FPGA IMPLEMENTATION OF MULTIPLIER USING SHIFT AND ADD TECHNIQUE

[1]S.V.PADMAJARANI, [2]M.MURALIDHAR

[1]Professor and HOD(ECE), [2]Principal, SreeVenkateswara College of Engineering and Technology, Chittoor, A.P, India
E-mail: [1]sv.padmajarani@gmail.com, [2]muralidhar6666@gmail.com

Abstract - Multipliers are one of the most important parts in signal processing or other computationally intensive applications. Therefore, designing multipliers that are high speed, low power and less area of substantial research interest. Many attempts have been made to reduce the number of partial products generated by multiplication process. The aim of this paper is to Implement a Multiplier block using shift and Add technique of multiplication in an FPGA. The implementation is done by using Xilinx 14.5 version of VHDL with the targeted device of Spartan 3E. The performance of Multiplier unit is evaluated for various parallel prefix adder variants, which are developed for high speed addition. The experimental results shows that the implemented Multiplier using hybrid parallel prefix adder is efficient in area, consume low power and high speed compared to existing parallel prefix adder models.

Keywords - Multiplier, Shift and Add, Field Programmable Gate Array (FPGA), Digital Signal Processing (DSP), Parallel Prefix Adder (PPA).

I. INTRODUCTION

In many digital signal processing operations – such as correlations, convolution, filtering and frequency analysis – one needs to perform multiplication. Multiplication algorithms will be used to illustrate methods of designing different cells so that they fit into larger structures[2]. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low power multiplier design has been an important part in low power VLSI system design [4]. There has been extensive work on low power multipliers at technology, physical, circuit and logic levels. A systems performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. Furthermore, it is generally the most area consuming. Hence optimizing the speed and area of the multiplier is a major design issue. However area and speed are usually conflicting constraints so that improving speed results mostly in large areas. As a result a whole spectrum of multipliers with different area-speed constraints has been designed with fully parallel structures [5-7].

The rest of the paper is organized as follows: In section II, some background information about the process of multiplication and various types of multipliers are given. In section III, multiplication using shift and add technique is discussed. The section IV describes various parallel prefix adder architectures and their implementation. Experimental results are presented in section V. conclusions and Future scope are drawn in section VI.

II. BACK GROUND

The multiplication process may be viewed to consist of the following two steps: 1. Evaluation of partial products, 2. Accumulation of the Shifted partial product. It should be noted that binary multiplication is equivalent to a logical AND operation. Thus evaluation of partial products consists of the logical ANDing of the multiplicand and the relevant multiplier bit. Each column of partial products must then be added and, if necessary, any carry values passed to the next column [2]. Multiplication schemes are commonly classified in three general types: sequential, parallel and array multipliers. This is not a universal classification and some hybrid multiplication schemes do not fall into exactly are of these categories. For example, as a compromise between sequential and parallel multipliers, partially combinational multiplier, are introduced to achieve higher performance but still keep the hardware small [12].

Sequential multipliers generate the partial products sequentially and add each newly generated partial product to the previously accumulated sum. The sequential multipliers were popular when the hardware was expensive and bulky. They are still in use in applications where the speed is not critical. Shift and Add multiplication is an example of sequential multiplier [11]. The parallel multipliers generate all partial products in parallel and then use fast multi-operand adders for their accumulation [12]. Array multipliers are essentially regular structures and are similar to expand. In array multiplication, we need to add as many partial products as there are multiplier bits [5]. Wallace tree multiplier is considerably faster than a simple array multiplier. The Wallace tree's wiring is much less regular and more complicated. As a result, Wallace Trees are often avoided by designers, while design complexity is a concern to them. Wallace tree styles are generally avoided for low power applications[4,5].Booth multiplier can be used in different modes such as radix-2, radix-4, radix-8, etc. modified Booth Encoding (MBE) is most often used to avoid variable

size partial product arrays[2,3,5,12]. Baugh Wooley multiplier is used for 2's compliment multiplication. It adjusts the partial products to maximize regularity of the multiplication array [5]. The FPGA implementation of multipliers using Hardware Description Language and their performance comparisons are given in literature [8-10].

