ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter15872020.pdf
https://doi.org/10.30534/ijeter/2020/15872020

Architectural Enhancement of Processor with 8 Bit Multiplier and 16 Bit Co-operative ALU using VHDL

Tanaji M. Dudhane[1], T. Ravi[2]
[1]Research Scholar, Sathyabama Institute of Science and Technology, Chennai, India.
tmdudhane@yahoo.com
[2] Research Supervisor, Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, India.ravivlsi123@gmail.com

ABSTRACT

In the new world of science and technology with the tremendous changes in electronic gadgets such as microprocessors and microcontrollers, becoming an important blocks of the electronics industry. For any processor, CPU has integral part which is ALU, responsible for the efficient work of the processor. This paper integrates the computing ability of 8 bit processor with 16 bit co-operative arithmetic and logic unit and 8by8 bit multiplication. In this technology, 8 bit processor keeps its ability with excellent input/ output control with maintaining its simple architecture and low power consumption. FPGA offers a platform for designing and implementing soft cores for improvement in execution time and developing IP cores. 16F84 RISC processor is designed module wise developed with HDL language and implemented using FPGA with enhanced architecture. The results are obtained in three ways as firstly designing and implementing 16 bit CALU with 8 bit processor, secondly 8by8 bit multiplier and thirdly combined integration of 16 bit CALU and 8by8 bit multiplier. Simulation result for 8 bit multiplier shows execution time of 109.241 ns, with 16 bit CALU is 674.66 ns and with multiplier and 16 bit CALU is 583.33 ns, 76.09 % cycle saving is achieved with the implementation using reconfigurable hardware, improving execution time from 674.66 ns to 583.33 ns. The design is configured on Xilinx 14.6 using FPGA and implemented using Verilog or HDL.

Key words : ALU, CALU,8by8 bit, FPGA, RISC VHDL.

1. INTRODUCTION

Normal approach of design is to reduce the machine cycles, variation in data bus width, reduction in cost and implementing using FPGA for better performance. In the modern technology of electronic design, FPGAs are the solutions for the core processors as well as for the different modules of the processors such as arithmetic and logic unit, multiplier, barrel shifter, MAC unit. PIC16 series of Microchip is popular for development of microcontroller using 4 states T1 to T4. Uses of Verilog or HDL languages ease the development for improvement in execution time of instructions. Modifying the size of program counter, additional instructions incorporated with 8 bit processor. System on Chip (SOC) has the important block as microcontroller with deep pipeline of 5 Stages. The pipelining stage having memory read and write back cycles, increments 67% in maximum clock rate with 86% increment in MIPS [1].

When a normal processor is integrated with maximum number of data lines for ALU, it enhances its computational ability. ALU is the important block of any modern processors, cause of its speed and size having major contributions in overall performance of the processor related to cost. Modern digital signal processors (DSPs) and Application Specific Integrated circuits (ASICs) were the solutions but having limited Space [2]. The use of 8 bit processor is suggested in Cellphones, PCs and robots with FPGA and in mathematical operations, the movement of data from one location to another is also suggested [3].

Now, PLDs (programmable logic devices) and CPLDs (complex programmable logic devices) are especially field programmable logic arrays (FPGAs), having their appearance in power electronics for modern control applications of conference paper in [4]. Device utilization is presented and integration concept is summarized in the conference paper [5]. The characteristics of FPGA, time to market, low price, low energy consumption and its use for implementing low to high end processors [6], is the work carried out by the author. FPGAs are suited for low-end to high-end processors. Design and development of processor of any kind using Verilog or HDL language on FPGA, is a solution for single chip processor and its different modules as ALU, control unit, decoder unit using combinational circuits as the example in the journal article in [7].

Reduced Instruction Set Computer is the design methodology for modern as well as pre-processors using FPGA, due to the increase in capacity of number of gates and processor performance in terms of speed and gate count [8]. While designing the processor, its instruction size also taken into consideration. Instruction length is minimized for minimum area occupation in memory. The length of the instruction is 22 bit wide, which acquires more area to store the instructions in the memory. Instead of using 22 bit size, we have designed 15 bits of instruction size from which, most significant bit (MSB) determines the operation of 8 bit or 16 bit ALU [9]. As example of journal article in [10], implements the CPU design and its mapping using FPGA. In

the context of high end processors, with mass production for 8 bit processors which are low-end having strong demand because of having moderate performance, low power consumption and having lower cost. The enhancement from a basic 8-bit processor would increase its usability also maintaining its simple and low power architecture [11].

