ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Design of low Power Application specific ALU

SumitKumar[1], Jasbir Kaur[2]

P.G. Research Scholar, Dept. of Electronics and Communication, PEC University of Technology, Chandigarh, India[1]

Assistant Professor, Dept. of Electronics and Communication, PEC University of Technology, Chandigarh, India[2]

ABSTRACT: In this paper an 8 bit ALU IS designed and its various parameter are optimized in terms of Speed, Power Consumption and Chip Area are optimized. ALU is firstly designed in Verilog language and then synthesized on Xilinx ISE simulator and cadence RTL schematic respectively. The key parameters to design any digital system are their logic delay, power consumption and chip area. Since different applications require the different optimization parameter. So choice of each module is determined by the type of application. Different logic families are used in the design for various logic modules. Further detailed analysis of various multiplier, adder and logic circuit is carried out in this paper.

KEYWORDS: Selection Multiplier, Adder, Logic operation, bitwise operation,ALU

I. INTRODUCTION

The main module of the Central Processing unit isArithmetic Logic Unit. It performs various arithmetic operations like addition, subtraction multiplication and division of two numbers and logical and bitwise operations such asOR, AND, INVERT, XOR and other Boolean functions. Conventional adder circuit is one of the basic units in an ALU to perform the arithmetic operations. A description of quaternary adder is also given which is also used for arithmetic operation but it uses the number in quaternary form.Multiplier circuit is the most important module as the performance of the system is determined by the performance of multiplier so detailed analysis of different multiplier i.e Array multiplier, Wallace multiplier, radix-2 multiplier, radix-4 multiplier and conventional multiplier is given in this paper. According to the system specification desired multiplier is selected from ALU. Some constraints of system specification are chip area, speed and the power consumption etc. for example if we have to select a multiplier based on area and power consumption then array multiplier is best suited multiplier from ALU. Some logical operator like logical AND or logical OR is also provided in the ALU. Similarly if we have to use bitwise operator, we can easily select from ALU. All these internal circuits are firstly designed in Verilog and then synthesised in Xilinx ISE simulator and cadence RTL schematic. Detailed analysis of internal element of ALU is given below.

II. SELECTION IN ADDERS

Adder is to add two numbers i.e X and Y. in this designed ALU we select either the conventional adder or the QFA adder.

1. *Conventional adder:*

The conventional adder circuit is simply a ripple carry adder implemented using cascaded full adders. Each full adder adds three single bits giving one bit sum and a carry bit as outputs in binary logic.

2. *QFA Adder:*

This is a multivalued logic adder having input in multivalued format and is used for addition. For high number of bits of input, this multiplier is used as it is having optimized parameter like less power consumption, less delay etc. To add two numbers by QFA adder in ALU, we simply provide the value to select line N equal to two shown in the part of Verilog code. RTL schematic of QFA adder is shown in figure 1.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

