A WALLACE TREE APPROACH FOR DATA AGGREGATION IN WIRELESS SENSOR NODES

D. Ramya Krishna M.Tech (DECS) and AMIE
[2]M.Kamaraju Ph.D, FIE;
Gudlavalleru Engineering College::Gudlavalleru

ABSTRACT:

Wireless Sensor Networks (WSN) refers to a gathering of spatially scattered and committed sensors used for to sense the environmental and physical conditions. The WSN collects and aggregates the data from all the sensor nodes and send it to the sink. But the delay required for Radio transmission of collected information to the sink is very high. If the delay of the network is high then the power consumption may be high it leads to decrease in node life time. So to avoid that problem the delay of the network must be kept at minimum in order to increase the node lifetime. If number of computations required for data aggregation process are low then automatically the delay of the network is also very less. At present a carry look ahead adder with parallel prefix algorithm for data aggregation is used but with this approach is having the disadvantages like high latency and memory. To avoid all those disadvantages a novel tree approach is proposed. The expected results are reduced in latency that is it increase the speed of data aggregation process in Wireless sensor nodes along with less memory requirement for that Tree structure.

KEY WORDS

Nodes, Wireless Sensor Networks (WSN), Folded tree, Wallace Tree

1. INTRODUCTION

A wireless sensor network (WSN) is a network consisting of possibly low size and low complex devices termed as nodes that can sense the environment and monitor that information through Wireless links. The information sensed by the sensors is shared between the nodes through relays and send to the sink by using internet. Wireless sensor Networks are used in many consumer and industrial applications like area monitoring, object monitoring, logistics, intelligent building and medical applications[1] like patient health monitoring, due to the large developments in wireless sensor networks they are used as battlefield surveillance in military applications.

2. CHARACTERISTICS OF WSN'S

Wireless Sensor Networks are having a number of unique characteristics[9] some of them are:

Communication paradigm: In WSN's all nodes are communicated[1]with each other and aggregates data and it is send to the sink.

DOI: 10.5121/ijci.2016.5411

Application specific: Depending upon the application Wireless Sensor Networks are developed to perform a specific task.

Scale and density: To cover large areas and to monitor the environmental conditions efficiently large number of nodes are placed[9] in one place that is density of the nodes are high.

Resource constraints: WSN nodes are small in size and battery so the power consumption must be low and delay of the network also kept at low for high speed of data aggregation. Like that WSN,s are having so many constraints.

Minimized Memory for data aggregation: The algorithms used for data aggregation[9] process in WSN,s are using the high memory. To avoid this problem memory required for data aggregation algorithms must be low.

3. REQUIREMENTS OF WSN'S

In order to make these sensor networks a reality, the node implementation and hardware should be optimized for the following three characteristics [9].

Low cost: The networks utility depends on high density of nodes. To make large scale deployments economically feasible, these nodes must be of very low cost.

Small size: The size of modules must be of small size in WSN networks in order to minimize power in sensor nodes.

Low power: WSN networks are having many nodes, battery replacement is very difficult and expensive or even impossible also. Nodes must have efficient energy [3] so that it can function for long periods without running out of power.

4. EXISTING WORK

WSN On-The Node Data Aggregation

Wireless sensor networks (WSN's) got researchers interest in recent years because those are used in wide range of applications. Wireless Sensor Networks could contain hundreds of sensors that collect and in some cases preprocess data before it is send to central node (Sink) for final processing. In most of the cases sensors are deployed to remote location then it is not possible to recharge or replace battery. In that cases solar and wind energy can be used, but such energy supplies are not practically suitable. Hence, increasing the life of wireless sensor networks is an very important issue.

Our existing work is developed with Prefix adder due to its less delay. In WSN ,sensors (nodes or programmable elements) are placed in sensing region. Those sensors are used to sense the data and send that data to sink (central location). All those programmable elements(PE's) are connected in different structures. Those structures are known as trees which performs data aggregation also.

