Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7642738

DESIGN AND IMPLEMENTATION OF NOVEL HYBRID MULTIPLIER FOR HIGH SPEED AND LOW DELAY APPLICATIONS

Tadikonda Prasad Babu and Dr. Rahul Mishra
Department of Electronics & Communication Engineering, Dr. A.P.J. Abdul Kalam University, Indore (M.P.) - 452010, India
Corresponding Author Email : tadikondaprasadbabu@gmail.com

ABSTRACT: The main intent of this paper is to design and implementation of novel hybrid Multiplier design at high speed and low delay applications. Basically, the implementation of hybrid Multiplier is based in VLSI chips because they are used as critical element. In the same way partial product generator is used in the multiplier to generate the propagator and generator signals. While designing dual partial product unit, optimized multiplier are used most widely. The novel hybrid Multiplier uses adequate hardware implementation. The transistor logic system only depends on the novel hybrid Multiplier to reduce the delay. Hence compared to adder system, the novel hybrid Multiplier systems gives effective results in terms of speed, area and delay.

Key Words: Novel Hybrid Multiplier, VLSI, CMOS, Hybrid adder, Partial products, Multiplier, Multiplicand, Carry generation unit.

I.INTRODUCTION

A circuit specially designed for the purpose of doing multiplication of two binary numbers is called as multiplier. In digital processors and several other digital devices employs multipliers for processing of signals. The addition, subtraction and multiplication of two binary numbers are very essential and frequent operations that occur in arithmetic logic units, processors and DSP applications. It is estimated that around 70% of the operations performed in various processors and DSP devices are addition and multiplication. As these operations occur most frequently the speed of the processor depends upon the speed of the multiplier. Therefore the speed of the multiplier and adder circuits governs the performance of the processors.

Advancements in computer technologies and signal processing developments insist much faster arithmetic units. Several multiplier designs have been proposed so far. The performance of multipliers is evaluated using parameters like operating speed, cost, and area and delay utilization.

In present age the development of coordinated circuit gadgets has expanded a great deal. The VLSI applications are given as computerized signal preparing and chip. These applications are most generally used to perform arithmetic and logical operation tasks. Alongside that duplication, deduction tasks are likewise performed. By utilizing a few modules this load of tasks are performed on the 1 bit full snake circuit.

Estimated adders are executed in a manner to adjust the compromise between precision versus execution/power. The spaces of picture and video preparing show up as great contextual analyses for the utilization of rough outcomes. The circuit configuration is tended at various levels. The plan boundaries of the circuit produce great compromise as far as speed and region [1]. Augmentation has been given cautious thought by using investigators, in light of the truth extension which is basically bitwise movement among two subject parts and the extra complicated errands, inversion and may be done with a couple of increments.

Montgomery's multiplier is grouped into three sorts, they are bit-sequential, piece equal, and digit sequential models. Spot equal shape is fast but it's far steeply-estimated in various expressions of area. Spot sequential construction is district effective, yet it's unreasonably slow for bounty bundles. The digit sequential design is adaptable which might change the space and speed, therefore, it accomplishes a moderate speed, sensible cost of execution and henceforth it is generally proper for useful use. Montgomery introduced a method for figuring measured augmentation beneficially. Montgomery acquainted with various depictions of numbers from the Zn to a substitute region, called Montgomery Residual depiction or Montgomery Domain [2-3].

Here with the end goal of safety, the PCs and correspondence framework carried with an interest from private area [4]. The Montgomery increase is the computation that grants successfully for enlisting. The cost of the specific duplication is identical to three entire numbers which expansions however the cost of the adjustment of the Montgomery region. However, if the huge scope task is an exponentiation at point the change of cost is unimportant appeared differently in relation to the amount of expansions executed in the Montgomery region. During the time spent Montgomery increase, pre-preparing unit and post-handling units are utilized [5]. The pre-preparing unit produces N-Residue operands and similarly post handling unit will wipe out the consistent factor 2n. Subsequently to shape N-Residue operands in the framework, measured exponentiation is utilized.

Here to displace division exercises, shifting undertakings are utilized. After the moving system the most un-basic pieces will stay zero. Presently to dispense with these devices in the measured increase, add items are utilized. After the most common way of disposing of the pieces, the leftover pieces are increased in the multiplicand. Thus from this it can see that the course of multiplicand is finished. Presently the yield is acquired after the deduction of pieces. Here on the off chance that the pieces are expanded, Montgomery bits additionally increments. Finally the multiplicand pieces are controlled without the utilization of deduction estimation.

