Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Single Cycle MIPS Design using High Performance ALU

Hemanth, Dr. Dola Sanjay S

Department of ECE, Visvesvaraya Technological University, Belgaum, Karnataka, India
Email: hemanthreddy32@gmail.com

Arithmetic and Logic Unit is the main processing unit of the MIPS processor. In ALU, the computational complexity is majorly due to adders and Multipliers used for the design. There exist many adder and multiplier designs in literature but the most adoptable ones are discussed in this paper. The various combinations of adders and multipliers are considered for evaluation. The parameters used for evaluation are Area, Delay, Power Dissipation and Power Delay Product. The designs are modeled in Verilog HDL and are functionally verify by using Xilinx ISE 14.5 and ISIM simulator. Among all the designs, the modified linear carry select adder along with vedic multiplier proves to be best for practical implementation of ALU in terms of power delay product. Also, the modified linear carry select adder along with shift add multiplier proves to be best for practical implementation of ALU in terms of area.

Keywords: ALU, Carry Select Adder, Carry Save Multiplier, Power Delay Product, Shift-add Multiplier, Vedic Multiplier.

1. Introduction

Processors are designed with an aim to suit at-speed operation for practical applications. The recent advances in technology initiate the compact and high speed designs especially for MIPS processor as shown in figure 1. MIPS stands for Microprocessor without Interlocked Pipelining Stages. The MIPS is faster due to fast access time of access a register as compared to a memory location, it is much faster to perform operations in on chip registers rather than in memory by using the load/store architecture. The computational complexity in any design is characterized by the adders and multipliers used. So if the adders and multipliers are chosen appropriate, then the ALU designed will be performing faster to suit the rated speed.

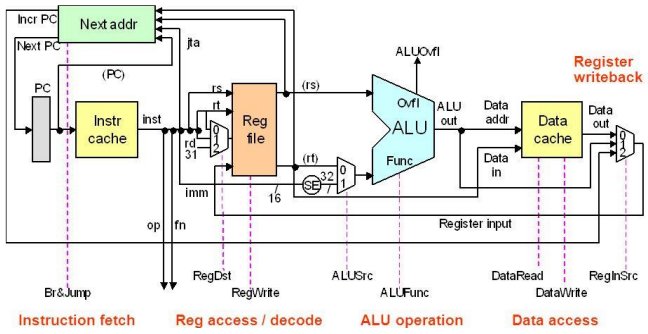


Fig.1: A Single Cycle Micro MIPS Data Path

Many researchers focused on development of various adders and multipliers with a focus on reducing the delay or making the design compact. In this paper, the focus is more on carry select adders and their modifications for adder design. Also multipliers like shift-add multiplier, carry save multiplier, vedic multiplier, etc are used for evaluation.

For the design of ALU, 64-bit data is considered and the basic architecture of ALU is as shown in figure 2. It comprises of Arithmetic unit, logic unit and multiplexer. The select line of multiplexer selects arithmetic unit if s[3] = 0 else selects logic unit. The lower order select lines i.e., s[2:0] select the functions as described in table 1.

Table 1. ALU Operation Select

Select Operation S[2:0]	Arithmetic Operation	Logic Operation
000	One's Complement of A	A
001	One's Complement of B	~ A
010	A + B	A & B
011	A - B	A | B
100	A * B	~ (A & B)
101	A / B	~ (A | B)
110	A % B	A ^ B
111	B + 1	~ (A ^ B)

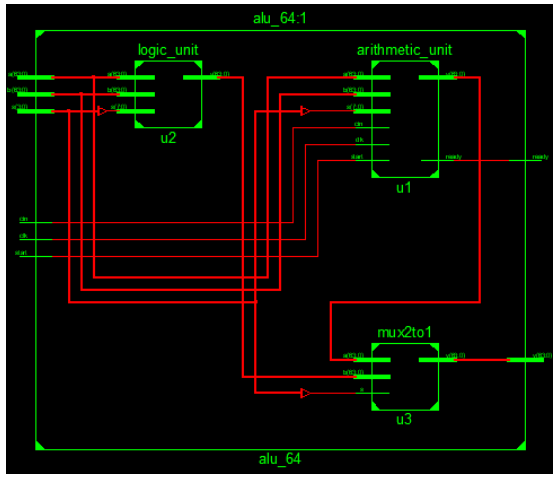


Fig.2: ALU architecture

The Paper is organized as section II describes the Adders Design. The section III details about the Multipliers Design. The section IV describes the ALU Designs using various Adders and Multipliers and the corresponding results along with parameters used for assessment and finally the paper is concluded.

