

Modern Processor Design: Superscalar and Superpipelining

Note: Some of the material in this lecture are
COPYRIGHT 1998 MORGAN KAUFMANN PUBLISHERS, INC.
ALL RIGHTS RESERVED.

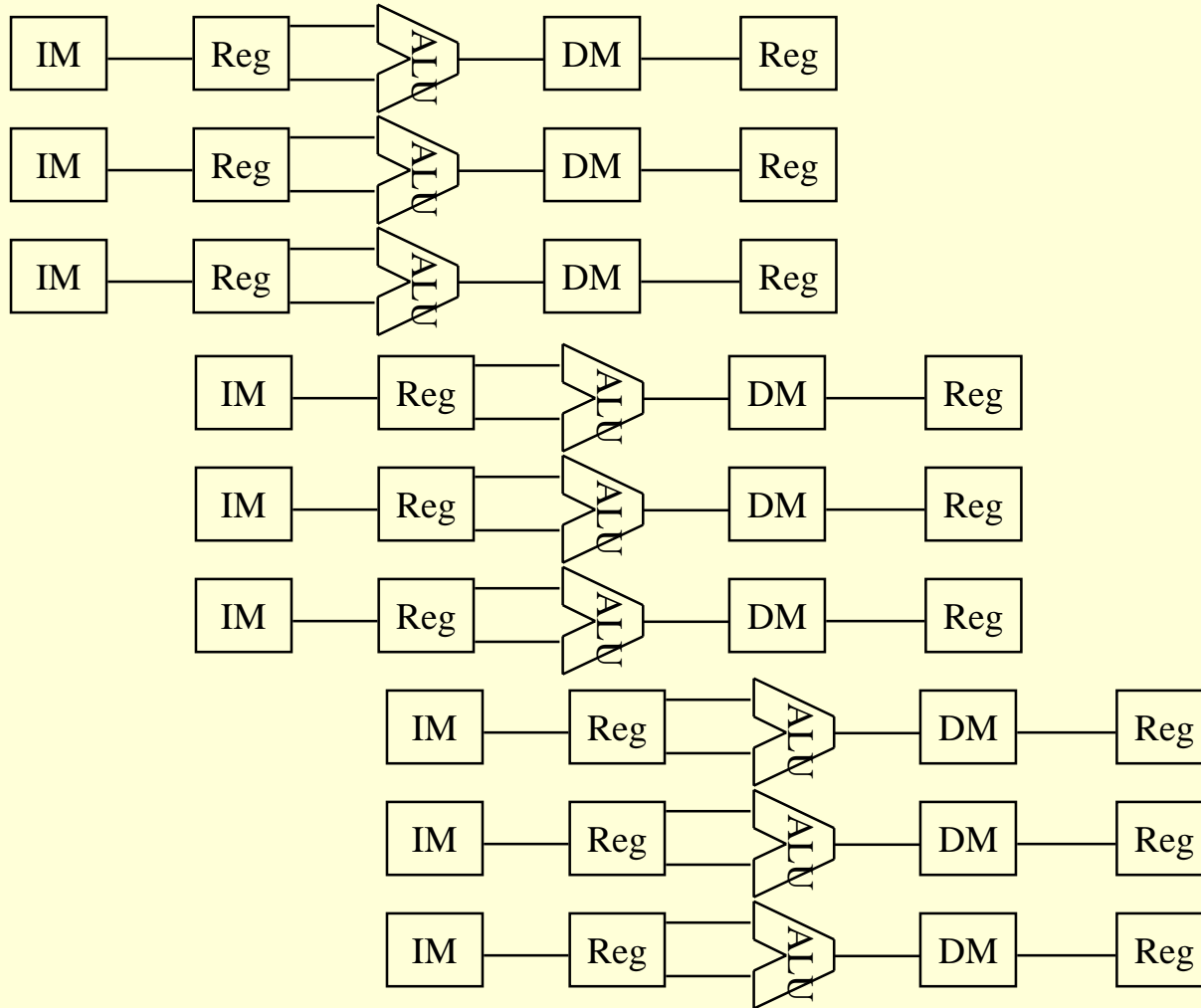
Figures may be reproduced only for classroom or personal education
use in conjunction with our text and only when the above line is included.

