Implementation of SAD Algorithm with Folded Tree Architecture using VHDL

Resma S[#1], Ragimol[#2]

[#1]*PG student , Department of Electronics and Communication Engineering, Sree Buddha College of Engineering, Kerala University, India*

[#2]*Assitant Professor, Department of Electronics and Communication Engineering, Sree Buddha College of Engineering, Kerala University, India*

Abstract* — The trend of smaller, portable and more capable electronic devices give rise to a number of design and implementation problem, mainly due to the energy consumption. The highest energy consumption in radio communication, so in order to reduce the energy and power, Folded tree architecture is beneficial. The existing architecture known as Binary tree architecture, is a tree data structure in which each node has at most two children, but this architecture requires large number of processing elements. Thus the Folded tree architecture is used, which has two phases, trunk and twig, this help in reducing the number of processing elements. Wireless Sensor Network (WSN) has wide range of application in medical monitoring, environmental sensing, industrial inspection and military surveillance. The data-driven nature of Wireless Sensor Nodes applications requires a specific data processing approach. Motion estimation is the most critical component of video coding system. Sum of Absolute Difference (SAD) algorithm is the most common matching criteria choosen for motion estimation because of its low complexity and good performance and it is a tree structure. Due to the structural similarity of Sum of Absolute Difference (SAD) algorithm, motion estimation can implement using binary tree and folded tree architectures. The area, power and delay are reduced in the proposed architecture. This paper describes the design and implementation of the newly proposed folded-tree architecture with Sum of Absolute Difference (SAD) algorithm for motion estimation.

Keywords* — Wireless Sensor Network (WSN), parallel prefix operation, binary tree, folded tree, Sum of Absolute Difference (SAD) algorithm, motion estimation.

I. INTRODUCTION

The development of Wireless Sensor Networks was motivated by military applications such as battlefield surveillance, such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on. Wireless Sensor Network node has three parts. They are sensors, radio, and micro controller. These three parts are combined with a limited power supply. Since radio transmissions are very expensive in terms of energy. The ratio of communication to computation energy cost can range from 100 to 3000 [1]. So data communication must be traded for on-the-node processing which in turn can convert the many sensor readings into a few useful data values. The data-driven nature of WSN applications needs a specific data processing approach.

The Wireless Sensor Network consist of different types of architectures. In this paper Binary tree architecture [2] and Folded tree architecture [3] are designed and implemented. Highest energy consumption in radio communication. So in order to reduce the energy and power, Folded tree architecture is better. The existing architecture known as Binary tree architecture, is a tree data structure in which consist of nodes, but this architecture needes large number of Processing Elements (PEs). Thus the Folded tree architecture is used, help in reducing the number of processing elements by reusing the Processing Elements (PEs). The Folded tree architecture is a low power Digital Signal Processor (DSP)architecture, which reduce the area, power and delay than Binary tree architecture. Hence using Folded tree architecture with Sum of Absolute Difference (SAD) algorithm [4] for implementing motion estimation.

Sum of Absolute Difference algorithm is a tree structure, which consist of sum elements and difference elements. The structural similarity of SAD with this two architecture , which can implemented using Binary tree and Folded tree, hence estimating the motion vector. Motion estimation is the most critical component of video coding system, determines motion vectors that describe the transformation from one 2D image to another. The Sum of Absolute Difference (SAD) algorithm is an extremely fast metric due to its simplicity. Different motion estimation algorithms are existing, but which is in the field of Image processing. In field of Very Large Scale Integration (VLSI), Sum of Absolute Difference algorithm is prefered, because it consist of addition and subtraction unit. In this work motion estimation done with existing architecture known as Binary tree and proposed architecture known as Folded tree architecture, which reduce area, power and delay in a netwok.