Implementation and design for ALU is discussed with reversible logic in [12]. Multiplier and accumulate unit (MAC) is used for the multiplication of two numbers and adding this product to the accumulator. The Presentation of conversion from an 8 bit to 16 bit is implemented in [13], with simple low cost and fast response architecture of low-end processors. The paper [14] contributes the implementation of 8 bit multiplier with RISC processor of 8 bit for the improvement in features of the processor. Before the earlier design, 6 to 10 ASICs, required and all the modules are assembled together and communicates through a common data path [15]. 8 bit RISC processor is mapped into FPGA, described in Verilog or HDL having simple 8 bit arithmetic and logic unit is presented in [16], and implemented using SPARTAN-6. As there is increment in speed, power decreases and increase in maximum computability. Different platforms for implementation of 8 bit microcontrollers are discussed with speed improvement and comparative work with 16 bit microcontrollers [17].

The basic concern is to design high performance processor, because there are tremendous advancements in the transistor technology. Multiplier is major part contributes for the performance of the processor [18]. Design and implementation using VHDL with integrated devices for the realization of mobiles and modern devices is the work carried out in [19]-[21].Microcontroller based monitoring of blood pressure and development of devices with GPS and SMS facility as in paper [22] where Arduino microcontroller is used. Different adders are compared and speed of arithmetic unit is enhanced using adder [23], the architecture is designed, implemented and evaluated on Xilinx. The author focuses on nanotube transistor using single wall as an element for digital circuit design and suggest the alternate technology as Nano for semiconductor technology [24].Power and path delay parameters are compared with proposed and existing adder in [25] showing the saving of power and path delay for the design of full adder with CLU adder. RISC processor having multiplier unit and 16 bit co-operative ALU implemented using Xilinx. As in the example of conference paper [26], implements barrel shifter and Universal Shift Register. Control unit developed for this processor, which gives signals to the ALU and multiplier and generates the desired signals.

For reduction of power for the use of processer implementation using reconfigurable hardware, number of gates reduced. Field programmable gate array offers the modified solution in the design tremendously, with minimum use of gate count. Different arithmetic operations can reduce the power and provide the maximum throughput is presented in the paper [27], as well as power consumption, data bus size, execution time year wise analysis. Power consumption at the level of execution of instruction depends on the machine cycle, with implementation of FPGA . FPGA and ASIC technology is compared for the execution of arithmetic, memory access and control instructions for consumption of dynamic power. The Comparison between AISC and FPGA have limitations as example in [28] as in ASIC, for full logical block required a single flip- flop while in FPGA may be fit in one single bit register .The adiabatic logic is developed for MIPS processor computation, CMOS implemented and uses the characteristics improvement of delay time, area and reduction in power [29].

The article in the journal [30] proposes the sub word parallelism (SWP) for RISC processor for multimedia applications. The performance of the processor is increased at instruction level and at operation level. There is vast change in the technological development of the processor software and hardware but there is gap of speed between the processor and the memory. On-chip memory is incorporated to reduce the gap between memory and processor [31]. For the sine and cosine transforms which are for video coding using CGRAs, is presented in [32]. The article proposes with 4.1W power dissipation with 200 MHZ operating frequency. We have proposed the architecture using reconfigurable hardware for improvement in speed and with execution of 11 instructions of 16 bits.

This paper is organized as follows: Section 2 explains the material and method used for the 16 bit Co-operative ALU with 8 by 8 bit multiplier to 8 bit RISC processor architecture. Result is discussed in Section 3.. The paper is concluded with section 4.

2. MATERIALAND METHOD

We have designed and implemented soft core RISC processor, with simulation and developed using FPGA platform, Xilinx 14.7 ISE as software, 8 bit 16F84 RISC processor is used as soft-core and Spartan-6 kit is used as FPGA.

Reduced instruction Set Computers (RISC) is the design mainstream using FPGA in these days. 8 bit processors, having their importance in modern world applications with changes in its architecture. The processor having its 8 bit arithmetic logic unit performing operations on 8 bit data. The size of the instruction is 14 bit wide.

In this paper, the size of the instruction is increased by one bit in MSB, making 15 bit instruction size, used for selection of 8 bit and 16 bit co-operative ALU. 16 bit CALU is designed and integrated with RISC Processor, performing operations on 16 bit instruction size. New 16 bit, 11 instructions are executed by CALU. If MSB bit in instruction size is 1, 16 bit operations are performed otherwise 8 bit operations.

