International Journal of Computer and Communication Technology

Volume 7 | Issue 1 Article 16

January 2016

A NOVEL IMPLEMENTATION OF 32-BIT VLIW-MISC PROCESSOR ON FPGA

M. KAMARAJU
Dept. of E.C.E, Gudlavalleru Engineering College, INDIA, madduraju@yahoo.com

M. ALEKHYA
Dept. of E.C.E, Gudlavalleru Engineering College, INDIA, alekhya.madivada@gmail.com

K.LAL KISHORE
JNTUA, Anantapur, lalkishorek@yahoo.com

A NOVEL IMPLEMENTATION OF 32-BIT VLIW-MISC PROCESSOR ON FPGA

M.KAMARAJU[1] M. ALEKHYA[2] & K.LAL KISHORE[3]

[1,2]Dept. of E.C.E, Gudlavalleru Engineering College, INDIA
[3]JNTUA, Anantapur
E-mail: madduraju@yahoo.com, alekhya.madivada@gmail.com, lalkishorek@yahoo.com

Abstract: The main objective of this work is to implement a 32-bit pipelined RISC processor without interlocking stages. It is developed by S.I.M.E (Single Instruction Multiple Execution) that is with single instruction scheme more executions can be done and is based on VLIW(Very Long Instruction Word) architecture processing is an optimal choice in the attempt to obtain high performance level in Embedded Systems. In VLIW based architecture, the effectiveness of the processor depends on the ability of compilers to provide sufficient instruction level parallelism (ILP). The processor has been designed with VHDL, synthesized using Xilinx tool.

Keywords: *MIPS, RISC, VLIW, SIME.*

I.INTRODUCTION

The last two decades have been a golden era of clocked architectures and circuit design. The design of embedded processors typically targets low cost and low power consumption. Greater functionality in less space and reduced cost are the main considerations of today's systems. In this work MIPS processing is taken into consideration. Full name of MIPS is Microcomputer without Interlocked Pipelined Stages [1]. In order to speed up processing, pipeline architecture is to be adopted. The pipelining can be performed by splitting the processing of single cycle architecture into a series of independent stages and inserting registers in between stages. Direct pipelining results five stage named IF (Instruction Fetch) stage, ID (instruction decode) stage, EX (execute) stage, MEM (memory) stage and WB (write back) stage [7]. Reduced Instruction Set Computer (RISC) mainly concentrates on reducing the number and complexity of instructions in the machine [8,9]. Field Programmable Gate Arrays (FPGA) is growing faster for being designing with cost reduction compared to ASIC design [10]. In addition, each generation of FPGAs delivers higher performance.

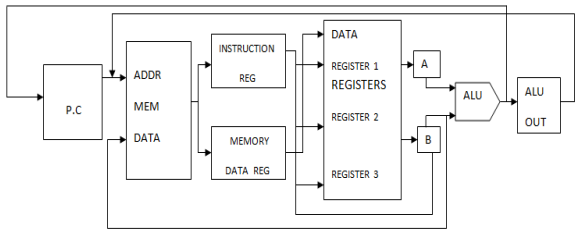

Fig 1: Block diagram of general processor.

The instruction is fetched from physical memory, RAM. Program counter (PC) is used to fetch the address of the instruction. This register is visible to Control unit of CPU but not visible to programmer. The instruction is copied to Instruction Register from memory. If the instruction is ADD is considered, the addition operation is performed on corresponding registers. This is performed by the unit ALU. For MIPS kind of RISC machine, memory access occurs for the load and store instructions. For the other instructions no operation is performed by ALU. Here data memory is accessed. The result of the operation being performed by ALU, is written to appropriate register in the register file, which is a hardware that stores all the registers.

Even though the RISC processors are having more advantages than CISC processors, it suffers from hazards that occurs in pipelining.

1. Data hazards: Data hazards occurs when instruction that exhibits data dependency that is modifying the data in different stages of pipelining.
2. Pipeline interlocks: In this, the pipeline divided into two parts called instruction fetch and instruction execute.

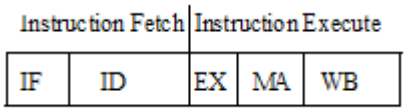

Fig 2: Sub division of instruction pipelining

The hardware to detect a data hazard and stall a pipeline until the hazard is cleared is called a pipeline interlock. In order to speed up the operation of MIPS processor, VLIW is the technique adopted by sub dividing the pipeline process as fetching and execution stages.

3. ALU forwarding: Here the result of ALU is used directly, without having to wait for the result to be written back to the register file.

