www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 7 July 2015, Page No. 13395-13402

Design and FPGA-Based Implementation of A High Performance 64-Bit DSP Processor

Jadapalli Pavani, K.Dhanunjaya(phd)
Department of Electronics and communication Engineering
Audisankara College of Engineering & Technology, gudur
(Autonomous)

Abstract— To meet the faster processing demand in consumer electronics, performance efficient DSP processor design is important. This paper presents a novel design and FPGA-based implementation of a 64 bit DSP processor to achieve high performance gain for reduced instruction set DSP processors. The proposed design includes a hazard-optimized pipelined architecture and a dedicated single cycle integer MAC to enhance the processing speed. Performance of the designed processor is evaluated against existing similar reduced instruction set DSP processor (MUN DSP-2000). Synthesis results and performance analysis of each system building component confirmed a significant performance improvement in the proposed DSP processor over the compared one.

Keywords—DSP processor; FPGA; Pipelined; Single cycle MAC; Hazard Handling.

I. INTRODUCTION

With the advent of personal computer, smart phones, gaming and other multimedia devices, the demand for DSP processor is ever increasing. The information world is migrating from analog to DSP based systems to support the high speed processing. In the past, successful research effort has been made to integrate complex signal processing modules with the conventional processors to optimize the speed [1]. This paper demonstrates a novel design and FPGA based implementation of a 64 bit pipelined Digital Signal Processor with reduced instruction set. The design is modeled with behavioral Verilog Hardware Description Language. The processor is designed to support the basic DSP operation like digital filtering (we have designed FIR filtering). The processor is integrated with a two stage pipeline which optimizes the speed by reducing propagation delay. Reduced propagation delay is ensured by allocating every step of an operation into independent pieces of hardware and running all operations in parallel. This two stage pipeline also provides a better Cycle per Instruction (CPI) of 1 because all the instructions need only 2 cycles to complete an operation.

The computation speed of a DSP processor can be enhanced by incorporating General Purpose Processors (GPP) architectures into DSPs by retaining the functions critical to DSP [2-3]. To enhance processing speed of the proposed processor, a subset of the complete instruction set of a multi cycle RISC processor is included in this design. In addition to this, by incorporating two stage of pipeline, a better throughput is achieved for less number of instructions of this processor. Moreover, the hazard optimization for the pipelined architecture ensures a better performance of the processor. A performance evaluation shows that by using two stage of pipeline, the proposed design has achieved 12.06 MB/s of throughput over 8 MB/s of an existing similar reduced instruction set DSP processor- MUN DSP-2000[4] which has a five stage of pipeline.

A DSP processor is also characterized by fast multiply-accumulate and multiple-access memory architecture [5]. The memory of a DSP processor is guided to optimize the overall speed of the processor. Data and instructions must flow into the numeric and sequencing sections of the DSP on every instruction cycle [6]. There can be no delays and everything about the design focuses on throughput. To ensure better throughput, Harvard architecture is used in which memory is typically uses two separate memory buses. By using Harvard architecture instead of Von Neumann architecture, it doubles the throughput of this processor because separation of data and instructions gives this DSP processor the ability to fetch multiple items on each cycle.

FPGAs are well suited for reducing combinational path as well as employing parallel operations which can provide a better solution for manipulating speed [7]. A design implemented on XILINX Spartan-3E has the ability to provide high throughput and avoid the lengthy development cycles, and the inherent inflexibility of conventional ASICs. In addition to this, digital filter implementation on FPGAs allow higher sampling rates than available from traditional DSP chips and lower cost [8].FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs [9]. This can help the designer to perform the basic processes faster. These advantages are the key reasons for choosing FPGA to implement this design work.

For digital filter applications, an efficient MAC operation requires one single system clock cycle to compute a successful filter output [10]. The proposed design is modeled and synthesized for a dedicated MAC unit so that FIR filtering computations can be done in one cycle. The following key features are conducive to achieve improved throughput and speed gain of the proposed DSP processor:

- Pipelined data-path design
- Separate instruction and data memory
- Single cycle MAC
- Hazard free Finite State Machine

The Performance analysis of the proposed design confirmed a speed gain of 13 MHz over 10 MHz of existing MUN DSP-2000. The details and extended version of the proposed design and implementation is published in the *Proceedings of International Journal of Scientific and Engineering Research* [11].

