International Journal of Control Theory and Applications
Volume 11, Number 1, 2018, ISSN : 0974-5572
available at http: www.serialsjournals.com

DESIGN OF HIGH SPEED LOW POWER MULTIPLIER FOR SENSOR NODES

Saji. M. Antony[1], S. Indu[2] and Rajeshwari Pandey[3]

[1] *ECE Department, Bharati Vidyapeeth's College of Engineering, New Delhi, India*
E-mail: saji.antony@bharatividyapeeth.edu
[2, 3] *ECE Department, Delhi Technological University, New Delhi, India*
E-mails: [2]s.indu@dce.ac.in,[3]rajeshwaripandey@gmail.com

Abstract: The life of a sensor network is mainly determined by its energy consumption. Commercially available sensor nodes are battery driven devices. Due to large number of sensor nodes that may be deployed and longer life times required for the system, replacing battery is not an option. We can achieve energy optimization by considering energy consumption in design and operation of sensor nodes**.** In the architecture of sensor nodes, multipliers are the main structure for designing an energy efficient processor. The main reason for power dissipation in multiplier circuit is due to power dissipation of full adder circuit. Low power multipliers can be designed by using low power adders. In this paper Dadda multiplier with multiplexer based adder is proposed. Verilog HDL code has been simulated in MentorGraphics ModelSim Edition 10.4a. The synthesis of the multipliers has been done on Virtex-6 device using Xilinx ISE 14.7. The performance of multiplier has been obtained for the conventional adders as well as the logic optimized adder. The analysis of the results shows that the proposed method leads to reduction in the delay and LUT (Look Up Table) count (an indicator of area) of the multiplier.

Keywords: Dadda multiplier, MUX based full adder, carry save adder, High speed low power multiplier, Sensor nodes.

I. INTRODUCTION

Sensor networks have been identified as one of the most important technologies of the present century. Advances in wireless communications and micro-electro-mechanical systems (MEMS) have led to the development of smaller size sensor nodes that are low-cost and low power consuming. These multifunctional sensor nodes have capabilities of sensing, data processing, communication, storage etc. A sensor network consists of many such nodes deployed in a region to monitor a particular phenomenon. Sensor network is a fresh research area with applications in military, environment monitoring, disaster management etc. Sensors are used to measure and monitor parameters that may vary with place and time which prompts the need for the Dynamic Sensor Network (DSN). Block diagram of sensor node is as shown in Fig. 1[1]. Speed is an important factor to determine the performance of a processor. In real time applications like controlling and measuring various environmental conditions, fast response of the processor is required to process the measured signals. In arithmetic operations, an important fundamental function is Multiplication. Operations based on multiplication are frequently used in critcal applications of digital signal processing (DSP) like convolution, Fast Fourier Transform (FFT), filtering etc. These are also used in arithmetic functions like inner products, MAC (multiply and accumulate) units etc. Multiplier is an important unit in digital image processing systems. Multipliers are used not only in ALU but also in other components of processor implementation, like various data path units. High

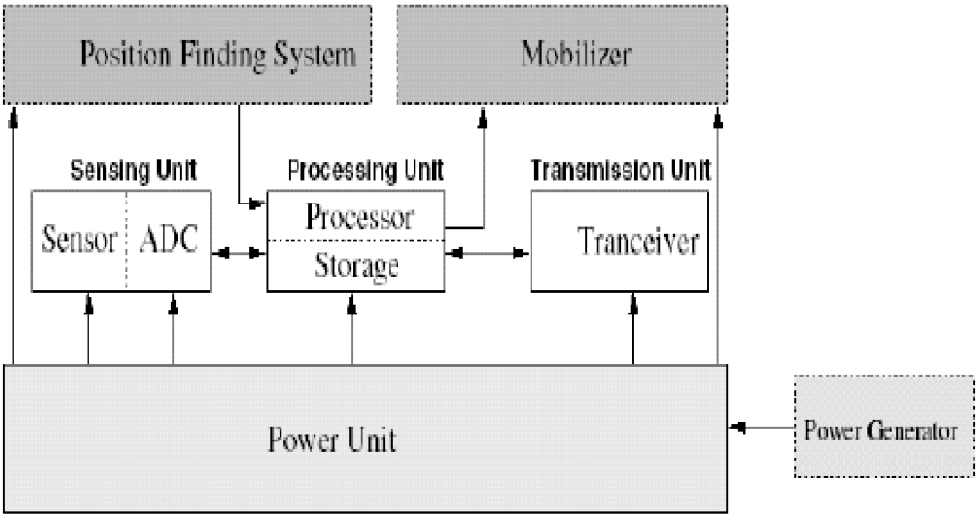


Figure 1: The components of a sensor node. [1]