For this there is no need to forward the result of the ALU directly backing to the ALU. Before the

result gets written to the destination register, it is return to the intermediate register. It is need to forward the contents of immediate results back into input of the ALU.

To overcome these hazards, MIPS is introduced. MIPS is a RISC micro processor defines, 32 bit General purpose Registers.

II. 32-bit VLIW-MISC PROCESSOR

In this proposed technique, first block that is program counter is used to generate the address of the next instruction

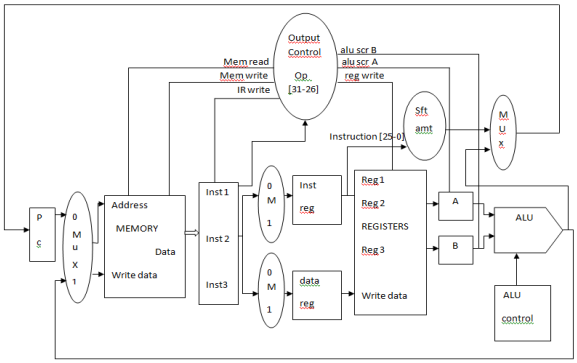

Fig 3: Block diagram of 32-bit VLIW-MISC processor.

For the fetched instruction, data and address are stored in the memory. Multiplexer is used to select the corresponding instruction format. In response to the instruction format fetched data is sub divided into instruction and data with separate memory access. The data is stored in temporary registers. First 0-25 bits are used for data and remaining 26-31 bits are used to select opcode selection. This op code is used for the operation which is performed on the data. Blocks A and B are used to apply data to ALU unit, in which execution is performing. Multiplexers are used to select four 32-bit data values. ALU control unit produces the control signals that are required for program execution. Then the result is write back to the memory. Next instruction address is generated by program memory. By adopting VLIW, at a time two instructions are executed with same operation or combination of operations for two different or same data values.

III. INSTRUCTION SET (VLIW-MISC)

In MISC processor, registers are hardwired and contains the value zero. MIPS only has three instruction types: I-type is used for load and store instructions, R-type is used for Arthematic instructions and J-type is used for jump instructions.

R-type is short for “register type”. The format of an R-type instruction shown in fig 4.

B31-26	B25-21	B20-16	B15-11	B10-6	B5-0
Opcode	Reg s	Reg t	Reg d	Shift amount	Function

Fig 4: R-type instruction format.

I-type is short for “immediate type”. The format of an I-type instruction shown in fig 5.

B31-26	B25-21	B20-16	B15-0
opcode	Reg s	Reg t	Immediate

Fig 5: I-type instruction format.

j-type is short for “Jump type”. The format of an J-type instruction shown in fig 6.

B31-26	B25-0
opcode	Target

Fig 6: J-type instruction format.

IV.RESULTS

VLIW-MISC processor is simulated by using ISE simulator. Programming is done in VHDL, which is effective and user friendly. This simulation is possible for all arithmetic and logical instructions. Also possible for load and store, jump and branch instruction.

Fig 7: Signal Diagram of VLIW-MISC processor

Signal diagram of VLIW-MISC processor description shown in fig 7. The signals are *clk* : This signal provides clock for the given inputs. *rst*: This signal resets the inputs when ever not required.*opcode*: operation applied for data. *instruction 1 &2[31:0]*: Input data values.*fop 1&2[31:0]*: output data values.*jop 1&2[31:0]*: jump output values.

Fig.8 Shows simulation result of addition of two set of data values using R type instruction format.In this format, data values are provided for instruction1 are, mode selection bit as 01, opcode given is addition, inputs are 8 and 9 in memory locations of 7 and 5. And for instruction 2 inputs are 10 and 4 in locations of 3 and 6. So the outputs obtained for these two instructions are 17 and 14.

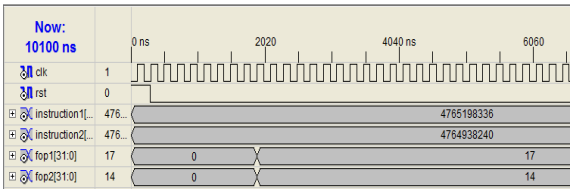

Fig.8 : Simulation result of addition of two set of datas using R type instruction format.

Fig.9 shows simulation result of NAND operation for first set of data value using I type format, addition operation for second set of datas using R type format.

In this format, data values are provided for instruction 1 are, mode selection bit as 10, opcode given for nand operation, input is 12 in memory locations of 6 and immediate value of 2084. And for instruction 2 opcode is addition, input is 8 in location 1 and immediate value of 660. Outputs obtained are 0 for both 1 inputs for nand,815 for addition opcode.