II. REVIEW OF MULTIPLIERS

The binary multiplication procedure is similar to the method of basic decimal number multiplication. In decimal multiplication process multiplicand is multiplied with each number in the multiplier. The product obtained in each step of multiplication is called as partial products. The partial products obtained with the multiplication of second digit are shifted one position left to the partial products of preceding number in ones place. In the same way all

numbers are multiplied with the multiplicand and results are placed in their corresponding positions.

The serial multiplier used technique of shift and adds method. In this multiplier the multiplier is shifts to right side sequentially. If the first bit in multiplier is '1' then the multiplicand is added to the accumulator value. If the bit is '0' then the result present in accumulator is shifted to right. The registers present in the multiplier figure holds the values of operands. A holds multiplicand, B holds multiplier and the registers R and B holds result of the operation.

In parallel multiplier, multiplication of each bit of the multiplier is done first for producing partial products. Then to generate resultant product P these partial products are summed up together. The parallel multiplication is considered as two parts, namely partial product generation and summation.
The pipe line multiplier can be implemented from the parallel multiplier which is operating in three step process. The performance of the parallel multiplier is improved further by adding an extra step to decrease the number partial product reduction following partial products production step. The reduced partial products are summed up using a ripple carry adder. In this Pipeline technique the instructions executed continuously by overlapping the operations.

To perform arithmetic or logic operations, the pipeline includes a sequence of stages for performing particular task is known as logic circuits. The circuit in first stage is determined to produce partial products. For performing pipeline operations synchronously the multiple steps of the pipeline are partitioned by the clocked latches, which are known as registers. The midway results during multiple steps of pipeline are stored in these registers. A common clock signal that is applied to the registers in all stages to store the data in registers is called a synchronous clock.

III. HYBRID ADDER DESIGN

The below figure (1) shows the block diagram of Novel hybrid adder. In this mainly carry look ahead adder generator, propagator & generator and carry generation cells are used.

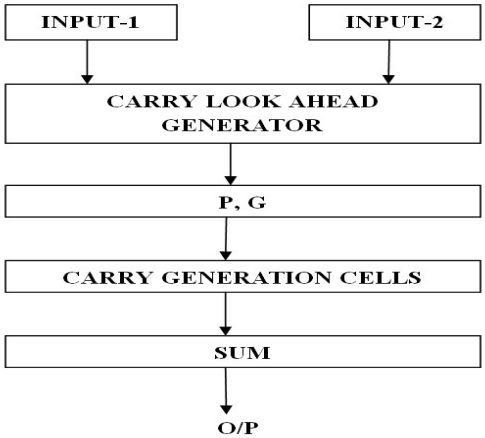

Fig. 1: BLOCK DIAGRAM OF NOVEL HYBRID ADDER

The n-bits of an operand are divided into groups with a CLA in each group and each group is connected by using a RCA. The n-bits are divided in equal size for easy design and modularity. In group CLA the group carry generation and propagation are represented by G*, P*.
The G*= 1 if carry out from the group is produced.
The p* = 1 If all propagation P_0, P_1, P_2, P_3 bits inside the group are 1.
Group-generated carry G*= 1 if a carry-out (of group) is produced with in the group.
Group-propagated carry P*=1 if a carry-in (to group) is propagated internally to produce a carry-out (of group).
For each group the carry generate and carry propagate equations are given below
$G^* = G_3+G_2P_3+G_1P_2P_3+G_0P_1P_2P_3$
$P^* = P_0P_1P_2P_3$
The CLA works on the principle of determining, whether carry will generate or not based on input signals even before the actual addition performed. Here, it is illustrated an example addition two binary operands, which shows the difficulty arising due to the delay in carry propagation.

As a solution to this problem CLA is implemented. This leads to less delay in propagation of the carry. To make the concept of CLA it is very clear to control the logical expressions corresponding to the full adder. The equations of full adder regarding carry propagate and generate are given as
Carry propagate $P_i = A_i \oplus B_i$
Carry generate $G_i = A_i \times B_i$
The above equations clearly shows that both signals G and P can be produced by using input bits with in time of one gate delay. The rewritten expressions for sum S_i and carry C_{i+1} in the form of P and G are given by:
$S_i = P_i \oplus C_{i-1}$

$$C_{i+1} = G_i + P_iC_i$$

There are only two chances in getting carry while doing addition. One chance is when two input bits A_i and B_i are 1. Second chance is when at least one of input bits is one and carry input C_i is 1.
The common equation for carry in CLA is

$$C_{i+1} = Gi + PiG_{i-1} + PiP_{i-1}G_{i-2} + \ldots\ldots P_iP_{i-1}\ldots P_2P_1G_0 + PiP_{i-1}\ldots P_1P_0C_0$$

The above expression can be realized using two stage circuit. But, in CMOS logic the delay of the circuit varies nonlinearly with the fan-in.