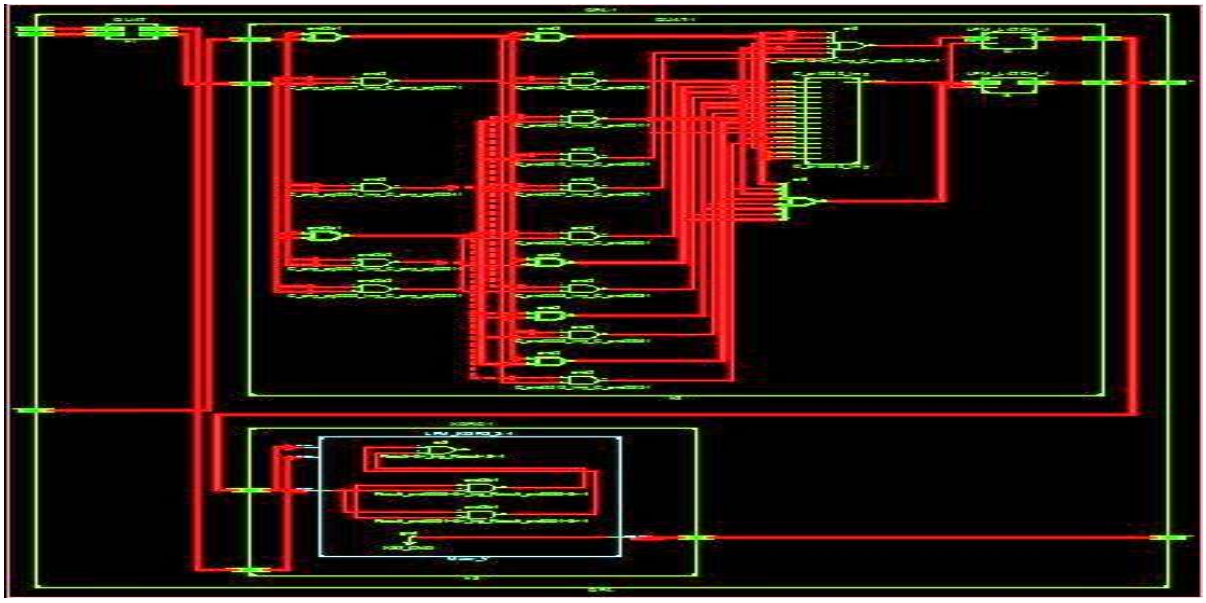

Figure 1 RTL Schematic of QFA ADDER

III. SELECTION IN MULTIPLIERS

1. *Array Multiplier:*

Array multiplier was used initially to multiply two numbers. To multiply two numbers using Array multiplier we firstly make bitwise anding to generate partial products of two number, partial products obtained from bitwise anding is either the multiplicand shifted by a particular amount when multiplier bit is one or 0 when multiplier bit is 0. These partial products are then added with the help of Ripple adders. Array multiplier in this paper is designed with the help of Verilog programming language and then output result is verified by using cadence RTL schematic. Array multiplier is a part of ALU. To multiply two numbers by array multiplier in ALU, we simply select the array multiplier by providing select line a value equal to 3. Figure 2 shows a 4*4 unsigned array multiplier synthesized in cadence RTL schematic using Verilog language.

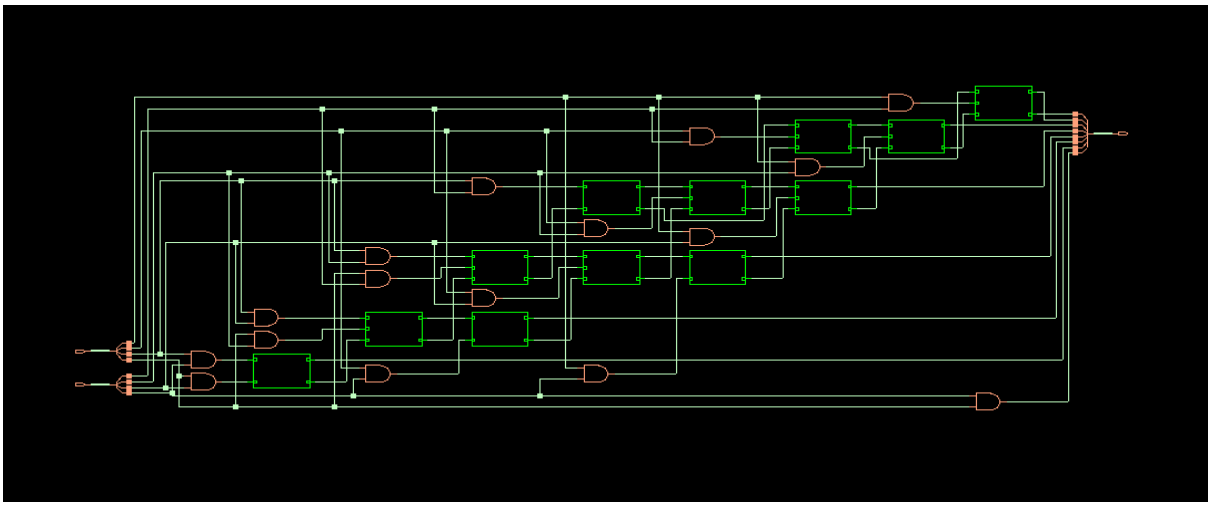

Figure 2 RTL schematic of 4*4 Array multiplier

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

2. *Radix-2 Multiplier :*

This multiplier was initially developed by O. L. Macsorley in 1961 for speeding up multiplication in Early computers by reducing number of partial product by a factor of two. This multiplier uses a booth table from which a binary number is recoded according to booth algorithm. This multiplier was initially designed for high bit parallel multiplier as for low bit for multiplier array multiplier was also efficient. The drawbacks of radix-2 multipliers is that number of add subtract operations and the number of shift operations becomes variable and becomes inconvenient in designing parallel multipliers and algorithm becomes inefficient when there are isolated 1's. To multiply two number by radix-2 multiplier in ALU, we simply provide the value to select line n equal to four. Technology schematic of radix-4 booth multiplier is shown in figure 3.

