Design of Area Efficient 32 Bit Arithmetic and Logic Unit (ALU) for DSP Processors

K. V. Ganesh, V. Malleswara Rao, Sreenivasa Rao Ijjada

ABSTRACT---Almost every electronic gadget contains the Digital signal processor (DSP) unit for the purpose of computations, whose role couldn't be specified with smaller words. Gadget's performance, efficiency and the importance could be measured with how best the specifications of the processors are. Arithmetic and Logical Unit (ALU) is the key circuit for any DSP processors, where large data computations can be performed. Hence, the ALUs design should be include high performance and large data handling capacity. An ALU is a digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers. The conventional ALU designs, design complexity rate proportionally increases with the performance demand. In this paper, an attempt has been given to design a low complex ALU with improved performance. Sub circuits designs comprise with new approaches to make the simple designs for higher performance of ALU. A 32 bit ALU design procedure has been demonstrated in this paper. For design, 90 nm CMOS technology and CADENCE virtuoso tools used.

Keywords: Arithmetic and logical unit, adders, multipliers, Digital signal processing, multiplexers, comparators.

I. INTRODUCTION

In the present scenario, VLSI plays a major role. Currently existing microprocessors (μp) operates at clock frequency of more than 5GHz and fabricating millions of transistors on a chip. Scaling of Transistor has made the IC faster and also enhanced the mode to be denser. Hence, they are more preferable for designing many DSP applications [4]. High operating frequency consists of more number of transistor which results more power consumption [1], for example Intel's Pentium has 42 million transistors on 0.18 μm CMOS [2][3]. ALU is the key component in the processors and processors have ALU's in execution different units. High speed digital adder architectures proposed new technology for carry propagation designs[5]. The PTL gains more speed advantage over CMOS due to high logic functionality for fabricating a high-speed adder, CCS resolves the problem of pass transistors connected in series for performing all these techniques 30% of the delay can be reduced when compared with ordinary CMOS ALUs [5]. A slice of 4 bit ALU has been introduced to design 32-bit rapid single-flux-quantum μp . ALUs are one of the most building blocks in processors which perform all arithmetic as well as logical operations[6] [7]. Traditional adder structures like Kogge-

Stone and Ladner-Fischer will perform carry propagation operations[8]. Multiplication is the most commonly used arithmetic operation in DSP processors, and the overall performance strongly depends on the multiplication and division [9]. Division operation is much similar to the multiplication, instead of addition in multiplier, performs subtraction in division and instead of left shift, performs right shift in division. In ALU division frequently right shifted divisor are from subtracted or added to the dividend or the resulting partial product. To carry out the partial product remainder will determines the quotient bit as well as whether the shifted divisor is to be added or subtracted on the next cycle [10].

In this paper, proposed a 32 bit ALU design for less power and more speed, which includes all the arithmetic and logic operations for 32 bit operations. In this work, initial the design starts with 1-bit ALU design, and is extended to 8bit, 16bit and 32bit ALU designs.

II. EXISTING WORK

ALU is a basic building block of many types of computing circuits, including the central processing unit (CPU), ALU performs bitwise arithmetic and logical operations on integer binaries, in contrast to a floating-point unit (FPU).

i). 1 Bit ALU design:

1-bit ALU has four levels of operations. In the first level all arithmetic operations are to be performed. 1 bit ALU as shown in fig.1. The circuit shows a 4X1 MUX and two select lines S0 and S1. It required to alter one of the input of the full adder (FA) depending on the logic needed [4]. A 4x1 MUX can be designed by three 2X1 MUXs. FA's one input is fixed and the other input is from the MUX depends on the S0 and S1. This circuit performs five basic arithmetic operations as listed in the table 1. If S0 = 0 and S1 = 0, then B is transferred to FA and addition operation can be performed, if both are high, decrement operation can be performed. Similarly if S1 = 0 and S0 = 1, then the B' is transferred to FA and subtraction can be performs. If reverse, once complement of A can be results.

Revised Manuscript Received on 14 August, 2019.

K.V.Ganesh, Senior Member, IEEE, Assistant professor, Dept of ECE, VITAM college of Engineering, Andhra Pradesh, India (Email: kona.venkat13@gmail.com)

Dr.V.MalleswaraRao, Senior Member, IEEE, Professor, Dept of ECE, GITAM University, Visakhapatnam, Andhra Pradesh, India. (Email: mraoveera@yahoo.com)

Sreenivasa Rao Ijjada, Senior Member, IEEE, Asst. Professor, Dept of ECE, GIT, GITAM University, Visakhapatnam, Andhra Pradesh, India. (Email: dr.ijjada2019@gmail.com)

Fig 1: 1-bit ALU diagram

Table 1: Functional table of 1 bit arithmetic circuit

S1	S0	C	Output of multiplexer	Operation
0	0	X	B	Addition
0	1	1	B'	Subtraction
1	0	0	0	Transfer A
1	0	1	0	Increment A
1	1	0	1	Decrement A

Second level need to operate 4 bitwise logical operations such as AND, OR, EX-OR, EX-NOR. The design is shown in the fig.2. The selection inputs S0 and S1 decide the operation. This circuit performs four logical operations as listed in the table 2. If both the selection inputs are logic 0s, then $Y=A.B$, if both are high, $Y=A \text{ xnor } B$, if $S1=0$ and $S0=1$, then $Y=A \text{ xor } B$, and if $S0=1$ and $S1=0$, then $Y=A+B$.

