ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of Pipelined 64-Set of Instruction Set High Performance Digital Signal Processor

R.Rajanikanth Reddy
Master of Technology VLSI Design,
TKR College of Engineering and Technology,
Medbowli, Meerpet, Saroornagar, Hyderabad.

Dr. D. Nageshwar Rao
Professor & HOD,
Department of ECE,
TKR College of Engineering and Technology,
Medbowli, Meerpet, Saroornagar, Hyderabad.

Abstract:

The goal of DSPs is usually to measure, filter and/or compress continuous real-world analog signals. Most general-purpose microprocessors can also execute digital signal processing algorithms successfully, but dedicated DSPs usually have better power efficiency thus they are more suitable in portable devices such as mobile phones because of power consumption constraints. DSPs often use special memory architectures that are able to fetch multiple data and/or instructions at the same time. A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable".

The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). In this paper, 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.

Keywords: DSP, FPGA, MAC.

1.INTRODUCTION:

Digital signal processing (DSP) is the mathematical manipulation of an information signal to modify or improve it in some way. It is characterized by the representation of discrete time, discrete frequency, or other discrete domain signals by a sequence of numbers or symbols and the processing of these signals. The goal of DSP is usually to measure, filter and/or compress continuous real-world analog signals.

Usually, the first step is conversion of the signal from an analog to a digital form, by sampling and then digitizing it using an analog-to-digital converter (ADC), which turns the analog signal into a stream of discrete digital values. Often, however, the required output signal is also analog, which requires a digital-to-analog converter (DAC).

Even if this process is more complex than analog processing and has a discrete value range, the application of computational power to signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression.

Digital signal processing and analog signal processing are subfields of signal processing. DSP applications include audio and speech signal processing, sonar and radar signal processing, sensor array processing, spectral estimation, statistical signal processing, digital image processing, signal processing for communications, control of systems, biomedical signal processing, seismic data processing, among others.

DSP algorithms have long been run on standard computers, as well as on specialized processors called digital signal processors, and on purpose-built hardware such as application-specific integrated circuit (ASICs).

Currently, there are additional technologies used for digital signal processing including more powerful general purpose microprocessors, field-programmable gate arrays (FPGAs), digital signal controllers (mostly for industrial applications such as motor control), and stream processors, among others.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

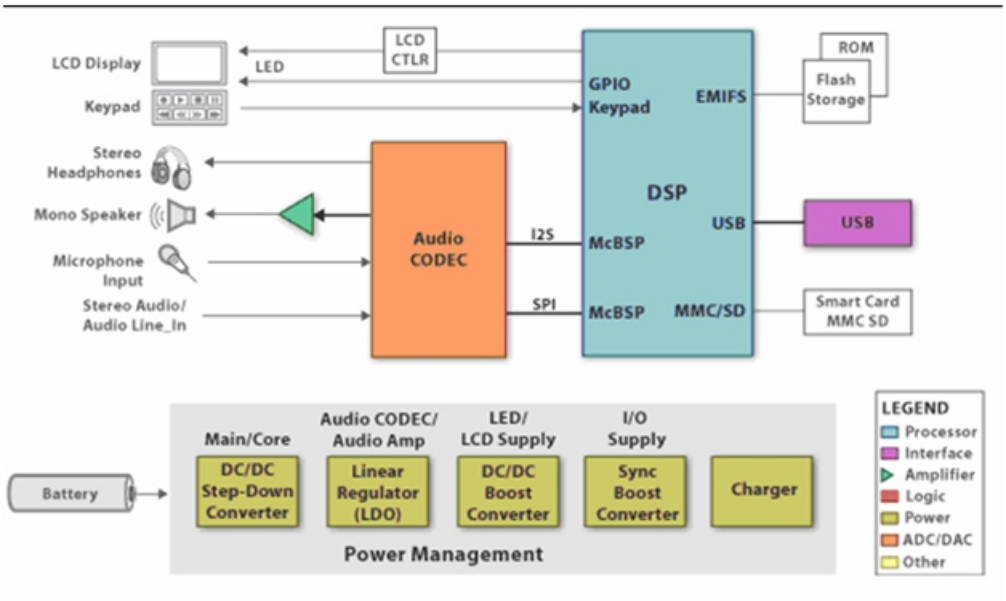

Figure 1 Basic block diagram of a DSP processor

2.INSTRUCTION PIPELINING:

Instruction pipelining has become an important element to achieve high DSP performance. It consists of dividing the execution of instructions into different stages and executing the different instructions in parallel stages. The net result is an increased throughput of the instruction execution. The whole process can be compared to a factory assembly line, which produces cars for instance: more than one car is in the assembly line at the same moment, at different stages of assembly. This provides a production higher than the case where only one car at a time is produced, where many specialized crews are idle waiting for the next car to require their work.

