International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.7, pp : 719-723 1 July 2013

# Concept, Design and Performance Evaluation of VLVIW Processor

**Ranjan Kumar Behera, Deepak Kumar, K. S. Pandey**

Computer Science and Engineering Department, National Institute of Technology Hamirpur
jranjanb.19@gmail.com ,deepakkumar.nith@gmail.com ,kumar@nith.ac.in

**Abstract: EPIC *processor is one of the best ways to exploit the instruction level parallelism where multiple instructions are issued explicitly by the compiler. VLIW processor is the evolution of EPIC processing paradigm. Very Long Instruction Word (VLIW) is multi-issue processors that try to extract parallelism statically by the compiler .It execute a long instruction that consist of multiple operation. Even if VLIW processor provides higher performance than conventional processor some of application like scientific computing, Digital signal processing, military application may need more processing speed. If instruction set is customize in such a way that its instruction would probably operate on linear array of number (also called as vector) instead of scalar operand, then performance could be increased tremendously. So a VLIW processor can be customize to work as a VLVIW processor which may result in high performance computation.*}**

**Keyword- VLIW, Superscalar, VLVIW, Trimaran, Vectorization**

## I. INTRODUCTION

Basic goal of multi issue processor is to increase the instruction level and data level parallelism [1] so as to increase the performance of the system. Multi issue processor comes in two flavor a) Superscalar processor [2] b) VLIW processor [2]. Superscalar processors issue varying numbers of instructions per clock cycle and are either statically scheduled or dynamically scheduled. Statically scheduled processors use in-order execution, while dynamically scheduled processors use out-of-order execution. Due to some of the limitation arise in superscalar processor, VLIW architecture is used for improve parallelism with less complexity.

Very long instruction word (VLIW) is one kind of processor design that try to achieve high level of instruction level parallelism(ILP) by executing long instruction word composed of multiple operation . The instruction word is also called as MultiOp consists of multiple arithmetic, logic and control operation. Each of the individual operation is a separate operation in case of RISC processor. VLIW concurrently execute the set of operation that are specified within a single MultiOp thereby achieving instruction level parallelism .Although RISC architecture can have temporal parallelism, VLIW can have both temporal and spatial parallelism. If a task is broken into multiple subtask and each subtask is executed in different processing segment simultaneously as in pipelining then it is known as temporal parallelism however if multiple similar task is executed in duplicate hardware simultaneously then that type of parallelism is known as spatial parallelism. Similar to superscalar architectures, the VLIW architecture can reduce the clock per instruction (CPI) factor by executing several operations concurrently. Most of the cases in VLIW all instruction scheduling is done statically.

One of the architecture that is used to controlling multiple functional units is vector processor. Vector processor provides vector instruction. Some properties of vector instruction are [1]:-

- The computation of each result is independent of the computation of previous results, allowing a very deep pipeline without any data hazards.
- A single vector instruction does a tremendous amount of work – it is the same as executing an entire loop. Thus, the instruction bandwidth requirement is reduced.
- Vector instructions that access memory have a known access pattern.

## II. RELATED WORK

The main contribution of this paper is to design and measure the performance of VLVIW processor by customizing the instruction set to get the mixed advantages of both VLIW and Vector processor. Some previous contribution to deeply pipelining in VLIW architecture has been made in paper [6] which shows that VLIW is one of the promising methods in HPC (High Performance Computing) applications. Another previous work has been made to measure the performance in cost-effective and high-performance dual-thread VLIW processor model in paper [7].An important contribution in IS customization has been made in paper [8] which involve only scalability issue i.e. varying the processing power (number of register file, functional units etc.) for evaluating the performance. However this paper shows the graphical results of performance which involves both scalability and Vectorization in VLIW architecture. Some extra effort was involved in a previous paper [12]to evaluate the effectiveness and performance of multimedia and DSP application in SIMD processor and VLIW architecture but very less effort has been made on combine effect of VLIW and SIMD support.

