Asian Journal of Engineering and Applied Technology (AJEAT)
Vol.2.No.1 2014pp 12.- 17.
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

LOW POWER WIRELESS SENSOR NETWORK BASED ON DVFS ALGORITHM

B. Imayavathi
Final Year , M.E. – VLSI DESIGN
Sethu Institute of Technology
Pulloor, Kariapatti,
E-Mail: bimayavathibeece@gmail.com

Dr. R. Ganesan, Ph.D.,
HOD, VLSI DESIGN
Sethu Institute of Technology
Pulloor, Kariapatti.
E-Mail: ganesanhod@gmail.com

Mr. V. Karthik, M.Tech.,
Assistant Professor, ECE Department
Sethu Institute of Technology
Pulloor, Kariapatti.
E-Mail: karthik2055@gmail.com

ABSTRACT

Wireless Sensor Networks (WSN) has become a very significant enabling technology in civil, military, radio communication and medical applications for collecting and processing of complex environmental data. Wireless sensor networks have several important attributes that require special attention to device design. These include the need for inexpensive, long-lasting, highly reliable devices coupled with very low performance requirements. Radio communication has highest energy consumption in wireless sensor nodes. Sensor nodes are battery driven and have limited power. Hence, sensor nodes lifetime on the order of months to years. Hence, energy consumption is the important factor in determining sensor nodes lifetime. Power consumption is high during data processing and data transmission process. Inorder to reduce number of data sensed by nodes Data Aggregation is used. Data Aggregation converts multiple sensed data into single data. Here proposing, data aggregation process implemented using DVFS (Dynamic Voltage and Frequency Scaling) algorithm used at input side along with folded tree architecture to reduce power by reducing number of processing elements and to improve performance of sensor nodes. Experimental results shows that using DVFS algorithm reduces power up to 13% compared to existing processing methods.

Key words: Folded tree, DVFS, processing elements, WSN

1. INTRODUCTION

Wireless Sensor Network (WSN) is an - emerging technology in recent years. Its application ranges from home appliances to military services. The WSN consist of sensor, transmitter, receiver, controller, memory and processor. Hundreds and thousands of sensor nodes are interconnected to one another to form a network. These intermediate sensor nodes are called as motes. Sensor nodes use batteries as a source of power supply. Batteries have limited energy and sensor nodes life time varies from months to years depending on its application. Data aggregation is used to process data which are sensed by sensor nodes into a single useful data using aggregate functions such as sum, average, etc..Hence, most of the power is consumed during sensed data processing. Due to power consumption nodes lifetime is limited and sensor nodes are expensive to buy. To increase the lifetime time of sensor nodes, power consumed during the processing of data is to be reduced. In order to reduce power consumed by PEs in sensor nodes, folded tree architecture along with DVFS (Dynamic Voltage and Frequency Scaling) algorithm is proposed in this paper. In section II existing works are described, in section III data aggregation and parallel processing steps are described, in section IV proposed work is explained, in section V results are compared with previous works and work is concluded in section VI.

2. EXISTING WORKS

Accelerator based computing and VDD gating to reduce power consumption in sensor nodes [1]. VDD gating is used to reduce power by shutting off the current supplied to blocks not in use during processing of sensed data. Power consumption is high and it’s a time consuming process. Power reduction using VDD gating is around 0.55V. Dynamic Voltage Scaling (DVS) where voltage is pre-computed and stored in look-up tables before data processing. Voltage given to elements during processing is increased or decreased depending upon its use. It has high delay margins, since output depend directly on given input voltage and is applicable only for short data length [2]. Sensor nodes designing vary depending on its application is discussed in [3]. Only simulation level architecture is implemented and not implemented in hardware. [4] uses Asynchronous QDI circuits. Hand shaking protocol is used in asynchronous circuits for communication between devices. In QDI circuits there is no assumption about delay of any circuit elements or wires but assume delay for output to reach two different destinations. It suffers from size overhead, no well developed test procedure and it's difficult to learn.Various power supplies used in sensor nodes from power supply, batteries to solar cell usage and their performances are discussed in [5].Folded Tree Architecture which reduces the number of processing elements used in data processing by using the concept of reusing [6]. It's complex for large number of sensed data.Multi – Objective Genetic Algorithm for synthesizing low power circuits [7]. A gate level description of the circuit is encoded into a single chromosome which is evolved using fitness function in genetic algorithm.Distributed Weighted Sampling Algorithm to sample sensed data to reduce energy consumption during processing [8].Genetic Algorithm to balance the loads of energy consumption of all sensor nodes using fitness values [9].Tree prefix to compute intermediate signals and computation graph is used for area and power optimization in [10] and [11].Bitwise Timing Constraints is used to minimize area and total switching activities [12], [13].

