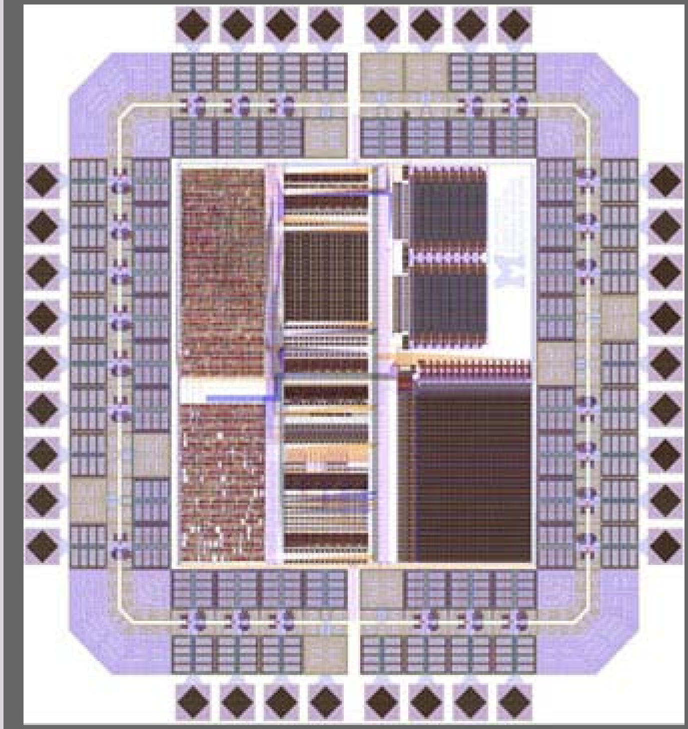


Course Name	Parallel Computer Architecture
Department	Computer Science and Engineering IIT Kanpur
Instructor	Dr. Mainak Chaudhuri

Next

Module 1: "Multi-core: The Ultimate Dose of Moore's Law"

Lecture 1: "Evolution of Processor Architecture"

The Lecture Contains:

Multi-core: The Ultimate Dose of Moore’s Law

A gentle introduction to the multi-core landscape as a tale of four decades of glory and success

- Mind-boggling Trends in Chip Industry
- Agenda
- Unpipelined Microprocessors
- Pipelining
- Pipelining Hazards
- Control Dependence
- Data Dependence
- Structural Hazard
- Out-of-order Execution
- Multiple Issue
- Out-of-order Multiple Issue

Previous Next

Mind-boggling Trends in Chip Industry:

- Long history since 1971
  - Introduction of Intel 4004
  - http://www.intel4004.com/
- Today we talk about more than one billion transistors on a chip
  - Intel Montecito (in market since July'06) has 1.7B transistors
- Die size has increased steadily (what is a die?)
  - Intel Prescott: 112mm2, Intel Pentium 4EE: 237 mm2, Intel Montecito: 596 mm2
- Minimum feature size has shrunk from 10 micron in 1971 to 0.045 micron today

Agenda:

- Unpipelined microprocessors
- Pipelining: simplest form of ILP
- Out-of-order execution: more ILP
- Multiple issue: drink more ILP
- Scaling issues and Moore's Law
- Why multi-core
  - TLP and de-centralized design
- Tiled CMP and shared cache
- Implications on software
- Research directions

Unpipelined Microprocessors:

- Typically an instruction enjoys five phases in its life
  - Instruction fetch from memory
  - Instruction decode and operand register read
  - Execute
  - Data memory access
  - Register write
- Unpipelined execution would take a long single cycle or multiple short cycles
  - Only one instruction inside processor at any point in time

Previous Next

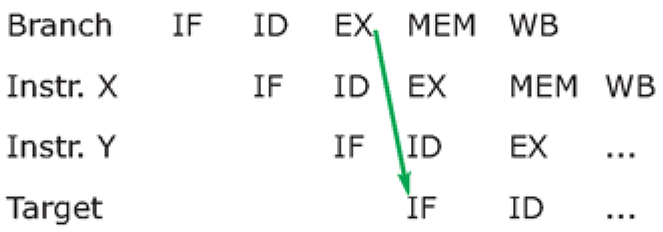
Pipelining:

