VLIW: Is It For Real This Time?

Wen-mei Hwu
Department of Electrical and Computer Engineering
University of Illinois, Urbana-Champaign

MICRO-27

Outline

- ILP Opportunities and Challenges
- Predicated Compilation and Execution
- Control and Data Speculation
- Interprocedural Compilation
- VLIW: Is it your Father's Oldsmobile?
- Instruction Set Architecture Change
- Outlook

1995 ILP Microprocessors

	Width	br	ld/st	alu	float	clock	SPEC int	SPEC float	issue
AMD K5	4	1	2	2	1				oo order
PowerPc 620	4	1	1	2	1	133MHz	225	300	oo order
MIPS R10000	4	1	1	2	2	200MHz			oo order
Alpha AXP21164	4	1	2	2	2	300MHz	325	500	in order
UltraSPARC	4	1	1	1	1	167MHz	250	300	in order
PA-8000	4	1	2	2	2				oo order

ILP Technology Challenges

- Performance of general code:

  control flow, memory dependence,
  function boundaries, memory latency
- Scalable implementations
- Efficient execution of existing binaries
- Smooth migration of existing binaries

ILP Technology Challenges Control Flow

- Frequent appearance of branches
  - execution limited by branch handling
- Frequent occurrence of mispredictions
  - pipeline break and recovery cost
- Compiler code parallelization
  - missed parallelization opportunities
- Compiler code motion and scheduling
  - wasted speculation and parallelism

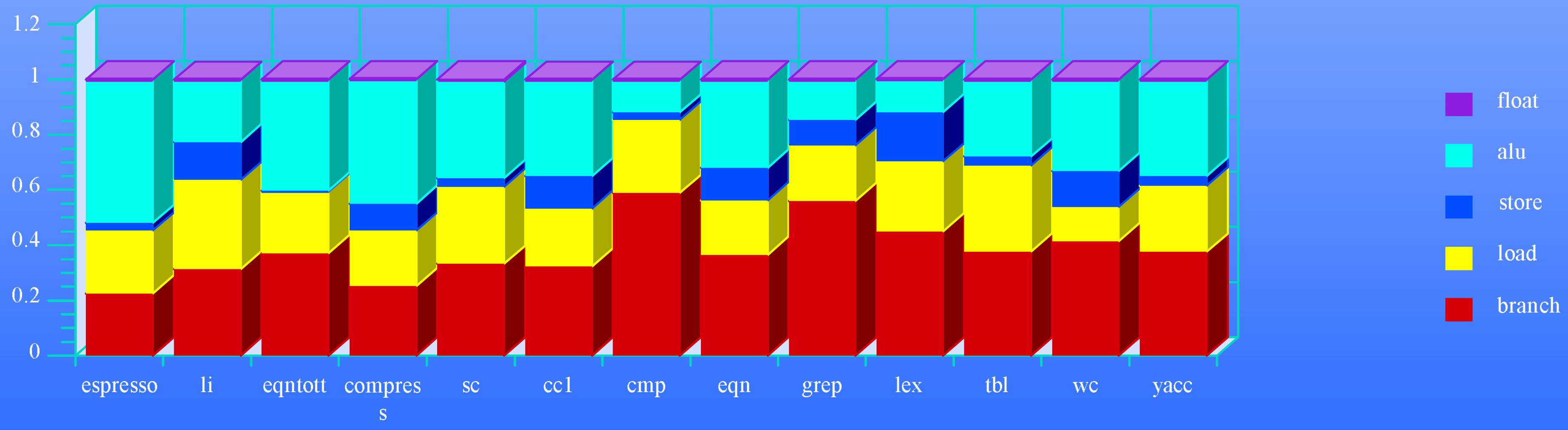
Predicated Execution and Compilation

- Predicated execution architecture
  - branch elimination
  - code explosion control
- Predicated Compilation
  - optimization along multiple paths
  - scheduling along multiple paths

