p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
May 2017

Design an Efficient Dual Logic Level Multiplier

K.V.V.N.Arunasree
M.Tech, E.C.E Department
A.K.R.G College of Engineering & Technology
Nallajerla, A.P . India

K.Chaitanya Lakshmi M.Tech
Assistant Professor , E.C.E Department
A.K.R.G College of Engineering & Technology
Nallajerla, A.P . India

ABSTRACT

This paper represents the design of a Dual Logic Level Multiplier for 32*32 bit number multiplication. Modern computer system has a dedicated and very high speed unique multiplier. Therefore, this paper presents the design of a dual logic level multiplier. This proposed system hasseveral interconnected blocks. By extending bit of the operands it generates an additional product the dual logic level multiplier.Multiplication is performed by the dual logic level in efficient manner with less area and also it reduces delay i.e., speed is increased.

I. INTRODUCTION

Multiplication is the most important fundamental operation in signal processing algorithms. Most of the Multipliers occupy large area, long latency and consume more power. Therefore low-power multiplier design plays an important part in low power VLSI system.The importance for low-power VLSI system arises from two main forces. First, due to the steady growth of operating frequency and processing capacity of each chip, large currents should to be delivered and the heat occurred due to large power consumption must be reduced by efficient cooling techniques. The battery life of portable electronic devices is limited to some extent. Low power design leads to lengthy operation time in the portable devices. Extensive work is going on for low-power multipliers at technology, performance of the multiplier unit, because the multiplier is generally the slowest element in the system. Multipliers generally occupy the most of the area in the circuit. So, optimizing the speed and area of the multiplier is most important issue. But, area and speed are conflicting constraints so improving speed results mostly in larger areas. As a result, the whole spectrum of multipliers with different area speed constraints has been designed with parallel design. Fully Parallel Multipliers at one end of the spectrum and fully serial multipliers are at another end of the system are implemented.

Multiplication is done in three steps: generation of partial products (PPG), reduction of partial products (PPR), and finally carry-propagate addition (CPA).In general there are sequential and combinational multiplier implementation designs. We only consider combinational case here because the scale of integration now is large enough to accept parallel multiplier implementations in digital VLSI systems. Different multiplication algorithms differ in the methods of PPG, PPR, and CPA. For PPG, radix-2 is one of the easiest ways. To reduce the number of PPs and consequently reduce the area/delay of PP reduction, one operand is generally recoded into high-radix digit set. The most popular among these is the radix-4 digit set {-2,-1, 0, 1, 2}. For reduction of PPR, two alternatives exist: reduction by a row, which isperformed by an array of adders, and second is the reduction by columns, which is performed by an array of counters. The final CPA requires a fast adder scheme because of its critical path. In many cases, final CPA is postponed if it is advantageous to keep redundant results from PPG for future arithmetic operations.

For the standard description of digital systems and portable devices, VHDL is used as input and output to various simulation, synthesis, and layout tools for efficient design systems. This language will provide the ability to describe systems, networks, and components at a high behavioral level as well as very low gate level also. It represents a top-down methodology and environment. Simulations can be carried out at any level from a generally functional analysis to a very detailed gate-level wave form analysis.

II.LITERATURE SURVEY

Floating point arithmetic

Many applications in this generation require numbers that are not integers. There are many ways that non-integers that is decimal points can be represented. Adding two such numbers can be done with integer add operation, whereas multiplication requires some extra shifting bits in addition. There are various ways for representing the number systems. But, only one non-integer representation has gained fame among all, and that is floating point.

Floating Point Importance:

In this method,floating point is divided into two parts, an exponent part and a significant part.The advantage of standardizing a particular representation is more efficient. The semantics of floating-point instructions are not as clear-cut as the semantics of the rest of the instruction set, and in the past the behavior of floating-point operations varied considerably from one family to the next family. The variations have involved in things like the number of bits allocated to the exponent and significant, In the range of exponents, how rounding was carried out, and the actions taken on exceptional conditions like underflow and over- flow are explained. Now a day's computer industry is rapidly converging on the format specified by IEEE standards. Theadvantage of using a standard variant of floating point is similar to those using floating point over other non-integer representations. IEEE arithmetic differs a lot from previous arithmetic ways.