Today's processors

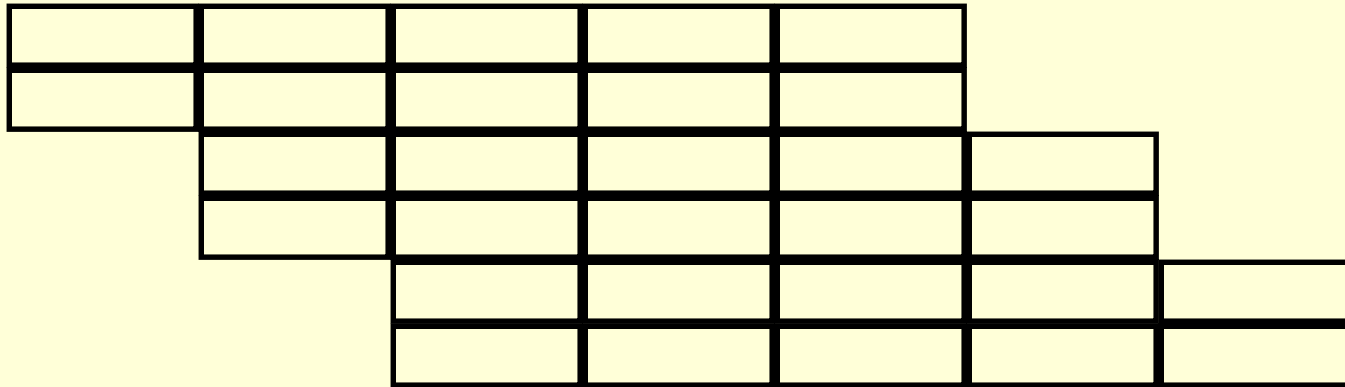
Not fundamentally different than the techniques we discussed, but ...

- Deeper pipelines (superpipelining)
 - Example: 20 stages in Pentium 4.
- Pipelining is combined with:
 - superscalar processing:
 - issuing more than 1 instruction per cycle (3 or 4 is common)
 - out-of-order execution
 - allowing instructions to jump ahead of others in line
 - VLIW (very long instruction word)
 - packaging instruction in group, always executed together

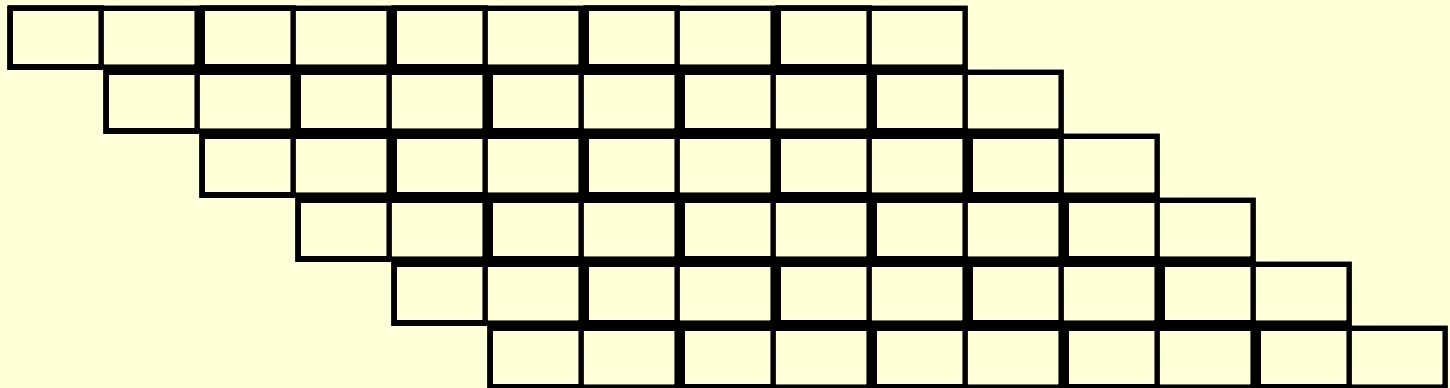
Superscalar Execution



Superscalar vs. superpipelined



(multiple instructions in the same stage, same CR as scalar)



