Design of Data Processing Based on Folded Tree for Wireless Sensor Nodes

A.Priyanka[1] S.Tamil Selvan[2]

[1,2]Kalaignar Karunanidhi Institute of Technology, Affiliated to Anna University, Coimbatore, Tamil Nadu, India

Abstract— Radio communication exhibits the highest energy consumption in wireless sensor nodes. The folded tree architecture consists of four processing element which reduce area compared to binary architecture. The four Processing Element structures use both carry look-ahead and square root carry select adder (SQRT CSLA) in the folded tree architecture. Both adder are operate on truck (transmitter) and twig phase (receiver). On Analyzing both carry look-ahead adder and square root carry select adder, the power and delay are reduced in SQRT CSLA than carry look-ahead adder. The SQRT CSLA will be implemented in wireless body area network in future extends.

Key words: Folded Tree, Wireless Sensor Node (WSN), Square Root Carry Select Adder (SQRT CSLA), Carry Look- ahead Adder (CLA)

I. INTRODUCTION

In many high- speed digital signal processing (DSP) and multimedia applications, the multiplier plays a very important role because it dominates the chip power consumption and operation speed. In electronics, adder is a digital circuit that performs addition of numbers. The main function of the digital signal processor are process discrete signals with digital filters ,filters minimize the effect of noise on a signal or enhance or modify the spectral characteristics of a signal, while analog signal processing requires complex hardware components, digital signal processors (DSP) requires simple adders, multipliers, and delay circuits and DSPs are highly efficient.

Adders are the basic building blocks of any processor or data path application. In adder design carry generation is the critical path. To reduce the power consumption of data path we need to reduce number of transistors of the adder. Carry Select Adder is one of the fast adder used in many data path applications. There is a chance to reduce the area, power and delay in the CSLA structure.

The unique operating environment and performance requirements of distributed micro sensor networks require fundamentally new approaches to system design. As an example, consider the expected performance versus longevity of the micro sensor 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 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 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, network protocols, and even the integrated circuits themselves.

To perform fast arithmetic operations carry select adder (CSLA) is one of the fastest adders used in many data-processing processors. Simple and efficient gate – level modification is used in order to reduce the delay and power of CSLA.

II. RELATED REQUIREMENTS FOR PROCESSING

Two key requirements are used to improve existing processing and control architectures can be identified.

A. Minimize Memory Access:

Modern micro controllers are based on the principles of divide and conquer strategy of ultra-fast processors. In addition the lack of task specific operations leads to inefficient execution, which results in longer algorithms and significant memory book keeping.

B. Combine Data and Control Flow Principles:

To manage the data stream and the instruction stream in the core functional unit, two approaches exist. Under control flow, the data 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. The data flow program identifies the data dependencies. The latter approach has been hugely successful in specialized high throughput applications, such as multimedia and graphics processing. The characteristics of wireless sensor networks are Data driven, many too few, Application specific.

III. EXISTING METHOD

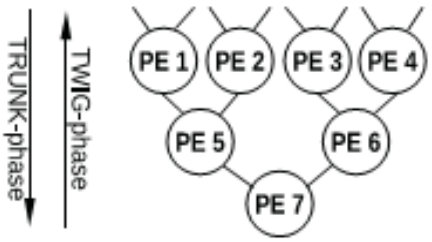

Fig. 3: Binary Tree

In Fig 3. Binary Tree is used as existing method. The disadvantages of this method is at a time only one node act as root nodes, other node act as leaves. So at a time only one data is send. Hence the power as well as energy is increased. Time requirement is high and interconnection is high. The

proposed approach gives the limited power and energy. The time requirement is low as well as interconnection path is increased. So Folded tree architecture is proposed to send the data in the way of wireless communication technique.

A. 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. A bitwise propagate-generate (PG) logic stage, a group of logic stage, and a sum stage. The output of bitwise PG stage is given below.