Multipliers are in fact complex adder arrays. This is an operation common to a large number of applications and the complexity of this function has lead to a large amount of research directed at speeding up its execution. Multipliers can be implemented using different algorithms. In the implementation of digital multipliers binary adders are essential components. The researchers are focused to evaluate and enhance the performance of adders, to improve the overall characteristics of multipliers [6,7].

This paper focuses on the implementation of Multiplier block that increases the speed of multipliers as well as reducing the area and power among the suitable architecture. The implementation of Multiplier block using Shift-and Add technique is discussed in next chapter.

III. MULTIPLICATION USING SHIFT AND ADD TECHNIQUE

Shift and Add technique of multiplication is a kind of sequential multiplication. Shift-and-add multiplication is similar to the multiplication performed by paper and pencil. To multiply two numbers by paper and pencil, the algorithm is to take the digits of multiplier one at a time from right to left, multiplying the multiplicand by a single digit of the multiplier and placing the intermediate product in the appropriate positions to the left of the earlier results.

In the case of binary multiplication, since the digits are 0 and 1, each step of the multiplication is simple. If the amplification digit is 1, a copy of the multiplicand (1Xmultiplicand) is placed in the proper positions; if the multiplier digit is 0, a number of 0 digits (0Xmultiplicand) are placed in the proper positions. The Circuit of shift-and-add method of multiplication for two n-bit numbers is shown in figure 1.

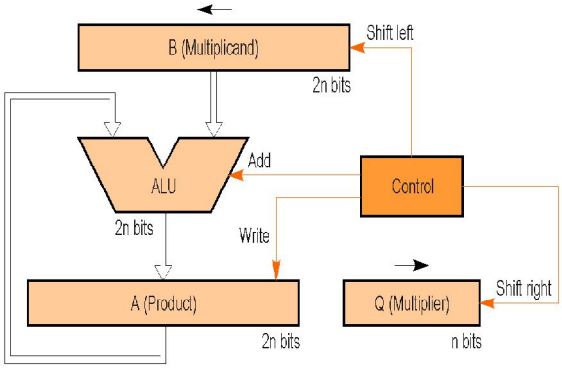

Figure 1 . A multiplier Circuit of Shift-and-Add method of multiplication for two n-bit numbers

The 2n-bit product register (A) is initialized to 0. Since the basic algorithm shifts the multiplicand register (B) left one position each step to align the multiplicand with the sum being accumulated in the product register, we use a 2n-bit multiplicand register with the multiplicand placed in the right half of the register and with 0 in the left half.

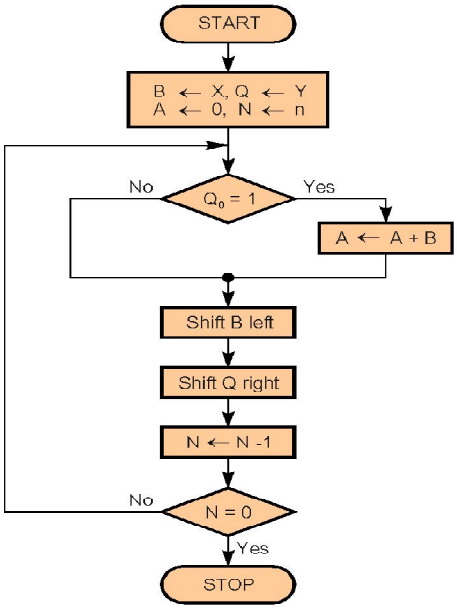

Figure 2. The multiplication algorithm

Figure 2, shows the basic steps needed for the multiplication. The algorithm starts by loading the multiplicand into the Bregister, loading the multiplier into the Qregister, and initializing the Aregister to 0. The counter Nis initialized to n. The least significant bit of the multiplier register (Q_0) determines whether the multiplicand is added to the product register. The left shift of the multiplicand has the effect of shifting the intermediate products to the left, just as when multiplying by paper and pencil. The right shift of the multiplier prepares the next bit of the multiplier to examine in the following iteration.

In the implementation of multiplier block, addition is performed by the parallel prefix adders, because parallel prefix adders are designed for high speed computing, which will speed up the multiplication process also. For shifting operation, barrel shifter is used, by loading it initially with the required value i.e the upper half with zeros and lower half with multiplicand bits.

IV. PARALLEL PREFIX ADDERS

The main disadvantage of conventional ripple carry adder structure is longest path delay. The parallel prefix adders are developed for high sped operation of addition operation. Various parallel prefix adders are proposed in literature of different constraints like area, timing, logic depth, fan-out capability and logic levels of implementation [14-17].

