

DESIGN A HIGH SPEED AND AREA EFFICIENT RECONFIGURABLE ARITHMETIC OPERATIONS USING AHL

Mr.AKKISETTI VN HANUMAN¹, Mr.R.VINAY KUMAR²

¹Associate Professor, International School of Technology and Sciences for Women,
Rajanagaram, Andhra Pradesh-533294.

²Associate Professor, International School of Technology and Sciences for Women,
Rajanagaram, Andhra Pradesh-533294.

ABSTRACT:

In this project, design a high speed and area efficient reconfigurable arithmetic operations using AHL is implemented. Basically, multipliers are key arithmetic circuits in many of these applications including digital signal processing (DSP). This ALU uses adder that limits its carry propagation. Partial products are used to generate propagate and generate signal's. AHL uses a sequential logic to increase the speed of operation. From results it can observe that the proposed system gives effective results and it is implemented using Xilinx 14.7 ISE design suite.

Key words: *DSP, AHL, ALU, multiplier, chip.*

I. INTRODUCTION

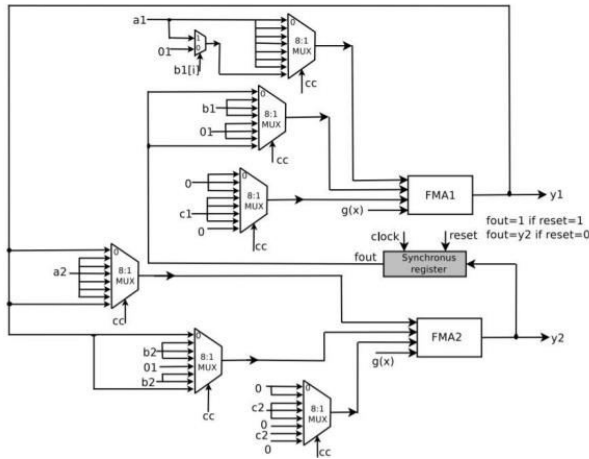
In the modern time, integrated circuit (chip) is widely applied in the electronic equipment. Almost every digital appliance, like computer, camera,

music player or mobile phone, has one or several chips on its circuit board. Very Large Scale Integration (VLSI), in general, comprises over an excess of one million transistors, an incredible figure that could not have been imagined a

decade ago. The ComputerAided Design (CAD) has further aided the growth in the complexity and performance of integrated circuits in the VLSI technology. With such a phenomenal increase in complexity, it is more crucial than ever before to manage the design process, to maintain the reliability, quality, and extensibility of a given design. The procedure incorporates "definition, execution and control of structure approaches in an adaptable and configurable manner". Speed of improvement in superior processing, media communications and customer hardware in a quickly evolving business sector, formative expenses, and cost engaged with instance of mix-ups, play a critical role in a commercial environment. Hence, it requires designs that can be processed quickly, cheaply and mistakes brought to the forefront at the earliest, perhaps, before fabrication stage. Nevertheless, a couple of inconveniences, for example, long structure and manufacture time and higher hazard to extend with

multifaceted nature of a great many parts prompts the expectation of quick calculation and formats near optimality age. The innovative work of circuit format (Physical Design) mechanization instruments could clear a path for future development of VLSI frameworks. The accepted norm about the layout of integrated circuits on chips and boards is that it is a complex process. Consequently, any problem arising as a result of optimization problems requires to be solved during the circuit layout. This refers to the fact that they are mostly Nondeterministic Polynomial (NP) - hard. The major implication of this recourse is that the optimal solutions cannot be achieved in polynomial time.

2. Existing method



a brief timeframe, both draw up and pull-down ways in the CMOS cell direct and the present streams from Dynamic power can be brought down by decreasing exchanging action and clock recurrence, which influences execution; and furthermore by the lessening capacitance and supply voltage. Dynamic power can likewise be diminished by cell determination quicker slew cells devour less unique power.

This calculation begins by putting away item esteem in LUT memory where multiplier is 2 and multiplicand is 0, 1, 2, 3, 4, 5, 6, and 7. The Address is the estimation of Multiplicand.

Stage 1: If the Multiplier is 2 and multiplicand is any an incentive from the range 0-7, at that point the item esteem put away at that specific memory area (characterized by multiplicand) is net yield. In any case, if the Multiplier is other than 2 and multiplicand is from 0-7 then re-look in memory to discover if the necessary yield is as of now

determined. In the event that indeed, take the yield from that specific location. Model: how about we guess the multiplier be 2 and multiplicand be 5, at that point $2 \times 5 = 10$ and 10 (10101)₂ is as of now put away at memory area 5. In any case, if the multiplier isn't 2 and item esteem is as yet accessible in memory (like $3 \times 4 = 12$ (1100)₂ previously put away at area 6), at that point essentially get the item esteem at yield put away at that specific area.

