International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Low Delay and Area of 8-bit & 16-bit Multiplier using Modified 4:2 and 7:3 Compressor

Juveria Khan[1], Prof. Rajesh Sharma[2], Prof. Abhishek Agwekar[3]

M. Tech. Scholar, Department of Electronics and Communication, T.C.S.T, Bhopal, India[1]

Assistant Professor, Department of Electronics and Communication, T.C.S.T, Bhopal, India[2]

Head of Department, Department of Electronics and Communication, T.C.S.T., Bhopal, India[3]

ABSTRACT: Multiplication is an important function in arithmetic operations. A CPU (central processing unit) devotes a considerable amount of processing time in performing arithmetic operations. Multiplication requires substantially more hard-ware resources and processing time than addition and sub-traction. Digital signal processors (DSPs) are the technology that is omnipresent in engineering Discipline. Fast multiplication is very important in DSPs for digital filter, convolution, Fourier transforms etc. In this proposed research work an attempt will make for making a novel multiplier using Nikhilam Sutra and Kogge stone adder. The proposed multiplier will have not only fast response but also having less number of component, area and power consumption.

KEYWORDS: Vedic Multiplier, Compressor, Xilinx Simulation.

I. INTRODUCTION

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) unit and inner products are some of the frequently used Computation- Intensive Arithmetic Functions currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filter circuits and in microprocessors in its arithmetic and logic unit (ALU). Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

In this work we have put into effect a high speed Vedic multiplier using barrel shifter. The sutra was implemented by modified design of "Nikhilam Sutra" due to its feature of reducing the number of partial products. The barrel shifter is used at different levels of designs to reduce the delay when compared to conventional multipliers. The hardware implementation of Vedic multiplier is using barrel shifter contributes to adequate improvement of the speed.

In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of algorithm. The speed of multiplication operation is great importance in DSP as well as in general processor. In past multiplication was implemented with a sequence of addition, subtraction and shift operations. There have been many algorithms proposals to perform the multiplication, and each offering different advantages and having in terms of speed, circuit complexity, area and power consumption.

The multiplier is a fairly large block of a computing system. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Latency is the real delay of computing a function, a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time multiplier is not only a high delay block but also a major source of power dissipation. That's why if one also aims to minimize power consumption, it is of great interest to reduce the delay by using various delay optimizations.

Advanced multipliers are the center parts of all the computerized signal processors (DSPs) and the rate of the DSP is generally controlled by the velocity of its multipliers. Two most basic duplication calculations followed in the computerized equipment are exhibit increase calculation and Booth augmentation calculation. The calculation time taken by the exhibit multiplier is relatively less on the grounds that the halfway items are ascertained autonomously in parallel. The postponement connected with the exhibit multiplier is the time taken by the signs to spread through the

entryways that shape the Multiplication cluster. Corner increase is another vital augmentation calculation. Extensive corner clusters are required for fast duplication and exponential operations which thus require expansive halfway aggregate and incomplete convey registers. Duplication of two n-bit operands utilizing a radix-4 corner recording multiplier requires roughly n/(2m) clock cycles to create the minimum noteworthy portion of the last item, where m is the quantity of Booth recorder snake stages. Hence, an extensive spread deferral is connected with this case.

The rest of the paper is sorted out as takes after: Literature survey of Vedic multiplier utilizing barrel shifter is introduced as a part of Section II. Brief portrayals of Vedic multiplier are displayed in Section III. Outline of Vedic multiplier utilizing barrel shifter structural planning as a part of Section IV. Confirmed the outcome in area V. Conclusion is introduced in Section VI.

II. LITERATURE REVIEW

B.Madhu Latha et al. [1], a 8-bit Vedic multiplier is enhanced as far as transmission deferral when contrast and the additional unsurprising multipliers. We have utilized 8-bit barrel shifter which desires for stand out clock cycle for "n" measure of movements in our anticipated configuration. The course of action is executed and checked utilizing FPGA and ISE Simulator. The focal part was executed on Xilinx Spartan-6 family xc6s1x75T-3-fgg676 FPGA. The transmission deferral complexity was excerpted from the blend report and static timing report as well. The basic configuration may achieve engendering postponement of 6.781ns by method for barrel shifter in base determination module and multiplier.