(more total stages, faster clock rate)

Superscalar Execution

To execute, say, four instructions in the same cycle, we must find four independent instructions.

- In a VLIW processor, the compiler produces groups of four that are guaranteed to be independent.
- In an in-order superscalar processor, the hardware executes as many of the next instructions (up to four) that it can, making sure that they are independent.
- In an out-of-order processor, the hardware finds four (not necessarily consecutive) instructions that are independent.

What do you think are the tradeoffs?

Superscalar Scheduling

- Assume the “modest superscalar processor” (in-order; can execute one R-type and one I-type together)

```
lw $6, 36($2)
```

```
add $5, $6, $4
```

```
lw $7, 1000($5)
```

```
sub $9, $12, $8
```

```
sw $5, 200($6)
```

```
add $3, $9, $9
```

```
and $11, $5, $6
```

When does each instruction begin execution?

Out-of-order Scheduling

- Starts execution of an instruction as soon as all of its dependences are satisfied, even if prior instructions are stalled.

lw \$6, 36(\$2)

add \$5, \$6, \$4

lw \$7, 1000(\$5)

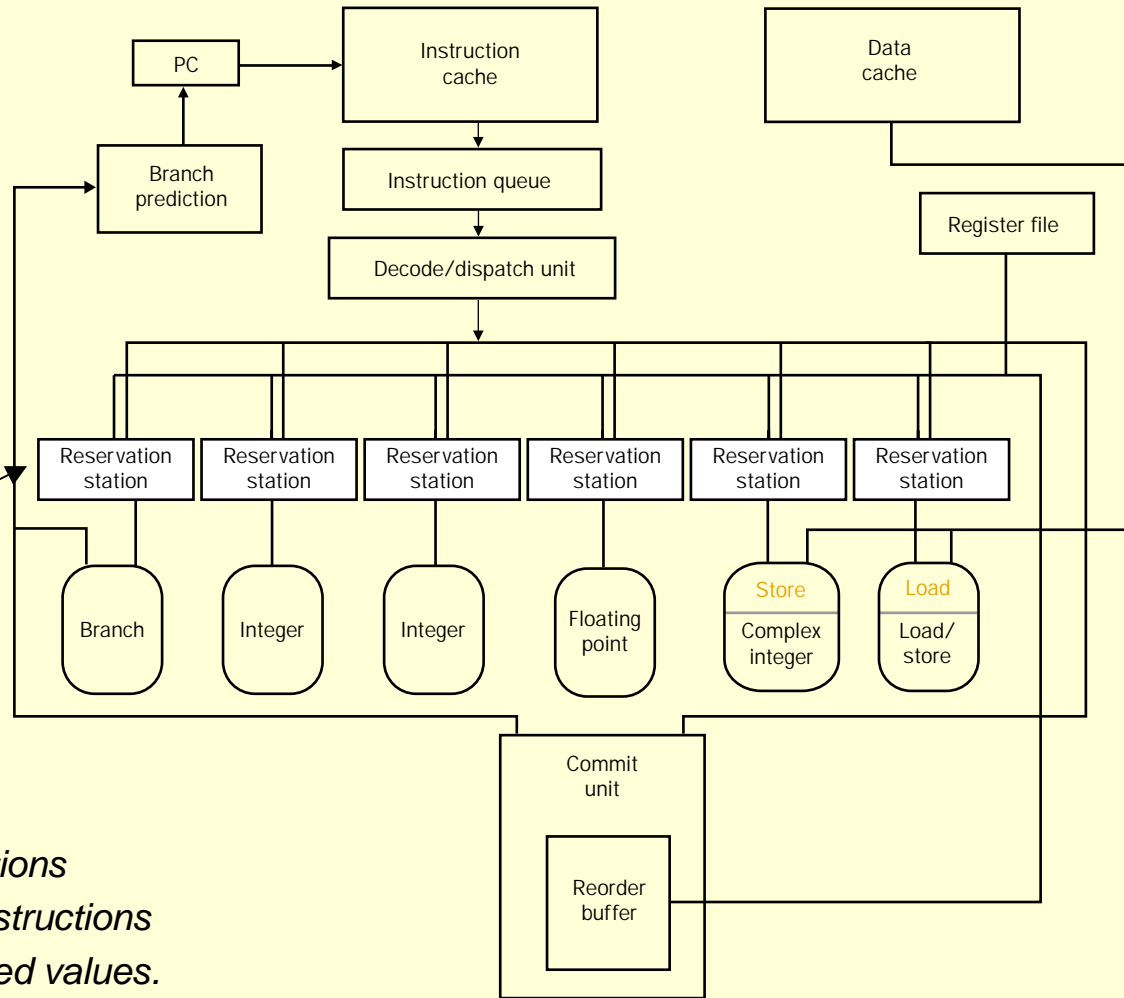
sub \$9, \$12, \$8

sw \$5, 200(\$6)

add \$3, \$9, \$9

and \$11, \$5, \$6

PowerPC 604, Intel Pentium



Reservation stations hold decoded instructions waiting for needed values.

Pipelining -- Key Points

- Execution time
= Number of instructions * CPI * cycle time
- *Data hazards* and *branch hazards* prevent CPI from reaching 1.0, but *forwarding* and *branch prediction* get it pretty close.
- To improve performance we must reduce cycle time (superpipelining) or reduce CPI below one (superscalar and VLIW).