3. DATA AGGREGATION AND PARALLEL PROCESSING

Data aggregation is the process of combining multiple row values into single significant value based on certain criteria. The functions used in data aggregation for combining multiple parallel values are sum, average, count, maximum, minimum, median, mode, etc.... data aggregation is done using parallel prefix operation, where the number of processing elements is reduced in successive i+1 level. Data aggregation is used to eliminate redundant data in data transmission and the number of data is reduced for transmission.

Various aggregation techniques are as follows:

- Centralized approach
- In network aggregation
- Tree based approach
- Cluster based approach

In centralized approach, data from sensor nodes are sent to a centralized node for transmission via shortest path. The centralized node is called leader. Leader is used to aggregate data from various nodes into a single data for transmission to its child node. In network aggregation, is used to process, gather and routing of data in a multi hop network, with the aim of reducing resource consumption and to increase node lifetime. Two approaches in in-network aggregation are network aggregation with size reduction and network aggregation without size reduction. In tree based approach, aggregation tree is constructed by considering root node as target and leaves nodes as source. Data flows from leaves to root where data aggregation is done at nodes present at level (i-1).

In cluster based approach, whole wireless network is clustered or divided into several clusters. Each cluster has a cluster head among its members and transmission is done via cluster head.

A. *Parallel Processing*

Parallel processing is the ability to carry out multiple operations or tasks simultaneously. Parallel processing is also called parallel computing. Parallel computing is the simultaneous use of more than one CPU or processor core to execute a program or multiple computational threads. Ideally, parallel processing makes programs run faster. **Advantages**: Faster execution time, so higher throughput. **Disadvantages** More hardware required, also more power requirements, not good for low power and mobile devices.

B. *Parallel Prefix Operation*

The parallel prefix or scan operation is a surprisingly versatile primitive and a basic building block in massively parallel algorithms for a variety of different problems. The parallel prefix operation can be explained as follows. Let p be the number of processing element (PE)s numbered consecutively from 0 to p − 1, and let a sequence of p elements xi with an associative, binary operation is given.

The inclusive parallel prefix operation computes for each PE j, 0 ≤ j < p the value Lji=0

$$x_i = x_0 \oplus x_1 \oplus x_3 \oplus \ldots\ldots\ldots\ldots\ldots. \oplus x_j$$

with the convention that Lji=j xi = xj (a one element sum is just that one element). The exclusive parallel prefix operation computes for each PE j, 0 < j < p except PE 0 the value Lj−1 i=0. With this definition, no neutral element for the operation ⊕ is required.

Sensors are used to gather data which is transmitted form source. The gathered data's have to be processed in parallel to reduce computation time. Parallel prefix operation is used to process sensed data. Inputs are given parallel to reduce computation time and prefix is used to denote that output depends on initial inputs. Carry look ahead adder is used to implement parallel prefix operation to reduce data processing time.

Three stages in parallel processing using carry look ahead adder are

- Pre-calculation of P_i and G_i for each stage.
- Calculation of carry for each stage.
- Combining P_i and C_i of each stage to generate final sum.