A Murali et al. [2], execution of Vedic multiplier is upgraded for spread postponement when contrasted and other ordinary multiplier like exhibit multiplier, Braun multiplier, altered corner multiplier and Wallace tree multiplier. For math duplications different Vedic augmentation methods are utilized. It has been found that Urdhva tiryakbhyam Sutra is most productive Sutra, giving least defer for increase of a wide range of numbers, either little numbers or huge numbers. In our configuration we have used 8-bit barrel shifter which requires stand out clock for "n" number of movements. The configuration is actualized and confirmed utilizing FPGA and Mentor Graphics Simulators. The center was actualized on Xilinx Spartan-3E crew. The engendering postponement examination was separated from the amalgamation report and static timing report also. The configuration could accomplish proliferation deferral of 6.771ns utilizing barrel shifter as a part of base choice module and multiplier.

Mrs.Toni J.Billore et al. [3], this paper portrays the usage of a 8-bit Vedic multiplier utilizing quick viper improved as a part of terms of proliferation postponement when contrasted and ordinary multiplier. In our outline of 8 bit Vedic multiplier utilizing quick snake, we have used 8-bit barrel shifter which requires one and only clock cycle for "n" number of movements. The configuration of 8 bit Vedic multiplier utilizing barrel shifter is executed and confirmed utilizing FPGA and ISE Simulator. The center utilized here was actualized on Altera Cyclone® II 2C20 FPGA gadget programming. The proliferation postponement between 8 bit Vedic multiplier utilizing barrel shifter utilizing barrel shifter and utilizing quick snake examination was removed from the union report and static timing report too. The configuration which is executed here could accomplish spread deferral of 6.781ns utilizing barrel shifter obstruct as a part of base determination module and multiplier of building design utilized. In our undertaking, we make a correlation between execution investigation of 8 bit Vedic multiplier utilizing barrel shifter and utilizing quick viper.

Pavan Kumar et al. [4], This paper describes the implementation of an 8-bit Vedic multiplier enhanced in terms of propagation delay when compared with conventional multiplier like array multiplier, Braun multiplier, modified booth multiplier and Wallace tree multiplier. In our design we have utilized 8-bit barrel shifter which requires only one clock cycle for 'n' number of shifts. The design is implemented and verified using FPGA and ISE Simulator. The core was implemented on Xilinx Spartan-6 family xc6s1x75T-3-fgg676 FPGA. The propagation delay comparison was extracted from the synthesis report and static timing report as well. The design could achieve propagation delay of 6.781ns using barrel shifter in base selection module and multiplier.

III. DIFFERENT TYPES OF ADDER

Ripple carry is a combinational circuit for adding more than two bit information. It is also called parallel adder. Ripple carry adder can be designed by using full adder in cascading form. Carry output of first full adder is connected with input of the next full adder, so carry is rippled from one adder to another adder. That is by it is called ripple carry adder.
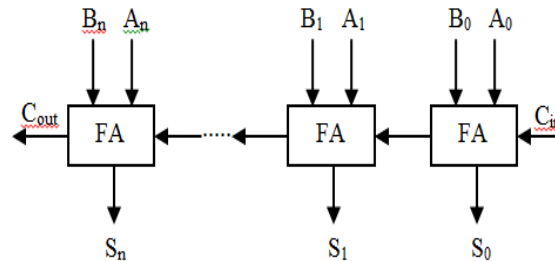
Figure 2: N-bit Ripple Carry Adder

$$S_0 = A_0 \oplus B_0 \oplus C_{in} \tag{1}$$

$$C_0 = (A_0 * B_0) + (B_0 * C_{in}) + (C_{in} * A_0) \tag{2}$$

$$S_1 = A_1 \oplus B_1 \oplus C_0 \tag{3}$$

$$C_1 = (A_1 * B_1) + (B_1 * C_0) + (C_0 * A_1) \tag{4}$$

$$S_2 = A_2 \oplus B_2 \oplus C_1 \tag{5}$$

$$C_2 = (A_2 * B_2) + (B_2 * C_1) + (C_1 * A_2) \tag{6}$$

$$S_n = A_n \oplus B_n \oplus C_{n-1} \tag{7}$$

$$C_n = (A_n * B_n) + (B_n * C_{n-1}) + (C_{n-1} * A_n) \tag{8}$$

IV. VEDIC MULTIPLIER

As specified prior, Vedic Mathematics can be isolated into 16 unique sutras to perform scientific counts. Among these the Urdhwa Tiryakbhyam Sutra is one of the most exceedingly favored calculations for performing increase. The calculation is sufficiently able to be employed for the duplication of whole numbers and also binary numbers. The expression "Urdhwa Tiryakbhyam" started from 2 Sanskrit words Urdhwa and Tiryakbhyam which mean "vertically" and "transversely" respectively. It depends on a novel idea through which the era of every single fractional item should be possible with the simultaneous expansion of these halfway items. The calculation can be summed up for n x n bit number. Since the incomplete items and their totals are figured in parallel, the multiplier is free of the clock recurrence of the processor. In this way the multiplier will require the same measure of time to figure the item and henceforth is free of the clock recurrence.