The basic pipelining stage is divided in to following three stages.

1. Fetch. The DSP calculates the address of the next instruction to execute and retrieve the op code, i.e., the binary word containing the operands and the operation to be carried out on them.
2. Decode. The op-code is interpreted and sent to the corresponding functional unit. The
Instruction is interpreted and the operands are retrieved.
3. Execute. The instruction is executed and the results are written onto the registers.

Basic pipelining stages Action

1.Fetch

» Generate program fetch address
» Read op-code

2.Decode

» Route op-code to functional unit
» Decode instruction
» Read operands

3.Execute

» Execute instruction
» Write results back to registers

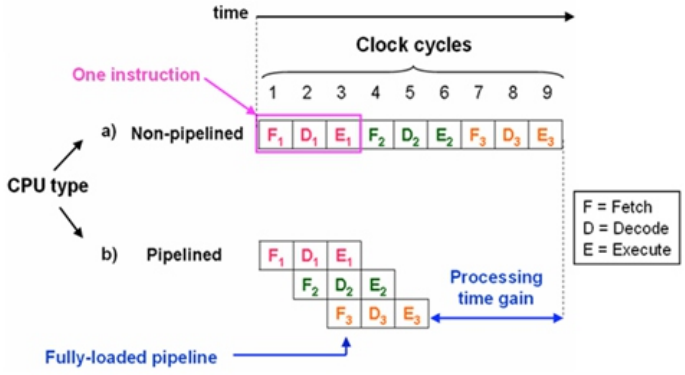

Figure 2 Pipelining architecture of an instruction set

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

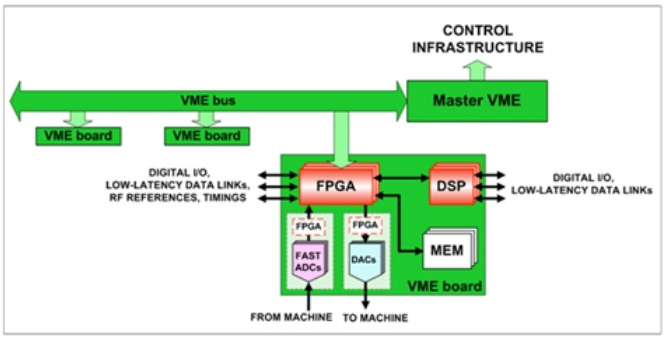

Figure 3 Typical DSP Core Design Using FPGA

3. PROPOSED METHOD:

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.

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 (32 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.

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.

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.

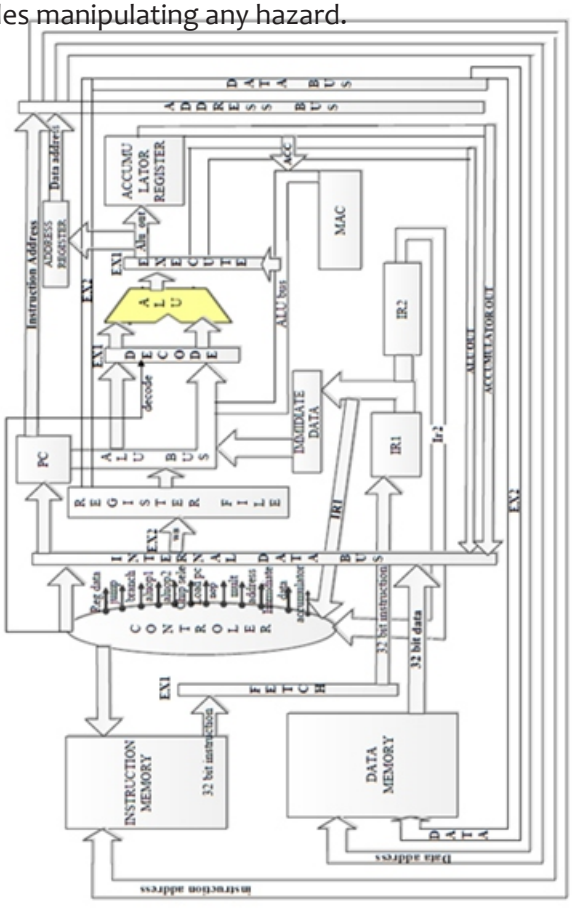

Figure 4 Basic block diagram of two stage pipelining

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Above diagram illustrate the 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.

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.

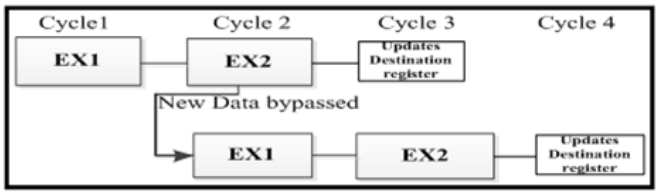

Figure 5 Bypassing technique for avoiding data hazard

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

4.SIMULATION RESULTS:

The 64 bit DSP processor is designed by using verilog and the internal blocks are verified by simulating all the blocks and the corresponding results are shown below figures.