The rest of the paper is organized as follows: Next few sections (II through VII) describe the design of the proposed DSP processor. Section VIII presents a performance analysis and discussion. Finally, this paper concludes in section IX with summarization of the design process and outlines to future works.

II. DEDICATED MAC

The proposed design contains a dedicated processing unit for multiply and accumulation. This design has implemented a 8 tap MAC. This MAC is dealing with sample values which are signed values. Signed numbers are converted to unsigned numbers before it goes to the shift register file. The corresponding MAC is modified to operate in one single cycle. The simulation result of the single cycle operation is shown in Fig. 1.

This is a dedicated processing unit for manipulating FIR filter operation. The MAC data path is kept apart from the general purpose data path where ALU is free from filtering task to deal with other instructions. For this two stage pipelined DSP processor, ALU needs 10 cycles to complete a two bit multiplication.

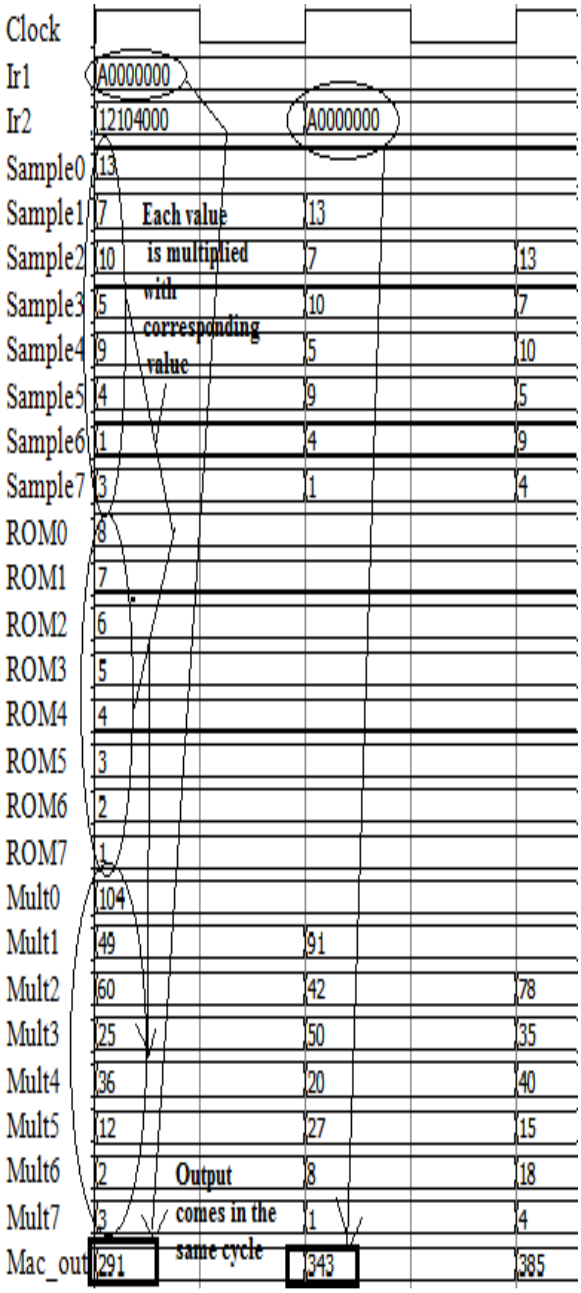

Figure 1. Simulation result of single cycle MAC

The comparison between ALU and single cycle MAC is given below where the total required cycles are shown in Table I to complete 2*2 multiplications.

TABLE I. COMPARISON TABLE BETWEEN ALU AND MAC FOR EXECUTING 2*2 MULTIPLICATIONS

Multiplication	**Using ALU**		**Using Single cycle MAC**	
	Operation	***No. of Cycles to complete operation***	***Operation***	***No. of Cycles to complete operation***
13 ×18 104 –PP1 13 -PP2 234-ADD (MULT Result)	Bit wise AND (13* 8)	2 cycles	External Multiplier 1	1 cycle
	Bit wise AND (13*1)	2 cycles	External multiplier 2	
	Partial Product 1	2 cycles	Addition	
	Partial Product 2	2 cycles		
	Addition	2 cycles		
Total no. of cycles to complete operation	10 cycles		2 cycles (Due to two stage pipelining)	

