International Journal on Emerging Technologies 10(1): 93-96(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Design and UVM Verification of High Speed ALU

J. Selva Kumar[1], Rohit Konathala[1], R. Manikandan[2], C. Ramesh[3] and Robbi Rahim[4]
[1]Department of Electronics and Communications, SRM Institute of Science and Technology, Kattankulatur - 603203
[2]School of Computing, SASTRA Deemed University, Thanjavur – 613401.
[3]Associate Professor, Department of Computer Science & Engineering,
Bannari Amman Institute of Technology, Sathyamangalam, (Tamilnadu), India
[4]Department of Management, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia

(Corresponding author: R. Manikandan)
(Received 27 February 2019, Revised 25 April 2019 Accepted 08 May 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: The high speed ALU is designed with a sole intent to reduce the delay of an ALU and in doing so dramatically boost the efficiency of the system. High speed of the ALU unit was achieved by porting high speed modules to the ALU logic, high speed modules such as Vedic Multipliers, carry look ahead adders, shifters, sequence detectors etc. In addition to the system, special emphasis has been laid on the design procedures taken to design the system. All the modules used in the system have been made such that they can easily to be reused for future projects.

Keywords: VLSI, Vedic Multiplier, Carry Look ahead Adder, ALU, UVM, VCS

I. INTRODUCTION

ALU's play an important role in contributing to the functioning of the entire system; they are most often the bottle neck as the system would have had to settle with a slower clock rate, just to synchronize with the ALU's service speed. This can improved by shifting from conventional modules to tried and tested and significantly faster modules such as Carry Look Ahead Adders, Vedic Multipliers, and Dividers etc described in the review later on. The module created here is that of 2, 8-bit inputs that have a select line of 3bit opcode to choose between the logic in the ALU. The other design consideration was to connect the inputs only to the intended logic only and hence keeping the other modules idle while getting the required output. This significantly helps in having accurate grasp of expected delay and also consumes significantly lesser power.

The design time had taken longer than expected as these are complex modules but special care has been taken into making the modules as convertible and reusable as possible. The design Principals used will be described in later sections. Vedic mathematics, on the one hand, simplifies the multiplication process and reduces the delay; while on the other hand, GDI technique helps in minimizing the transistor count (TC) and reduction in power [7]. Each of the modules was checked and verified before being instantiated. On the test bench front, while its easier to drive a targeted test bench, it make better sense to create a UVM test bench to detect the corner case scenarios where targeted, human feed inputs could not have imagined or missed due to the sheer size of inputs that would have to be fed.

II. OVERVIEW OF THE ALU STRUCTURE

A. Skeletal Design of the ALU.

The proposed ALU has 2 input ports each capable of transmitting 8 bit data. The ALU is sports a 3 bit control line (opcode) that helps switch between logical operations in the ALU. In total 16 modules/logical operations can be ported onto the ALU. This ensures future expandability of the unit to incorporate high speed modules that would be worked upon in future works.

As said, emphasis has been laid on making the unit as modular as possible in order to be able to reuse and modify the modules individually in this project or for future project instantiations.

B. Design overview of the proposed ALU

The ALU outputs are two, 8-bit outputs called A & B, there is also an 8bit Flag register provided and for comparison results output - lesser than, greater than and equal to are also provided.

The logic/modules ported are a Adder & Subtractor combined module, a 8 bit Vedic multiplier, a signed multiplier, comparison logic, sequence shifter, a Vedic divider a sequence detector.

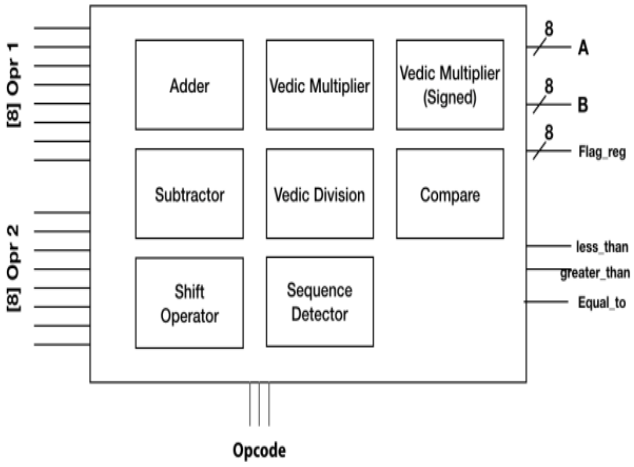

Fig. 1. Overview of the ALU design (top level).

