

A Noval Programmable ALU Architecture for DLX Processor

B.Geetha Rani, M.Tech Student, Sri Venkateshwara College of Engineering, Email:balakageetharani24@gmail.com¹
M.Chandra Sekhar Reddy, Assistant Professor, Sri Venkateshwara College of Engineering,
Email: Chandra.2030@ gmail.com²

ABSTRACT: In today's fast paced growing embedded industry, design's complexity has increased and hence designers must have extensive knowledge of various design and modeling aspects. In this paper these features and approach has been elaborated using the example of DLX ALU at a basic level. The ALU is designed using Verilog HDL and a model has been extracted with the help Xilinx System Generator. The proposed methodology mainly focuses on the design of 32-bit pipelined RISC processor based on the DLX architecture to perform fast convolution and correlation operations.

Keywords- DLX; RISC processor; Verilog; FPGA; HDL

1 INTRODUCTION

Nowadays, computers and mobile phones is indispensable tool for most of everyday activities. This places an increasing burden on the embedded microprocessor to provide high performance while retaining low power consumption and small die size, which increase the complexity of the device .

However, as products grow in complexity more processing power is required while the expectation on battery life also increases. With the rapid development of silicon technology RISC includes extensions to RISC concepts that help achieve given levels of performance at significantly lower cost than other systems. The main features of RISC processor are the instruction set can be hardwired to speed instruction execution. With reconfigurable devices such as FPGAs based core design any set of task can be configured. It sustains any system level change without costly hardware replacement and thus the design process is very fast and cost effective.

The DLX Microprocessor is a RISC Processor designed by John L. Hennessy and David A.

Patterson, the principal designers of the MIPS and the Berkeley RISC designs (respectively), the two benchmark examples of RISC design. The DLX is essentially

simplified MIPS with simple 32-bit load/store architecture. The DLX design is widely used in university-level computer architecture courses to help students to get knowledge about the RISC processor in terms of instruction encoding & decoding, pipelined operations, functions of each component within the processor, types of instruction set and operations etc. It is based on classic Harvard architecture of Separate instruction and data memories to allow simultaneous instruction fetching and data memory transactions. It supports 3 types of instruction formats (I, R & J) & uses immediate & displacement addressing modes. In non-pipelined execution, each instruction takes 5-clock cycles.

II ARCHITECTURE AND DESIGN

In RISC architecture, the instruction set of processor is simplified to reduce the execution time. It uses small and highly optimized set of instructions which are generally register to register operations. The speed of the execution is increased by using smaller number of instructions .This uses pipeline technique for execution of any instruction.

The figure shown below is the architecture of RISC processor, which uses separate instruction and data caches and their access paths also different. There is one instruction per machine cycle in RISC processor

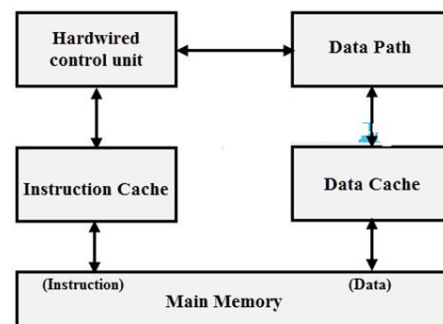
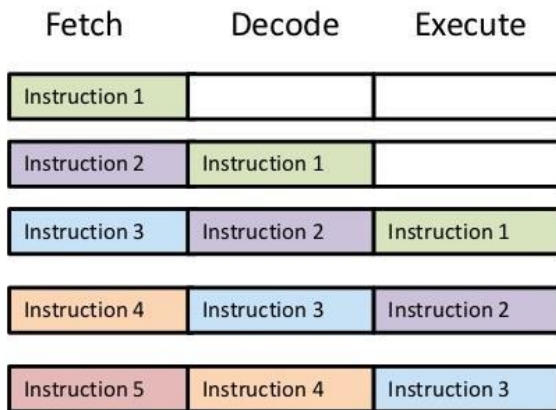


Fig 1; Risc processor

The pipelining technique allows the processor to work on different steps of instruction like fetch, decode and execute instructions at the same time. Below is image showing execution of instructions in pipelining technique. Generally, execution of second instruction is started, only after the completion of the first instruction. But in pipeline technique, each instruction is executed in number of stages simultaneously. When the first stage of first instruction is completed, next instruction enters into the first stage. This process continues until all the instructions are executed.



