

You have two hours (120 minutes) to complete this test.

20pts 1. Consider the iron law of performance. Note that we have used two versions of the Law.

RISC: $1/\text{Performance} = \text{Cycles / Instruction} * \text{Seconds / Cycle} * \text{Instructions / Task}$

4pts each

a. How did RISC attempt to improve the first term (CPI)?

Pipelining - to get close to 1 CPI

b. How did RISC attempt to improve the second term (SPC)?

Simplifying ctrl. logic

c. Would you expect RISC to make the third term better or worse?

worse - each instrh does less work

Superscalar: $\text{Performance} = \text{Instructions / Cycle} * \text{Cycles / Second} * \text{Task / Instructions}$

d. How did Superscalar attempt to improve the first term (IPC)?

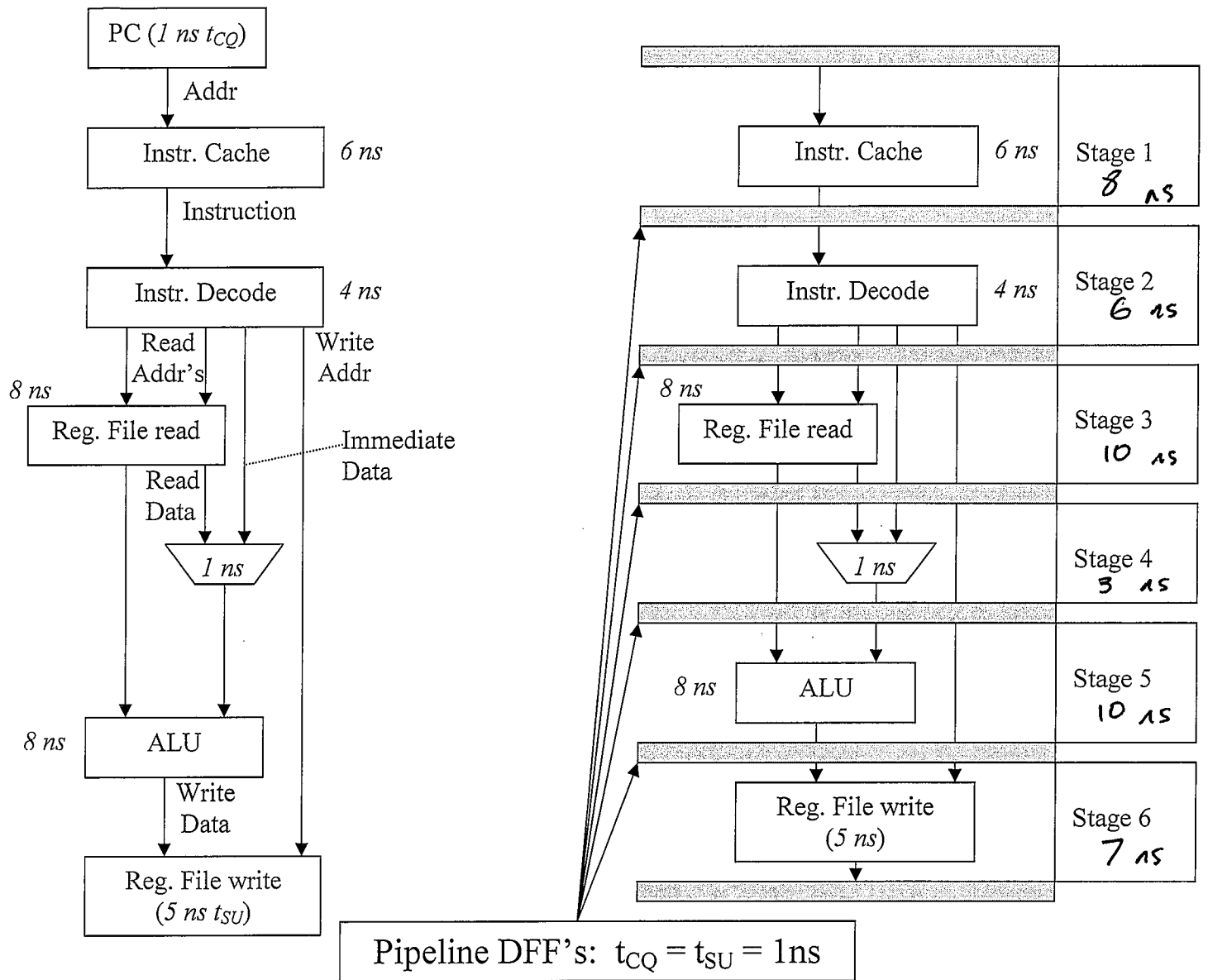
By adding multiple hardware elements in parallel