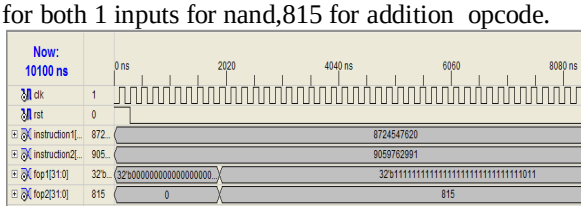

Fig.9: Simulation result of NAND operation for first set of data value using I format, addition operation for second set of datas using r-format.

Fig.10 Shows Simulation results of jump instruction using J type format. Whenever the instruction is correct then cursor will go fir that location, if not continue next instruction execution.

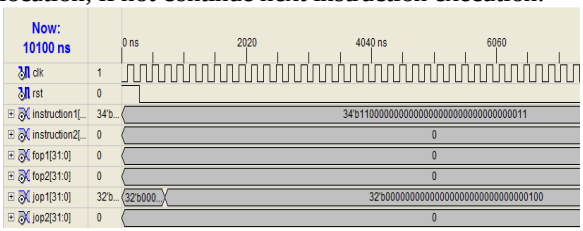

Fig.10: Simulation results of jump instruction using J type format.

Outputs of I and J type formats are obtained at final output (fop1) and final output (fop2), for J type format output is obtained at jump output (jop1) and jump output (jop2).

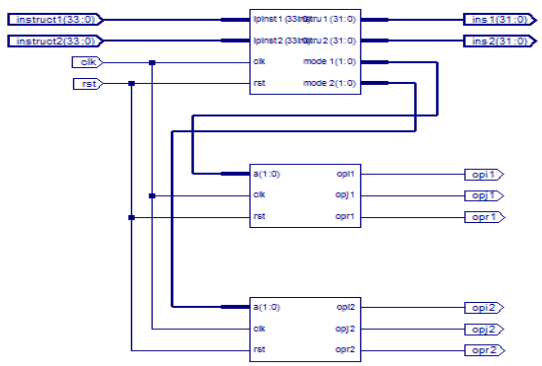

Fig.11: RTL schematic of VLIW-MISC processor.

Fig.11 shows the RTL schematic of VLIW-MISC processor. For the given input of 34-bit length, corresponding mode is selected based on 34th and 33th bits. Remaining inputs are selected based on corresponding format. a(1,0) are used to select the mode. Using VLIW two instructions are executed simultaneously.

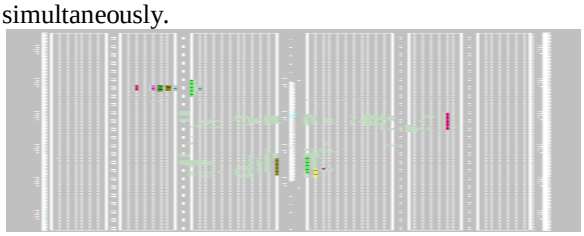

Fig 12: Layout of VLIW-MISC processor on FPGA

Figure 12 represents layout of MISC processor implemented on a vertex 4 family. Components of VLIW-MISC processor that are represented by colored area on FPGA.

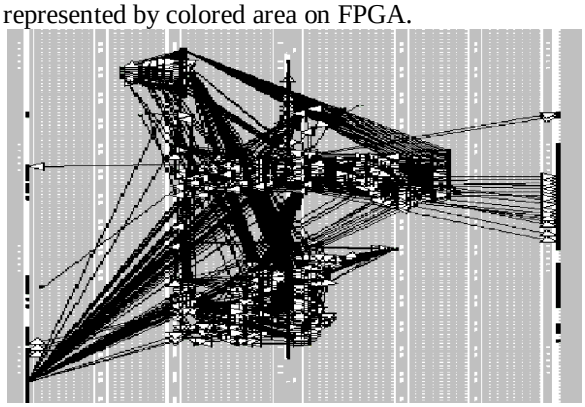

Fig 13: Place and Route Diagram of MISC processor.

Figure 13 represents place and route diagram of VLIW-MISC processor. Placement involves the placing of electronic components, circuitry and logic elements in less amount of space. Where as routing involves designing of all wires needed to connect placed components exactly based on corresponding rules and limitations.

Figure 16 and 17 represents the effect of frequency and voltage variations on power.