$$Pi=Ai \text{ (xor) } Bi \text{ , } Gi = Ai \text{ . } Bi$$

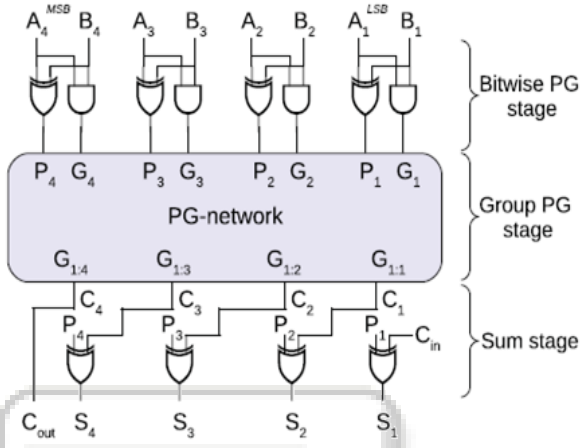

Fig. 3.1: Carry Look Ahead Adder

Group PG logic stage, which implements the following expression

$$(Pi,Gi)\ (Pi+1,Gi+1) = (Pi.Pi+1,\ Gi+Pi.Gi+1)$$

Fig 3.1 Carry Look Ahead Adder used as 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. This topology will form the fixed part of our approach, but in order to serve multiple applications, flexibilities also required. The tree-based data flow wills, therefore, bevy executed on a data path of programmable PEs, which provides this flexibility together with the parallel prefix concept. To perform fast arithmetic operations, carry select adder (CSLA) is one of the fastest adders used in many data-processing processors. The structure of CSLA is such that there is further scope of reducing the area, delay and power consumption .Square Root CSLA (SQRT CSLA) in terms of area, delay and power consumption. The result analysis shows that the proposed structure is better than the Carry Look-ahead Adder.

IV. PROPOSED APPROACH

The overview of the implemented folded tree design. Each consisting of a data path with Programmable controller and bit instruction memory. They are interconnected through the request-acknowledge handshaking trigger bus and the bidirectional data bus in the folded way

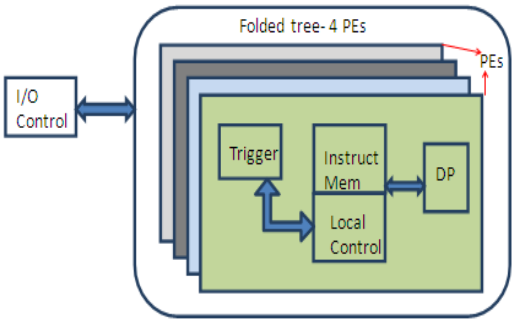

Fig. 4: The Folded Tree of 4 PEs in SQRT CSLA

Handshaking triggers activate the PEs only when new data is available and in such a way that hey functionally become a binary tree in both directions of the trunk- and twig-phase. Within each data path ,muxes select external data, stored data or the previous result as the next input for the data path. The data path contains an algorithmic logical unit (ALU) with four-word deep register files (RF-A and RF-B) at the inputs A and B for operand isolation in Fig 4.

These RFs comprise the distributed data memory of the whole system; with a combined capacity of 4 kB.They are the only clocked elements within the data path. As data flows through the tree, it is constantly kept local to its designated operation. This is one of the goals of this paper, which effectively removes the von Neumann bottleneck and saves power.

The ALU operation operate son both the carry look-ahead adder and carry select adder. Comparing the both adders of the carry look-ahead adder and square root carry select adder. The power Based on the analysis, square root carry select adder is best compare to carry look-ahead adder. The architecture of folded tree are use a four processing element in the same structure.

A. Folded Tree:

To use folded tree architecture the area and power is reduced. The idea here is to fold the tree back onto the itself to maximally reuse the PEs. In doing so, P becomes proportional to n/2 and the area is cut in half. Note that also the interconnect is reduced. On the other hand, throughput decreases by a factor of log (n) but since the sample rate of different physical phenomena relevant for WSNs does not exceed 100 kHz, this leaves enough room for this trade off to be made. There are two phase

1) Trunk Phase
2) Twig Phase

B. Trunk Phase:

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. Note that each time, a store and calculate operation is executed.

C. Twig Phase:

The twig phase starts, during which data moves in the opposite direction, from the root to the leaves. Now the incoming value, beginning with the sum identity element 0 at the root, is passed to the left child, while it is also added to the previously saved Lsave and passed to the right child. In the end, the reduced prefix set is found at the leaves.

Consider the given example A= "1001" and B= "0101" are added together. The bitwise PG logic ofLSB first noted A={1001} and B={1010} returns the PG pairs for these values are(P,G)={(0,1);(0,0);(1,0);(1,0)}The carry array results are G = {1,0,0,0}.The sum results are S={0,1,1,1}.The prefix element of the ordered set [3,1,2,0,4,1,1,3] is Σai=15.

V. PROGRAMMING AND USING THE FOLDED TREE

First the trunk phase is considered. The figure shows four processing elements. The letters L and R indicates left and right value of inputs A and B. According to Blelloch approach, L is saved as Lsave and the sum L+R is passed.

To see exactly how the folded tree functionally becomes a binary tree, all nodes of the binary tree are assigned numbers that correspond to the PE, which will act like that node at that stage. As can be seen, PE1 and PE2 are only used once, PE3 is used twice and PE4 is used three times. This corresponds to a decreasing number of active PEs while progressing from stage to stage. The first stage has all four PEs active. The second stage has two active PEs: PE3 and PE4. The third and last stage has only one active PE: PE4

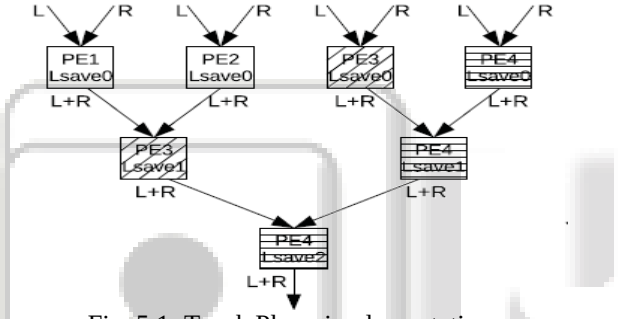

Fig. 5.1: Trunk Phase implementation

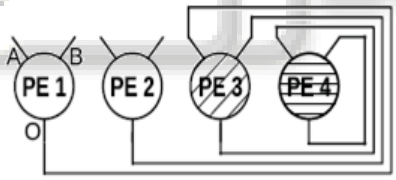

Fig. 5.2: Implications of using a folded tree.