Carry adder output is calculated using following formula

$(G_i ,P_i) = (g_i , p_i) \text{ o } (G_{i-1} , P_{i-1})$

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 P_i , this yields the sum S = {0111}, which corresponds to "1110."

For example, if ◦ is a simple addition, then the prefix element of the ordered set [3, 1, 2, 0, 4, 1, 1, 3] is a_i = 15.Blelloch's procedure is used to calculate the prefix-operations on a binary tree requires two phases:

(i) Trunk phase
(ii) Twig phase

At the end of Trunk phase parallel prefix element is found. At the end of Twig phase reduced prefix set is found. In trunk phase, the left most value is saved in processor and is added with the right input and the result is passed towards root. In twig phase, value 0 is given as a input to root node and is sent unchanged to left leaf and value is added with saved value and is sent to right leaf node. The output is taken at the leaf nodes.

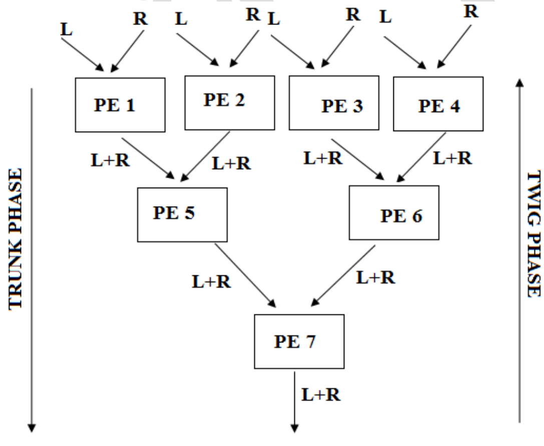


Fig. 1 Parallel prefix operation

4. PROPOSED WORK

In our proposed approach, Data aggregation is implemented using folded tree architecture along with DVFS (Dynamic Voltage and Frequency Scaling) algorithm which is used to reduce power and number of processing elements in sensor nodes and to improve performance and life time of sensor nodes. DVFS is used at input side to vary voltage and frequency based on sensed data value.

A. *Folded Tree*

However, parallel prefix operation implemented using binary tree approach with n inputs require p = n − 1 PEs and consume more area and power. Pipelining is used to reduce area and power and to increase throughput. With binary tree approach, as soon as PEs at level i finishes processing, the results are passed on to level i+1 where processing is done independently.

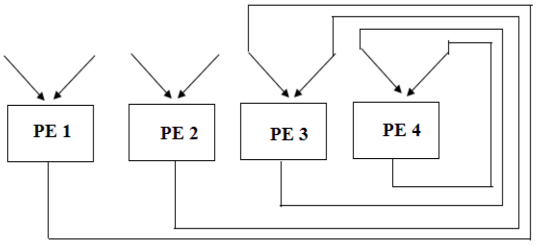


Fig. 2 Folded tree

The structure is modified i.e. structure of parallel tree is folded back onto itself to maximally reuse the PEs. Hence, p becomes proportional to n/2 and the area is cut in half. Throughput decreases by a factor of $\log_2(n)$ but since the sample rate of various WSNs does not exceed 100 kHz [12]. The proposed folded tree topology is depicted in Fig. 1 on the right, which is functionally equivalent to the binary tree on the left.

B. *DVFS*

DVFS (Dynamic Voltage and Frequency Scaling) is an effective method to reduce considerable power in low power circuits. It exploits correlation in run time distribution and satisfies deadline constraints. Frequency is set to the ratio of remaining work-to-do (workload prediction) to remaining time-to-deadline. Software work load is classified into three as follows:

- Data dependency
- Control flow dependency
- Architectural dependency

Workload prediction results in pessimistic voltage and frequency scaling by setting voltage/frequency values corresponding to operation mode.

C. *Block Diagram*

Schematic overview of the implemented folded tree design is given in fig.3. Data path with controller and memory are interconnected through request acknowledge handshaking trigger bus and the bidirectional data bus in the folded way. Handshaking triggers activate the PEs only when new data is available and in such a way that they functionally become a binary tree in both directions of the trunk- and twig-phase.