The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient.

V. VEDIC MULTIPLIER USING COMPRESSOR BASED ADDER

The multiplication of two numbers is done by using Urdhwa Triyakbhyam. Here first the least significant bits of the two digits are multiplied. Then the intermediate digits are cross multi-plied and added together. After this the most

significant digits are multiplied. For the 16X16 bit multiplication small block of 2X2 or 4X4 or 8X8 multiplier were used in parallel to make the process easy and efficient.

Compressor Based Adder
4:2 Compressor
A 4:2 compressor is capable of adding 4 bits and one carry, in turn producing a 3 bit output. The 4-2 compressor has 4 inputs X_1, X_2, X_3 and X_4 and 2 outputs Sum and Carry along with a Carry-in (C_{in}) and a Carry-out (C_{out}) as shown in figure 3. The input C_{in} is the output from the previous lower significant compressor.

The C_{out} is the output to the compressor in the next significant stage. The critical path is smaller in comparison with an equivalent circuit to add 5 bits using full adders and half adders.
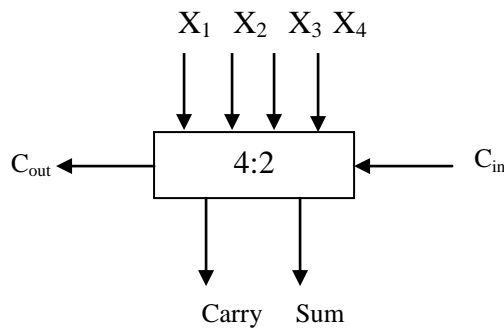

Figure 3: Block Diagram of 4:2 Compressors

Similar to the 3-2 compressor the 4-2 compressor is governed by the basic equation

$$X_1 + X_2 + X_3 + X_4 + C_{in} = sum + 2*(Carry + C_{out})$$

The standard implementation of the 4-2 compressor is done using 2 Full Adder cells as shown in figure 4. When the individual full Adders are broken into their constituent XOR blocks, it can be observed that the overall delay is equal to 4*XOR.
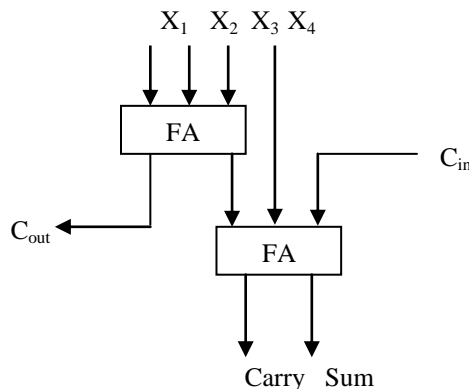

Figure 4: Logic Diagram of 4:2 Compressors

Modified 4:2 Compressor

The block diagram in figure 3 shows the existing architecture for the implementation of the 4-2 compressor with a delay of 3*XOR. The equations governing the outputs in the existing architecture are shown below

$$Sum = X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus C_{in}$$
$$C_{out} = (X_1 \oplus X_2).X_3 + (X_1 \oplus X_2).X_1$$
$$C_{arry} = (X_1 \oplus X_2 \oplus X_3 \oplus X_4).C_{in} + (\overline{X_1 \oplus X_2 \oplus X_3 \oplus X_4}).X_4$$