## III. VLVIW PROCESSOR ARCHITECTURE

VLVIW is a multi-issue processor design that try to achieve high level of instruction level parallelism (ILP) by executing long instruction word composed of multiple operations where all operands are vector. In VLVIW architecture all the scheduling is static. This means that they are not done at runtime by the hardware but are handled by the compiler. Since here all scheduling is done at compile time, they tends to have simpler control path and therefore hardware complexity is reduced .VLVIW use multiple independent functional units (FUs) rather than issuing multiple independent instruction into a functional unit. VLIW is a static multiple issue processor which uses compiler to decide which instruction to issue and execute simultaneously .VLVIW issue packet which is a set of instructions (operands are vector) that are bundled together and issue in one clock cycle. The compiler does static branch prediction and code scheduling to reduce the control hazard and eliminate the data hazard .VLVIW architecture moves the complexity from the hardware to compiler. So complexity is paid only once when the compiler is written and not every time when the chip is fabricated.

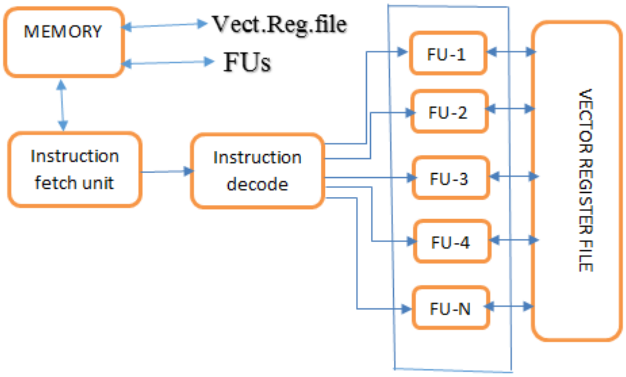


Fig. 1 VLVIW Architecture

Some of the key features of VLVIW processor are

- Multiple functional units are connected through a global shared vector register file
- A central controller is used to issue a long instruction word in every clock cycle.
- Each instruction consists of multiple parallel independent operations where all operands are linear array of number known as vector. Each operation require statically known number of cycle to complete

## IV. FRAMEWORK USED FOR ERFORMANCEEVALUATION

Retargetable compiler and configurable simulator is necessary for research in instruction level parallel architecture like VLIW and superscalar processor. Trimaran framework is best suitable for this kind of research as it has inherently built in tool for compiling and simulating and also it can target variety of ILP processor

Trimaran compiler infrastructure comprises of following components as shown in the Fig. 2:-

1. A machine description facility ,MEDS for describing ILP architecture
2. A parameterized ILP architecture called HPL-PD
3. A compiler front end called as OpenIMPACT for C, is used for parsing, code profiling, type checking machine independent optimization, function inlining

etc.

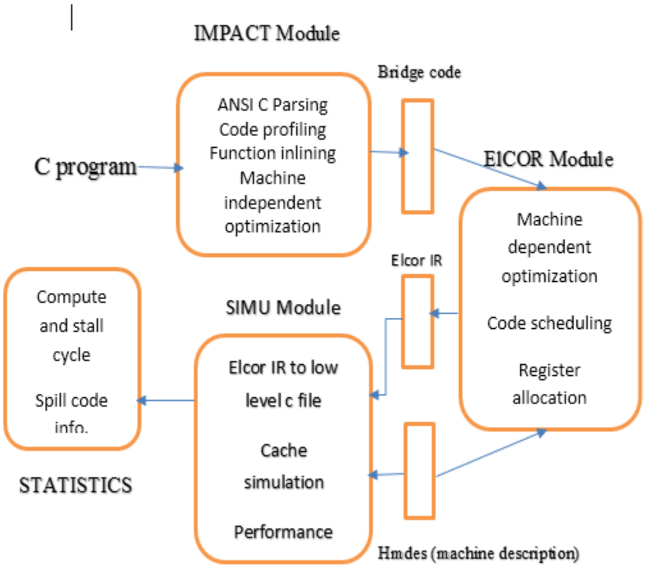


Fig. 2 Trimaran Compiler Infrastructure [9]

4. A compiler back end, called Elcor, parameterized by a machine description for targetable processor (here VLVIW processor). It perform instruction scheduling, register allocation and machine dependent optimization. Each stage of compilation process can easily be changed as per research requirement.
5. A cycle level simulation component which can be configurable by machine description and provide runtime information on execution time, branch frequency and resource utilization.

