Design and Implementation of 4 bit Ripple Carry Adder Multiplier and 8 bit ALU on FPGA using VHDL

G. VENKATA GIRISH
Dr. A. KAMALA KUMARI
V. YASWANTH
B. MANOJ KUMAR
girishguntupalli99@gmail.com
Dept. of INSTRUMENT TECHNOLOGY
ANDHRA UNIVERSITY COLLEGE OF ENGINEERING (A), VISAKHAPATNAM
ANDHRA PRADESH, INDIA

ABSTRACT :- **The aim of this paper is to design and implement the Arithmetic circuits (adder & multiplier) and ALU on fpga board using VHDL and it is accomplished using Xilinx family Spartan-6 XC6SLX9 Field Programmable Gate Array (FPGA) device. Since in this technological era, the application of digital signal processing is very wide and it will enhance in the future as well. The operations performed in the digital signal processing domain are the combination of addition and multiplication, the arithmetic circuits (adder & multiplier) are the core of DSP hardware. In this paper, different types of adder multiplier circuits have been implemented on Field Programmable Gate Array (FPGA) and an analysis has been done to find out the most applicable arithmetic circuits for FPGA based digital signal processing implementation. ALU of digital computers is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. The hardware can only perform a relatively simple and primitive set of Boolean & arithmetic operations and are based on a hierarchy of operations that are built by using algorithms employing the hardware.**

Index Terms:**--4 bit Ripple Carry Adder, 4 bit Multiplier, FPGA, 8 Bit ALU, Xilinx ISE 14.7,Edge Spartan 6 board.**

I.INTRODUCTION

Since there is rapid advancement in communication and multimedia systems, the demand of large capacity data processing and real-time signal processing is rising. The adder and multiplier are an important element of digital signal processing such as convolution and filtering.

The binary adder is the mandatory element in most of the digital circuit design including digital signal processors and microprocessors data path units. The design of these adders should have low power consumption with low cost and reduced area. This paper attempts to examine the features of certain adder circuits such as 4 bit ripple carry and which guarantees superior performance compared to existing circuits. In fast techno world designing adder circuits is so fast but the designed model should satisfy the necessary conditions like less power consumption, minimized use of area and reliable delay.This paper concerns about implementation and analysis of the structure and operation of ripple carry adder calculating its efficiency and performance based on its simulation results and implemented using FGPA.

Field Programmable Gate Arrays (FPGAs) combine limited cost and reconfigurability with very high integration capability and performances. Such characteristics, along with reduced low volume costs make them a valid alternative to the more complex and time to market demanding Application Specific Integrated Circuits (ASICs). Addition is the main operation of each arithmetic circuit, thus improving speed performances and reducing the area occupancy of adder circuits is still an enterprising research topic. Unfortunately, as it is well known, designing efficient adders using an FPGA platform is not trivial. In fact, many of the fast adder architectures existing in literature cannot optimally exploit dedicated routing resourcesavailable within FPGAs.

This paper is based on the design and implementation of fixed point arithmetic circuits (adders & multipliers circuits)and ALU with respect to delay calculation. This paper is mainly focused on enhancement of FPGA based design and implementation of Ripple carry adder ,multiplier and ALU for digital signal processing application. It is relatively extensive work that provides very good analysis for FPGA based design of the arithmetic circuits for DSP application. Here,a Spartan-6 XC6SLX9 FPGA is used for implementation purpose due to its arithmetic-support features.

II. FPGA

Field Programmable Gate Arrays (**FPGAs**) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. They are so called "Field Programmable" because **FPGAs** can be reprogrammed to desired application or functionality requirements after manufacturing.

Xilinx Spartan board contain FPGA board as an integrated circuit that can be customized for a specific application.

Unlike traditional programmable logic blocks that can be wired in different configurations. These blocks create a

physical array of logic gates that can be used to perform different operations. Because the gates are customizable, FPGAs can be optimized for any computing task. This gives FPGAs the potential to perform operations several times faster than a hard-wired processor. Field-programmable gate arrays are typically customized using a hardware description language, or HDL. A programmer can use HDL commands to configure the gate interconnects (how the gates connect to each other) as well as the gates themselves. For example, a gate may be assigned a boolean operator, such as AND, OR, or XOR. By linking several gates together, it is possible to perform advanced logic operations. FPGAs have a wide variety of field applications.
Examples include telecommunications, data centers, scientific computing, and audio/video processing. Besides being used in servers and high-end computers, they can also be implemented in electronic devices, such as TVs, radios, and medical equipment.

