

Analysis of High Performance Parallel Computing Instruction Sets

M. Balamurugan¹, S. Vijaykumar^{1*} and S. G. Saravanakumar²

¹School of Computer Science, Engineering and Applications, Bharathidasan University, Trichy - 620024, Tamil Nadu, India; mmbalmurugan@gmail.com, indianid@gmail.com

²6th SENSE, An Advanced Research and Scientific Experiment Foundation, kumbakonam - 612001 Tamil Nadu, India; saravanakumarsg@gmail.com

Abstract

This study explores existing design principles of the processor architecture and identifies future design approach that will help to solve existing business problems that are operable on the scalable environment. We considered the two broader classifications of the instruction sets- RISC and CISC and analyzed the ways to improve the performance of the existing processor design approach. Findings show that all the design principles have been made for different engineering level points to work on the different kinds of task that are specific to the respective fields and SIMD data can be handled well in vector processing environments. Improvements can be made while choosing the design environment based on the business problem and significant design improvement can make to overcome the existing performance-related issues.

Keywords: CISC, MIMD, MISD, RISC, SIMD, SISD, Vector Processing, VLIW

1. Introduction

In the recent trends, many processors are available in the market by stating that it is specialized for the particular purpose for example systems specialized for PC gaming, Mobile Workstations, all day programming models, Business laptops etc. All the manufacturers are pushing for high-end machines to show their computers are being better than the existing models. This results in adding new and more complex design to the existing design so that the computer can handle all types of complex operation capabilities. Already many organizations started using big data systems to support their day to day business operations and there is a big demand for buying high-end systems to support their scalable environment. In this paper, we analyzed various factor that needs to be taken into account while identifying suitable processor design for an infrastructure and for a specific business function. Identified that instruction set plays a major role in the processor design because it closely works with switches and transistors for each and every step of the processor and all the multi-core processors are aiming to

achieve parallelism on all the operations to obtain better execution results.

2. Instruction Set

An instruction set is a collection of commands for an internal processor to execute the relevant switch between transistors in the CPU design. It allows the processor to process the user programmed instructions without any hazards. An instruction set architecture is set of processor design techniques it redefines the design used, in an each processor.

In a contribution to the instruction set architecture many virtual machines that support bytecode to execute the instructions in the machine language such as Smalltalk, the Java virtual machine, and Microsoft's Common Language Runtime. In addition to these virtual machines just in time compilation techniques execute less frequently used code paths by interpretation. In overall all the instruction set types, Virtual machines and other interpretation techniques allow the programmer to code their programming logic without worrying about the

* Author for correspondence

physical processor, CPU design, and machine language instructions. A Single instruction set that works on the single hardware will perform better than the one which is spilled as a several simpler instructions and accessing a large volume of memory.

3. Classification of Instruction Sets

Instruction sets are classified into three types.

3.1 CISC (Complex Instruction Set Computer)

A Complex Instruction Set Computer (CISC) design has a large set of instructions which are rarely used by the real world programs. In fact, many of the instructions can be used only if we do assembly program. This instruction set computer design can process single instructions executed as several low-level operations or it is capable of processing multi-step operations within single instructions set. This type of instruction set design are used mainly in Intel, AMD processors where it does not worry about the resource consumption and it is mainly used for processing the complex instructions such as calling the remote procedure call to another system, accessing the resources from complex addressing modes, accessing the elements in the complex data structure, infinite looping etc.

3.2 RISC (Reduced Instruction Set Computer)

A Reduced Instruction Set Computer (RISC) simplifies its number of instruction set and it instruct the processor to implement the instructions that are small, highly optimized instructions through user programs, There is

a wrong assumption has been spread across in the design world that RISC design approach reduces the number of instructions set from the CPU design but it actually uses the highly optimized instruction set rather using more versatile set of instructions to achieve better execution time¹.

The term ‘reduced’ denotes it reduces the number of execution step which is executed by the single instruction by executing in the single cycle rather executing in multiple execution cycles like CISC. In this design approach, the performance of the executing is purely based on the quality of the code it is executing. RISC design is mainly used in smartphones and other small scale electronic devices. If any complex instructions need to be executed then the algorithm needs to split each instruction into small pieces as input for the RISC to achieve the better execution results. In RISC architecture the number of the register used is huge when compared to other architecture to avoid direct memory interactions².