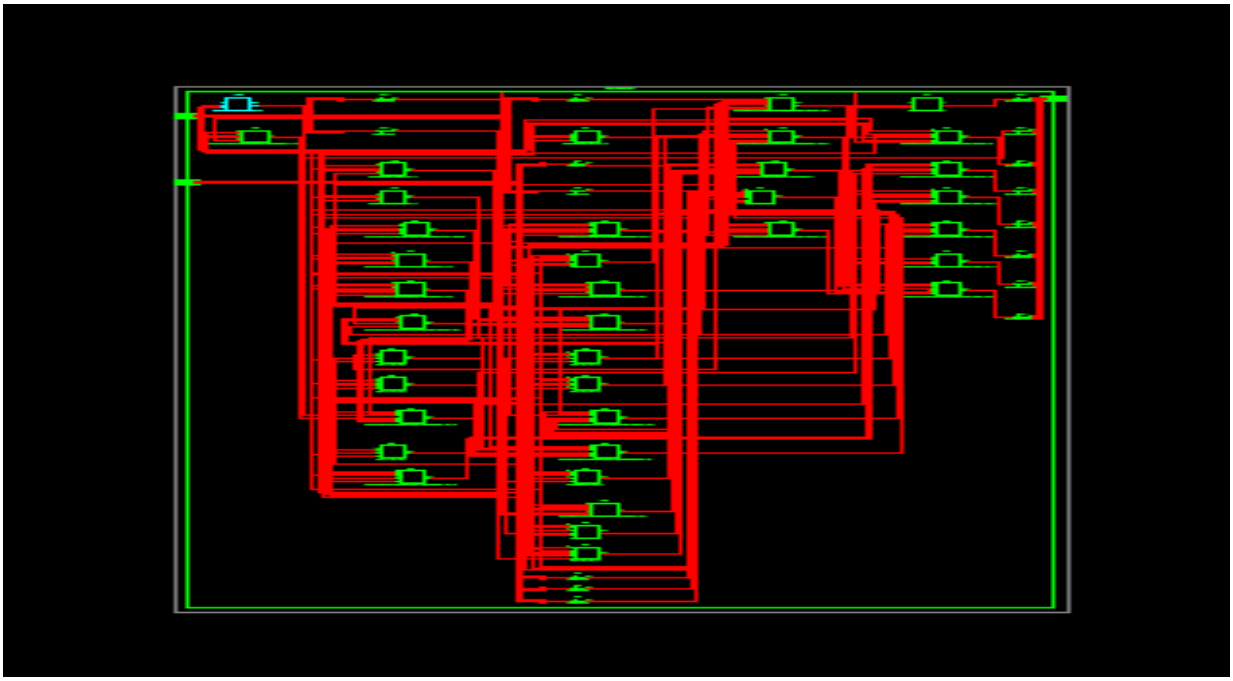

Figure 3 Technology Schematic of radix2 multiplier

3. *Radix-4 Multipliers:*

Problems of Radix-2 booth multiplier can be overcome by radix-4 multiplier. Radix-4 booth multipliers recode three bits at a time which results into decrease of partial products by a factor of three. Recoding method of radix-4 multiplier is similar to radix-2 booth multiplier i.e we have to recode multiplier according to radix-4 booth table and then implement radix-4 algorithm. The radix-4 algorithm is designed in Verilog language and then synthesized in cadence RTL schematic. An example is given to multiply two number by radix-4 multiplier. Let the two numbers to be multiplied are X = 18 and Y = -15. To multiply two number by radix-4 multiplier in ALU, we simply provide the value of select line N equal to five.