In every bit(i) of the two operand block, the two input signals (a_i and b_i) are added to the corresponding carry-in-signal ($carry_i$) to produce sum output (sum_i). The equation to produce the sum output is:

$$sum_i = a_i \oplus b_i \oplus carry_i \ldots. (1)$$

Parallel prefix adder architectures contain tree like structures which compute the required carry input ($carry_i$) for every bit of addition. The tree structures are developed by using different operators, which receive generate and propagate signals of previous level and compute generate and propagate signals for next level [14,15]. The realization of these operators, for enhancement of speed characteristic is also developed [16]. The hybrid architecture of parallel prefix tree is presented [17] for high speed area efficient characteristics.

In this paper the following parallel prefix adders are considered for the implementation.

*Brunt-Kung Adder(BK Adder)
*SkalanskyAdder(SK Adder)
*Kogge-Stone Adder(KS Adder)
*Han-Carlson Adder(HC Adder)
*Ladner-Fischer Adder(LF Adder)
*Knowles Adder(Kn Adder)
*Hybrid Adder

The architecture of 16-bit Hybrid adder with four types of operators is shown in figure 3 [17].

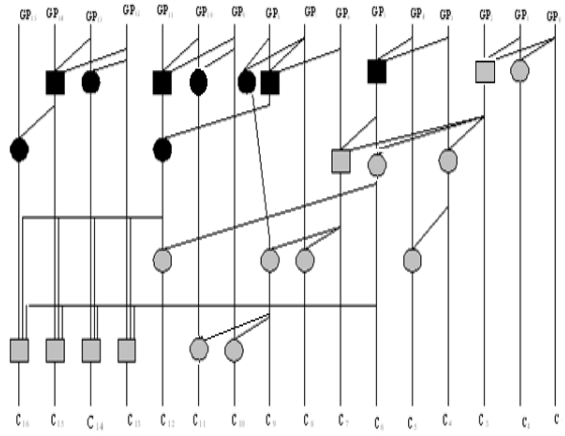

Figure 3. Hybrid 16-bit parallel prefix adder [15]

V. EXPERIMENTAL RESULTS

The Multiplier using Shift and Add technique is simulated using Xilinx 14.5 version by writing VHDL Code and implemented with the targeted device XC3S500E. The multiplier is designed for 8-bit wide operands. The addition operation is done by using parallel prefix adder (16-bit). The performance of Multiplier block is tested for various parallel prefix adder variants such as BK, Skalansky, KS, HC, LF, Knowles models and also the proposed hybrid model of parallel prefix adder architecture [17].

The results of Multiplier block, based on various parallel prefix adder variants are tabulated in table 1. The comparison is done for three factors: speed, area and power consumption. The speed performance is evaluated with respect to delay parameter. The area requirement can be estimated from the utilization of number of slices and Look-up tables. The power consumption is analyzed by taking switching power (dynamic power) in account which mainly depends on the input test vectors that can be applied through the test bench. Static power is not considered because lack of ASIC tools available.

Table 1. Results of Multiplier block with various parallel prefix adder variants

Adder type	Slices	LUT	Number of logic levels	Average fan out	Delay (ns)	Power (mw)
BK	98	184	20	3.49	22.357	46
Skalansky	104	191	22	3.64	23.955	46
KS	178	343	18	3.44	23.365	49
HC	118	220	17	3.30	21.772	47
LF	103	194	23	3.71	24.593	47
Knowles	185	347	18	3.54	22.586	48
Hybrid	117	225	18	3.34	22.767	47

The design summary of synthesis, simulation results and power analysis results of hybrid parallel prefix added based, Multiplier are given in Figure 4,5 and 6 respectively.