2. Adder Design

The Regular carry select adder is shown in fig.3. it uses the parallel processing capability for calculation of carry for next block of bits to be added. Hence the focus is on reducing the delay corresponding to carry generation [1-5].

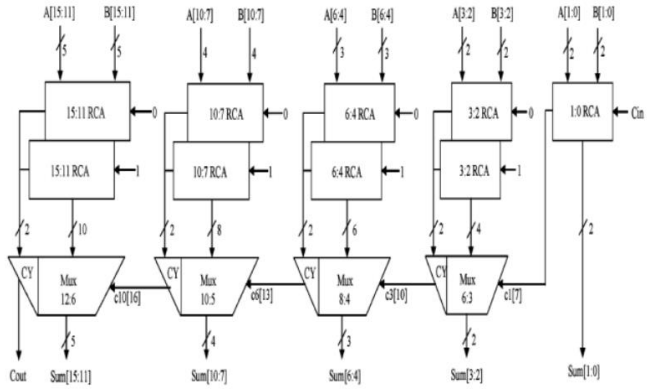


Fig.3 Regular Carry Select Adder

The figure 4 shows the modified carry select adder uses a binary to excess converter by considering carry input to that block as 1. Thus it offers the reduction in area and is known as square root carry select adder.

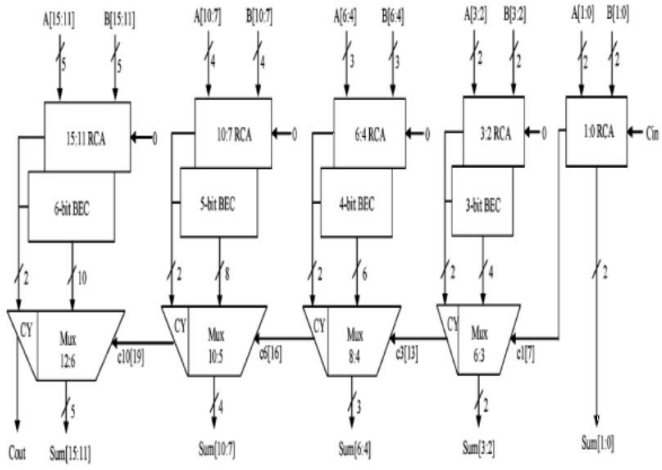


Fig.4: Modified Carry Select Adder

The table 2 shows the comparison of ALU Designs with various adders. CSA64 represents the 64-bit basic carry select adder. CSAVAD64 represents the variable blocks where instead of fixed length of input bits variable lengths are considered like initially 2-bits then 3 bits and so on.

MOD64LINCSLA represents the modified 64-bits linear carry select adder and MOD64SQRTCSLA represents the modified 64-bits square root carry select adder.

Table 2. Comparison of ALU Designs with Various Adders

Parameter	Alu_64_csa64.v	Alu_64_csavad64.v	Alu_64_mod64lincsla.v	Alu_64_mod64sqrtcsla.v
Number of Slices	913	929	936	942
Number of 4-input LUTs	1771	1803	1809	1822
Delay, ns	74.734	60.356	46.025	51.123
Power, mW	3.26	3.26	3.34	3.42

From table 2, it is clear that the area occupied in terms of 4-Input LUTs or Slices is less by 3% for 64-bit ALU with basic carry select adder. The delay is less by 38.4% for 64-bit ALU with modified linear carry select adder when compared with other designs. The power dissipation is less by 4.67% for 64-bit ALU with basic or variable carry select adder when compared with other designs.