Three major vendors provide FPGA hardware.

(1) Xilinx inc

(2) Altera corp

(3) Laltice semiconductor corp

III. XLINX ISE

Xilinx ISE (Integrated Synthesis Environment)[3] is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a source code editor (Workplace), an output console (Transcript), and a processes tree (Processes). The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure.For single-chip designs there may be one main module, with other modules included by the main module, similar to the `main()` subroutine in C++ programs. Design constraints are specified in modules, which include pin configuration and mapping.

IV. SPARTAN-6 BOARD

Spartan-6 FPGAs offer the best solution for high-volume logic designs, consumer-oriented DSP designs, and cost-sensitive embedded applications. Spartan-6 FPGAs are the programmable silicon foundation for Targeted Design Platforms that deliver integrated software and hardware components that enable designers to focus on innovation as soon as their development cycle begins.

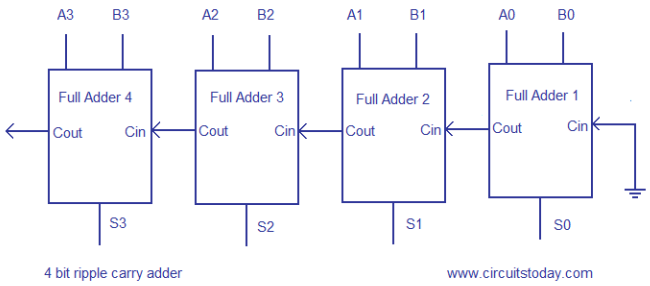
FEATURES:

Spartan-6 Family:
• Spartan-6 LX FPGA: Logic optimized
• Spartan-6 LXT FPGA: High-speed serial connectivity
• Designed for low cost
• Multiple efficient integrated blocks
• Optimized selection of I/O standards
• Staggered pads
• High-volume plastic wire-bonded packages
• Low static and dynamic power
• 45 nm process optimized for cost and low power
• Hibernate power-down mode for zero power
• Suspend mode maintains state and configuration with multi-pin wake-up, control enhancement
• Lower-power 1.0V core voltage (LX FPGAs, -1L only)
• High performance 1.2V core voltage (LX and LXT FPGAs, -2, -3, and -3N speed grades)
• Multi-voltage, multi-standard SelectIO™ interface banks
• Up to 1,080 Mb/s data transfer rate per differential I/O
• Selectable output drive, up to 24 mA per pin
• 3.3V to 1.2V I/O standards and protocols
• Low-cost HSTL and SSTL memory interfaces
• High-speed interfaces including: Serial ATA, Aurora, 1G Ethernet, PCI Express, OBSAI, CPRI, EPON, GPON, DisplayPort, and XAUI.