2.1 CALU with Processor

Based on integration technology, 16 bit CALU is integrated with 8 bit processor as shown in Figure 1.

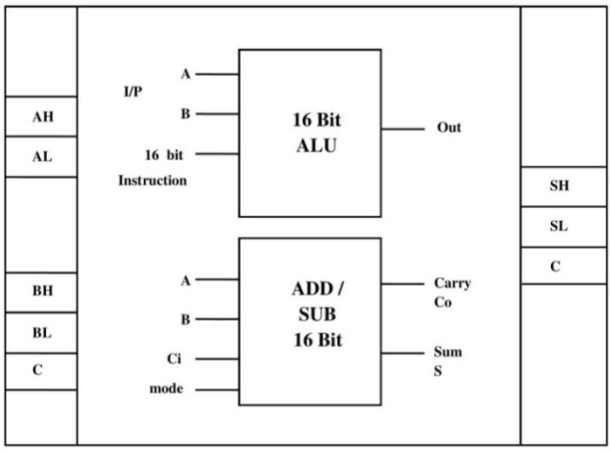

Figure 1: Integrated 16 bit CALU with RISC processor.

The basic architecture of the processor with 8 bit is not disturbed but the 16 bit arithmetic and logic unit calling as co-operative, executing the instructions having 16 bits. The addresses for the input register A, B and sum (S) are mapped as the special function registers in the internal memory. Designing and implementing co-operative ALU enhances the computing ability of the processor that it executes additional 16 bit instructions. Multibyte operations are performed in a single cycle, increasing performance of the 8 bit processor. All the 16 bit instructions were performed in single cycle, enhancing its speed of execution of instructions. 11 cycles for 11 instructions with integration of 16 bit CALU required instead of total 46 cycles required for the same instructions execution with 8 bit RISC processor.

With the, integration 16 bit CALU, there is change in the original instruction size of 14 bit, 15 bit is used. There is distinction between 8 bit and 16 bit, which are 8 bit and 16 bit instructions. Proposed format of the instruction set is shown in Figure 2.

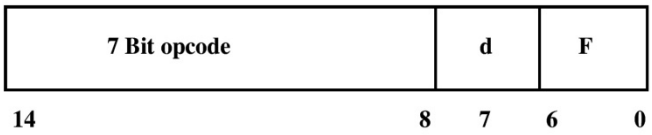

Figure 2: Proposed format of Instruction Set

F= 7 bit file register address.
d=1 for the destination F.
d=0 for the destination W, default d = 1.
The bits from 6 to 0 (7 bits) are used for address of file register,'d' is used for the location of the result when '1', destination is 'f' otherwise destination is working register W. The instructions are categorized in bit, byte oriented, literal, control operations and CALL, GOTO instructions separately. Table 1 shows the 8 bit instructions implemented by the 8 bit RISC processor without the integration of multiplier and 16 bit CLAU.

For bit oriented instructions, three bit address is determined by 'b' and f is the 7 bit file register address. For byte oriented instructions, 'f' determines file register7-bit address and 'd' determines the location of the result. Literal and control operations are carried with k having 8 bit immediate value.

Table 1: Instruction set for 8 bit ALU

Mnemonics	Operands	Description
Single Bit Manipulation		
bcf	f,b	Clear bit b of reg. f ,where b = 0 to 7
bsf	f,b	Set bit b of register f ,where b=0 to 7
Data transfer and clear		
clrw		Clear W
clrf	f	Clear f
movlw	k	Move literal value to W, putting result into W
movwf	f	Move W to f
movf	f,F	Move f to F
movf	f,d	Move f to W or F depending on d bit
swapf	f,d	Swap nibbles of f, putting result into F or W
Increment /Decrement /Complement		
incf	f,d	Increment f, putting result in F or W reg.
decf	f,d	Decrement f, putting result in F or W reg.
comf	f,d	Complement f, putting result in F or W reg.
Logical		
Andlw	k	AND literal value into W
Andwf	f,d	AND W with f , putting result in F or W reg.
Iorlw	k	Inclusive - OR literal value into W
Iorwf	f,d	Inclusive - OR W with f, putting result in F
Xorlw	k	Exclusive - OR literal value into W
Xorwf	f,d	Exclusive - OR W with f, putting result in F
Arithmetic		
Addlw	k	Add literal valve into W
addwf	f,d	Add w and f putting result in F
sublw	k	Substract W from literal value ,putting result in W
subwf	f,F	Substract W from f putting result in F
Rotate		
rlf	f,F	Copy f into F,rotate F left through carry bit
rrf	f,F	Copy f into F, rotate F right through carry bit
Conditional branch		
btfsc	f,b	Test bit b of register f where b = 0 to 7,skip if clear
btfss	f,b	Test bit b of register f where b = 0 to 7, skip if set
decfsz	F,d	Decrement f, putting result in F or W reg. , skip if zero
incfsz	F,d	Increment f, putting result in F or W reg. , skip if zero