e. In general, superscalar's internal control logic is much more complicated. Would you expect superscalar to make the second term (CPS) better or worse?

worse - opposite of RISC

Mostly tautology = -3 ✓

2. Given a non-pipelined and pipelined design, both shown below, answer the following questions. *30pts*



8pts a. Compute the cycle time and latency of both designs.

Non-pipe: $1+6+4+8+1+8+5 = 33\text{ ns}$ latency/cycle time

Pipe: 10 ns cycle time

$6 \times 10 = 60\text{ ns}$ latency

Math err: -2
2pts/answer

7pts b. Compute the speedup of the pipelined design relative to the non-pipelined design.

$$\text{Speedup} = \frac{33}{10} = 3.3$$

- 8 pts
4 pts/7. c. In the pipelined design, how many stages have a delay that is less than the cycle time? What is the average stage delay?
- ↓
4 stages
- ↓
 $\frac{44}{6} = \frac{22}{3} = 7\frac{1}{3} = 7.33 \text{ ns}$

- 7 pts d. Can pipeline stages be removed from the pipelined design *without increasing the clock-cycle time*? If so, which stages?
- ↓
No!

30 pts 3. Consider a processor with a 32-bit virtual address and a 32-bit physical address.

Level 1 Cache: 64kByte, 16 byte cache line size, 1% miss rate, 1 ns access time

Level 2 Cache: NONE

RAM: 15 ns access time

TLB: 4 kByte page size

The Level-1 cache and TLB are accessed in parallel.

- 6 a. Is the L-1 cache addressed with virtual or physical addresses?
- virtual

- 6 b. For the L-1 cache, how big are the offset, index, and tag fields?
- offset = 4 bits
index = 12 bits
tag = 16 bits

- 6 c. For the TLB, how big is the VPN (virtual page number)?
- 12-bit offset \Rightarrow 20-bit VPN

- 6 d. How many bits of address are needed to resolve cache aliasing?
- 4 bits

- 6 e. Calculate the average access time in ns.

$$\begin{array}{rcl} 99\% * 1 \text{ ns} & = & 0.99 \text{ ns} \\ 1\% * 15 \text{ ns} & = & 0.15 \text{ ns} \\ \hline & & 1.14 \text{ ns} \end{array}$$

Div $\div 2 = -4$

4. Consider a RISC processor with a 6-stage pipeline, and how to convert the processor from RISC to Superscalar.

20 pts

- § a. In changing to a processor that is 3-way superscalar (that is, a processor that tries to execute 3 instructions in parallel), would you expect the complexity to go up by less than a factor of 3, exactly a factor of 3, or more than a factor of 3?

more than 3

- § b. What types of buffers have to be added before and after the functional units, and what are the purposes of the buffers?

Reordering buffers -

before: dispatch when instr's are ready

after: retire in original instr. order

- § c. Since it is Superscalar, what has the responsibility for checking the instruction dependencies - the hardware or the compiler?

No purpose: - 3

hardware

- § d. If the processor were VLIW instead of Superscalar, which would have the responsibility for checking dependencies - hardware or compiler?

compiler