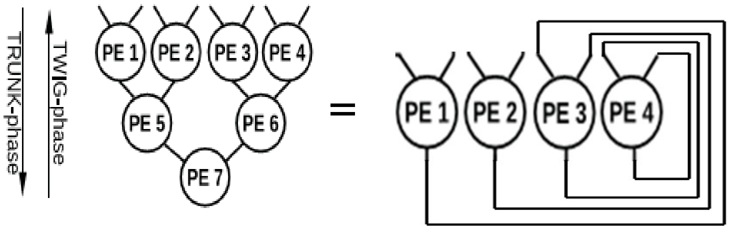


Fig. 1. Left: A binary tree with 7 PEs

Right: Folded tree with 4 PEs which is functionally equivalent to binary tree.

In binary tree 4 PE's (PE_1, PE_2, PE_3, PE_4,)are collected data and that data is added and send it to two PE's (PE_5, PE_6,) then those two data's are aggregated and send it to PE_7.Then PE_7 send that aggregated information to sink.(As shown in Fig.1.Top)

In normal binary tree[1] it requires 7 Programmable elements(PE's) to aggregate two 16 bit inputs and that produces one output but in folded tree structure only 4 PE's are enough to add two 16 bit numbers/data.(as shown in Fig.1.bottom)

Parallel Prefix adders

Prefix adders[7] are well suited for digital circuits to add two numbers then we get a Sum and Carry bit. Carry Look-Ahead Adder is developed by using parallel prefix operation we can categorize into three different stages. They are:-

First stage is the Pre-processing stage where we obtain the Group Generate and Group Propagate signals.

Second stage is the Carry generation stage where we generate the carry using the Group Generate and Group Propagate signals. ($Pi = Ai \oplus Bi$ and $Gi = Ai \cdot Bi$)
Third and Final stage is where we obtain the Sum bit using the Carry bit and the Propagate signal.

The steps illustrated above are as shown in the fig.2.
$(P_i, G_i) \quad (P_{i+1}, G_{i+1})=(P_i.P_{i+1},G_i+P_i.G_{i+1}) \circ$

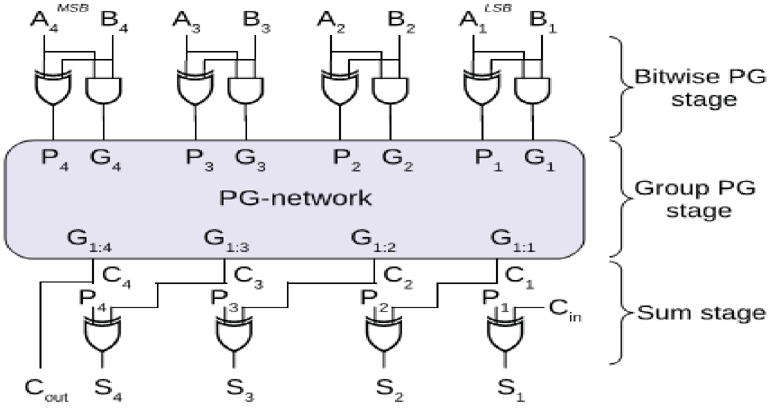

A_4 MSB B_4
A_3 B_3
A_2 B_2
A_1 LSB B_1
Bitwise PG stage
P_4 G_4 P_3 G_3 P_2 G_2 P_1 G_1
PG-network
$G_{1:4}$ $G_{1:3}$ $G_{1:2}$ $G_{1:1}$
Group PG stage
C_4 C_3 C_2 C_1
P_4 P_3 P_2 P_1 C_{in}
Sum stage
C_{out} S_4 S_3 S_2 S_1

Fig. 2. Prefix adder Folded Tree

In Fig.1. binary tree is implemented in Blelloch's approach as shown in Fig. 1. It requires p = n – 1 PEs area for n inputs .To reduce area and delay, here a fold the tree back onto itself to reuse the PEs the it is called as Folded Tree. This folded tree requires p = ((n – 1)/2) PEs for n number of inputs that is number of PEs are reduced to half.

Folded Tree is implemented in two phases. Those are

Trunk Phase
Twig Phase

5. FOLDED TREE PROGRAMMING

First consider trunk-phase in Folded Tree as shown in fig.3.Bottom. At the Fig. 3.top, a folded tree is designed with four PEs in which PE3 and PE4 are connected differently. In Trunk phase of the folded tree functionality is equivalent to the binary tree that is center again shows how data moves from leaves to the root.