Since computations are done by a large number of hardware components in only one cycle, the duration of the clock cycle must be longer (twice) than the summation of all propagation delays of individual hardware components in a single cycle implementation of MAC. To reduce the propagation delay, eight external multipliers are working parallelly which reduces clock period. This enhances speed of the processor which improves overall system performance. Based on the timing analysis, the worst delay of one multiplier is 23.764 ns. Using one multiplier for overall calculation, the delay would be 23.764*8 = 190.112 ns. By using eight traditional multipliers, the delay is now only 23.764 ns as all the multiplications are performed parallelly to compute the filter output. This improves the overall speed. Then all the multiplications results were added by a dedicated adder. Again flexibility is kept in this design where the MAC could either

manipulate with four sample values or eight sample values. Four extra MUX were used to pass zero to the multiplier when number of taps is four. This helped to eliminate invalid data from the filter value. To conduct parallel operations by the multipliers, it is required to receive all the sample values simultaneously. This design performs filtering operation only after all the shift registers are filled with sample values..

III. FIR FILTER

Finite Impulse Response (FIR) filter design task is the recurring technical task in the development of digital signal processing products and systems [12]. The digital filtering part of this processor is designed such a way so that it can perform continuous filtering until user stops the filtering. The sample program is shown in Table II and Fig. 2 shows Functional simulation for FIR filter of this DSP processor.

TABLE II. FIR FILTER SAMPLE PROGRAM FOR PROPOSED PROCESSOR

Machine code (HEX)	Instruction	Hardware operation		Result
"A0000000" (FIR FILTER)	RF(0) ← mac_out	Instruction Register 1	Instruction Register 2	RF(0) ← mac_out
		Accum ← mac_out	RF(0) ← mac_Accum	

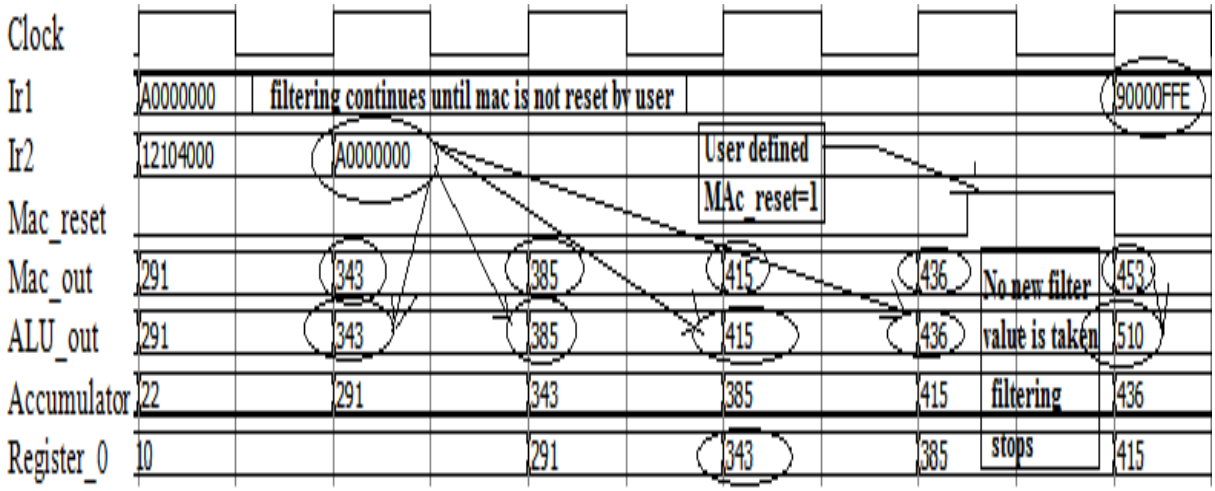

Figure 2. The functional simulation for FIR filter of proposed DSP processor

It can be seen from this simulation that, at cycle-3 the FIR filter output which is computed by external MAC, is passed by ALU. The output (values used are 232 and 253) is passed through the ALU to the accumulator register. At next active clock edge, this filter output is loaded to the register file. The FIR filtering is a continuous filtering of a real world signal and the filter operation requires the information to stop filtering. A user dependent signal Mac_reset is checked for this purpose. If the Mac_reset is found low then filtering is continued and no new instruction will be fetched. The simulation shows the next instruction fetched is a FIR filter instruction as user has not yet stopped the filter operation (Mac_reset = 0 at cycle 4). The reset signal is pressed before cycle-5 which stops the filtering and fetches a new instruction. After the reset signal is pressed, no filter output is loaded to the register which demonstrates the successful filtering operation of the proposed DSP processor.