Figure 3 shows the internal structure of 8 bit ALU, which implements eight, 8 bit instructions. Addlw, subwf, etc. Any instruction is ORed together and ANDed with T3 state for enabling of the multiplexer. Output of one multiplexer is 'a'

and output of second multiplexer is 'f' which is 8 bit. Third multiplexer is used for the carry, if generated. Byte adder is used for addition of 8 bits of 'f' and 'w'. The most significant two bits of the address of instruction are used for select lines of 2:1 multiplexer and 4 bits from 11-8 position bits in instruction register are used for select lines of multiplexer.

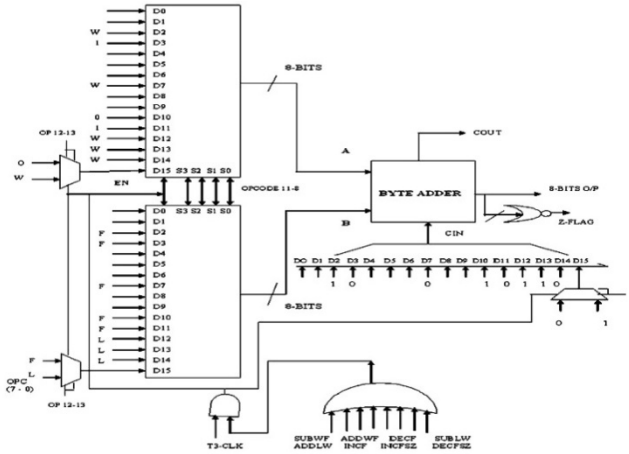

Figure 3: Proposed internal structure of 8 Bit ALU for FPGA.

2.2 Design of Multiplier for 8 Bit RISC Processor

Original 8 bit RISC processor, not having the multiplier. Its computability is increased by integrating 8 by 8 bit multiplier. There is addition of multiplication operation. For multiplication process add/shift method is used. Figure 4 shows the important blocks of multiplier as 8 bit multiplicand, 8 bit multiplier, adder of 16 bit and control logic for add and shift logic. After multiplication, result is shifted left by 1 bit position 8 times. Shifting one bit towards left whether it is 0 or 1, decides the action of control unit to add 0 or previous result, gets added in multiplicand. Figure 4 shows the proposed diagram of the 8by8 bit multiplier.

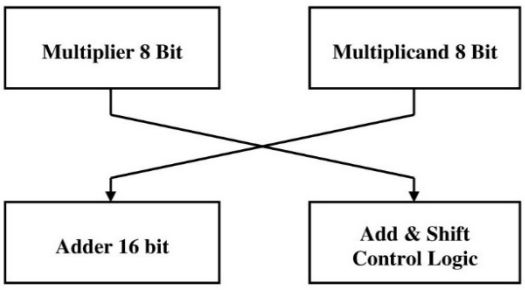

Figure 4: Proposed Diagram of the 8 by 8 Bit multiplier.

The multiplier is based on the shift and add method. Two 8 bit data is multiplied and 16 bit register is designed for the result of multiplication. This multiplier adds one more instruction, which is very important in iterative operations and in prototype before the complete fabrication of the device. Add and shift control logic works on the right shifting bits of multiplier and left shifting 8 bits of multiplicand.

