

Computer Science 703
Advance Computer Architecture
2010 Semester I

Lecture Notes 7

16Mar10

ILP: (Dynamic Scheduling),
Multiple Issue Processors

James Goodman



Lectures This Week

- Today: ILP: H&P Sections 2.4-2.5
 - Finish Dynamic Scheduling
 - Hardware-based Speculation
 - VLIW Processors
- Thursday: Memory Systems

Suggested (Online) Readings

- Wikipedia: “Cache memory”
http://en.wikipedia.org/wiki/CPU_cache
- Hill & Smith, “Evaluating Associativity in Caches”
ftp://ftp.cs.wisc.edu/markhill/Papers/toc89_cpu_cache_associativity.pdf
- Jon Stokes, “Understanding CPU caching and performance”
<http://arstechnica.com/old/content/2002/07/caching.ars>
- Hennessy & Patterson, “Eleven Advanced Optimizations of Cache Performance,” Section 5.2 from H&P, pp. 293-309.
http://books.google.co.nz/books?id=pqY13SWkA64C&pg=PA293&clpg=PA293&cdq=11+advanced+optimizations+cache+performance&source=bl&ots=0O9XEiFGRG&sig=pGojOwj3FomulP2wMjIzqAjuqmnM&chl=en&ei=4qaeS_TwjY_MsQOI5u2_Aw&sa=X&oi=book_result&ct=result&resnum=1&cved=0CAgQ6AEwAA#v=onepage&q=11%20advanced%20optimizations%20cache%20performance&f=false
Note: Book is available online at Google, but “total pages displayed will be limited.” In this case, that is pp. 238-298, thus including half this reference.

Reservation Station

- Op: Operation to perform on source operands
- Qj, Qk: Reservation stations supplying operands (zero indicates already received)
- Vj, Vk: Value of the source operands
- (A: information for memory address calculation)
- Busy: indicates the reservation station/functional unit are occupied
 - How the dispatch unit knows when a functional unit becomes available

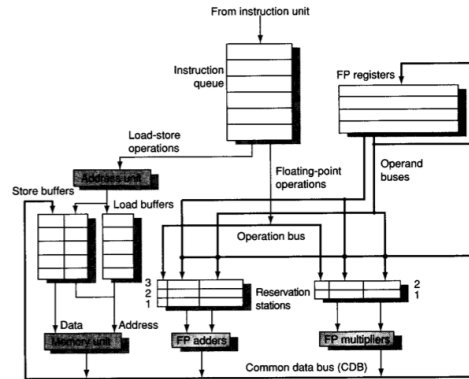


Figure 2.9 The basic structure of a MIPS floating-point unit using Tomasulo's algorithm. Instructions are sent from the instruction unit into the instruction queue from which they are issued in FIFO order. The reservation stations include the operation and the actual operands, as well as information used for detecting and resolving hazards. Load buffers have three functions: hold the components of the effective address until it is computed, track outstanding loads that are waiting on the memory, and hold the results of completed loads that are waiting for the CDB. Similarly, store buffers have three functions: hold the components of the effective address until it is computed, hold the destination memory addresses of outstanding stores that are waiting for the data value to store, and hold the address and value to store until the memory unit is available. All results from either the FP units or the load unit are put on the CDB, which goes to the FP register file as well as to the reservation stations and store buffers. The FP adders implement addition and subtraction, and the FP multipliers do multiplication and division.

Code Example

1. L.D F6, 32(R2)
2. L.D F2, 44(R3)
3. MUL.D F0, F2, F4
4. SUB.D F8, F2, F6
5. DIV.D F10, F0, F6
6. ADD.D F6, F8, F2

Exceeding the Flynn Limit

- Michael Flynn observed that every uniprocessor built or proposed had a maximum rate of execution of one instruction per clock cycle
- Major challenge: how to detect hazards and guarantee correctness while issuing multiple instructions simultaneously?

Multiple Issue Processors

Common name	Issue structure	Hazard detection	Scheduling	Distinguishing characteristic	Examples
Superscalar (static)	dynamic	hardware	static	in-order execution	mostly in the embedded space: MIPS and ARM
Superscalar (dynamic)	dynamic	hardware	dynamic	some out-of-order execution, but no speculation	none at the present
Superscalar (speculative)	dynamic	hardware	dynamic with speculation	out-of-order execution with speculation	Pentium 4, MIPS R12K, IBM Power5
VLIW/LIW	static	primarily software	static	all hazards determined and indicated by compiler (often implicitly)	most examples are in the embedded space, such as the TI C6x
EPIC	primarily static	primarily software	mostly static	all hazards determined and indicated explicitly by the compiler	Itanium

Figure 2.18 The five primary approaches in use for multiple-issue processors and the primary characteristics that distinguish them. This chapter has focused on the hardware-intensive techniques, which are all some form of superscalar. Appendix G focuses on compiler-based approaches. The EPIC approach, as embodied in the IA-64 architecture, extends many of the concepts of the early VLIW approaches, providing a blend of static and dynamic approaches.