Computer(s) of the Day

- A brief history of supercomputers:
 - 70's & 80's: Vector computers (usually CRAY)
 - Set up pipeline (e.g. $y[i] = a * x[i] + b * y[i-1]$)
 - Once it gets going, delivers one result per cycle
 - Very fast memories (usually SRAM)
 - 90's: Supercomputers based on "commodity processors"
 - SMP's "Shared Memory" or "Symmetric MultiProcessor"
 - Many processors sharing same memory (address space)
 - Communicating by writing/reading to common locations
 - MPP's ("Massively Parallel Processors") or "Multicomputer"
 - Processors have separate memories
 - Communicate via I/O (message passing)
 - Both styles are harder to program than Vector machines, but give more FLOPS per \$\$.

Computer(s) of the Day

- And today we have ... the “SMP cluster”
 - Multiple processors sharing memory within a “node”.
 - Multiple nodes communicating via message passing.
 - the worst of both SMP & MPP (from programmers perspective)
- Ten fastest computers (on “Top500” list)
 - ... based on Linpack benchmark performance
 - 5 are IBM SP's,
 - 2 DEC Alpha clusters,
 - 1 cluster of Intel Pentiums,
 - 1 SGI cluster
 - 1 Hitachi vector

Computer(s) of the Day

- IBM “Blue Horizon” at UCSD’s San Diego Supercomputer Center (SDSC)
 - Installed (2/2000); 8th fastest in world (now 18th).
 - Fastest available to academic scientists.
 - Power3 processor – 375 MHz, 4 Float Ops/cycle
 - 8 processors per node
 - 144 nodes → 1152 processors → 1.7 TeraFLOP peak
 - 4 (or more) GByte DRAM/node → 576 GB memory
 - 5.1 TeraBytes of disk storage
 - Used to simulate colliding galaxies, beating heart, chemical activity in brain; to look for patterns in DNA; to factor large numbers; etc.