Predicated Execution

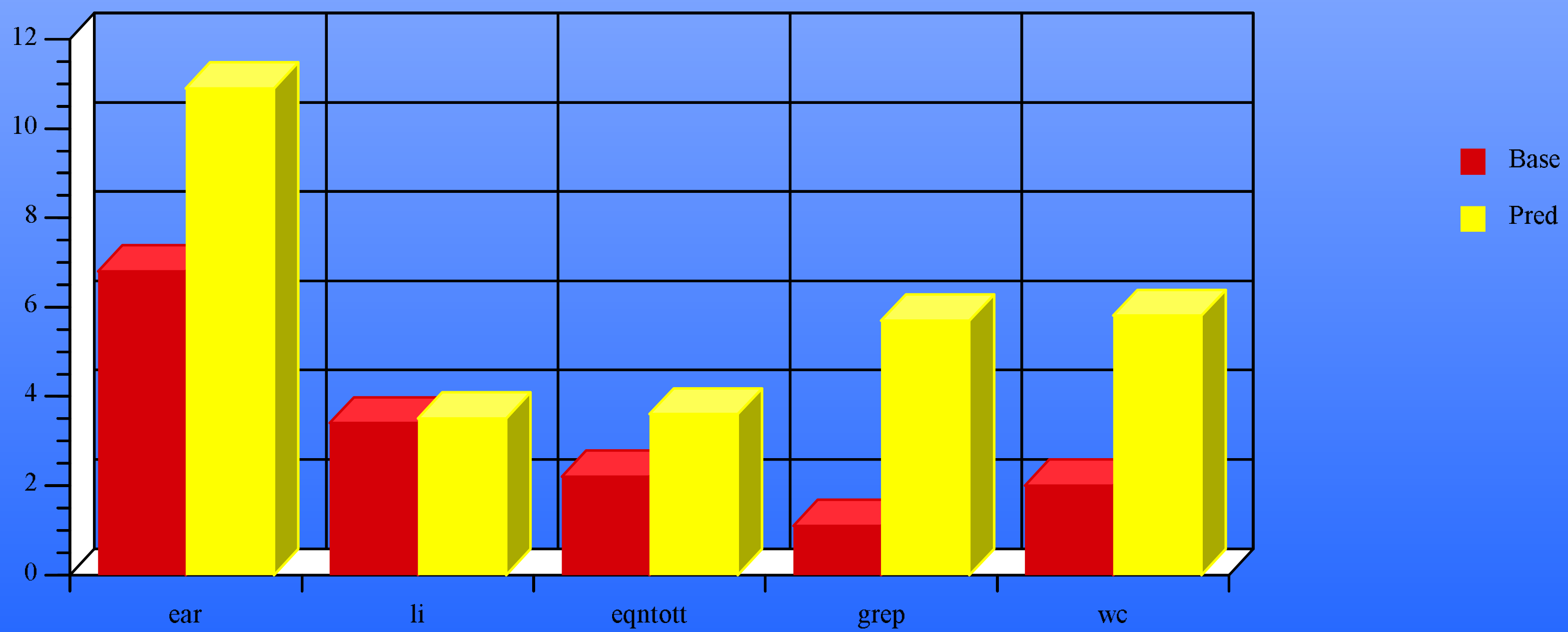
- Mask generation and use in vector pipelines [Cray, etc.]
- Cydra-5 *stuff* instructions and predicated instructions [Rau *et al.*]
- Guarded execution to hide branch delay [Hsu and Davidson]
- *Cmov*, *nullify* and *select* in Alpha, SPARC V9 and PA-RISC
- PlayDoh predicate define instructions and predicated instructions [Kathail *et. al.]*