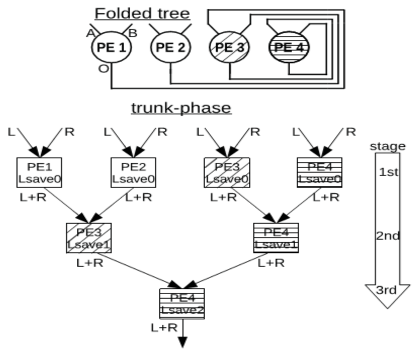


Fig. 3. Top: Four-PE folded tree &Bottom: The trunk-phase program code of the prefix-sum algorithm on a 4-PE folded tree.

In the this Trunk phase of folded tree[1] (as shown in fig.3.Bottom). The data flows by following the below steps :

The left value is stored as L save in Programmable element(PE).

The left and right values are added (L+R) and pass that output to next stage.

Now, the Twig-phase in Folded Tree is considered (as shown in Fig. 4). In Twig Phase the Folded Tree The tree operates in opposite direction to the Trunk phase.

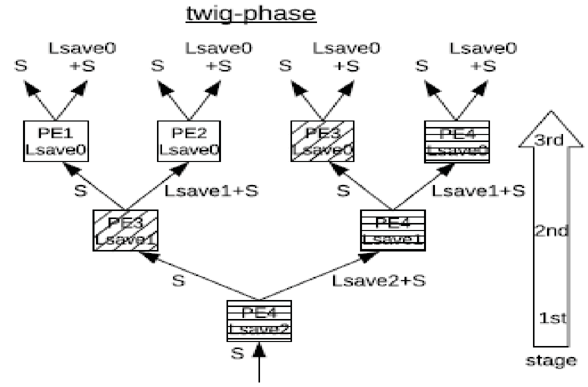


Fig. 4. Twig-phase of 4-PE folded tree.

According to Blelloch's approach in this Twig phase of Folded Tree the data flow is as shown in following steps , (as shown in fig. 4.)

The bottom value(S) is added with Programmable element value (Lsave +S) and propagate that value to right side of the PE.

At left side of the PE only bottom value(value entered into PE) is only propagated.

6. PROPOSED WORK

Our proposed architecture is Wallace tree that performs data aggregation by performing multiplication of data from various sensor nodes and send that multiplied output to the sink. In multiplication process Wallace tree reduces the number of partial products and those are added by using Sklansky adder structure.

Wallace tree:

The Structure of Wallace tree[2] has three steps. Those are,
Partial Product Generation Stage
Partial Product Reduction Stage

Partial Product Addition Stage

Partial Product Generation Stage

Partial product generation is the first step in Wallace tree multiplier. All these are the intermediate product terms generated based on the value of multiplicand and multiplier. If the multiplier bit is '1', then the multiplicand is copied as it is and forms a partial products row and if it is'0', then all the partial products in a row is also zero. From the 2nd bit multiplication onwards, each partial product row is shifted one unit to the left. In signed multiplication process, the sign bit is also extended to the left side. For a conventional multiplier Partial product generators consisting of a series of logic AND gates as shown in Fig.5.

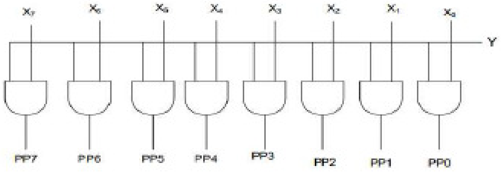


Fig.5. Partial product selection logic for simple

Partial Product Reduction Stage:

In the proposed architecture, partial product reduction is done by the use of 4:2, 5:2 compressor structures and the final stage of addition can be performed by a Sklansky adder. The latency or delay in the Wallace tree multiplier can be reduced by decreasing the number of adders in the partial products reduction stage.

