, so many sensor readings can be reduced to a few useful data values.

Application-Specific: A "one-size-fits-all" solution does not exist since a general purpose processor is far too power hungry for the sensor node's limited energy budget. ASICs, on the other hand, are more energy efficient but lack the flexibility to facilitate many different applications.

Apart from the above characteristics of WSNs, two key requirements for improving existing processing and control architectures can be identified.

Minimize Memory Access: Modern micro-controllers (MCU) are based on the principles of a divide-and-conquer strategy of ultra-fast processors on the one hand and arbitrary complex programs on the other hand. But due to this generic approach, algorithms are deemed to spend up to 40–60% of the time in accessing memory, making it a bottleneck.

Data Flow and Control Flow Principles: To manage the data stream (to/from data memory) and the instruction stream (from program memory) in the core functional unit, two approaches exist. Under control flow, the data stream is a consequence of the instruction stream, while under data flow the instruction stream is a consequence of the data stream. Traditional processor architecture is a control flow machine, with programs that execute sequentially as a stream of instructions. In contrast, a data flow program identifies the data dependencies, which enable the processor to more or less choose the order of execution. The latter approach has been hugely successful in specialized high-throughput applications, such as multimedia and graphics processing.

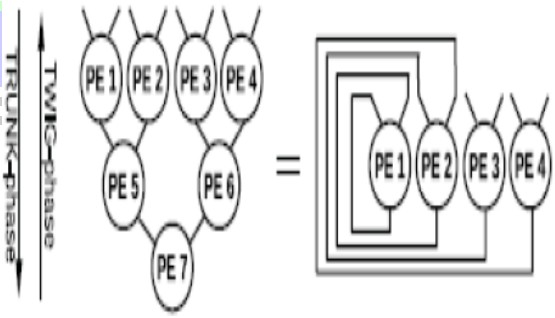


Fig2:A binary tree (left, 7 PEs) is functionally equivalent to the novel folded tree topology (right, 4 PEs) used in this architecture.

International Journal of Engineering In Advanced Research Science and Technology ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:79829-35

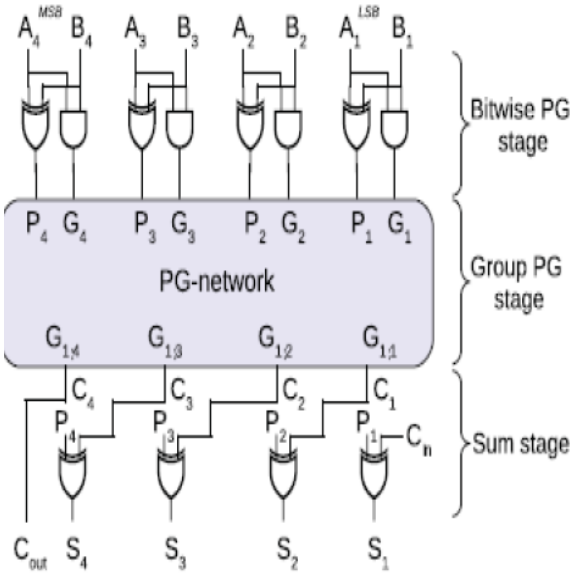


Fig3: Addition with propagate-generate (PG) logic.

ON-THE-NODE DATA AGGREGATION:

Notwithstanding the seemingly vast nature of WSN applications, a set of basic building blocks for on-the-node processing can be identified. Common on-the-node operations performed on input data collected directly from the node's sensors or through in-the-network aggregation include filtering, fitting, sorting ,and searching[7].Prefix operations can be calculated in a number of ways, but we chose the binary tree approach because its flow matches the desired on-the-node data aggregation. This can be visualized as a binary tree of processing elements (PEs) across which input data flows from the leaves to the root (Fig. 2, left). This topology will form the fixed part of our approach, but in order to serve multiple applications, flexibility is also required. The tree-based data flow will, therefore, be executed on a data path of programmable PEs, which provides this flexibility together with the parallel prefix concept.

Parallel Prefix Operations

In the digital design world, prefix operations are best known for their application in the class of carry look-ahead adders. The addition of two inputs A and B in this case consists of three stages (Fig. 3): a bitwise propagate generate (PG) logic stage, a group PG logic stage, and a sum-stage. The outputs of the bitwise PG stage (Pi = Ai + Bi and Gi = Ai · Bi) are fed as (Pi , Gi)-pairs to the group PG logic stage, which implements the following expression:

(Pi , Gi) ◦ (Pi+1, Gi+1) = (Pi · Pi+1, Gi + Pi · Gi+1) (1)

It can be shown this ◦-operator has an identify element I = (1, 0) and is associative.

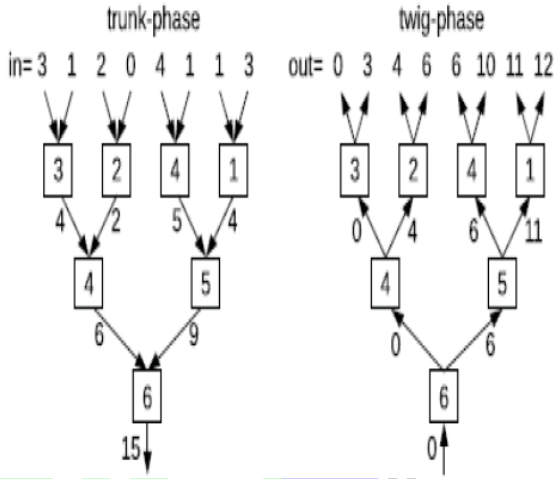


Fig. 4. Example of a prefix calculation with sum operator using Blelloch's generic approach