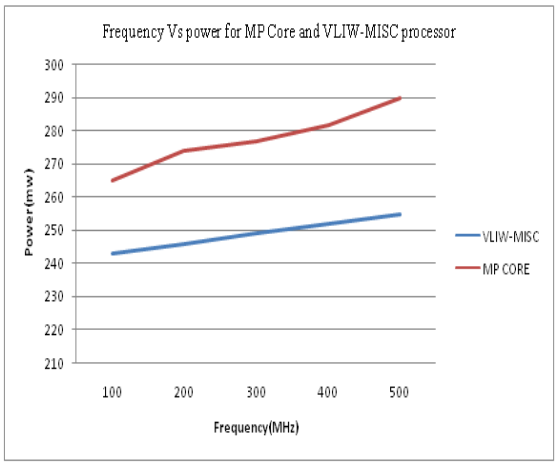

Fig 14: Graph for frequency and power.

Figure 14 represents variation of frequency with power. For the linear proportionality of frequency and power, this MISC processor can be operated for high frequencies, with slight increase in power consumption. Compared to MP Core, VLIW-MISC processor consumes less amount of power. For example, at frequency 500MHz power consumption of VLIW-MISC and MP Core[9] are 255mW and 290mWs respectively. So that power can be saved upto 35% using new design.

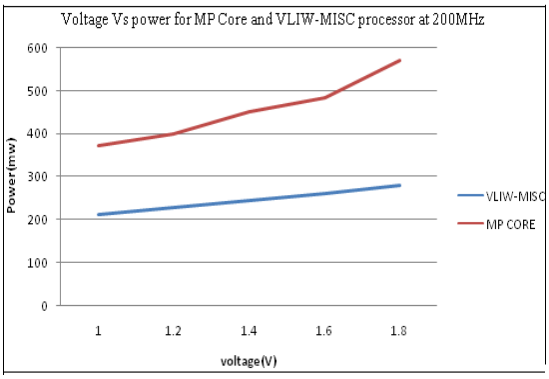

Fig 15: Graph for power and voltage

Figure 15 represents variation of voltage with power. For the increase in voltage, power is also slightly varies. So that this processor is operated with different voltages.

When MIPS operated at 1.4v, power consumption is 270mW where as MP Core consumes power of 480mW generated at same voltage. Table1 shows the characteristics of VLIW-MISC processor.

Table1: Comparison of VLIW-MISC processor with MP core [9].

Parameter	MP core[9]	VLIW-MISC
Total no of gates	7360	6227
slices contains related logic	496	339
Power consumption	401.53mW	271.75mW
Memory usage for the synthesis	247.5MB	240MB
Maximum frequency	0.928GHz	1.25GHz

CONCLUSION

As a result more number of instructions can be executed by adopting S.I.M.E. methodology to MIPS RISC processor. The designed MISC processor consumes less amount of power and occupies less area. Since its operating frequency is high, it can be operated for different applications. Instruction execution is based on VLIW technique, so that 32-bit data is accessible at a time. Speed of operation is also maintained by reducing number of gates. It can be used in real time applications while developing Embedded Systems.

REFERENCES

[1] Rupali S. Balpande ,Rashmi S. Keote "Design of FPGA Based Instruction Fetch & Decode Module of 32-bit RISC (MIPS) Processor", International Conference on Communication Systems and Network Technologies 2011, pp 409-413.

[2] Xiaotao chang,Mingming Zhang, Zhimni Zhang and Zim Wang "Adaptive clock gating techniques for low power IP core in SoC design" IEEE international symposium on circuits and systems, may (2007), pp.2120-2123.

[3] Kui YI, Yue-Hua DING "32-bit RISC CPU Based on MIPS-Instruction Fetch Module Design", International Joint Conference on Artificial Intelligence2009, pp 754-760.

[4] Katarzyna Leijten-Nowak, Jef L. van Meerbergen,"An FPGA Architecture with Enhanced Datapath Functionality" International conference on Field Programmable logic and application 2009, pp 212-217.

[5] Forrest Brewer, Steve Haynal "Transaction Brief -Symbolic NFA Scheduling of a RISC Microprocessor" IEEE transaction on VLSI syatems 2002, VOL. 10.

[6] MIPS32 4KTMProcessor Core Family Software User's Manual, MIPS Technologies Inc.

[7] John L. Hennessy, and David A. Patterson, "Computer Architecture A Quantitative Approach", 4th Edition; 2006

[8] Vincent P. Heuring, and Harry F. Jordan, "Computer Systems Design and Architecture", 2nd Edition, 2003

[9] M.Kamaraju, K.LalKishore,A.V.N.Tilak, "A Novel Implementation of Application Specific Instruction-set processor(ASIP) using Verilog" International Journal of Science, Engineering and Technology, Issue 59, Nov 2011, pp 382-387