II. RELATED REQUIREMENT FOR PROCESSING

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

A. Minimize Memory Acess

Modern micro controllers are working on the principle of divide and conquer strategy of ultra fast processors [5]. The lack of task specific operations leads to insufficient execution, that results in longer algorithms and significant memory book keeping.

B. Combine Data and Control Flow Principles

Two approaches are exist to manage the data stream and instruction stream in core functional unit. Under control flow, the data stream is a consequence of instruction stream and the instruction stream is the consequence of data stream.A traditional processor architecture is a control flow machine, that execute programs sequentially as a stream of instructions. The data flow program identifies the data dependencies.

III. EXISTING METHOD

The existing method for implementing motion estimation with SAD algorithm is binary tree. 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 required interconnection is high. The proposed approach gives the limited power and energy for motion estimation with SAD. The Folded tree

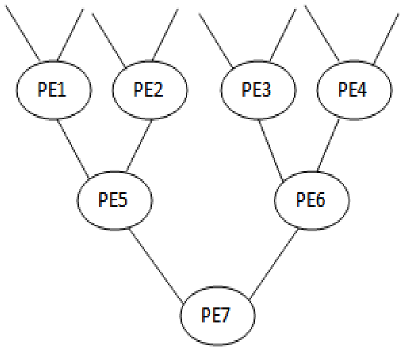


Fig. 1. A binary tree

architecture is proposed to send the data in the way of wireless communication technique.

In the Fig. 1, binary tree [2] is constructed using Processing Elements (PEs). The first stage is constructed using PE1,PE2,PE3 and PE4. In the second stage, which constructed using PE5 and PE6 and the third stage is using PE7. The number of Processing Element (PE) required to construct this simple network is seven. So inorder to reduce the number of processing element proposing a new architecture known as Folded tree architecture, which reuse the processing element and reducing the area and power. . SAD is implemented using Binary tree, which is a tree structure and is made of four partual SAD [4].

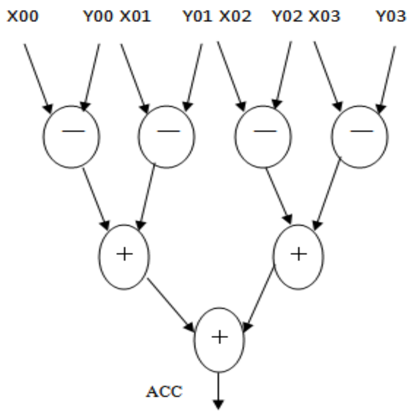


Fig. 2. A SAD Structure

A partual SAD consist of 4 substraction unit and 3 addition unit. The current coding pixels X00,X01,.... and the reference pixel Y00,Y01,..... are read from memory. The partual SAD value is accumulated in accumulator (ACC). The structure of SAD is similar to a tree, so it can implemented using binary tree and folded tree.

IV. PROPOSED METHOD

Parallel prefix operation is used for on-the-node data processing in wireless sensor network. Prefix operation can be calculated in different ways [6], due to the flow matches the desired on-the-node data aggregation, the binary tree and folded tree approach is prefered. On-the-node data aggregation can visualized in binary tree and folded tree of Processing Elements (PEs). The tree based data flow will executed on the data path of programmable PEs which provide the flexibility together with the parallel prefix operation.

A.Parallel Prefix Operation

Prefix Operation normally used in the world of digital design. Carry look-ahead adder [7] is the suitable example. The carry look-ahead adder consist of tree stages- Bitwise Propagate Generate (PG) stage, Group PG stage, Sum stage and A and B are the two inputs.The output of bitwise PG stage is Pi= Ai xor Bi ; Gi= Ai and Bi. For A={1001} and B={1010} then output P={0011} and G={1000}; the sum out finally obtaining depends upon the Cin, sum={0111}.