In the sections that follow the key speed improving models are explained in detail, supported by diagrams and design style considerations taken into account

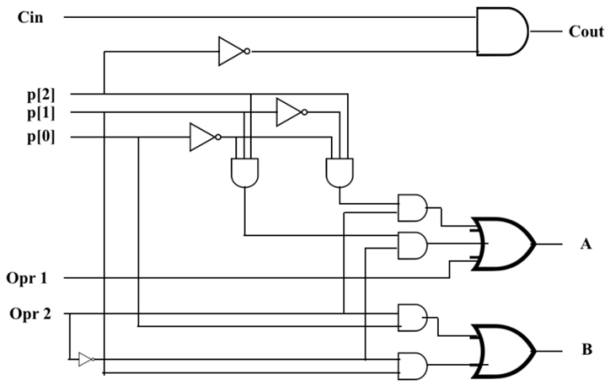

Fig. 2. Generator Block diagram of Carry look ahead adder with select line for subtraction [1].

III. ADDER AND SUBTRACTOR UNIT

The adder and the subtractor have been decided upon to be made as one module. This was done to reduce cumulative design work time and to simplify the module. There is a select line in the module which helps trigger between adder and subtractor. The architecture used here is that of a carry look ahead adder and the select port is one that was connected to one of the inputs via a xor gate. This dramatically helps reduce design complexity and carry look ahead adder are in relative comparison faster to ripple carry adders and other types. Hence the decision made proved useful and has helped in the making of the proposed module convenient. In the diagram is a very abstract level explanation of the designed adder/subtracted and that the select line used. There are 2 inputs opr1 and opr2 and a carry in. On the output front we have connect the output ports to the output ports of the ALU. The select lines P were selectively connected to the 3 bit select line of the ALU [1].

IV. VEDIC MULTIPLIER

A. Authors and Affiliations

Vedic multipliers work on simple ancient sutras (algorithms) which are essentially tricks used for fast mental mathematics and calculations. This has been used to dramatically cut down on delay time.

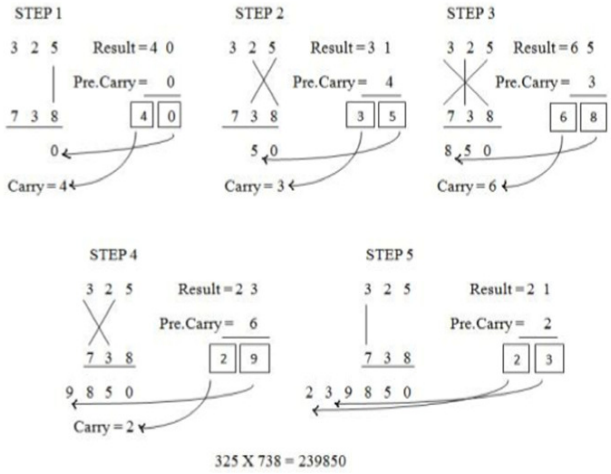

Fig. 3. Vedic multiplication methodology [2].

However Vedic calculators require significantly higher lateral space, I.e multiple operation need to take place concurrently at the same time. This has significant increase in the no of LUT that would be required to build the system. The most advantageous of all would be to use a filed programmable gate array board for emulation to get the best results.

B. Working of a Vedic Multiplier

Unlike with normal multiplication, Vedic multiplication requires the several simultaneous multiplications and additions to execute, figure above show the multiplication of 2, 3-bit numbers in the Vedic multiplication process. The numbers are multiplied in a particular sequence and then adder, the carry is taken over to the next step of the calculation

C. Construction of 4 bit Vedic Multiplier

The construction of the 8 bit Vedic multiplier begins with the development of half adder and full adder. Following which it is possible to generate a 2 bit multiplier using he half adders. After the development of the 2 bit Multiplier we can go for the making of the 4 bit Vedic Multiplier. The block diagram is shown in the figures below