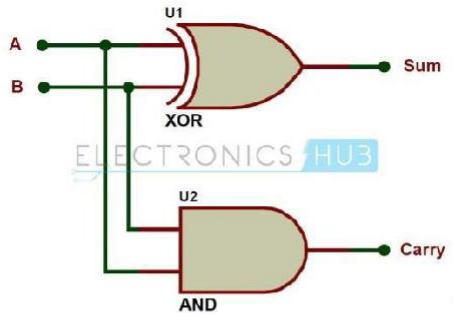
V.DESIGN METHODOLOGIES

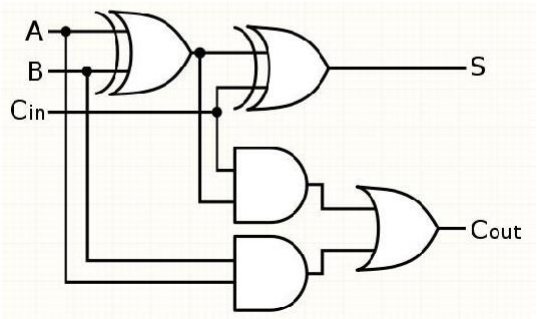
A. DESIGN OF RIPPLE-CARRYADDER

Multiple full adder circuits can be cascaded in parallel to add an N-bit number. For an N- bit parallel adder, there must be N number of full adder circuits. A ripple carry adder is a logic circuit in which the carry-out of each full adder is the carry in of the succeeding next most significant full adder. It is called a ripple carry adder because each carry bit gets rippled into the next stage. In a ripple carry adder the sum and carry out bits of any half adder stage is not valid until the carry in of that stage occurs.Propagation delays inside the logic circuitry is the reason behind this. Propagation delay is time elapsed between the application of an input and occurance of the corresponding output. Consider a NOT gate, When the input is “0” the output will be “1” and vice versa. The time taken for the NOT gate’s output to become “0” after the application of logic “1” to the NOT gate’s input is the propagation delay here. Similarly the carry propagation delay is the time elapsed between the application of the carry in signal and the occurance of the carryout (Cout) signal. Circuit diagram of a 4-bit ripple carry adder is shown below.



Sum out S0 and carry out Cout of the Full Adder 1 is valid only after the propagation delay of Full Adder 1. In the same way, Sum out S3 of the Full Adder 4 is valid only after the joint propagation delays of Full Adder 1 to Full Adder 4. In simple words, the final result of the ripple carry adder is valid only after the joint propogation delays of all full adder circuits inside it.





A Full adder can be made by combining two half adder circuits together (a half adder is a circuit that adds two input bits and outputs a sum bit and a carry bit).

B. DESIGN OF MULTIPLIER

The most basic form of multiplication consists of forming the product of two binary numbers m and n. (m×n) bit multiplication can be viewed as forming n partial product of m bits each, and then summing appropriately shifted partial products to produce an (m+n) bit result P. Binary multiplication is equivalent to a logical AND operation.
The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below:

Y= yn-1 yn-2 .....y2 y1 y0(multiplicand)
X= xn-1 xn-2..... X2 x1 x0(multiplier).

AND gates are used to generate the Partial Products, If the multiplicand is N-bits and the Multiplier is M-bits then there is N* M partial product. The way that the partial products are generated or summed up is the difference between the different architectures of various multipliers . Multiplication of binary numbers can be decomposed into additions.
Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

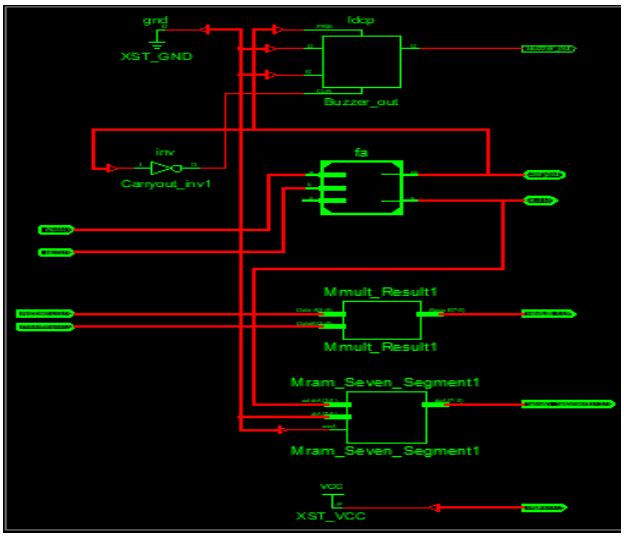
C. DESIGN OF 8 BIT ALU

The Arithmetic Logic Unit (ALU) is a fundamental building block of the Central Processing Unit (CPU) of a computer. Even one of the simplest microprocessor contains one ALU for purposes such as maintaining timers. We can say that ALU is a core component of all central processing unit within in a computer and is an integral part of the execution unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations. The ALU takes as input the data to be operated on (called operands) and a code from the control unit indicating which operation to perform. The output is the result of the computation. The ALU implemented will perform the following operations: Arithmeticoperations (addition, subtraction, increment , decrementtransfer) .
Logic operations (AND, NOT, OR, NAND, NOR, EX-OR, EX-NOR). A digital system can be represented at different levels of abstraction. This keeps the description and design of complex systems manageable. The highest level of abstraction is the behavioral level that describes a system in terms of what it does (or how it behaves) rather than in terms of its components and interconnection between them.This ALU operate on 8 bit input. It performs arithematic and logical operations. This gives appropriate output. 8 bit ALU performs 8 operations. There are 2 data inputs and one select line. According to select line input appropriate operation is performed between 2 inputs. Output of this 8 bit ALU is connected between ROM and RAM. A number of basic arethematic and bitwise logic functions are performed in the ALU.