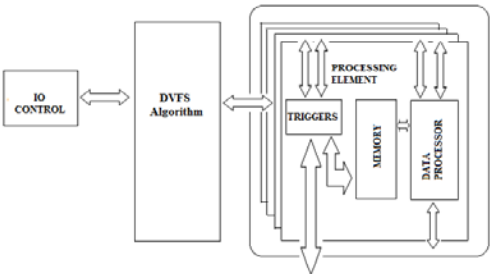


Fig. 3 Folded tree architecture with DVFS

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. These RFs comprise the distributed data memory of the whole system. This reduces memory bottleneck in system. DVFS is used to reduce voltage at input based on signal amplitude sensed by nodes which is applied to processing elements.

D. *Circuit Diagram*

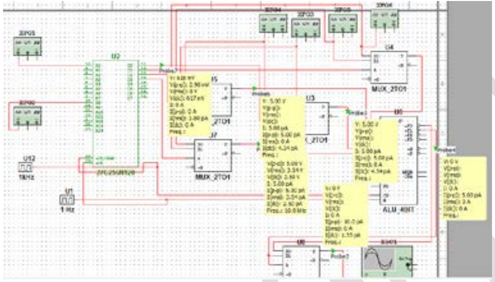


Fig. 4 Circuit diagram

Input is given to the circuit from function generator , where given input frequency, voltage and duty cycle can be changed. Input is given to memory where the processed data is saved. Output from memory is given to multiplier circuit for selecting signals based on select line value. Output from multiplier is the signal with less distortion and noise value. Then the output is given to ALU unit and signal is processed and then the output is measured using probe and output signal is viewed through oscilloscope.

5. EXPRIMENTAL ANALYSIS

A. *Workload Prediction:*

Workload is precalculated which provides amount of work to be done based on data set values. The remaining workload is calculated which gives us with minimum energy consumption. Remaining workload is denoted as W (i, j). Transition probabilities are different for different set of data sets. So energy optimization will be different for different set of data. Energy consumption (E_i) is calculated using the formula given below:

$$E_i = p\,(n_{i,j}\,,\, n_{i+1,j}).(f^2_{i,j}\; x_{i,j} + f^2_{i+1,j}\, x_{i+1,j}) +$$
$$p\,(n_{i,j}\,,\, n_{i+1,j+1}).(f^2_{i,j}\; x_{i,j} + f^2_{i+1,j+1}\, x_{i+1,j+1}) +$$
$$p\,(n_{i,j+1}\,,\, n_{i+1,j}).(f^2_{i,j+1}\; x_{i,j+1} + f^2_{i+1,j}\, x_{i+1,j}) +$$
$$p\,(n_{i,j+1}\,,\, n_{i+1,j+1}).(f^2_{i,j+1}\; x_{i,j+1} + f^2_{i+1,j+1}\, x_{i+1,j+1})$$

$f_{i,j}$ is the ratio of remaining workload to remaining time. By integrating energy consumption with density function, average energy consumption E_{avg} is given by the formula:

$$E_{avg} = w^2_{i,j}.\cancel{x_{i,j}} + z_{i,j}.\int p_i/(1- (x_{i,j}/w_{i,j}))^2\, dx_i +$$
$$w^2_{i,j+1}.\; \cancel{x_{i,j+1}} + z_{i,j+1}\,.\int p_i/(1-(x_{i,j+1}/w_{i,j+1}))^2\, dx_i$$

The average energy consumption (E_{avg}) is continuous with respect to $w_{i,j}$ and $w_{i,j+1}$, $w_{i,j}$ and $w_{i,j+1}$ which minimize the average energy consumption. In similar method, workload for N nodes is calculated.

B. *Voltage / Frequency Selection*

The range of runtime called bin index is predicted first using the runtime history of each node. This process is called bin prediction. The remaining workload for each node is predicted using the bin index value. Frequency is given by

$F_i = w_i/T_i$, Where w_i represent remaining workload and T_i represent remaining time.