Fig 2: 1-bit logic unit

Table 2: Functional table of 1 bit ALU

S0	S1	Y(Operation)
0	0	A.B
0	1	A+B
1	0	A xor B
1	1	A xnor B

In third level, uses another 2X1 MUX with selection input S2 in order to get all nine operations. Here, the FA sum and output (Y) are the inputs and intermediate output is the output as shown in the fig.3. If selection input $S2=0$, then arithmetic operations can be performed, otherwise logical operations can be resulted.

In the final stage, one more 2X1 MUX is connected with EA as the selection line, intermediate output and MUL&DIV are the input signals as shown in the fig.4. Here, the final output is either stage 3 output or the MUL& DIV output. This stage performs all the operations.

Fig 3: 1bit ALU without multiplier & divider

Fig 4: one bit ALU

III. PROPOSED WORK

In most of the ALU designs, Due to the complexity of the Division and multiplication circuits, usually separate circuits were designed to do multiplication and division operations separately, which is not area efficient and not power efficient. Here, we proposed a single circuit which can make possible of doing both multiplication and division operations. This is the novelty of this work.

i) Combined circuit for both multiplier and divider:

1 bit multiplier/divider circuit is shown in the fig.5, an AND gate, an XOR gate, two 2X1 MUXs and 1 full adder/subtraction circuits are the fundamental building blocks for the circuit. A 2X1 MUX has been designed for this circuit using pass transistor logic to reduce the transistor count and since there is less number of series transistors, there is no voltage degradation. Whenever the select input is low PMOS will be turned ON and the input1 will be appeared at the output, whenever the selection input is high then nMOS transistor will be turned on and pMOS will be turned off and input 2 will be appeared at the output. Fig.6 shows the results

Fig 5: 1-bit combined multiplier and divider

Fig 6: 1 bit multiplication and division results

ii) Proposed 1-bit ALU design:

With inclusion of the above multiplier/divider circuit a 1-bit ALU has been proposed with less hardware requirement. The circuit shown in fig.7 can perform the following operations, if $EA=0, N=0, S2=0, S1=0, S0=0, CIN=X$, then ADDITION operation can be performed, if $EA=0, N=0, S2=0, S1=0, S0=1, CIN=1$, SUBTRACTION can be performed. If $EA=0, N=0, S2=0, S1=1, S0=0, CIN=0$ TRANSFER A can be performed, if $EA=0, N=0, S2=0, S1=1, S0=0, CIN=1$ INCREMENT A will be performed, if $EA=0, N=0, S2=0, S1=1, S0=1, CIN=1$ DECREMENT A will perform, if $EA=1, N=0, S2=0, S1=0, S0=0, CIN=0$ MULTIPLICATION can perform, if $EA=1, N=0, S2=0, S1=0, S0=0, CIN=1$ DIVISION can be perform. If $EA=1, N=0, S2=1, S1=0, S0=0, CIN=0$ EX-OR operation can be performed, if $EA=1, N=0, S2=1, S1=0, S0=0, CIN=1$ EX-NOR operation can be performed. If $EA=0, N=0, S2=1, S1=0, S0=0, CIN=0$ AND operation can be performed, if $EA=0, N=0, S2=1, S1=0, S0=0, CIN=1$ OR

operation would performed, if $EA=0, N=1, S2=1, S1=0, S0=0, CIN=0$ NAND operation and if $EA=0, N=1, S2=1, S1=0, S0=0, CIN=1$ NOR operation can be performed as shown in the table 3.

Table3: Function table of 1-bit alternative arithmetic and logic circuit

EA	N	S2	S1	S0	C-IN	OPERATION
0	0	0	0	0	X	ADDITION
0	0	0	0	1	1	SUBTRACTION
0	0	0	1	0	0	TRANSFER A
0	0	0	1	0	1	INCREMENT A
0	0	0	1	1	1	DECREMENT A
1	0	0	0	0	0	MULTIPLICATION
1	0	0	0	0	1	DIVISION
1	0	1	0	0	0	EX-OR
1	0	1	0	0	1	EX-NOR
0	0	1	0	0	0	AND
0	0	1	0	0	1	OR
0	1	1	0	0	0	NAND
0	1	1	0	0	1	NOR

Fig 7: Proposed 1-bit ALU

Fig 8: 1-bit ALU schematic

Transistor count, area of the design and power utilization of the proposed design has been compared with existing design and the results are presented in the table 4. From the table it can be observed that, proposed design uses 38 transistors for 1 bit ALU, the design area is $1190.95\mu\text{m}^2$ and

the power consumption is $21.369\mu\text{w}$. There are 65 transistors less compared to the previous design, 68% area and 27% power efficient.