- One simple observation
  - Exactly one piece of hardware is active at any point in time
- Why not fetch a new instruction every cycle?
  - Five instructions in five different phases
  - Throughput increases five times (ideally)
- Bottom-line is
  - If consecutive instructions are independent, they can be processed in parallel
  - The first form of instruction-level parallelism (ILP)

Pipelining Hazards:

- Instruction dependence limits achievable parallelism
  - Control and data dependence (aka hazards)
- Finite amount of hardware limits achievable parallelism
  - Structural hazards
- Control dependence
  - On average, every fifth instruction is a branch (coming from if-else, for, do-while,…)
  - Branches execute in the third phase
- Introduces bubbles unless you are smart

Control Dependence:



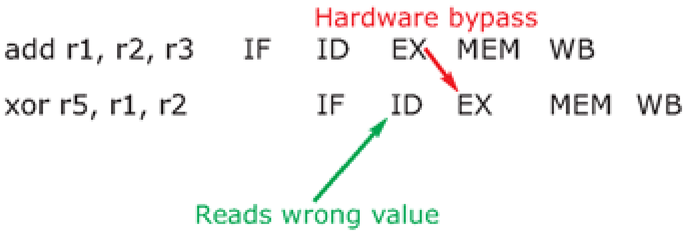
Branch	IF	ID	EX	MEM	WB	
Instr. X		IF	ID	EX	MEM	WB
Instr. Y			IF	ID	EX	...
Target				IF	ID	...

What do you fetch in X and y slots?

Options: nothing, fall-through, learn past history and predict (today best predictors achieve on average 97% accuracy for SPEC2000)

Previous Next

Data Dependence:

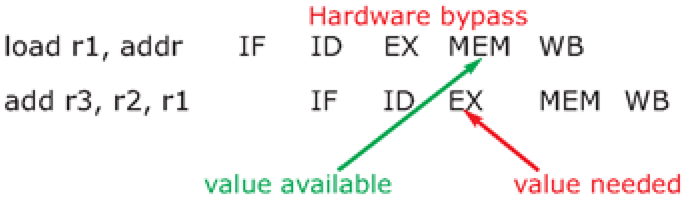

Hardware bypass
add r1, r2, r3 IF ID EX MEM WB
xor r5, r1, r2 IF ID EX MEM WB
Reads wrong value

Take three bubbles?

Back-to-back dependence is too frequent

Solution: hardware bypass paths

Allow the ALU to bypass the produced value in time: not always possible

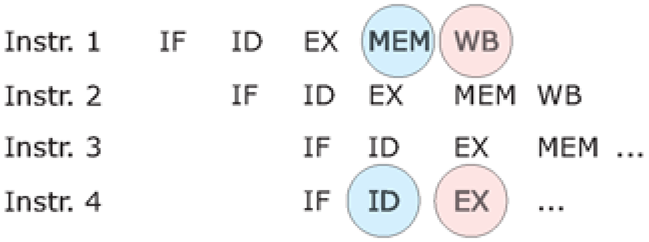


Need a live bypass! (requires some negative time travel: not yet feasible in real world)

No option but to take one bubble

Bigger problems: load latency is often high; you may not find the data in cache

Structural Hazard:



Usual solution is to put more resources

Previous Next

Out-of-order Execution:

Previous Next

Multiple Issue:

Out-of-order Multiple Issue:

- Some hardware nightmares
  - Complex issue logic to discover independent instructions
  - Increased pressure on cache
- Impact of a cache miss is much bigger now in terms of lost opportunity
- Various speculative techniques are in place to “ignore” the slow and stupid memory
- Increased impact of control dependence
- Must feed the processor with multiple correct instructions every cycle
- One cycle of bubble means lost opportunity of multiple instructions
- Complex logic to verify

Previous Next