

## Advanced Pipelining

CSE 141

Dean Tullsen

## Pipelining and Exceptions

- Again, exceptions represent another form of control dependence.
- Therefore, they create a potential branch hazard
- Exceptions must be recognized early enough in the pipeline that subsequent instructions can be flushed before they change any permanent state.
- We also have issues with handling exceptions in the correct order and “exceptions” on speculative instructions.
- Exception-handling that always correctly identifies the offending instruction is called *precise interrupts*.

CSE 141

Dean Tullsen

## Pipelining in Today’s Most Advanced Processors

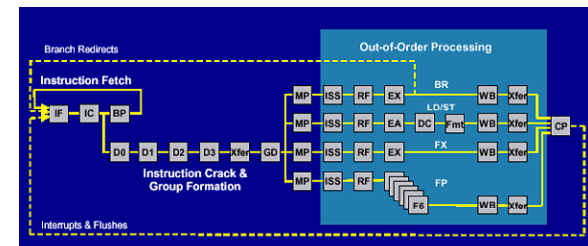
- Not fundamentally different than the techniques we discussed
- Deeper pipelines
- Pipelining is combined with
  - **superscalar** execution
  - **out-of-order** execution
  - **VLIW** (very-long-instruction-word)

CSE 141

Dean Tullsen

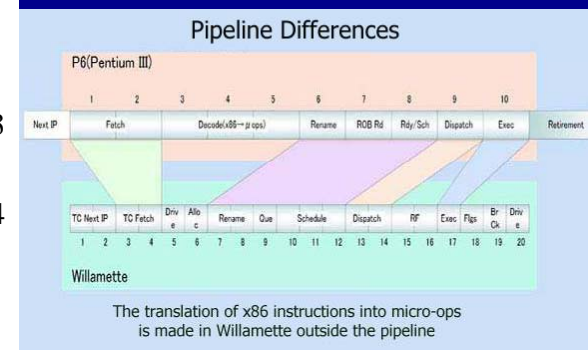
## Deeper Pipelines

- Power 4



- Pentium 3

- Pentium 4

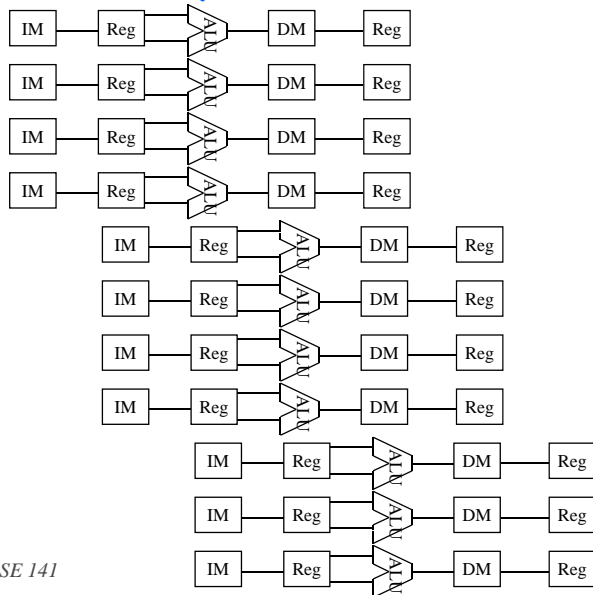


The translation of x86 instructions into micro-ops is made in Willamette outside the pipeline

CSE 141

Tullsen

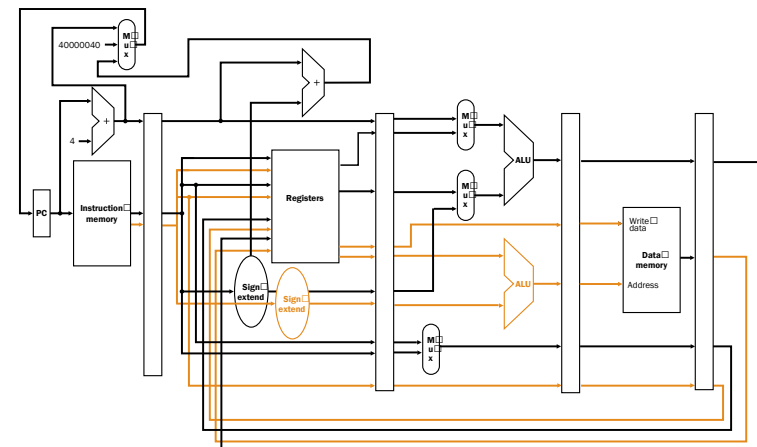
## Superscalar Execution



CSE 141

Dean Tullsen

## A modest superscalar MIPS



- what can this machine do in parallel?
- what other logic is required?
- Represents earliest superscalar technology (eg, circa early 1990s)