In show the delay and area all the modified 4:2 compressors. 4:2 compressors replacing some XOR blocks with multiplexer's results in a significant improvement in delay. Also the MUX block at the SUM output gets the select before the input arrive and thus the transistors are already switched by the time they arrive.
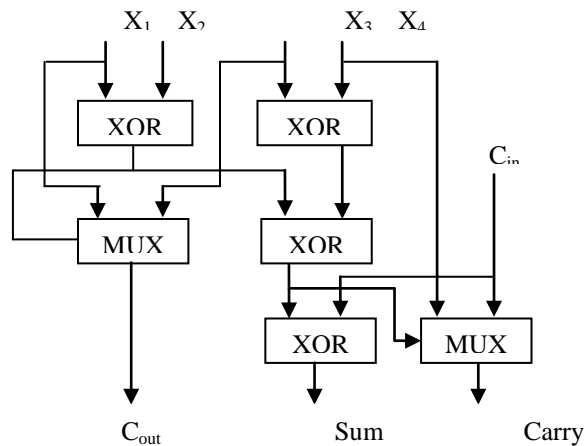

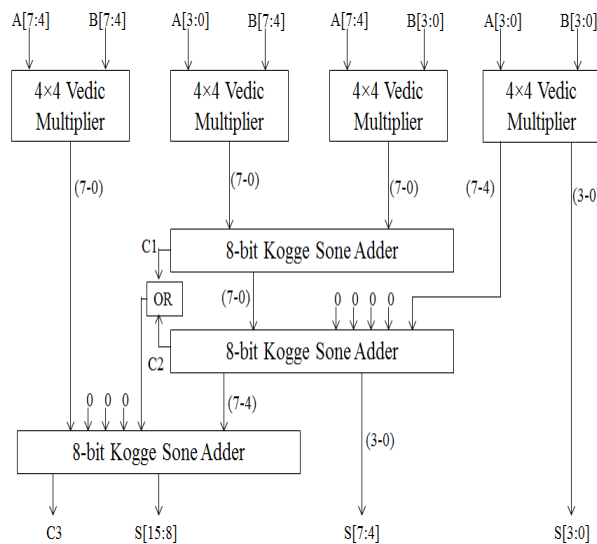Figure 5: Logical Diagram of Modified 4:2 Compressor

Figure 6: Logic Diagram of 8-bit Vedic Multiplier using Kogge Stone Adder

In our proposed method the high speed carry select adder is replaced by the carry select adder along with Kogge Stone (KS) adder which claims to provide a better speed and less propagation delay. Here we have used four multiplier of 8 bit to perform 16 bit multiplication. The method used is the addition of all partial product formed by the cross multiplication of one bit with another. The LSB bits of first multiplier P1 (7-0) gives the LSB bits Q (7-0) of the final output. Another bits of first multiplier P1 (15-8) are added in series with LSB 8 bits of second multiplier to form the 16 bits, which in turn get added with 16 bits of third multiplier by using KS Adder. The LSB bits of the output of KS adder forms the Q (15-8) bits of the final output. The remaining 8 bit P2(15-8) is then added with the left 8 bits of KS output to from 16 bits, which is then added with 16 bits of the fourth multiplier by using KS 2 adder. The output from KS 2 adder forms the Q (31-16) bits. This is how the 32bit output is achieved in the less possible time.

VI. SIMULATION RESULT

All the designing and experiment regarding algorithm that we have mentioned in this paper is being developed on Xilinx 14.1i updated version. Xilinx 9.2i has couple of the striking features such as low memory requirement, fast debugging, and low cost. The latest release of ISE^{TM} (Integrated Software Environment) design tool provides the low memory requirement approximate 27 percentage low. ISE 14.1i that provides advanced tools like smart compile technology with better usage of their computing hardware provides faster timing closure and higher quality of results for a better time to designing solution. ISE 14.1i Xilinx tools permits greater flexibility for designs which leverage embedded processors. The ISE 14.1i Design suite is accompanied by the release of chip scope Pro^{TM} 14.1i debug and verification software. By the aid of that software we debug the program easily. Also included is the newest release of the chip scope Pro Serial IO Tool kit, providing simplified debugging of high-speed serial IO designs for Virtex-4 FX and Virtex-5 LXT and SXT FPGAs. With the help of this tool we can develop in the area of communication as well as in the area of signal processing and VLSI low power designing. To simplify multi rate DSP and DHT designs with a large number of clocks typically found in wireless and video applications, ISE 14.1i software features breakthrough advancements in place and route and clock algorithm offering up to a 15 percent performance advantage. Xilinx 14.1i Provides the low memory requirement while providing expanded support for Microsoft windows Vista, Microsoft Windows XP x64, and Red Hat Enterprise WS 5.0 32-bit operating systems.
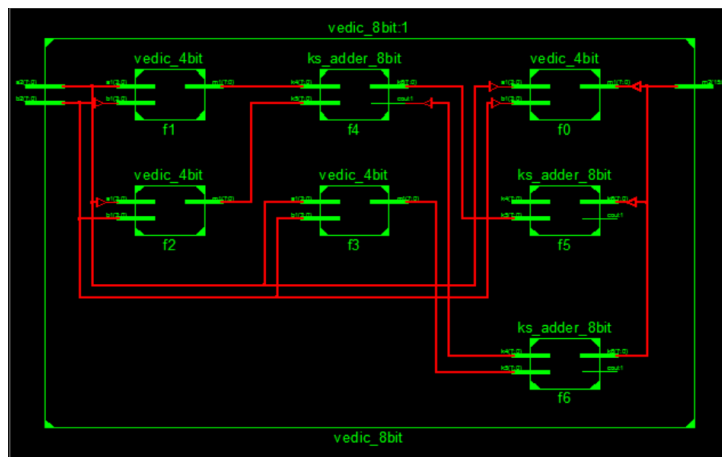