Some of the brief descriptions of above components are:-

**MDES**:-The machine description model (MDES) in TRIMARAN allows the user to develop a machine description for the HPL-PD family of processors in a high level language, which is then translated into a low-level representation for efficient use by the compiler. High level language allow the user to specify detail execution constraints in an easy to understand, maintainable and retargetable manner. Low level representation is design to allow the compiler to check execution constraint with time/space efficiency. The target processor (VLVIW) is to be described in MDES component of Trimaran.

**HPL-PD**: -HPL-PD is a parameterized ILP architecture which serves asa vehicle for processor architecture and compiler optimization research. It admits both VLIW and superscalar implementations. The HPL-PD parameter space includes number and types of fictional units, number and type of registers, width of the instruction word and instruction latencies.

**Open IMPACT**: - Impact module is the front end of Trimaran compiler infrastructure. It is generalized C compilers that generate optimize code for various architecture. It perform parsing, function inlining, code profiling, machine independent optimization, basic block formation.

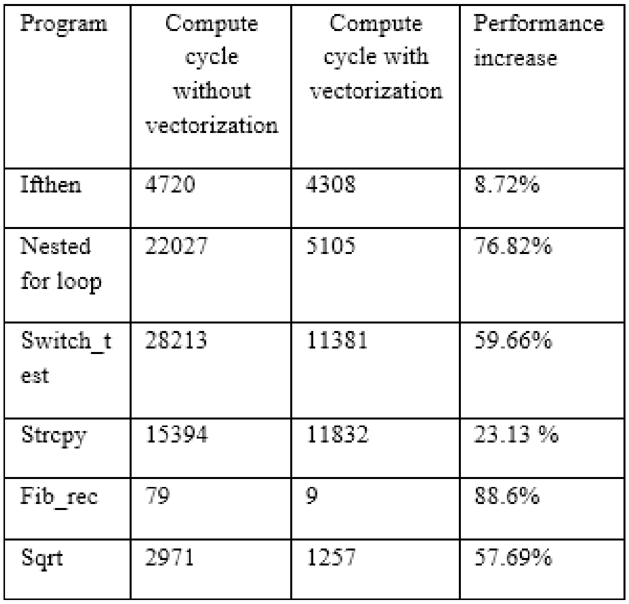
**ELCOR** :-Elcor compiler is a VLIW compiler that takes largely machine independent assembly code and compile it for specific machine describe in the given machine description (MDES) Elcor forms the back end of Trimaran infrastructure . It has several components and each component read, analyze and convert the Lcode into another IR form known as Rebel code. Machine description of our target processor is given to this module. Using the processor description, here machine dependent optimization, register allocation, instruction bundling vectorization is performed

**SIMU**: - It is a Trimaran simulator used to generate multiple statistic to summarize the program execution. Simulator has the static component and dynamic component .the static component is called Codegen and dynamic component is called Emulib. The first generate the low level C file similar to assembly file from the Rebel IR .the file is then compiled using the host native C compiler and linked to the emulation library (Emulib) to simulate the application.

## V. EXPERIMENTAL RESULT

Vectorization is a technique which identify and exploit data level parallelism in a program for efficient execution on architecture with SIMD support .This technique in Trimaran is called as selective vectorization .It creates highly efficient schedule of instruction by distributing computation between scalar and vector processing units in order to improve resource utilization and performance .This type of technique is mostly applicable in multimedia application.