The architecture of 16 bit RISC processor has been shown in the figure. It comprises of Control unit, general purpose register, ALU, Barrel shifter, universal shift register and accumulator. The control unit consists of two registers i.e instruction register and instruction decoder. Instruction and data are fetched sequentially in order to reduce the latency in the machine cycle. Pipeline structure has been incorporated that further utilizes three execution cycle fetch, decode and execute. This pipeline structure helps in enhancing the speed of operation. In fetch cycle, instruction and relevant data are inferred from the memory while in decode cycle, instruction and data drawn from the memory are bifurcated to activate component and data path for execution and in the execution cycle instruction is executed, data is manipulated and result is stored in the accumulator.

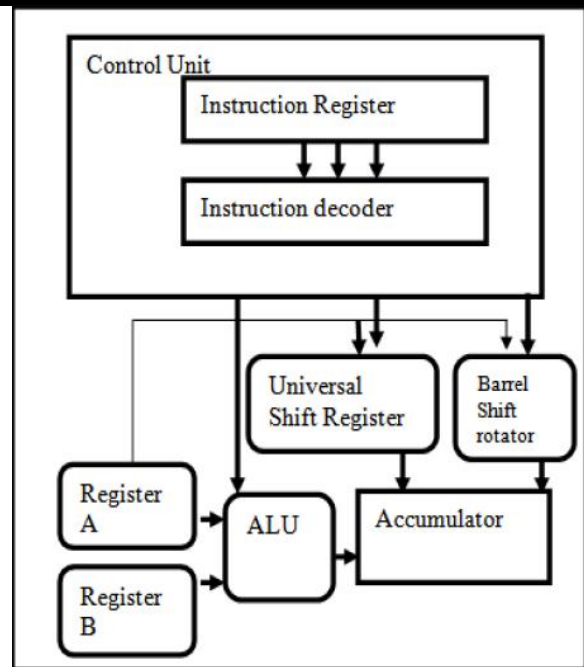


Fig 2; Architecture of RISC processor

The control unit accept the opcode and generate the signal that triggers the components and data path to work accordingly and perform the desired function. The control unit has two instruction decoders. These two decoders decode the instruction bits and direct the signal to either into ALU, universal shift register or barrel shift rotator. The operands are received from register A or register B. Upon receiving the operands from registers and the decoded instruction bits arithmetic and logical unit perform arithmetic and logical functions. Universal shift register and barrel shift rotator receives the input from register A and depending upon the decoded

information perform the desired operation of either shifting or rotation and the result is stored in the accumulator register.

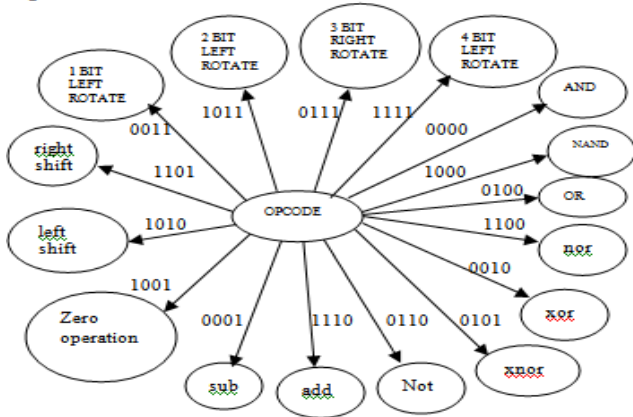


Fig 3: State diagram

A) Memory:

- Memory of 32 locations depth and 32 bit width
- Receives address from operand1 which is give to the instructions decoder
- If r/w is high it will assign the data to DATA_OUT
- The memory block will get activated only when CHIP_SE is high.

