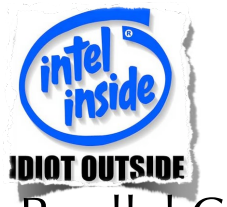


2110412 Parallel Computer Architecture: Multi-core Tech.

2: Approaches to ILP

Krerk Piromsopa, Ph.D.
Department of Computer Engineering
Chulalongkorn University

Today's Goal:

- Pipeline (Revisit)
- Instruction Level Parallelism
- Dynamic Scheduling
  - Scoreboarding, Tomasulo Algorithm
- Multiple Issue
  - SuperScalar, VLIW
- Speculation

Pipeline

Pipeline CPI = Ideal pipeline CPI + Structural stalls + Data Hazard stalls + Control stalls

What is ILP?

- basic block
  - a straight-line code sequence with no branches in except to the entry and no branches except at the exit
- dynamic branches is between 15-20%
- So, we want ILP across multiple basic blocks.
- Instruction-level parallelism (ILP) is a measure of how many of the operations in a computer program can be performed simultaneously (from wikipedia)

How about loop?

```
for (i = 0; i < 1024; i++)
  C[i] = A[i]*B[i];
```

Vector?

```
for (i = 0; i < 1024; i+=4)
   C[i:i+3] = A[i:i+3]*B[i:i+3];
```


Data Dependences

- 1. e = a + b
- 2. f = c + d
- 3. g = e * f

Name Dependences

- antidepence
  - instruction j writes a register or memory location that instruction i reads
- output dependence
  - instruction i and instruction j write the same register or memory location

Data Hazards

- RAW
- WAW
- WAR
- Solution: preserve order

Control Dependences

- if p1
  - S1;
- if p2
  - S2;

- An instruction that is control dependent on a branch cannot be moved before the branch
- An instruction that is not control dependent on a branch canot be moved after the branch

Dynamic Scheduling

- Hardware rearranges the instruction execution to reduce the stalls while maintaining data flow and exception behavior.
- Out-of-order execution (introduce WAR and WAW)
- Advantages
  - runtime binding (some dependence may not known at compile time)
- Issues
  - Imprecise exceptions

Out-of-order

- Split ID into two stages
  - Issue
  - Read Operands

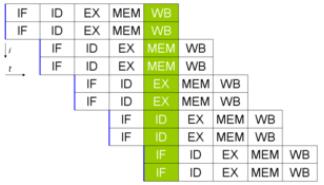
Tomasulo's Algorithm

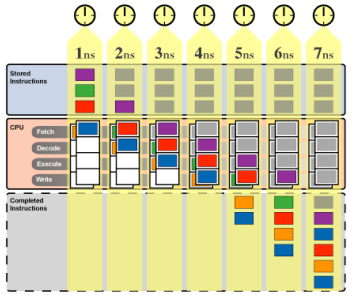
- Register renaming --- reservation stations
- See demo

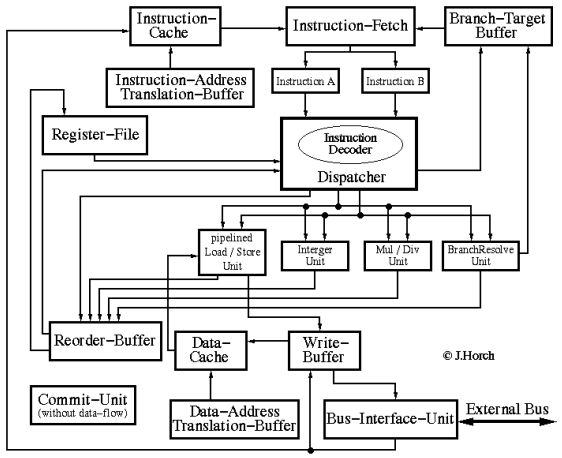
Multiple Issue

- Allow multiple instructions to issue in a clock cycle.
- Two flavors
  - superscalar processors
    - static, dynamic, speculative
  - VLIW (Very long instruction word)

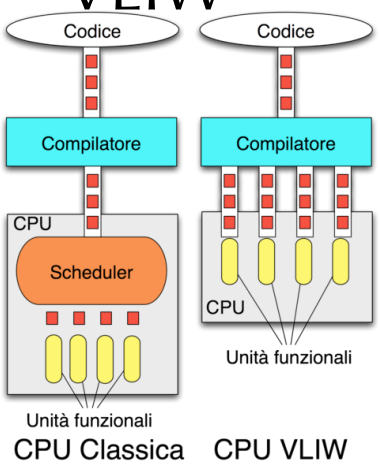
SuperScalar







VLIW



- Compiler Approach (Itanium)
- No backward compatibility
- f12=f0*f4, f8=f8+f12, f0=dm(i0,m3), f4=pm(i8,m9);

Speculation

- aka speculative execution
- Guess next instruction and start immediately
  -

Multiple Issue with Speculation