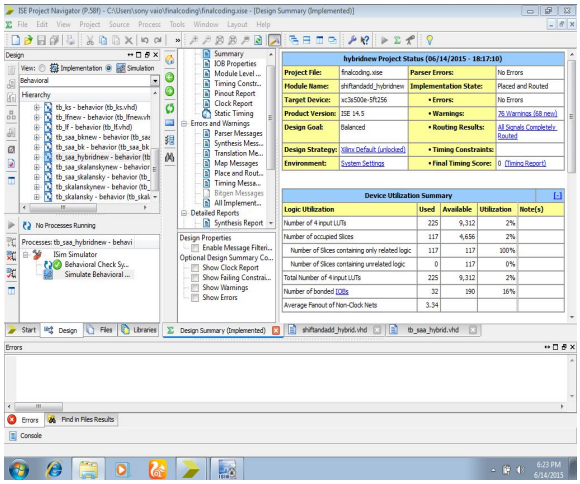

Figure 4. Design Summary of hybrid parallel prefix added based Multiplier

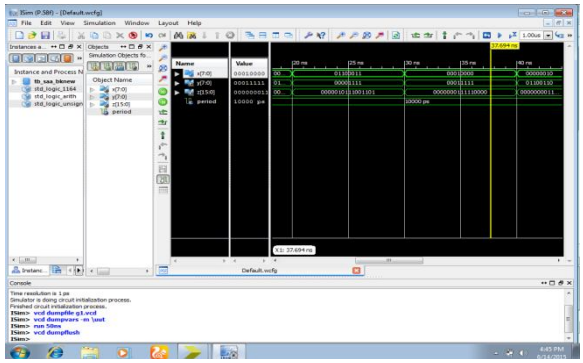

Figure 5. Simulation results of hybrid parallel prefix added based Multiplier

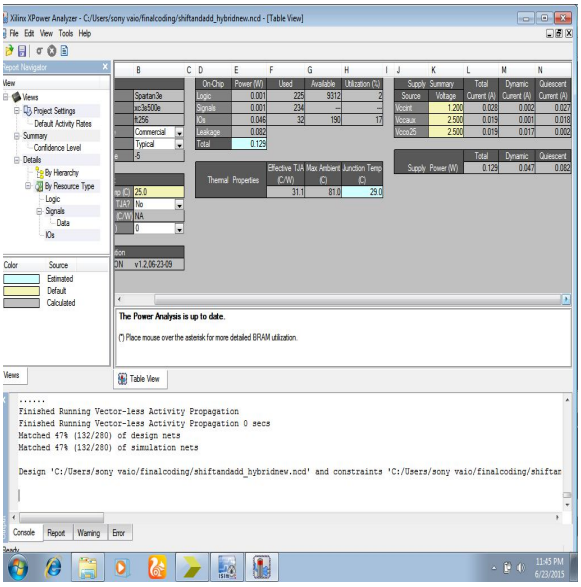

Figure 6. Power analysis results of hybrid parallel prefix added based Multiplier

For better comparison of the results of Multiplier based on various parallel prefix adder implementation are presented in figures 7 to 12.