Blelloch's generic approach [2] is used for prefix calculation, which consist of two phase trunk phase and twig phase. In trunk phase, the left value L is saved locally as Lsave and it is added to the right value R, which is passed towards the root. This continues until the parallel-prefix element 15 is found at the root. Here a store-and-calculate operation is executed. . The twig-phase starts, during which data moves in the opposite direction, from the root to the leaves. The incoming value, beginning with the sum identity

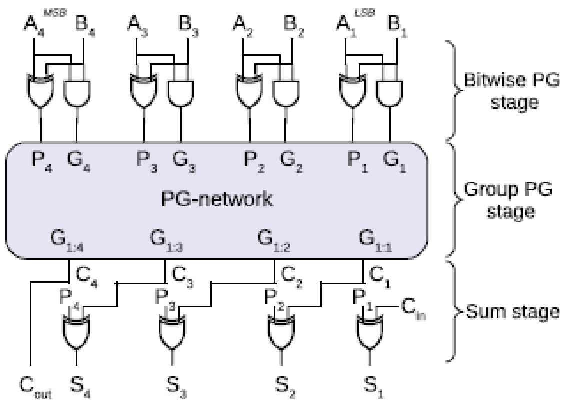


Fig.3. Carry look-ahead adder

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. Finally the reduced-prefix values is found at the leaves.

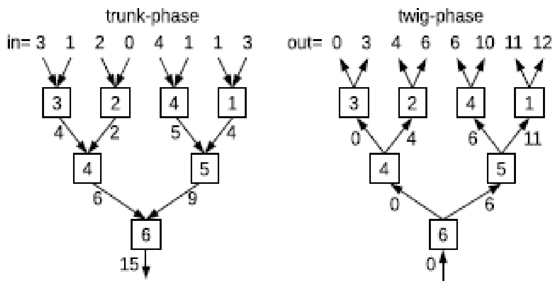


Fig. 4. Example of a prefix calculation with sum operator using Blelloch generic approach in a trunk and twig phase.

B. Folded Tree

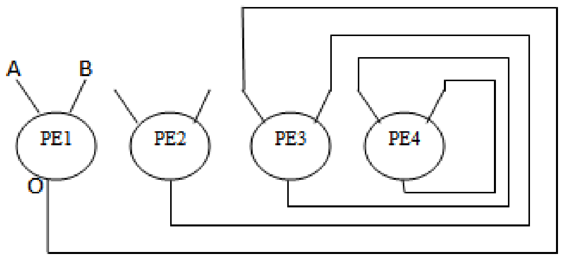


Fig. 5. Folded tree

The area and power is reduced by using the folded tree architecture. The idea here is to fold the tree back onto the itself to maximally reuse the PEs. In doing so,if n is the input, cost P becomes proportional to n/2 and the area is reduced to half. Note that also the interconnect is reduced. And 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 [8] , this leaves enough room for this trade off to be made.

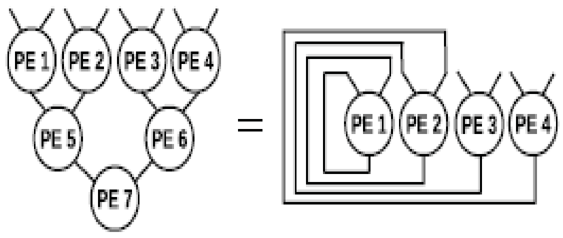


Fig. 6. A binary tree (left, 7 PEs) is functionally equivalent to the novel folded tree topology (right, 4 PEs) used in this architecture.

In folded tree which fold the tree back onto itself to maximally reuse the PEs, so the number of PEs is reduced . In the Fig. 5, the number of PEs reduced to four.

V. PROGRAMMING EXISTING AND PROPOSED APPROACH

The binary tree and folded tree architectures are programmed on VHDL.VHDL (VHSIC Hardware Description Language) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. VHDL can also be used as a general purpose parallel programming language.