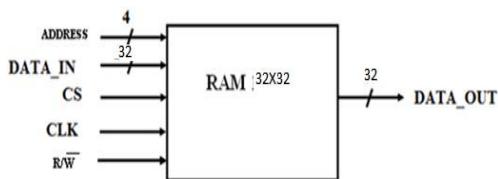


Fig 4: Memory block

B) ALU

Arithmetic and logical unit is a digital circuit that performs arithmetic and logical operations. The proposed design performs seven logical functions and two arithmetic functions. The logical operations to be executed are AND, NAND, OR, NOR, XOR, XNOR and NOT while two logical operations are performed Addition and Subtraction. ALU will receive instruction bits from control unit and will execute the desired operation. For example, if input to control unit is 0000, the decoded bits will be 10000000 and after receiving the instruction bits from the decoder AND operation is performed by ALU according to the operands from register A and register B.

ALU Instructions:

Opcode	CIN	Operation
0000	0	AND
0001	0	OR
0010	0	XOR
0011	0	NOT
0100	0	NAND
0101	0	NOR
0110	0	XNOR
0111	0	LOGICAL LEFT SHIFT OF DATA[OPERAND_1]
1000	0	UNSIGNED ADDITION
1000	1	UNSIGNED ADDITION with CARRY
1001	1	UNSIGNED SUBTRACTION
1001	0	UNSIGNED SUBTRACTION with BORROW
1010	0	UNSIGNED MULTIPLICATION
1011	0	2'S COMPLEMENT OF DATA[OPERAND_1]
1100	0	SIGNED ADDITION
1100	1	SIGNED ADDITION WITH CARRY
1101	1	SIGNED SUBTRACTION
1101	0	SIGNED SUBTRACTION with BORROW
1110	0	SIGNED MULTIPLICATION
1111	0	ARITHMETIC RIGHT SHIFT OF DATA[OPERAND_1]

C)Control Unit:

The control unit is based on state diagram as depicted in the figure. The state machine performs the functions of arithmetic, logical, shifting and rotating functions. If bit instruction is 0100 then OR operation is performed as soon as next instruction is received then appropriate operation is performed. The control unit consists of two decoders. The first decoder performs arithmetic and logical function and the second decoder performs shifting and rotating operations.

D)Barrel Shifter:

Barrel shifter is shown in the figure. It is a digital circuit that shifts the number of bits by specified times. It will receive the decoded instruction bits from the second instruction decoder inside the control unit and performs the desired operation depending upon operand from register A and select lines.

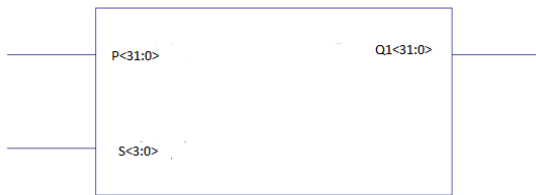


Fig 5: Barrel shifter

E) Universal Shift Register

The architecture is shown in the figure. This architecture performs four main functions as follows loading the value, left shift and right shift and no change. If s4 and s5 both are low while z is equal to 01000000 the value is loaded. If s4 is low and s5 is high with decoded output z as 00100000 left shift operation is performed.

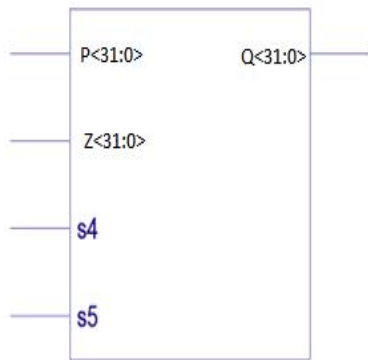


Fig 6: universal shift register

F) Accumulator Register:

Accumulator register top block is shown in the figure .The result from ALU or universal shift register or barrel rotator is stored in the accumulator register. If reset is set to high then accumulator register is cleared otherwise 8 bit result is stored in the accumulator register.