CSE 141

Dean Tullsen

## Superscalar Execution

- To execute four instructions in the same cycle, we must find four independent instructions
- If the four instructions fetched are guaranteed by the compiler to be independent, this is a *VLIW* machine
- If the four instructions fetched are only executed together if hardware confirms that they are independent, this is an *in-order superscalar* processor.
- If the hardware actively finds four (not necessarily consecutive) instructions that are independent, this is an *out-of-order superscalar* processor.
- What do you think are the tradeoffs?

CSE 141

Dean Tullsen

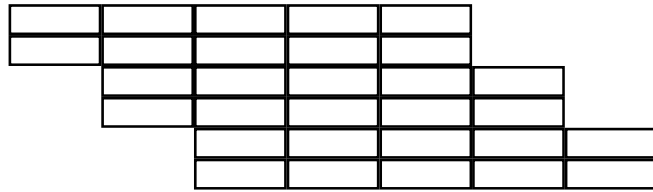
## Superscalar Scheduling

- assume in-order, 2-issue, ld-store followed by integer
  - lw \$6, 36(\$2)
  - add \$5, \$6, \$4
  - lw \$7, 1000(\$5)
  - sub \$9, \$12, \$5
- assume 4-issue, any combination (VLIW?)
  - lw \$6, 36(\$2)
  - add \$5, \$6, \$4
  - lw \$7, 1000(\$5)
  - sub \$9, \$12, \$5
  - sw \$5, 200(\$6)
  - add \$3, \$9, \$9
  - and \$11, \$7, \$6
- When does each instruction begin execution?

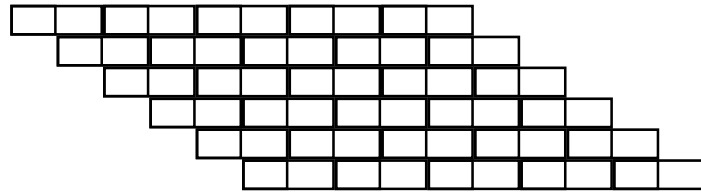
CSE 141

Dean Tullsen

## Superscalar vs. superpipelined



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



(more total stages, faster clock rate)

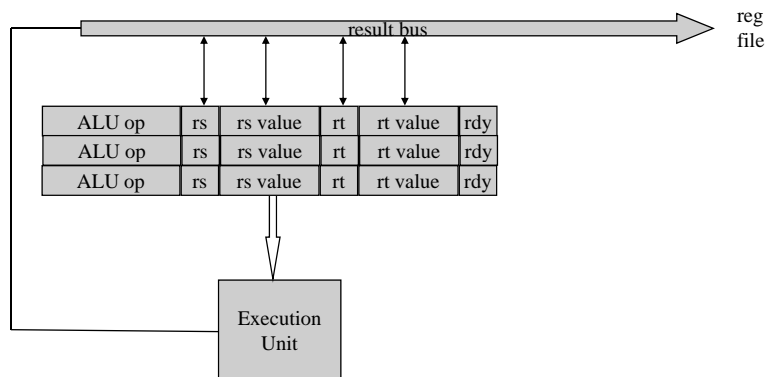
## Dynamic Scheduling or Out-of-Order Scheduling

- Issues (begins execution of) an instruction as soon as all of its dependences are satisfied, even if prior instructions are stalled. (assume 2-issue, any combination)

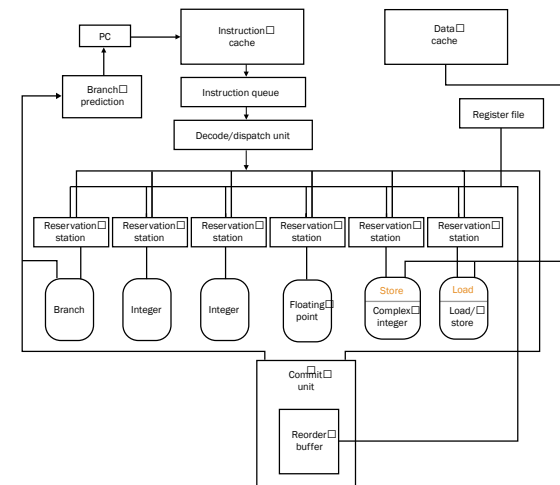
```
lw $6, 36($2)
add $5, $6, $4
lw $7, 1000($5)
sub $9, $12, $8
sw $5, 200($6)
add $3, $9, $9
and $11, $5, $6
```