speed processing is essential in real time applications, for processing of acquired signals. So demand for low power consuming multiplier circuit with high speed for sensor nodes has been continuously increasing [2].

This paper aims to develop a technique which can lead to the reduction in delay and the area, utilized by the Dadda multiplier. Since the speed of multiplier is mainly determined by the speed of adders employed for addition of partial products, to increase the speed of multiplier, it is proposed to use full adders designed with multiplexers. The multiplier is modeled as a digital system and described using Verilog HDL. Our aim is to reduce the delay and area of Dadda multiplier by replacing the basic structural unit of the multiplier, which is an adder. A modified multiplexer-based adder will be used instead of the conventional full adder to increase speed and reduce area [2].

II. RELATED WORK

Due to the demand of high speed processing and wide range of applications, a lot of research has been done on various multiplication algorithms and various kinds of adders. According to K. M. Mhaidat and A.Y. Hamzah[3], tree multiplier or parallel multiplier are faster but have more area and power consumption. Analysis of Wallace tree and the Dadda multiplier has been done on a variety of platforms. Using Leonardo spectrum, Lee *et al.* [4] compared the array multiplier, Wallace tree, and Dadda multiplier. Analysis shows that Dadda is not always faster than Wallace but uses minimum logic. Wallace tree is suited for high speed, only when area is not a priority. Swee and Hiung [5] have also implemented array, Wallace tree, Dadda and radix-4 booth multiplier on Leonardo spectrum on different synthesis platforms (speed optimized, area optimized and auto optimized). Wallace tree is fastest for speed-optimized whereas Dadda is fastest for area optimized. With DSCH-2 and microwind tools Anitha and Ramanathan have designed a hybrid multiplier using both Wallace tree and dadda multiplier which has less area and consumes lesser power [6]. The design of multipliers such as Dadda Multiplier (DM), Ripple Carry Multiplier with Row bypassing (RCM), Array Multiplier (AM), Wallace Tree Multiplier (WT), Modified Radix-2 booth multiplier (MRBM and Vedic Multiplier (VM) are discussed in reference [7]. For 8 bit multiplication, Dadda Multiplier (DM) has optimum performance results of delay and area, compared with corresponding conventional multiplier architectures. Dinesh et al [8]

have also compared regular multiplier and tree based multipliers and concluded that Dadda has considerable improvement over Wallace tree multiplier. Lots of efforts have been done to improve the delay, area and power performance of these multipliers. In Ref [9], tanner tool (350nm technology) is used to implement Wallace tree and dadda multiplier with the final stage ripple carry adder (RCA) replaced by a sklansky tree adder or a parallel prefix adder. Comparison of delay and power shows that dadda is better than Wallace tree.

Adders have an important role, in the implementation of an efficient multiplier. Weighted carry save adder is used by Park *et al.* [10] to reduce area due to the trapezoid structure of normal CSA multiplier. It reduces the addition of number of bits in the final stage. The consumption of power in carry save adder can be reduced by employing double pass transistor using asynchronous adiabatic logic by Bennet and Muflin [11]. A comparison between carry save adder, carry ripple adder, carry increment adder and carry look-ahead adder is done. It has been found out that CSA offers least delay. In [12], Javali et al have replaced the final stage RCA with using a carry look-ahead adder. This causes a trade off between area and speed i.e. the delay reduces but area increases. A modified carry save adder is proposed by Mahalakshmi and Sailatha [13]. In place of ripple carry adder, parallel multiplexers are used in a specific way to reduce the propagation delay of the final stage (divided in groups so that each can be processed parallel).

