Superscalar Machines

- Increasing pipeline length eventually leads to diminishing returns
  - longer pipelines take longer to re-fill
  - data and control hazards lead to increased overheads, removing any performance advantage
  - yet there is still space for more circuitry on-chip
- An alternative is the multiple-issue processor
  - i.e. issue more than one instruction each clock cycle
    • that is, start more than one instruction
- Such processors are called Superscalar processors

Characteristics of superscalar processors

- Superscalar processors issue
  - more than one instruction each cycle
  - the number of instructions issued will depend on the instructions in the instruction stream
  - instructions are often re-ordered to fit the processor architecture better
  - typically, between 1 and 8 instructions can be issued each cycle
- The simplest possibility is a machine which issues
  - one integer instruction
  - one floating-point instruction
    • per clock cycle
- ...but this only gives appreciable improvements if there are many FP instructions to be executed.
Ordering instructions 1

• Instructions are produced in a particular order by the compiler, but need not (necessarily) be executed in that order.
• They may be
  – issued in order and completed in order
  – issued in order, but completed out of order
  – issued out-of-order, but completed in order
  – issued out of order and completed out of order.

Ordering instructions 2

• The best order for instructions in a particular superscalar architecture depends on
  – the architecture itself
  – the precise dependencies between instructions
• the actual order they are executed in may be
  – set up by the compiler
    • in which case it must “know” the architecture
    • complex code-generating compiler
  – or the instructions may be reordered by the CPU itself
    • many instructions are fetched, but the precise order of execution is decided at execution time
    • this is more common
Register renaming

- one of the main techniques for dealing with data hazards is register renaming
  - registers are dynamically allocated by the hardware
    - out of a register set which is larger than the programmer’s model of the architecture
  - subsequently executing instructions use whichever register is correct for them.
  - Usually the actual register which a programmer’s register refers to is re-allocated from time to time

- Limits to multiple issue machines
  - there are inherent limits to the extent to which sequential instructions can be executed concurrently
    - these show up as data and control hazards
  - there are limits to the amount of underlying hardware that can be produced
    - these show up as structural hazards

Branch Prediction

- Since branches can cause major delays in pipelines, some machines attempt to predict what the outcome of a branch will be
- Like the 68040, the Pentium attempts to limit the problems which branches cause by fetching instructions from both the target address and the branch address
- However, it predicts which will occur
  - from what has happened previously
  - and (speculatively) executes from that address.
- Other techniques can use the compiler to predict whether the branch is more or less likely to be taken
Superscalar architecture DLX: superscalar version

Superscalar DLX Features

- Pipelined, superscalar
  - Two instructions per clock
  - Branch-Target buffer
  - Reorder-Buffer to commit instructions in program order
  - Precise exception processing

- Four execution units with Reservation-Station
  - Branch-Resolve unit
  - Arithmetic-Logic unit
  - Multiply-Divide unit
  - Load-Store unit

- Write-Buffer
Example 2: DEC ALPHA AXP 21064

- DEC ALPHA AXP 21064
  - superscalar architecture with
    - multiple functional units
    - 7 stage 5ns/stage integer pipeline
  - the CPU can issue simultaneously
    - any load/store in parallel with any operate
    - an integer operate in parallel with a FP operate
    - a floating operate and a floating branch
    - an integer operate and an integer branch
  - the most recent version of the ALPHA can issue 4 instructions simultaneously.
    - 4 integer execution units
    - 2 FP execution units

Pentium architecture

- Superscalar architecture
  - 2 independent integer pipelines
  - one Floating point pipeline
  - but control unit can issue either 2 integer instructions or 1 (occasionally 2) floating point instructions

- Branch prediction by loading code to cache from target address
- separate on-chip instruction and data caches
  - both 8Kbytes long

- Integer pipelines are not identical
  - u pipeline includes barrel shifter, but v is simpler (and can't execute all integer instructions)

- Some elements of microcoded control using control ROM.
Pentium 3

- Superscalar processor
  - 3 integer pipelines
  - 2 FP pipelines
- 16KB instruction cache
- 16KB data cache
- L2 cache on chip
- Microcode system to deal with CISC instructions
  - some x86 instructions -> 1 microOp
  - others into 2 or 3 microOps
  - others into many more
Pentium P4

- Similar to P3, but designed for ultra-high clock rates
- Quad-pumped ("pipelined") system bus
- 20 stage pipeline (!)
  - P3 has 10
- Sophisticated branch prediction using BTB
  - crucial for performance with long pipelines.
- Ultrafast integer ALU (2x clock speed) for simple integer operations
- Execution trace Cache of uOPs
  - instead of Instruction cache
- Smaller Data cache
  - to allow 1-cycle access
- New FP instructions
  - but is anyone using them?