## Reservation Stations

- are a mechanism to allow dynamic scheduling (out of order execution)



## PowerPC 604, Pentium Pro (II, III)



## Pentium 4

- Deep pipeline
- Dynamically Scheduled (out-of-order scheduling)
- Trace Cache
- *Simultaneous Multithreading* (HyperThreading)

### Basic Pentium® III Processor Misprediction Pipeline

1	2	3	4	5	6	7	8	9	10
Fetch	Fetch	Decode	Decode	Decode	Rename	ROB Rd	Rdy/Sch	Dispatch	Exec

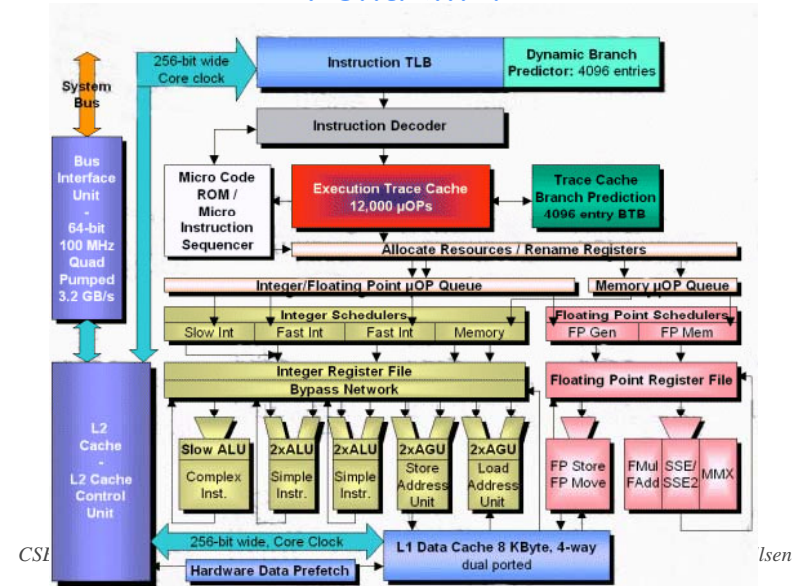
### Basic Pentium® 4 Processor Misprediction Pipeline

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
TC Nxt IP	TC Fetch	Drive	Alloc	Rename	Que	Sch	Sch	Sch	Disp	Disp	RF	RF	Ex	Flgs	Br Ck	Drive			

CSE 141

Dean Tullsen

## Pentium 4



CSE

Isen

## Modern (Pre-Multicore) Processors

- Pentium II, III – 3-wide superscalar, out-of-order, 14 integer pipeline stages
- Pentium 4 – 3-wide superscalar, out-of-order, simultaneous multithreading, 20+ pipe stages
- AMD Athlon, 3-wide ss, out-of-order, 10 integer pipe stages
- AMD Opteron, similar to Athlon, with 64-bit registers, 12 pipe stages, better multiprocessor support.
- Alpha 21164 – 2-wide ss, in-order, 7 pipe stages
- Alpha 21264 – 4-wide ss, out-of-order, 7 pipe stages
- Intel Itanium – 3-operation VLIW, 2-instruction issue (6 ops per cycle), in-order, 10-stage pipeline

CSE 141

Dean Tullsen

## Latest Developments – Multicore Processors

- IBM Power 4, 5, 6
  - Power 4 dual core
  - Power 5 and 6 dual simultaneous multithreading (2 threads each) cores
- Sun Niagara
  - 8 cores, 4 threads/core (32 threads).
  - Simple, in-order, scalar cores.
- Sun Niagara 2
  - 8 cores, 8 threads/core.
- Intel Quad Core Xeon
- AMD Quad Core Opteron
- Intel Nehalem
  - Up to 8 cores, each core SMT (2 threads)

CSE 141

Dean Tullsen

## Pipelining -- Key Points

- $ET = \text{Number of instructions} * CPI * \text{cycle time}$
- *Data hazards* and *branch hazards* prevent CPI from reaching 1.0, but *forwarding* and *branch prediction* get it pretty close.
- Data hazards and branch hazards need to be detected by hardware.
- Pipeline control uses combinational logic. All data and control signals move together through the pipeline.
- Pipelining attempts to get CPI close to 1. To improve performance we must reduce CT (superpipelining) or CPI below one (superscalar, VLIW).