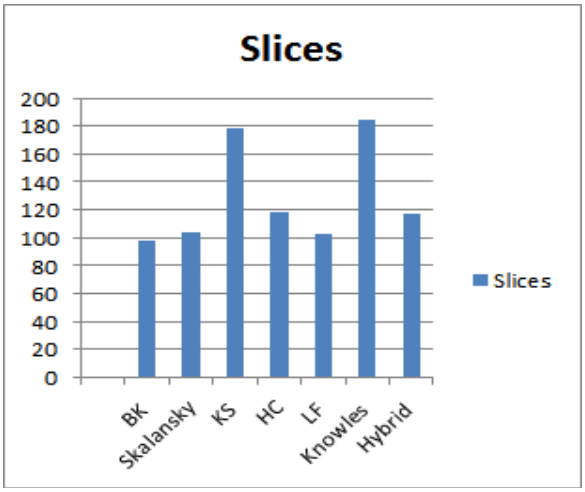

Figure 7. Utilization of Slices of Multiplier with various adder variants

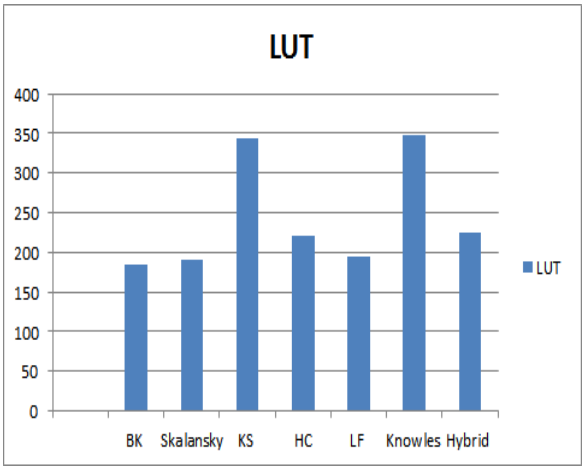

Figure 8. Utilization of LUTs of Multiplier with various adder variants

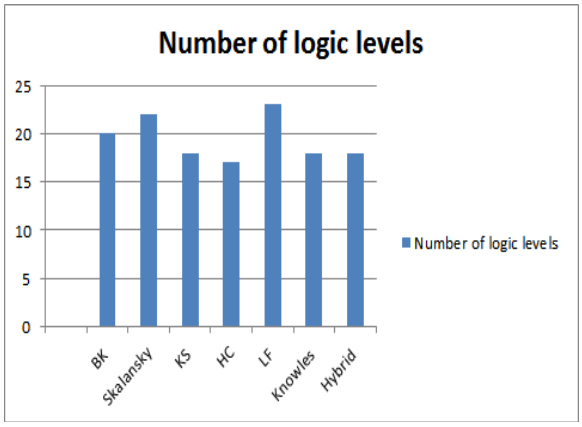

Figure 9. Number of Logic levels of implementation of Multiplier with various adder variants

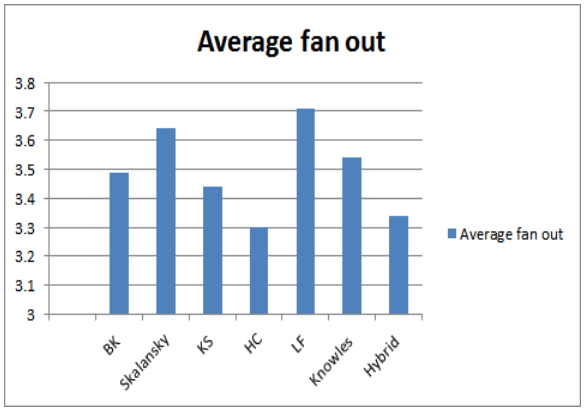

Figure 10.Average fan out of LUT's of Multiplier with various adder variants

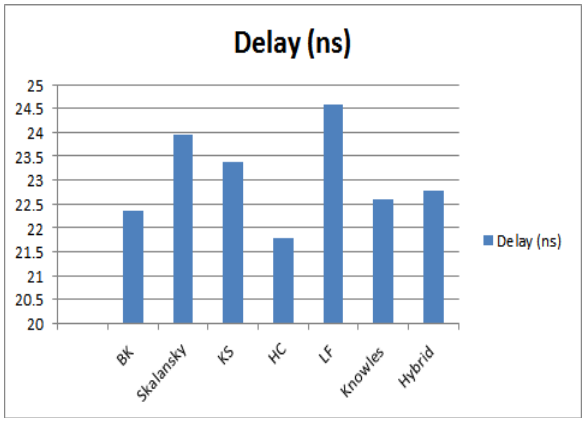

Figure 11.Delay of Multiplier with various adder variants

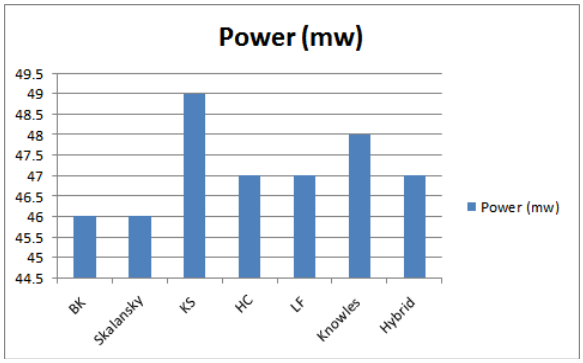

Figure 12. Dynamic Power consumption of Multiplier with various adder variants

CONCLUSIONS

This paper presents the implementation of 8-bit Multiplier using Shift and Add technique. The required addition operation is performed by various parallel prefix adder variants. To evaluate the performance of this Multiplier, Xilinx 14.5 version of VHDL is used with the targeted device of Spartan 3E. The experimental results of Multiplier based on hybrid PPA is compared with various PPA variants.