IV. ARITHMETIC LOGIC UNIT (ALU)

The arithmetic logic unit (ALU) is an essential part of a computer processor. The most time consuming operations in ALU operation are addition and subtraction [4]. With a ripple adder design, the adder propagates the carry from the lowest bit to the highest sequentially. The most significant bit of the sum must wait for the sequential evaluation of the previous thirty one (31) 1-bit adders, which creates large propagation delay. Theoretically, the carry input without waiting for it to be generated by the previous 1-bit adder component. This can be done by applying some calculations on the two operands and the carry input to the least significant bit of the adder. Delay for this kind of adder will be in order of *log2N*, where *N* is the bit number of the operands (64 in this case), instead of *N* provided by the usual ripple adder [13]. Therefore, to increase the speed, a fast parallel adder, the Carry Look Ahead adder is used in this proposed design.

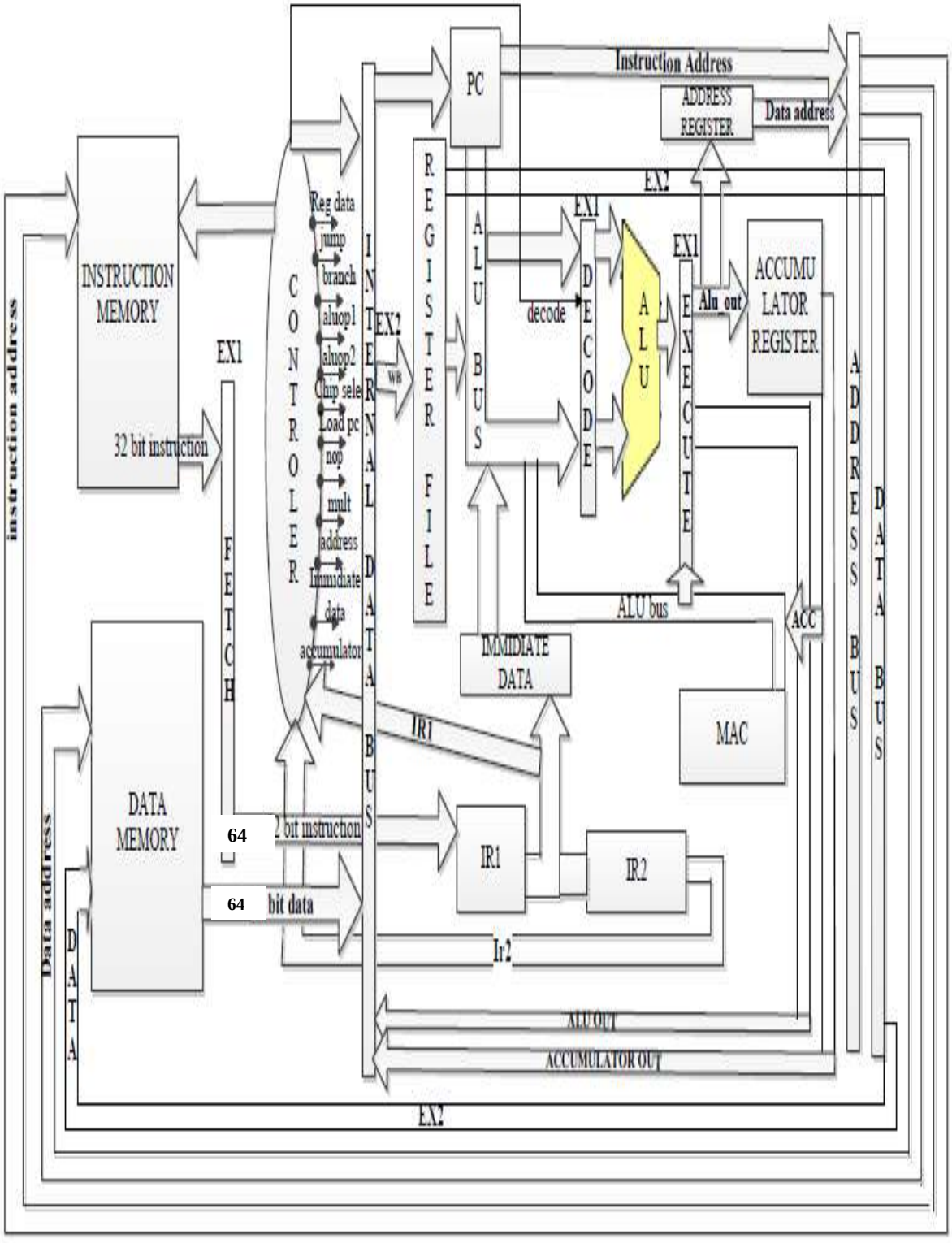

Figure 3. Block diagram of two stage of pipelining of proposed DSP Processor

V. MEMORY DESIGN

To perform fast filtering, the two stage pipelined architecture of this DSP processor needs to access both data and instruction memory simultaneously. To achieve higher performance in DSP operation and successful execution of other instructions, this design follows Harvard architecture which has separate instruction and data memories. For the safe design purpose or control over unwanted memory output, a tristate buffer is used in the output of data memory. The purpose is to keep the data bus unconnected when it is not accessed by the processor. This helps to save the calculation from unwanted data.

VI. TWO STAGE PIPELINING

Two stage pipelined architecture is one important feature of the design. In this design, both data paths include a single

pipeline with two stages: 1) Execution Stage1, and 2) Execution Stage2. The primary reason for separating the stages is to keep the system clock operates faster with less combinational delay. To enhance the speed, the general purpose data path is modified by adding some dedicated registers. For the load or store type operation, a special register is used which helped to complete any load or store type operation only in two clock cycles. These registers also help to execute all instructions of this processor within two clock cycles manipulating any hazard. The two stage pipelined diagram of proposed DSP processor is illustrated in Fig. 3.

VII. HAZARD HANDLING

This design is modeled with a hazard free pipelined architecture which could handle two instructions simultaneously as it is a two stage pipelined processor. Since two instructions work in parallel, the FSM is designed such a way that helps to detect hazards and can control the data path to complete computation by resolving the hazard. The FSM of this processor can manipulate both data and structural Hazards.

For the structural hazard, a MUX is used instead of ALU, at the second stage of pipelining. The reason of using MUX for this modeling is to save ALU calculation from unwanted data because a new instruction is using ALU at its first stage of calculation.

For the data hazards, the controller is designed for receiving the data direct from the internal data bus which keeps the most recent data fetched from memory or stored value of accumulator register. By taking the value from internal data bus, the new data is bypassed before decode and execution stage from accumulator register. This improvement saves 1 cycle as this is not taken from the register bank which needs 1 more cycle to update with the corresponding value. Fig. 4 shows the bypassing technique of the DSP processor.

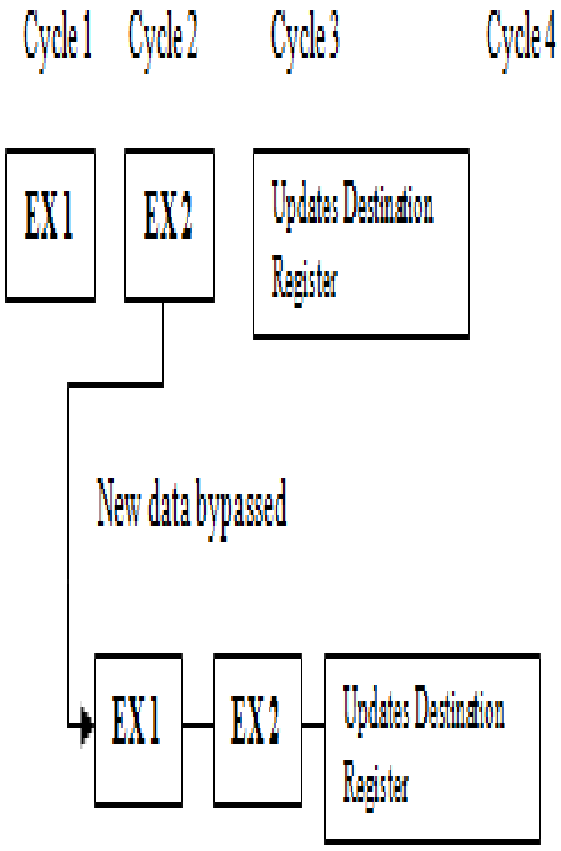

Figure 4. Bypassing technique for avoiding data hazard

The improvement for hazard free FSM can be identified by the comparison of the pipelined architecture with and without hazard handling capability.