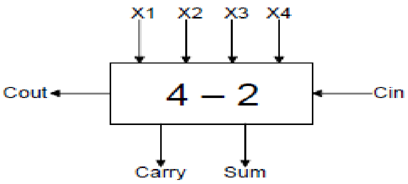
The Wallace tree is constructed by considering all the four rows of input bits in each stage at a time and compress them by using compressors. Thus, compressors form the compulsory requirement of high speed multipliers. The area, speed and power consumption of the multipliers will be directly proportion to the efficiency of the compressors. Thus, in order to satisfy the requirement of speed and power this Wallace tree is developed by using these compressors.

4-2 Compressor:

4-2 compressor (as shown in Fig.6.) has four inputs X1, X2, X3 and X4 and two outputs Sum and Carry along with a Carryin (Cin) and a Carry out (Cout) as shown in Fig. 6. The input Cin is the output from the previous lower significant compressor. The Cout is the output to the compressor in the next significant stage.

5-2 Compressor:

The 5-2 Compressor (as shown in Fig.7.) block has 5 inputsX1,X2,X3,X4,X5 and 2 outputs, Sum and Carry, along with 2 input carry bits (Cin1, Cin2) and 2 output carry bits (Cout1,Cout2) as shown in Fig.7. The input carry bits are the outputs from the previous lesser significant compressor block and the output carry is passed on to the next higher significant compressor block.



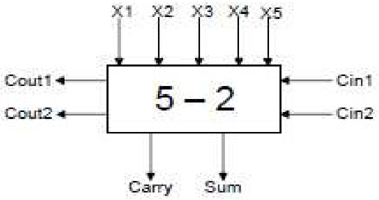


Fig.6. 4:2 compressors Fig.7. 5:2 compressors

Now the Fig.8. Shows the partial product reduction stage in Wallace tree by using 4:2,5:2 compressors.

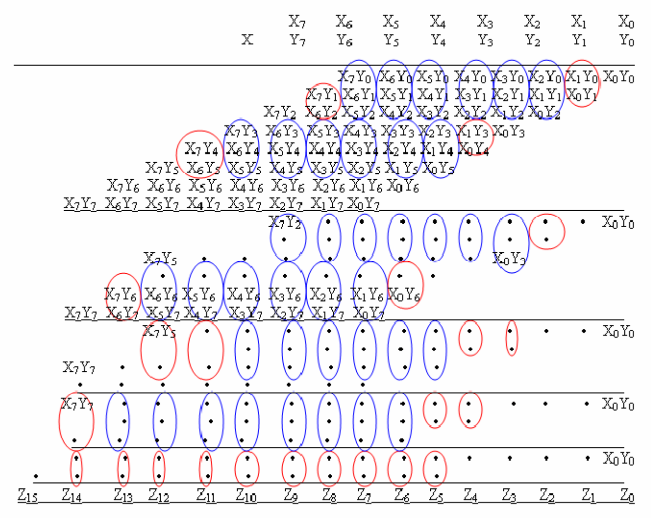

X
X7 X6 X5 X4 X3 X2 X1 X0
Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0
X7Y0 X6Y0 X5Y0 X4Y0 X3Y0 X2Y0 X1Y0 X0Y0
X7Y1 X6Y1 X5Y1 X4Y1 X3Y1 X2Y1 X1Y1 X0Y1
X7Y2 X6Y2 X5Y2 X4Y2 X3Y2 X2Y2 X1Y2 X0Y2
X7Y3 X6Y3 X5Y3 X4Y3 X3Y3 X2Y3 X1Y3 X0Y3
X7Y4 X6Y4 X5Y4 X4Y4 X3Y4 X2Y4 X1Y4 X0Y4
X7Y5 X6Y5 X5Y5 X4Y5 X3Y5 X2Y5 X1Y5 X0Y5
X7Y6 X6Y6 X5Y6 X4Y6 X3Y6 X2Y6 X1Y6 X0Y6
X7Y7 X6Y7 X5Y7 X4Y7 X3Y7 X2Y7 X1Y7 X0Y7
X7Y2
X0Y0
X7Y5
X0Y3
X7Y6 X6Y6 X5Y6 X4Y6 X3Y6 X2Y6 X1Y6 X0Y6
X7Y7 X6Y7 X5Y7 X4Y7 X3Y7 X2Y7 X1Y7 X0Y7
X7Y5
X0Y0
X7Y7
X7Y7
X0Y0
X0Y0
Z15 Z14 Z13 Z12 Z11 Z10 Z9 Z8 Z7 Z6 Z5 Z4 Z3 Z2 Z1 Z0

Fig.8. Partial product reduction stage in Wallace tree

Partial Product Addition Stage

Sklansky tree is commonly known as the divide-and-conquer tree it is as shown in Fig.9.