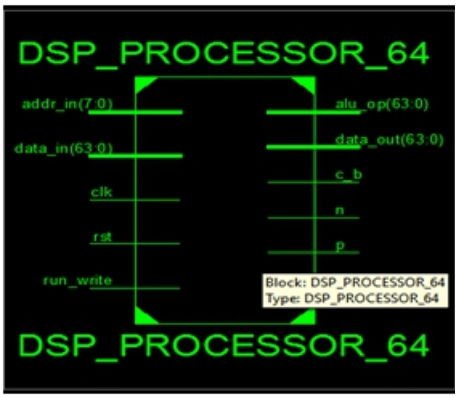

Figure 6. DSP module.

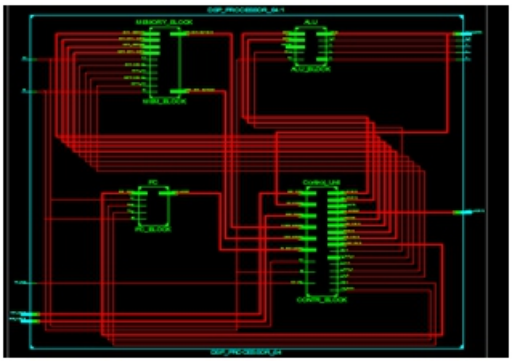

Figure 7 Internal blocks of DSP module.

The above two diagrams indicates the rtl schematic view of fir fitler architecture with mac design which enables us to enhance the filter architecture in less number of clock cycles with much lesser delay overhead while comparing with general techniques of filter designes. This schematic consists of the blocks namely alu,address and data memory blocks along with mac design which depicts a generalized view of a DSP processor architecture.

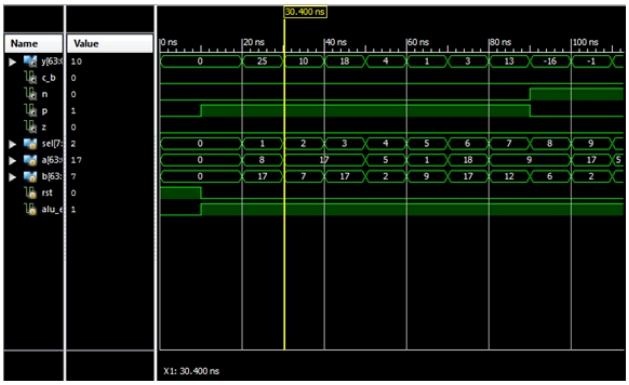

Figure 8 Simulated waveforms for ALU block

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The above wave forms represents the arithmetic and logic block which consists of general arithmetic operations needed by processor to process data namely addition,subtraction ,multiplication, division etc..

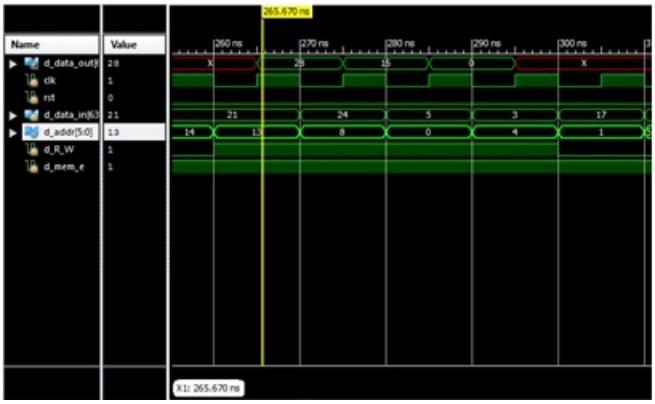

Figure 9 simulated waveforms for data memory

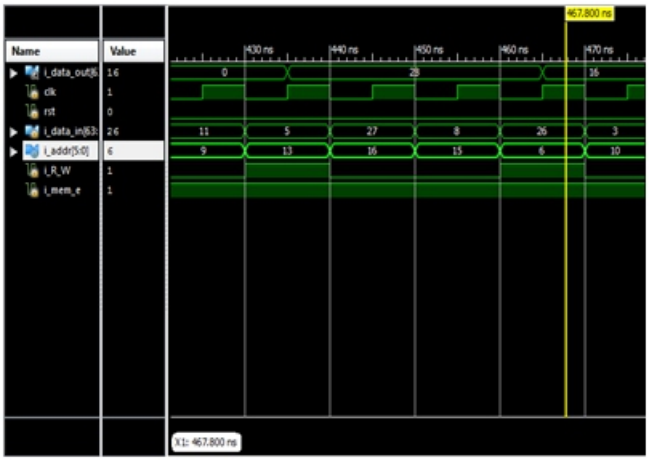

Figure 10 simulated waveforms for instruction memory

The above program describes the memory allocation of our given data in which we can observe the input data is been stored in respective addresses accordingly with enable and mode signals.Here based on the signal i_r_w signal,the data will be read by memory or write data to other temporary memory module.