The basic unit of the multiplier is an adder, so delay of multiplier can be reduced by reducing the delay of adder. In ref [14], Nimmagadda and Pal have proposed a full adder using transmission gates and have shown that the performance is better in terms of energy consumption and delay. Non-volatile full adder using race track memory is implemented which reduces power by 5.9 times and area by 50%. Resistor coupled Josephson logic is used to make a 4 bit full adder which has very high speed. Dynamic full adder and transmission gate full adder topology is mixed, which causes very fast carry computation [15-17]. In [18], different configurations of CMOS full adders have been proposed and simulated using HSPICE which show difference in their delay parameters. Instead of bulk CMOS, FINFET is used by Rapolu and Nikoubin [19] for ultra low power design of the full adder, which shows a reduction in the delay. A reduced complexity Wallace tree multiplier with lesser area and power dissipation is implemented by Khan et al [20], employing energy efficient CMOS full adder in place of conventional full adder. This causes power and gate count to decrease. Uma and Dhavachelvan have analyzed various kinds of implementation of a simple full adder by using different Boolean equations possible for the outputs of the full adder. Concept of logical effort has been used to bring out the differences in performance of the various full adder configurations [21]. This concept has been used in this work so that the effect of using different full adder configurations can be seen on the performance of the multipliers. The proposed work presents a gate-level analysis using Xilinx ISE 14.7.

III. PROBLEM FORMULATION

The aim of this work is to implement a high speed and low power multiplier, which is suitable for sensor nodes. It is proposed to integrate the high speed Dadda multiplier, which is fastest for area optimization and MUX based adders for delay minimization.

A. DADDA Multiplier

The Dadda multiplier was invented by a scientist Luigi Dadda [22] in 1965. It is resembling the Wallace tree multiplier, but is little faster than the latter and need lesser number of gates (for all but the smallest operand sizes).

The Dadda multiplier has following three steps:

1. Multiply (AND) each multiplier bit with corresponding multiplicand bit. Thus for n-bit multiplier and multiplicand, this step yields n^2 bits as result. The n^2 bits are arranged as n partial products, each of n-bits in the same way as is done in the usual manual multiplication.
2. We start with n partial products in this step. Carry save addition is employed to decrease the number of partial products in every stage. In each stage, the reduction is done such that the ratio of partial products in the current and the next stage should not be greater than 1.5. This step is continued till the partial products are reduced to two.
3. The two numbers are finally added using conventional adders like the ripple carry adder etc. to obtain the final product.

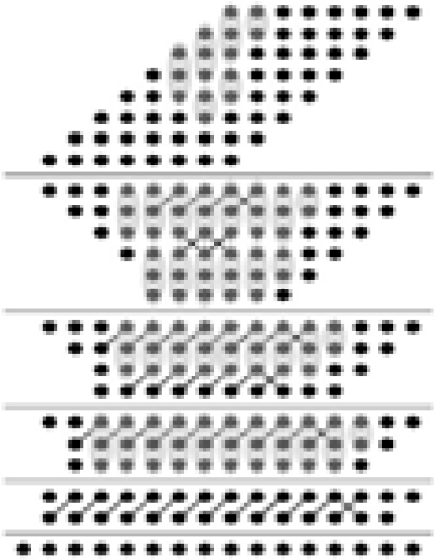


Figure 2: Dadda Mutiplication Process

B. Carry save Adder (CSA)

A carry-save adder (CSA) is used to calculate the sum of three or more n-bit binary numbers. The main difference between CSA and other adders is that, its partial sum bits and carry bits outputs have the same dimensions as the inputs (n-bits). Thus the CSA is a high-speed,multi-operand adder.

Multi-operand addition is often required in multiplication and division. Thus we require adders, which can add more than two numbers at a time. In a CSA, the carry is saved rather than propagate. This means that we save(store) the carry-out instead of using it immediately to compute a final sum. Carry save adder is ideal to use with several operands together. Thus it can increase the computation speed by saving the time spent in carry propagation.

The basic carry save adder (CSA) takes three n-bit inputs as operands and generates two n-bit outputs, one n-bit sum, and one n-bit carry. A CSA thus, reduces the number of operands to be added from 3 to 2, without any propagation of carry. If we want to obtain the normal addition result of the operands, then we have to combine sum and carry. A second CSA receives these two bit sequences as inputs (sum and carry of the first three operands) and a first operand as input, and produces a new

sum and carry. This technique is very powerful as any number of operands can be added together using this method. A carry propagating addition like the ripple carry addition is required for the recombination of the final carry and sum to produce the (n+1) bit result.

The delay for n-bit is equal to delay of a single full adder as the full adders are not connected to each other and hence no delay due to the propagation of the carry signal is caused. This can be seen in fig. 3.

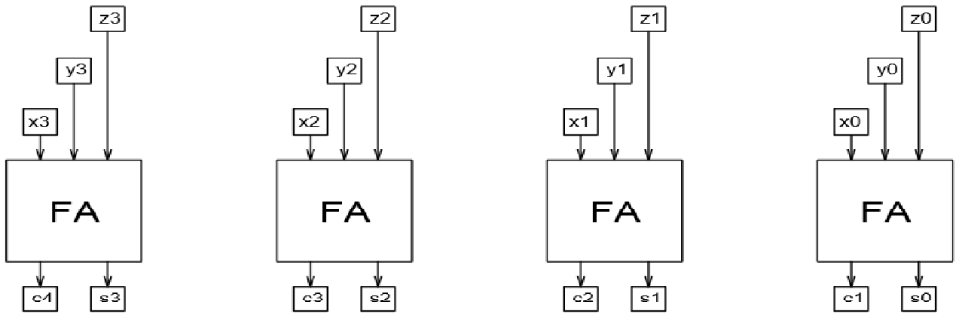


Figure 3: Carry Save Addition Principle

If the purpose of the adder is to add two numbers and produce a result, the carry-save addition is useless, because the two results (sum and carry) will have to be converted to a single sum which means that the carries have to propagate from right to left. CSA is mostly used to accumulate partial sums in a multiplication.

The carry-save adder made of n full adders. Each full adder gives a single sum and carry bit outputs, based only on the corresponding bits of the three input operands. For three n bit numbers x, y, z, it produces a partial sum s and a carry c which are given by the following equations.

$$Si = x_i \oplus y_i \oplus z_i \tag{1}$$

$$c_{i+1} = (x_i \wedge y_i) \vee (x_i \wedge z_i) \vee (y_i \wedge z_i) \tag{2}$$

IV. MUX BASED ADDERS

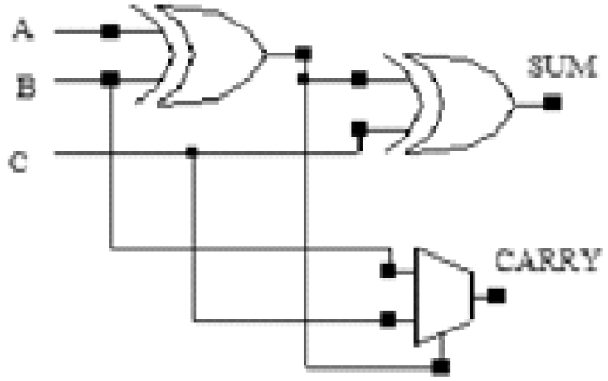
Adders are important units for the implementation of multipliers with highefficiency. Reduction in the delay of full adder results in increased speed of the multiplier. As shown in [21], [23] and [24] it can be concluded that full adder with XOR gate and multiplexer gives best performance, in terms of delay and power dissipation. It contains one 2x1 MUX and two XOR gates as given in Fig. 4.

Using logical effort method, we can obtain delay mathematically rather than employing simulation tools. This results in a simple way to decide the best logical construct or topology [21]. Delay (d) for a single stage network can be calculated by using equation (3), where 'h' would be the electrical effort, 'g'the logical effort and 'p' parasitic delay.

$$d = g^*h + p; \tag{3}$$

in which 'h' denotes the ratio of output capacitance to input capacitance and 'g' denotes the output current production ability of logic gate (how much better or worse the output current can be produced by the logic gate, compared to inverter) and 'p' represents delay of gate due to internal capacitance. Table I shows that, for any number of inputs, though the logical effort of MUX will be constant (equals

to 2), the logical effort for NOR, NAND and XOR gate will be higher, which are used in conventional full adder circuits. Thus, full adder circuit implemented with MUX results in minimum delay when compared with conventional full adders.