A.Binary Tree

The binary tree in Fig. 6 consist of seven PEs, each stage consist of different PEs. The right and left input values are added , passed to the next stage and left input value is saved in PEs. For example in Fig. 4, trunk phase input applied to the PE1 is 3 and 1. The left value 3 is stored in the PE1, left input 3 and right input 1 is added and passed to the next stage. This process will continues until output value is obtained from PE7. Using this criteria and stuctural similarity with SAD motion estimation is implemented using binary tree. The programming of binary tree with two inputs, binary and decimal. But binary input is prefered because the number of bit in the input and output is reduced, so complexity of design is reduced.

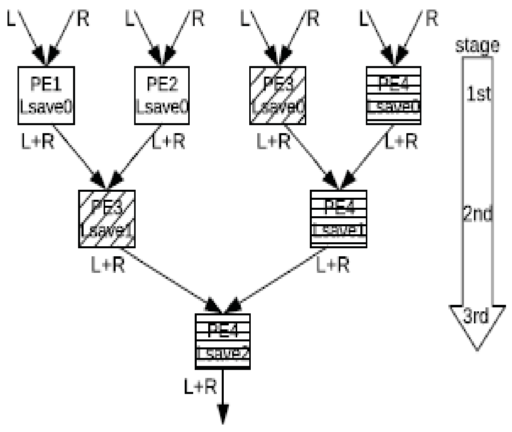


Fig. 7. Trunk Phase Implementation

B. Folded Tree

The Blelloch generic trunk phase is considered. 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.

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 implies to a decreasing number of active PEs while progressing from stage to stage. In the first stage, all four PEs are active. The second stage has two active PEs: PE3 and PE4. The third and last stage has only one PE is active: PE4.

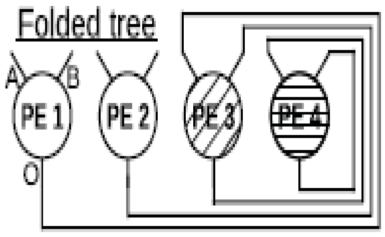


Fig. 8. Implications of using Folded tree

More importantly, it can 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 have only keep one: Lsave0. The trunk phase PE program has three instructions, which are identical, apart from the different RF addresses 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. That is why PE4 , PE3 and PE1 and PE2 needs three, two , one instruction respectively.

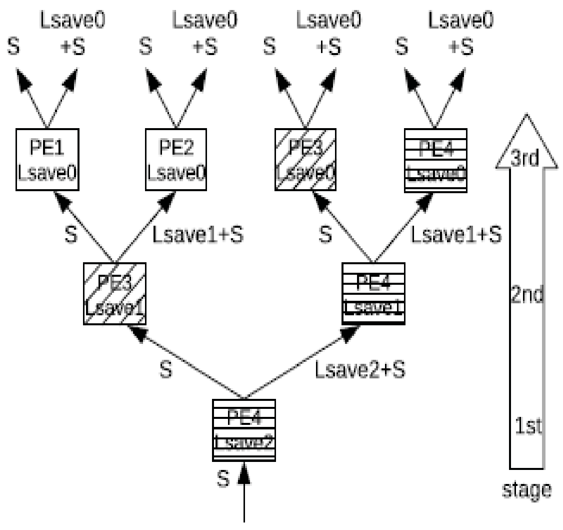


Fig. 9. Annotated twig phase graph of 4 PE folded tree

In twig phase the tree operates in the opposite direction. According to Blelloch's approach, S is passed to the left and the sum S+Lsave is passed to the right, 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 happen.

In Fig. 10, SAD converted to folded tree, here always the Blelloch's trunk phase is applied and programmed. Normal SAD algorithm is a type of binary tree, so programming and implementation is simple, but number of processing element is high. In the case of SAD with folded tree, the number of processing elements is reduced and motion is estimated.