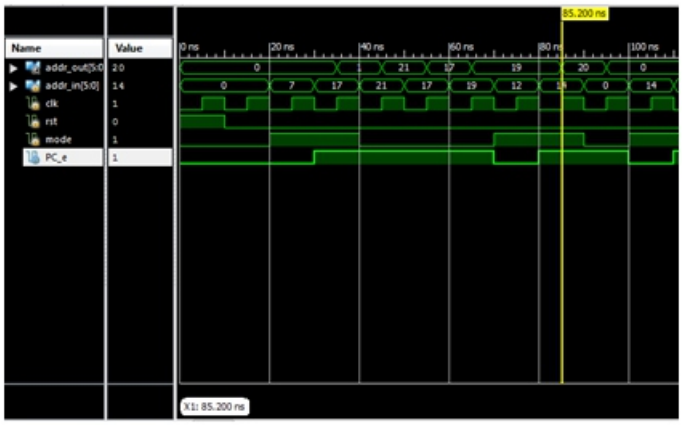

Figure 11 simulated waveforms for program counter

The program counter counts up the operations to be done by alu and it will do the counting operations of data lengths in order to send correct length of data for every time to and from the alu block.

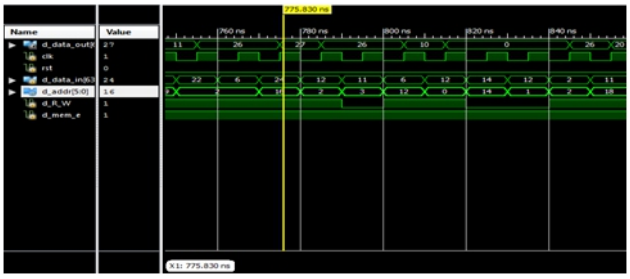

Figure 12 final output of the DSP

This wave form represents the finalized output of our architecture in which we can observe the data read write operations along with arithmetic and logic operations done by ALU block based on the d_addr and d_mem_en signals.

5.CONCLUSION:

32-bit Processor core has been design and implemented in hardware on Xilinx 3E FPGA. The design has been achieved using VHDL and simulated with Xilinx 9.2i. Spartan 2E development board has been used for the hardware part. Most of the goals were achieved and simulation shows that the processor is working perfectly, Future work will be added by increasing the number of instructions and make a pipelined design with less clock cycles per instruction and more improvement can be added in the future work.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

REFERENCES:

[1] Tasnim Ferdous, Design and FPGA-based Implementation of a High Performance 32-bit DSP Processor, Computer and Information Technology (ICCIT), 2012 15th International Conference

[2] R. Uma, "Design and Performance Analysis of 8-bit RISC processor using Xilinx Tool", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol-2, Mar-Apr-2012.

[3] J. Poornima, G.V.Ganesh, M. Jyothi, M. Shanti and A.Jhansi Rani,"Design and implementation of pipelined 32-bit Advanced RISC processor for various D.S.P Applications", Proceedings of International Journal of Computer Science and Information Technology, ISSN: 3208-3213, Vol-3(1), June-2012.

[4] Xiao Li, Longwei Ji, Bo Shen, Wenhong Li, Quianling Zhang, " VLSI implementation of a High-performance 32-bit RISC microprocessor", Communications, Circuits and Systems and West Sino Expositions, IEEE , International Conference, ISSN:1458- 1461,Vol-2,2002.

[5]http://elearning.vtu.ac.in/12/enotes/Adv_Com_Arch/Pipeline/Unit2-KGM.pdf.

[6] Asmita Haveliya "Design and simulation of 32-point FFT using Radix-2 Algorithm for FPGA Implementation" IEEE, 167-171, 2012.

[7]The pipelined RISC-16 ENEE 446: Digital Computer Deisgn, Fall 2000 by Prof. Bruce Jacob.

[8] Preetam Bhosle, Hari Krishna Moorthy," FPGA Implementation of Low Power Pipelined 32-bit RISC Processor", Proceedings of International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Vol-1, Issue-3, August 2012.

[9]Charles H.Roth Jr, "Deisgn Systems Design using VHDL", Prentice Hall 2nd Edition, 2000.

About Author's:

R.Rajanikanth Reddy
Master of Technology VLSI Design,
TKR College of Engineering and Technology,
Medbowli, Meerpet, Saroornagar, Hyderabad.

Dr. D. Nageshwar Rao
Professor & HOD,
Department of ECE,
TKR College of Engineering and Technology,
Medbowli, Meerpet, Saroornagar, Hyderabad.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal