

IMPLEMENTATION OF AN ARITHMETIC LOGIC USING AREA EFFICIENT CARRY LOOK-AHEAD ADDER

Navneet Dubey and Shyam Akashe

¹ECE Department, ITM University, Gwalior (M.P), India

²Associate Professor, ECE Department, Gwalior (M.P), India

ABSTRACT

An arithmetic logic unit acts as the basic building blocks or cell of a central processing unit of a computer. And it is a digital circuit comprised of the basic electronics components, which is used to perform various function of arithmetic and logic and integral operations further the purpose of this work is to propose the design of an 8-bit ALU which supports 4-bit multiplication. Thus, the functionalities of the ALU in this study consist of following main functions like addition also subtraction, increment, decrement, AND, OR, NOT, XOR, NOR also two complement generation Multiplication. And the functions with the adder in the arithmetic logic unit are implemented using a Carry Look Ahead adder joined by a ripple carry approach. The design of the following multiplier is achieved using the Booths Algorithm therefore the proposed ALU can be designed by using verilog or VHDL and can also be designed on Cadence Virtuoso platform.

KEYWORDS

Arithmetic Logic Unit, Booth Multiplier, Carry Look- Ahead Adder, VLSI

1. INTRODUCTION

Arithmetic operation such as additions and subtraction and also in multiplication and division are widely used and play an important role in various digital systems such as digital signal processor (DSP) architecture and also for microprocessor further being used in microcontrollers and data process unit. Although adders are the logic circuits that had being designed to perform some of the high speed arithmetic as well as logical operations and are important components in digital system because of their extensive use in other basic operations such as subtraction and multiplication and good to be likely used for division. Nowadays in many of computer and other kinds of processor and adders are likely to be used hence not only in the arithmetic logic unit functions but also in other important parts of the processors where they are used to calculate addresses and table indices and same kind of operations performed thus the very basic arithmetic operations is the adding of two binary digits bits. A conventional ALU can be used to perform basic arithmetic and logic operations such as AND, OR, NOT, ADD, Subtract. The Arithmetic logic unit take two operands and also performs the desired operations between those units also the control signal is to be used to select the output from the operations that had been performed thus; the control unit is designed by using a multiplexer which selects the required operations.

All the operations are performed in one cycle but only the one that is required in the output is selected by the Multiplexer and an Arithmetic logic unit does not perform multiplication between two operands and also an extra circuitry is being required along with the Arithmetic logic unit which increase the chip area therefore, in this paper we propose the design of an ALU which supports multiplication. The Multiplier is designed using the Booth's algorithm is a multiplication algorithm that multiplies two numbers which are binary by using two's complement notation also in an conventional array multiplier requires a large number of devices. Thus the complexity of the circuit increase implementing the following algorithm is more efficient approach which performs signed multiplications and the circuit complexities is considerably less than any other type of multiplier circuit. Thus it contributes in building a faster logic circuit with a lower area requirement. The adder unit is constructed by using following adder as Carry Look-Ahead adder which is an conventional ripple-carry adder which indicates some considerable amount of delay. The adder takes care of the problem and is thus better equipped to perform faster operations. Addition is a fundamental arithmetic operation that is used in many VLSI design systems like DSP architecture, microprocessor, microcontroller and data process unit. This VLSI system requires fast addition which impacts The overall performance of digital system, these addition operations are done by using adders. Various adder structures can be used to execute addition such as serial and parallel structures and most of researches are done on the design of high-speed, low-area, or low-power adders.

2. PREVIOUS WORK

2.1 Adder

There have been various adder circuits used in previous design and other Arithmetic logic unit which is used to perform the addition operations also conventional ripple-carry adders have been used while designing logical circuits thus when two individual operands are applied as input to adder therefore it takes less time before giving out the valid output result. This happens because each full adder in the combination introduces a certain delay. If one full adder introduce a delay of time 't' then for an 'n' bit circuit the total delay after which Cout is obtained is '(n-1)t'. The delay incurred depends on the size of the operands. Thus when higher bit number is used the delay becomes highly unacceptable.

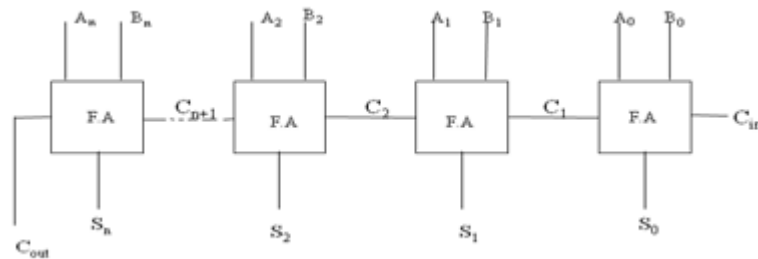


Figure 1: Ripple carry adder

2.2 Multiplier

Multiplication of two four-bit unsigned binary numbers X₃X₂X₁X₀ and Y₃Y₂Y₁Y₀ by using an array multiplier is shown in figure 2. Full adder block is the basic building block and total

number of full adder blocks required is $4 \times 3 = 12$. Output generated by the full adder block is (a) $SUM = X \oplus Y \oplus CI$. (b) Carry Out = $X.Y + Y.CI + CI.X$. The partial produce are realized using an AND Gates.