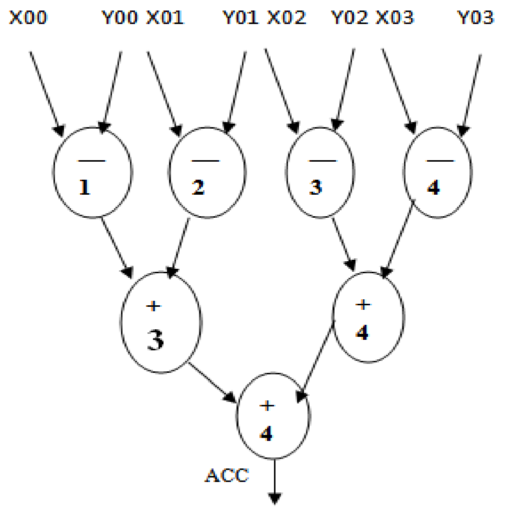
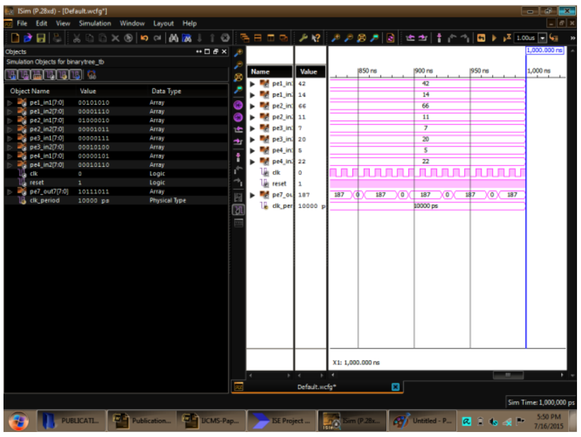


Fig. 10. SAD Algorithm with Folded Tree Architecture

VI. EXPERIMENTAL RESULTS


ISim (P.28xd) - [Default.wcfg*]

Simulation Objects for binarytree_tb

Object Name	Value	Data Type
pe1_in1[7:0]	00101010	Array
pe1_in2[7:0]	00001110	Array
pe2_in1[7:0]	01000010	Array
pe2_in2[7:0]	00001011	Array
pe3_in1[7:0]	00000111	Array
pe3_in2[7:0]	00010100	Array
pe4_in1[7:0]	00000101	Array
pe4_in2[7:0]	00010110	Array
clk	0	Logic
reset	1	Logic
pe7_out7[7:0]	10111011	Array
clk_period	10000 ps	Physical Type

Name	Value
pe1_in	42
pe1_in	14
pe2_in	66
pe2_in	11
pe3_in	7
pe3_in	20
pe4_in	5
pe4_in	22
clk	0
reset	1
pe7_ou	187
clk_per	10000 p

Fig. 11. Output of Binary tree architecture with decimal input

Fig. 11, shows binary tree of eight decimal input is applied and a single out is obtained. Fig .12 shows binary tree of eight binary input with four bit and a single output with seven bit. Fig. 13, shows the motion vector for binary tree architecture with SAD algorithm. Fig. 14 shows folded tree of eight decimal input is applied and a single out is obtained. Fig. 15 shows folded tree of eight binary input with four bit and a single output with seven bit. Fig.16 shows the motion vector for folded tree architecture with SAD algorithm. At 250ns the data was transmitted in folded tree with SAD. Hence the time requirement is low.