Sum $S = A \oplus B \oplus C;$

Carry $C_{out} = \overline{(A \oplus B)}\ .B + (A \oplus B).C$

Figure 4: Full adder with MUX and XOR gates [10]

Table I
Logical Effort Against Various Cmos **Gate Inputs**

		Gate Type			
No of Inputs	*Inverter*	*NAND*	*NOR*	*XOR(parity)*	*Multiplexer*
1	1				
2		4/3	5/3	4	2
3		5/3	7/3	12	2
4		6/3	9/3	32	2
5		7/3	11/3		2
6		(n+2)/3	(2n+1)/3		2

V. RESULTS

A. Simulation

The simulation platform used is Modelsim by Mentorgraphics. The Verilog code is synthesized and implemented on the Virtex 6 FPGA family using Xilinx ISE Design Suite v 14.7. Structural style of modeling has been used.

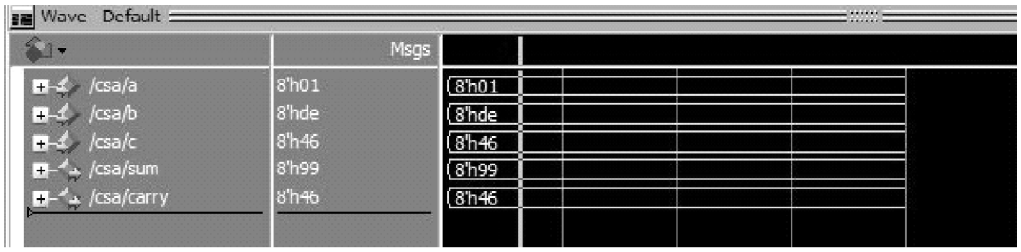


Figure 5: CSA 8 bit

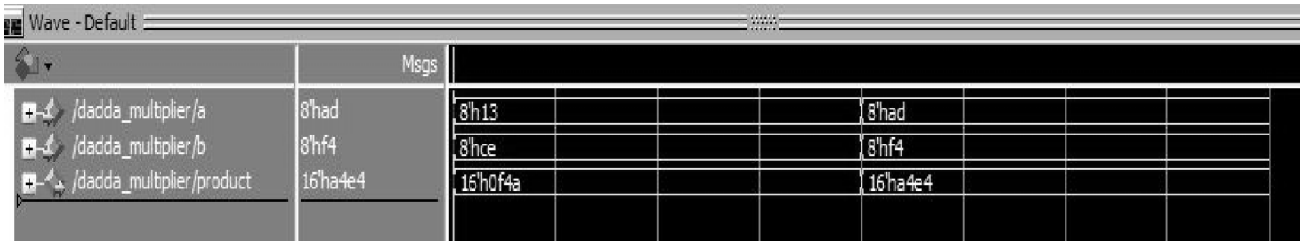


Figure 6: Dadda 8bit

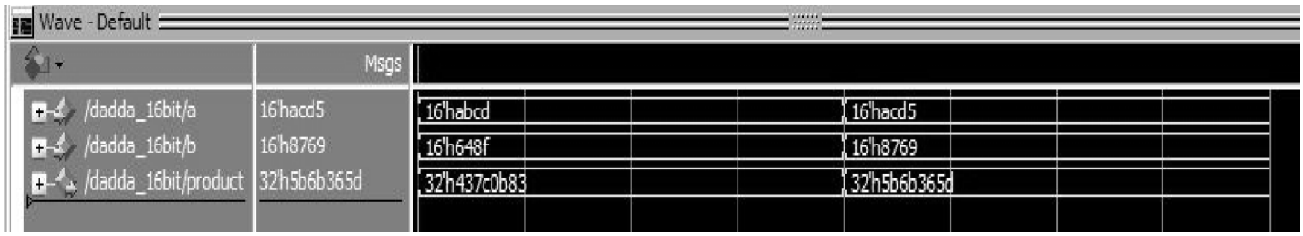


Figure 7: Dadda 16 bit

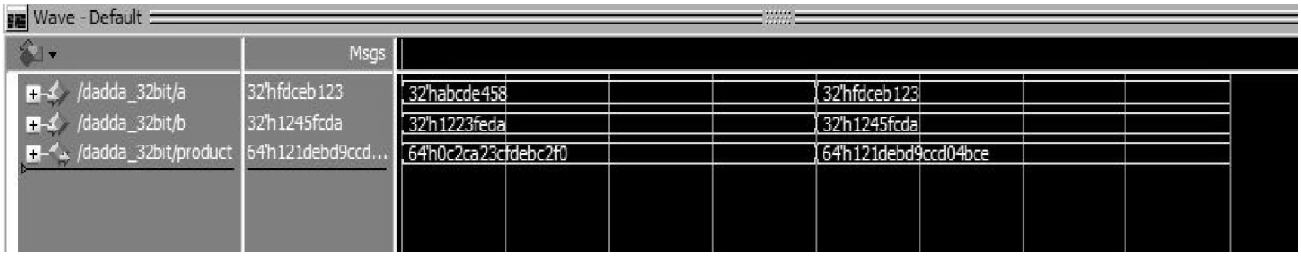


Figure 8: Dadda 32 bit

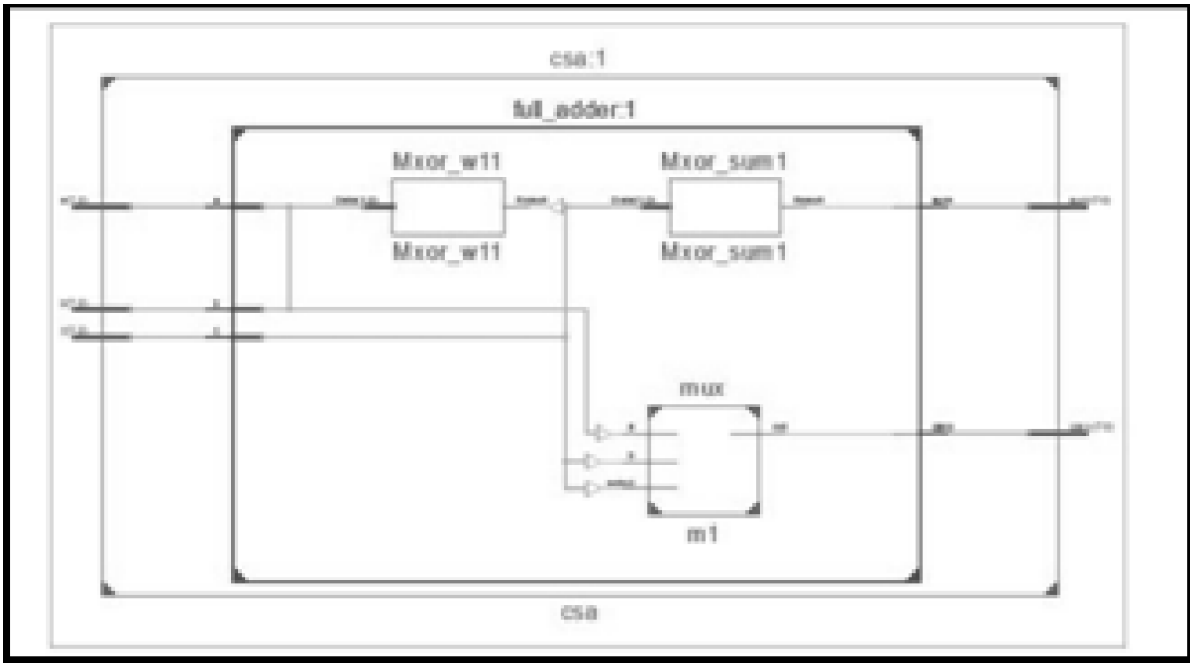


Figure 9: CSA Inner Block Diagram

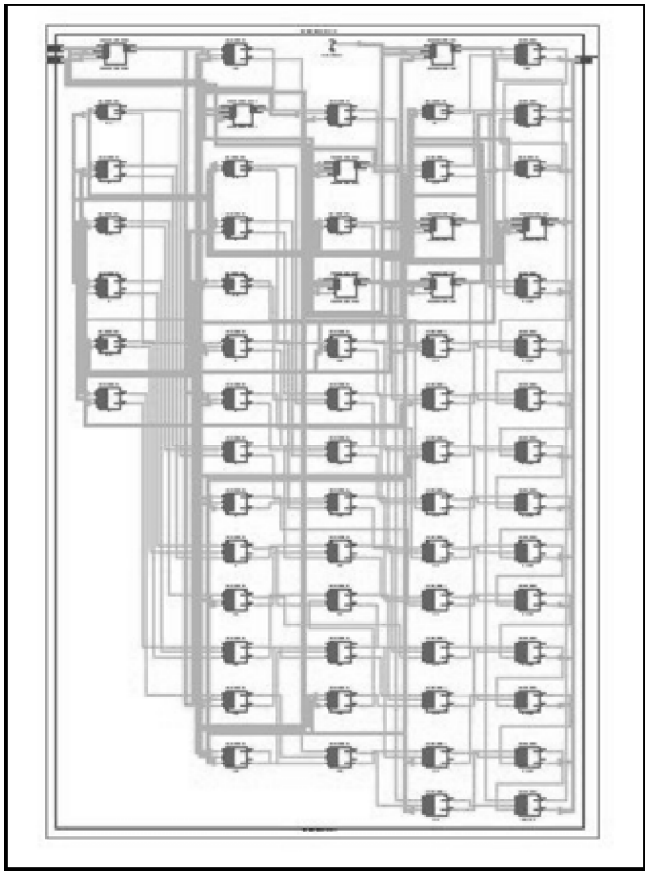