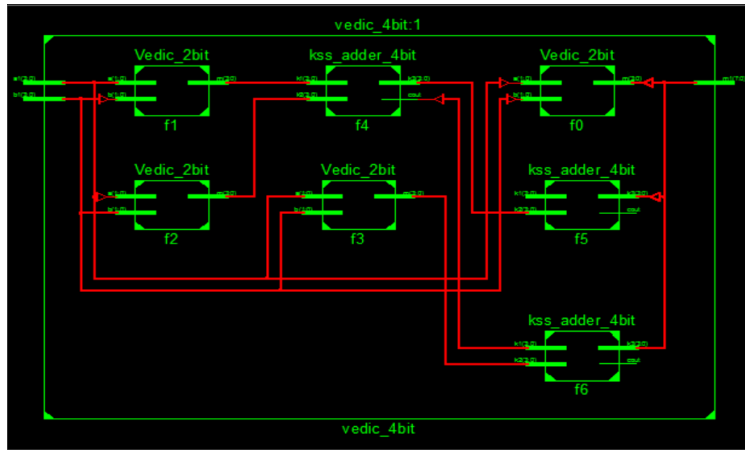Figure 7: RTL View of 8-bit Vedic Multiplier
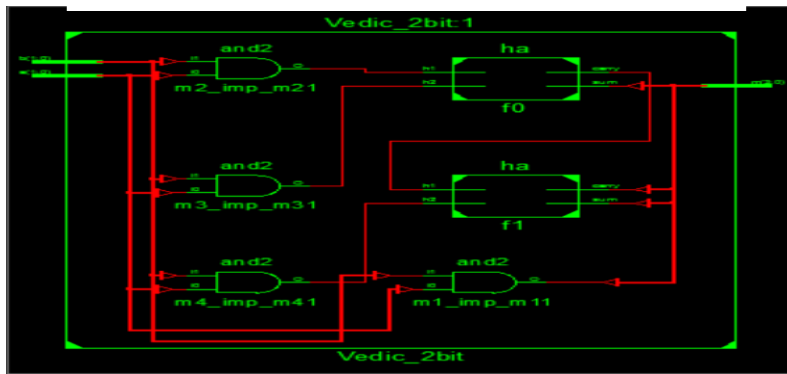
Figure 8: RTL View of 4-bit Vedic Multiplier

Figure 9: RTL View of 2-bit Vedic Multiplier

```
Device utilization summary:
---------------------------

Selected Device : 3s50tq144-5

 Number of Slices:                   113  out of    768    14%
 Number of 4 input LUTs:             200  out of   1536    13%
 Number of IOs:                       32
 Number of bonded IOBs:               32  out of     97    32%
```

Figure 10: Device Utilization 8-bit Vedic Multiplier

```
Timing Summary:
---------------

Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 24.180ns
```

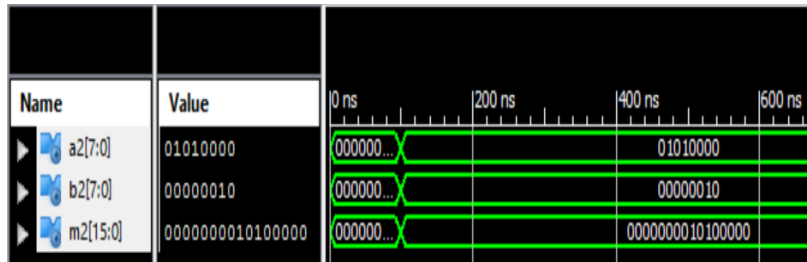Figure 11: Timing Summary for 8-bit Vedic Multiplier