Dynamic Instruction Mix

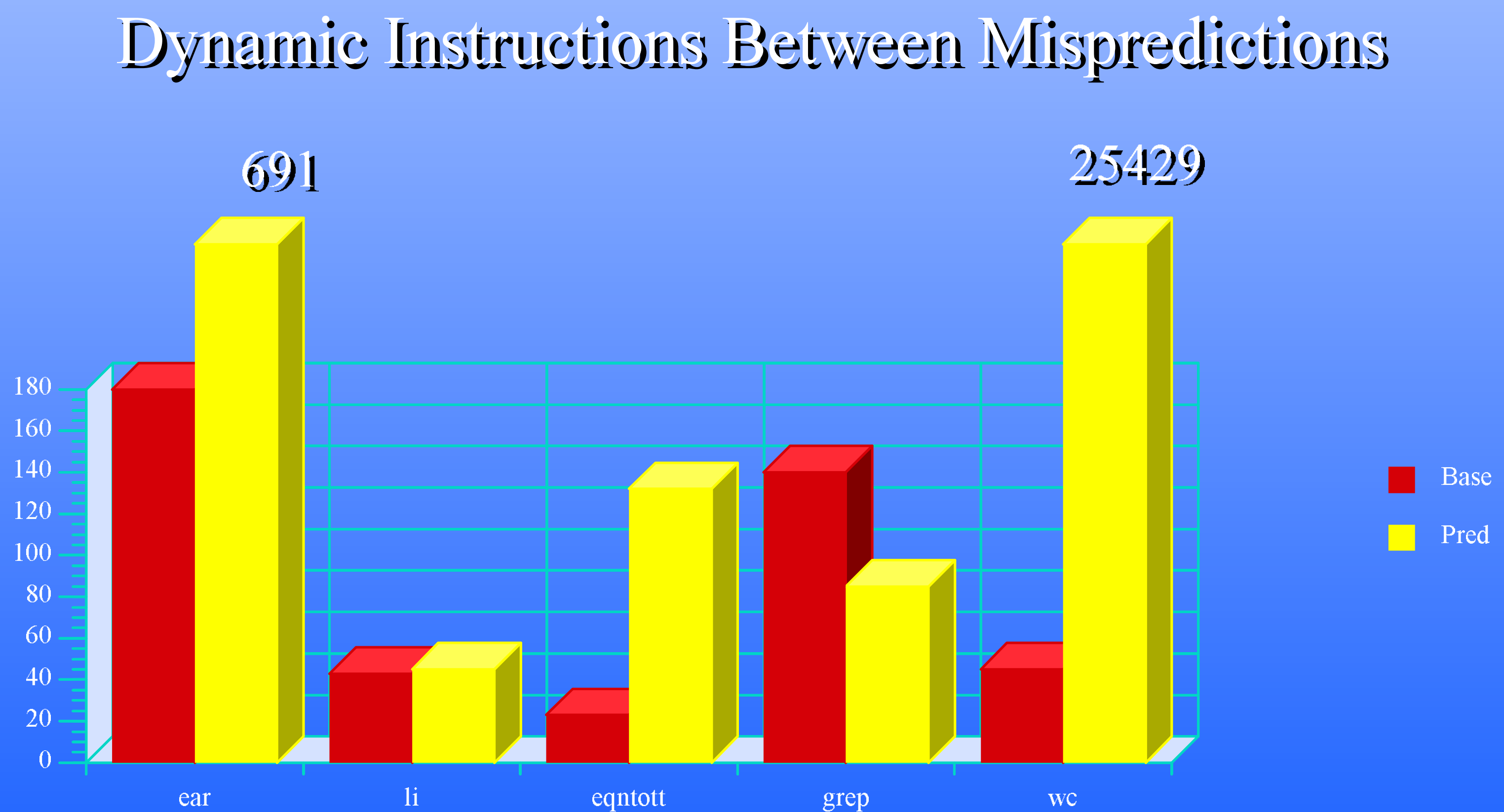


Predicated Execution

Dynamic Instructions between Branches



Predicated Execution



Predicated Compilation

- Parallelize and schedule along multiple control flow paths
- If-Conversion to generate predicated code eliminate branches [Park and Schlansker]
- Hyperblock formation, optimization and scheduling (PHG) [Mahlke et al.]
- Height reduction [Kathail et al.]
- Reverse If-Conversion to decouple from predicated execution [Warter et al.]

Predicated Compilation
Evolution of Modulo Scheduling

- Modulo scheduling for single basic block loops [Rau *et al.*]
- Cydrome predication for loops with simple control flow [Schlansker *et al.*]
- Hierarchical reduction for loops with simple control flow [Wood, Lam]
- PlayDoh predication for loops with arbitrary control flow [Kathail *et al.*]

Predicated Compilation Evolution of Modulo Scheduling

(Cont.)