Floating Point Rounding

When we are rounding a result to the nearest floating-point number, it picks the one that is even. It rounds to nearest by default, but it is also having three other rounding modes.

It has sophisticated facilities for handling exceptions. Whenwe are operating on two floating-point numbers, the result is a number that cannot be exactly represented as another floating- point number. This should be rounded to two digits. In the IEEE standard, such cases are rounded to the number whose low-order digit is even. The standard generally had four rounding modes. The default is round to nearest, which rounds to an even number as just explained. The other modes are rounding towards 0, rounding towards $+\infty$, and rounding towards $-\infty$.An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a specific use, rather than for general-purpose use. For example, a chip is designed only to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products.

As the design sizes have shrunk and design tools are improved from the years, the maximum complexity possible in an ASIC had grown from 5,000 gates to over 100 million gates.Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and many other large building blocks.Such an ASIC is often termed assystem-on-a-chip. Designers of digital ASICs

p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
May 2017

use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs.Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts, programmable logic blocks and programmable interconnects allow the same FPGA to be used in different applications. For smaller designs and lower production volumes, FPGAs may be more cost effective than an ASIC design.An application-specific integrated circuit is an integrated circuit (IC) customizedfor a particular use, rather than intended for general-purpose use. A Structured ASIC falls between an FPGA and a Standard Cell-based ASIC Structured ASIC's are used mainly for mid-volume level design. The design task for structured ASIC's is to map the circuit into a fixed arrangement of known cells.

III.EXISTINGSYSTEM

The multiplier was based on the variable-latency technique and it is used to regulate the AHL circuit to attain consistent operation for reducingthe error and re-execution of clock cycle. The adaptive hold logic circuit was utilized to determine whether the input needs one or two cycles to execute the required operation and can regulate the judging criteria to make sure that there is less error detection and re-execution of clock cycle. In the given example of the 4 x4 column bypass multiplier, the architecture consists of seven column parts and seven row parts. Further 8 x 8 bit architecture the numbers of parts are increased, this increases the area and number of logical operation. But when compared to the normal AM, speed is increased in this technique.To decrease the number of logical operations and number of parts, layers and improve the speed as better to this technique, we should propose the Dual Logic Level Multiplier.

IV.PROPOSED SYSTEM

DUAL LOGICLEVELMULTIPLIER

The high performance Dual Logic Level Multiplier gives less delay when compared with other multipliers. A dual logic level shares its logical operation that depends on the preference of the logical operation that is executed. It consists of the three layers and the parts depend on the bit size. In 2 bit size there are 3 Parts, 3 bit size there are 5 parts and 4 bit size there are 7 parts.As the bit size increases the parts are double in the architecture. In the architecture the main thing is depending on the third layer. In the third layer there are two levels of operations aredone. This depends on the preference, one level of operation is performed and the second one vice versa.

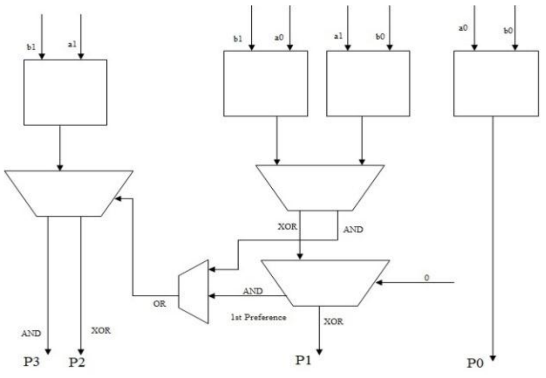

Fig. 4.1: 2 x 2 Architecture of Dual Logic Level Multiplier

The dual logic level multiplier architecture depends on the third layer operation .In the third layer two levels of operations are performed. Depending on the preference, one level of operation is performed next the other.But in the third layer AND & XOR operations are performed.In the AND gate one gate is used to perform operation but in the XOR five gates are used for operation. Due to that first preference is given to AND gate and the second preference is given to XOR gate. So for that purpose, depending on the operation, which is performed quickly, is preferred first after that second operation is performed. In that architecture in third layer is first preference AND to do operation next XOR operation is performed.

