The Intel x86 approach to the RISC model

Giovan Carlo Germoglio

Departamento de Informática
Universidade do Minho
4710 Braga, Portugal
gicage@hotmail.com

Abstract – *The new decodification and translation features IA-32 instructions into RISC instructions, called micro-operations, kept the Intel´s processors as competitive as the RISC architecture processors. This paper will show the characteristics of this architecture.*

1. Introduction

With the fast advances in the development of integrated-circuits technology, the Intel, that is one of the most knew brand that develop processors, come doing many efforts and changes, trying to following the trends. The microprocessor families Intel x86 is characterized by its abundant instruction sets, multiple addressing modes, and multiple instruction formats and sizes. The control units are considered complexity because they need to know the differences between a large number of opcodes, addressing modes and formats. These characteristics of architecture are known as Complex Instruction Set Computer [1].

As can be seen in [1], an attempt by the designers to make an architecture with less number of instructions, addressing mode, and multiple formats and sizes than the CISC, came up a new option of architecture, called RISC – Reduced Instruction Set Computer, become a direct opposite of the CISC.

To make sure of the importance of RISC architecture, many studies about programs´ behaviour have shown that a majority of CISC instructions are never used, that is, 25% of the instructions in the instruction set make up 95% of execution time. In this case, the hardware-supported instructions often are not used at all. If the CISC machines taken out some these instruction from the instructions set, control unit, that decoding instructions, they will be less complex [1].

Although the importance of architecture RISC, according to [2], the Intel had success maintaining its compatibility with old version of processors, increase the number of transistors led Intel to approaching RISC architecture, using RISC instructions internally and externally ensuring the compatibility with x86 instruction set.

2. IA-32 evolution

The Intel 80x86 architecture, differently of RISC architectures, was improving the processors characteristics through 20 years, adding news features to the instruction set, more transistors in its integrated-circuits and finally making some changes in the 80x86 microarchitecture, approaching RISC architecture.

Starting of IA-32, following a brief evolution of this architecture to P6 microarchitecture, remembering that the old version of this processors family had very importance to reach new technologies and improvements seen today[3][4]:

- The architecture 80286, also knew as 286 that was a 24-bit architecture was extended by 80386 to 32-bits. Not only added 32-bit register and 32-bit address space, but new features of addressing modes and additional operations, became the 80386 nearly a general-purpose register machine;
- After 80386, appeared the 80486 that maintain a similar instruction set to its predecessor, adding few extras new instructions. This processor came up in 1989. The principal impacts of improvements was in the hardware level, where it has one on-chip unified instruction and data cache, an on-chip floating-point and an enhanced bus interface unit;
- The main changes that the P5 microarchitecture, which is used by Pentium, made in 486 architecture was support a superscalar architecture allowing more than one instruction per clock, that is a RISC characteristic. Also by add Single Instructions Multiple Data, more knew as MMX, only later models, and 64 path data, the quantity of information pulled from the memory on each fetch;
- The P6 microarchitecture, used by Pentium Pro, Pentium II and Pentium III, improved the hardware utilization, decoding IA-32 instructions to RISC instruction, called by Intel micro-operations, and adopting the out-of-order execution. This architecture will be the focus this paper, trying to show with more details how this architecture approach still more to RISC architecture.

3. Understanding the P6 Microarchitecture

The Pentium Pro, Pentium II, and Pentium III have P6 microarchitecture as the base of their architectures. The P6 adopted new features in its microarchitecture. These new characteristics brought, according to [4], high hardware utilization. The new features in P6 microarchitecture are decoding the IA-32 instructions to a series of RISC instructions, called micro-operations (uops) by Intel, after being executed by the various pipeline stages. Another new characteristic this architecture is because these RISC instructions can be executed out-of-order [3].

In each clock cycle, three IA-32 instructions can be fetched, decoded, and translated into RISC instructions. But only six RISC instructions or micro-operation can be generated by each clock cycle. If the IA-32 instruction needs more than four uops, they will be generated in multiple clock cycle, being the first four uops to the first IA-32 instruction and the others to the remaining instructions [3].