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

TABLE 1. Radix-4 Booth Multiplier Example

								0	0	0	1	0	0	1	0	18
								1	1	1	1	0	0	0	1	-15
						0	0	0	0	0	1	0	0	1	0	PP1
				0	0	0	0	0	0	0	1	0	0	1	0	PP2
		1	1	1	1	1	0	1	1	1	1	0	0	1	0	PP3
1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	PP4
1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	-270

4. *Wallace multiplier:*

Various algorithm and techniques have been designed in past with objective of improving the speed of the parallel multiplier. Wallace Tree multiplier is a very good technique to improve the speed of parallel multiplier. In Wallace tree architecture, instead of adding the partial product in ripple fashion all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. At the last stage of Wallace multiplier a fast adder is used at the end to produce the final result. Since the addition of partial products now becomes 0 (IogN) the speed of Wallace multiplier greatly improves. If we design an ALU considering only multiplier then technology schematic of the ALU is shown in the figure 4.

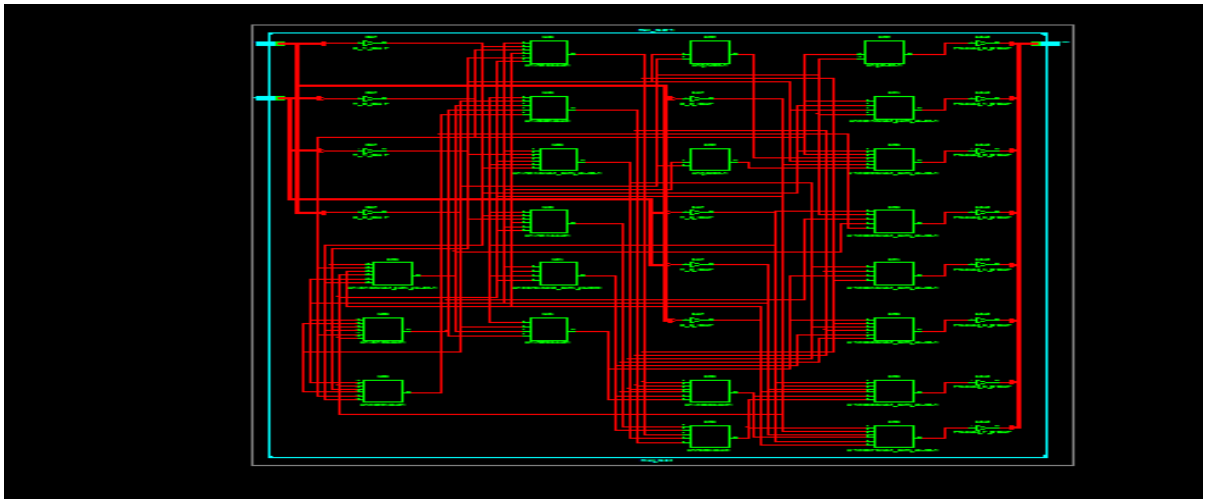

Figure 4 Technology Schematic of selection of multiplier in ALU

IV. LOGICAL OPERATION

Logical operation includes logical operatorsi.e logical-AND, logical-OR and logical-NOT. Logical operators always evaluate to l-bit value. If resultant expression is equal to zero, then output bit is equivalent to a logical o (false condition). If the resultant expression is not equal to zero, then the resultant value is equivalent to a logical 1 (true condition). If any operand bit is X or Z, then resultant expression is equivalent to X (ambiguous condition) and is normally treated by simulators as a false condition. Example for different logical operation is given below:

Suppose A = 2; B = 0;

A && B (Logical-and) Expression evaluates to 0. Equivalent to (logical-1 && logical-0)

A || B (Logical-or) Expression evaluates to 1. Equivalent to (logical-l || logical-0)

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

! A (Logical-not) Expression evaluates to 0.
! B (Logical-not) Expression evaluates to 1.

V. BITWISE OPERATION

In bitwise operation, bitwise operators perform a bit-by-bit operation on two operands. They take each bit in one operand and perform the operation with the corresponding bit in the other operand and if one of the operand is shorter than the other, then that operand will be bit extended with zeros to match the length of the other operand. Example for different Bitwise operation is given below:

Let X=4'b1010, Y=4'b1101
~X // Bitwise Negation. Output Result is 4'b0101.
X & Y // Bitwise and. Output Result is 4'b1000.
X | Y // Bitwise or. Output Result is 4'b1111.
X^Y // Bitwise xor. Output Result is 4'b0111.