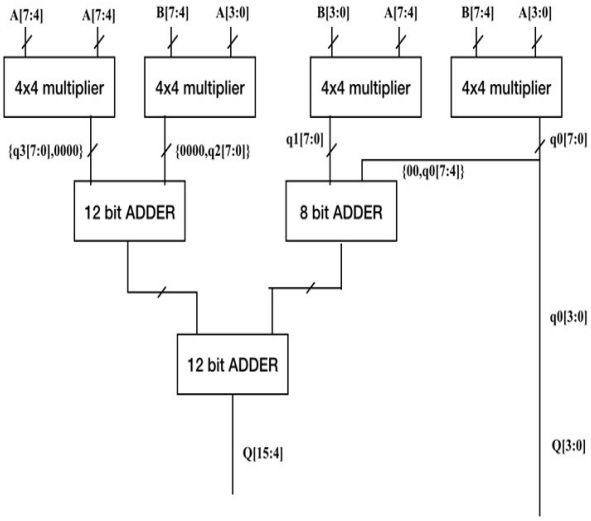

Fig. 4. Block diagram of Vedic 4 bit multiplier.

The above image is that of the place and hold sequence generated by DVE (dump) tool that supports Synopsys VCS.

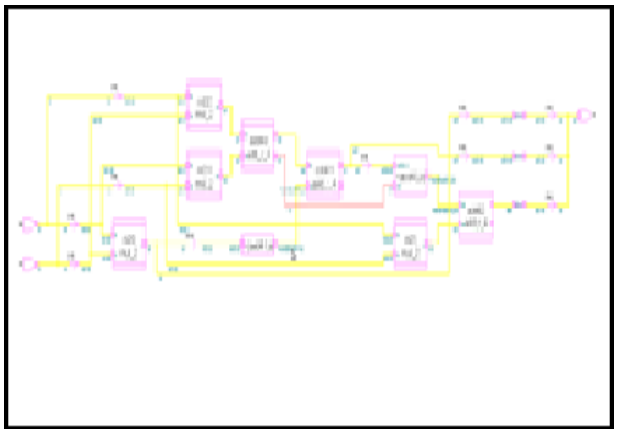

Fig. 5. Place and Hold layout schematic for 4 bit multiplier.

D. Construction of an 8 bit Vedic multiplier

Since the design was focused on building up on modules, here to make the 8 bit vdiic multiplier a similar patter as that of 4 bit Vedic multiplier is used using 4 bit multiplier and 8 bit adders, 12 bit adders.

E. Figures and Tables

The comparison of the proposed Vedic Multiplier is done again conventional booth multipliers for 3 stages 4x4, 8x8, 16x16. The figures are referenced from journal paper [2]. As can been seen here, Vedic multiplier outscores the booth multiplier by a huge margin as shown table 1. It dramatically reduces the delay and hence has been chosen for the making of the high speed ALU [2].

Table 1. Comparison of Booth and Vedic multiplier.

Sr No.	Multiplier	Booth Multiplier	Vedic Multiplier
1	4×4	17.68ns	14.62ns
2	8×8	32.28ns	23.7ns
3	16×16	75.24ns	32.7ns

V. UNIVERSAL VERIFICATION METHODOLOGY (UVM) - RANDOMIZED TEST BENCH VERIFICATION

While targeted Test Bench might help in partially verifying the basic functioning of modules, it's got several limitations and is labor intensive. In the big picture it is not feasible as one cannot possible enter all combinations of the input when the input capacity is larger. While it worked for verification of smaller modules in the project, for the test bench evaluation of the entire ALU with 2, 8 bit inputs will require a randomized stream of input data, this helps in the detection of corner case scenarios that would have otherwise been missed by the verification engineer.

The industry has set up a verified standard for testing and verification, called UVM. This programming language deals with class and inheritance properties from System Verilog. It contains predefined classes that can be extended to make the test bench environment required.

A. Brief overview of UVM

The most elemental unit of a test bench is called a transaction. It contains the input types and all control units that are expected to be driven though a DUT. A group of transactions makes a Sequence. The Sequence is driven to the driver using a class called Sequencers which acts as an intermediary. The driver looks into the enable signals of the interface BUS and the DUT clock cycle and drives the sequences at appropriate time. The driven inputs run through the DUT and the output is collected at the Monitor.