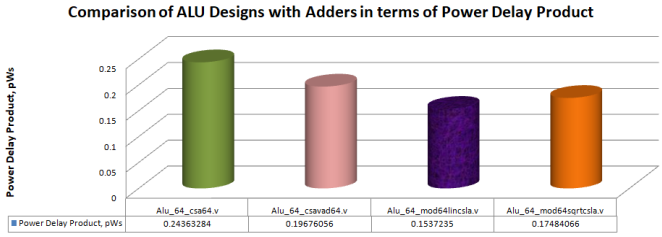


Fig. 5: Comparison of ALU Designs with Various Adders in terms of Power Delay Product

Figure 5 shows comparison of ALU Designs with various adders in terms of figure of merit i.e., power delay product. It shows that the 64-bit ALU with modified linear carry select adder improves by 36.9% when compared with other ALU designs using various carry select adders.

3. Multiplier Design

The complexity of multiplier is more when compared with adders in ALU [6-15]. This section deals with various fast adders. The shift- add based multiplier is shown in figure 6. Here the multiplicand is added directly whereas the multiplier is shifted right and added for final product result. This concept is applicable to n number of bits of inputs.

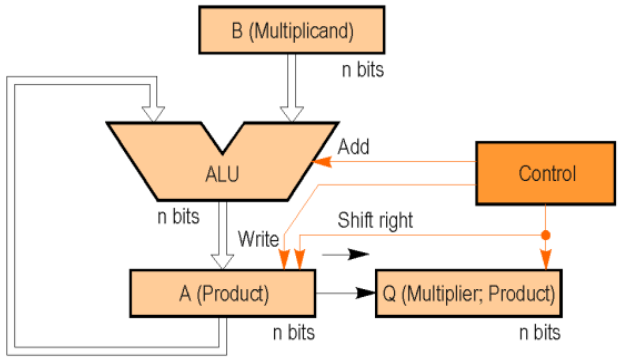


Fig.6: Shift – Add Multiplier

The Vedic multiplier is shown in figure 7, here the n bits are divided into low and high order bits and they are multiplied by using basic multiplication concept then they are added as shown in figure to obtain the final product result as fast as possible. By using this concept, the higher order input bits are obtained in the similar manner where the lower order bits again use the concept of vedic multiplication by using parallel evaluation.

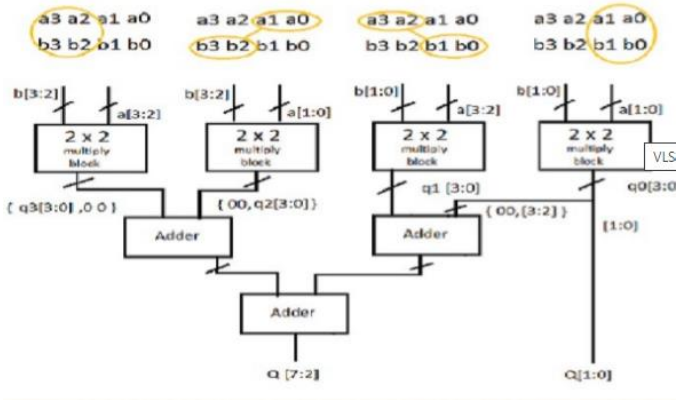


Fig.7: Vedic Multiplier

The carry save multiplier is shown in figure 8, where the half and full adders are used to find the final product. The time delay is equal to the delay of three half adders and four full adders.

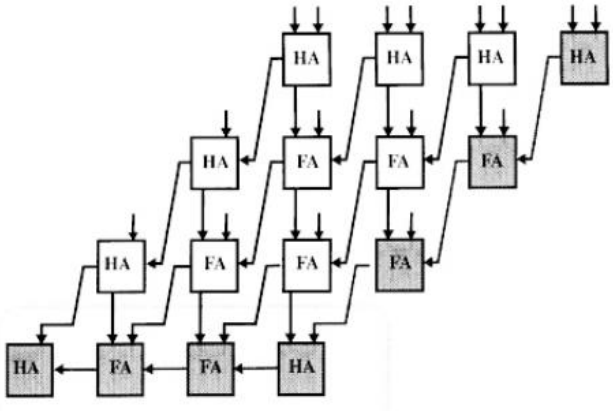


Fig.8: Carry – Save Multiplier

From table 3, it is clear that the area occupied in terms of 4-Input LUTs or Slices is less by 20% for 64-bit ALU with shift add multiplier. The delay is less by 14% for 64-bit ALU with vedic multiplier when compared with other designs. The power dissipation is less by 4.56% for 64-bit ALU with shift add multiplier when compared with other designs.