Feasibility check is the process of checking whether deadline is satisfied when frequency is set. Feasibility condition is given below:

$$((\, WCEC_i / f_i\,) + T_{tr}\,) + T^{min}_{i+1} \le T_i$$
$$T^{min}_{i+1} = (\, WCEC_{i+1,end} / f_{max}\,) + T_{tr}$$

$WCEC_i$ and $WCEC_{i+1,\,end}$ denote worst case execution cycle of i^th^ and i+1^th^ instruction to end of the program. The voltage is also scaled which gives minimum energy consumption when processor is working at calculated frequency level.

C. *PE Measurements:*

The dynamic energy and leakage power for one PE running at 20 MHz and 1.2 V supply consumes 42 μW or 2.1 μW/MHz, including 0.03 μW leakage. The register-based instruction memory consumes about 6 μW. In idle mode, a PE will consume 60% less power than in Active mode.

At each stage, the number of active PEs is cut in half as 8, 4, 2, 1. The number of idle PEs increases accordingly as 0, 4, 6, 7. Therefore a total of 15 active PEs and 17 idle PEs. Overall, the folded tree processor consumes 255 μW or 13 pJ/cycle, including memories.

D. *Energy Per Instruction*

The normalized energy-per-instruction values are calculated using the following formula:

$$E_{norm} = E_{orig} \times 130\ \text{nm}/L \times (1.2\ V/V_{dd})^2 \times 16\ \text{bit}/W\ \ .$$

E. *Simulation Results*

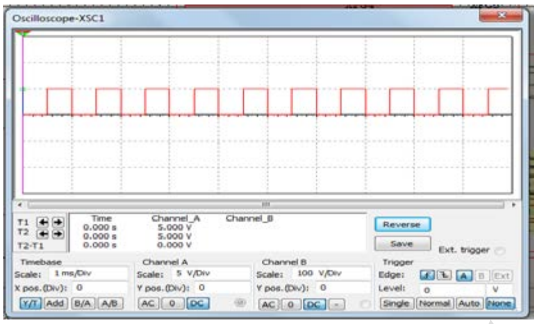
The folded tree consumes 8–10× in terms of energy and at least 2–3× in terms of execution time.

CURRENT (pA)	VOLTAGE (V)	POWER (kW)
1.87	2	0.00374
3.83	2	0.00766
4.99	2	0.00998
	4	0.01996
	4.25	0.0212075

	5	0.02495

Table 1 Power calculation

Speed gain can be made even more energy- efficient execution by lowering the supply voltage until an equal throughput is reached. Operating at half the frequency (10 MHz) and a minimal supply voltage of 0.79 V, the processor consumes about half the energy. A single active PE core will now only consume 0.95 μW/MHz, including leakage. Overall, the folded tree processor now consumes down to 80 μW or 8 pJ/cycle.



	Time	Channel_A	Channel_B
T1	0.000 s	5.000 V	
T2	0.000 s	5.000 V	
T2-T1	0.000 s	0.000 V	

Timebase: Scale: 1 ms/Div; X pos.(Div): 0
Channel A: Scale: 5 V/Div; Y pos.(Div): 0
Channel B: Scale: 100 V/Div; Y pos.(Div): 0
Trigger: Edge: A; Level: 0 V

Fig. 5 Simulated output

Power consumption during simulation of processor architecture using Multisim 11.0 platform is about 4V, which is equivalent to 0.00998 W. Simulated power analysis output is shown in fig. 4.