Figure 10: Dadda 8 bit Multiplier Inner Block Diagram

Table I
Delay Comparison for 8, 16 and 32 bit Dadda Multiplier

Type of full adder used	*8-bit (ns)*	*16-bit (ns)*	*32-bit (ns)*
XOR, AND, OR	9.693	20.038	40.780
XOR, MUX (PROPOSED MODEL)	8.474	17.034	37.317

Table II
Area Comparison (lut) for 8, 16 and 32 bit Dadda Multiplier

Type of full adder used	*8-bit*	*16-bit*	*32-bit*
XOR, AND, OR	106	485	2020
XOR, MUX (PROPOSED MODEL)	103	471	1986

VI. CONCLUSION

This paper proposes to use a high speed Dadda multiplier with MUX based full adders. Proposed design gives much less delay with less LUTs, compared with conventional Dadda multipliers. This multiplier can be used for processors in sensor nodes. For future work, instead of other conventional designs ALU and MAC unit can be designed with the proposed multiplier.

REFERENCES

[1] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey, In IEEE Wireless Communications, Volume 11, No.6, pp. 6-28, 2004.

[2] S. M. Antony, S.S.R Prasanthi, S.Indu and R.Pandey "Design of High Speed Vedic Multiplier Using Multiplexer Based Adder" International Conference on Control Communication and Computing India(ICCC), DOI : 10.1109/ICCC.2015.7432938, pp:448-453, Nov 2015.

[3] K. M. Mhaidat and A. Y. Hamzah, "*A New Efficient Reduction Scheme To Implement Tree Multipliers On FPGAs*", 2014 9th Int. Design and Test Symp. (IDT), Algiers, Dec. 2014, pp. 180-184.

[4] C. Y. H. Lee et al., "*A Performance Comparison Study On Multiplier Designs*", Intelligent and Advanced Systems (ICIAS), 2010 Int. Conf. on, Kuala Lumpur, Jun. 2010, pp. 1-6.

[5] K. L. S. Swee and L. H. Hiung, "*Performance Comparison Review Of 32-Bit Multiplier Designs*", Intelligent and Advanced Systems (ICIAS), 2012 4th Int. Conf. on, Kuala Lumpur, Jun. 2012, pp. 836-841.

[6] P . Anitha and P . Ramanathan, "*A New Hybrid Multiplier Using Dadda And Wallace Method*", Electronics and Communication Systems (ICECS), 2014 Int. Conf. on, Coimbatore, Feb. 2014, pp. 1-4.