III DLX PROCESSOR

The proposed method mainly concentrates on the design of 32-bit pipelined RISC processor based on the DLX architecture to perform fast convolution and correlation operations, which is most required for almost all signal processing applications. The processor is designed to handle both integer and IEEE-754 floating point data formats making it effective for any application requiring floating point operations, namely graphics and scientific. Additionally, many modern multimedia applications requires the use of concurrent data calculations, which it can handle smoothly because of three pipeline stages. The processor design follows classic Harvard architecture of separate instruction and data memories to allow simultaneous instruction fetching and data memory transactions. The processor is designed by a powerful and flexible hardware descriptive language VERILOG. VERILOG model of the DLX microprocessor consists of various functional units which were created as components. These components were coded as separate modules which were composed, simulated and then instantiated in the main entity. This microprocessor contains the following functional modules as shown in fig.

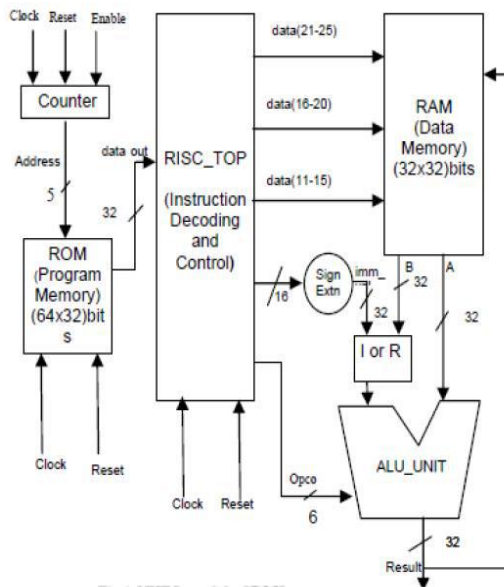


Fig 7: Model of DLX processor

Implementation of DLX Processor

Assembler: This module converts the assembly code written in text editor into equivalent binary instructions by following dlx instruction formats.

Counter Unit: Functions as instruction fetching module. Once it fetches one instruction from ROM the program counter is automatically incremented to fetch further instructions from the instruction memory.

Instruction decoding and control unit: Instruction fetched is decoded based on I-type or R-type format. The controller then fetches operands from the RAM(data memory) if data is reside in register specified in register fields of the instruction otherwise 16- bit immediate data is sign extended to make 32- bit data and issues command to ALU as per the opcode decoded by decoder.

ROM(Program Memory): General purpose ROM memory module for instruction storage, in which instructions are downloaded from the in binary form by following the DLX instruction format.

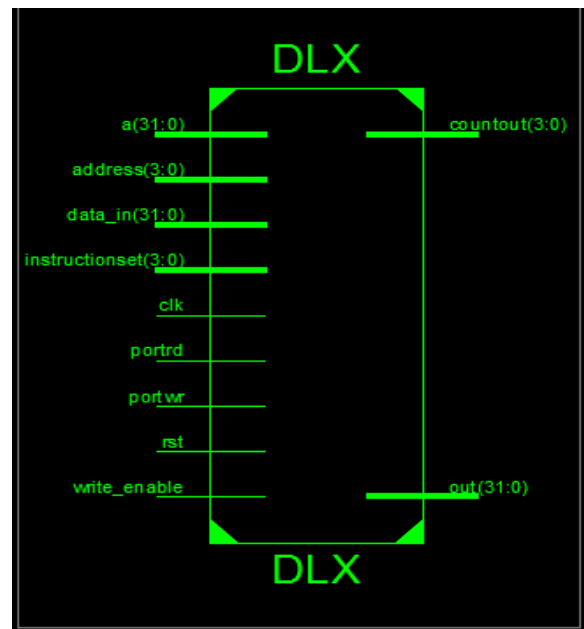
RAM (Data Memory): Data memory is also called register file that contains all the 32-bit general purpose registers and

32-bit floating point registers of the microprocessor. The register file is not containing any reserved registers, such as the PC, the status register, or other special registers so that all register can be used for all operation.

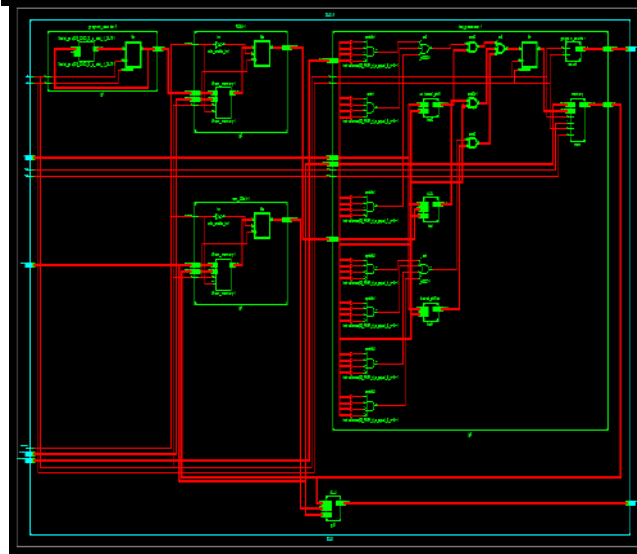
ALU: Arithmetic and logic unit, performs particular operation on the given operators and operands. Various command operations that can be performed are Addition, Subtraction, AND, OR, XOR, logical comparisons, shift (left / right), divide, multiply Increment, decrement etc.

1V RESULTS

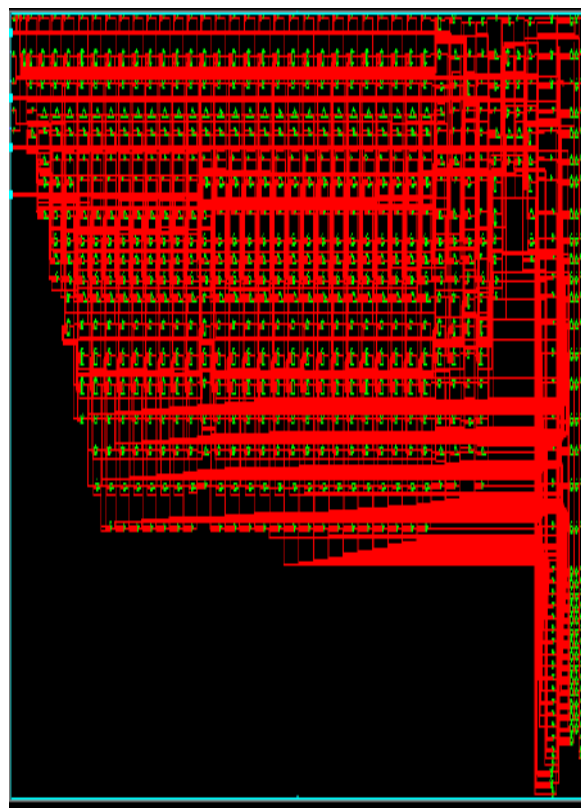
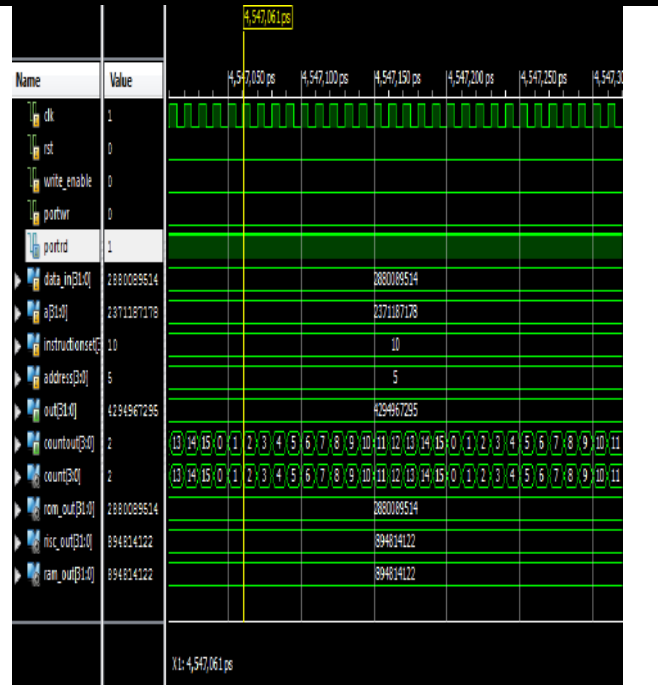
BLOCK DIAGRAM