After the IA-32 instructions are decoded into RISC instructions or into a series of RISC instructions, if it needs more than 4 uops, they will be executed in an out-of-order pool of pending instructions, where these instructions can be executed without following the same order of program instructions, considering that there is not a dependency between them, rising the hardware utilization [5].

Briefing how the steps described above works, the Figure 1 below show how the P6 microarchitecture is divided and the path followed by the instructions.

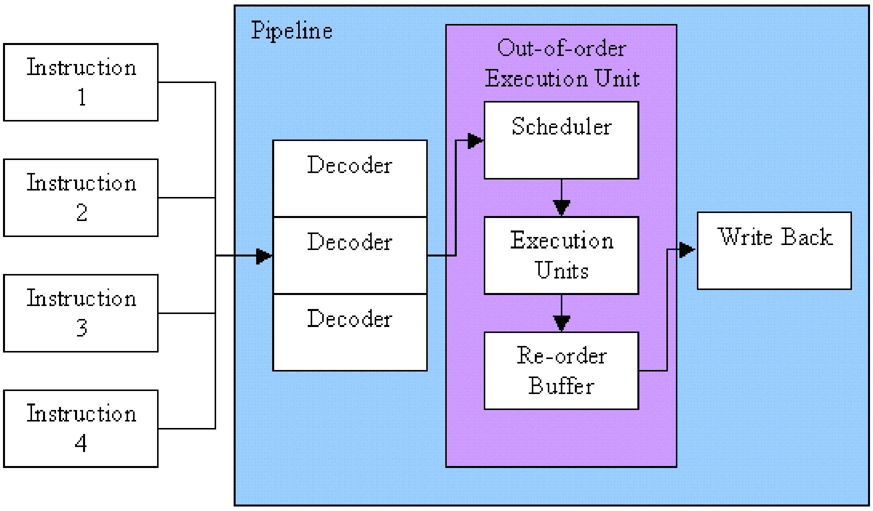

Figure 1 – P6 microarchitecture[4]

As show the Figure 1, the P6 microarchitecture has three stages. The first, called front-end, has eight pipeline stages that fetch, decode, translate and dispatch the program in-order instruction into RISC instructions. The second, called out-of-order core, has three pipeline stages that execute the RISC instruction in each clock cycle. The out-of-order core is integrated with the integer, float point, branch, memory address and memory access units. The last stage, represented by in-order retirement logic, put the instructions in the original program order. This stage has three pipeline stages, where can be take out three RISC instructions per clock cycle [5].

There are different latencies and repeat rates when a uop is ready to take out. The table below show this numbers according to the operation [2]:

Instruction name	**Pipeline stages**	**Repeat rate**
Integer ALU	1	1
Integer load	3	1
Integer multiply	4	1
FP add	3	1
FP multiply	5	2
FP divided (64-bit)	32	32

Table 1 The latency and repeat rate for common uops in the P6 microarchitecture

4. Conclusion

With the improvements that the P6 microarchitecture added to 80x86 Intel family became more difficult to distinguish what the processors have an architecture CISC or RISC, because they are become a hybrid architecture, maintaining the compatibility with predecessors, ensuring IA-32 instructions externally that will be decoded and translated into RISC instructions internally.

New features were added in the P6 microarchitecture, like a number of pipeline stages, rising from 14 to 20, an improvement in the out-of-order engine and others, calling NetBurst microarchitecture. But the P6 microarchitecture was a big step to approach the Intel CISC architecture to a RISC architecture.

References

[1] Jamil, T. 1995. RISC versus CISC. *IEEE Potentials* 14(3): 13-16.

[2] Hennessy, John L. and Patterson, David A. 2003. *Computer Architecture – A quantitative approach*, Third edition, Elsevier Science USA, 260-272.

[3] Hennessy, John L. and Patterson, David A. 2003. Appendix D - An Alternative to RISC: The Intel 80x86, *in*: *Computer Architecture – A quantitative approach*, Third edition, Elsevier Science USA.

[4] Sarmiento, S. Recent History of Intel Architecture - a Refresher, *in:* http://www.intel.com/cd/ids/developer/asmo-na/eng/44015.htm?prn=Y&.

[5] Bhandarkar, D. 1997. RISC versus CISC: a tale of two chips. *ACM SIGARCH Computer Architecture News* 25(1): 1-12.