In the internal operation of 2 x 2 dual logic level multiplier, it consists of three layers and three parts. The inputs are given to the first layer in the first part a_0, b_0 inputs given to the first layer AND gate and the result go down. In the second part a_0, b_1 and a_1, b_0 inputs are given to the first layer two AND gates the outputs of the AND gates is given to the inputs of the second layer multiplier. In this two operations are performed XOR, AND. The second layer is having two outputs, one output is XOR and second output is AND, the XOR output is given to the input of third layer multiplier, AND output is given to one of the input of the third layer OR gate.In the third layer XOR, AND operations are performed, the XOR operation is performed which requires five gates for the operation but in the AND operation only one gate is used. So in the third layer first preference is given to AND gates next the XOR gate to perform the operations. In the third layer multiplier XOR output goes down AND output is given to the one input of OR gate and second input of the OR gate is coming from second layer multiplier AND output. In the third part first layer inputs a1, b1 is given to the AND gate and the output is given to the second layer of multiplier and the second input is coming from third layer OR gate output. The two outputs of the multipliergoes down. This is the total internal operation of 2 x2 Dual Logic Level Multiplier. The number of possible outputs in this multiplier is 2 x m where m may be multiplier or multiplicand.2 x2 multiplier having 2 x 2 = 4 outputs.

Algorithm for Dual Logic Level Multiplier

Dual LogicLevel Multiplier works by reading of two input values known as multiplicand and multiplier. Input registers read the two input values multiplicand and multiplier. Both the inputs are passed to dual level logic multiplier, after the completion of multiplication operation in multiplier the product will be selected by the logic family according to the preferences.

Dual LogicLevel Multiplier contains adder cells which performs XOR and AND operations. AND operation is given first preference because the result of AND operation is given as carry to the next adder cells. Later the XOR operation is done and it is taken as sum bit. All adder cells will be performed in parallel and there is no delay in the carry like Adaptive Hold Logic Multiplier. So the disadvantage of Adaptive Hold Logic is overcome by Dual LogicLevel Multiplier in this way.