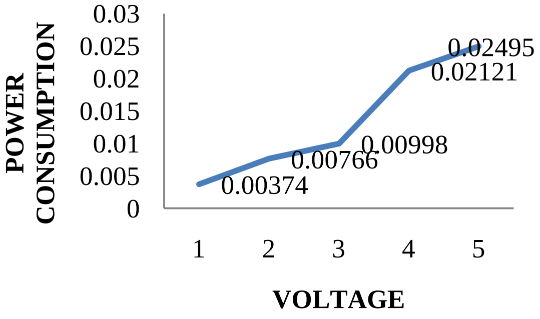


Fig. 6 power comparison chart

Above figure shows, power consumption at different voltage levels. Power consumption is reduced when voltage is scaled from higher value to lower value by applying DVFS algorithm method.

6. CONCLUSION

In this paper, folded tree architecture with DVFS algorithm is proposed to implement data aggregation process. In data aggregation data set is reduced by using parallel prefix. Parallel prefix is implemented using folded tree where the number of processing elements is reduced to half and power consumption of processors is reduced by reducing sleep to active mode power of processors. Mathematical formulations are used to minimize workload given to node and also energy consumed by nodes. Energy consumed by single processor is calculated from which variation in processor power during processing can be found. It results in reduced area and power consumption of processors. Number of data used for processing is reduced by using data aggregation which is implemented using folded tree. DVFS is used to scale down frequency and voltage and to improve power consumed by processors. Thus, voltage and number of processor is reduced considerably and performance is improved.

REFERENCES

[1] M. Hempstead, D. Brooks, and G. Wei, "*An accelerator-based wireless sensor network processor in 130 nm cmos*," J. Emerg. Select. Topics Circuits Syst., vol. 1, no. 2, pp. 193–202, 2011.

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

[3] B. A. Warneke and K. S. J. Pister, "*An ultra-low energy micro- controller for smart dust wireless sensor networks*," in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers. , pp. 316–317, Feb. 2004.

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

[5] Z. Watral, A. Michalshi, "*Power sources for WSN*" Instrumentation and Measurement magazine, IEEE – vol.16, pp – 37-43,Feb 2013.

[6] C. Walravens, W. Dehaene, "*Low power digital signal processor architecture for wireless sensor networks*", in IEEE transactions for VLSI systems, Vol. No-22,pp-313-321,Feb 2014.

[7] T. Arslan, D.H. Horrocks and A.T. Erdogan, "*Overview and design directions for low power circuits and architecture for digital signal processing*", Institution of Electrical Engineers,pp-611-615,1995.

[8] B.V. Elsevier, "*Information processing and data management in Wireless Sensor Networks*", Signal processing, pp-2859-2860,2007.

[9] J. Sklansky, "*Conditional sum additions logic*", Electronic computers, IRE transactions, Vol. EC-9,pp- 226-231, Aug-2009.

[10] P. Richard Brent, H. T. Kung, "*A regular layout for parallel adders*", IEEE transactions on computers, Vol. C-31, Issue 3, pp-260-264, Aug-2006.

[11] T. Matsunaga and Y. Matsunaga, "*Timing constrained area minimization algorithm for parallel prefix adders*", IEICE Transactions on Fundamental of Electronics , Communication and Computer Science, Vol. No- E90,Issue 12,pp-2770-2777,Dec-2007.

[12] T. Matsunga, Y. Matsunga, and S. Kimura, "*Synthesis of parallel prefix adders considering*

switch activities", IEEE Conference on computer design, pp-404-409, Oct-2008.

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

[14] V. N. Ekanayake, C. Kelly, and R. Manohar, "*BitSNAP: Dynamic significance compression for a lowenergy sensor network asynchronous processor*," in Proc. IEEE 11th Int. Symp. Asynchronous Circuits Syst., Mar. 2005, pp. 144–154.

[15] Marc S. Bright and Tughrul Arslan,*" Synthesis of Low-Power DSP Systems Using a Genetic Algorithm*" IEEE Transactions On Evolutionary Computation, Vol. 5, No. 1, Feb. 2001.

[16] Nandini. S. Patil, Prof. P. R. Patil," *Data Aggregation in Wireless Sensor Network*", 2010 IEEE International Conference on Computational Intelligence and Computing Research.