An example is illustrated here for the comparison. For easier identification, some values (R1= 2, R2 = 5, R3= 8, R4= 9, R5= 4) are assumed for the instructions (R3 <- R1+ R2, R5 <- R3+ R4). When FSM is not handing hazard, R3 <- 7 and R5 <- 17 (adding previous value of R3 (8) and R4 (9)) since R3 needs one more cycle to update with the most recent value of addition (7) of R1 and R2 due to two stage of pipelining. This hazard is avoided by the FSM of proposed processor which updated R5 with the result of addition (16) of most recent value of R3 (7) and R4 (9). Table III is used here to illustrate the comparison.

TABLE III COMPARISON TABLE BETWEEN PIPELINE WITH AND WITHOUT HAZARD HANDLING CAPABILITY

Signal	Operation	Pipeline with hazard handling capability				Pipeline without hazard handling capability			
Clock	-	4	5	6	7	4	5	6	7
PC	-	3	4	5	6	3	4	5	6
Present State	-	2				2			
Instruction register 1 (IR1)	Desired operation	R3 <- R1+ R2	R5 <- R3+ R4	New inst	New inst	R3 <- R1+ R2	R5 <- R3+ R4	New instruction	New instruction
	Hardware operation	Accumulator <- R1 + R2	(Accumulator <- Accumulator + R4)	-	-	Accumulator <- R1 + R2	(Accumulator <- R3 + R4)	-	-
Instruction register 2 (IR2)	Desired operation	Old instruction	R3 <- R1+ R2	R5 <- R3+ R4	New inst	Old instruction	R3 <- R1+ R2	R5 <- R3+ R4	New instruction
	Hardware operation	-	R3 <- Accumulator	R5<- Accumulator	-	-	R3 <- Accumulator)	R5<- Accumulator	-
Accum	-	-	7	16	-	-	7	17	-
R1	-	2	2	2	2	2	2	2	2
R2	-	5	5	5	5	5	5	5	5
R3	-	8	8	7	7	8	8	7	7
R4	-	9	9	9	9	9	9	9	9
R5	-	4	4	4	16	4	4	4	17

VIII. PERFORMANCE ANALYSIS AND DISCUSSION

The simulation of the proposed 64-bit DSP processor is carried out by using Verilog HDL in Xilinx tool. The top module view and simulation results of the 64-bit DSP processor are shown in below figures.

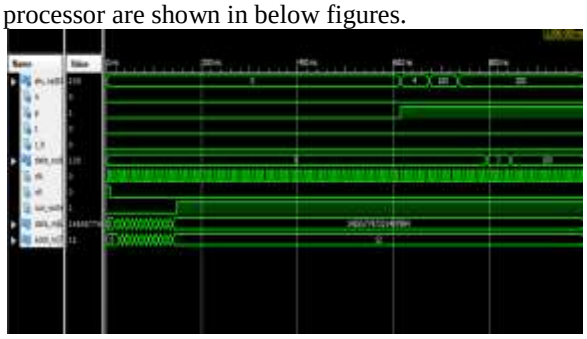

Figure 5. Simulation results of the 64-bit DSP processor

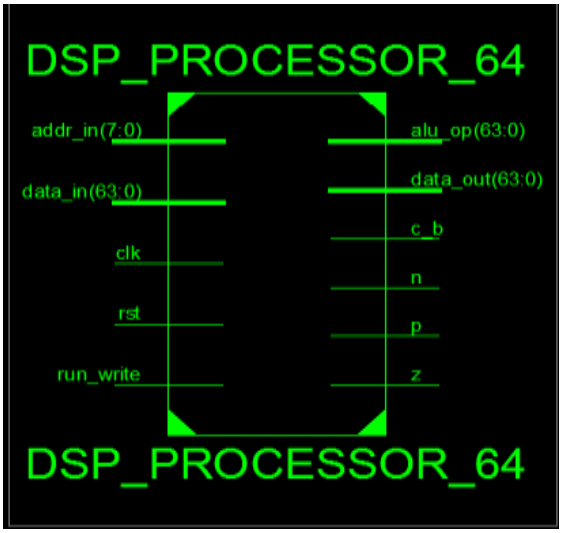

Figure 6. Top module view of 64-bit DSP processor

IX. CONCLUSIONS AND FUTURE WORK

In this paper, a pipelined DSP processor with reduced instructions set is illustrated for performance optimization. Primary focus of the design is to achieve better throughput and higher speed gain over the compared one (MUN DSP-2000). The design is simulated using Modelsim 6.5. Each system building component is synthesized using the Xilinx 8.2i. The improved performance of this processor is analyzed by comparing throughput with MUN DSP-2000

(another reduced instruction set processor). The comparison shows that a better throughput (12.06 MB/s) can be achieved with the new design. In addition to this, the maximum delay of the proposed design is also compared with existing system, and it is found that the new design consumes less delay in each system building components.