Table 3. Comparison of ALU Designs with Various Multipliers

Parameter	Alu_64_shiftaddmulv	Alu_64_vedicv	Alu_64_csmulv
Number of Slices	5706	7126	5883
Number of 4-input LUTs	10161	12655	10469
Delay, ns	47.133	40.487	40.588
Power, mW	25.13	25.23	26.33

Figure 9 shows comparison of ALU Designs with various multipliers in terms of figure of merit i.e., power delay product. It shows that the 64-bit ALU with vedic multiplier by 13.75% when compared with other ALU designs using various multipliers.

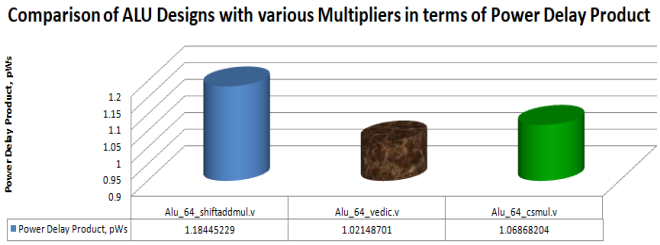


Fig. 9: Comparison of ALU Designs with Various Multipliers in terms of Power Delay Product

4. ALU Designs

The ALU designs are developed by using the various combinations of adders and multipliers. These designs are modeled in verilog HDL. They are functionally verified for Zynq 7000 series FPGA with device XC7Z020, Package CLG484 with a speed grade of -1,

which is a 28nm FPGA, in Xilinx ISE 14.5 and ISIM simulator.

From table 4, it is clear that the area occupied in terms of 4-Input LUTs or Slices is less by 17% for 64-bit ALU with basic carry select adder and shift add multiplier. The delay and power dissipation are less by 38.05% and 4.67% respectively for 64-bit ALU with modified linear carry select adder along with vedic multiplier when compared with other designs.

Table 4. Comparison of ALU Designs with Various Adders and Multipliers

Parameter	Alu_64_csa64_vedic.v	Alu_64_csavad64_vedic.v	Alu_64_mod64lincsla_vedic.v	Alu_64_mod64sqrtcsla_vedic.v	Alu_64_csa64_shiftaddmul.v	Alu_64_csavad64_shiftaddmul.v	Alu_64_mod64lincsla_shiftaddmul.v	Alu_64_mod64sqrtcsla_shiftaddmul.v	Alu_64_csa64_csmul.v	Alu_64_csavad64_csmul.v	Alu_64_mod64lincsla_csmul.v	Alu_64_mod64sqrtcsla_csmul.v
Number of Slices	3794	3810	3817	3821	3154	3172	3178	3180	3231	3245	3255	3258
Number of 4-input LUTs	6769	6797	6807	6818	5646	5678	5686	5694	5780	5804	5820	5829
Delay, ns	75.044	60.634	46.484	49.067	75.027	60.636	46.728	47.944	75.034	60.636	46.728	48.998
Power, mW	13.94	14.44	13.67	14.07	14.2	14.18	14.32	13.92	14.13	14.22	14.28	14.34

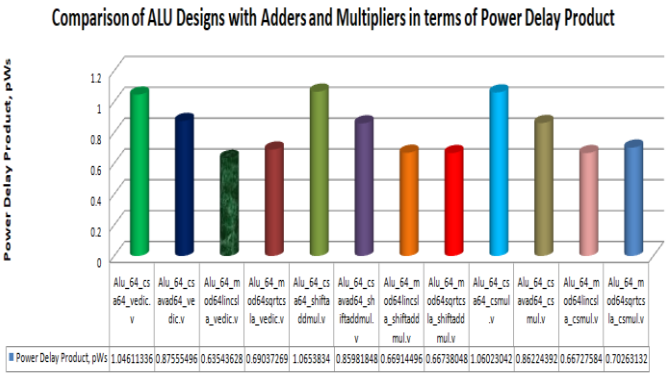


Fig. 10: Comparison of ALU Designs with Various adders and Multipliers in terms of Power Delay Product

Figure 10 shows comparison of ALU Designs with various adders and multipliers in terms of figure of merit i.e., power delay product. It shows that the 64-bit ALU with modified linear carry select adder and vedic multiplier by 40.35% when compared with other ALU designs using various adders and multipliers. Hence, the ALU design with modified linear carry select adder and vedic multiplier is best suited for fast and low power ALU design and ALU with basic carry select adder and shift add multiplier is the best option for area optimized ALU design. The simulation result is as shown in figure 11.