RTL SCHEMATIC



TECHNOLOGY SCHEMATIC



SIMULATION RESULTS

V CONCLUSION

The primary objective of coding DLX processor using Verilog HDL language was successfully completed. The designed architecture is examined by running both integer and floating point instruction. Finally the correlation and convolution algorithms were written and results are obtained by simulation as well as by realizing the code on vertex 7 FPGA board. On future the performance of the proposed design is to be embedded for signal processing applications involving convolution and correlation, optimizing its speed of operation.

VI REFERENCES

[1] Indu, Arun Kumar, “Design of Low Power Pipelined RISC Processor”, International Journal of Advanced Research in Electrical & electronics & instrumentation Engineering, vol.2, no.3, pp.3747-3756 , August 2013.
 [2] Priyanka Trivedi , Rajan Prasad Tripathi “low Power pipelined RISC processor: A Review,” IJSRD vol.2, no.4, pp. 526-528, July 2014.
 [3] Jagrit Kathuria, M. Ayoubkhan, Arti Noor, “A Review of Clock Gating Techniques”, MIT International Journal of Electronics and Communication Engineering vol 1,no. 2, August 2011.

[4] B. Ramkumar and Harish M Kittur “Low-Power and Area-Efficient Carry Select Adder” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no.2, 2012, pp. 371-75.

[5] R.uma , “ Design and Performance analysis of 8 bit RISC Processor Using Xilinx Tool”, International Journal of Engineering Research and Application, vol.2, no.2,pp. 53-58, April 2012.

[6] Samiappa Sakthikumaran ,S. Salivahanan,V.S ,kanchan. Bhaaskaran, “16 bit RISC Processor Design For Convolution Application ” IEEE - International. Conference on Recent Trends in Information technology pp.394 - 397, June 2011.

[7] Li Li and Ken Choi “SeSCG: Selective Sequential Clock Gating for Ultra - low-Power Multimedia Mobile Processor Design,” IEEE EIT Conference, May 2010.

[8] Arora H,Gupta A ,Singhai ,R, Purwar D,“ Design space Exploration of RISC Architecture Using Retargetability” IEEE , pp. 1-3 ,Jan 2015.

[9] Geun-young Jeong, Ju-sung Park , “Design of 32-BIT RISC Processor and Efficient Verification ” Proceeding -of the 7th Korea - Russia International Symposium, Korws ,2003.

[10] Hai Li, Swarup Bhunia, Yiran Chen, Kaushik Roy, “DCG: Deterministic Clock - Gating for Low - Power Microprocessor Design ”, IEEE Trans. On VLSI Systems, vol. 12, no. 3, March 2004.

[11] J mirald.”The future of high performance COTS signal processing hybrid FPGA/DSP architecture, the optimal solution in DSPFPGA.com resorce guide-2006

[12] Z.Guo.W Najjar F Vahid and K.Vissers. “ A antitative analysis of the speedup factors FPGAs over processors, in ACM/SIGDA International Symposium on Field Programmable Gate arraya_FPGA Monterey C.A22-24 Febraury 2004 vol 12 pp 162-170

[13] Patterson D Hennessy,J.”Computer Architecture A Quantitative approach” Morgan Kaufmann Publishers 1996, in Second edition.

[14] Patterson D Hennessy,J. “Computer Architecture A Quantitative approach” Morgan Kaufmann Publishers 1996, in Third edition.

[15] Ellard et al. “On the design of a new CPU architecture for pedagogical purposes”, Proceedings of the 2002

workshop on Computer architecture education: Held in conjunction with the 29th International Symposium on Computer Architecture, May 2002.