- Reverse If-conversion to decouple from predicated execution [Warter *et al.*]
- Control recurrence height reduction [Kathail, Schlansker and Anik]
- Hyperblock for loops with side exits and excluded paths [Lavery *et al.*]
- Several more in progress

Memory Disambiguation

- Most advanced compiler techniques work at the source code level
- Most code optimization and scheduling work at the machine code level
- The difficult issue is to provide accurate memory disambiguation at the machine code level

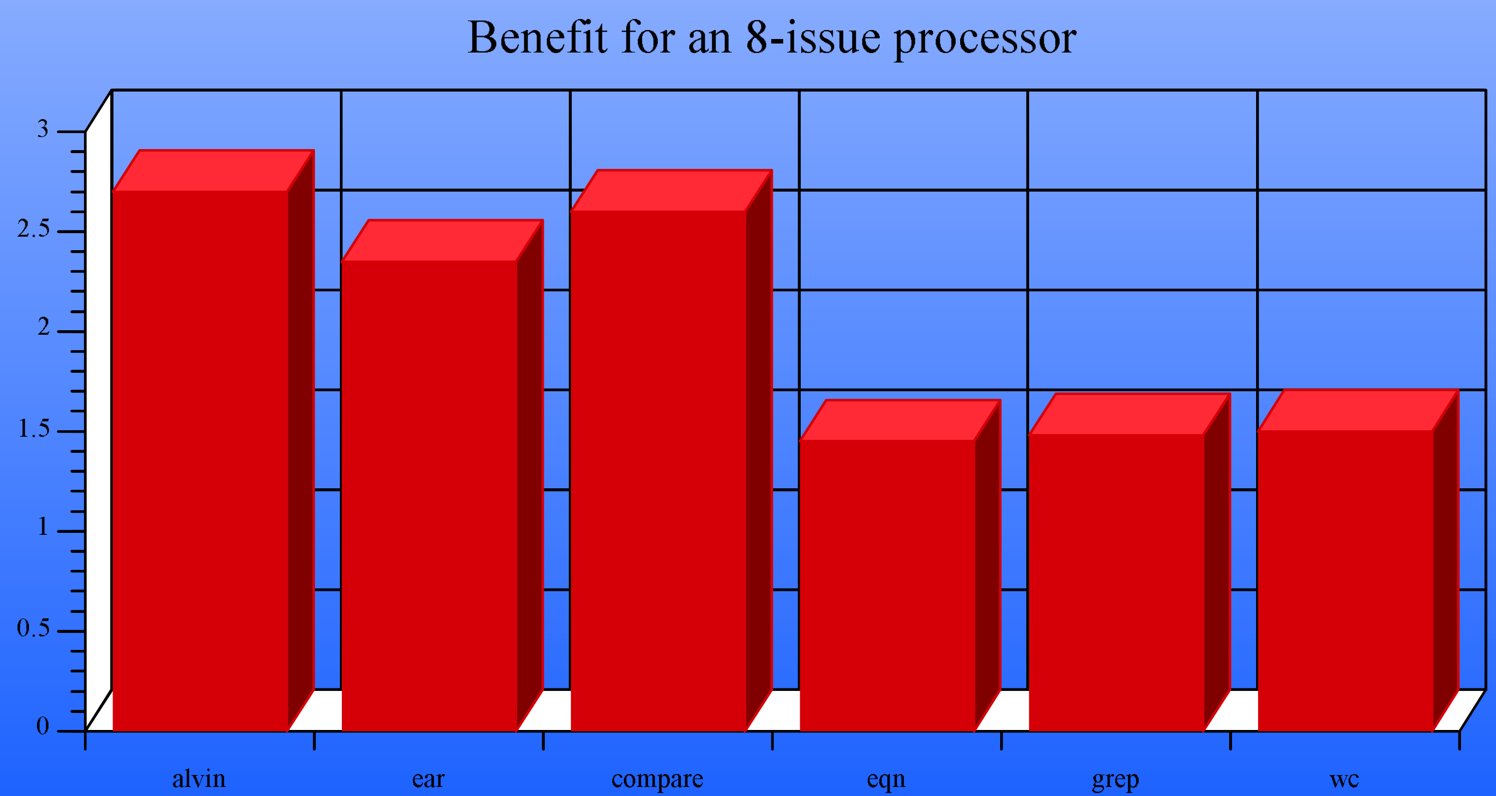
Memory Disambiguation Maintaining Dependence Arcs