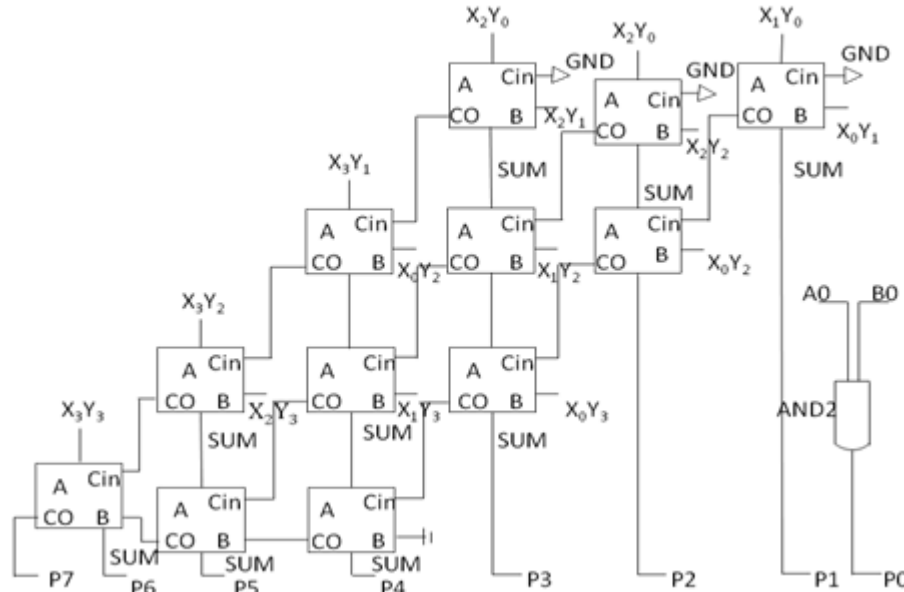


Figure 2: Array multiplier

3. ARITHMETIC LOGIC UNIT SUPPORTING BOOTH MULTIPLICATION

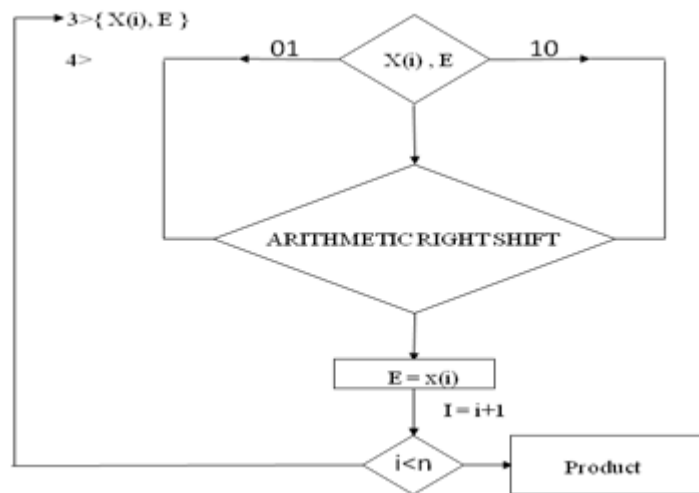
Functionalities that had been performed by the proposed Arithmetic Logic Unit are NOT, NAND, NOR, AND, OR, XOR, Addition, Increment also Two's complement generations, Subtraction, Decrement and Multiplication thus; the inverter, NAND and NOR blocks can be realized in the conventional manner. The inverter circuit consists of a positive channel metal oxide semiconductor (PMOS) and an NMOS transistor where the drain of the P type is connected to the supply voltage and the N type source is grounded. The inverter output is taken between the PMOS source and NMOS drain. An -bit inverter can be designed from the single bit inverter. The NAND and NOR circuits, both constitute a pull -up and pull-down in output generation of framework. The pull-up block in both the logic circuits is made up of PMOS transistor whereas NMOS transistors make up the pull-down block. The drain of the PMOS are connected each other which, in turn are to be connected to the supply voltage. The sources of both NMOS transistors are connected to the ground. The output is taken between the sources of the pull up block and also the drain of the pull down block therefore the pull-up network of NAND gate consists if two PMOS transistors connected in parallel with each other and the pull-down consists of two NMOS transistors connected in series with each other similar to the NOR circuit's pull up network has two transistors in series and the pull-down network has which are connected to two NMOS transistors in to the direction of parallel. An bits of NOT, NAND and NOR block can be erected from these three circuits. The AND and the OR block can be derived from the NAND and NOR blocks by utilizing the inverter which is used to prevents hardware complexity thus reducing the total Area there by making the ALU more efficient.

3.1 Adder

In digital adders the speed of addition is limited by the time required to propagate the carry signal towards the adder also in an elementary adder the generation of sum for each bit position takes places in a sequentially only after the preceding bit position has been summed and a carry is propagated in to the ascending next position, an carry look-ahead helps us in eliminating the delay caused by the propagation of the Carry signal in a binary adder thus; the delay caused in the addition operation is because of the carry signal.

