More on Pipelining and Pipelines in Real Machines

CS 333

Fall 2006

Main Ideas

- Data Hazards
  - RAW
  - WAR
  - WAW
- More pipeline stall reduction techniques
  - Branch prediction
    - » static
    - » dynamic
      - bimodal branch prediction
- Multiple-issue
  - static
  - dynamic (superscalar)
- Pipelines in real processors - Pentium 4, AMD Athlon (next class)

Pipelining

- Exploits **parallelism** between instructions
  - **Instruction-level parallelism (ILP)**
  - Gain performance
- Pipelining overlaps more instructions
  - Subdivides instruction execution into subtasks

Data Hazards

- Read after Write (RAW)

```
add r0, r1, r2
sub r4, r3, r0
```

- Write after Write (WAW)

```
add r0, r1, r2
sub r0, r4, r5
```

- Write after Read (WAR)

```
add r2, r1, r0
sub r0, r3, r4
```

WAW and WAR
Only possible if instructions can be executed in parallel or out-of-order

register renaming

Control Hazards

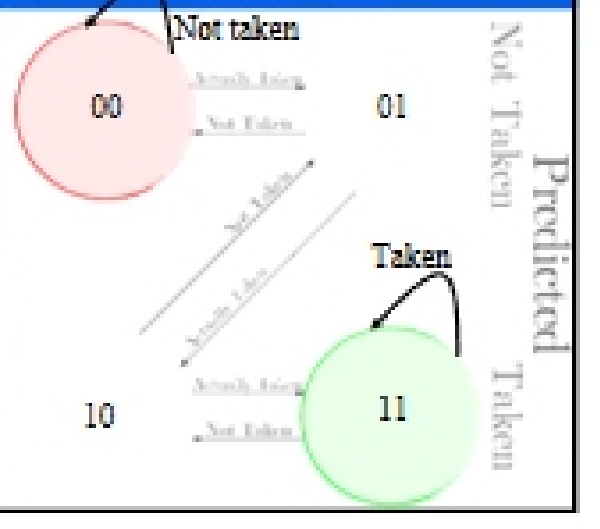
- Resolution of branch condition does not occur until **MEM** stage
  - Causes stalls
- Approaches to handling:
  - Stall (pipeline bubbles)
  - Delayed branch slot
  - Static prediction
    - » Does not rely on runtime information
    - » various approaches:
      - always taken (or, always not taken)
      - backwards taken, forwards not taken (loops)

If branch is mispredicted, need to undo all instructions in the pipeline along the incorrect path (need extra resources)

Dynamic Branch Prediction

- Use execution behavior to make a prediction

bimodal branch prediction
- 2 bit counters
- use recent branch behavior to predict future



Multiple Issue

- Another approach to increasing instruction level parallelism
- Launch many instructions in each stage
  - Need more resources (instead of 1 washer and dryer, may need multiple washers and dryers, AND more people to help fold clothes)
  - Disadvantage
    - » Need additional resources to keep all resources busy

Multiple Issue

- Additional tasks
  - Packaging instructions into **issue slots**
    - » **how many instructions?**
    - » **which instructions?**
  - Handling data and control hazards

Static Multiple Issue

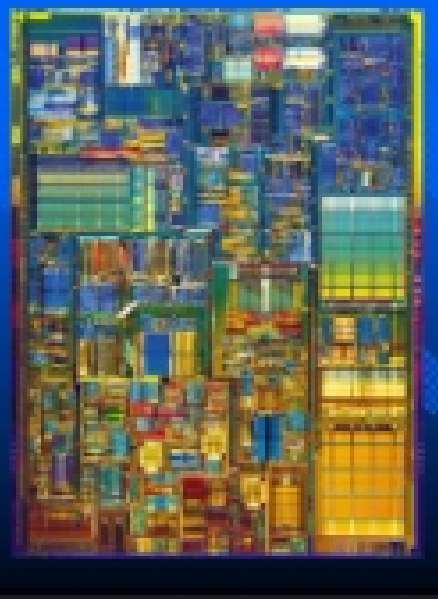
- Uses **compiler** to determine:
  - **issue packets**
  - handle data and control hazards
    - » **register renaming**
      - remove WAR and WAW dependences

add **r0**, r1, r2

sub **r0**, r4, r5

Rename r0 in sub to r3

Intel Pentium 4 Processor



Outline

- Pentium 4 – 20 stages
  - Instruction Set Architecture
  - Instruction Stream
- Pentium 4 revisions
  - Northwood (1/2002) – 21 stages
  - Prescott (2/2004) – 31 stages