IV. NOVEL HYBRID MULTIPLIER DESIGN

The below figure (2) shows the architecture of novel hybrid multiplier design. The entire system is divided into four modules. The four modules are multiplier register, multiplicand register, partial product and hybrid adder using GDI. This operation is mainly used in the coprocessor in serial format. This system provides the functionality for coprocessor. Basically, there are two registers available in this block diagram there are multiplicand Md and Multiplier

Mr registers. First the alignment of partial products will be done. After that partial products takes this registers and generates the propagate and generate signals. After this dual propagator generator alignment will be processed.

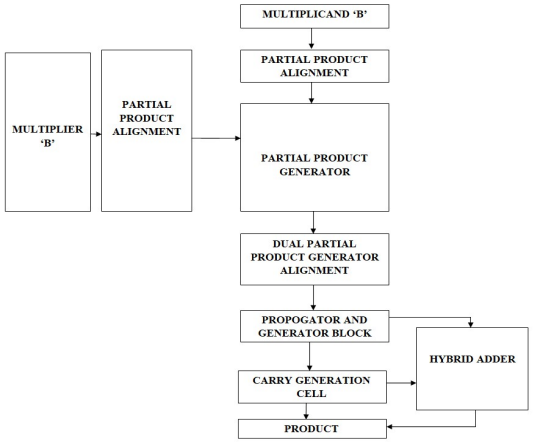

Fig. 2: BLOCK DIAGRAM OF NOVEL HYBRID MULTIPLIER DESIGN

Here firstly, the operands are loaded in the multiplier by using registers. The arithmetic operations like addition and multiplication operations are performed by using arithmetic circuit. The obtained result of this will be saved in the register. Here irreducible polynomial function is not used in the system. The main intent of register multiplier is to store the bit representation and give polynomial output a(t). Here parallel load operation is performed in the most significant bit position. In the same way left shift operations are performed in MSB bit. The multiplicand bit is used b(t) value to store the value in register. The parallel load operation is also applied in the multiplicand. The obtained value is stored in the register. The right shift operation is performed in the multiplicand register block. Crypto core processor is used to transfer the data in multiplicand register.

The multiplier register consists of root and load mr and this are taken as input to this block. The multiplier register is generally attached to the finite field arithmetic circuit. In the same way, multiplicand register consists of shift, data_in and load_md bits which are taken as input to the multiplicand register. It will shift the data and as well as load the data in effective way. Result register consists of output and saves the entire arithmetic result. Compared to existed system, the proposed system gives effective results.

The result multiplier and multiplicand is saved in the result register block. The both a(t) and b(t) values are assigned in the multiplier and multiplicand register blocks. The obtained values in the multiplier register block will shift the bits to finite field arithmetic block. This block will perform the arithmetic operations like addition and multiplication. After performing particular operation, the bits are shifted to the result register. This result register will save the output of finite field arithmetic circuit. At last the multiplicand register will perform the parallel operation in effective way.

V. RESULTS

The below table (1) shows the comparison table of parameters for hybrid multiplier and novel hybrid multiplier. In this total delay is divided into two types logic delay and route delay. Compared with hybrid multiplier, novel hybrid multiplier gives effective outcome in effective way.

Table. 1: COMPARISON OF PARAMETERS

S.No	Parameter	Hybrid Multiplier	Novel Hybrid Multiplier
1	Total delay	High	Low
2	Logic delay	High	Low
3	Route delay	High	Low
4	Memory used	High	Low

The below figure (3) shows the comparison of delay's. Total delay is classified into two types logic delay and route delay. Compared with hybrid multiplier, novel hybrid multiplier will give results in effective way.

