

# Pipelining - Ch. 6.1

Note Title

10/15/2008

1	2	3	4	5		$x_2$	$x_1$	$y_2$	$y_1$	P
I-Fetch	Decode	ALU	Mem	Reg Store		0	0	0	0	0
2ns	1ns	2ns	2ns	1ns	= 8ns	0	0	0	1	1
					$\frac{8ns}{instr}$					
					?					
					$\frac{instr}{sec}$					
						</				

In the limit how much does pipelining help?

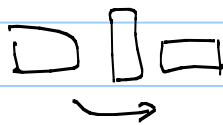
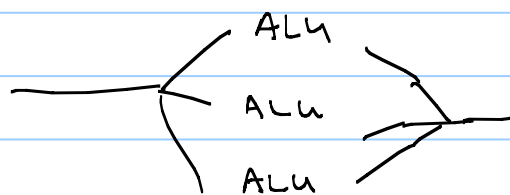
In the laundry: how often is a load completed?  
every 30 minutes

in the limit the speedup is  $\frac{\text{Time for a complete job}}{\text{time for one stage}} =$   
# of stages.  
Laundry Speedup  $\equiv 4$

What's wrong w/ this simple approximation?

- assumption all stages are equal

if not all equal:  $\frac{\text{Time for a complete job}}{\text{time for slowest stage}} = \text{speedup}$

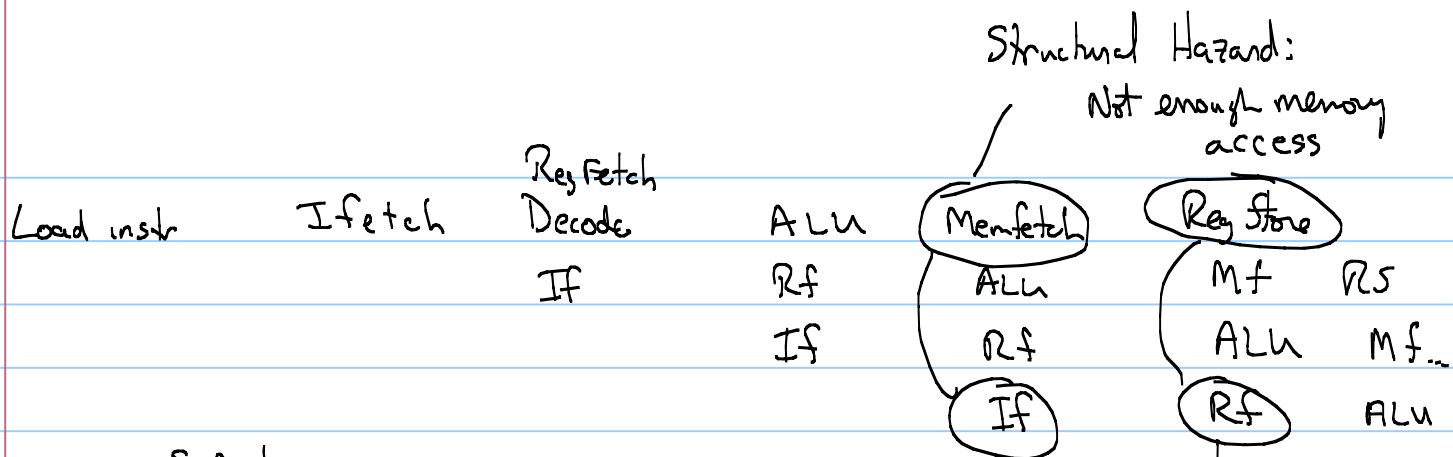


Time for latching and propagating from one  
Stage to the next.

## Pipeline

Hazards : situations that make a pipeline slow down.

- Structural - not enough hardware
  - Control - decision for a branch depends on previous result
  - Data - instruction depends on previous result.
- pipeline slowing down is called stalling;  
when stalled, bubbles are introduced.



### • Solutions

- Design memory to allow 2 fetches "at once."

Structural Hazard on Register Access

Characteristics of MIPS favorable to pipelining.

- all insts are same size and do about same amount of work.
- load/store architecture - operations are only on registers except for load and store

compare intel:

one operand from reg.

one from memory - compute effective address

store in memory