VI. RESULTS

From the individual simulation of all the circuit on RTL schematic we find out that for low bit of inputs, conventional adder is an optimized circuit for addition while for high bit of numbers we use QFA adder as it is having optimized parameter for higher bit numbers. Table 2 represents different parameter of all the multiplier discussed in this paper. From these parameter we can easily select required multiplier according to application by providing specific value to select line.

Table 2 Comparison of different 4*4 multiplier

S.no	Multiplier	Delay(ns)	AREA	Power Consumption(uW)	Area-Delay Product(AD)	Area-Power Product(AP)
1	Array Multiplier	1.791	78	1.369	139.698	106.782
2	Radix-2 booth multiplier	1.744	158	2.555	275.552	402.9
2	Radix-4 booth multiplier	.779	96	1.18	74.78	113.28
4	Wallace Multiplier	1.517	87	1.573	131.97	136.851
5	Conventional Multiplier	0.992	85	1.453	84.32	122.51

A part of the Verilog program of ALU showing how to select one of the parameter using select line N is given below:

ISSN(Online): 2320 - 9801
ISSN (Print) : 2320 - 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

```
case(N)

1: conventional_adderc1(A,B,CIN,SUM,COUT);
2: QFA q1(A,B,CIN,SUM,COUT );
3: arraymul a1(X,Y,Product );
4: radix2 r1(X,Y,Product);
5: radix4  r2(X,Y,Product);
6: wallace w1(X,Y,Product );
7: simplemul s1(X,Y,Product);
8: LOGICOR  L1 (X,Y,LOGICOP);
9: LOGICAND   L4  (X,Y,LOGICOP);
10: BITWISEOR  B1  (X,Y,LOGICOP);
11: BITWISEAND     B2   (X,Y,LOGICOP);
12: BITWISEXOR   L3   (X,Y,LOGICOP);

endcase
```

VII. CONCLUSION

In this Paper, An 8-bit arithmetic logic unit with 16 functions is designed in Verilog and the design is validated in Xilinx ISE simulator and cadence RTL schematic respectively. The 8-bit ALU design consists of different adder configuration to add, multiplier configuration to multiply and different operators. Different logic families are preferred for different application. For less number of bit of input, we select conventional adder in ALU as it provide optimum result but for higher number of input we select QFA as default adder, for multiplication, we select the multiplier depending on application. If application requires less area then we select array multiplier as a default multiplier. But if we require optimization in all parameter, then we select radix-4 multiplier as a default multiplier. Different operator like logical operator, bitwise operator is also provided in ALU. To choose any operator we have to provide corresponding value to select line. The selection of adder and multiplier algorithm helps the designer to select according to the area, power and delay parameter.

REFERENCES

1. R. D. Blanton and J. P. Hayes,'Design of a Fast, Easily Testable ALU', IEEE Conference Publications, pp. 9-16, 1996.
2. Yu Zhou and HuiGuo,' Application Specific Low Power ALU Design', IEEE Conference Publications, Vol. 1, pp. 214 - 220, Aug. 2008.
3. Na Gong, Jinhui Wang and RamalingamSridhar,'Application-driven power efficient ALU design methodology for modern microprocessors', IEEE Conference Publications, pp.184 - 188, 2013.
4. L. Gopal, N. SyahiraMohdMahayadin, A.K Chowdhury, A.A Gopalai and A. Kumar Singh,'Design and synthesis of reversible arithmetic and Logic Unit (ALU)', IEEE Conference Publications, pp. 289 – 293, September 2014.
5. A. G. Rao and A. K. D. Dwivedi,'Design of Multifunctional DR Gate and Its Application in ALU Design', International Conference on Information Technology, pp.339 - 344, December 2014.
6. N. Ravindran and R. M. Lourd,'An optimum VLSI design of a 16-BIT ALU',International Conference on Information and Communication Technology Research, pp. 52-55, May 2015.
7. N.Agrawal, M. Agarwal, S.Singh, A.Kumar and B.Pandey,'Different I/O standard based Wi-Fi enable 32-bit ALU design on 90nm FPGA. Communication', Control and Intelligent Systems, IEEE Conference Publications, pp. 382 - 389, November 2015.
8. J.Kaur, P.Sapra,'Multi Valued Logic Based Design of Quaternary Adders in VHDL', International Journal of Innovative Research in Science, Engineering and Technology, pp. 1443-1448,Feburaray 2016.