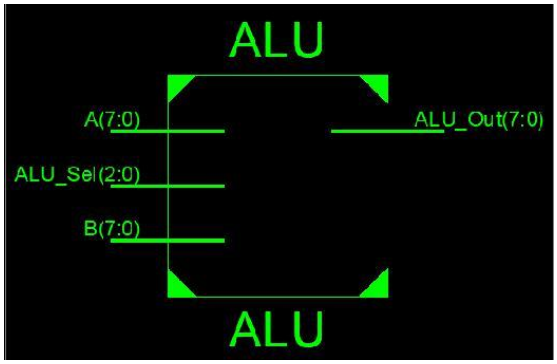
VI.RTL VIEWS OF THE PROPOSED CIRCUITS

A. RTL SCHEMATIC OF 4 BIT RIPPLE CARRY ADDER AND MULTIPLIER.

The following figure shows the register transistor logic view 4 bit ripple carry adder and 4*4 multiplier



B.RTL SCHEMATIC OF 8 BIT ALU



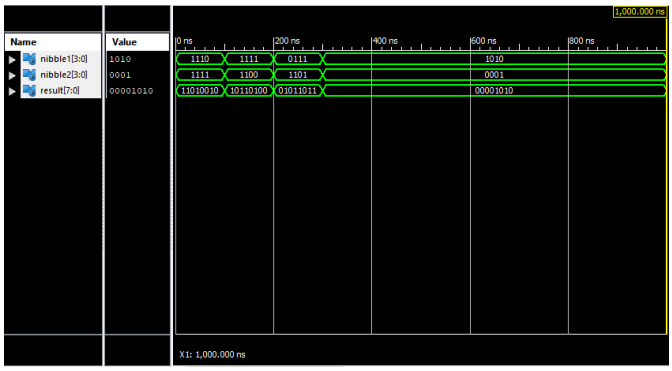
VII. SIMULATION RESULTS:

A. SIMULATION RESULTS OF 4 BIT RIPPLE CARRY ADDER AND 4*4 MULTIPLIER

The below figures show the simulated output of the 4 bit ripple carry adder and 4x4 binary multiplier designed in VHDL using Xilinx platform of ISE version 14.7.

The above circuits are implemented on hardware component i.e.. FPGA Spartan 6 edge board.

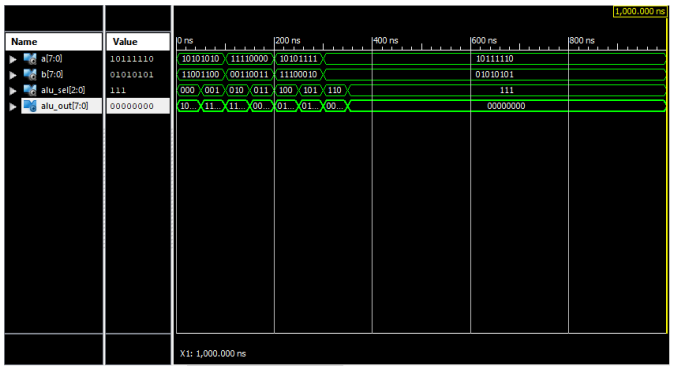
The above figure shows the simulation result of 4 bit ripple carry adder for the given inputs ‘a’, ‘b’, ‘cin’. And gives the output ‘s’, ‘Carryout’ respectively. Here ‘a’ and ‘b’ are the 4 bit inputs given by the user with an input carry ‘cin’ and ‘s’ is the generated 4 bit sum for the given inputs and carryout is the output carry generated.



The above figure shows the simulation result of 4 bit binary multiplier for the given inputs ‘Nibble1’, ‘Nibble2’ and gives the output ‘Result’ respectively. Here ‘Nibble1’ and ‘Nibble2’ are 4 bit inputs given by the user and ‘Result’ is the generated 8 bit product of the given inputs.