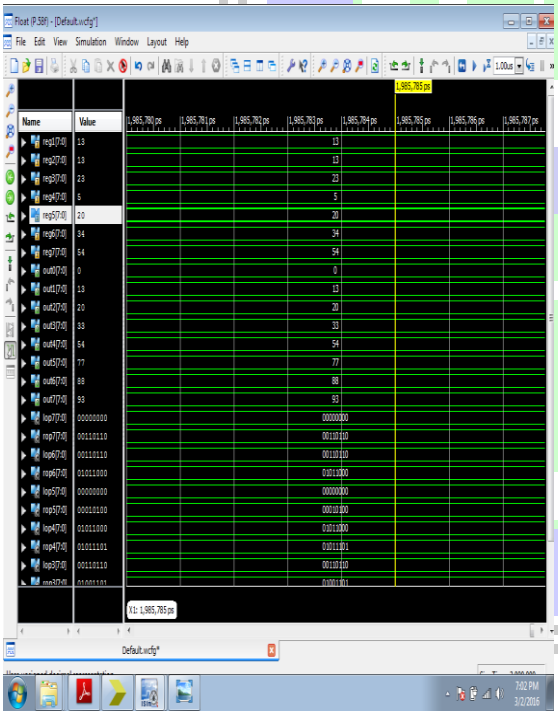
For example, the binary numbers A = "1001" and B= "0101" are added together. The bitwise PG logic of LSB-first noted A = {1001} and B = {1010} returns the PG-pairs for these values, namely, (P, G) = {(0, 1); (0, 0); (1,0); (1, 0)}.Using these pairs as input for the group PG-network, defined by the ◦-operator from (1) to calculate the prefix operation, results in the carry-array G = {1, 0, 0, 0} [i.e., the second element of each resulting pair from (1)]. In fact, it contains all the carries of the addition, hence the name carry look ahead. Combined with the corresponding propagate values Pi, this yields the sum S = {0111}, which corresponds to "1110." The group PG logic is an example of a parallel prefix computation with the given ◦-operator. The output of this parallel prefix PG-network is called the all-prefix set defined next.

For example, if ◦ is a simple addition, then the next prefix element of the ordered set [3, 1, 2, 0, 4, 1, 1, 3] is Σai = 15.Blelloch's procedure to calculate the prefix operations on a binary tree requires two phases (Fig. 3). In the trunk-phase, the left value L is saved locally as Lsave and it is added to the right value R, which is passed on toward the root. This continues until the parallel-prefix element 15 is found at the root.

FOLDED TREE:

However, a straightforward binary tree implementation of Blelloch's approach as shown in Fig.3 costs a significant amount of area as n inputs require p= n−1 PEs. To reduce area and power, pipelining can be traded for throughput. With a classic binary tree, as soon as a layer of PEs finishes processing, the results are passed on and new calculations can already recommence independently. The idea presented here is to fold the tree back onto itself to maximally reuse the PEs. In doing so, p becomes proportional to n/2 and the area is cut in half. The interconnect is reduced. On the other hand, throughput decreases by a factor of log2(n) but since the sample rate of different physical phenomena relevant for WSNs does not exceed 100 kHz, this leaves enough room for this tradeoff to be made. This newly proposed folded tree topology is depicted in Fig.1 on the right, which is functionally equivalent to the binary tree on the left.

RESULT:



CONCLUSION:

This paper presented the folded tree architecture of a digital signal processor for WSN applications. The design exploits the fact that many data processing algorithms for WSN applications can be described using parallel-prefix operations, introducing the much needed flexibility. Energy is saved thanks to the following: 1) limiting the data set by pre-processing with parallel-prefix operations; 2) the reuse of the binary tree as a folded tree; and 3) the combination of data flow and control flow elements to introduce a local distributed memory, which removes the memory bottleneck while retaining sufficient flexibility.

REFERENCES:

[1] Agrawal.B, Chong.F.T, Mysores.S and Sherwood.T,"Exploring the processor and ISA design for wireless sensor network applications",inProc. 21th Int. Conf. Very-Large-Scale Integer. (VLSI) Design, pp. 59–64, 2008.

[2] Backus.J, "Can programming be liberated from the von neumann style?", in Proc. ACM Turing Award Lect., pp. 1–29,1997.

[3] Blelloch. "Scans as primitive parallel operations," IEEE Trans. Comput., vol. 38, no. 11, pp. 1526–1538,1987.

[4] Blelloch G.E "„Prefix sums and their applications", Carnegie Mellon Univ., Pittsburgh, PA: USA, Tech. Rep. CMU-CS-90,1990.

[5] Ekanayake V.N, Kelly.C, and Manohar.R , "SNAP/LE: An ultra-lowpower processor for sensor networks", ACM SIGOPS Operat. Syst. Rev.- ASPLOS, vol. 38, no. 5, pp. 27–38,2004.

[6] Ekanayake V.N ,Kelly.C,and R. Manohar, "Bit SNAP: Dynamic significance compression for a lowenergy sensor network asynchronous processor",inProc. IEEE 11th Int. Symp. Asynchronous Circuits Syst.pp. 144–154,2005

[7] Hempstead.M, ,Lyons and G.-Y. Wei, "Survey of hardware systems for wireless sensor networks", Low Power Electron., vol. 4, no. 1, pp. 11–29,2008.

[8] Hempstead.M, Brooks.W,Welsh.D, "Tinybench: The case for a standardized benchmark suite for TinyOS based wireless sensor network devices", in Proc. IEEE 29th Local Comput. Netw. Conf., Nov. 2004, pp. 585–586,2002.

[9] Hempstead.M Brooks.D, and Wei.G, "An accelerator-based wireless sensor network processor in 130 nm cmos",vol. 1, no. 2, pp. 19 202,2011.