A survey of new research directions in microprocessors

A. J. Sikc[1], B. T. Ungerer[2], and C. B. Robic[2]
[1]Computer Systems Department, Jozef Stefan Institute, Jamova 39, 1001 Ljubljana, Slovenia
[2]Department of Computer Design and Fault Tolerance, University of Karlsruhe, Germany
[2]Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Abstract - *Today's microprocessors applies the instruction-level parallelism by a deep processor pipeline and the superscalar instruction issue technique. Very-Large-Scale-Integration (VLSI) technology provides considerable solutions for the exploitation of the instruction-level parallelism in future generations of microprocessors. This kind of technology progress will take over the gate delay by on-chip wire delay as the main impediment to increasing the chip complexity and cycle rate. In this paper, we observe the current methods to solve this problem. Moreover, we compare several new research directions whose solutions are based on the complex uniprocessor architecture. Because of the connotation of the microarchitecture is functionally partitioned designs with strict nearest neighbor connections must be developed and the problem, facing by the microprocessor designers, is the utilization of a higher degree of speculation in combination with functional partitioning of the processor. Therefore, we found an uniprocessor share some similar attributes with classical dataflow limit imposed by data dependences—super-speculative techniques, trace cache, multi-scalar, and data-scalar. The discussion and comparison in this paper will be focused on the performance potential of those complex uniprocessors.*

Keywords: Super-speculative processor, Trace processor, Multi-scalar processor, Data-scalar processor

1 Introduction

Current microprocessors are the heir of the von Neumann computer and its architecture is a sequential control flow resulting in a consecutive instruction stream. The sequential operating principle of the von Neumann architecture is still the foundation for nowadays' most widely used high-level programming languages of the instruction sets of all modern microprocessors. Despite the main goal of the von Neumann design—minimal hardware structure—is far outweighed by the goal of maximum performance, the architectural characteristics, which has amount of difference between contemporary microprocessor on its internal structure, of the von Neumann design are still well-founded since the sequential high-level programming languages that are used today follow the von Neumann architectural prototype.

We known that current superscalar microprocessors are much more different than the original von Neumann computer, however, the order of the instruction flow as seen from outside by the compiler or assembly language programmer still remain the consecutive program order which is defined by the von Neumann architecture. For the time being, present microprocessors aim to extract as much fine-grained or coarse-grained parallelism from sequential program flow as can be reached by the hardware. Nonetheless, a huge portion of the exploited parallelism is speculation parallelism; that is incorrect speculation. Hence, the result of von Neumann architecture poses a severe obstacle.

The following are the possible developments of von Neumann computer that we can be categorized.

1. Microarchitectures that preserve the same architecture principle of von Neumann. In spite of instruction execution is internally performed in a highly parallel fashion, only instruction-level parallelism can be profitable from the contemporary microprocessors. Moreover, the superscalar principle applied in commodity microprocessors. A lot of similar research topics all fall within the same class of implementation techniques.

2. Processors that almost deviate from the von Neumann architecture, but live with the use of sequential von Neumann languages. Programs are compiled to the new instruction set principles.

3. Processors that increase the throughput of a multi-programming workload by executing multiple threads of control simultaneously. The different approaches are the single-chip multiprocessor and the simultaneous multi-threaded processor.

4. Architectures that breaks with the von Neumann principles. They need to use a modern language such as dataflow with dataflow single-assignment languages, or hardware—software code-sign with hardware description languages. The processor-in-memory, reconfigurable computing, and the asynchronous processor ways also point in that direction.

2 Major contributions of the paper

In this paper, the authors describe several directions in uniprocessor design and explain selected technical terms

concerning sophisticated uniprocessor architecture as well as some challenges and requirement for future microprocessors.

Processor architecture covers two aspects of microprocessor deign: the instruction set architecture, and the microarchitecture, also called internal organization of a processor.

2.1 Microarchitecture

What is the uniprocessor architecture? It is nothing but a deep pipelined, superscalar RISC processor.

The pipelining starts with the IF stage that fetches several instructions from the instruction cache into a fetch buffer. To avoid pipeline interlocking due to jump or branch instructions, the BTAC contains the jump branch target addresses that are used to fetch instructions from the target address. In the instruction decode stage, a number of instructions are decoded; the operand and result registers are renamed. Then the instructions are enclosed in an instruction buffer, instruction window. Instructions, in the instruction window, not only free from control dependence due to branch prediction, but also free from name dependence due to register renaming. In addition, the program order of the issued instructions is stored in the reorder buffer. Instruction issue from the instruction window can be in order or it can be out-of-order. Moreover, it can be either subject to simultaneous data dependence and resource constraints, or divided into two stages, checking structural conflict in the first and data dependence in the next stage. Instructions await their operands in the reservation stations, and then it is dispatched from a reservation station to the functional unit when all operands are available, and execution starts. When the functional unit finishes the execution of an instruction and the result is ready for forwarding and buffering, the instruction is represented "complete". During completion, the reservation station is freed and the state of execution is noted in the reorder buffer. After completion, operations are occur in order. By or after commitment, the result of an instruction is made permanent in the architectural register set. As to the branch prediction, which is now a well-developed part of the state-of-art microarchitecture design. Sophisticated microprocessors use either a two-bit prediction scheme, a correlation-based prediction, or a combination of both. A prediction must therefore mispredict twice before the prediction is changed. This predictor scheme works well with nested loops, because only a single misprediction occurs at the exit point of each inner loop cycle. Mentioned correlation-based predictors use pattern history table, branch prediction table, which is indexed by the branch history register. In order to decrease conflicts, one set of correlation-based predictors uses a hash function into the pattern history table instead of indexing the table.