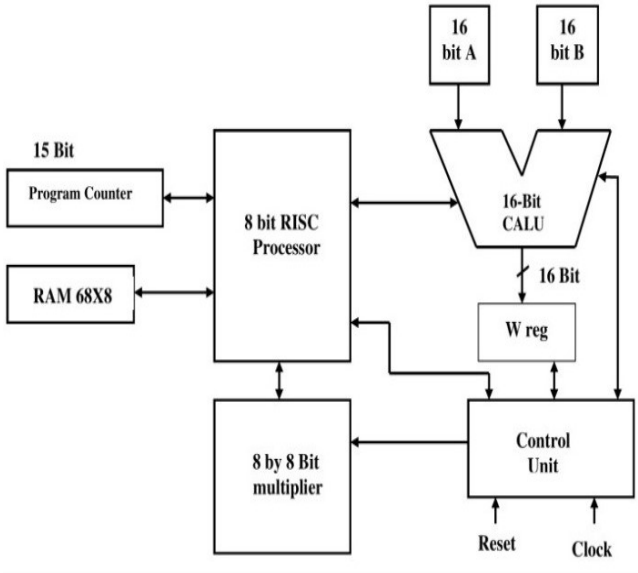

Figure 5: Proposed block diagram of enhanced RISC processor

Figure 5 shows the proposed diagram of enhanced processor with 16 bit CALU and 8 bit multiplier. The design is integrated with additional 16 bit arithmetic and logic unit performing 16 bit instructions in single cycle and with 8 bit multiplier, enhances the processing capability. Multiplier unit is designed which is the new feature of 8 bit RISC processor, enhancing its computability. While designing 16 bit CALU with 8 bit RISC processor and 8-bit multiplier, operations are carried one by one in a single device at a time. So when 8 bit ALU is performing its operations other elements are idle and when 16 bit CALU is performing operations on 16 bit data, remaining devices are idle and vice-versa. In parallel computation, design is less effective because of working a single device at a time.

3. RESULT AND DISCUSSION

FPGA platform used for simulation of the design, verified on Spartan using Xilinx software. Three different simulation results are achieved, namely using only 16 bit CALU, then with 8 bit multiplier and finally with both the devices. Figure 6 shows the simulation of 16 bit instructions. Addition of 2, 16 bit operands is carried out, implementing 11 instructions.

3.1 Simulation of co-operative ALU

Result is tabulated in the form of using number of slice registers, LUTS, bonded IOBs, showing the occupational area.

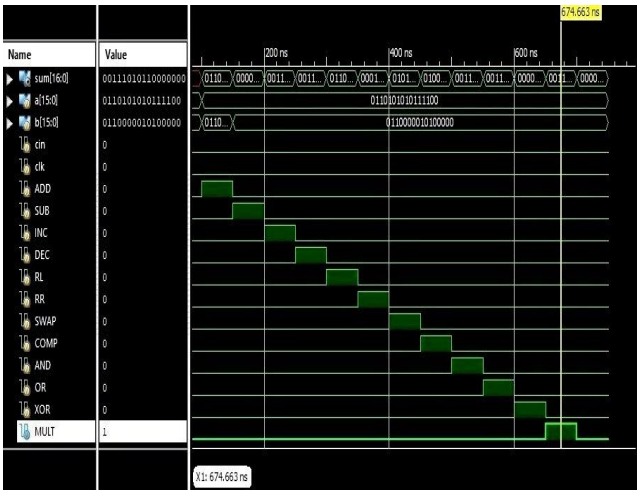

Figure 6: Simulation diagram of 16 Bit CALU with 11, 16 bit instructions.

3.2 Test bench waveforms of 8 by 8 bit multiplier

The 8 bit operands that is multiplier and multiplicand, gets multiplied and result is obtained in simulation. Device utilization is calculated for the multiplier. Figure7 shows the test bench waveform of 8 bit multiplier. Two 8 bit numbers are multiplied and 16 bit register is used for the storage of the result. The number of cycles are saved with 109.241 ns are recorded.

Simulation results are captured for 16 bit ALU with 674.66 ns, for 8 by 8 bit multiplier having 109.241 ns and with combined CALU and multiplier 583.33 ns.

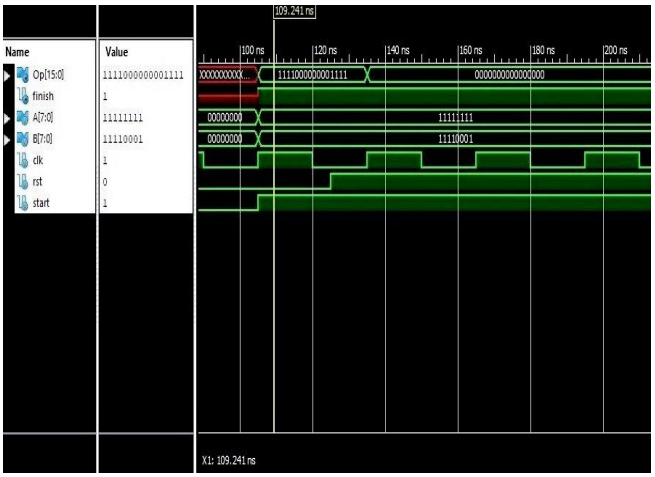

Figure 7: Test bench waveform of 8 by 8 bit multiplier

3.3 Waveform of 8 bit CALU and multiplier

Figure 8 shows the simulation waveform of the proposed RISC processor, with both the element as integrated with RISC processor.