- Targeting recompilation [Gallagher *et al.*]
  - Pass dependence arcs from source code level to machine code level
  - Maintain dependence arcs through optimizations and code transformations
  - Take advantage of dependence arcs in all aspects of machine level compilation

Memory Disambiguation Data speculation

- Targeting binary translation and compilation of difficult languages
  - Run-time disambiguation using branches [Nicolau *et al*.]
  - Speculative disambiguation using predication [Huang *et al*.]
  - Smart load [Schlansker *et al*.]
  - Memory Conflict Buffer [Chen et al.] [Gallagher et al]

Data Speculation Benefit



Data Speculation Generality of ILP Technology

- Confine the problem of memory disambiguation to the back-end.
- Provide ease entrance into difficult languages such as C and C++
- Lower technology requirement for binary translation
- A difficult concept for many!

Interprocedural Compilation

- Go beyond interprocedural analysis
- Exploit ILP across function boundaries
  - interprocedural innermost loops
  - interprocedural code optimization
  - interprocedural code motion

Interprocedural Compilation Interprocedural Regions

- Identify a region of code that tend to execute together
- Each region may span multiple functions
- Extract region code from functions and package them as a function
- Apply traditional intraprocedural optimization and scheduling techniques
- Under joint development between HPL and U. Illinois [Hank *et al.*]

VLIW:
Is it your fathers oldsmobile?

- Common Criticism on VLIW
  - Technology lacks generality
  - Do not support existing binaries
  - No scalable implementations
  - Excessive demand on the compiler

Cydrome Cydra-5 [Rau *et al.*]

- Compiler Technology
  - Modulo Scheduling
  - If-conversion
- Architecture Technology
  - Predicated Execution
  - Rotating Register File
  - Latency interlock

Multiflow Trace [Fisher *et al.*]

- Compiler Technology
  - Trace scheduling
  - Control speculation
  - Distributed register file allocation
- Architecture Technology
  - Non-trapping instructions
  - Partial predication

HPSm [Patt *et al.*]

- Wide instruction word (3-issue)
  - no dependence checking within wide word
- Out of order execution
- Branch prediction and speculative execution
- ISCA-86 (architecture)
- IEEE JSSC, Jan 1992 (VLSI design)

Instruction Set Architecture Change Are we due for another wave?

- VAX ISA lasted about 15 years.
- 68K ISA lasted about 15 years.
- All major RISC's are about 10 years old.
- 386 ISA is also about 10 years old.
- Can the industry avoid another major architecture change by the end of the decade?

Instruction Set Architecture Change The case of DEC Alpha

- X86 emulation not significantly more cost-effective than Pentium.
- No smooth software migration path from X86
  - no mix and match of Alpha and X86 code
- Raw speed alone does not make business.
- The bottomline is cost-effectiveness of running existing code and ease of software migration.

Instruction Set Architecture Change

The case of M.I.A. PowerPC

- X86 emulation not significantly more cost-effective than Pentium
- No support for smooth software migration from X86
- 68K emulation cost effective to take over Mac market

ISA Change
Where are the research results?

- Companies tend not fund research until it is too late.
- Government tends not to fund research needed in the foreseeable future.
- Hence, two major types of research prevail
  - over due research that should have been done a long time ago
  - never due research that will have to be redone when industry needs it

Successful ISA Change
Hwu's Law

- Opportunities come up every 3-5 years.
- Add the new ISA into a successful ISA.
- Expect 5 years of latency from customers and software vendors.

Outlook
The Hwu Prediction

- ILP microprocessors will sustain 4-6 instructions per cycle by 1997.
- They will be marketed as VLIW.
- They will be upward compatible with major current architectures.
- Binary translation will become more important than compilation.

Summary and Outlook

- ILP techniques have made tremendous progress in the past decade.
- ILP architecture and software technology have moved in as the mainstream.
- There is a wide range of new research opportunities looking ahead.
- Industry must do the next ISA generation today to sustain growth 5 years from now.