3.2 Multiplier

Latency, area and the design complexity, throughput is some of the factors which would help us to choose and the booth multiplier algorithm accelerates multiplications. Let's take an example where we see the result of the algorithm for signed and unsigned form of multiplications. Consider two operands x and y which are the multiplier and the multiplicand respectively. The values of x and y are which are varied in three cases and the result of multiplications is observed using the algorithm.



- Case 1 $x = 3, y = 5$. Product = +15
- Case 2 $x = -3, y = -5$. Product = +15
- Case 3 $x = -3, y = 5$. Product = -15
- 1> Read x, y
- 2> $i = 0, z = 0, E = 0$

3.3 Subtraction

We construct the subtractor circuit by implementing the carry look-ahead adder and an inverter. For an n-bit number a complement $2N$ is what the two's complement means. The result obtained after we subtract the number by $2N$ also the two's complement exhibits the behaviour of the negative to be in use of the original number.

4. INCREMENT AND UNIT DECREMENT

The incrementer and decrement units are built using the adder and subtractor units. For increment one of the addends is given the input value logic '1'. For decrement unit the subtrahend of the subtractor is provided with logic '1' value.

Two's complement generation:

The Two's complement generator circuit is achieved by using an inverter and also an incremental thus; the input operands is passed by an inverter which inverts its digit also the output of the inverter is as input to the incremental. The incremented add as logic 1 to the inverted bits. Therefore, the output generated is the two's complement of the input bit.

The multiplexer is used to select the output. All operations are performed in a single cycle but only the required operation is selected by the multiplexer.

| Operation | Opcode | Operation | Opcode |
|-----------|--------|------------------|--------|
| AND | 0 | SUM | 110 |
| OR | 1 | Decrement | 111 |
| Subtract | 10 | Increment | 1000 |
| NAND | 11 | NOT | 1001 |
| NOR | 10 | Two's Complement | 1010 |
| XOR | 101 | Multiply | 1111 |

Table 1: Operations and their Op-code

5. CONCLUSION

The need for smaller-sized integrated circuits is evolving at a very fast rate. The VLSI field requires logic circuits with reduced complexity and circuit area and a higher that of processing rate and the proposed Arithmetic logic unit consist of an adder circuit which provide less delay than a conventional adder the Booth's multiplication algorithm reduces the hardware complexity and provide the output faster than other multipliers. The AND and OR units are realized using NAND and NOR blocks thereby minimizing the area required by the logic circuits. Less area and faster operation makes the ALU ideal for an efficient logic circuit. It can be coded using Verilog or VHDL. We can design it using the Cadence Virtuoso platform.

ACKNOWLEDGEMENT

The author would like to thank Institute of Technology and Management, Gwalior for providing the Tools and Technology for the work to be completed.

REFERENCES

- [1] Akbar Bemana, A new Simulation if a 16 bit Ripple-Carry Adder and a 16-bit skip Carry Adder, International Journal of Scientific and Technology Research, Volume 1, Issue 2, March 2012.
- [2] U. Sreenivasulu & T.Venkta Sridhar, Implementation of a four bit ALU using Low Power and Area efficient Carry select Adder, International Conference on Electronics and Communication Engineering, 20th May 2012, Bangalore.
- [3] Neil H.E Wete, CMOS VLSI Design: A circuit and system Perspective (Pearson Education India, 1 Sep. 2006).
- [4] Mousam Halder, Jagannath Samanta, Performance Analysis of High Speed Low Power Carry-Look Ahead Adder Using Different Logic Style, International Journal of Soft Computing and Engineering, Vol. 2, Issue 6 Jan. 2013.
- [5] Fazal Noorbasha, VLSI Implementation of 200 MHz, 8 bit, 90nm CMOS Arithmetic and Logical Unit Processor, Leonardo Electronic Journal of Practices and Technology 19 July-December 2011.
- [6] Michael Andrew Lai, Arithmetic Units of a High Performance Digital signal processor, University of California, Davis, 2002. [7] J. M. Rabey, Digital Integrated Circuits, A Design Perspective, Prentice-Hall, 1996.
- [7] N. H. E. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison Wesley, New York, NY, 1985.
- [8] Sung-Mo Kang and Yusuf Leblebici, CMOS Digital Integrated Circuits Analysis and Design, McGraw-Hill, 2002.
- [9] N. West and K. Eshraghian, Principles of CMOS VLSI Design: A System Perspective, 2nd edition, Addison Wesley, NY, pp. 513-536, 1992.
- [10] R.W. Doran, "Variants of an improved carry-lookahead adder," IEEE Trans. Computers, vol. 37, pp. 1110-1113, Sep. 1988
- [11] G. A. Ruiz, Compact four bit carry look-ahead CMOS adder in multi-output DCVS logic, Electronic Letters, vol. 32, no. 17, pp. 1556-1557, Aug. 1996.