Flow chart of Dual Logic Level Multiplier

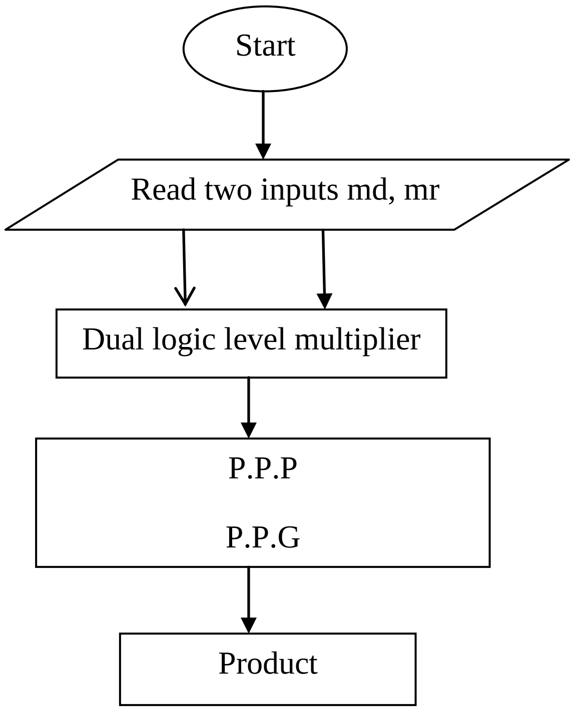

Fig 4.2: Flow chart of dual logic level multiplier

V.RESULTS

PROPOSED SYSTEMRTL

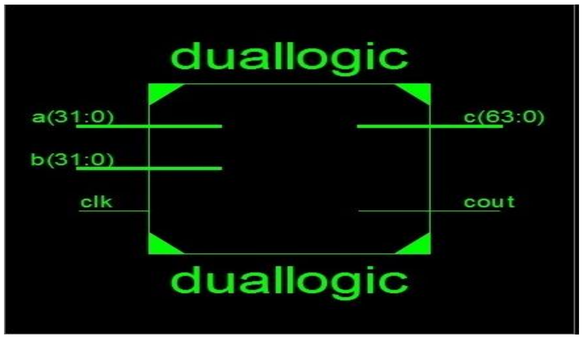

Fig5.1 :RTL Schematic of 32*32 Dual Logic Level Multiplier

TTL SCHEMATIC

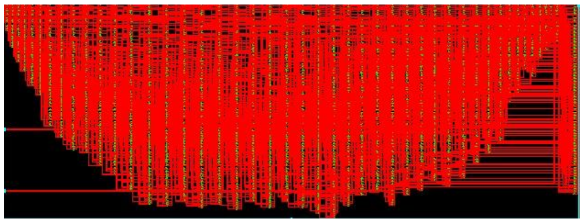

Fig 5.2: TTL Schematic of 32*32 Dual Logic Level Multiplier

OUTPUT WAVEFORM

The output waveform is shown in below figure for proposed system

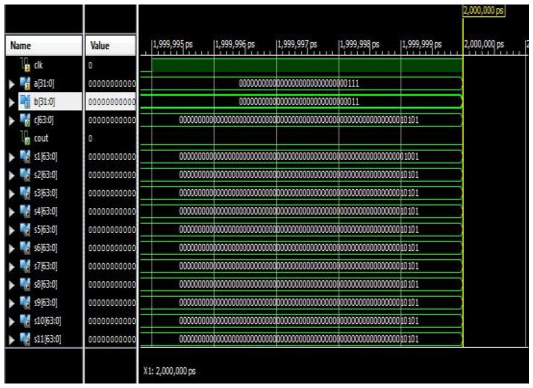

Fig 5.3:Simulated Waveform of 32*32 Dual Logic Level Multiplier

COMPARISON TABLE:

S.No	Type of Parameter	Existed System	Proposed System
1	Area	1.4	1.08
2	Delay	3.6	1.1

VI .CONCLUSION

The proposed system consists ofinterconnected blocks. The each block consists of gates and the number of rows in the architecture is lesser than existed multiplier. By generating a product with dual logic level multiplier is obtained. Multiplication operation that is performed by the dual logic level unit has better performance than existed multiplier. The required hardware and the chip memory is reduced and it reduces delay i.e., speed is increased.

VII.REFERENCES

[1] Ing-Chao Lin, Member, IEEE, Yu-Hung Cho, And Yi-Ming Yang"Aging-Aware Reliable Multiplier Design With Adaptive Hold Logic" IEEE Transactions On Very LargeScaleIntegration(VLSI)Systems

[2] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, “NBTI-aware flip-flop characterization and design,” in *Proc. 44th ACM GLSVLSI*, 2008, pp. 29–34

[3] S. V. Kumar, C. H. Kim, and S. S. Sapatnekar, “NBTI-aware synthesis of digital

circuits,” in *Proc. ACM/IEEE DAC*, Jun. 2007, pp. 370–375.

[4] A. Calimera, E. Macii, and M. Poncino, “Design techniqures for NBTItolerantpower-gating architecture,” *IEEE Trans. Circuits Syst., Exp.Briefs*, vol. 59, no. 4, pp. 249–253, Apr. 2012.

[5] K.-C. Wu and D. Marculescu, “Joint logic restructuring and pin reorderingagainst NBTI-induced performance degradation,” in *Proc. DATE*,2009, pp. 75–80.

[6] Y. Lee and T. Kim, “A fine-grained technique of NBTI-aware voltagescaling and body biasing for standard cell based designs,” in *Proc. ASPDAC*,2011, pp. 603–608.

[7] M. Basoglu, M. Orshansky, and M. Erez, “NBTI-aware DVFS: A newapproach to saving energy and increasing processor lifetime,” in *Proc.ACM/IEEE ISLPED*, Aug. 2010, pp. 253–258.

[8] K.-C. Wu and D. Marculescu, “Aging-aware timing analysis and optimizationconsidering path sensitization,” in *Proc. DATE*, 2011, pp

[9] K. Du, P. Varman, and K. Mohanram, “High performance reliablevariable latency carry select addition,” in *Proc. DATE*, 2012,pp. 1257–1262.

[10] A. K. Verma, P. Brisk, and P. Ienne, “Variable latency speculativeaddition: A new paradigm for arithmetic circuit design,” in *Proc. DATE*,2008, pp. 1250–1255.