Figure 12: Output Waveform of the 8-bit Vedic Multiplier

Table I: Device Summary

Design	Width	Area count	MCPD
P. Y. Bhavani [2014]	8-bit	1545	41.696 ns
G. Gokhale [2015]	8-bit	1380	45.678 ns
G. Gokhale [2015]	8-bit	1293	44.358 ns
Proposed Vedic Multiplier	8-bit	978	24.180 ns

Table II: Device Summary

| Logic Utilization	Previous 16-bit Vedic Multiplier	Implemented 16-bit Vedic Multiplier
Number of Slice	493	309
Number of LUTs	1243	538
Number of IOBs	66	64
MCPD (ns)	38.82	17.600
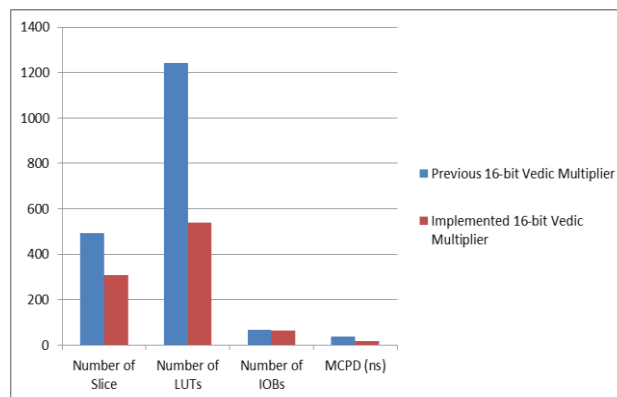
VI. CONCLUSION

The high speed implementation of such a multiplier has wide range of applications in image processing, arithmetic logic unit and VLSI signal processing. The proposed 8x8 Vedic multiplier architecture has been designed and synthesized using on Spartan 3 XC3S400 board. The proposed Vedic Multiplier with carry select adder is compared with the existing Vedic multiplier using Carry select adder along with Common Boolean Logic and can be inferred that proposed architecture is faster compared to existing Vedic multiplier. In future the proposed multiplier performance

parameters can be improved by high level pipelining operations and applied in signal processing applications like image processing and video processing.

REFERENCES

[1] G. Gokhale and P. D. Bahirgonde, "Design of Vedic Multiplier using Area-Efficient Carry Select Adder", 4th IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015), Kochi, August 10-13, 2015, India.

[2] G. Gokhale and Mr. S. R. Gokhale, "Design of Area and Delay Efficient Vedic Multiplier Using Carry Select Adder", 4th IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015), Kochi, August 10-13, 2015, India.

[3] Pavan Kumar, Saiprasad Goud A, and A Radhika had published their research with the title "FPGA Implementation of high speed 8-bit Vedic multiplier using barrel shifter", 978-1-4673-6150-7/13 IEEE.

[4] B.Madhu Latha1, B. Nageswar Rao, published their research with title "Design and Implementation of High Speed 8-Bit Vedic Multiplier on FPGA" International Journal of Advanced Research in Electrical ,Electronics and Instrumentation Engineering, Vol. 3, Issue 8, August 2014.

[5] A Murali, G Vijaya Padma , T Saritha, published their research with title "An Optimized Implementation of Vedic Multiplier Using Barrel Shifter in FPGA Technology", Journal of Innovative Engineering 2014, 2(2).

[6] Sweta Khatri , Ghanshyam Jangid, "FPGA Implementation of 64-bit fast multiplier using barrel shifter" Vol. 2 Issue VII, July 2014 ISSN: 2321-9653.

[7] Toni J.Billore, D.R.Rotake, "FPGA implementation of high speed 8 bit Vedic Multiplier using Fast adders" Journal of VLSI and Signal Processing, Volume 4, Issue 3, Ver. II (May-Jun. 2014), PP 54-59 e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197.

[8] S. S. Kerur, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V A, "Implementation of Vedic Multiplier for Digital Signal processing" International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011.

[9] Vaibhav Jindal, Mr. Navaid Zafar Rizvi, Dinesh Kumar Singh "VHDL Code of Vedic Multiplierwith Minimum Delay Architecture" National Conference on Synergetic Trends in engineering and Technology (STET-2014) International Journal of Engineering and Technical Research ISSN: 2321-0869, Special Issue.