Stage 2: If the Multiplier is other than 2 and the worth isn't accessible in memory; search for close by esteem (one less or one more prominent of anticipated item) and flip the last piece. Model: $3 \times 5 = 15$ and this worth isn't accessible in memory, so take the close by esteem 14 (1110)₂ and flip the last piece to make 14(1110)₂ to 15(1111)₂.

Stage 3: If the close by esteem is absent, search for any least factor of the item esteem and annex the bit(s) in the last. For example, $4 \times 5 = 20$. Here 10 (1010)₂ is minimal factor of 20 (10100)₂. So take the yield 10 accessible at memory area 5

and annex zero in the last to get the twofold of 10 that is 20. So also, assume multiplier is 3 and multiplicand is 7. Here the item would be 21(10101)₂. Be that as it may, 21 are not accessible in memory. Along these lines, this can be accomplished by taking 10 (1010)₂ at the yield and afterward annexing 1 in the last. This will change the 10 (1010)₂ in to 21(10101)₂.

Stage 4: If above advances doesn't give the necessary yield, attach two bits in the last to get the necessary information. For instance, if there should be an occurrence of 7×5 coming about item esteem is 35 (100011)₂ and when we annex (11)₂ in the last to 8 (1000)₂, we can without much of a stretch get the yield 35(100011)₂.

3. PROPOSED SYSTEM

As machine learning algorithms are getting more popular, there is an increasing demand for developing hardware accelerators for them. In particular deep neural networks such as

Convolutional Neural Networks (CNNs) have multiple traits that make them very attractive for hardware acceleration, such as high structural regularity, high computational complexity, and yet wide applicability and high recognition performance. FPGAs are one of the most preferred platforms due to their high flexibility and at the same time high parallelism. Hence much effort has been made to create better CNN accelerators on FPGAs. A unique option available to hardware implementations of DNNs is the flexibility in data width of arithmetic operations. GP-GPUs, for instance, have long provided only two options— either single-precision or double-precision floating point— since integer arithmetic on modern GPGPUs has zero or negative performance advantage. Recently half-precision was introduced on some select models, but this is a one-time change and not customizable by user. By contrast an ASIC (Application-Specific Integrated Circuit) implementation can choose whatever precision sufficient for the

target CNN application. As recent work suggests that 8-bit fixed-point is often enough for inference, even for deep CNNs, there is a good opportunity to increase performance for free by using lower precision without affecting output quality. FPGAs, too, have the flexibility, and using reduced precision means potentially higher throughput on the same FPGA. In practice, however, since most arithmetic operations are implemented using DSP blocks, and DSP blocks, too, support only a limited set of precisions, it is not easy to achieve higher performance through reduced arithmetic precision. For example, the DSP block of Xilinx FPGAs, DSP48E1, can perform a 25x18-bit multiplication only, and there is no way to perform two 8x8-bit multiplications simultaneously on the same DSP block for higher throughput. This paper is about how to turn an ordinary DSP block of an off-the-shelf FPGA device into a 2-way SIMD (SingleInstruction Multiple-Data) MAC (Multiply-and-Accumulate) unit, that can deliver 4 ops/cycle by

performing two multiply-and-add operations simultaneously with reduced data width.

Here firstly, the operands are loaded in the multiplier. The arithmetic operations like addition and multiplication operations are performed. The obtained result of this will be saved in the barrel shifter. 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.

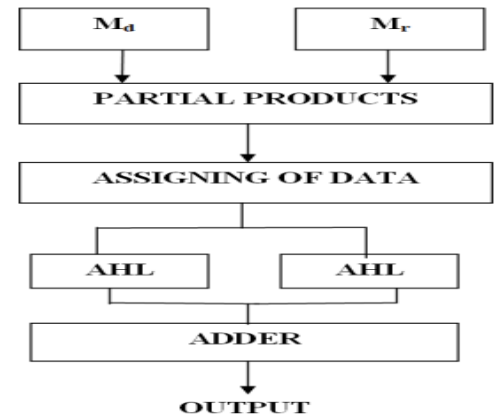


Fig.2. Proposed system.

The barrel shifter consists of root and load m_r 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 barrel shifter. 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 barrel shifter block. The both $a(t)$ and $b(t)$ values are assigned in the barrel shifter blocks. The

obtained values in the barrel shifter block will shift the bits to adder block. This block will perform the addition operation. After performing particular operation, the bits are shifted to the result register. This result register will save the output as product. At last the barrel shifter will perform the parallel operation in effective way.

PARTIAL PRODUCTS Partial-Product Multiplication is an alternative method for solving multi-digit multiplication problems. This is a strategy that is based on the distributive (grouping) property of multiplication. The first partial product is created by the LSB of the multiplier, the second partial product is created by the second bit in the multiplier, etc. The final partial products are added with a accurate adder circuit.