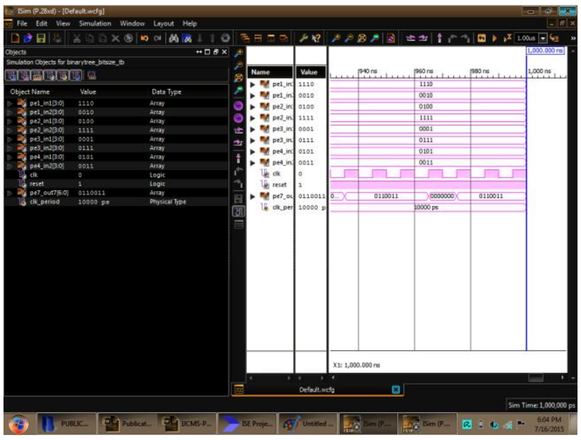


Fig. 12. Output of Binary tree architecture with binary input

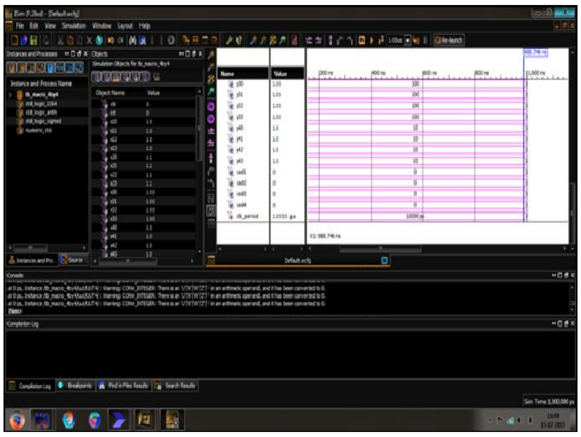


Fig. 13. Output of Binary tree architecture with SAD

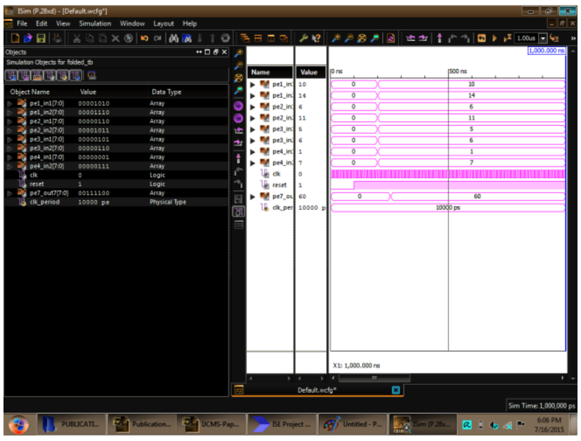


Fig. 14. Output of Folded tree architecture with decimal input

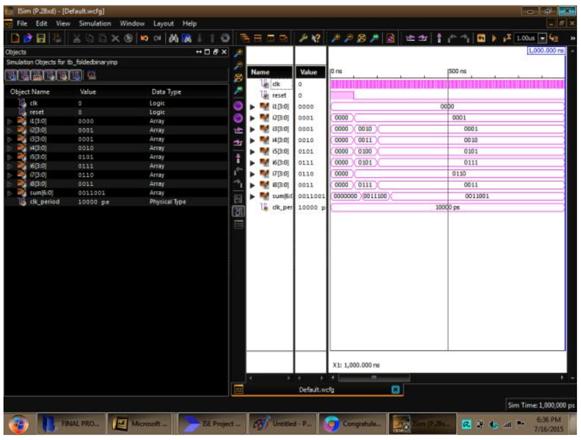


Fig. 15. Output of Folded tree architecture with binary input

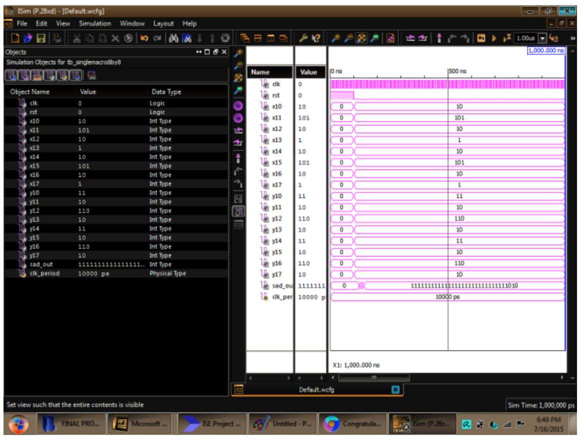


Fig. 16. Output of Folded tree architecture with SAD

TABLE I
DEVICE UTILIZATION SUMMARY OF BINARYTREE WITH SAD FOR MOTION ESTIMATION

Logic Utlization	Used	Available	Utlization
Number of Slices	956	2448	39%
Number of Slice Flip Flop	1056	4896	20%
Number of 4 Input LUTs	1324	4896	27%
Number of IOBs	896	108	829%
Number of GCLKs	1	24	4%