Table 4: Comparison report of 1 bit ALU circuits

Model	Power(μ w)	Area(μ m ²)	No of transistors
1-bit ALU[8]	29.37	3731.616	103
Proposed 1-bit ALU	21.369	1190.95	38

iii) 8/32 bit ALU designs:

With the above proposed 1 bit ALU, any number of bits ALU can be designed. Here designed 8 and 32 bit ALUs , 8 bit ALU takes two 8-bit binary numbers as input and produces 8-bit output in case of the basic operations, and produces 16-bit output in case of multiplier and 8-bit quotient and 8-bit remainder in case of division operation. The design of 8-bit basic ALU can be done by simply connecting eight 1-bit ALU's in a series manner by giving output carry of previous ALU is given as input carry to the present ALU.

Fig 9: 32-bit ALU with cascading of four 8 bit ALUs

Fig 10: 32-bit ALU Schematic

In this whenever performing an arithmetic operation, the carry output generated by the ALU is taken as carry output. But in case of logical operations, the carry generated by ALU is neglected. In case of designing 8-bit multiplier and

divider we require 64 one-bit combined multiplier and divider blocks in order to obtain the desired results. In order to differentiate multiplication and division use input carry bit. If the inputs carry bit is logic high then the operation performed is DIVISION, else the operation performed is MULTIPLICATION. The design of 8 bit ALU can be simply extended to 32-bit by cascading of four 8 bit ALUs as shown in the fig.9, and its schematic is in the fig.10. In 32 bit ALU the inputs are two 32 bit binary numbers and its operation is exactly as explained above for the 8 bit operation expect the length of the inputs.

IV. SIMULATION RESULTS

In ALU design, output carry should be taken care, since all logical operations depend on the input carry bit. This may results false outputs if applied output carry directly to the next ALU in the cascading. To overcome the issue, use a multiplexer to transmit output carry bit. In ARITHMETIC operations, else it will transmit input carry if the performing operation is LOGICAL operation [5]. The simulation results are shown below.

i) Arithmetic Operations 32-Bit ALU:

Consider two 32 binary numbers A31-A0 and B31-B0 as inputs as shown in fig.11 to fig.16, and the results are shown in the fig.17 and fig.18.

Fig 11: Selection inputs and input bits (A31-A25)

Fig 12: A24-A12 input bits

Fig 13: A11-A0 input bits

Fig 14: B31-B19 input bits

Fig 15: B18-B6 Input bits

Fig 16: B5-B0 input bits and out31-out25 output bits ()

Fig 17: out25-out13 output bits

Fig 18: out12-out0 output bits

ii) Logical Operations:

logical outputs are shown in the fig 19 and fig. 20 as shown below.

Fig 19: logical output bits out24-out12

Fig 20: logical output bits out11-out0

iii) Multiplication and division operation:

Consider two 32 binary numbers A31-A0 and B31-B0 as inputs as shown in fig.21 to fig.28, and the results are shown in the fig. 29 to fig.33.

Fig 21: selection input (c-in) and input bits (A62-A51)

Fig 22: input bits (A50-A38)

Fig 23: input bits (A37-A25)

Fig 24: input bits (A24-A12)

Fig 25: input bits (A11-A0)

Fig 26: containing input bits (B1-B19)

Fig 27: input bits (B18-B6)

Fig 28: input bits (B5-B0) and output (m63-m57)

Fig 29: output bits (mr6-mr43)

Fig 30: output bits (mq42-mq30)

Fig 31: output bits (mq29-mq17)

Fig 32: output bits (mq16-mq4)

Fig 33: out bits (mq0-mq3)

From the table 5, it is observed that the proposed ALU design is the best compared to the other designs shown in the table.