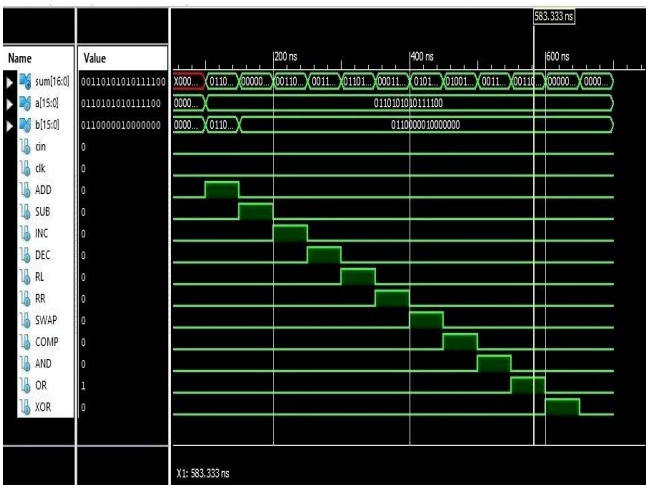

Figure 8: Simulation waveform with 16 bit CALU and 8 bit multiplier.

Above simulation implements the 11, 16 bit instructions, with one more additional multiplication instruction. Arithmetic instructions Add, Sub, Inc, Dec, rotate instructions as Rotate left, Right, Swap and logical instruction as And, Or, Xor and Complement are implemented with multiplication instruction of 2, 8 bit operands.

Table 2: Performance analysis of proposed 8 bit processor with CALU and multiplier

Parameter Summarization		For 16 Bit CALU	For 8 by 8 Bit Multiplier	with CALU and Multiplier
No. of slice LUTs	Used	135	155	300
	Available	101400	101400	101400
	% Util.	0.13	0.15	0.3
Slice Logic Distribution	Used	48	53	99
	Available	25350	25350	25350
	% Util.	0.19	0.21	0.39
Bonded IoB	Used	61	36	62
	Available	400	400	400
	% Util.	15.25	9	15.5

Table 2 shows the performance and device utilization of the three different devices. Utilization summary shows that, with integrating one element, there is slight increment in utilization towards integrating second and third element. Thus, in case of bonded IOB percentage utilization, increases from 15.25% to 15.5% with multiplier. Use of FPGA is the crucial part for improvement in performance in terms of slice LUTS and slice logic distribution. Total number of slice LUTs used are 0.3 %, logic distribution used is 0.39 % and bonded IOB with 15.5 % in the simulation.

Figure 9: Addition of two 16 bit numbers with 8 bit processor

Figure 9 shows 8 bit processor adding two 16 bit numbers.
16-bit addition with 16 bit CALU
Addword: requires one clock cycle.
16-bit addition original 8 bit core requires 6 cycles.

Table 3: Comparison between 8 bit and 16 bit

16 BIT addition	**No. Of Cycles**
With 8 bit core	6
With 16 bit CALU core	1

Table 3 shows the required number of cycles for the execution of 16 bit instruction. The simulation result of Addition instruction of two 16 bit operands, shows that, with 8 bit processor, it requires 6 cycles for execution and with 16 bit CALU, it performs the same operation in single cycle. Thus there is improvement in 83.33% of saving in cycle for a single instruction, with 16 bit CALU.

4. CONCLUSION

The 8 bit processors with their demand in different areas of applications with small modifications in architecture, they are performing in efficient manner with small device utilization. The constraints are power, area, computing ability, data bus size and addressing bus. The developed design, having the ability for implementing more number of 8 bit and 16 bit instructions. Though there are 16, 32 bit processors, this processor is suitable for low power, low area occupancy using FPGA. The saving in device utilization is achieved in terms of IOB's, LUTS.

From the above result it is concluded that 8 bit processor, with 8 bit multiplier and 16 bit CALU, performing efficiently and capable for executing 8 bit and 16 bit instructions. This processor with integrating elements, using FPGA platform, suitable for mathematical operations with improvement in execution of instruction.