B. SIMULATION REULTS OF 8 BIT ALU

The below figure is the generated waveform for the simulation of the 8 bit ALU implemented using VHDL language in Xilinx software. The values for A and B are varied using the test bench code for every 100ns and the alu select line for every 50ns and the simulated form shows the generated output i.e., ALU_Out for every interval.



VIII. HARDWARE APPROACH

FPGA is a hardware circuit that a user can program to carry out one or more logical operations. Taken a step further, FPGAs are integrated circuits, or ICs, which are sets of circuits on a chip that’s the “array” part. Those circuits, or arrays, are groups of programmable logic gates, memory, or other elements.

With an FPGA, there is no chip. The user programs the hardware circuit or circuits. The programming can be a single, simple logic gate (an AND or OR function), or it can involve one or more complexfunctions, including functions that, together, act as a comprehensive multi-core processor.

The VHDL code which implies the hardware part of the propsed circuits are downloaded on FPGA processor using JTAG cable interfacing PC and the hardware element. A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device i.eFPGA, and then it is the actual hardware being configured, rather than the VHDL code being "executed" as if on some form of a processor chip

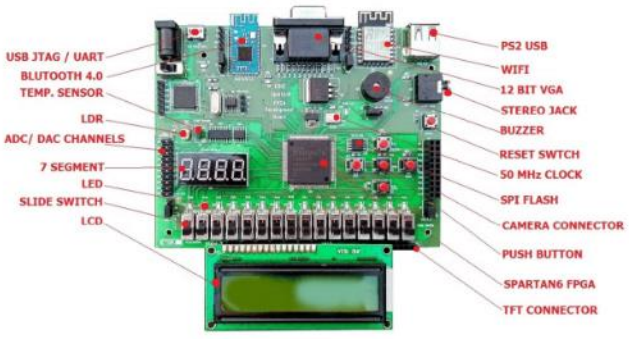
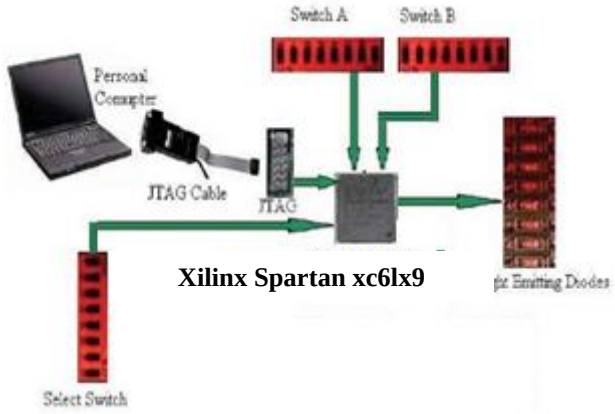


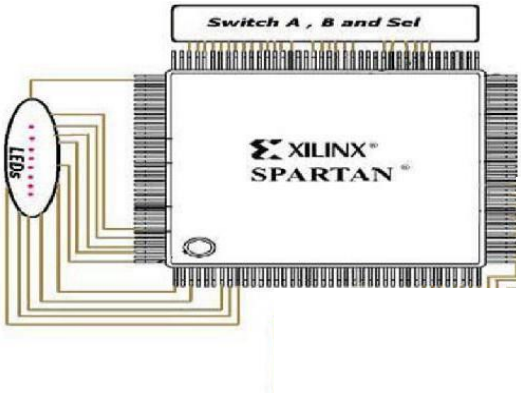
Figure 1: Features of EDGE Spartan 6 FPGA Development Board

IX. IMPLEMENTATION OF CIRCUITS ON FPGA BOARD

The VHDL coding of this paper design is compiled and simulated using Xilinx ISE 14.7 i and has been downloaded in FPGA using Spartan XC6LX9 kit as shown in Figure. The data is updated in the kit using two separate select inputs A and B each carrying 3 bits.