3.3 VLIW (Very Long Instruction Word)

Very Long Instruction Word (VLIW) design proposes the design where the processor receives multiple instructions set which is executed as a single long instruction. Here compiler plays a major role in deciding which instructions should execute in parallel and schedule each task.

4. Instruction Types

Instruction types have been described in Table 1.

5. Need for Parallelism

In the recent trends, all the instructions sets are required to process in parallel to achieve the better execution

Table 1. Instruction types

| | |
|-------------------------------------|--|
| Data handling and memory operations | Transfer of data from one operand to another. Register to register. Register to memory. Memory to register. Load and store architecture. |
| Arithmetic and logic operations | Add, Subtract, Multiply and divide Compare two values, bitwise operations |
| Control flow operations | Branch operations, control flow to another, Making calls to another operations, indirect calls. |
| Coprocessor instructions | Load/ store data between register and co-processor operations |
| Complex instructions | Loading bulk data to the registers, moving large memory blocks, Complicated mathematical operations such (sin, cos, tan etc.) |

time. Many organizations are already started deploying big data systems into their production environment to support their day to day growing data. Now in the recent trends, even our local desktop computers are come up with multicore processors like (i3, i5, i7 etc.) to achieve parallelism which is enabled with Hyper-Threading model where single core processor act as multicore to satisfy today's programmer needs. Currently, many parallel programming frameworks are popular among all the programmers where they intend to use it in their production environment to handle a huge load. Parallelism can be classified into three types: Bit-level parallelism, Task parallelism, Instruction level parallelism.

6. Why Instruction-Level Parallelism

Instruction level parallelism mainly focuses on executing all the instructions in parallel. To achieve higher performance both machine level parallelism and instruction level parallelism is required. All the recent processor having multi-stage pipelines of instructions it helps to process in each stage of the pipeline the processor works on the different set of instructions and the combined pipelined result will be published³. The example of pipelined processor is RISC processor contains five stages described in Table 2.

By comparing Bit-level parallelism and task parallelism the instruction level parallelism requires

Table 2. RISC processor design

| | |
|--------------------------|---|
| Instruction Fetch (IF) | Instruction fetch will cache the values within one memory cycle |
| Instruction Decode (ID) | There is no microcode conversion hence simple combination logic is applied. |
| Execute (EX) | Execution begin with ALU, Bit shifter, multiple cycle's multiplier and divider. |
| Memory Access (MEM) | Data access from the memory occurs if needed |
| Register Write Back (WB) | Write back to a file or memory |

Table 3. Flynn's Taxonomy for Parallel Computing

| | |
|---|---|
| SISD: Single Instruction Single Data stream exploits no parallelism either in instruction or in data. It works on the single instructions | Mainframe computers. |
| SIMD: Single Instruction Multiple Data (SIMD) refers to all parallel instruction units share the same instruction set, but it operates on the multiple data elements. For example an array elements works on the single instruction set operation | e.g. ADD operation on multiple data set, Graphics Processing Units (GPUs), Microsoft's Direct 3D 9.0, Array processor |
| MISD: Multiple Instruction Set Operates on the single data elements. This type of design is mostly implemented in the fault tolerant systems where the switch over of the system can happen during system failures ⁸ | Fault tolerant systems. |
| MIMD: Multiple Instruction Multiple Data stream design has multiple independent instruction sets that works on the multiple independent data set in parallel. | VLIW, HPC systems |

low-cost hardware design and it is more efficient when compare to others. In Instruction level parallelism Flynn's taxonomy classifies different types of parallel computing described in Table 3.

7. Flynn's Taxonomy for Parallel Computing

Parallel computers can be Classifies as distributed computing, Grid computing, multi-core computing, cluster computing, and Vector processors.

Among all the parallel computers vector and multi-core processors are mainly focus on the different kind's instruction sets while other focus on the framing the system setup.