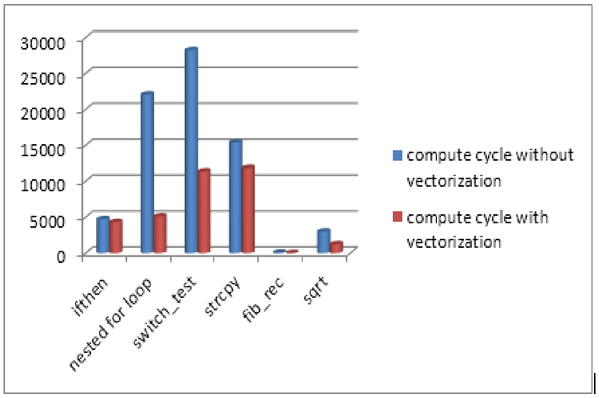
Table 1 show the performance of different benchmark with and without vectorization:-



| Program | Compute cycle without vectorization | Compute cycle with vectorization | Performance increase |
|---|---|---|---|
| Ifthen | 4720 | 4308 | 8.72% |
| Nested for loop | 22027 | 5105 | 76.82% |
| Switch_test | 28213 | 11381 | 59.66% |
| Strcpy | 15394 | 11832 | 23.13 % |
| Fib_rec | 79 | 9 | 88.6% |
| Sqrt | 2971 | 1257 | 57.69% |

Table 1.compute cycle of different benchmark with and without vectorization

Fig 3. Comparison of performance for different source code with vectorization and without vectorization.



The target VLVIW processor can be specified by changing the some of the parameter in machine description file in the Trimaran infrastructure .VLVIW processor take advantages of both VLIW and VECTOR processor. Parameter changes made are issue slots to 6 so that compiler would bundle 6 independent operation in a single instruction known as MultiOps instruction and all six operation would issue in a single cycle . Number of integer Functional unit to 4, number of floating units to 2, number of memory unit to 2.number of branch unit to 1.

After all changes made in MDES file, the source code needs to be compile in the vectorization mode using SUIF dependency library so that the target VLVIW processor design can be achieved .

In Table 2 we have shown the measured performance of different program with single issue vector processor, VLIW processor and VLVIW processor:-

Table2 comparison for different source code for Vector, VLIW and VLVIW architecture

| Benchmark | Compute cycle in vector processor | Compute cycle in VLIW processor | Compute cycle in VLVIW processor | Performance increase as compare to | |
|---|---|---|---|---|---|
| | | | | Vector Processor | VLIW processor |
| Nested for Loop | 5105 | 15623 | 3863 | 24.33% | 75.27% |
| Switch_test | 11381 | 18975 | 7910 | 30.50% | 58.31% |
| ifthen | 4308 | 3209 | 3206 | 25.58% | 0.01% |
| strcpy | 11832 | 9321 | 5613 | 52.56% | 39.78% |
| Fib_rec | 9 | 47 | 5 | 44.45% | 89.36% |
| sqrt | 1257 | 1106 | 1057 | 15.91% | 4.43% |

In the above Table 2 compute cycle involve in execution of different benchmark has been calculated using SIMU components of Trimaran for Vector, VLIW and VLVIW processor. Performance of VLVIW processor is compared with both Vector and VLIW processor in the last column. Fibonacci program which involve recursion (fib_rec) improve performance more than 89% in VLVIW processor when compared with VLIW processor and more than 44% when compare with Vector processor. Similarly loop involve program (Nested for Loop) improve performance more than 75% and 24% when compare with VLIW and Vector processor respectively. Likewise all program where more computation needs to be performed shows better performance in VLVIW processor as compare to both Vector and VLIW processor