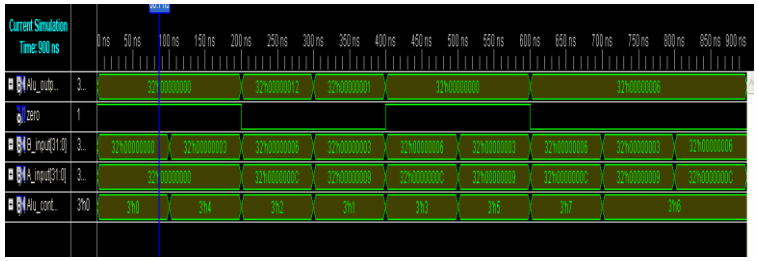
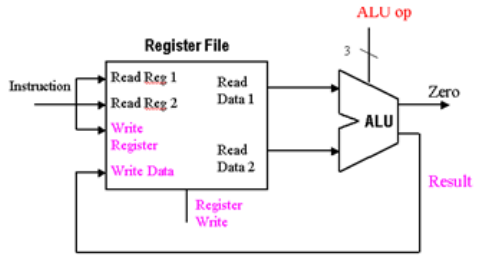


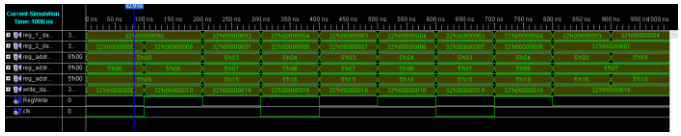
Fig.11. Simulation Result of ALU

When the MIPS design includes the 64-bit ALU with modified linear carry select adder and vedic multiplier the design was faster with less area occupied. The implementation of the data path for R-format instructions is fairly straightforward - the register file and the ALU are all that is required. The ALU accepts its input from the DataRead ports of the register file, and the register file is written to by the ALUresult output of the ALU, in combination with the RegWrite signal [16-17].

The instruction format is opcode r1, r2, r3 and the architecture is as shown in figure 12 with the corresponding simulation output.



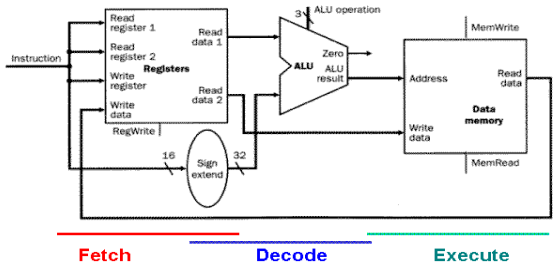
(a)



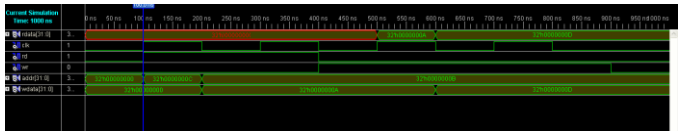
(b)

Fig.12: (a) R – instruction Implementation (b) Simulation waveform

The load/store instruction data path architecture and its simulation waveform are as shown in figure 13.



(a)



(b)

Fig.13: (a) Load/ Store Instruction Data Path (b) Simulation Result

But the single cycle data path has limitations like the critical path (longest propagation sequence through the datapath) is five components for the load instruction that takes atleast 4 or 5 units of time. This enables the use of multiple cycles of a much faster clock where the datapath actions can be interleaved in time i.e., MIPS with pipelining.

The area occupied was only 11 slices with 23.364ns delay for single cycle MIPS using the high performance ALU.

