

ILP in the Undergraduate Curriculum

Daniel Tabak
ECE Dept., George Mason University,
Fairfax, VA 22030-4444
Tel. (703) 993-1598, FAX (703) 993-1601
e-mail: dtabak@osf1.gmu.edu

ABSTRACT

The paper discusses the teaching of instruction level parallelism (ILP) in undergraduate electrical engineering (EE) and computer engineering (CpE) curricula. An argument is made for justifying the teaching of this topic, usually taught in graduate courses, at the undergraduate level. A detailed account of the way this topic is actually taught at the author's University is given. The paper discusses the specific ILP subjects, presented to the students, along with the technical literature sources used.

1. Introduction.

The study of instruction level parallelism (ILP) has been relegated primarily to textbooks intended for graduate studies [1]. It is also the practice in many Universities to teach this topic at the graduate level in most cases. At the same time, it should be realized that practically all modern computers, be they RISC or CISC, are implementing ILP on a constantly growing scale. Some of the latest products, worth mentioning, are Intel Pentium 4, Intel and Hewlett-Packard (HP) IA-64 architecture Itanium, AMD Hammer (64-bit Intel x86, or IA-32, architecture), Sun Microsystems UltraSPARC, Silicon Graphics Inc.(SGI) MIPS R10000, and others.

The author's department of Electrical and Computer Engineering (ECE) at the George Mason University (GMU) has two engineering curricula: electrical engineering (EE) and computer engineering (CpE), leading to all three degrees (BS, MS, Ph.D.). The author has been teaching for many years a senior course on computer design. This course is required

for the BS degree in CpE, and it is a technical elective for the BS in EE.

It has been realized by the author, who developed this course from scratch, that students graduating with the BS degree and going into industry (in most cases) or to graduate studies, should be knowledgeable not only of the basic engineering principles of computer organization and architecture, but of the most recent design techniques and practices, implemented in modern processors. For this reason, the course content has been constantly changed and revised from year to year (sometimes, from semester to semester), to reflect the perpetual innovations in computer design.

As ILP began to be one of the main topics of research and practice of microarchitecture, it was introduced, in a timely manner, into the senior course on computer design. Recently, the subject of ILP also started to appear in textbooks intended primarily for undergraduate curricula, such as [2], chapter 8 and [3], chapter 5. The details of the ILP topics, covered in this course, are described in this paper. The course program and its literary sources are presented in the next section. Section 3 lists the examples of actual ILP processors, presented to the students. Section 4 includes concluding comments.

2. ILP in the Computer Design Course.

Prior to going into ILP, the students are exposed to a very detailed study of scalar pipelining. The primary textbook of the course is [1]. It was used in this course since its first edition in 1989. Chapter 3 in [1] has a very exhaustive coverage of pipelining. A good coverage of pipelining can also be found in [2], chapter 8, and [3], chapters 4 and 5.

After going over the basic principles of pipelining, using the examples in [1], chapter 3, the students are exposed to what can go wrong in pipelines; namely, to the possible pipeline hazards:

- Structural hazards
- Data hazards
- Control hazards

The above hazards, and some of their possible remedies, are discussed in detail. It is later pointed out that these hazards are only more serious in case of ILP.

Subsequently to pipelining, the discussion of ILP is initiated, using chapter 4 of [1] and other sources [4-6]. Sources [4,5] were chosen because they constitute extensive surveys on the subject with relatively large references lists. Report [6] was included because it contains very useful material on branch prediction, not available in such concentrated form elsewhere. In addition, material was taken from [7-9]. These are some of the earliest ILP publications, containing basic material. Superscalar, superpipelined, and very large instruction word (VLIW) operations are defined. However, the course concentrates primarily on superscalar operation, because of its prevalent implementation in industry. With the advent of the Intel-HP IA-64 architecture, more weight to VLIW may be given in the future.

Initially, problems involved with data dependence in ILP operations are discussed in detail. The concepts of name dependence, antidependence, output dependence, and control dependence, are defined, and some examples are given. The examples are taken both from [1] and some are supplied by the instructor. In addition, the following terms, associated with this topic, are defined and pointed out in the examples:

- Rear After Write – RAW
- Write After Read – WAR
- Write After Write – WAW

Subsequently, the following methods, approaches, and special data structures, having to do with data dependence, are studied in detail:

- Register renaming
- Speculative execution
- Out-of-order execution
- Scoreboarding
- Reorder buffer (ROB)
- Reservation stations (RS)
- Trace caching

All of the above topics are well covered and exemplified in chapter 4 of [1]. Other sources, such as [4,5,7] are also used. It is pointed out to the students that instead of using an RS in front of each functional unit (FU), one can use one central window with more entries, to forward operands to all FUs [7]. Some processors are indeed implementing this option.

Some topics in chapter 4 of [1], having a strong software “flavor”, such as loop unrolling, are skipped. It has been the experience of the author, that engineering majors do not willingly accept topics involving programming. Had the course been given to computer science majors, the above topics would also be included.