In the future, the design can be extended to perform more operations. Moreover, the calculation speed can be enhanced by providing support for floating point operations.

REFERENCES

[1] M. E. A. Ibrahim, M. Rupp , and H.A. H. Fahmy, "Power Estimation Methodology for VLIW Digital Signal Processors," in *Proc. ACSSC'08*, 2008, paper 169593.

[2] K. Anand and S. Gupta, "Designing Of Customized Digital Signal Processor" B.T. Thesis, Indian Institute of Technology, Delhi, May, 2007.

[3] Chattopadhyay, W. Ahmed, K. Karuri, D. Kammler, R. Leupers, G. Ascheid, H. Meyr, "Design Space Exploration of Partially Re-conFig.urable Embedded Processors," in *Proc. Design, Automation & Test in Europe Conference & Exhibition'07*, 2007, paper 10.1.1.86.3489, p.319.

[4] C. Li, L. Xiao, Q. Yu, P. Gillard and R. Venkatesan, "Design of a Pipelined DSP Processor - MUN DSP2000," in *Proc. NECEC'00, 2000*, paper 10.1.1.11.3142

[5] K. Karuri and R. Leupers, Application Analysis Tools for ASIP Design: Application Profiling and Instruction-set Customization, 1st ed., Springer, 2011.

[6] D. Skolnick and N. Levine, (1997), Analog devices. [Online]. Available: http://www.analog.com/library/analogDialogue/archives/31-1/DSP.html

[7] "Xilinx Spartan-3 FPGA Family Data Sheet," Product Specification, California, USA.

[8] C.-J. Chou, S. Mohanakrishnan and J. B. Evans, "FPGA implementation of digital filters," in *Proc. ICSPAT'93,* paper 10.1.1.57.857.

[9] J. Cong, B. Xiao, "mrFPGA: A Novel FPGA Architecture with Memristor-Based Reconfiguration," *ACM International Symp. Nanoscale Architectures*, 2011.

[10] J. Becker, M. Glesner, "A Parallel Dynamically Reconfigurable Architecture Designed for Flexible Application-Tailored Hardware/Software Systems in Future Mobile Communication", *The Journal of Supercomputing*, vol.19, no.1, 2001.

[11] T. Ferdous, "Design, Synthesis and FPGA-based Implementation of a 32- bit Pipelined Digital Signal Processor," *International Journal of Scientific and Engineering Research (IJSER)*, vol. 3, issue 7, 2012.

[12] J. Treichler, (2009) Retrieved from the Connexions. [Online]. Available: http://cnx.org/content/col10553/1.3/

[13] J.P.Uyemura, *Introduction to VLSI Circuits and Systems*, 1st ed., John Wiley & Sons, Inc. 2002.

[14] M. R.S. Balpande, M.R.S. Keote, "Design of FPGA based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor," in *Proc. ICCSNT'11*, 2011, paper 10.1109, p. 409.

[15] Altium PPC405A 32-bit RISC Processor, Altium, 2006

Author's Profile:

Jadapalli Pavani received her B.Tech degree in Electronics and Communication Engineering from Narayana Engineering College Gudur,Nellore District, affiliated to JNTU Anantapur.She is currently pursuing M.Tech VLSI in Audisankara college of Engineering and Technology, Gudur(Autonomous), SPSR Nellore (Dist), affiliated to JNTU Anantapur.

K.Dhanunjaya received his B.Tech Degree in Electronics and Communication Engineering from G. Pulla Reddy Engineering College, Kurnool, AP in 1998. He received his M.Tech in ECE from Jawaharlal Nehru Technological University, Anantpur. He has 14 years teaching experience, presently working as Professor and Head of the department of ECE, Audisankara College of Engineering and Technology (Autonomous), Affiliated to JNTU, Anantpur. Andhra Pradesh, India