ACKNOWLEDGMENT

Tanaji M Dudhane ,the author would like to grateful thanks Rajgad Dnyanpeeth's Shri Chhatrapati Shivajiraje College of Engineering, Pune for giving opportunity for the research work.

REFERENCES

[1] X. Li, LongweiJi, B. Shen, wenhong Li, and Qianlingzhang. **"VLSI Implementation of a high-performance 32-bit RISC Microprocesser,"** *IEEE, International Conference on Communications, Circuits and Systems and West Sino Expositions*, Chengdu, China, Vol. 2, pp. 1458- 1461, 2002.

[2] Paul Metzgen, and Holmers Farm Way. **"A High performance 32-bit ALU for Programmable Logic,"** *Proceeding of the 2004 ACM/SIGDA 12th International Symposium on Field Programmable gate array*. pp. 61-70, 2004. https://doi.org/10.1145/968280.968291

[3] E. Ayeh, K. Agbedanu , Y. Morita, O Adamo, and P. Guturu**. "FPGA Implementation of an 8-bit Simple Processor,"** *IEEE. Region 5, Conference Kansas City, MO*, pp.01-05, 2008.

[4] Florin M, Mihai C, Constantin S, and Dan F.**"Implementation of a 32-bit RISC Microcontroller-FPGA prototype,"** *4th International Conference on Inter disciplinary in Education ICIE'09.* Lithuania. pp. 193-199, 2009.

[5] Jesse Benson, Ryan Cofell, and chrisFrericks. **"Design, integration and implementation of the DySER hardware accelerator into OpenSPARC,"***IEEE, International Symposium on High-performance computer Architecture, New Oreleans,* LA. pp. 01-12, 2011. https://doi.org/10.1109/HPCA.2012.6168949

[6] Eric Monmasson, M. arcian, and N. cirstea. **"FPGAs in Industrial Control Applications,"** *IEEE Transactions on Industrial Informatics*. Vol. 7, Issue 2. pp. 224-243, 2011. https://doi.org/10.1109/TII.2011.2123908

[7] S. Kaliamurthy, and MUS sir. **"VHDL Design of FPGA Arithmetic Processer,"** *Global Journal of Research in Engineering.* [S.I.]. Vol.11, no.6, pp. 31-35, 2011.

[8] R. Uma. **"Design and Performance Analysis of 8-bit RISC Processor using Xilinx Tool,"** *International journal of Engineering Research and Applications.* Vol.2. issue 2,pp. 053-058, 2012.

[9] A. J. Salim, S. I. M. Salim, N. R. Samsudin, and Y. Soo.**"Instruction set Extension Through partial Customization of Low-EndRISC Processor,"** *Australian Journal of Basic and Applied Sciences*, Vol. 7. Issue. 6. pp. 678-687, 2013.

[10] Cesar Giacomini, Penteado, Edward David Moreno, and S. T. Kofuji. **"An integrated - approach for designing and Testing specific processors,"** *International Journal of VLSI design and Communication System*, Vol.4. Issue. 5. pp. 1-17, 2013.

[11] A. J. Salim, and SIMsalim. **"Conversion of an 8 bit Soft core RISC Processor,"** *Journal of Electronics*, 2013.

[12] K. Arunachalam, M. Perumalsamy, C. Kalyana Sundaram, and J. S.kumar. **"Design and Implementation of a Reversible Logic Based 8-Bit Arithmetic and Logic Unit,"** *International Journal of Computers and Applications.*Vol.36, Issue.2,pp. 49-55, 2014.

[13] Khoi-Nguyen LE-HUU, Anh-Vu, Thanh VU, Quoc-Minh, and Vy LUU. **"A Proposed RISC Instruction Set Architecture for the MAC Unit of 32-bit VLIW DSP processer Core,"** *International Conference on Computing, Management and Telecommunication*. pp. 170-175, 2014.

[14] Tanaji M Dudhane, and T. Ravi. **"Enhancement of Features of 8 Bit RISC Processor by Implementing 8Bit Shift/Add Multiplier,"** *International Journal on Emerging Technologies*, 11(2), pp. 770-774, 2020.

[15] Nupur Gupta, Pragati Gupta, Himanshi Bajpai, RichaSingh, and Shilpa Saxena. **"Analysis of 16 Bit Microprocesser Architecture on FPGA Using VHDL,"***International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,*Vol.3, Issue. 4, pp.8979-8986, 2014.