[7] M. Sai Kumar *et al.*, "*Design And Performance Analysis of Multiply-Accumulate (MAC) Unit*", Circuit, Power and Computing Technologies (ICCPCT), 2014 Int. Conf. on, Nagercoil, Mar. 2014, pp. 1084-1089.

[8] B. Dinesh *et al.,* "*Comparison of Regular and Tree Based Multiplier Architectures with Modified Booth Encoding for 4 Bits on Layout Level Using 45nm Technology*", Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 Int. Conf. on, Coimbatore, Mar. 2014, pp. 1-6.

[9] T. Arunachalam and S. Kirubaveni, "*Analysis of High Speed Multipliers*", Communications and Signal Processing (ICCSP), 2013 Int. Conf. on, Melmaruvathur, Apr. 2013, pp. 211-214.

[10] Bong-II Park *et al.,* "*A Regular Layout Structured Multiplier Based On Weighted Carry- Save Adders*", Computer Design, 1999. (ICCD '99) Int. Conf. on, Austin, TX, Dec. 1999, pp. 243-248.

[11] B. Bennet and S. Maflin, "*Modified Energy Efficient Carry Save Adder*", Circuit, Power and Computing Technologies (ICCPCT), 2015 Int. Conf. on, Nagercoil, Mar. 2015, pp. 1-4.

[12] R. A. Javali *et al.*, "*Design of High Speed Carry Save Adder Using Carry Lookahead Adder*", Circuits, Communication, Control and Computing (I4C), 2014 Int. Conf. on, Bangalore, Nov. 2014, pp. 33-36.

[13] R. Mahalakshmi and T. Sasilatha, "*A Power Efficient Carry Save Adder And Modified Carry Save Adder Using CMOS Technology*", Computational Intelligence and Computing Research (ICCIC), 2013 IEEE Int. Conf. on, Enathi, Dec. 2013, pp. 1-5.

[14] M. R. Nimmagadda and A. Pal, "*Low Power, Energy-Efficient Full Adder for Deep-Submicron Design*", IEEE Computer Society Annu. Symp. on VLSI, Chennai, July 2011, pp. 345-346.

[15] K. Huang *et al.,* "*A Low Power and High Sensing Margin Non-Volatile Full Adder Using Racetrack Memory*", IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, pp. 1109-1116, Feb. 2015.

[16] J. Sone *et al.*, "*High Speed Four-Bit Full Adder With Resistor Coupled Josephson Logic*", Electron Devices Meeting, 1983 International, Washington DC, U.S.A., Dec. 1983, pp. 682-685.

[17] M. Alioto and G. Palumbo, "*Very High-Speed Carry Computation Based on Mixed Dynamic/Transmission-Gate Full Adders*", Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conf. on, Seville, Aug. 2007, pp. 799-802.

[18] T. M. Shafiqul Khalid, "A *Fast Optimal CMOS Full Adder*", Circuits and Systems, IEEE 39th Midwest Symp. on, Ames, IA, Aug. 1996, pp. 91-93 vol 1.

[19] S. Rapolu and T. Nikoubin, "*Fast and Energy Efficient Finfet Full Adders with Cell Design Methodology (CDM)*", 2015 6th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), Denton, TX, July 2015, pp. 1-5.

[20] S. Khan *et al.,* "*Performance Analysis Of Reduced Complexity Wallace Multiplier Using Energy Efficient CMOS Full Adder*", Renewable Energy and Sustainable Energy (ICRESE), 2013 Int. Conf. on, Coimbatore, Dec. 2013, pp. 243-247.

[21] R.Uma and P.Dhavachelvan, "*Logic Optimization Using Technology Independent Mux Based Adders In FPGA*", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.4, August 2012.

[22] L. Dadda, "Some schemes for parallel multipliers", Alta Frequenza, vol. 34, pp. 349–356, 1965.

[23] Maroju Sai Kumar, Dr. P. Samundiswary "Design and Performance Analysis of Various Adders using Verilog", IJCSMC, Vol. 2, Issue. 9, September 2013, pp.128 –138.

[24] N. Prathima, K. HariKishore, "Design of a low power and high performance digital multiplier using a novel 8T adder", (IJERA),Vol. 3, Issue 1, January -February 2013, pp.1832-1837.