8. Vector vs Multi-Core

A multi-core processor is best suited for MIMD type of instruction sets where multiple independent instruction sets are required to process the data independently in parallel to work on the multiple data streams. Need for executing such complex instruction in the real time is very rare. Still, in many of the scientific or research fields, they are working on the simple instruction set with a large set of data rather than complex instructions (due to increase in real-time data to work on same instruction set).

When there is a need for SIMD type of instruction sets then vector processing model is the best-suited approach

where it will execute a single instruction set instructions on the large data sets. Generally SIMD type of design used to process graphical inputs where it processes RGB color variations in the matrix.

9. Discussion on GPU Part in Map Reduce Clusters and Big Data Applications

GPU works on the vector processing model where it will process the array of data elements in the single instruction stream. GPU process the instructions based on the vector processing model that's why many processor designs are come up with vector processing model in the recent processors to enable the scaling of graphical images without any delay. There is a discussion on improving the overall efficiency when implementing Map reduce programs in the multi-core GPU machines and researchers have been identified certain types of Map Reduce tasks well-suited to GPMR (GPU Map Reduce)⁴.

Another research shows that when they working on the neural networks to implement big analytics techniques they have experimented the execution with GPU in the testing phase of large-scale neural network and the testing results show that the new design with GPU implementation able to attain 2 ~ 11× speed-up when compare to the normal CPU implementation. During the testing, they have found that the new design can scale up a recurrent neural network with GPU proves the accuracy of 47 % on the Microsoft Research Sentence Completion Challenge⁵.

10. Revisiting the RISC-CISC

In this section compares two design approach by using various factors. Measurements have been calculated for the time taken by a processor to complete a process are determined by three factors.

- Number instructions the process have been divided.
- Average clock cycles that are required to execute the instruction set.
- Clock cycle time to complete the operation.

CISC design approach can handle a huge set of instructions by converting it into complex operations. On the other hand, RISC will reduce the number of

instruction set into many pieces to assign in the pipeline for further processing. RISC design will reduce the average clock cycles since it works on the principle of the single clock cycle. Finally, both the design approach will take the advantage of its design to reduce the total number clock cycle time to complete the operation⁶.

Many types of research have tried to take consideration of both the design architecture of RISC and CISC. They have understood each of their importance in specific areas of fields and they have performed real-time comparison study on the RISC and CISC architecture and found that combining both RISC and CISC would be more powerful but due to some implementation and design issue still it is not proposed for implementation. CISC ISA, the x86 continues to be populated nowadays it translates the CISC instructions into RISC-based instruction styles so that it can achieve better execution results and it has the ability to work on complex instruction set instructions as well. This can easily allow CISC processors to approach RISC performance but the CISC ISA has the burden of translating the instruction sets into RISC-based instruction sets. The study identifies that the CISC ISA can be implemented to yield similar performance as RISC processors by considering the extra computation to convert the instructions into RISC-based instruction sets⁷.

Another research proves that ISA being with RISC or CISC seemed to be irrelevant when they compared the performance. Because the in the ISA design it converts the instruction set into another instruction based on the esign what we choose and they have identified that ARM and x86 are made for different engineering level points to work on the different kinds of operations and nothing gets improved by adding ISA class with another⁸.

There is an attempt for combining both RISC and CISC processors to achieve more power, performances etc. and the hardware and software issues have been discussed⁹.

Many suggestions have been raised to commercialize the RISC-based design in all the suitable products and in the future workstations¹⁰. Designing RISC-based hardware takes less time because the time taken to test and debug is very less since there is no microprogramming is involved in the chip and the size of the control unit is very small¹¹. In addition to this, the architecture is very simple and fewer chances of having design errors which lead to the less defective product during the production and cheaper manufacturing cost increased the demand among users^{12,13}.

There is a trial performed on the VAX and MIPS machine it shows that the RISC approach offers fewer cycles per instruction but it has a lot of small instructions per process since the all the large instruction set is divided into small sets and it is pipelined for further processing. Research results show from the monitor on the MIPS machine and a hardware monitor on the VAX, resulting advantage in cycles per program ranges from slightly under a factor of 2 to almost a factor of 4, with a geometric mean of 2.7¹⁴.