ADDER The accurate adder is decided by architecture/system-level applications. A self configuration technique has been proposed for the scenarios where architecture/system-

level choice is either unclear or difficult. A carry is propagated through several consecutive bits because of the actual path delay is large. When the actual carry propagation chain is short, there is no need to use approximation configuration, which is intended to cut carry chain shorter.

SIMULATION RESULTS

Viewing an RTL schematic opens an NGR file that can be viewed as a gate-level schematic. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND gates, and OR gates, that are independent of the targeted Xilinx device.

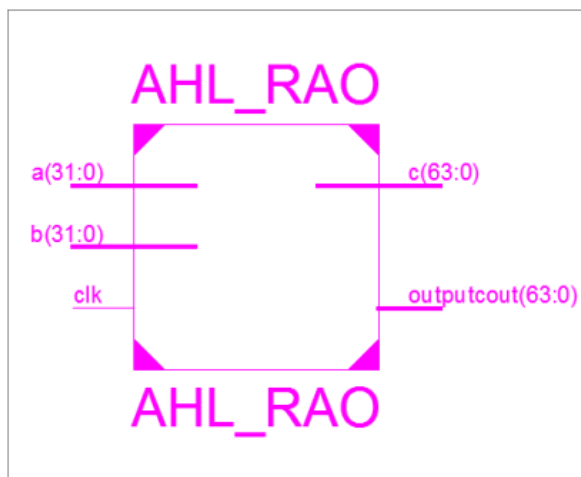


Fig.3. RTL SCHEMATIC.

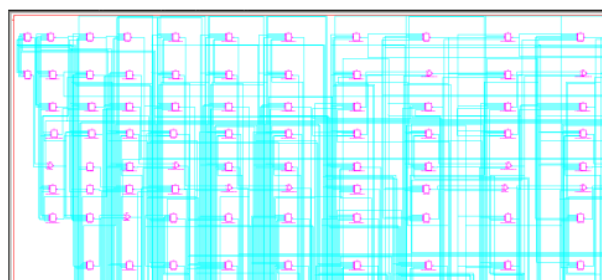


Fig.4. TECHNOLOGY SCHEMATIC.

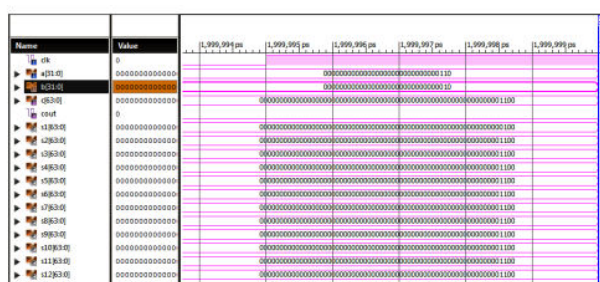


Fig.5. Fig. OUTPUT WAVEFORM.

The synthesis report of proposed system. In this report the delay and memory usage is given in detail manner. Total delay is

actually divided into two types. They are logic delay and route delay. Memory usage of proposed system is also given in detail manner.

```

LUT6:I3->O      5  0.086  0.618  p31/x2/c231 (p31/x2/c23)
LUT6:I3->O      5  0.086  0.618  p31/x2/c251 (p31/x2/c25)
LUT6:I3->O      5  0.086  0.618  p31/x2/c271 (p31/x2/c27)
LUT6:I3->O      4  0.086  0.675  p31/x2/c291 (p31/x2/c29)
LUT6:I0->O      2  0.086  0.905  p31/x2/Mxor_sum<30>_Result1 (*30<62>)
LUT6:I0->O      1  0.086  0.286  p32/x2/Mxor_sum<30>_Result1 (c_62_OBUF
OBUF:I->O        2.144  c_62_OBUF (c<62>)
-----
Total              56.862ns (9.502ns logic, 47.360ns route)
                    (16.7% logic, 83.3% route)

-----

Total REAL time to Xst completion: 58.00 secs
Total CPU time to Xst completion: 58.95 secs

-->

Total memory usage is 450024 kilobytes
Number of errors   : 0 ( 0 filtered)
Number of warnings : 3 ( 0 filtered)
Number of infos   : 0 ( 0 filtered)

```

Fig.6. SYNTHESIS REPORT.

V. CONCLUSION

Hence in this project design a high speed and area efficient reconfigurable arithmetic operations using AHL was implemented. The multiplier performs the multiplication in very fast. The proposed system will reduce the switching activities that are produce in the system. To reduce the delay partial product unit is introduced. This system is mainly used in the applications of low delay and high speed applications.

REFERENCES

- [1] Tayab D Memon, Aneela Pathan, "An Approach to LUT Based Multiplier for Short Word Length DSP Systems", 978-1-5386-5689-1/18/\$31.00 ©2018 IEEE.
- [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 lookahead 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 ultralow 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. 71

[11] Fatemeh Kashfi, S. Mehdi Fakhraie, Saeed Safari,”