[16] M. Herrera and F. Viveros. **"Asynchronous 8-bit processor mapped into an FPGA device,"***2014 IEEE Colombian Conference on communications and computing (COLCOM),* Bogota, pp. 1-7, 2014.

[17] I. mezzah, H. chemali, S. Mezzah, O. kermia, and O. Abdelmalek. **"MCIP: High Configurable 8-bit Microcontroller IP-Core,"** *Science and Information conference,* London, (Uk). pp.1387-1390, 2015.

[18] M. Jhamb, Garima,and H. Lohani. **"Design, Implementation and Performance comparison of multipliers topologies in Power-delay space,"** *Engineering Science and Technology, an International Journal.*pp.01-09. doi.10.1016/j.jestch.2015.08.006, 2015.

[19] V. V. Balpande, A. B. Pande, and M. J. walke**, "Design and Implementation of 16 Bit Processer on FPGA,"** 2015.

[20] M. F. Tolba, A. H. Madian, and A. G. Radwan**. "FPGA realization of ALU for mobile GPU,"***3rd International Conference on Advances in Computational Tools for Engineering Applications, (ACTEA),* Beirut. pp.16-20, 2016.
https://doi.org/10.1109/ACTEA.2016.7560104

[21] M. Kantawala**. "Design and Implementation of 8 bit and 16 bit ALU using Verilog Language,"** *International Journal of Engineering Applied Sciences and Technology.* Vol. 3. Issue 2. pp.30-34, 2018.

[22] E. B. Panganiban. **"Microcontroller-based Wearable Blood Pressure Monitoring Device with GPS and SMS Feature Through Mobile App,"** *International Journal of Emerging Trends in Engineering Research*. Vol. 7(6), pp.32-35, June 2019.
https://doi.org/10.30534/ijeter/2019/02762019

[23] T. Ravi**. "10-Nanometer Carbon nano tube field effect transistor based high celerity transposed polyphase decimation filter,"** *Materials Today Proceedings*, Vol.3, Issue.6, pp.1799-1807, 2016.
https://doi.org/10.1016/j.matpr.2016.04.077

[24] R. Rashvenee, D. Roshinikeerthana, T. Ravi, and P. Umarani.**"Prominent Speed Arithmetic Unit Architecture for Proficient ALU,"***ARPN Journal of Engineering and Applied Sciences*, Vol. 11. Issue. 15, pp. 9013-9018, 2016.

[25] M. S. Kumar, F. Noorbasha, S. Inthiyar, M. Jameela, A. Sandhya, Md. Imran, S. Kumar and Tulasi**. "Low power Carry Look-Ahead Adder using Transmission Gate Multiplexer,"** *International Journal of Emerging Trends in Engineering Research*. vol.8(1),pp.13-17,Jan.2020.
https://doi.org/10.30534/ijeter/2020/03812020

[26] P. Trivedi, and R.P. Tripathi.**"Design and Analysis of 16 bit RISC processor using low Power Pipelining,"** *International Conference on computing, communication and Automation (ICCCA), IEEE*. pp.1294-1297, 2015.

[27] V. C. Rodriguez. **"ALU Designs for Power and Speed Optimization in the 21st century",** *Microelectronics*, vol.45, no.5.pp.1-4.

[28] C. C. Glavan, M. Marcu, A. Amaricai, S. Fedeac, M. Ghenea, Z. Wang, and A. Chattopadhyay**. "Direct FPGA based Power profiling for a RISC Processor,"** *IEEE, Instrumentation and Measurement Society*, pp.1-6.

[29] C. O. Campos, R. Celis, I. K. Hanninen, C.S Lent, A. O. Orlov, and G. L. Snider. **"A Mini-MIPS Microprocessor for Adiabatic Computing,"** *IEEE*. 2016.

[30] S. Khan, M. Rashid, and F. Javaid**. "A high performance processor architecture for multimedia applications,"** *Computers and Electrical Engineering,*2017,
https://doi.org/10.1016/j.compeleceng. 2017.09.027.

[31] D. Efnusheva, and A. Tentov**. "Design of Processor in memory with RISC- modified memory- Centric Architecture,"** *Advances in Intelligent Systems and Computing,*. pp. 70-81,2017.
https://doi.org/10.1007/978-3-319-57264-2_7

[32] S. Nouri, R. Ghaznavi, and J. Nurmi. **"Design and Implementation of multi- purpose DCT/DST-Specific Accelerator on Heterogeneous Multicore Architecture,"** *IEEE*, 2018.
https://doi.org/10.1109/NORCHIP.2018.8573457