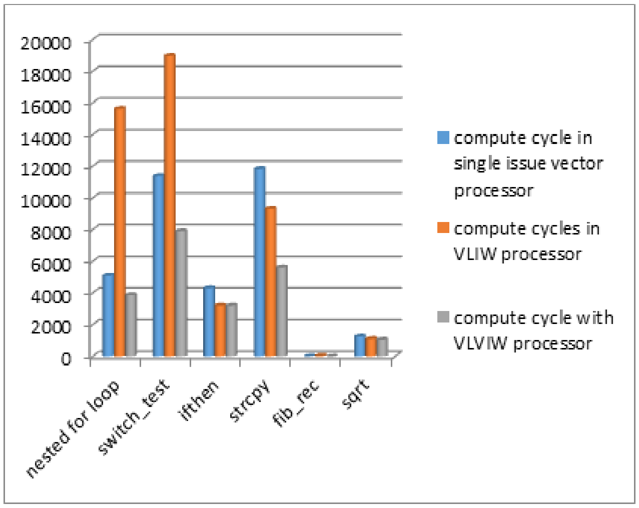


Fig4.Graphical comparison of performance in Vector, VLIW and VLVIW architecture

From the above Fig 4 it has been shown that some of the program where multiple similar operation present like recursion , looping ,switch case perform better in case of vector processing rather than VLIW processor architecture and some other program like string copy ,square root perform better in VLIW architecture rather than vector processing but when we combine the architecture of both VLIW and vector processor most of the program perform better , yielding less compute cycle than both VLIW and Vector processor .This is our target architecture known as VLVIW (Very Long Vector Instruction Word)

VLVIW can be one of the promising architecture in High Performance Computing field.

## VI. CONCLUSION

A new architecture that combine the advantages of both vector and VLIW processor known as VLVIW architecture can be a promising architecture in high performance computing application. Some of the benchmark that are given in Trimaran tool are tested against VLIW, Vector and VLVIW architecture which resulted the best performance for VLVIW architecture as compare to both Vector and VLIW architecture .It can be concluded that for most of the scientific application, DSP application, weather forecasting where large sets of data are used for processing VLVIW can be a good choice in term of performance metrics.

## References

i. *J. Hennessy and D. Patterson. Computer Architecture: A Quantitative approach, Morgan Kaufmann, San Francisco, CA, third edition 2002, page 172-181*

ii. *.Superscalar and VLIW processors, www.ida.liu.se[3].MDES_manual http://trimaran.org/documentation.shtml*

iii. *HPL-PD Architecture Specification: Version1.1 http://trimaran.org/documentation.shtml*

iv. *Joseph A. Fischer, Very Long Instruction Word Architectures and the ELI-512, Proceedings of the 10’Th Symposium on Computer Architectures, pp.140-150, IEEE, June, 1983*

v. *Pipelining and Bypassing in a VLIW Processor Arthur Abnous and Nader Bagherzadeh IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 5, NO. 6, JUNE 1994*

vi. *“Pipelining and Bypassing in a VLIW Processor” Arthur Abnous and Nader Bagherzadeh, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 5, NO. 6, JUNE 1994*

vii. *P. Faraboschi, G. Brown, J. A. Fisher, G. Desoli, and F. Homewood. Lx: A technology platform for customizable VLIW embedded processing. In Proceedings of the27th Annual International Symposium on Computer Architecture, pages 203–13, Vancouver, June 2000.*

viii. *Trimaran4_manual,http://trimaran.org/documentation.shtml*

ix. *A VLIW Processor With Reconfigurable Instruction Set for Embedded Applications Andrea Lodi, Mario Toma, Fabio Campi, Andrea Cappelli, Roberto Canegallo, and Roberto guerrieri IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 38, NO. 11, NOVEMBER 2003*

x. *High-Performance and Low-Cost Dual-Thread VLIW Processor Using Weld Architecture Paradigm”,Emre O¨ zer and Thomas M. Conte, Fellow,IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 16, NO. 12, DECEMBER 2005*

xi. *“Evaluating VLIW and SIMD Architectures for DSP and Multimedia Applications” DeepuTalla Department of Electrical and Computer Engineering the University of Texas at Austin*