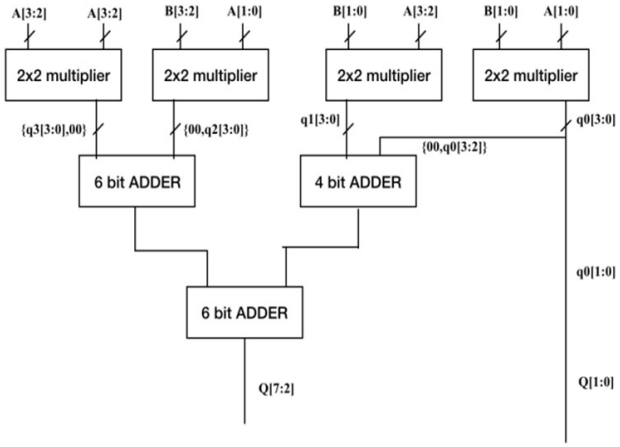

Fig. 6. Block diagram of Vedic 8 bit multiplier.

Here, for this project, we have used a dual monitor set up where in the first monitor takes in the output of the DUT and processes it, while the second monitor is driven the test bench inputs and is programmed the RTL, the output of this monitor is the expected output. The two monitors drive their values to the Score Board.

At the score board the inputs for a specific input are synchronized from the two monitors and compared. Should the values be the same, an OK flag is triggered, if not an error signal is triggered.

While preparing the transactions for this test case, we have Randomize the inputs and the select lines to detect for corner case errors as well.

Fig. 7. Place and Hold layout schematic for 8 bit multiplier.

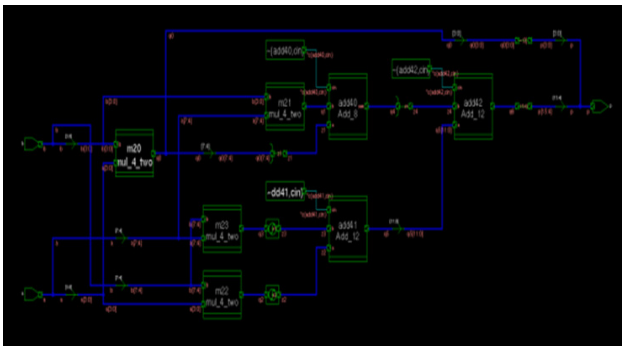

Fig. 8. Test Bench results for ALU unit.

B. Proposed Project test bench Result

The inputs are randomly triggered as are the select lines. The output wave forms for the respective select line configurations are shown and simultaneously the comparison of the input values is also acknowledged.

CONCLUSION

The proposed module is highly efficient in terms of power by selective logic enabling and in delay constraint by using significantly faster logical modules. Though the project modality and professional design standards were followed for reusability and easy instantiation and modification.

Conflict of Interest: Nil

Acknowledgment: We would like to acknowledge Mr. Selvakumar, Assistant Professor, SRMIST for guidance and mentoring and insight though-out the project. The project was conducted using the resources and training at RedPine Signals, Hyderabad under the external guidance of Mr. Mahesh Devarasetty, Sr. Design engineer, RedPine.

REFERENCES

[1]. Nautiyal, P., Madduri, P., & Negi, S. (2015). Implementation of an ALU using modified carry select adder for low power and area-efficient applications. In *2015 International Conference on Computer and Computational Sciences (ICCCS)* (pp. 22-25). IEEE.

[2]. Pichhode, K., Patil, M.D., Shah, D., & Rohit, B.C. (2015). FPGA implementation of efficient vedic multiplier. In *2015 International Conference on Information Processing (ICIP)* (pp. 565-570). IEEE.

[3]. "UVM Guide for Beginners", colourlesscube.com

[4]. Sharma, A., & Tiwari, R. (2016). Low power 8-bit ALU design using full adder and multiplexer. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2160-2164). IEEE.

[5]. Prakash, P., & Saxena, A. K. (2009). Design of Low Power High Speed ALU Using Feedback Switch Logic. In *2009 International Conference on Advances in Recent Technologies in Communication and Computing* (pp. 899-902). IEEE.

[6]. Ravi, S., & Kittur, H. M. (2017). Design of high performance double precision hybrid ALU for SoC applications. In *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)* (pp. 1-6). IEEE.

[7]. Garg, A., & Joshi, G. (2018). Gate diffusion input based 4-bit Vedic multiplier design. *IET Circuits, Devices & Systems*, **12**(6): 764-770.

How to cite this article: Kumar, J. Selva, Konathala, Rohit, Manikandan, R. Ramesh, C. and Rahim, Robbi (2019). Design and UVM Verification of High Speed ALU. *International Journal on Emerging Technologies,* **10**(1): 93-96.