Table 5: Comparison table of different adder circuits

Design	Supply voltage (V)	Technology	Power Consumption	Operation Frequency	Delay
This work	1.0	0.09µm	13.672mW 8.75mW	5GHz	147.9pS
1[2005]	1.5	0.18 µm	201mW	1.7GHz	455pS
2[2001]	1.8	0.18 µm	185mW	4 GHz	317PS
3[2005]	1.2	0.09µm	351m	2.5GHz	171pS
4[2016]	3.2	0.1 µm	283mW	1.9GHz	525 nS
5[2018]	1.2	0.09µm	213mW	2 GHz	431ps
6[1993]	2.5	0.25 µm	307mW	2GHz	1.9ns
7[2006]	1.8	0.18 µm	2.18mW	231M Hz	769nS
8[2005]	2.1	0.09µm	150mW	9 GHz	159ns
9[2002]	1.37	0.13 µm	172mW	6 GHz	210ns

V CONCLUSIONS

The proposed ALU is compared with different ALUs in terms of power, propagation delay and frequency of operation. The proposed ALU has less power with 13.672mw and low propagation delay with 147.9ps the overall performance and design can be made using cadence 90nm technology operating at 5GHZ of frequency. All the designs are performed using Cadence Virtuoso EDA tools.

REFERENCES

- Guang-Ming Tang , Pei-Yao Qu, Xiao-Chun Ye, and Dong-Rui Fan, "Logic Design of a 16-bit Bit-Slice Arithmetic Logic Unit for 32-/64-bit RSFQ Microprocessors" IEEE transactions on Applied Superconductivity, Vol. 28(4),2018.
- Makoto Suzuki, Norio Ohkubo, ToshinobuShinbo, Toshiaki Yamanaka, Akihiro Shimizu, Katsuro Sasaki, and YoshinobuNakagome, "A 1.5-ns 32-b CMOS ALU in Double Pass-Transistor Logic" IEEE Journal of Solid-State Circuits, vol. 28(11), 1993.
- Guang-Ming Tang, Kensuke Takata, Masamitsu Tanaka, Akira Fujimaki, Kazuyoshi Takagi, and Naofumi Takagi, "4-bit Bit-Slice Arithmetic Logic Unit for32-bit RSFQ Microprocessors" IEEE transactions on Applied Superconductivity, Vol. 26(1), 2016.
- Sanu K. Mathew, Mark A. Anders, Brad Bloechel, Trang Nguyen, Ram K. Krishnamurthy, and

ShekharBorkar, "A 4-GHz 300-mW 64-bit Integer Execution ALU With Dual Supply Voltages in 90-nm CMOS" IEEE Journal of Solid-State Circuits, Vol. 40(1), 2005.

- BhaskarChatterjee and ManojSachdev "Design of a 1.7-GHz Low-Power Delay-Testable 32-ALU in 180-nm CMOS Technology" IEEE transactions on VLSI Systems, Vol. 13(11), 2005.
- Glenn Hinton, Michael Upto avid J. Sager, Darrell Boggs, Douglas M. Carmean, Patrice Roussel, Terry I. Chappell, Thomas D. Fletcher, Mark S. Milshtein, Milo Sprague, SamieSamaan, and Robert Murray "A 0.18-um CMOS IA-32 Processor With a 4-GHz Integer Execution Unit" IEEE Journal of Solid-State Circuits, Vol. 36(11), 2001.
- Steven K. Hsu, AmitAganval, Kaushik Roy, Ram K. Krishnamurthy, ShekharBorkar"An 8.3GHz Dual Supply/Threshold Optimized 32b Integer ALU-Register File Loop in 90nm CMOS" ISLPED'OJ, August 8-10, 2005, San Diego, California, USA.Copy right 2005.
- Makoto Suzuki, Norio Ohkubo, ToshinobuShinbo, ToshiakiYamanaka, Akihiro Shimizu, Katsuro Sasaki, and YoshinobuNakagome, "A 1.5-ns 32-b CMOS ALU in Double Pass-Transistor Logic" IEEE Journal of Solid-State Circuits, Vol. 28(11), 1993.
- Hyejung Kim, Byeong-Gyu Nam, Ju-HoSohn, Jeong-Ho Woo, and Hoi-Jun Yoo, "A 231-MHz, 2.18-mW 32-bit Logarithmic Arithmetic Unit for Fixed-Point 3-D Graphics System" IEEE Journal of Solid-State Circuits, Vol. 41(11), 2006.
- SriramVangal, MarkA. Anders,NitinBorkar, ErikSeligman,VenkateshVsanthaErraguntla, Howard Wilson,AmareshPangal,VenkatVeeramachaneni,James W. Tschanz, Yibin Ye, Dinesh Somasekhar, Bradley A. Bloechel, Gregory E. Dermer, Ram K. Krishnamurthy, K. Soumyanath, Sanu Mathew, Siva G. Narendra,Mircea R. Stan,Scott Thompson, Vivek De, and ShekharBorkar "5-GHz 32-bit Integer Execution Core in 130-nmDual-VT CMOS" IEEE Journal of Solid-State Circuits, Vol. 37(11), 2002.
- David H. K. Hoe, Chris Martinez and Sri Jyothsna Vundavalli Department of Electrical Engineering The University of Texas, Tyler "Design and Characterization of Parallel Prefix Adders using FPGAs", IEEE 43rd Southeastern Symposium on System Theory,2011