5. Conclusion

The MIPS is used to develop a customized fast processor. The ALU forms the basis for any computations in processor. Thus the complexity of ALU must be reduced. The designs developed in this paper use various forms of adders and multipliers. These designs are modeled in verilog HDL. They are functionally verified for Zynq 7000 series FPGA with device XC7Z020, Package CLG484 with a speed grade of -1, which is a 28nm FPGA, in Xilinx ISE 14.5 and ISIM simulator. Among the designs developed, the ALU design with modified linear carry select adder and vedic multiplier has improvement of 40.35% which proves to be best suited for fast and low power ALU design and ALU with basic carry select adder and shift add multiplier has an improvement of 17% and proves to be the best option for area optimized ALU design. When the combination of 64-bit ALU with modified linear carry select adder and vedic multiplier was used in MIPS, the design was faster with only 23.364ns with less area occupied i.e., 11 slices in the intended FPGA. To overcome critical path delay of load store instruction based MIPS, multi cycle MIPS can be implemented in future.

References

1. B. Ramkumar, Harish M Kittur “Low power and Area efficient carry select adder,”IEEE Trans,Vol.20,Feb 2012.
2. Shivani Parmar and Kirat pal Singh , ”Design of high speed hybrid carry select adder” , IEEE 2012.
3. T. Y. Ceiang and M. J. Hsiao,”Carry-select adder using single ripple carry adder”, Electron Let, vol.34, no.22, oct-2013
4. Ms. S.Manjui, Mr. V. Sornagopae, “ An Efficient SQRT Architecture of Carry Select Adder Design by Common Boolean Logic” ,IEEE, 2013.
5. U,Sreenivasulu and T.Venkata Sridhar, “Implementation of an 4-bit ALU using Low power and Area efficient carry select adder ”, International Conference on Electronics and Communication Engineering,20 May2012.
6. Hemanth. J, Dr. Dola Sanjay. S, “High Performance Vedic Multiplier Design Using Variable Precision”, TEST Engineering and Management, The Mattingley Publishing Co., Inc., March - April 2020 ISSN: 0193-4120 Page No. 13549 – 13552.
7. Haider Ali and Bashir M. Al-Hashimi “Architecture Level Power-Performance Tradeoffs for Pipelined Designs” IEE Circuits, Devices and Systems Series 18, 2006.
8. Erle, M.A.; Hickmann, B.J.; Schulte, M.J.; IBM, Macungie, PA “Decimal Floating-Point Multiplication” Computers, IEEE Transactions on Issue Date: July 2009 Volume: 58, Issue: 7.
9. Yogita Bansal, Charu Madhu and Pradeep Kaur, “High Speed Vedic Multiplier Designs: A Review”, Proceedings of 2014 RAECS UIET Punjab University Chandigarh, 06-08 March,

2014.
10. D.M.Smith, “Using Multiple Precision Arithmetic”, Computing in Science &Engineering IEEE publication, July/august 2003.
11. M.J. Schulte, E.E. Swartzlander, “A Family of VariablePrecision Interval Arithmetic Processors”, IEEE Trans on Computers, Vol. 49, No. 5, May 2000.
12. Serdar S. Erdem ,Çetin K. Koç¸ “A Less Recursive Variant of Karatsuba-Ofman Algorithm for Multiplying Operands of Size a Power of Two”, 16th IEEE Symposium on Computer Arithmetic, 2003.
13. Siva Kumar. G, B.K.N.Srinivasarao “Design of Area and Power Efficient Floating-point Co-processor” (ijaest) international journal of advanced engineering sciences and technologiesvol no. 10, issue no. 2, 249 – 251
14. A. Karatsuba, Ofman Yu, Multiplication of multiple numbers by mean of automata”, DokadlyAkad. Nauk SSSR 145, no 2, pp293-294, 1962.
15. MN Kumar, J Hemanth, KD Prasad, “VLSI Implementation of DWT Using Systolic Array Architecture”, International Journal of Recent Technology and Engineering (IJRTE) 1, 67-73, 0, 2012.
16. P. V. S. R. Bharadwaja, K. R. Teja, M. N. Babu and K. Neelima, "Advanced low power RISC processor design using MIPS instruction set," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 2015, pp. 1252-1258, doi: 10.1109/ECS.2015.7124785.
17. P.V.S.R.Bharadwaja, M.Venkata Naresh, Neelima Koppala, J Sai Krishna, “Design of Step-up Inexact MAC (IMAC) Unit for DSP Applications”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6S, March 2019, pp. 360 - 364.