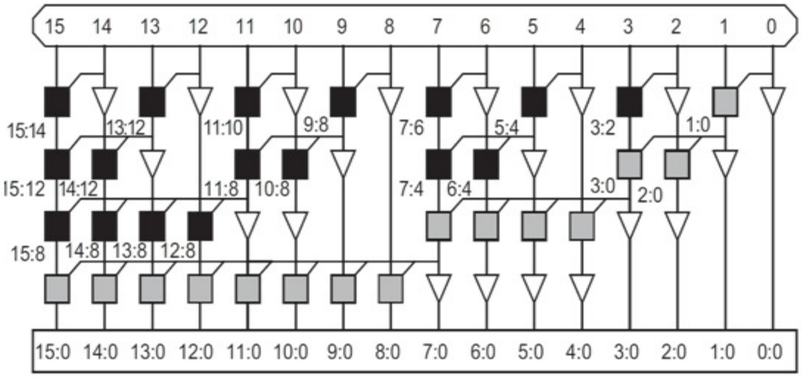
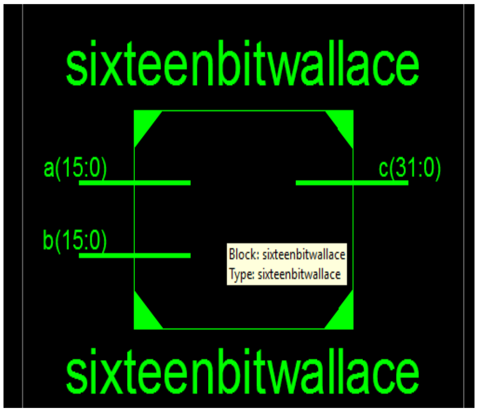


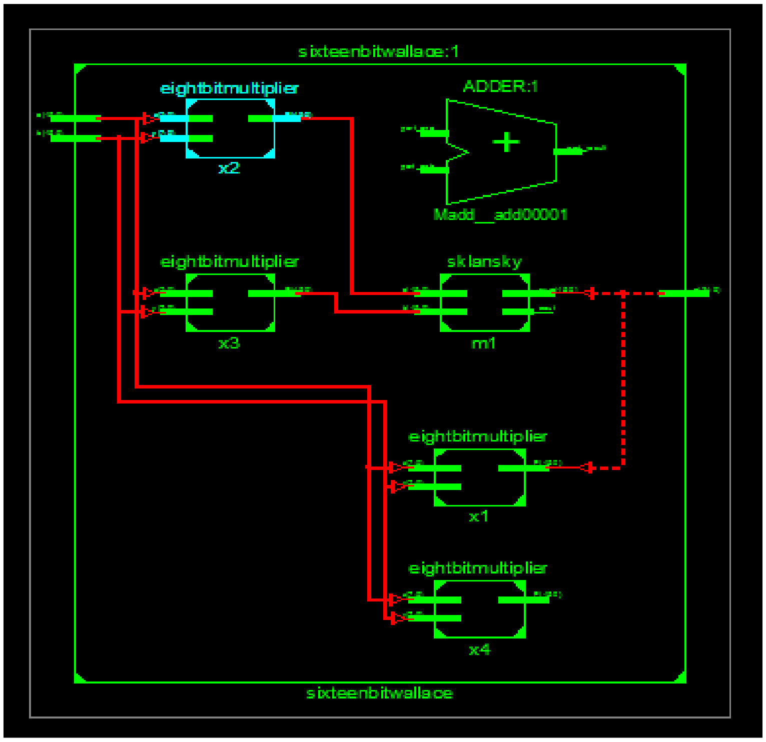
Fig.9. Structure of Sklansky adder

At the final stage of the Wallace Tree this Sklansky adder is used. It reduces the delay to log_2N stages. Hence delay of the Wallace Tree this Sklansky adder is very low.