The function of FPGA is embedded on the kit along with PROM, LCD, LEDs and DIP switches. A Joint Test Action Group (JTAG) interface connects the FPGA chip with PROM and leads to PC through a serial interface. Since FPGA is a user programmable, therefore JTAG is of core significance. PROM has several postulates in the shape of data storage and debugging, permanent storage of data, consistency of operation, low cost, high speed and compactness. PROM IS used in this design of ALU, which is equipped with the inbuilt circuitry to support and store complex functions.



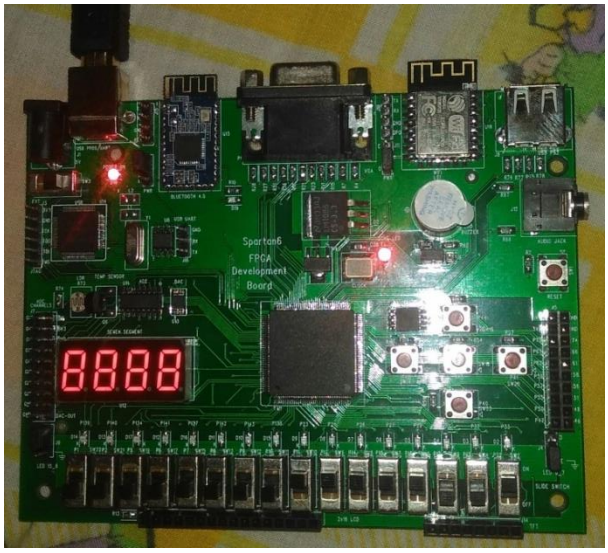
X. HARDWARE RESULTS:

A. 4 BIT RIPPLE CARRY ADDER , 4*4 MULTIPLIER

The 4 Bit Full adder and multiplier are implemented on the Spartan 6 FPGA board and the results are obtained . The inputs of four 4 bit data is given through the 16 slide switches. First 8 switches are used to input first two 4 bit data i.e A and B. The last 8 switches to assign the two 4 bit input data for multiplier that is the nibble1 and nibble2. The push buttons are used as selection line inputs.the carryin for the full adder is given through the one of the five push buttons provided on the board. The result or output after undergoing the operations on the given input are determined by the LED 'S present on the board. The 16 leds present on the board are assigned to the two sets in which the first set of 8 LEDs give the full adder output and the next set gives the 8 bit multiplier output and the carryoutis determined by the buzzer provided on the board which works in such a way that when there is carryout present in the output of the full adder the buzzer makes a sound and vice versa.

EXAMPLES TAKEN FOR IMPLEMENTATION:

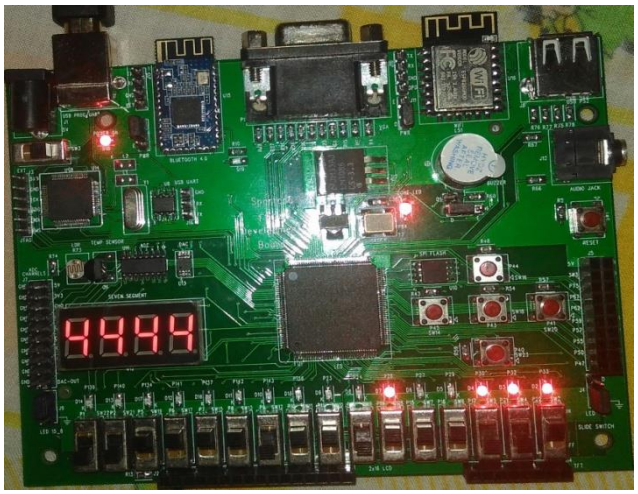
In case of Ripple carry adder the inputs A and B taken are as follows
Input A = 1001; Input B = 1111; Carry in = 0;
The ripple carry adder output after implementation are
Output = 1000 (decimal = 8(in 7 seg dispaly)) ;
Carryout = 1; (buzzer in response to high carryout)



In case of 4*4 the inputs Nibble1 and Nibble 2 taken are
Nibble1 = 1101 (13) ; Nibble2= 0011 (3) ;

The 4*4 multiplier output after implementation are
Output = 00100111 (13*3 = 39);(output on LEDs).