2.2 Challenges and requirements for future microprocessors

Today's overall trend in microprocessor design is driven by several architectural challenges. In this part, I will just point out some topics for instance, scalable and faster busses, memory latency, fault-tolerant chip, and so on. Moreover, processor architecture must take into consideration the technological aspects of the hardware, such as logic design and packaging technology. In addition, the long interconnect wire delay problem requires a strict functional partitioning within the microarchitecture and a floor planning that avoids long interconnects. As for co-ordinate these processing elements, to serve as a single unified processor will require an additional level of microarchitecture hierarchy for both control and data. Therefore, it has yet to be seen whether a modular design is cost-effective for a very complex single instruction stream general-purpose processor, or whether the pendulum will swing back to less complex processors.

2.3 Future microprocessor architectures

Superscalar processors utilise instruction-level parallelism to speed-up single-thread performance, nevertheless, instruction-level parallelism between successively taken branches is limited, especially for integer-dominated programs. Hence, the excess of resources in issue-bandwidth, pipeline breadth and functional units, which are possible with modern's processor chips, is utilised by speculation. One of the major challenges is to identify ways of expressing and exposing more parallelism to the processor. Because the language structures of von Neumann languages used today limit the amount of extractable parallelism, not enough instuction-level parallelism is available—Higher-level parallelism allows smaller computational units to work in parallel on multiple threads and thereby favours a modular design approach. Another class of architectural proposals contains the multithreaded architecture, in particular simultaneous multithreaded architecture, which is able to pursue two or more threads of control in parallel within the processor pipeline. Moreover, multithreading techniques use coarse-grain parallelism to speed-up the computation of a multithread workload through better utilisation of the resources of a single processor. However, single-thread performance may even slightly deteriorate because of minimal register set and thread overheads. Those examples of architecture can be found in the multiscalar processor, the trace processor, and the datascalar processor. Approaches to those architectural try to speed-up single thread performance by exploiting coarse-grain parallelism, in addition, to insturction-level parallelism.

2.3.1 Five research directions in complex uniprocessor architectures

1.Advanced superscalar processors: wide-issue superscalar processing core with speculative execution.

2.Super-speculative processors: wide-issue superscalar processing core with aggressive data and control speculation.

3.Multi-scalar processors: several processing cores speculatively execute different statically generated program segments.

4.Trace processors: several processing cores speculatively execute different dynamically.

5.Data-scalar processors: several processing cores redundantly execute the same sequential program with different data sets.

3 Absorbing knowledge from this paper

At the first sight of this paper, we can understand what are the main contributions of this paper quickly based on the abstract and introduction. Firstly, they introduce some preliminary concept of microprocessors and its architecture. Secondly, they introduce the concept of microprocessor, which is descendants of the von Neumann, and also describe some major difference between today's microprocessor and von Neumann. This kind of concept gives us some preliminary idea of what are von Neumann and its future development. Thirdly, as for the next section, they explain each step of the superscalar pipeline comprehensively. Most of them we already learn from the class, but when I saw this part of the material again; it totally enhances my knowledge in my mind. When they talk about the challenges and requirements, it let me know some barriers about current microprocessor, and also profound insight of the architecture. Despite the author uses a lot of terms that I am not fully understanding, after multiple time of reading this paper, I become more used to it, and have more rudimentary knowledge about this topic. Hence, this paper not only gives me another instructive skill but also broaden my realm on the microprocessors.

4 Unfamiliar concepts

When the author talks about the future microprocessor architecture, I become lost its attention. Even though I do some annotation on this paper, search the meaning of the term online, and try to find more information, I still cannot fully understand. Most of them are using a similar concept, but the result is totally different. I know what they try to mention on the paper and also indicate the direction of their survey. However, it is just a preliminary aspect of me. It is hard to digest correctly with this kind of knowledge of my own. I hope some of the forthcoming development topic could discuss during the class.

5 Conclusions

As a microprocessor generation becomes mature the next generation starts to become visible. In the paper, author surveyed the uniprocessor alternatives, such as advanced superscalar, super-speculative, multiscalar, trace, and data-scalar processor, which are all examples of new microarchitectures suitable for the next generation. All the microarchitecture techniques described increase the memory bandwidth requirements compared to today's superscalar microprocessor. Thus, all these microarchitecture techniques may be grouped in future with the processor-in-memory or intelligent RAM approaches with DRAM memory on the same chip to solve the memory latency bottleneck. Furthermore, there are also research directions in highly parallel chip architecture that deviate from the von Neumann architecture model. Those two such direction are concentrated on the processor-in-memory approach and on reconfigurable processors.