[10] Bhavin D Marul, Altaf Darvadiya "VHDL Implementation of 8-Bit Vedic Multiplier Using Barrel Shifter" International Journal for Scientific Research & Development| Vol. 2, Issue 01, 2014 | ISSN (online): 2321-0613.
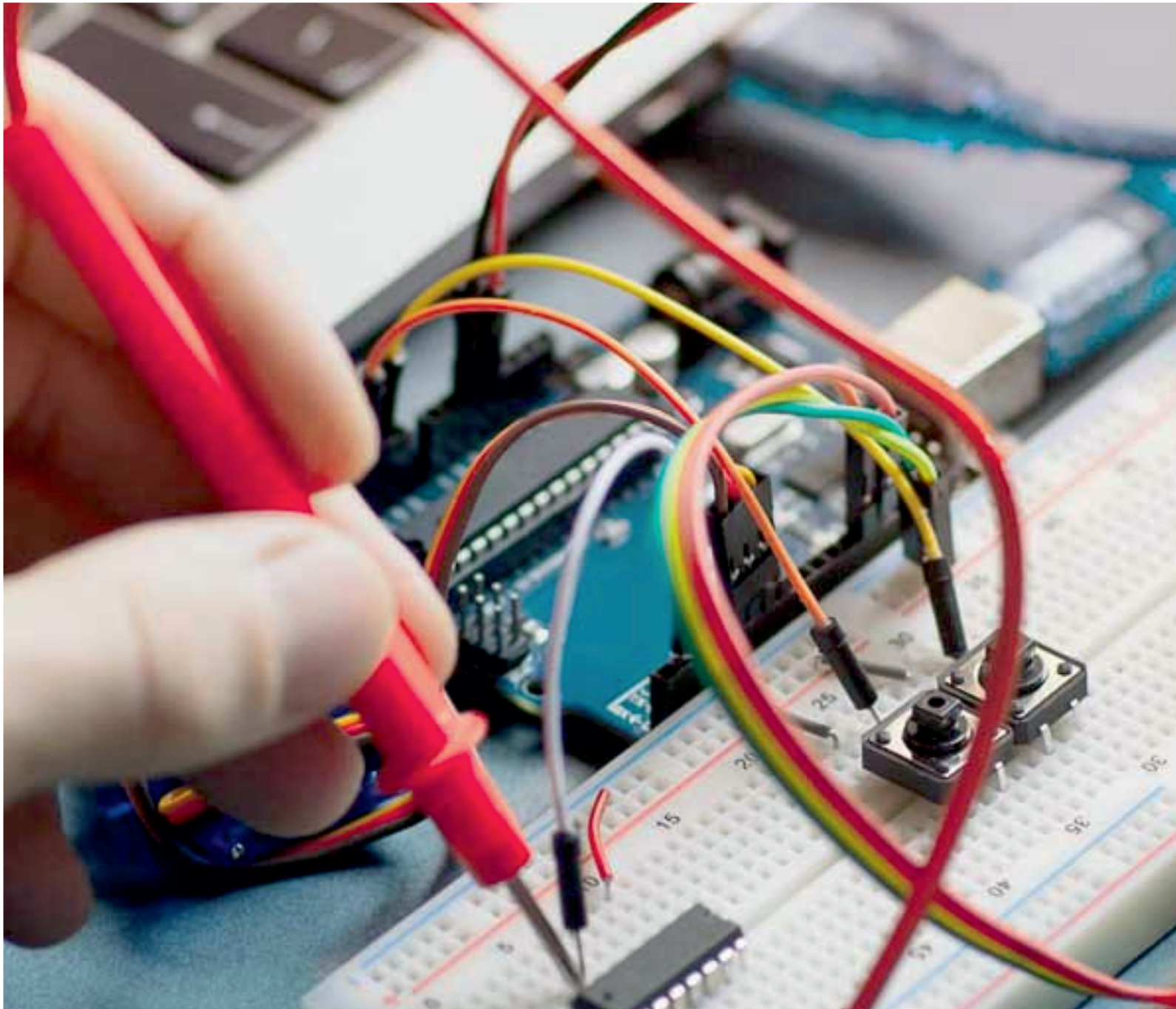
International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

9940 572 462   6381 907 438   ijareeie@gmail.com