The speed of the Multiplier is estimated with the delay parameter. The delay of Multiplier with hybrid PPA is 22.767ns. The speed performance of Multiplier is enhanced to 7.5% with hybrid PPA when compared to LF model. The area occupied by the design is evaluated from the utilization of Slices and LUTs. 35% of area is saved by the Multiplier when compared to KS and Knowles models. The dynamic power consumption of Multiplier has reduced to 4% when compared to KS model. The average fan out of LUTs is resulted as lowest when compared to all models. The number of logic levels of Multiplier implementation of FPGA is least in case of hybrid PPA model.

The lower value of fan out and less number of logic levels of implementation of Multiplier with hybrid PPA model indicate that the speed performance of Multiplier with higher order bits of implementation and its utilization in higher level circuits enhances the speed performance of that system, this can be further tested and utilized by the digital signal processing applications such as correlations, convolutions, FIR filtering.

REFERENCES

[1] John F.Wakerly, "Digital Design", 2nd edition, Pearson Education, 2004

[2] Neil H.E.Weste and Kamran Eshraghian, "Principles of CMOS VLSI Design", 2nd edition, Pearson Education, 2004.

[3] Pucknell and Eshraghian, "Basic VLSI Design", 3rd edition, PHI, 1994.

[4] Z.Huang, "High-Level optimization Techniques for Low power Multiplier Design", PhD dissertation, University of California, LOS Angeles, June 2003.

[5] L.Raja, B.M.Prabhu, K.Thanushkodi, "Design of low power Dual Threshold Voltage Adder Module", Elsevier, International Conference on Communication Technology and System Design, 2011.

[6] MohmmadJaveed, GellaRavikanth, "Design and Implementation of 64 BIT Multiplier by using carry save Adder", Proc. 10th IRF International Conference, pp. 45-47, Oct.2014.

[7] Sukdev Singh, Puneeth Jain, Pankaj Sharma, RamandeepChalal, "Design and synthesis of various Multipliers using VHDL: Performance Analysis Approach", International Journal of Electronics and Computer Science Engineering, Vol.3, N0.3, pp.339-347.

[8] Sarita Singh and Sachin Mittal, "VHDL design and Implementation for optimum Delay & Area for Multiplier and Accumulator unit by 32 bit- sequential Multiplier", International Journal of Engineering Trends and Technology(IJETT), Vol.3, No.5, pp.683-686, 2012.

[9] Ruchi Sharma, "Analysis of Different multiplier with Digital Filters using VHDL Language", International Journal of Engineering and Advanced Technology(IJEAT),Vol.2, No.1, pp.45-48, Oct.2012.

[10] K.S.Ganesh Kumar, J.DevaPrasannam &M.Anitha Christy, "Analysis of Low power Area and High Speed Multipliers for DSP applications", International Journal of Emerging Technology & Advanced Engineering (IJETAE), Vol.4, No.3, pp.278-282, March 2014.

[11] Giovanni D Aliesio, "8-by-8 Bit Shift/Add Multiplier", Digital Design and & synthesis COEN 6501, Department of Electrical & Computer Science Engineering, Concordia University, Dec 2003.

[12] SaeeidTahmasbiOskuii, "Design of Low power reduction Trees in parallel Multipliers", PhD dissertation, Norwegian University of Science and Technology, Trondheim, Norway.

[13] K.H.Tsoi, P.H.W.Leong, "Mullet- a parallel Multiplier generation", International Conference on FPGAs, pp. 691-694, 2005.

[14] S.V.Padmajarani and M.Muralidhar, "A Hybrid Parallel Prefix Adder for high speed computing", Proc. 7th National Conference on Advances in Electronics and Communications(ADELCO), 2011.

[15] S.V.Padmajarani and M.Muralidhar, "Comparison of Parallel Prefix Adders Performance in an FPGA", International Journal of Engineering Research and Development (IJERD), Vol.3, No.6, pp. 62-67, September -2012.

[16] S.V.Padmajarani and M.Muralidhar, "A New Approach to implement Parallel Prefix Adders in an FPGA", International Journal of Engineering Research and Applications(IJERA) ,Vol.2, No.4,pp. 1524-1528, July-August, 2012.

[17] S.V.Padmajarani and M.Muralidhar, " Design and Implementation of a Hybrid High Speed Area Efficient Parallel Prefix Adder in an FPGA", International Journal of Computer Applications(IJCA),Vol.58, No.1, November, 2012

★★★