11. Summary and Conclusion

In an overview, many research studies have been carried out to explore new ideas in bringing new design approach that supports all kinds of operations and it can be produced with low manufacturing cost with a simple design so that it should have flexible way of operating environment (By without transforming one instruction set form into another instruction set). Some of the research proposals are come up with new design approach and it also discussed the practical limitation of implementing those designs in the production environment. Researchers have identified that all the design principles have been made for different engineering level points to work on the different kinds of task that are specific to the respective fields. In addition to that, we are in the need for achieving parallelism in all the tasks and it pushes the manufacturing companies to implement complex designs. Our analysis identified that the operating single instructions multiple data set are widely used in all the fields and complex operations are rarely used on the CISC environment. We also identified that SIMD data can be handled well in vector processing environments it will solve the issue of operations with large data sets. Future analysis will be in identifying suitable instruction set architecture to support identified business problem and along with achieving parallelism to handling large data sets.

12. References

1. Bhandarkar D, Clark DW. Performance from architecture: Comparing an RISC and a CISC with similar hardware organization. In Proceedings of the 4th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS IV; New York, NY, USA: ACM. 1991. p. 310–9. DOI: 10.1145/106972.107003.
2. George AD. An overview of RISC vs CISC. Proceedings of the 22nd Southeastern Symposium on System Theory; 1990. p. 436–8. DOI: 10.1109/SSST.1990.138185.
3. Dornika DM, Ravi KS, Krishna PG. An efficient identification of dynamic faults using CAN and ARM7 in a wind turbine. *Indian Journal of Science and Technology*. 2016; 9(17). DOI: 10.17485/ijst/2016/v9i17/93112.
4. Jeff AS, Owens JD. Multi-GPU MapReduce on GPU clusters. Proceedings of the 2011 IEEE International Parallel and Distributed Processing Symposium; Washington, DC, USA: IEEE Computer Society. 2011. p. 1068–79. DOI: 10.1109/IPDPS.2011.102.
5. Wang Y, Li B, Luo R, Chen Y, Xu N, Yang H. Energy efficient neural networks for big data analytics. 2014 Design, Automation Test in Europe Conference Exhibition (DATE); 2014. p. 1–2. DOI: 10.7873/DATE.2014.358.
6. RISC versus CISC architecture. Available from: <http://www2.cs.siu.edu/~cs401/Textbook/ch4.pdf>
7. Isen C, John LK, John E. A tale of two processors: Revisiting the RISC-CISC debate. Proceedings of the 2009 SPEC Benchmark Workshop on Computer Performance Evaluation and Benchmarking; Berlin, Heidelberg: Springer-Verlag. 2009. p. 57–76. DOI: 10.1007/978-3-540-93799-9_4.
8. Blem E, Menon J, Sankaralingam K. Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures. 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA); 2013. p. 57–76. DOI: 10.1109/HPCA.2013.6522302.
9. Garth SCJ. Combining RISC and CISC in PC systems. In IEE Colloquium on RISC Architectures and Applications, 10/1-10/5; 1991.
10. Khazam J, Mowery D. The commercialization of RISC: Strategies for the creation of dominant designs. *Research Policy*. 1994 Jan; 23(1):89–102. DOI: 10.1016/0048-7333(94)90028-0.
11. Vijaykumar S, Saravanakumar SG, Balamurugan M. Unique sense: Smart computing prototype for industry 4.0 revolution with IOT and bigdata implementation model. *Indian Journal of Science and Technology*. 2015 Dec; 8(35). DOI: 10.17485/ijst/2015/v8i35/86698.
12. Dinesh D, Kumar RM. Physical design implementation of 16 bit RISC processor. *Indian Journal of Science and Technology*. 2016 Sep; 9(36). DOI: 10.17485/ijst/2016/v9i36/102911.
13. Kumar JV, Raju BN, Babu MV, Sreelekha K, Ramanjappa T. Implementation of low power pipelined 64-bit RISC processor with unbiased FPU on CPLD. *Indian Journal of Science and Technology*. 2016 Sep; 9(33). DOI: 10.17485/ijst/2016/v9i33/89815.