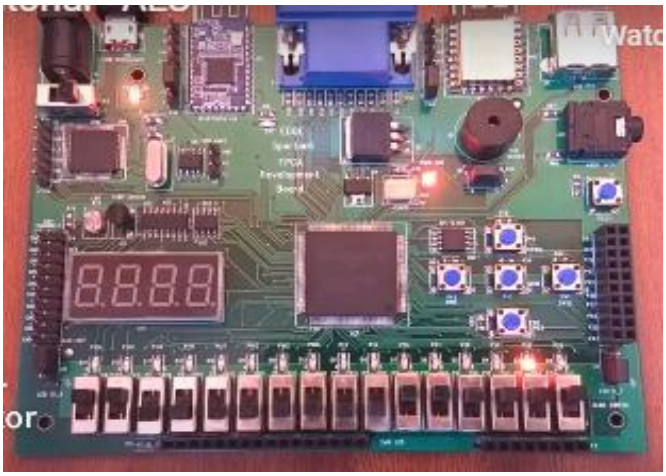
The constraints file in the Xilinx software determines the several number of pins on the board which are assigned to the respective inputs and outputs that can be decided by the user and the results can be determined through those pin assigned slide switches , push buttons and LEDs.



B. 8 BIT ARITHMETIC LOGIC UNIT

The 8 bit Arithmetic Logic Unit is implemented on the Spartan 6 FPGA board and the results are obtained . The inputs of two 8 bit data is given through the 16 slide switches. First 8 switches are used to input first 8 bit data i.e A and the last 8 switches to assign the 8 bit data B. The push buttons are used as selection line inputs . Different combinations of these

selection lines determine which operation has to be carried by the ALU .The proposed ALU has several operations that can be accessed by these combinations. The result or output after undergoing the operations on the given input are determined by the LED 'S present on the board. The 16 leds present on the board are assigned to the 16 bit alu output and through those the hardware result is determined.



XI. CONCLUSION

In this paper, various adder, multiplier and ALU circuits have been designed and implemented. The implementation has been done using Xilinx Spartan-6 XC6SLX9 field programmable gate array device because it confers special features that support arithmetic operation. This study helped to understand the complete flow of RTL design, starting from designing a top level RTL module for 8- bit ALU using hardware description language, VHDL. Verification of the designed RTL code using simulation techniques. Synthesis of certain efficient arithmetic circuits for binary and decimal arithmetic suitablefor quantum applications have been proposed in this paper.

This logic is used and the designs are realized using simple efficient reversible logic gates. As more features are integrated within battery powered electronic devices like cell phones, laptops nowadays, it drains the battery quickly.
So the need for new technology or circuit technique is necessitated to reduce power dissipation and area of arithmetic units used in processing elements. These fixed point adder and multiplier circuits are applicable in digital signal processing. The increasing demand for low-power very large scale integration (VLSI) can be addressed at different design levels, such as the architectural, circuit, layout, and the process technology level. The simulation results shows improvement in delay of output signal & decrease the distortion of the waveforms at the output stages. Due to the major advantages the proposed design can be suitable in DSP applications.

XII. REFERENCES

[1]. Lakshmi Narayanan G. and Venkataramani B., "Optimization Techniques for FPGA-Based Wave Pipelined DSP Blocks" IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol.13.no 7.
[2]. Shanthala S and S. Y. Kulkarni, "VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique".
[3]. K. V. Suresh Kumar, S. Shabbir Ali, E. Chitra, " Implementation of 32-bit high-valency ling adder in modified FIR filter using APC-OMS approach," International Journal of Electrical and Electronics Engineering ISSN 2231 – 5284, Volume-3, Issue-1, 2013
[4]. Chen K.H., Chen Y. M., and Chu Y. S., "A versatile multimedia functional unit design using the spurious power suppression technique".
[5] Stephen P. Boyd, Seung-Jean Kim, Dinesh D. Patil, Mark A. Horowitz, Digital Circuit Optimization via Geometric Programming, Operations Research.
[6] P. Prem Kumar, K. Duraiswamy, and A. Jose Anand, "An optimized device sizing of analog circuits using genetic algorithm,".
[7] Digilent, Inc.., Spartan 3E Starter Board,
Date Accessed June 2000
http://www.digilentinc.com
[8] Fraunhofer IIS, "From VHDL and Verilog to System".
[9] Xilinx Technologies, Xilinx Data Sheet for XC3S100E.