Problems due to branches in ILP, particularly those dealing with the conditional ones, are handled next. The topics of speculative and out-of-order execution are raised again. In addition, the following topics and data structures are studied:

- Branch prediction (local, global, bimodal)
- Branch target buffer (BTB)
- History table (HT)
- Counter structure (Counts)

This material is also covered in chapter 4 of [1]. In addition, references [4-7] are used. Of particular importance on this topic is the report [6].

The topic of data prediction is not covered in the undergraduate curriculum, since it belongs in the realm of basic research, as opposed to current industrial practice. It is relegated to a subsequent graduate course in computer architecture, along with other more advanced topics (such as explicitly parallel instruction computing - EPIC, for instance).

3. Examples of ILP Systems

Examples of actual processors, both of the RISC and CISC type, implementing ILP, are presented to the students. Special data structures and methods, discussed in the previous section, are pointed out to the students, as they are encountered in the processor examples. Some of the examples are brought up during the discussion of various topics in section 2. Reference [1] contains a number of examples. Reference [4] contains examples of SGI MIPS R10000, Compaq (Digital) Alpha 21164 (actually used in the primary computing system on GMU campus), and AMD K5. A number of ILP examples (including the R10000 and Alpha 21164) can be found in [10]. Another source to which students are directed is the Internet (websites such as www.intel.com, developer.intel.com, www.extremetech.com and others).

The main ILP implementation example, illustrated in detail in this course, is the Intel-HP IA-64 architecture with its first product, the Itanium. Most of this material comes from the Intel and HP websites on the Internet.

In conjunction with the study of the IA-64 architecture, the students are familiarized with the concept of predication, along with illustrative examples of its implementation. The concepts of EPIC [11,12] are briefly covered. The details of EPIC are relegated to a subsequent graduate course on computer architecture. In the Itanium example, ILP features, discussed in general earlier, such as register renaming, scoreboarding, branch prediction, and multiple FUs, are pointed out to the students.

The Intel IA-32 architecture products are also included in the examples, particularly the latest Pentium 4. Also here as for the Itanium, the ILP features such as out-of-order execution, trace caching, branch prediction, and multiple FUs, are stressed. The multiple register files (128 registers) in both Itanium and the Pentium 4 are pointed out to the students. In the Pentium 4, those are of course rename registers along with the old x86 architecture 8 “general purpose” registers (not quite “general”, because of their special tasks).

Other examples, such as the Alpha architecture processors (actually used on the GMU campus), the Sun UltraSPARC, and the SGI MIPS R10000, are also covered.

4. Concluding comments

Because of the prevalence of ILP implementation in industrial products, it is obvious that the topic should be included in undergraduate curricula, preparing engineers and computer specialist for the information technology industry. A sample of a possible undergraduate coverage of ILP, as practiced in a senior EE and CpE course at GMU, has been presented. This program has been constantly revised and modified in the past few years, to follow up the developments of the state of the art and engineering practice. This development and constant revision of the course is intended to continue.

REFERENCES

1. J.L.Hennessy, D.A.Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed., M.Kaufmann, San Francisco, CA, 1996.
2. C.Hamacher, Z.Vranesic, S.Zaky, *Computer Organization*, 5th ed., McGraw Hill, NY, 2002.
3. J.P.Hayes, *Computer Architecture and Organization*, 3rd ed., McGraw Hill, NY, 1998.
4. J.E.Smith, G.S.Sohi, The Microarchitecture of Superscalar Processors, Proc.IEEE, vol.83, no.12, pp.1609-1624, Dec.1995.

5. A.Moshovos, G.S.Sohi, Microarchitecture Innovations, Proc.IEEE, vol.89, no.11, pp.1560-1575, Nov.2001.
6. S.McFarling, Combining Branch Predictors, WRL Technical Note, TN-36, June 1993.
7. M.Johnson, *Superscalar Design*, Prentice Hall, Englewood Cliffs, NJ, 1990.
8. N.P.Jouppi, D.W.Wall, Available Instruction-Level Parallelism for Superscalar and Superpipelined Machines, In Proc. ASPLOS III, pp.272-282, Boston, MA, April 1989.
9. N.P.Jouppi, The Nonuniform Distribution of Instruction-Level and Machine Parallelism and its Effect on Performance, IEEE Trans. on Computers, vol.38, no.12, pp.1645-1658, Dec.1989.
10. D.Tabak, *RISC Systems and Applications*, RSP, UK and Wiley, NY, 1996.
11. M.S.Schlansker, B.R.Rau, EPIC: Explicitly Parallel Instruction Computing, IEEE Spectrum, vol.33, no.2, pp.37-45, Feb.2000.
12. M.S.Schlansker, B.R.Rau, EPIC: An Architecture for Instruction-Level Parallel Processors, HP Laboratories Report, HPL-1999-111, Feb.2000.