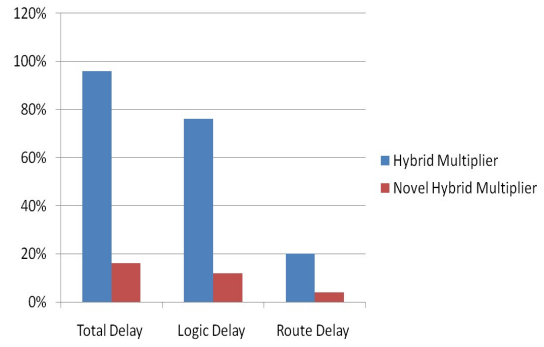

Fig. 3: COMPARISON OF DELAY'S

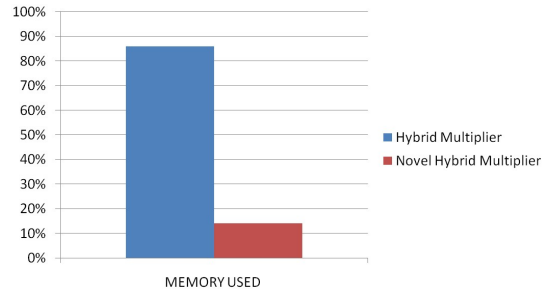

Fig. 4: COMPARISON OF MEMORY USED

The above figure (4) shows the comparison of memory used. Compared with hybrid multiplier, novel hybrid multiplier will reduce the memory in effective way.

VI. CONCLUSION

Hence in this paper design and implementation of novel hybrid Multiplier design at high speed and low delay applications is presented. The novel hybrid multiplier performs the multiplication is very fast. The proposed system uses registers to save the values in effective way. The proposed system will reduce the switching activities that are produce in the system. To reduce the delay partial product unit is introduced. Hence the area of complexity is reduced in novel hybrid multiplier. This system is mainly used in the applications of low delay and high speed applications.

VII. REFERENCES

[1] "Design and implementation of high speed modified booth multiplier using hybrid adder", Divya Govekar , Ameeta Amonkar, 2017 International Conference on Computing Methodologies and Communication (ICCMC).

[2] "Power-delay-area efficient design of vedic multiplier using adaptable manchester carry chain adder", Raghava Katreepalli , Themistoklis Haniotakis, 2017 International Conference on Communication and Signal Processing (ICCSP).

[3] "Low power array multiplier using modified full adder", S. Srikanth , I. Thahira Banu , G. Vishnu Priya , G. Usha, 2016 IEEE International Conference on Engineering and Technology (ICETECH).

[4] "Design of high speed multiplier using modified booth algorithm with hybrid carry look-ahead adder"
R Balakumaran , E Prabhu, 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT).

[5] "Design of area and delay efficient Vedic multiplier using Carry Select Adder", G. R. Gokhale , S. R. Gokhale, 2015 International Conference on Information Processing (ICIP).

[6] "Design of ultra low power multipliers using hybrid adders", Thottempudi Pardhu , N.Alekhya Reddy, 2015 International Conference on Communications and Signal Processing (ICCSP).

[7] "Comparative study of performance vedic multiplier on the basis of adders used", Josmin Thomas , R. Pushpangadan , S Jinesh, 2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE).

[8] "Design of area efficient and low power multipliers using multiplexer based full adder"
S. Murugeswari , S. Kaja Mohideen, Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014.

[9] "A vertical-MOSFET-based digital core circuit for high-speed low-power vector matching", Yitao Ma , Tetsuo Endoh , Tadashi Shibata, 2011 International SoC Design Conference.

[10] H. Hinkelmann, P. Zipf, J. Li, G. Liu, and M. Glesner, "On the design of reconfigurable multipliers for integer and Galois field multiplication," Microprocessors Microsyst., vol. 33, no. 1, pp. 2–12, Feb. 2009.

[11] P. K. Meher and C.-Y. Lee, "Scalable serial-parallel multiplier over GF(2m) by hierarchical pre-reduction and input decomposition," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2009, pp. 2910–2913

[12] J. Fan and I. Verbauwhede, "A digit-serial architecture for inversion and multiplication in GF(2m)," in Proc. IEEE Workshop Signal Process. Syst. (SiPS), Oct. 2008, pp. 7–12.

[13] P. K. Meher, "High-throughput hardware-efficient digit-serial architecture for field multiplication over GF(2m)," in Proc. 6th Int. Conf. Inf., Commun. Signal Process. (ICICS), Dec. 2007, pp. 1–5.

[14] W. Tang, H. Wu, and M. Ahmadi, "VLSI implementation of bit-parallel word-serial multiplier in GF(2233)," in Proc. 3rd Int. IEEE-NEWCAS Conf., Jun. 2005, pp. 399–402.

[15] H. Wu, "Bit-parallel finite field multiplier and squarer using polynomial basis," IEEE Trans. Comput., vol. 51, no. 7, pp. 750–758, Jul. 2002.