More importantly, it can also be seen that PE3 and PE4 have to store multiple Lsave values. PE4 must keep three: Lsave0 through Lsave2, while Pe3 keeps two: Lsave0 and Lsave1. PE1 and PE2 each only keep one: Lsave0. The trunk phase PE program here has three instructions, which are identical, apart from the different RF addresses that are used. Due to the fact that multiple Lsave"s have to be stored, each stage will have its own RF address to store and retrieve them. This is why PE4 needs three instructions, PE3 needs two instructions and PE1 and PE2 need one instruction.

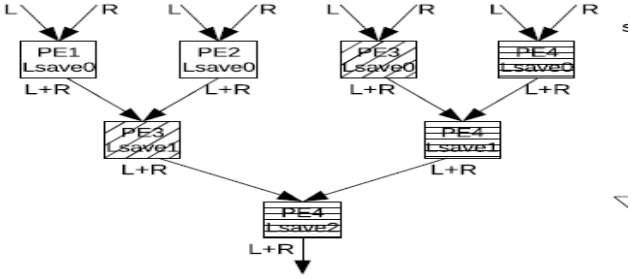

Fig. 5.3: Annotated twig phase graph of 4 PE folded tree.

In twig phase the tree operates in the opposite direction. According to Blelloch approach S is passed to the left and the sum S+L save is passed to the right. Note that here as well none of these annotations are global. The way the PEs are activated during the twig phase again influences how the programming of the folded tree must happen

VI. EXPERIMENTAL RESULTS

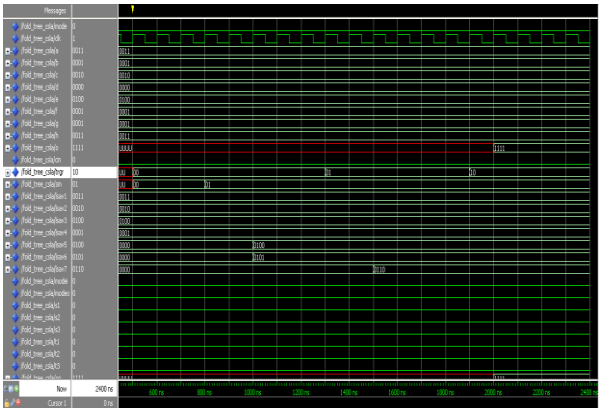

Fig. 6.1: Output of folded tree architecture

This result shows eight inputs and only one output. Sad signal is used to select the path. At 250ns the data was transmitted. At this time the power and energy is saved. Hence the time requirement is low.

Fig 6.1 output gives the dynamic energy and leakage power for one PE core running at 20 MHz and 1.2 V supply under full stress with varying data inputs. It consumes 53 mW and Maximum output required time after clock: 7.247ns.Minimum period is 6.823ns and Maximum Frequency: 146.563MHz.The register based instruction memory power values are presented in last column of the PE table and consume 6 µW. When going into idle mode, a processing element will consume 60 percentages (60%) less than in active mode.

VII. CONCLUSION

This paper describes 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, introducing the much needed flexibility energy is saved to the following conditions.

1) Limiting the data set by preprocessing.
2) The reuse of the binary tree as a folded tree

The future Scope of this project is the end of architecture router is included. It is used to reduce the delay as well as congestion.

REFERENCES

[1] Agrawal. B, Chong. F. T, Mysores. S and Sherwood. T,"Exploring the processor and ISA design for wireless sensor network applications", in Proc. 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-low power 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 low energy sensor network asynchronous processor", in Proc. 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 Computer. Network. 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. 193–202, 2011.
[10] Hennessy. J and Patterson. D, "Computer Architecture A Quantitative Approach", 4th ed. San Mateo, CA: Morgan Kaufmann, 2007.
[11] Karl. H and Willig. W,"Protocols and Architectures for Wireless Sensor Networks", 1st ed. New York: Wiley, 2005.
[12] Park. S Raghunathan. V, Schurgers. C, and Srivastava. M. B "Energy aware wireless microsensor networks", IEEE Signal Process. Mag., vol. 19, no. 2, pp. 40–50, 2002.
[13] Sanders.P and Träff.J, "Parallel prefix (scan) algorithms for MPI", in Proc. Recent Adv. Parallel