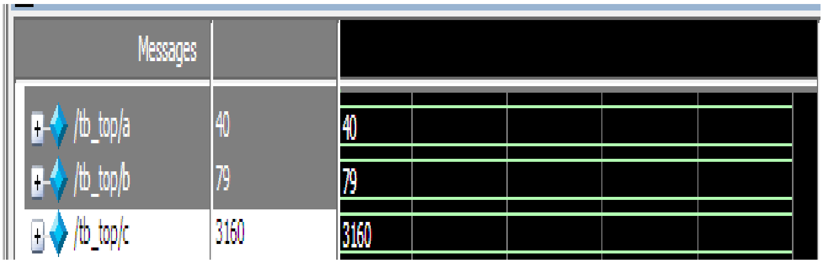
7. RESULTS

RTL Schematic Diagram for 16 bit Wallace tree with Sklansky adder:





Simulation Waveforms for 16 bit Wallace tree with Sklansky adder:



```
Messages
/tb_top/a   40     40
/tb_top/b   79     79
/tb_top/c   3160   3160
```

The table 1 describes that 16 bit Wallace tree is having less delay and memory compared to 16 bit folded tree.

Table 1: Comparison of 16 bit folded Tree and 16 bit Wallace Tree:

Parameter	Total values for 16 bit Folded Tree	16 bit Walla Tree in Sklansky adder
Delay(ns)	29.922	23.322
Memory(KB)	329720	179228

8. CONCLUSION

A 16 bit Wallace tree architecture was developed for data aggregation in Wireless Sensor Nodes.

This architecture gives better results compared to folded tree architecture. Those are

1. Less Delay (Speed of Data aggregation process is increases)
2. Memory of the Wallace architecture is also less.

REFERENCES

[1] Cedric Walravens, and Wim Dehaene, " Low Power Digital Signal Processor Architecture for Wireless Sensor Nodes" IEEE transactions on very large scale integration (VLSI) systems, vol. 22, no. 2, pp. 313–321,February 2014.

[2] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energyaware wireless microsensor networks," IEEE Signal Process. Mag., vol. 19, no. 2, pp. 40–50, Mar. 2002.

[3] C. Walravens and W. Dehaene, "Design of a low-energy data processing architecture for wsn nodes," in Proc. Design, Automat. Test Eur. Conf. Exhibit., Mar. 2012, pp. 570–573.

[4] Dakupati.Ravi Sankar1, Shaik Ashraf Ali2 " Design of Wallace Tree Multiplier by Sklansky Adder". International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 ,Vol. 3, Issue 1, January -February 2013, pp.1036-1040.

[5] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective. Reading, MA, USA, Addison Wesley, 2010.

[6] L. Nazhandali, M. Minuth, and T. Austin, "SenseBench: Toward an accurate evaluation of sensor network processors," in Proc. IEEE Workload Characterizat. Symp., Oct. 2005, pp. 197–203.

[7] P. Sanders and J. Traff, "Parallel prefix (scan) algorithms for MPI," in Proc. Recent Adv. Parallel Virtual Mach. Message Pass. Interf., 2006, pp. 49–57.

[8] G. Blelloch, "Scans as primitive parallel operations," IEEE Trans. Comput., vol. 38, no. 11, pp. 1526–1538, Nov. 1989.

[9] H. Karl and A. Willig, Protocols and Architectures for Wireless Sensor Networks, 1st ed. New York: Wiley, 2005.

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

AUTHOR'S PROFILE

D. Ramya Krishna is M.Tech student in Gudlavalleru Engineering College, Andhra Pradesh. She has complete B.Tech from Mallineni Perumallu Educational Socities Guntur.

Dr.M. Kamaraju is Professor & Head of the Department in E.C.E of Gudlavalleru Engineering College. He has about 22 years of teaching experience and he presented papers in several international and national conferences and published papers in international journals. His main insteresting domain is Low Power VLSI design.