TABLE II
DEVICE UTILIZATION SUMMARY OF FOLDEDTREE WITH SAD FOR MOTION ESTIMATION

Logic Utlization	Used	Available	Utlization
Number of Slices	43	2448	1%
Number of Slice Flip Flop	47	4896	0%
Number of 4 Input LUTs	79	4896	1%
Number of IOBs	29	108	26%
Number of GCLKs	1	24	4%

Comparing Table I and Table II, the number of slices utlized by folded tree is very less compared to binary tree architecture. That is the area is reduced in folded tree. The number of slice flipflop required for binarytree 20% higher than folded tree and number of LUTs required for folded tree is very less. From this tables, reaches to the infornation that folded tree is more efficient than the binary tree.

TABLE III
PARAMETER COMPARISON

Architectures	Area in Percentage	Power in Percentage	Delay
Binary tree Architecture	39	65	107.743
Folded tree Architecture	1	57	71.537

The comparison table gives the binary tree consumes 39% area, folded tree consume only 3% area. Power consumption in both architecture is milliWatt range, so the measurement of power is so deficult. But number of device included in each architecture can measure. From this measurement power consumption is obtained. Binarytree utlizing 65% power, but the folded tree utlizing only 57% power. And the delay is always reduced from 107.743 in binarytree to 71.537 in folded tree.That is the Folded tree architecture with SAD algorithm for motion estimation is area efficient, low power and reducing delay.

VII.CONCLUSION

This paper presented the folded tree architecture with SAD algorithm for motion estimation. The SAD algorithm for motion estimation can be described using binary tree, but it utlizing high power and area. Reducing the area, power and delay in folded tree by reusing the PEs. The simplicity of the programmable PEs, which constitute the folded tree architecture resulted in high integration, reduced area, reduced delay and low power consumption. The binary tree can be reused as folded tree.

REFERENCES

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

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

[3] Cedric Walravens, 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, february 2014.

[4] S.C.Hsia, P.Y.Hong, " Very large scale integretion(VLSI) implementation of low-complexity variable block size motion estimation for H.264/AVC coding" , *IET Circuits Devices Syst.*, 2010, Vol. 4.Iss. 5, pp. 414-424.

[5] J. Hennessy and D. Patterson, *Computer Architecture A Quantitative Approach*, 4th ed. San Mateo, CA: Morgan Kaufmann, 2007.

[6] 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.

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

[8] 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.

[9] 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.

[10] H. Karl and A. Willig, *Protocols and Architectures for Wireless SensorNetworks*, 1st ed. New York: Wiley, 2005

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

[12] Yu-Wen Huang, Shao-Yi Chien, Bing-Yu Hsieh, and Liang-Gee Chen, "Global Elimination Algorithm and Architecture Design for Fast Block Matching Motion Estimation" , *IEEE transactions on circuits and systems for video technology*, vol. 14, no. 6, june 2004

[13] J. Vanne, et al., .A high-performance sum of absolute difference implementation for motion estimation,. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 876.883, July 2006.

[14] Y.S. Jehng, L.G. Chen and T.D. Chiueh, "An efficient and simple VLSI tree architecture for motion estimation algorithms," *IEEE Trans. on Signal Processin.g*, vol. 41, no.2,pp.889-900,Feb.1993.

[15] Y. H. Yeh and C. Y. Lee, "Cost-effective VLSI architectures and buffer size optimization for full-search block matching algorithms," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 345–358, Sept. 1999.